

RESEARCH

Open Access

An open-source rule-based syllabification tool for Brazilian Portuguese

Nelson Neto^{*}, Willian Rocha and Gleidson Sousa

Abstract

Background: The automatic syllabification process is an essential prerequisite for speech synthesis systems. However, the task is not trivial, and several techniques have been adopted over the last decade. Furthermore, while there are many public resources for some languages (e.g., English and Japanese), the resources for Brazilian Portuguese (BP) are still limited. This paper discusses ways to diminish this drawback, through the implementation of an open-source syllabification system for BP.

Methods: The proposed tool is based on published rule-based algorithms, with some new proposals, especially in the treatment of words with diphthongs and hiatus.

Results: Computer experiments were performed on a randomly chosen extract of the CETEN-Folha text corpus, and the results showed the percentage of correctly syllabified words of 99%.

Conclusions: A subjective evaluation was also conducted in order to compare the elaborated syllabification algorithm with the reference one within a text-to-speech system for BP. All developed codes and databases are publicly available.

Keywords: Automatic syllabification; Speech synthesis; Brazilian Portuguese

Background

Text-to-speech (TTS) is considered not just very innovative but also a very mature technology and an important tool to provide or increase the functional abilities of people with disabilities. It consists in converting natural language texts into synthesized speech [1]. The TTS procedure consists of two main phases. The first one is Natural Language Processing (NLP), also known as the front-end module, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is Digital Signal Processing (DSP), or the back-end module, where the acoustic output is produced from this phonetic and prosodic information. A simplified version of the procedure is presented in Figure 1.

The main goal of the NLP module is to produce the phonetic representation from an input text along with the prosody information. The NLP module is divided into three steps: the text normalization step, where numerals, special characters, abbreviations, and acronyms are

expanded into full words; the pronunciation analysis step, which typically assigns phonetic transcriptions, stress marks, syllable boundaries, and part-of-speech (POS) information to each word, including homographs, proper names, and foreign words; and the prosodic analysis step, where the prosodic features of speech are determined.

The NLP module informations (or labels) are used as input for the DSP module. The DSP module is language independent, and its objective is to produce synthesized speech. Although a few speech synthesis techniques exist, the approach wherein speech is synthesized through the selection and concatenation of natural speech waveform units has been largely applied [2], including for the Portuguese language [3-6]. Nevertheless, for this technique, the synthesis of voices with different styles and emotions as well as the obtainment of high quality itself requires the availability of large corpora.

A trainable approach in which the speech waveform is synthesized from parameters directly derived from hidden Markov models (HMMs) has been reported to work well for several languages, though the first version was implemented for Japanese [7]. The HMM-based synthesizers

^{*}Correspondence: nelsonneto@ufpa.br
Federal University of Pará, Augusto Corrêa, 1, Belém 66075-110, Brazil

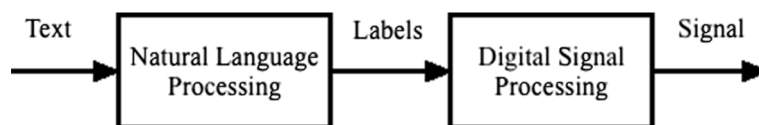


Figure 1 A simple diagram for TTS systems.

have been widely used because it is possible to obtain a good quality synthetic voice from a small database, and due to the fact that the voice features could be easily modified. In [8], the authors present topics related to the application of the HMM-based speech synthesis approach to Brazilian Portuguese (BP). The results obtained from [8] work are very relevant for BP and have encouraged another research groups expanding the linguistics and statistics knowledge. In 2008, for instance, Microsoft decided to develop an HMM-based BP synthetic voice, together with other languages for mobile and desktop interfaces [9].

Since nowadays the DSP is a stable module, to make a TTS system robust, efficient, and reliable, it is crucial to have a good language-dependent NLP module. In this paper, the syllabification front-end task is focused. Turning to the Portuguese language case, considerable advances have been reported concerning automatic syllabification for TTS systems. However, the location of the syllable boundaries is a problem of non-consensual resolution in Portuguese, mainly when the acoustic-phonetic constraints are taken into account [10]. Furthermore, most of the contributions given to the Portuguese language are not freely available.

In a more specific way, this work proposes an open-source syllabification tool for BP TTS systems that is based on a set of linguistic rules described in [11,12]. The motivations are to complement these previous initiatives and release resources of syllable transcription with stress vowel indication. The implemented system establishes a baseline and enables the comparison of results among research groups [13].

Aside from the description of the syllabification rules in details, the results of tests performed on parts of the CETEN-Folha text corpus [14] are also shown, in order to measure the designed algorithm effectiveness in treating words of long length and high complexity. A subjective experiment was also carried out in order to investigate the naturalness (or speech quality) added by the developed syllabification algorithm to a TTS system for BP [15].

This work is part of the FalaBrasil project [16], which aims at developing and deploying resources and tools for BP speech processing. Its public database allows one to establish baseline systems and reproduce results across different sites. Due to aspects such as the increasing importance of reproducible research, the FalaBrasil

project achieved good visibility and is now fomented by a very active open-source community.

