

Implementação de um Separador Silábico Gratuito Baseado em Regras Linguísticas para o Português Brasileiro

Willian Rocha¹, Nelson Neto¹

¹ Laboratório de Processamento de Sinais - LaPS
Universidade Federal do Pará - UFPA
Rua Augusto Correia, 1 – 660750-110 – Belém, PA, Brasil

willian.rocha@itec.ufpa.br, nelsonneto@ufpa.br

Abstract. *The automatic syllabification process is an important prerequisite for speech synthesis systems. However, the task is not trivial and several techniques have been adopted over the last decade. Furthermore, there is a few number of researches and public resources dedicated to Brazilian Portuguese compared to another languages. This paper discusses efforts to reduce this disabilities, implementing and testing a syllabification algorithm based on linguistic rules. All developed codes and databases will be publicly available.*

Resumo. *A divisão automática das palavras em sílabas é um importante pré-requisito para sistemas que envolvem síntese de voz. Contudo, a tarefa não é trivial e diversas técnicas vêm sendo adotadas ao longo da última década. Além disso, são poucos os estudos na área e recursos públicos dedicados ao Português Brasileiro quando comparado a outras línguas. Este trabalho discute esforços para reduzir essa deficiência, implementado e testando um separador silábico baseado em regras linguísticas. Todos os códigos e bases de dados desenvolvidos serão livremente disponibilizados.*

1. Introdução

A síntese de voz (TTS, de “text-to-speech”) consiste em produzir a fala humana artificialmente, através da geração automática do sinal de voz a partir de texto. Apesar dos conhecidos problemas, como a falta de ritmo da fala, a geração de voz sintetizada é vista, atualmente, como uma tecnologia madura e um importante instrumento para a comunicação natural humano-computador [Taylor 2009]. A mesma é capaz de aumentar o número de cidadãos com acesso às tecnologias de informação, em especial as pessoas que possuem algum tipo de limitação física ou necessidade especial.

Os sistemas TTS são compostos por módulos de *software* que convertem textos escritos em linguagem natural em voz sintetizada que seja inteligível e mais natural possível. A tarefa dos sistemas TTS é bastante complexa, visto que envolve a imitação (*mimicking*) de como os seres humanos realizam a leitura de um texto [Dutoit 2001]. Por isso, é fundamental a presença de um módulo responsável pelo pré-processamento da língua (NLP, de “natural language processing”). Esse módulo realiza um conjunto de tarefas que atribuem transcrição fonética à cada palavra e marcam o texto com informações prosódicas.

Para qualquer língua, é essencial identificar corretamente a unidade silábica quando tratamos de sistemas TTS, tanto para transcrição fonética, como para a geração de prosódia, já que na sílaba tônica há um aumento de duração e intensidade. E para a língua

portuguesa não é diferente. Verificou-se que tanto as principais propostas de separadores silábicos para o Português Europeu (PE) [Gouveia et al. 2000, Oliveira et al. 2007, Braga e Jr. 2007], como para o Português Brasileiro (PB) [Madureira et al. 1999, Jr. et al. 2004, Silva et al. 2008], relatam que só conhecendo a alternância entre as sílabas tônicas e átonas nas palavras é possível ditar o ritmo da fala em sistemas TTS.

Com exceção de [Barros e Weiss 2006], que optou por uma abordagem em máxima entropia, e [da Silva e de Oliveira 2012], que propôs um método de divisão silábica automático a partir de arquivos de fala com base na envoltória do sinal, em todos os sistemas de separação silábica estudados seguiu-se uma abordagem linguística, ora usando um conjunto de regras pré-definidas, ora usando transdutores de estados finitos para a geração automática dessas regras. É consenso que a opção por regras linguísticas se baseia na regularidade no ritmo da fala existente na língua portuguesa, cadenciado pelas sílabas, que são a base para a formação das palavras no nosso idioma. Importante salientar que poucos autores revelam seus algoritmos e ainda menos publicam os resultados de testes de desempenho comparativamente com outros sistemas.

