

Separação das Sílabas e Determinação da Tonicidade no Português Brasileiro

Denilson C. Silva, Daniela Braga e Fernando G. V. Resende Jr

Resumo—Este artigo apresenta dois algoritmos, baseados em regras lingüísticas, capazes de realizar a separação das sílabas e a determinação da tonicidade em palavras do português brasileiro. Os algoritmos propostos foram implementados e testados em um extrato aleatório de texto da base de dados CETEN-Folha, atingindo uma taxa de erro de 0,71% na separação das sílabas e 1,1% na determinação da tonicidade.

Palavras-Chave—Texto-fala, síntese de voz, separação silábica, ritmo, vogal tônica, prosódia

Abstract—This paper presents two algorithms, based on linguistic rules, capable to achieve the syllabification and stress determination on Brazilian Portuguese words. The proposed algorithms were implemented and tested on a randomly chosen extract of the CETEN-Folha text database, reaching an error rate of 0.71% on syllabification and 1.1% on stress determination.

Keywords—Text-to-speech, speech synthesis, syllabification, rithm, stressed vowel, prosody

I. INTRODUÇÃO

NOS últimos anos, sistemas de conversão texto-fala (TTS, *text-to-speech*) têm sido desenvolvidos como uma das principais inovações entre as tecnologias de voz, com o objetivo de converter a linguagem de texto em voz audível. Seja no âmbito internacional [1]-[3], bem como no português europeu [4]-[7], ou brasileiro [8], [9], é essencial realizar o mapeamento das sílabas quando tratamos de sistemas TTS, tanto para transcrição fonética, como para a geração de prosódia, já que numa sílaba tônica há um aumento de duração e intensidade.

Considerando que algumas regras do conversor grafema-fone (G2P, *grapheme-to-phone*) utilizam a informação de tonicidade, além de já ter sido comprovado que o uso de regras lingüísticas é uma boa escolha para sistemas TTS, pois o português brasileiro (PB) é fonologicamente regular, a solução para separação silábica e determinação da tonicidade por regras apresenta duas vantagens em relação a solução por dicionário: pouca utilização de memória e a capacidade de sempre poder ler uma nova palavra.

Propomos neste artigo um algoritmo composto de 20 regras lingüísticas para realizar a separação das sílabas e um algoritmo baseado em 19 regras para determinação da tonicidade no PB. Os algoritmos tiveram as suas regras testadas com um extrato aleatório de texto da base de dados CETEN-Folha [10].

Denilson C. Silva, Programa de Engenharia Elétrica, PEE/COPPE, UFRJ, Rio de Janeiro, RJ, E-mail: denilson@lps.ufrj.br

Daniela Braga, Microsoft Language Development Center, Porto, Portugal, E-mail: i-dbraga@microsoft.com

Fernando G. V. Resende Jr, DEL/Escola Politécnica, PEE/COPPE, UFRJ, Rio de Janeiro, RJ, E-mail: gil@lps.ufrj.br

O teste com o algoritmo para separação das sílabas, composto de 18.037 sílabas, atingiu a taxa de erro de 0,71% e o teste com o algoritmo para determinação da tonicidade, composto de 15.974 palavras, atingiu a taxa de erro de 1,1%.

Este artigo está organizado da seguinte forma: A Seção II apresenta o conjunto de regras para separação das sílabas. Na Seção III, apresentamos o conjunto de regras para determinação da tonicidade. Na Seção IV, temos a descrição dos testes realizados. Na Seção V são apresentadas as conclusões e trabalhos futuros.

II. ALGORITMO PARA SEPARAÇÃO DAS SÍLABAS

Nesta seção, apresentamos um algoritmo composto por 20 regras lingüísticas, utilizado para realizar a separação das sílabas nas palavras do PB. Na Tabela I a simbologia é descrita.

