A Rule-Based Method for Homograph Disambiguation in Brazilian Portuguese Text-to-Speech Systems

Denilson C. Silva, Daniela Braga, and Fernando Gil V. Resende Jr.

Abstract— **This work presents a rule-based algorithm set used to decide the pronunciation of homographs applied to a Brazilian Portuguese (BP) text-to-speech (TTS) system. The proposed approach is composed of a morphosyntactic analysis, which deals with homographs that belong to different part-of-speech (POS), and a semantic analysis, which deals with homographs that belong to the same POS. The algorithms were implemented to solve ambiguities for 111 homograph pairs organized into 23 disambiguation algorithms, and tested with three types of texts: news, Bible and literature. Computer experiments showed that a correct homograph pronunciation is obtained in 99.00% of the occurrences.**

Index Terms— **Text-to-Speech, Homograph, Speech Synthesis, Morphosyntactic Analysis, Semantic Analysis**

I. Introduction

IN text-to-speech (TTS) systems, the decision on the pronunciation of heterophonic homographs is a nontrivial problem. In Brazilian Portuguese (BP), whenever a homograph appears, the algorithms that undertake grapheme-phone conversion (G2P) need to decide between two possible situations: whether the stressed vowel is opened ([E]/[O]) or closed ([e]/[o]) [1]. Words such as <seca> (noun, "the *drought*", and verb, "he *dries*") have the same spelling, but different meanings and pronunciation. If those words are not correctly analyzed, they may give rise to a wrong phonetic transcription.

The number of homographs usually represents a small percentage of the analyzed text (about 1.0% in the text database used in this work), but in the context of speech synthesis, mistaken phonetic transcriptions produce a bad evaluation of the TTS system, even if it occurs in a small number of times. Therefore, minimizing G2P errors for homographs is fundamental to obtain a satisfactory evaluation of a TTS system.

Homographs are a subject widely analyzed in several languages: [2] presents a typology of homograph pairs in the English language and some traditionally used techniques for disambiguation, such as bayesian classifiers, n-gram taggers and decision trees, as well as the proposal of a hybrid system, combining the best of the three described approaches. In [3], the subject is treated in languages such as Thai, Chinese and Japanese, in which the words have no word-boundary delimiter, and a pattern recognition approach called "winnow" has been proposed to solve both word segmentation and homograph ambiguity problems altogether. [4] presents a study on the relation between Chinese characters and their pronunciations and also considers a solution for the disambiguation of polyphonic characters. Regarding disambiguation in European Portuguese TTS systems, [5] and [6] use morphosyntactic information, while in [7], the disambiguation is obtained through morphosyntactic as well as semantic information. For Brazilian Portuguese, in [8] and [9] a morphosyntactic analyzer is applied, and in [10] and [11], both morphosyntactic and semantic approaches are presented, but the algorithms were designed for only one homograph.

In this work a rule-based algorithm set is proposed to solve homograph disambiguation applied to a BP TTS system [12]. The proposed approach is composed of a morphosyntactic analysis, which deals with problems of homographs that belong to different POS, and a semantic analysis, which deals with problems of homographs that belong to the same POS. Modifications produced by a recent orthographic agreement in Portuguese language [13] are also taken into account. The algorithms were implemented to solve ambiguities for 111 homograph pairs organized into 23 disambiguation algorithms, and tested with three types of texts: news, Bible and literature. The overall homograph correct pronunciation rate achieved through computer experiments is 99.00%.

This work is organized as follows. In Section II, the proposed method for homograph disambiguation and its characteristics are described. In Section III, computer experiments with data extracted from CETENFolha text database [14], Holy Bible [15] and Brazilian literature [16] are presented. Finally, Section IV contains our conclusions.

II. Applied methodology

In Table I, the homograph set used in this work is shown. The following libraries were developed:

- Homograph library, with 111 homograph pairs grouped in 23 types;
- A closed POS library for articles, conjunctions, contractions, interjections, numerals, prepositions and pronouns;
- A morphemes library, with noun, verb, adverb and adjective suffixes, prefixes, Latin and Greek affixes;

Denilson C. Silva is with the Brazilian Air Force, Rio de Janeiro, Brazil. e-mail: speechsolutions@speechsolutions.com.br.