Syllable and syllabification

The syllable is a unit relatively easy to identify and segmental if the splitting rules stipulated by the language orthography are followed. However, as a phonological unit, there is no consensus about its basic structure, as discussed in [17]. For most authors, a syllable is defined so that its nucleus, canonically a vowel, constitutes a peak in the curve of audibility that is preceded (onset) and/or followed (coda) by a sequence of segments (none or more consonants), with progressively decreasing sonority values. The nucleus and coda are sometimes lumped together to form what is called the rhyme. By applying these principles, the syllable is a speech unit of rhythmic organization, although other authors disagree, stating that the syllable should not be seen in parts but as a whole.

Syllabification means the determination of the place of syllable boundaries in a word. This is not a consensual procedure. Sometimes automatic syllabification methods deal more with a concept of 'syllable' that corresponds to the written form, while in other situations, they are correlated in some way with audibility, in an attempt to reconcile specific phonological aspects with technical requirements of the TTS systems [18]. For instance, Brazil congregates different accents, some are primarily syllable-timed (e.g., the 'paulista' accent), while others are stress-timed (e.g., the 'mineiro' accent). This has an impact on the quality of vowels and their deletion in pronunciation, which can certainly influence the syllabification process.

Adjacent vowel sounds also produce good examples of different pronunciations for the same Portuguese word [19]. In some cases, the pronunciation given to the vowels (faster or slower?) is uncertain and variable, a phenomenon which causes the double possibility of articulating the vocalic sequence both as rising diphthong (e.g., 'náusea' <náu-sea>, 'étereó' <é-te-reo>) and as hiatus (e.g., <náu-se-a>, <é-te-re-o>).

Recent studies emphasize the potential of using statistical information methods to increase specification of Portuguese syllables [10], but they are ongoing works, and a description of the syllable structure prototypes deriving from acoustic-phonetic properties is still required for the Portuguese language.

Evaluation of the syllabification task

In Portuguese case, it has been reported by several studies [20,21] that the syllabification task is essential in enhancing the quality of speech produced by synthesizers, since detecting the alternation between stressed and unstressed syllables will help in using them to model certain acoustic traits, like intensity and duration, in order to improve the synthesized speech intonation. According to [22], it is commonly accepted by the specialists of prosody models that the syllable is useful in the determination of prosodic parameters. The subjective experiments performed by [8] showed that the lack of features related to syllable stress information strongly degrades the quality of the synthesized speech.

There are many strategies that can be used to automatic syllabification. Choosing a method to splitting syllables depends mostly on the language, which is supposed to be applied for. For instance, the dictionary-based approach relies on keeping a large file containing all the words of a certain language and the corresponding linguistic informations (the use of such file eventually requires a large memory, which can be a problem, depending on the application). Another problem of using this method is that the lexical collection of any language is constantly evolving and TTS systems based on dictionaries need frequent updates.

Even now, there are few probabilistic approaches for Portuguese syllabification in the literature. The work by [23] introduces a method based on maximum entropy, while [24] proposes a new algorithm of syllabic splitting, which is based on the envelope of speech signals. In [25], the authors present a probabilistic neural network approach to predict the syllable boundaries as a mean to improve the performance of continuous speech recognition systems. The results achieved show that the application of syllable segmentation information in speech recognizers improves their overall performance, opening good perspectives for the use of this kind of information in other contexts.

Based on prior work focused in Portuguese, we conclude that most of the syllabification approaches are based on preestablished linguistic rules [26,27] or use weighted finite state transducers to automatically build these rules [28]. Rule-based systems are useful to aggregate new instances or to build new corpora for machine learning approaches. On the other hand, these systems are expensive while needing a linguist expert to setup all rules and exceptions needed to produce the results. Anyway, the use of rules is admittedly a reliable method to generate standard syllabification to Portuguese synthesis applications [29]. This interest comes from the observation that in Portuguese, the speech rhythm is regular and cadenced by syllables, which is considered the phonological basis of words.

However, it is important to note that very few studies reveal their algorithms and test corpora. As a consequence, authors often regret the impossibility to compare their results with another implementations, because they do not have the knowledge of other published researches about automatic syllabification with measured results. A motivation of this work is to complement these previous initiatives and release an open-source syllabification algorithm for BP TTS systems. The goal is to establish a baseline system and enable the comparison of results among research groups.

The remainder of the paper is organized as follows. 'Methods' section describes, briefly, the reference algorithms used in this work. 'A syllabification tool for BP' section presents the developed syllabification tool. A summary of the results is presented and discussed in 'Results and discussion' section. Finally, 'Conclusions' section summarizes our conclusions and addresses future works.

Methods

This paper extends the linguistic rules presented in [11,12]. The main idea of these algorithms is that all the syllables have a vowel as a nucleus, and this vowel can be surrounded by consonants, semi-vowels (or glides), or other vowels. Hence, one should locate the vowel that composes the syllable nuclei and isolate it from the other graphemes or not.

The algorithm proposed by [11] is based on orthography but also considers phonological preestablished criteria (e.g., the spelling double <r> is represented by a single phoneme [R] so is grouped and associated to only one potential syllable, like in the word 'carro' <ca-rro>). There is a set of 20 original rules and a hierarchical order of application is assumed. First, the more specific rules are considered until a general case (the last rule) is reached. The rules, as previously mentioned, consider the kind and the arrangement of graphemes to split the syllables of an input word. However, there are words, especially the ones with diphthongs (i.e., combinations of vowels and semi-vowels), whose syllabification is very difficult to perform correctly when we consider only these two criteria, since a number of very specific and well-elaborated rules would be required in order to deal with just a few examples.