As regras são baseadas na busca das vogais existentes em cada palavra, seguida de análise dos caracteres existentes à esquerda e à direita, para então decidir qual ação deve ser tomada. O conjunto de ações está listado na Tabela II. 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 já separados. As regras lingüísticas são apresentadas nas Tabelas III e IV. As cinco primeiras regras tratam de vogais no início de sílabas, os demais casos analisam sílabas com vogais posicionadas internamente ou terminais. Este algoritmo foi desenvolvido baseado na ortografia [11], mas prevalecendo a fonologia das palavras, na tentativa de conciliar as teorias fonológicas da língua com as necessidades técnicas do sistema TTS [12], como é o caso das Regras 2, 5, 16 e 18, onde temos situações em que a fonologia prevalece em palavras como “burro” (*bu-rro*), “arredondar” (*a-rre-don-dar*) e “assado” (*a-ssa-do*).

As regras são dispostas hierarquicamente desde a primeira até a Regra 19. Este procedimento faz com que a carga computacional não seja muito grande, fator importante em sistemas que requerem uma resposta rápida, uma vez que, sendo a vogal analisada enquadrada em uma das regras, passa-se imediatamente para a vogal seguinte.

III. ALGORITMO PARA DETERMINAÇÃO DA TONICIDADE

Inicialmente, apresentamos o significado dos símbolos utilizados nas regras para determinação da tonicidade:

- O símbolo “^” é um ponteiro para os caracteres das palavras, a partir do final. Por exemplo, ^(0) é o último caractere de uma palavra, ^(1) é o penúltimo caractere da mesma.

TABELA I

SÍMBOLOS USADOS NAS REGRAS DE SEPARAÇÃO DAS SÍLABAS.

Símbolo	significado
V	Vogal (a, e, o, á, é, ó, í, ú, â, ã, ô, õ, ê, ã, ò, à, ù)
G	Semivogal (i, u)
C	Uma consoante qualquer (<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 except <lh>)
CN	Nasal (m; n)
SP	Espaço
F	Final de linha ou final do arquivo
p ⁰	Início da sílaba
^(+1) = C	Primeiro grafema à direita da vogal é uma consoante qualquer
^(+2) = G	Segundo grafema à direita da vogal é semivogal
^(+3) = V	Terceiro grafema à direita da vogal é vogal
^(-1) ≠ CN	Primeiro grafema à esquerda da vogal não é consoante nasal
^(-1) = CO	Primeiro grafema à esquerda da vogal é oclusiva

TABELA II

CASOS E AÇÕES TOMADOS PELO ALGORITMO.

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 separada 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 separada do terceiro grafema.
Caso 6	V é unida ao grafema anterior e a todos os grafemas seguintes até o final da palavra.

- O símbolo “T” indica a posição da vogal tônica em uma determinada palavra. Por exemplo, T = 2 significa que o antepenúltimo caractere é tônico.

As regras para a determinação da tonicidade são mostradas nas Tabelas V e VI. Este algoritmo funciona mediante uma análise das seqüências grafêmicas, partindo do último caractere, e das regras de acentuação gráfica da língua portuguesa [11]. As palavras são submetidas à seqüência de regras, desde a Regra 1 até a Regra 19. Caso a palavra esteja enquadrada em uma das regras, ela tem a sua sílaba tônica marcada, passando, então, para a palavra seguinte. As palavras átonas, tais como preposições, contrações, pronomes, conjunções, são tratadas como exceção no contexto das regras por não possuírem tonicidade.

A primeira regra já faz uma varredura na palavra buscando marcação gráfica prévia, obedecendo a questão da precedência. A Regra 10 trata de uma exceção da Regra 11, já que “porque” é oxítônica no PB. A Regra 18 também é uma exceção da Regra 19, para que o <u> não seja equivocadamente marcado no lugar do <e>, na palavra “quem”. As Regras 4, 6, 9, 12 e 15 determinam a tonicidade de palavras no plural, até então associadas a regras anteriores quando no singular.