Esta atual proposta contribui com um programa *open-source* baseado nas regras fonológicas descritas em [Silva et al. 2008] para divisão silábica em PB. Melhorias às regras originais também são propostas, pois, apesar de apresentarem uma excelente taxa de acerto, não se mostraram suficientes para resolver todos os casos, principalmente no trato dos encontros vocálicos. Em seguida, o desempenho do separador silábico proposto é avaliado e comparado com outros algoritmos e um dicionário disponível na Internet. Com a disponibilização gratuita dos códigos e bases de dados [Fal 2013] implementados, pretende-se estimular a cooperação e a reprodução de resultados entre diferentes grupos de pesquisa.

Este trabalho está organizado da seguinte maneira. A Seção 2 apresenta as estratégias utilizadas na implementação do separador silábico. A Seção 3 descreve os testes realizados e avalia os resultados obtidos. A Seção 4 conclui e sugere pesquisas futuras.

2. O separador silábico proposto

Comprovadamente, o emprego de regras linguísticas é uma boa escolha para sistemas de conversão texto-fala, considerando que o Português é uma língua com bastante regularidade fonológica [Martelotta 2008]. Além disso, a solução para separação silábica por regras apresenta duas vantagens em relação, por exemplo, à solução por dicionário, que consiste na criação de uma tabela contendo a palavra e a sua respectiva separação (*hash table*): pouca utilização de memória e a capacidade de sempre poder ler uma nova palavra [Braga e Jr. 2007]. A seguir serão apresentadas a arquitetura e as regras usadas no separador silábico proposto neste trabalho.

2.1. Arquitetura do algoritmo

O separador silábico proposto é uma codificação em C# das 20 regras fonológicas descritas em [Silva et al. 2008], com algumas modificações. Sua ideia principal é que todas as sílabas possuem uma vogal como núcleo, que pode ser cercado por consoantes ou outras vogais (semi-vogais ou glide). Então, é possível localizar a vogal que compõe o núcleo da sílaba e isolar as consoantes e as semi-vogais. A simbologia utilizada na descrição das regras é apresentada na Tabela 1.

Tabela 1. Simbologia usada no algoritmo de separação silábica.

Símbolo	Significado
V'	Vogal (a, e, o, á, é, í, ó, ú, â, ê, ô, ü)
G	Semi-vogal (i, u)
V	V', G
C	Consoante (<lh>, <nh>, CO, CF, CL, CN)
CO	Oclusivas (p, t, c+a, o, u, qu+e, i, b, d, g+a, o, u, gu+e, i)
CF	Fricativas (f, v, s, c+e, i, ç, z, ss, ch, j, g+e, i, x)
CL	Líquida (l, r, rr, exceto <lh>)
CN	Nasal (m, n)
CM	Mudas (b, g, p, c, d, f, t; não seguidas de (l, r, t, V'))
F	Fim da palavra
p^o	Início da sílaba
$\hat{(+n)} = C$	Grafema n à direita igual a uma consoante qualquer
$\hat{(-n)} = V$	Grafema n à esquerda igual a uma vogal qualquer
$\hat{(+n)} \neq CN$	Grafema n à direita não é uma consoante nasal

Tabela 2. Casos e ações considerados.

Caso	Ação
Caso 1	V é separada do grafema seguinte
Caso 2	V é unida ao próximo grafema da direita e é separada dos subsequentes
Caso 3	V é unida ao grafema anterior e é separado dos seguintes
Caso 4	V é unida ao grafema anterior e ao seguinte e é separada dos subsequentes
Caso 5	V é unida aos dois grafemas seguintes e é separado do terceiro grafema
Caso 6	V é unida ao grafema anterior e a todos os grafemas até o final da palavra

Na arquitetura do algoritmo as regras são dispostas hierarquicamente desde a primeira até a penúltima, sendo a última regra o caso geral que finaliza a análise. As regras são baseadas na busca das vogais existentes em cada palavra, seguida da análise dos caracteres existentes à esquerda e à direita, para decidir que ação tomar. Como as vogais são a base da sílaba, a ação tomada é no sentido de unir, ou não, a referida vogal ao conjunto de grafemas que a cerca. O conjunto de ações está listado na Tabela 2.