Daniela Braga is whith the Microsoft Corporation, Bellevue, USA. email: dbraga@microsoft.com.

Fernando Gil V. Resende Jr. is with the Department of Electronic Engineering and Computer Science (DEL/Escola Politecnica), and with the Program of Electrical Engineering (PEE/COPPE), of the Federal University of Rio de Janeiro (UFRJ), Bl. H-219, Rio de Janeiro, Brazil, P.O. Box:68564. email: gil@lps.ufrj.br.

This work was partially presented at the 27th Brazilian Telecommunications Symposium (SBrT'09), September, 2009, Blumenau-SC, Brazil

TABLE I
HOMOGRAPH SET SPLITTED BY TYPE.

Type	Homograph set
1	acerto, apelo, aperto, apreço, começo, concerto, conserto, desemprego, desespero, emprego, enredo, erro, esmero, espeto, flagelo, gelo, governo, interesse, interesses, modelo, pego, peso, rego, selo, testo e zelo.
2	aborto, acordo, adorno, aforro, almoço, apoio, arrojo, arroto, choco, choro, conforto, consolo, contorno, controle, coro, desgosto, despojo, destroço, encosto, endosso, esforço, estorvo, folgo, gosto, jogo, logro, namoro, olho, piloto, reforço, rodo, rogo, rolo, sopro, suborno, sufoco, toco, toldo, topo, torno, troco e troço.
3	rola e rolha.
4	colher e meta.
5	desses, deste e destes.
6	fora.
7	seco, seca e secas.
8	boto.
9	este.
10	leste.
11	sobre.
12	rota, rotas, tola e tolas.
13	corte, cortes, forma, formas, molho e soco.
14	cerca.
15	pega e pegas.
16	pelo, pela e pelas.
17	besta e bestas.
18	sede e sedes.
19	medo e medos.
20	termos.
21	cor.
22	lobo e lobos.
23	bola e bolas.

TABLE II
EXAMPLES WITH HOMOGRAPHS THAT BELONG TO DIFFERENT POS.

Type	Stress alternations and Grammatical oppositions	Example
1	[e] Noun / [E] Verb	Nosso erro foi muito grande. Eu erro bastante.
2	[o] Noun / [O] Verb	Ele fechou o olho esquerdo. Eu olho para cima.
3	[o] Noun / [O] Verb	Eu vi uma rola branca. Ele deita e rola.
4	[e] Noun / [E] Verb	É época de colher o tomate. Essa é a nossa meta.
5	[e] Contraction / [E] Verb	Ele ganhou dois desses prêmios. Era bom que nunca desses a notícia.
6	[o] Verb / [O] Adverb	Ele fora uma pessoa honesta. Eu estou fora do jogo.
7	[e] Adjective or Noun / [E] Verb	O rio estava muito seco. Eu seco os pés na entrada.
8	[o] Adjective or Noun / [O] Verb	Ele viu um boto na praia. Eu boto azeite na salada.
9	[e] Demonstrative / [E] Adjective or Noun	Este armário é meu. Norte, sul, este, oeste.
10	[e] Verb / [E] Adjective or Noun	Leste a notícia?. Seguiu para o leste.
11	[o] Preposition / [O] Verb	Comentou sobre o fato. É bom que sobre uma garrafa.
12	[o] Adjective or Verb [O] Noun /	Ela andava toda rota. Nós seguimos a rota.
13	[o] Noun / [O] Verb / Noun	Ela comprou pão de forma. De qualquer forma iremos ao passeio.
14	[e] Preposition / Noun / [E] Verb	Eles andaram cerca de dez kilômetros. Ele cerca seu terreno com arame farpado.
15	[e] Noun / [E] Verb / Noun	Aquela ave parece uma pega. Olha que essa moda ainda pega.
16	[e] Contraction / Noun [E] Verb / Noun	Nós passamos pela rua. Ela pela o pelo do corpo.