In order to overcome this difficulty, the work developed in [12] added two new rules to the set proposed by [11] considering also the stress feature. The main motivation for analyzing the 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 [17]). Another point that ratifies the focus adopted in [12] is the fact that diphthongs, especially the ones with rising sonority, have

presented misconceptions in the syllabification performed by [11].

Then, the first rule deals with the vocalic sequences $\langle a, e, o + i, u \rangle$. If the second part of the sequence is the stressed vowel grapheme, then it belongs to another syllable (e.g., ‘contrair’ $\langle \text{con-tra-ir} \rangle$, ‘graúdo’ $\langle \text{gra-ú-do} \rangle$). The second one deals with diphthongs that varies with hiatus (glide + vowel). If the word contains this kind of sequence in the final position and the glide or the vowel is the stressed grapheme, then they must be separated (e.g., ‘tamanduá’ $\langle \text{ta-man-du-á} \rangle$). Finally, when the rising diphthong is not at the end of the word, the glide is always separated from the vowel (e.g., ‘bioma’ $\langle \text{bi-o-ma} \rangle$). Due to the fact that diphthongs and hiatus need this special treatment in their syllabification, it was defined that these rules must be incorporated before the original ones (see Figure 2).

In addition to that, the work by [12] improved the rule 19 [11]. If the analyzed vowel is not the first grapheme in the syllable to be formed and it is followed by another vowel that precedes a consonant, then the analyzed vowel must be separated from the following graphemes. This new version of the rule 19 fixes errors observed in the syllabification of words like ‘teólogo’, for example (the correct is $\langle \text{te-ó-lo-go} \rangle$, instead of $\langle \text{teó-lo-go} \rangle$, as shown in [11]).

The next section presents the developed syllabification algorithm with stress vowel determination and describes the baseline evaluation process.

A syllabification tool for BP

The proposed syllabification algorithm for BP is implemented by means of C# program language and is based on the 20 rules designed in [11], plus the two rules added by [12], including the stressed vowel brand and new improvements. All the rules are based on orthography and do not focus in any BP dialect. In fact, phonological criteria are also considered in this work, but only the classical ones, where the grapheme sequence is admittedly represented by a single phoneme. Figure 2 illustrates the architecture of the proposed system.

Each linguistic rule of the algorithm is basically composed of a condition to be evaluated and actions to be executed, considering that every syllable must have a vowel as a nucleus. The writing conventions used in the algorithm and the six possible syllabification actions are described in Tables 1 and 2, respectively.

Each condition evaluates all the graphemes that surround the syllable nucleus (the 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.

The algorithms on [11,12] were subject to a baseline evaluation process with 150,000 words (and their respective syllabification) extracted from the Dicio database [30]. Dicio is a dictionary of contemporary Portuguese, consisting of definitions, meanings, examples, and rhymes featuring over 400,000 words. During this phase, the analysis of the problems gave the input for some improvements on the rules proposed by [11]. First of all, the group composed by the rules 1 to 5 was analyzed, and the suggested modifications can be seen in Algorithm 1. This initial set of rules deals with syllables that begin with vowels.

Algorithm 1 Modifications proposed to the rules 2 to 4

Rule 2

```

if  $V = p^o$  and  $\hat{+}(1) = C$  and  $\hat{+}(2) = C$ 
  and ( $\hat{+}(3) = CO$  or  $\hat{+}(3) = CL$ ) then
  if  $\hat{+}(3) = CO$ 
    Case 5
  else
    Case 2
  end if
end if

```

Examples: *ins-tan-te*, *ex-tra-ção*

Rule 3

```

if  $V = p^o$  and  $\hat{+}(1) = G, \langle s \rangle, \langle r \rangle, \langle l \rangle, CN,$ 
   $\langle x \rangle$  and  $\hat{+}(2) = C$  and  $\hat{+}(2) \neq \langle s \rangle, \langle h \rangle,$ 
   $\langle r \rangle$  then
  if  $\hat{+}(1) = \langle x \rangle$  and  $\hat{+}(2) = \langle c \rangle$ 
    Case 1
  else
    Case 2
  end if
end if

```

Examples: *e-xce-to*, *e-xce-len-te*, *al-tu-ra*

Rule 4

```

if  $V = p^o$  and  $\hat{+}(1) = C$  and  $\hat{+}(2) = C$  and
   $\hat{+}(3) = V$  then
  if  $\hat{+}(1) = CM$ 
    Case 02
  else
    Case 01
  end if
end if

```

Examples: *ab-do-mi-nal*, *a-flo-rar*

It was added the check of liquid consonants to the conditional loop of rule 2, besides the original presence of stop consonants. This addition corrected the syllabification of few words, such as ‘adpresso’ $\langle \text{ad-pre-ss}o \rangle$ and ‘eplexia’