A questão da acentuação lexical secundária no PB não foi considerada, para efeitos de geração de voz, um vez que se trata de um fenômeno de micro-prosódia, possivelmente com pouco impacto na inteligibilidade do sistema, como em *cafezinho* (acento primário em “zi” e secundário em “e”) e *impreterivelmente* (acento primário em “men” e secundário em “ri”).

TABELA III

REGRAS UTILIZADAS NA SEPARAÇÃO DAS SÍLABAS (DE 1 A 11).

Regra 1 if V = p ⁰ and V ≠ <ã>, <õ> and ^(+1) = V and ^(+1) ≠ G then Caso 1 end if Exemplo: <i>a-eronave, a-inda</i>
Regra 2 if V = p ⁰ and ^(+1) = C and ^(+2) = C and ^(+3) = CO then Caso 5 end if Exemplo: <i>o-bstar, a-dstrito</i>
Regra 3 if V = p ⁰ and ^(+1) = G, <s>, <r>, <l>, CN, <x> and ^(+2) = C and ^(+2) ≠ <s>, <h>, <r> then Caso 2 end if Exemplo: <i>am-bos, an-te, as-pas, al-tura, ar-gúcia, eu-ropa</i>
Regra 4 if V = p ⁰ and ^(+1) = C and ^(+2) = C and ^(+3) = V then Caso 1 end if Exemplo: <i>as-tral, ex-por, ei-ra, ai-po, o-ptar, a-dvogar, a-gnóstico, a-florar, a-fia</i>
Regra 5 if V = p ⁰ and ^(+1) = C and ^(+2) = V, CL then Caso 1 end if Exemplo: <i>a-rrendar, a-tlas, a-lho, a-mor, a-clamado</i>
Regra 6 if V ≠ p ⁰ and ^(-1) = C and ^(+1) = C and ^(+2) = V then Caso 3 end if Exemplo: <i>ca-lha, ca-la, me-ta, ca-choeira</i>
Regra 7 if V ≠ p ⁰ and ^(-1) = C, G and ^(+1) = G and ^(+2) = C and ^(+2) ≠ <m>, <n>, <r>, <s> seguido de SP, F then Caso 4 end if Exemplo: <i>cai-ro, rai-va, cau-sa, quei-xar</i>
Regra 8 if V ≠ p ⁰ and ^(-1) = C and ^(+1) = G and ^(+2) = <s> and ^(+3) = CO then Caso 5 end if Exemplo: <i>claus-tro</i>
Regra 9 if V ≠ p ⁰ and ^(-1) = C and ^(+1) = G and ^(+2) = V, SP then Caso 4 end if Exemplo: <i>prai-a, mei-a</i>
Regra 10 if V ≠ p ⁰ and ^(-1) = G and ^(+1) = C and ^(+2) = V then Caso 3 end if Exemplo: <i>pia-da, via-gem, sua-da</i>
Regra 11 if V ≠ p ⁰ and ^(-1) = C and ^(+1) = G and ^(+2) = <r> and ^(+3) = C then Caso 3 end if Exemplo: <i>ca-irmos</i>

O conjunto de regras apresentado foi inicialmente proposto em [8] com uma taxa de erro de 1,47%. Neste artigo, obtivemos uma sensível melhora ao tratarmos algumas daquelas regras. Naquela ocasião, o algoritmo proposto apresentava um conjunto de 20 regras. As melhorias implementadas para este trabalho estão relacionadas com a fusão das Regras 7 e 8 de [8] (aqui, Regra 7); um ajuste nas Regras 12 e 13 (aqui,

TABELA IV

REGRAS UTILIZADAS NA SEPARAÇÃO DAS SÍLABAS (DE 12 A 20).