TABLE III
EXAMPLES WITH HOMOGRAPHS THAT BELONG TO THE SAME POS.

Type	Stress alternations and Grammatical oppositions	Example
17	[e] Noun / [E] Noun	Ele é metido a besta. Ele conseguia disparar a besta.
18	[e] Noun / [E] Noun	Ele estava com uma sede insuportável. A sede da empresa fica em Paris.
19	[e] Noun / [E] Noun	Ela estava com medo de morrer. Eles venceram todo o Império Medo-Persa.
20	[e] Noun / Verb [E] Noun	Estes são os nossos termos. A termos tinha café quente.
21	[o] Noun / [O] Noun	O vestido era cor de rosa. Sabia tudo de cor e salteado.
22	[o] Noun / [O] Noun	Na estória não tinha lobo mau. Ele feriu o lobo temporal.
23	[o] Noun / [O] Noun	Só amassei a bola de carne. Eu não tenho bola de cristal.

- A lemmas library, which features the Portuguese Jspell dictionary with approximately 34 000 morphologically annotated words [17];
- An irregular verbs library, with the inflexion forms of the main existing irregular verbs in the BP;
- A library consisting of the verb “to be” in the third person followed by an adjective;
- A restrict lexical combinations library, with idiomatic expressions, proverbs, or fixed expressions with one or more words. This library is only used in the semantic analysis;
- A Wordnets library, developed under the concept of Wordnets [18], [19], with words that are semantically and cognitively related with the analyzed homograph. This library also is required only in the semantic analysis.

In the processing, the text is split into words and phrases. The system carries through the search for every homograph, and applies the corresponding algorithm type.

The homographs that belong to different POS and to the same POS are shown in Table II and in Table III, respectively. As shown in Table II, the grammatical oppositions are more frequent between nouns and verbs, according to the morphological concept, and between [e]/[E] and [o]/[O], according to the phonetic concept. The evidence is that in nouns the stressed vowel is typically closed, while in verbal forms the stressed vowel is opened. Type 1 and 2 homographs represent 61.3% of the total number of homographs in the test library. Type 13, 14, 15 and 20 homographs need both morphosyntactic and semantic analysis.

In the Appendix all the proposed algorithms, from Algorithm 1 (Homograph type 1) up to Algorithm 23 (Homograph type 23) can be found. The symbols used in the algorithms can be seen in Table IV.

The Algorithm 16 was included to attend to the recently signed Orthographic Agreement [13]. This agreement is only orthographic; therefore, it is restricted to the written language and does not affect any aspect of the spoken language.

III. COMPUTER EXPERIMENTS

The proposed algorithms were tested with three different types of texts: news, Bible and literature. The results can be found in Tables V, VI and VII.

The CETENFolha text database is a corpus containing approximately 24 million words in BP extracted from Folha de São Paulo newspaper [14] built by the Computational

TABLE IV

APPLIED SYMBOLOGY IN THE DISAMBIGUATION ALGORITHMS.

Symbol	Meaning
P-1, P-2, P+1	last word, second last word and the next word, respectively.
F0, F-1, F+1	current phrase, last phrase and the next phrase, respectively.
P_DEM	demonstrative pronoun.
P_IND	indefinite pronoun.
P_INT	interrogative pronoun.
P_POSS	possessive pronoun.
A_IND	indefinite article.
P_RELA	relative pronoun.
PREPO	preposition.
CONTR	contraction.
P_PESS_SU	personal pronoun subject.
P_PESS_O	personal pronoun object (<me>,<mim>,<te>, <ti>, <se>, <si>,<nos>, <vos>,<lhe(s)>,<no-lo(s)>, <no-la(s)>, <vo-lo(s)>, <vo-la(s)>, <lho(s)> or <lha(s)>).
CS	subordinative conjunction.
CC	coordinative conjunction.
HN	"a", "o", "as" or "os" (pronoun or definite article).
nc	common noun.
adv	adverb.
ad	adjective.
NUM	numeral.
DESV	verbal suffixes set.
PART	participle.
BC	restrict lexical combination.
WN	wordnet.
V	vowel.