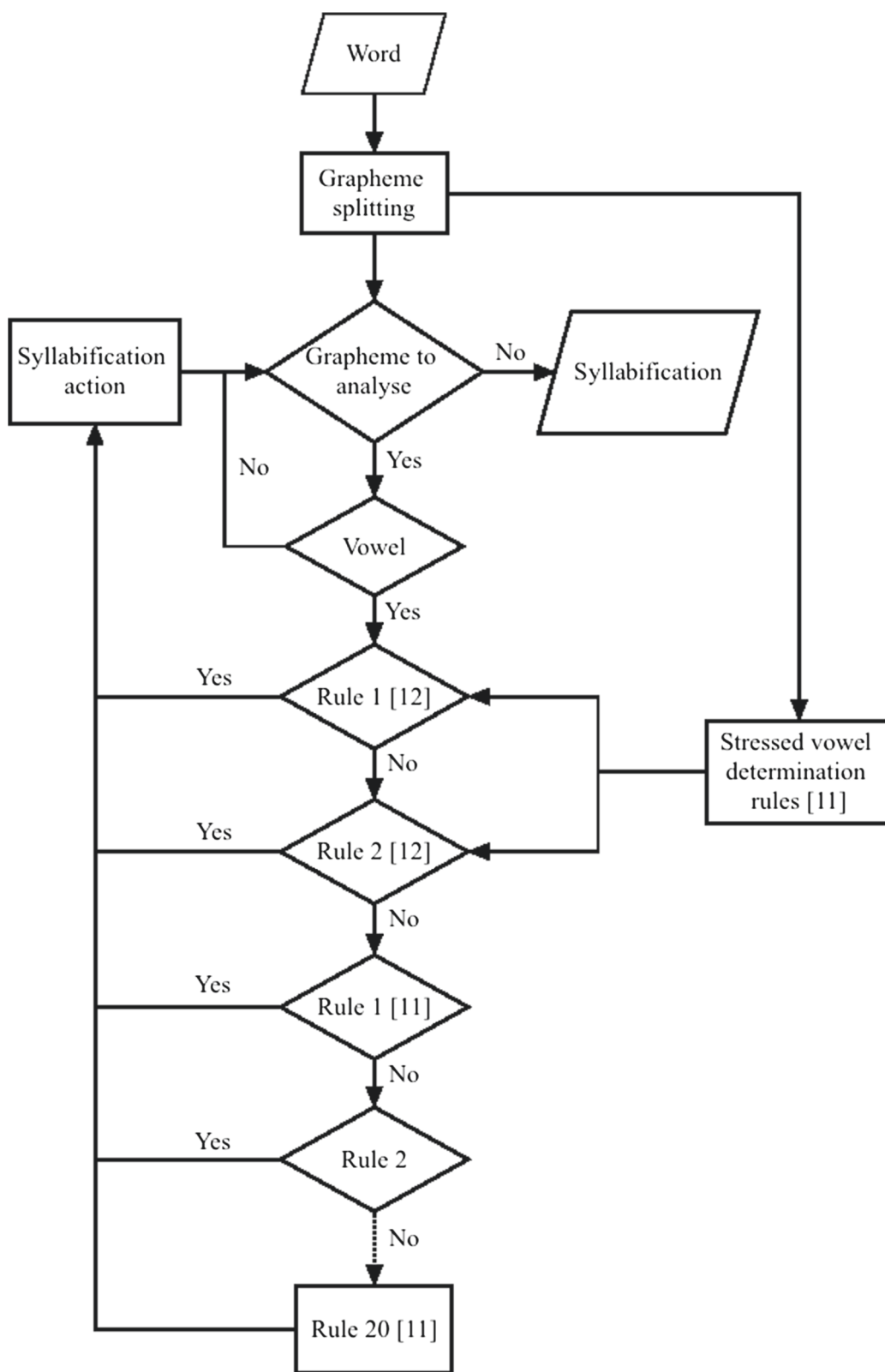


Figure 2 Block diagram of the developed syllabification tool. There is a set of splitting rules, and a hierarchical order of application is assumed, at the moment the algorithm identifies a vowel. First, the more specific rules are considered until a general case rule (rule 20) is reached, which restarts the process. Note that some rules proposed by [11] were improved during the research, like the rule 2, as described below.

Table 1 Annotation symbols and conventions used in the syllabification algorithm

Symbol	Meaning
V'	Vowel (a, e, o, á, é, í, ó, ú, â, ê, ô, ü)
G	Semi-vowel (i, u)
V	V', G
C	Any graphic consonant (< lh >, < nh >, CO, CF, CL, CN)
CO	Occlusive (p; t; c+a, o, u; qu+e, i; b; d; g+a, o, u; gu+e, i)
CF	Fricative (f; v; s; c+e, i; ç; z; ss; ch; j; g+e, i; x)
CL	Liquid (r, rr, l except < lh >)
CN	Nasal (m, n)
CM	Voiceless (b, g, p, c, d, f, t, not followed by l, r, t, V)
F	End of word
p ^o	Beginning of syllable
$\hat{\sim}(+n) = C$	n-grapheme on the right is equal to any graphic consonant
$\hat{\sim}(-n) = V$	n-grapheme on the left is equal to any graphic vowel
$\hat{\sim}(+n) \neq CN$	n-grapheme on the right is not equal to a nasal consonant

<ec-ple-xi-a>, which are not treated by the reference algorithm [11].

The modification made to the rule 3 reached words like 'exceto,' which the original syllabification follows the orthography (<ex-ce-to>). The letter <x> represents the most variable consonant sound in Portuguese. Its pronunciation depends on the letters before and after it. In this context, [19] suggested that the phonology conditional statement: < xc > → [s] is needed for the < xc > combination, to prevent other rules from forming a doubled sound [ss]. Therefore, the rule 3 was modified to keep the consonantal sequence < xc > in the same syllable (<e-xce-to>).