O conjunto de regras proposto em [Silva et al. 2008] não considera a análise da vogal tônica no processo de separação silábica, apesar de existem palavras cuja divisão silábica torna-se bastante complicada (e muitas vezes falha) sem a análise de tonicidade, especialmente na presença de encontros vocálicos. Para contornar essa dificuldade, [Monte et al. 2011] propôs duas regras com verificação de vogal tônica que foram adicionadas ao conjunto original de [Silva et al. 2008]. A primeira aborda os ditongos decrescentes (“vogal + semi-vogal”), já a segunda trata dos ditongos que variam como hiato (“semi-vogal + vogal”). Com isso, [Monte et al. 2011] apresentou uma melhora significada na separação silábica de encontros vocálicos em comparação a [Silva et al. 2008]. As regras extras de [Monte et al. 2011] também foram usadas no algoritmo proposto.

2.2. Alterações nas regras propostas em [Silva et al. 2008]

Os testes de validação, realizados com 150.000 palavras da base de dados do Dicionário Online de Português [dic 2013], mostraram a necessidade de alteração em algumas regras propostas em [Silva et al. 2008]. Primeiramente, analisou-se as Regras 1 a 5, que tratam sílabas iniciadas com vogais, e as modificações sugeridas podem ser vistas na Tabela 3.

Tabela 3. Regras utilizadas na separação das sílabas (2, 3 e 4).

<p>Regra 2</p> <p>if $V = p^o$ and $\hat{(+1)} = C$ and $\hat{(+2)} = C$ and $(\hat{(+3)} = CO$ or $\hat{(+3)} = CL)$ then</p> <p style="padding-left: 2em;">if $\hat{(+3)} = CO$</p> <p style="padding-left: 4em;">Caso 5</p> <p style="padding-left: 2em;">else</p> <p style="padding-left: 4em;">Caso 2</p> <p style="padding-left: 2em;">end if</p> <p>end if</p> <p>Exemplo: <i>ins-cri-ção, ex-tra-ção</i></p>
<p>Regra 3</p> <p>if $V = p^o$ and $\hat{(+1)} = G, < s >, < r >, < l >, CN, < x >$ and $\hat{(+2)} = C$ and $\hat{(+2)} \neq < s >, < h >, < r >$ then</p> <p style="padding-left: 2em;">if $\hat{(+2)} = < s >$ and $\hat{(+3)} = F$</p> <p style="padding-left: 4em;">Caso 6</p> <p style="padding-left: 2em;">else if $\hat{(+1)} = < x >$ and $\hat{(+2)} = < c >$</p> <p style="padding-left: 4em;">Caso 1</p> <p style="padding-left: 2em;">else</p> <p style="padding-left: 4em;">Caso 2</p> <p style="padding-left: 2em;">end if</p> <p>end if</p> <p>Exemplo: <i>e-xce-to, in-su-ce-ssô</i></p>
<p>Regra 4</p> <p>if $V = p^o$ and $\hat{(+1)} = C$ and $\hat{(+2)} = C$ and $\hat{(+3)} = V$ then</p> <p style="padding-left: 2em;">if $\hat{(+1)} = CM$</p> <p style="padding-left: 4em;">Caso 02</p> <p style="padding-left: 2em;">else</p> <p style="padding-left: 4em;">Caso 01</p> <p style="padding-left: 2em;">end if</p> <p>end if</p> <p>Exemplo: <i>ab-do-mi-nal, di-a-gra-ma</i></p>

Na Regra 2, adicionou-se a verificação de consoantes líquidas no laço condicional, além das originalmente presentes consoantes oclusivas. Essa modificação corrigiu a separação silábica de um restrito conjunto de palavras, como “adpresso” (*ad-pre-ssô*) e “ecplexia” (*ec-ple-xi-a*), que no algoritmo de referência não recebiam tratamento.