TABLE V

TESTS WITH PROPOSED ALGORITHM - CETENFOLHA.

Type	Occurrence	Hits	Rate
1	3 409	3 365	98.71%
2	3 046	2 965	97.34%
3	11	10	90.91%
4	95	90	94.74%
5	637	636	99.84%
6	482	471	97.72%
7	90	80	88.89%
8	5	5	100.00%
9	825	825	100.00%
10	169	169	100.00%
11	2 335	2 321	99.40%
12	47	45	95.74%
13	826	813	98.43%
14	866	863	99.65%
15	43	43	100.00%
16	6 656	6 653	99.95%
17	11	10	90.91%
18	148	141	95.27%
19	130	130	100.00%
20	108	101	93.52%
21	68	68	100.00%
22	39	39	100.00%
23	262	262	100.00%
TOTAL	20 308	20 105	**99.00%**

Processing of Portuguese Project. The system was tested with a random extract containing 1 564 591 words, of which 20 308 homograph pairs were detected (1.30% of the processed text). The text was processed and a correctness rate of 99.00% was achieved.

The other database is a version, in text format, of the Holy Bible in BP [15]. It is composed of 750 000 words, presenting a more formal style than that of the CETENFolha database. This test detected 7 904 homographs (1.05% of the processed text) and a correctness rate of 99.00% was achieved.

The text from Brazilian literature [16] is composed of 70 000 words. It is a romance narrated in the first person.

TABLE VI

TESTS WITH PROPOSED ALGORITHM - HOLY BIBLE.

Type	Occurrence	Hits	Rate
1	209	205	98.09%
2	322	311	96.58%
3	5	4	80.00%
4	27	25	92.59%
5	333	321	96.40%
6	428	422	98.60%
7	61	56	91.80%
8	0	—	—
9	984	984	100.00%
10	5	4	80.00%
11	2 740	2 726	99.49%
12	11	10	90.91%
13	65	61	93.85%
14	51	49	96.08%
15	5	5	100.00%
16	2 345	2 344	99.96%
17	46	45	97.83%
18	107	97	90.65%
19	82	81	98.78%
20	60	58	96.67%
21	3	2	66.67%
22	14	14	100.00%
23	1	1	100.00%
TOTAL	7 904	7 825	**99.00%**

TABLE VII

TESTS WITH PROPOSED ALGORITHM - BRAZILIAN LITERATURE.

Type	Occurrence	Hits	Rate
1	36	36	100.00%
2	73	72	98.63%
3	0	—	—
4	3	3	100.00%
5	30	30	100.00%
6	52	50	96.15%
7	6	6	100.00%
8	0	—	—
9	86	86	100.00%
10	0	—	—
11	35	35	100.00%
12	2	1	50.00%
13	5	5	100.00%
14	2	2	100.00%
15	1	1	100.00%
16	123	123	100.00%
17	7	7	100.00%
18	1	1	100.00%
19	22	22	100.00%
20	5	4	80.00%
21	17	17	100.00%
22	0	—	—
23	4	4	100.00%
TOTAL	510	505	**99.02%**

This test detected 510 homographs (0.73% of the total text) and a correctness rate of 99.02% was achieved.

The overall result is obtained as follows:

$$\begin{aligned}\textbf{Overall result} &= \frac{20\ 105 + 7\ 825 + 505}{20\ 308 + 7\ 904 + 510} * 100\%. \quad (1)\\ &= 99.00\%.\end{aligned}$$

It could be observed that most of the errors occur while running Algorithms 1 and 2 when the homograph was followed by a preposition or contraction, or anteceded by conjugated verbal forms. The performance of the proposed algorithm did not vary signifcantly with the type of text.

IV. Conclusions

In this work it was presented an algorithm set based on linguistic rules for homograph disambiguation applied to a BP TTS system. The proposed algorithms are capable of determining the correct pronunciation of 111 pairs of homographs in BP. The algorithms are based on morphosyntactic and semantic analysis. The algorithm set was implemented and tested on a randomly chosen extract of a newspaper text database, the Holy Bible and a text from Brazilian literature. An overall correct pronunciation result of 99.00% was achieved through computer experiments.