The rule 4 is a discussion of consonants not followed by vowel (or voiceless consonants). On the original rule, the voiceless consonant is joined to the next syllable (e.g., 'advogar' <a-dvo-gar>). Although the existence of an empty nucleus is hypothesized [19] (i.e., the word is

Table 2 Syllabification actions used in the algorithm

Case	Action
Case 1	V is separated from the next grapheme
Case 2	V is attached to the next grapheme and separated from the subsequent graphemes
Case 3	V is attached to the previous grapheme and separated from the subsequent graphemes
Case 4	V is attached to the previous and next graphemes, and separated from the subsequent graphemes
Case 5	V is attached to the next two graphemes and separated from the third grapheme
Case 6	V is attached to the previous grapheme and all the graphemes until the final position

in actually pronounced with an additional syllable, like 'advogar'), this work chose to follow the orthography and other studies [27] and kept the voiceless consonant in the same syllable (e.g., 'advogar' <ad-vo-gar>).

The group formed by the rules 6 to 20 treats the syllables that begin with consonants. A summary of the modified rules can be seen in Algorithm 2. Initially, it was observed that uncommon words were not correctly separated by the rule 13 (e.g., 'acampsia' <a-campsia>). Thus, the nasal consonants were added to the original condition in order to solve these faults. As a consequence, the exemplified syllabic splitting was updated to <a-camp-si-a>. A conditional statement was also added to treat hiatus formed by the grapheme sequence < ui >, like in the verb 'fluir' <flu-ir>).

Algorithm 2 Modifications proposed to the rules 13, 15, and 19

Rule 13

```

if V ≠ po and  $\hat{\sim}(-1) = C$  and  $\hat{\sim}(+1) = CL, CN, < i >$ 
then
  if  $\hat{\sim}(+2) = F$ 
    Case 6
  else if V = < u > and  $\hat{\sim}(+1) = < i >$ 
    Case 1
  if  $\hat{\sim}(+1) = CN$  or  $\hat{\sim}(+2) = < s >$ 
    Case 5
  else
    Case 2
  end if
end if
end if
Examples: dis-pen-sar, ru-im, trans-plan-te, hi-per-pnéi-a
    
```

Rule 15

```

if V ≠ po and  $\hat{\sim}(+1) = CO, < f >, < v >, < g >$  and
 $\hat{\sim}(+2) = CL, CO$  and  $\hat{\sim}(+3) = V$  then
  if  $\hat{\sim}(+1) = CM$ 
    Case 02
  else
    Case 01
  end if
end if
Examples: cap-tar, e-le-tro
    
```

Rule 19

```

if V ≠ po and  $\hat{\sim}(+1) = V$  and  $\hat{\sim}(+2) = C$  then
  if  $\hat{\sim}(+1) = V'$ 
    if  $\hat{\sim}(-1) = < g >, < q >$  and V = < u >
      Case 2
    else
      Case 1
    end if
  else if  $\hat{\sim}(+3) = V$ 
    Case 2
  else
    Case 1
  end if
end if
end if
Examples: san-guí-neo, se-me-ar
    
```

The proposed rule 15 follows the same logic applied to the rule 4 with respect to the voiceless consonants. For instance, the syllabic splitting defined for the word ‘captar’ is <cap-tar>, instead of <ca-ptar>, as presented in [11]. Finally, the rule 19 was improved to treat specific hiatus (vowel + vowel) not considered in the analyses made by [11,12]. Since this kind of vocalic sequence has a strong presence in the Portuguese lexicon, this modified rule is an important contribution of this work. Examples can be seen in ‘Evaluation of the syllabification tool’ section.

Regarding the algorithm for determining the stressed vowel, it is based on a set of rules described in [11]. The 19 original rules (including the general case) were implemented in hierarchical order and the output character (i.e., the stressed vowel) is used as input to the splitting rules designed by [12].

All the developed syllabification resources like source codes, libraries, dictionaries, test results, and log files are publicly available [31]. Having presented a summary of the proposed syllabification tool for BP, the next section discusses some results achieved.

Results and discussion

This section presents the baseline results. The experiments evaluate the current syllabification tool effectiveness and its influence in a TTS system performance.

Evaluation of the syllabification tool

The syllabification tool described in ‘A syllabification tool for BP’ section was compared with two other systems. The first one was based on the set of 20 rules proposed by [11], and the second system one followed the approach described in [12]. The three algorithms were chosen because they represent the evolution of our research in this topic, with implementing the algorithm proposed by [11] being the first attempt, which was followed by improving it.

The test corpus used to measure the effectiveness of each algorithm, considering orthographic criteria, was composed of 10,000 words randomly selected from the CETEN-Folha database [14], which is a text corpus extracted from *Folha de São Paulo*, a Brazilian newspaper, and contains around 24 million words.

The evaluation process was carried out in three stages. In the first one, only words contain vocalic sequences were used to evaluate the algorithms. Just verbs and adjectives were used in the second stage. No filter was applied in the last one. For each stage, four lists of thousand words, selected from the test corpus, were used to perform the experiments. The average was considered in the performance assessment.

The comparison between the different algorithms was performed in terms of correctly syllabified words (or word accuracy), which was obtained automatically, using the

Dicio database as reference. Since the syllabic splitting provided by the reference dictionary is based on orthography [30], intentional errors caused by adopted phonological criteria were not considered (e.g., ‘russo’ <ru-ss0>). The results are shown in Table 3.