Regra 12 if $V \neq p^0$ and $\hat{(-1)} = C, \{<u>, <ü>$ antecidos de $<q>, <g>$ and $\hat{(+1)} = CL, CN, <c>, <x>$ and $\hat{(+3)} = V, <h>, <l>, <r>$ then if $\hat{(+2)} = <h>, <l>, <r>$ Caso 1 else Caso 4 end if end if Exemplo: <i>car-ta, mal-dade, con-tar, esquer-da, aguar-do</i>
Regra 13 if $V \neq p^0$ and $\hat{(-1)} = C$ and $\hat{(+1)} = CL, CN, <i>$ then if $\hat{(+2)} = SP$ Caso 6 else if $\hat{(+2)} = <s>$ Caso 5 end if end if Exemplo: <i>va-ral, cais</i>
Regra 14 if $V \neq p^0$ and $\hat{(+1)} = V$ igual or $\hat{(+1)} = V$ and $\hat{(+2)} = V$ then Caso 1 end if Exemplo: <i>co-operação, bu-eiro</i>
Regra 15 if $V \neq p^0$ and $\hat{(+1)} = CO, <f>, <v>, <g>$ and $\hat{(+2)} = CL, CO$ and $\hat{(+3)} = V$ then Caso 1 end if Exemplo: <i>su-blime, ra-pto</i>
Regra 16 if $V \neq p^0$ and $V = <i>$ and $\hat{(-1)} = C$ and $\hat{(+1)} = <a>, <o>$ and $\hat{(+2)} = F$ then Caso 4 end if Exemplo: <i>políci-a, sóci-o, democraci-a</i>
Regra 17 if $V \neq p^0$ and $V = <ã>, <ô>$ and $\hat{(-1)} = C$ and $\hat{(+1)} = <o>, <e>$ and $\hat{(+2)} = SP, F, <s>$ then Caso 6 end if Exemplo: <i>gamão, limões</i>
Regra 18 if $V \neq p^0$ and $\hat{(-1)} = C, <u>$ antecido de $<q>$ and $\hat{(+1)} = C$ and $\hat{(+2)} = C$ then if $\hat{(+1)} = \hat{(+2)}$ Caso 1 else if $\hat{(+1)} = <s>$ and $\hat{(+2)} \neq <s>$ Caso 2 else if $\hat{(+2)} = <s>$ and $\hat{(+3)} = CO$ Caso 5 end if end if Exemplo: <i>bu-rro, bes-ta</i>
Regra 19 if $V \neq p^0$ and $\hat{(+1)} = V$ and $\hat{(+2)} = C$ then if $\hat{(+3)} = V$ Caso 2 else Caso 1 end if end if Exemplo: <i>teó-logo</i>
Regra 20 Se nenhum dos casos anteriores for verificado e a palavra terminar, V forma sílaba com os grafemas que restarem até o espaço em branco, sinal de pontuação ou hífen. Exemplo: <i>cas-to, des-cer</i>

TABELA V

REGRAS PARA DETERMINAÇÃO DA TONICIDADE (DA 1 ATÉ A 10)

Regra 1 Se existe acento, a vogal marcada é tônica. O acento agudo e o circunflexo tem precedência sobre o til. Exemplos: <i>órfão, órfã bênção, bênções.</i>
Regra 2 if $\hat{(0)} = \{r, l, z, x, n\}$ then T = 1 end if Exemplos: <i>propor, carrossel, rapaz, triplex, durex.</i>
Regra 3 if $\hat{(0)} = \{m\}$ and $\hat{(1)} = \{i, o, u\}$ then T = 1 end if Exemplos: <i>puđim, bombom, comum.</i>
Regra 4 if $\hat{(0)} = \{s\}$ and $\hat{(1)} = \{n\}$ and $\hat{(2)} = \{i, o, u\}$ then T = 2 end if Exemplos: <i>puđins, bombons, comuns.</i>
Regra 5 if $\hat{(0)} = \{i\}$ and $\hat{(1)} = \{u, ü\}$ and $\hat{(2)} = \{q, g\}$ then T = 0 end if Exemplos: <i>caqui, aqui, sagüi.</i>
Regra 6 if $\hat{(0)} = \{s\}$ and $\hat{(1)} = \{i\}$ and $\hat{(2)} = \{u, ü\}$ and $\hat{(3)} = \{q, g\}$ then T = 1 end if Exemplos: <i>caquis, sagüis.</i>
Regra 7 if $\hat{(0)} = \{i, u\}$ then if $\hat{(1)}$ é vogal $\neq \{u\}$ T = 1 else T = 0 end if end if Exemplos: <i>caiu, grau, pneu, caju, javali.</i>
Regra 8 if $\hat{(0)} = \{s\}$ and $\hat{(1)} = \{i, u\}$ and $\hat{(2)}$ é vogal then T = 2 end if Exemplos: <i>andaís, paúis, graus.</i>
Regra 9 if $\hat{(0)} = \{s\}$ and $\hat{(1)} = \{i, u\}$ and $\hat{(2)}$ não é vogal then T = 1 end if Exemplos: <i>cajus, javaliis.</i>
Regra 10 if $\hat{(0)} = \{e\}$ and $\hat{(1)} = \{u\}$ and $\hat{(2)} = \{q\}$ and $\hat{(3)} = \{r\}$ and $\hat{(4)} = \{o\}$ and $\hat{(5)} = \{p\}$ then T = 0 end if Exemplos: <i>porque.</i>