Acknowledgment

The authors would like to thank the reviewers for their constructive remarks.

References

[1] SAMPA Website, Speech Assessment Methods Phonetic Alphabet, 1993, http://www.phon.ucl.ac.uk/home/sampa/home.htm, visited on 02/23/2008.

[2] D. Yarowsky, "Homograph disambiguation in Text-to-Speech Synthesis," *Progress in Speech Synthesis* (Jan van Santen, Richard Sproat, Joseph Olive, and Julia Hirschberg, editors), pp. 159-174, New York: Springer, 1996.

[3] V. Tesprasit, P. Charoenpornsawat, and V. Sornlertlamvanich, "A context-sensitive homograph disambiguation in Thai text-to-speech synthesis," in *Proc. HLT-NAACL'2003*, short papers, vol. 2, 2003.

[4] H. Dong, J. Tao, and B. Xu, "Grapheme-to-phoneme conversion in Chinese TTS system," in *Proc. 2004 International Symposium on Chinese Spoken Language Processing*, pp. 165-168, 2004.

[5] R. Ribeiro, L. C. Oliveira, and I. Trancoso, "Using Morphossyntactic Information in TTS Systems: Comparing Strategies for European Portuguese," in *Proc. of the 6th Workshop on Computational Processing of the Portuguese Language (PROPOR'2003)*. Springer-Verlag, Heidelberg, pp. 143-150, 2003.

[6] R. Ribeiro, L. C. Oliveira, and I. Trancoso, "Morphossyntactic Disambiguation for TTS Systems," in *Proc. of the 3rd Intl. Conf. on Language Resources and Evaluation*. vol. 5. pp. 1427-1431, 2002.

[7] D. Braga, L. Coelho, and F. G. V. Resende Jr., "Homograph Ambiguity Resolution in Front-End Design for Portuguese TTS Systems," in *Proc. of Interspeech 2007*, pp. 1761-1764, Aug. 2007.

[8] I. Seara, S. Kafka, S. Klein, and R. Seara, "Considerações sobre os problemas de alternância vocálica das formas verbais do Português falado no Brasil para aplicação em um sistema de conversão Texto-Fala," in *Proc. 2001 19th Brazilian Telecommunication Symposium (SBrT2001)*, 2001.

[9] I. Seara, S. Kafka, S. Klein, and R. Seara, "Alternância vocálica das formas verbais e nominais do Português Brasileiro para aplicação em conversão Texto-Fala," *Revista da Sociedade Brasileira de Telecomunicações*. vol. 17, no. 1, pp. 79-85, June 2002.

[10] L. Ferrari, F. Barbosa, and F. G. V. Resende Jr., "Construções gramaticais e sistemas de conversão texto-fala: o caso dos homógrafos," in *Proc. of the International Conference on Cognitive Linguistics*, Braga, Portugal, 2003.

[11] F. Barbosa, L. Ferrari, and F. G. V. Resende Jr., "A methodology to analize homographs for a Brazilian Portuguese TTS system," in *Proc. of the 6th Workshop on Computational Processing of the Portuguese Language (PROPOR'2003)*. Springer-Verlag, Heidelberg, pp. 57-61, 2003.

[12] R. Maia, H. Zen, K. Tokuda, T. Kitamura, and 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, pp. 58-71, Aug. 2006.

[13] Acordo Ortográfico da Língua Portuguesa, Decreto no. 6583, de 29 de setembro de 2008. D6583.htm. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Decreto/, visited on 10/13/2009.

[14] CETENFolha Database, Corpus de Extratos de Textos Eletrônicos NILC/Folha de São Paulo (CETENFolha), 2003, http://www.linguateca.pt/cetenfolha/index_info.html, visited on 02/23/2008.

[15] The Holy Bible Database, J. F. Almeida version, 1993. biblia.rtf. [Online]. Available: http://www.culturabrasil.pro.br/zip/, downloaded on 03/16/2009.