Clearly, the rules proposed by [11] to process vocalic sequences achieved the worst performance. Most errors were due to hiatus and rising diphthongs. For instance, the syllable division <saí-da> performed by [11] to the word ‘saída’ is incorrect, because the word has a 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, and therefore, the accepted syllabification is <sa-í-da>. Another mistaken treatment was given to words like ‘criada’, for example, where the two vowel sounds occurring in adjacent syllables (<cri-a-da>), and not in the same syllable (<cria-da>), as defined by [11]. Due to the improvements proposed by [12], such inconsistencies were not observed in other two algorithms.

As expected, the modified rule 19 increased the performance of this current proposal and allowed it to outperform [12]. This rule correctly handled the hiatus present in the words ‘campeonato’ <cam-pe-o-na-to>, ‘joelho’ <jo-e-lho>, and ‘israelense’ <is-ra-e-len-se>, for example. In turn, [11,12] assumed these hiatus as diphthongs and mistakenly maintained them in the same syllable.

The current algorithm presents errors in words containing vocalic sequences in the final position. For instance, the word ‘euforia’ presents two vowels (<ia>) that should be separated to compose nuclei for two different syllables (<eu-fo-ri-a>), but the three algorithms keep the vowels in the same syllable (<eu-fo-ria>). This type of mistake is caused by inconsistencies in determining the stressed vowel and is subject of ongoing research. Faults are still observed in the vocalic sequence <ui>, despite the changes made in the rule 13 (e.g., ‘constituição’ <consti-tui-ção>). Lastly, errors due to prefixation are noticed, like in the word ‘teleinformática’ <te-lei-nfor-má-ti-ca>.

Regarding the second stage of tests, the CETEN-Folha corpus is suitable for this purpose since all sentences are morphologically annotated in terms of POS tags, from which it is easy to obtain adjectives and verb forms for

Table 3 Experimental results of the syllabification algorithms against words with vocalic sequences (stage 1), verbs and adjectives (stage 2), and general words (stage 3)

Algorithm	Stage 1	Stage 2	Stage 3
Silva et al. 2008 [11]	68.90	83.20	78.12
Monte et al. 2011 [12]	93.80	96.20	94.90
Current proposal	98.80	99.55	99.05

It was considered the average of the word accuracy (%) of the four word lists created for each stage.

Table 4 Set of sentences used to evaluate the syllabification algorithms working on a TTS system

Number	Sentence
1	sandra regina machado: acho que ela enfim criou juízo.
2	no total, sete mísseis foram disparados contra o encrave.
3	em florianópolis, foi registrado dois graus celsius na manhã de domingo.
4	as situações ditas embaraçosas são resolvidas com os dados.
5	conseguiram eliminar áreas supérfluas ou que antes eram desperdiçadas.
6	uma lata de leite em pó integral vale um ingresso.
7	a maioria dos passageiros do barco naufragado era de crianças.
8	a provável causa do acidente foi excesso de lotação a bordo.
9	a secretaria estadual de saúde distribuirá cem mil preservativos no carnaval.
10	são essas qualidades que inspiraram o plano real desde a sua criação.

the same lemma. The initial idea was to study the conjugated forms of the verbs, since it contains cases of vowels' sequences that is an important context to evaluate for syllabification [32]. But the reference dictionary gives essentially the infinitive forms and few examples in the first and third person. Thus, a manual check would be required to evaluate the conjugated form syllabification, which was not done in this work.

On the other hand, an extra test was performed on 776 verbs (also present in the reference dictionary). The word accuracy was 94.58% and 96.90% for [11,12], respectively, and the errors still have focused on vocalic sequences, like in the words 'viola' <vio-la> and 'realizar' <rea-lizar>. The proposed algorithm diverged from the dictionary only in the word 'oro.' The problem is that Dicio erroneously considered such conjugated verb as a monosyllable at the time of testing. However, a recent update

fixed this fault (i.e., the actual syllabification proposed by Dicio is <o-ro>, as well as in this work).

Finally, the third experiment was carried out without any filter, and the proposed algorithm achieved again the best result. The errors remained concentrated in vocalic sequences, adding failures caused by foreign words and acronyms.

Using the syllabification tool within a TTS system

This experiment evaluates the designed syllabification tool working on an open-source TTS system for BP [15]. The chosen system uses the MARY framework [33] to build HMM-based acoustic models (or voices).

In order to built HMM-based voices, one needs a labeled corpus with transcribed speech. So, the speech training corpus used in this work consists of 1,000 phonetically balanced phrases [34] recorded by a man speaker, corresponding to approximately 1.6 h of audio. This well-known set of sentences was obtained from CETEN-Folha through a genetic algorithm, seeking to minimize the number of speech synthesis units (triphones) not present in the collection.

Then, to evaluate the influence of the syllabification rules on the synthesized speech quality, the TTS system was trained under the conditions of syllable information provided by the proposed algorithm and the other two algorithms, separately. In other words, three HMM-based voices for BP were built, and the only difference between them was the syllable division.