As modificações inseridas na Regra 3 atingiram palavras como “exceto”, cuja separação original segue a ortografia (*ex-ce-to*). Contudo, entendeu-se que a fonologia deveria prevalecer nesses casos, com o encontro consonantal “xc” permanecendo na mesma sílaba por emitir um único som (*e-xce-to*). Esse tipo de correção é importante para atender as necessidades técnicas subjacentes ao desenvolvimento de um sistema TTS. Essa característica é seguida por [Silva et al. 2008] em outras regras, como na palavra “arrendar” (*a-rren-dar*) tratada na Regra 5.

A Regra 4 recai na discussão sobre as consoantes sem vogal na sílaba (ou “mudas”). Na regra original essas consoantes são unidas a sílaba adjacente, como na palavra “advogar” (*a-dvo-gar*), por exemplo. Já este trabalho optou por seguir a ortografia e outras referências [Braga e Jr. 2007], ou seja, a divisão silábica foi alterada para “advogar” (*ad-vo-gar*).

Para as Regras 6 a 20, que tratam as sílabas iniciadas em consoantes, também

foram propostas algumas modificações (vide Tabela 4). Nos testes de validação observou-se que palavras inusuais da língua portuguesa obedeciam a condição de entrada da Regra 13, no entanto, nenhuma ação era tomada dentro da mesma, por exemplo, na palavra “acampsia” (*a-campsia*). Para resolver esses casos, incluiu-se outras consoantes típicas de final de palavra, como a consoante “r” e as consoantes nasais, além da consoante “s” originalmente posta. Assim, a divisão silábica da palavra tomada como exemplo foi corrigida para “acampsia” (*a-camp-si-a*). Já na Regra 15, adicionou-se a condição de consoante sem vogal na sílaba, seguindo a mesma linha de raciocínio apresentada na Regra 4. Como exemplo de alteração, podemos citar a palavra “captar” (*ca-ptar*), cuja divisão silábica foi modificada para (*cap-tar*).

Tabela 4. Regras utilizadas na separação das sílabas (13, 15 e 19).

Regra 13

if $V \neq p^o$ and $\hat{(-1)} = C$ and $\hat{(+1)} = CL, CN, < i >$ **then**
 if $\hat{(+2)} = F$
 Caso 6
 else if $\hat{(+1)} = CM, CN$ or $\hat{(+2)} = < s >, < r >, CN$
 Caso 5
 else
 Caso 2
 end if

end if

Exemplo: *dis-pen-sar, cir-cuns-pec-to, quar-tzo*

Regra 15

if $V \neq p^o$ and $\hat{(+1)} = CO, < f >, < v >, < g >$ and $\hat{(+2)} = CL, CO$
 and $\hat{(+3)} = V$ **then**
 if $\hat{(+1)} = CM$
 Caso 02
 else
 Caso 01
 end if

end if

Exemplo: *cap-tar, e-le-tro*

Regra 19

if $V \neq p^o$ and $\hat{(+1)} = V$ and $\hat{(+2)} = C$ **then**
 if $\hat{(+1)} = V'$
 if $\hat{(-1)} = < g >, < q >$ and $V = < u >$
 Caso 5
 else
 Caso 1
 end fi
 else if $\hat{(+3)} = V$
 Caso 2
 else
 Caso 1
 end fi

end if

Exemplo: *san-guí-neo, se-me-ar*

Finalmente, a Regra 19 foi modificada para tratar encontros vocálicos, especificamente hiatos, até então não considerados na análise feita pelas regras elaboradas em [Silva et al. 2008, Monte et al. 2011]. Essa é uma contribuição importante deste tra-

balho, dada a presença marcante desses agrupamentos de vogais em inúmeras palavras da língua portuguesa. Mais exemplos de divisões silábicas corrigidas serão apresentados na próxima seção, que aborda o desempenho do separador silábico diante das alterações propostas às regras usadas como referência.