[16] The Brazilian Literature text Database - Dom Casmurro, M. de Assis, 1899 (2007), ABL, http://www.machadodeassis.org.br, downloaded on 03/16/2009.

[17] The JSpell Dictionary, 2009. jspell.pt.ao1990. [Online]. Available: http://softwarelivre.sapo.pt/projects/bigorna/browser/trunk/dicionario/, downloaded on 10/31/2009.

[18] C. Fellbaum, *WordNet: An Electronic Lexical Database*, The MIT Press, 1998.

[19] About Wordnet, 2008, http://wordnet.princeton.edu, visited on 02/23/2008.

Denilson da Cruz da Silva received the B.Sc. degree in telecommunication engineering from Federal Center of Technological Education of Rio de Janeiro (CEFET-RJ), Rio de Janeiro, Brazil, in 1999, and the M.Sc and D.Sc degree in electrical engineering from Federal University of Rio de Janeiro (UFRJ/COPPE), Rio de Janeiro, Brazil, in 2005 and 2011, respectively. Currently he is working with the Brazilian Air Force. His research interests include emotional speech synthesis, natural language processing and robust speech recognition.

Daniela Braga holds a degree in Linguistics (2000) from the University of Oporto, Portugal, a Master's in Linguistics from the University of Minho, Portugal and an European PhD in Speech Synthesis (2008) from the University of A Coruña, Spain. From 2000 to 2006 she was a Researcher in Speech Technology in the University of Oporto (Portugal) as well as Assistant Lecturer in the Universities of Oporto (Portugal) and A Coruña (Spain). She has been participated in national and international R&D projects and consortia (FP6 and FP7 funded networks and projects, COST actions, QREN-national funded projects) since 2001. From Nov. 2006 to Nov. 2010, she was the head of the Text-to-Speech and Language Expansion team at MLDC - Microsoft Language Development Center (Lisbon), where she has been responsible for end-to-end product life cycles and several linguistic-related feature areas. From Nov. 2010-Nov.2011, she was a Program Manager in the Speech team in Microsoft in Beijing, where she was the technical manager responsible for the Prosody enhancement, technology roadmap and for the TTS release for Windows 8. Since Nov. 2011, she moved to the Microsoft headquarters in the US, joining the Information Platform and Experiences team in Redmond, WA, being responsible to drive the Crowdsourcing data collection strategy for IPE, including the Speech team. She is author and co-author of over 70 papers in Text-to-Speech Conversion, Speech Synthesis, Phonetics, Prosody, and Speech Recognition and has been member of scientific committees of several international conferences and Journals on Speech and Language Processing.

Fernando Gil Vianna Resende Junior received the B.Sc. degree from Military Institute of Engineering (IME), Brazil, in 1990, and the M.Sc. and Ph.D. degrees from Tokyo Institute of Technology (TIT), Japan, in 1994 and 1997, respectively, all in electrical engineering. Since 1998 he has been with the Department of Electronic Engineering and Computer Science, Polytechnic School, Federal University of Rio de Janeiro (UFRJ), as Associate Professor. Also, since 2003 he has been with the Program of Electrical Engineering, COPPE/UFRJ. His research interests are in the areas of natural language processing, speech synthesis, speech and speaker recognition, and speech coding.

APPENDIX
PROPOSED ALGORITHMS

Algorithm 1

```
if (Word is a homograph of the type 1) then
  if (P-1 = P_DEM, P_IND, P_INT or P_POSS) or (P-1, P-2 or P-3 = A_IND) or (P-1 or P-2 = HN, CONTR or PREPO) or (P+1 = <que> or P_RELA) then
    V = [e]
  else if (P-1 = P_PESSO_SU, P_PESS_O_1 or CS) or (P+1 = PREPO, CONTR, P_PESS_O_1 or HN) or (P+1 = A_IND e P+2 = nc) or (P-1 or P-2 = <não> or <nunca>) then
    V = [E]
  else
    V = [e]
  end if
else
  Go to Algorithm 2
end if
```