In the sequel, ten sentences were randomly extracted from the training corpus to be used in the test stage. The search space was the first subset of 20 phrases described in [34]. The sentences are shown in Table 4. Each test sentence was then synthesized into the three HMM-based voices. Finally, the 30 audio files were randomly played, and the listener had to give a score for the speech quality, from bad (1) to excellent (5),

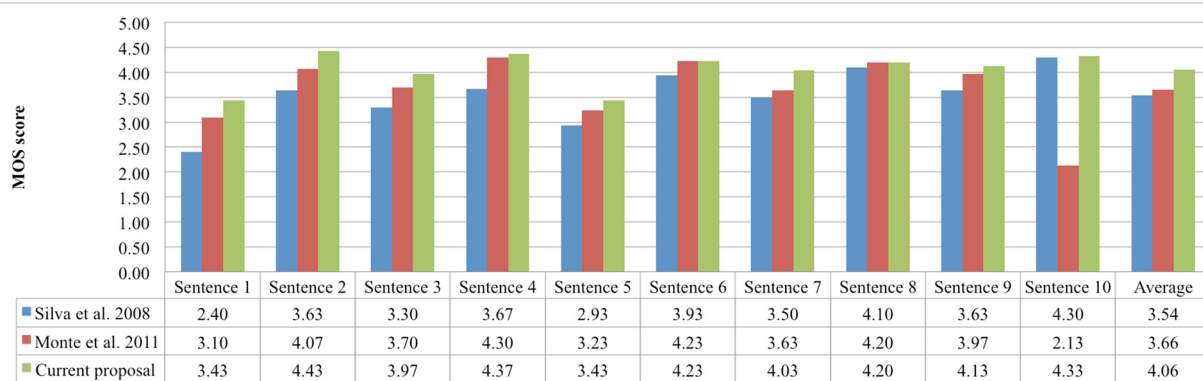


Figure 3 Overall result of the speech quality test comparing the syllabification algorithms within a TTS system. It was considered the average of the MOS score for each built HMM-based voice.

according to the mean opinion score (MOS) protocol [35].

A total of 30 Brazilian subjects participated in the test. Since the intention was to evaluate the overall quality of the synthesized speech from the viewpoint of the general user, the chosen listeners had no training and were not familiarized with the speech processing area. The results are shown in Figure 3.

The sentences 1 and 2 were classified as slightly annoying (i.e., a value of 3.0 to 3.5) in the current proposal. The phrase 1 may have sounded without subject-verb agreement due to lack of intonation given by the voices with respect to colon. In the phrase 2, we realized that many listeners did not understand the word 'supérfluas,' which certainly has an impact on the score. Regarding the outlier in the sentence 10, there is no apparent cause for this disorder.

As expected, the syllable boundaries influence the quality of the synthesized speech, and the algorithms proposed by [11,12] had almost the same performance, while both were outperformed by the current proposal in average. In addition to the empirical improvement on the speech quality, this work had also contributed to the development of an open-source syllabification tool that can be easily incorporated to any TTS system for BP.

Conclusions

The description of a syllabification tool with its corresponding characteristics was performed in this paper. A set of rules that marks the stressed vowel was also presented. In fact, after making available resources [31], the goal is to establish a baseline system and enable the comparison of results among research groups. According to tests carried out on the CETEN-Folha database, it was shown that application of rules is a good method for BP automatic syllabification. It can keep up with new words and does not need large computational resources. Aiming at investigating the value of the information input to a TTS system, it was verified, according to some subjective experiments, that the determination of the syllable boundaries represents an important issue in order to achieve synthesized speech with good quality. Also, according to a MOS test performed with listeners not familiarized with the speech processing area, the syllabification tool in question performed well when compared to previous researches. Future work include:

- Improving the stress determination rules;
- Checking the feasibility of implementing new types of syllable structures based on a phonetic and acoustic perspective only [10];
- Comparing the proposed rule-based algorithm with some machine learning approaches, since this work releases a big dictionary;
- Verifying the new orthographic form of Portuguese. Would the new one represent any difference in the proposed syllabification splitting? It seems to be very likely, since the graphic accent rules were modified, and some words are no longer marked.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors have contributed to the different methodological and experimental aspects of the research. All authors read and approved the final manuscript.

Acknowledgements

This work was financially supported by the Federal University of Pará (UFPA), Brazil, project no. 07/2013 - PROPEP.