$\hat{(2)}$ ou $\hat{(3)} = \{g\}$. Outra providência foi ajustar a Regra 16 (aqui, Regra 15), introduzindo a exceção para $\hat{(5)} \neq \{q, g\}$. Estas melhorias proporcionaram uma redução na taxa de erro considerável (de 1,47% para 1,1%), com um extrato de texto três vezes maior do que o testado em [8].

IV. RESULTADOS OBTIDOS

A. Resultados com o algoritmo de separação das sílabas

As regras para separação das palavras em sílabas foram testadas com um extrato aleatório de texto da base de dados CETEN-Folha [10], composto de 18.037 sílabas. Os resulta-

Regras 11 e 12), com uma expansão para o caso de termos

TABELA VI

REGRAS PARA DETERMINAÇÃO DA TONICIDADE (DA 11 ATÉ A 19).

Regra 11
if $\hat{\alpha}(0) = \{e\}$ and $\hat{\alpha}(1) = \{u\}$ and $\hat{\alpha}(2) = \{q,g\}$ then
if $\hat{\alpha}(3)$ é vogal
T = 3
else
T = 4
end if
Exemplos: <i>alambique, Henrique, destaque, bosque.</i>
Regra 12
if $\hat{\alpha}(0) = \{s\}$ and $\hat{\alpha}(1) = \{e\}$ and $\hat{\alpha}(2) = \{u\}$ and $\hat{\alpha}(3) = \{q,g\}$ then
if $\hat{\alpha}(4)$ é vogal
T = 4
else
T = 5
end if
Exemplos: <i>alambiques, Henriques, destaques.</i>
Regra 13
if $\hat{\alpha}(0)$ é vogal and $\hat{\alpha}(1) = \{i,u\}$ and $\hat{\alpha}(2)$ é vogal then
T = 2
end if
Exemplos: <i>meia, seio.</i>
Regra 14
if $\hat{\alpha}(0) = \{s\}$ and $\hat{\alpha}(3)$ são vogais and $\hat{\alpha}(2) = \{i,u\}$ and $\hat{\alpha}(1)$ não é vogal and $\hat{\alpha}(4) \neq \{q,g\}$ then
T = 3
end if
Exemplos: <i>cadeira, queima, louco.</i>
Regra 15
if $\hat{\alpha}(0) = \{s\}$ and $\hat{\alpha}(1)$ and $\hat{\alpha}(4)$ são vogais and $\hat{\alpha}(3) = \{i,u\}$ and $\hat{\alpha}(2)$ não são vogais and $\hat{\alpha}(5) \neq \{q,g\}$ then
T = 4
end if
Exemplos: <i>cadeiras, queimas, loucos.</i>
Regra 16
if $\hat{\alpha}(0) = \{a,e,o\}$ and $\hat{\alpha}(1)$ é consoante and $\hat{\alpha}(2) = \{n\}$ and $\hat{\alpha}(3) = \{i,u\}$ and $\hat{\alpha}(4)$ é vogal then
T = 3
end if
Exemplos: <i>ainda, caindo, fluindo, incluindo.</i>
Regra 17
if $\hat{\alpha}(k) =$ penúltima vogal and $\hat{\alpha}(k) = \{i,u\}$ and $\hat{\alpha}(k+1)$ é vogal and $\hat{\alpha}(k-1)$ não é vogal and $\hat{\alpha}(k+2)$ não é $\{q,g\}$ then
T = k+1
end if
Exemplos: <i>outro, claustro.</i>
Regra 18
if $\hat{\alpha}(0) = \{m\}$ and $\hat{\alpha}(1) = \{e\}$ and $\hat{\alpha}(2) = \{u\}$ and $\hat{\alpha}(3) = \{q\}$ then
T = 1
end if
Exemplos: <i>quem.</i>
Regra 19
T = penúltima vogal da palavra
Exemplos: <i>casa, homem, guerra.</i>