3. Testes de desempenho

As regras para separação silábica foram testadas com um extrato de 10.000 palavras escolhidas de forma aleatória na base de dados CETENFolha [Cet 2013]. Tomou-se o cuidado de evitar acrônimos. O algoritmo proposto foi, então, comparado com outros dois algoritmos: um contendo apenas as 20 regras originais de [Silva et al. 2008]; e outro adicionando as duas regras para tratamento de encontros vocálicos descritas em [Monte et al. 2011].

Ao todo foram realizados três diferentes tipos de experimentos. Primeiramente, 1.000 palavras com foco em encontros vocálicos foram selecionadas aleatoriamente no extrato e submetidas aos algoritmos. Já no segundo teste, o desempenho dos algoritmos foi avaliado diante de mil palavras, sempre verbos ou adjetivos, também extraídas de forma aleatória. No último teste novamente foram selecionadas randomicamente 1.000 palavras dentro do extrato, mas dessa vez sem qualquer tipo de filtro. Todos os testes foram repetidos quatro vezes para que a média fosse considerada.

O Dicionário Online de Português [dic 2013] foi usado como referência de divisão silábica e a métrica usada para avaliar o desempenho dos algoritmos foi a taxa de erro por palavra, ou seja, basta uma sílaba separada incorretamente para a palavra ser contabilizada como erro. Claro que o Dicionário Online segue a ortografia ao fornecer a divisão silábica, com isso erros intencionais provocados por regras com apelo fonológico não foram considerados, como em “excesso” (*e-xce-ssó*), por exemplo. A Tabela 5 mostra o desempenho dos três algoritmos quando submetidos aos diferentes testes propostos.

Tabela 5. Desempenho dos algoritmos considerando a taxa de erro por palavra.

Algoritmos	Encontros Vocálicos	Verbos e Adjetivos	Sem Filtro
[Silva et al. 2008]	68,90 %	83,20 %	78,12 %
[Monte et al. 2011]	93,80 %	96,20 %	94,90 %
Algoritmo proposto	98,80 %	99,55 %	99,05 %

É visível a queda de desempenho do algoritmo original [Silva et al. 2008] na presença de encontros vocálicos. Isso se dá pelo fato do mesmo considerar uma sílaba fonológica as unidades constituídas por semi-vogal seguida de vogal, como em “criada” (*cria-da*), por exemplo. Esse tipo de tratamento não é correto, dado que claramente existem nesses casos dois esforços vocálicos e, conseqüentemente, duas sílabas (*cri-a-da*). O mesmo tratamento falho foi dado no encontro formado por vogal seguida de semi-vogal, como em “saída” (*sai-da*). Em função do uso das regras para tratamento de ditongos elaboradas em [Monte et al. 2011], tais inconsistências não foram observadas nos outros dois algoritmos.

A vantagem obtida pelo algoritmo proposto em relação a [Monte et al. 2011] foi conquistada no tratamento dedicado aos hiatos (alteração proposta na Regra 19). Esse

fato pode ser comprovado na correta divisão silábica fornecida para as palavras “campeonato” (*cam-pe-o-na-to*), “joelho” (*jo-e-lho*) e “israelense” (*is-ra-e-len-se*). Por sua vez, [Silva et al. 2008, Monte et al. 2011] consideram esses hiatos como ditongos e os mantêm equivocadamente na mesma sílaba.

As falhas encontradas no algoritmo proposto concentraram-se nos encontros vocálicos em final de palavra, como em “euforia” (*eu-fo-ria*), onde o ditongo crescente “ia” é mantido erroneamente na mesma sílaba. Isso acontece porque as regras propostas em [Monte et al. 2011] ainda possuem inconsistências na determinação da vogal tônica, o que se reflete no processo de divisão silábica. Logo, um estudo mais aprofundado na análise de tonicidade se faz necessário.

Na segunda bateria de testes, agora focando em verbos e adjetivos, observou-se um desempenho mais equilibrado entre os três algoritmos. A maioria dos erros continuou concentrada na separação dos encontros vocálicos. Por fim, foi feita uma avaliação sem qualquer tipo de filtro e, como esperado, o algoritmo original perde desempenho devido a presença marcante de encontro vocálicos no extrato. Todos os algoritmos sofreram com erros decorrentes de palavras estrangeiras presentes na base de teste.