Received: 1 May 2014 Accepted: 5 December 2014

Published online: 28 January 2015

References

1. Taylor P (2009) Text-to-speech synthesis. Cambridge University Press, New York
2. Hunt A, Black A (1996) Unit selection in a concatenative speech synthesis system using a large speech database. In: IEEE international conference on acoustics, speech, and signal processing (ICASSP), IEEE, Atlanta, GA. pp 373–376
3. Simões F, Violaro F, Barbosa P, Albano E (2000) Um sistema de conversão texto-fala para o Português falado no Brasil. *Revista da Sociedade Brasileira de Telecomunicações* 15:70–77
4. Nicodem M, Kafka S, Seara Junior R, Seara R (2007) Refinamento da segmentação fonética em aplicações de síntese de fala. In: XXV simpósio brasileiro de telecomunicações, pp 1–6
5. Freitas D, Braga D (2002) Towards an intonation module for a Portuguese TTS system. In: 7th international conference on spoken language processing (ICSLP), pp 161–164
6. Carvalho P, Trancoso I, Oliveira L (2003) WFST based unit selection for concatenative speech synthesis in European Portuguese. In: 15th international congress of phonetic sciences, pp 2333–2336
7. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: 6th European conference on speech communication and technology, pp 2347–2350
8. Maia R, Zen H, Tokuda K, Kitamura T, Resende F (2006) An HMM-based Brazilian Portuguese speech synthesizer and its characteristics. *J Commun Inf Syst* 21:58–71
9. Braga D, Silva P, Ribeiro M, Henriques M, Dias M (2008) HMM-based Brazilian Portuguese TTS. In: PROPOR special session: Applications of Portuguese Speech and Language Technologies
10. Candeias S, Perdigão F (2011) Investigating new syllables prototypes for the Portuguese language. In: 17th international congress of phonetic sciences, pp 388–391
11. Silva D, Braga D, Resende Jr. F (2008) Separação das sílabas e determinação da tonicidade no Português Brasileiro. In: XXVI simpósio brasileiro de telecomunicações
12. Monte A, Ribeiro D, Neto N, Cruz R, Klautau A (2011) A rule-based syllabification algorithm with stress determination for Brazilian Portuguese natural language processing. In: 17th international congress of phonetic sciences, pp 1418–1421
13. Vandewalle P, Kovacevic J, Vetterli M (2009) Reproducible research in signal processing - what, why, and how. *IEEE Signal Process Mag* 26:37–47
14. Corpus de Textos Eletrônicos NILC/Folha de S. Paulo (2014) Núcleo Interinstitucional de Linguística Computacional, São Carlos, SP. www.linguateca.pt/cetenfolha/. Accessed 10 April 2014
15. Couto I, Neto N, Tadaiesky V, Klautau A, Maia R (2010) An open source HMM-based text-to-speech system for Brazilian Portuguese. In: 7th international telecommunications symposium

16. Projeto FalaBrasil (2014) Laboratório de Processamento de Sinais da Universidade Federal do Pará, Belém, PA. www.laps.ufpa.br/falabrasil. Accessed 10 April 2014
17. Collischonn G (2005) A Sílabas em Português. In: BISOL, Leda (org.). *Introdução a, Estudos de Fonologia do Português Brasileiro*. EDIPUCRS, Porto Alegre, pp 95–126
18. Oliveira C, Moutinho LC, Teixeira A (2006) On automatic European Portuguese syllabification. In: *III congresso internacional de fonética experimental*, pp 1–11
19. Faria A (2003) *Applied phonetics: Portuguese text-to-speech*. Technical Report, University of California, Berkeley
20. Madureira S, Barbosa P, Fontes M, Spina D, Crispim F (1999) Post-stressed syllables in Brazilian Portuguese as markers. In: *XIV international congress of phonetic sciences*, pp 917–920
21. Seara Jr. R, Kafka S, Seara I, Pacheco F, Klein S, Seara R (2004) Parâmetros linguísticos utilizados para a geração automática de prosódia em sistemas de síntese de fala. In: *XXI simpósio brasileiro de telecomunicações*, pp 1–6
22. Teixeira JP (2004) *A prosody model to TTS systems*. PhD Thesis, Faculdade de Engenharia da Universidade do Porto
23. Barros MJ, Weiss C (2006) Maximum entropy motivated grapheme-to-phoneme, stress and syllable boundary prediction for Portuguese text-to-speech. In: *IV jornadas en tecnologías del habla*, pp 177–182
24. Silva EL, Oliveira HM (2012) Implementação de um algoritmo de divisão silábica automática para arquivos de fala na língua portuguesa. In: *XIX congresso brasileiro de automática*, pp 4161–4166
25. Meinedo H, Neto J, Almeida L (1999) Syllable onset detection applied to the Portuguese language. In: *6th European conference on speech communication and technology*, pp 5–9
26. Gouveia P, Teixeira JP, Freitas D (2000) Divisão silábica automática do texto escrito e falado. In: *V encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR)*, pp 65–74
27. Braga D, Resende F Jr. (2007) Módulos de processamento de texto baseados em regras para sistemas de conversão texto-fala em Português Europeu. In: *XXI encontro da associação portuguesa de linguística*, pp 141–156
28. Oliveira C, Moutinho LC, Teixeira A (2005) On European Portuguese automatic syllabification. In: *9th European conference on speech communication and technology*, pp 2933–2936
29. Braga D, Freitas D, Ferreira H (2003) Processamento linguístico aplicado à síntese de fala. In: *3th congresso luso-moçambicano de engenharia*, pp 1–12
30. *Dicionário Online de Português* (2014) 7Graus, Porto, Portugal. www.dicio.com.br. Accessed 10 April 2014
31. Projeto FalaBrasil - Downloads (2014) Implementação de um separador silábico gratuito para o Português Brasileiro. www.laps.ufpa.br/falabrasil/files/separador_silabico.tar.gz. Accessed 10 April 2014
32. Marquiável V, Shulby C, Veiga A, Proença J, Candeias S, Perdigão F (2014) Rule-based algorithms for automatic pronunciation of Portuguese verbal inflections. In: *11th international conference, PROPOR*. Springer International, Publishing Switzerland, pp 36–47
33. Schroder M, Trouvain J (2001) The German text-to-speech synthesis system MARY: a tool for research, development and teaching. *Int J Speech Technol* 6:365–377
34. Cirigliano R, Monteiro C, Barbosa F, Resende F, Couto L, Morais J (2005) Um conjunto de 1000 frases para o Português Brasileiro obtido utilizando a abordagem de algoritmos genéticos. In: *XXII simpósio brasileiro de telecomunicações*, pp 110–114
35. Jobson R (2014) *Methods to objectively evaluate speech quality*. Technical Report, Teraquant Corporation. www.teraquant.com. Accessed 10 April 2014

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