TABELA VII

TABELA COM ERROS ENCONTRADOS NA SEPARAÇÃO DAS SÍLABAS.

Tipo de erro	Número de erros	% de erro
Estrangeirismos	43	0,24%
Acrônimos	8	0,04%
Outros erros	77	0,43%
TOTAL	128	0,71%

dos obtidos podem ser verificados na Tabela VII, onde foi alcançada uma taxa de erro de 0,71%.

Já era esperado que houvesse um número razoável de erros decorrentes de acrônimos e estrangeirismos, pois um conteúdo jornalístico, normalmente, apresenta muitos casos destes tipos,

TABELA VIII

TABELA DE ERROS ENCONTRADOS NA DETERMINAÇÃO DA TONICIDADE.

Tipo de erro	Número de erros	% de erro
Nomes próprios	86	0,54%
Estrangeirismos	53	0,33%
Acrônimos	20	0,13%
Outros erros	16	0,10%
TOTAL	175	1,10%

que não seguem as regras lingüísticas vigentes na língua. Por exemplo, "ABRAVEST" (*Associação Brasileira do Vestuário*), que teve a sua última consoante isolada e formando sílaba (A-BRA-VES-T), o que não ocorre no PB, pois toda sílaba, obrigatoriamente deve possuir uma vogal.

B. Resultados com o algoritmo de determinação da tonicidade

As regras para determinação da sílaba tônica foram testadas com um extrato de texto aleatório da base de dados CETEN-Folha, composto de 15.974 palavras.

Os resultados obtidos podem ser verificados na Tabela VIII, onde foi alcançada uma taxa de erro de 1,1%.

Os erros relacionados a nomes próprios justificam-se pelo fato de que, muitas vezes, esses nomes não estão de acordo com as regras vigentes no PB, como por exemplo "Sônia", que comumente é grafado como "Sonia", determinando uma falsa informação de tonicidade (Son[i]a). Verificamos que a ocorrência deste tipo de erro em número acentuado foi devido ao fato de que um conteúdo jornalístico apresenta diversas informações e notícias relacionadas às pessoas. Erros de estrangeirismos são justificáveis por si só, já que dificilmente estarão enquadrados nas regras lingüísticas.

Por fim, os acrônimos também não ficam enquadrados no contexto das regras. Podemos citar como exemplos para este caso as siglas "ABRAVEST" e "CRECI", que fonologicamente possuem marcação tônica como "ABRAV[E]ST" e "CR[E]CI" (Conselho Regional de Corretores de Imóveis), mas que pelas regras propostas têm a sua tonicidade definida como "ABR[A]VEST" e "CREC[I]".

Diante destes resultados, o conjunto de regras proposto pode ser considerado uma boa escolha para a separação das sílabas e para a marcação de tonicidade em sistemas TTS no PB.