4. Conclusões e trabalhos futuros

Este trabalho apresentou um estudo sobre o emprego de regras fonológicas para separação silábica em PB. Verificou-se que tal processo não exige grandes recursos computacionais e com poucas regras é possível atingir taxas de acerto razoáveis. Os recursos desenvolvidos encontram-se disponíveis ao público [Fal 2013], o que permite a reprodução dos resultados e a melhora gradual da eficiência das regras com a ajuda da comunidade acadêmica.

Apesar dos avanços aqui conquistados, especialmente no tratamento dos hiatos, é evidente que os erros ainda podem ser reduzidos, através do refinamento de algumas regras e manipulação dos estrangeirismos, que evidentemente não se encaixam nas regras da língua portuguesa. Além disso, testes subjetivos para avaliar a naturalidade da voz produzida pelo sistema TTS devem ser feitos para verificar se a abordagem aqui proposta é mais eficiente, ou não, que as regras usadas como referência.

Referências

- (Visitado em Julho, 2013). FalaBrasil: Tecnologias de Voz para o Português Brasileiro. Disponível em: <http://www.laps.ufpa.br/falabrasil/>.
- (Visitado em Março, 2013). Corpus de Textos Eletrônicos NILC/Folha de S. Paulo. Disponível em: acdc.linguateca.pt/cetenfolha/.
- (Visitado em Março, 2013). Dicionário Online de Português. Disponível em: www.dicio.com.br/.
- M. J. Barros e C. Weiss (2006). Maximum entropy motivated grapheme-to-phoneme, stress and syllable boundary prediction for Portuguese text-to-speech. *IV Jornadas en Tecnologías del Habla*, páginas 177–182.
- D. Braga e F. Resende Jr. (2007). Módulos de processamento de texto baseados em regras para sistemas de conversão texto-fala em Português Europeu. *XXI Encontro da Associação Portuguesa de Linguística*, páginas 141–156.

- E. L. da Silva e H. M. de Oliveira (2012). Implementação de um algoritmo de divisão silábica automática para arquivos de fala na língua portuguesa. *XIX Congresso Brasileiro de Automática*.
- Thierry Dutoit (2001). *An Introduction to Text-To-Speech Synthesis*. Kluwer.
- P. Gouveia, J. P. Teixeira e D. Freitas (2000). Divisão silábica automática do texto escrito e falado. *Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada*, páginas 65–74.
- R. Seara Jr., S. Kafka, I. Seara, F. Pacheco, S. Klein e R. Seara (2004). Parâmetros linguísticos utilizados para a geração automática de prosódia em sistemas de síntese de fala. *XXI Simpósio Brasileiro de Telecomunicações*, páginas 1–6.
- S. Madureira, P. Barbosa, M. Fontes, D. Spina e K. Crispim (1999). Post-stressed syllables in Brazilian Portuguese as markers. *XIV International Congress of Phonetic Sciences*, páginas 917–920.
- Mário Eduardo Martelotta (2008). *Manual de Linguística*. Contexto.
- A. Monte, D. Ribeiro, N. Neto, R. Cruz e A. Klautau (2011). A rule-based syllabification algorithm with stress determination for Brazilian Portuguese natural language processing. *17th International Congress of Phonetic Sciences*, páginas 1418–1421.
- C. Oliveira, L. Moutinho e A. Teixeira (2007). On European Portuguese automatic syllabification. *III Congreso Internacional de Fonética Experimental*, páginas 461–473.
- D. Silva, D. Braga e F. Resende Jr. (2008). Separação das sílabas e determinação da tonicidade no Português Brasileiro. *XXVI Simpósio Brasileiro de Telecomunicações*, páginas 1–5.
- Paul Taylor (2009). *Text-To-Speech Synthesis*. Cambridge University Press.