V. CONCLUSÕES

Neste trabalho, apresentamos um algoritmo baseado em regras lingüísticas, num total de 20 regras, capaz de realizar a separação silábica de palavras do PB. O algoritmo proposto foi implementado e testado com um extrato de texto aleatório da base de dados CETEN-Folha e uma taxa de erro de 0,71% foi obtida.

Apresentamos também um algoritmo baseado em regras lingüísticas, num total de 19 regras, capaz de realizar a identificação da vogal tônica em palavras do PB. O algoritmo proposto foi implementado e testado com um extrato de texto aleatório da base de dados CETEN-Folha e uma taxa de erro de 1,1% foi atingida, superando os resultados apresentados em [8]. Como trabalhos futuros, desejamos analisar os erros ocorridos e ajustar ou criar novas regras. Pretendemos também

realizar novos testes com textos de natureza não-jornalística, bem como a predição do acento secundário e o estudo do seu impacto na naturalidade do sistema de síntese da fala no PB.

AGRADECIMENTOS

Este trabalho foi realizado no âmbito de projetos que receberam apoio financeiro do CNPq e da Faperj.

REFERÊNCIAS

- [1] R. Weerasinghe, A. Wasala and K. Gamage, *A rule based syllabification algorithm for Sinhala*, Proc. of Second International Joint Conference on Natural Language Processing (IJCNLP-05), pp. 438-449, Jeju Island, Korea, 2005.
- [2] G. Kiraz and B. Mobius, *Multilingual syllabification using weighted finite-state transducers*, In Proc. of the Third ESCA Workshop on Speech Synthesis, Jenolan Caves, Australia, 1998.
- [3] B. Mobius, *Word and syllable models for German text-to-speech synthesis*, Proc. of the Third ESCA Workshop on Speech Synthesis, 1998.
- [4] C. Oliveira, L. C. Moutinho and A. Teixeira, *On automatic european portuguese syllabification*, 9th European Conference on Speech Communication and Technology, pp. 2933-2936, 2005.
- [5] J. P. Teixeira, P. Gouveia and D. Freitas, *Divisão silábica automática do texto escrito e falado*, Proc. of PROPOR'2000, 2000.
- [6] J. P. Teixeira, E. R. Paulo, D. Freitas and M. da G. Pinto, *Acoustical characterisation of the accented syllable in Portuguese; a contribution to the naturalness of speech synthesis*, Proc. of the European Conf. on Speech Communication and Technology (EUROSPEECH), 1999.
- [7] D. Braga, *Algoritmos de processamento da linguagem natural para sistemas de conversão texto-fala em português*, Departamento de Galego-Português, Francês e Lingüística, Faculdade de Filoloxía da Universidade da Coruña, A Coruña, Espanha, Fev 2008.
- [8] D. C. Silva, A. A. Lima, R. Maia, D. Braga, J. F. Moraes, J. A. Moraes and F. G. V. Resende Jr., *A rule-based grapheme-phone converter and stress determination for Brazilian Portuguese natural language processing*, Proc. of the International Telecommunications Symposium, pp. 992-996, Fortaleza, Brasil, 2006.
- [9] R. Seara Jr., I. C. Seara, S. Kafka, F. S. Pacheco, R. Seara and S. Klein, *Parâmetros lingüísticos utilizados para a geração automática de prosódia em sistemas de síntese de fala*, Simpósio Brasileiro de Telecomunicações, 2004.
- [10] Corpus de Extratos de Textos Eletrônicos NILCS/Folha de São Paulo (CETEN-Folha), Online: <http://acdc.linguatca.pt/cetenfolha>, acessado em 11 Abr 2007.
- [11] E. Bechara, *Moderna Gramática Portuguesa*, Lucerna, 2002, Rio de Janeiro, RJ, Brasil.
- [12] R. S. Maia, H. Zen, K. Tokuda, T. Kitamura, F. G. V. Resende Jr., *A HMM-based Brazilian Portuguese Speech Synthesizer and its Characteristics.*, Revista da Sociedade Brasileira de Telecomunicações, vol. 21, no. 2, p.p 58-71, 2006.