Algorithm 2

```
if (Word is a homograph of the type 2) then
  if (P-1 = P_DEM, P_IND, P_INT or P_POSS) or (P-1, P-2 or P-3 = A_IND) or (P-1 or P-2 = HN, CONTR or PREPO) or (P+1 = <que> or P_RELA) then
    V = [o]
  else if (P-1 = P_PESSO_SU, P_PESS_O_1 or CS) or (P+1 = PREPO, CONTR, P_PESS_O_1 or HN) or (P+1 = A_IND e P+2 = nc) or (P-1 or P-2 = <não>, <nunca> or <ainda>) then
    V = [O]
  else
    V = [o]
  end if
else
  Go to Algorithm 3
end if
```

Algorithm 3

```
if (Word is a homograph of the type 3) then
  if (P+1 = <pelo>, ad or adv) or (P-2 or P-3 = A_IND or HN) or (P-1 = <que>, <ele>, <ela>, <se>, <não>, <já>, <as>, nc, CC or CS) or (P-1 or P-2 = P_DEM, P_IND, P_INT or P_POSS) or (P+1 = <e> e P+2 = <rebola>) then
    V = [O]
  else if (P-1 = <à> e P-2 = <tiro> or <caça>) or (P-1 = <uma> or <a>) or (P-1 or P-2 = CONTR or PREPO) or (P+1 = <brava>) then
    V = [o]
  else
    V = [O]
  end if
else
  Go to Algorithm 4
end if
```

Algorithm 4

```
if (Word is a homograph of the type 4) then
  if Homograph = <colher> then
    if (P+1 = <de>) or (P-2 begins with <met-> e P-1 = <a>) or (P-1 = <à>, HN or A_IND) then
      V = [E]
    else if (P+1 = HN or A_IND e P+2 = nc) or (P+1 = P_DEM, P_POSS or P_IND e P+2 = nc) or (P+1 = <em>, <no>, <na>, <nos> or <nas>) or (P+1 ends by <-os>, <-as>, <-ões>, <-ães>, <-ãs>, <-res> or <-es>) or (P-1 or P-2 = <não>) then
      V = [e]
    else
      V = [E]
    end if
  else if Homograph = <meta> then
    if (P-1 = P_DEM, P_IND, P_INT or P_POSS) or (P-1, P-2 or P-3 = A_IND) or (P-1 or P-2 = HN, CONTR or PREPO) or (P+1 = <que> or P_RELA) then
      V = [E]
    else if (P-1 = P_PESSO_SU, P_PESS_O_1 or CS) or (P+1 = PREPO, CONTR, P_PESS_O_1 or HN) or (P+1 = A_IND e P+2 = nc) or (P-1 or P-2 = <não>, <nunca> or <que>) then
      V = [e]
    else
      V = [E]
    end if
  end if
else
  Go to Algorithm 5
end if
```

Algorithm 5

```
if (Word is a homograph of the type 5) then
    if (P-1 = NUM) then
        V = [e]
    else if (P-1, P-2 or P-3 = <tu>, <vós>, <ontem>, <se>, <talvez>, <oxalá> or CS) or (P+1 = P_PESS_O_1 or P_PESS_O_2) or (P-1 or P-2 = <não> or <nunca>) or (P-1 = P_PESS_O_1) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 6
end if
```

Algorithm 6

```
if (Word is a homograph of the type 6) then
    if (P+1 or P+2 termina com <-ndo>, <-ado>, <-ada>, <-ido> or <-ida>) or (P+1 or P+2 = PART_IRR) or (P+1 = <apenas>, A_IND or HN) or (P-1 = <eu>, <ele>, <ela>, <você>, <onde>, <como>, <quando> or <quem>) then
        V = [o]
    else if (P+1 = <de>, <do>, <da>, <dos>, <das> or CONTR) or (P-1 or P-2 = <lá>, <cá> or <aí>) or (P-1 or P-2 ends by <-mente>) or (P-1 or P-2 begins with <deit->, <deix->, <atir->, <empat->, <consider->,<fic->, <est-> or <jog->) or (P-1 = <borda>, <jantar>, <comer>, <noite>, <mundo>, <dia>, <tarde>, <por>, <de> or <para>) or (P-1 ends by <-ar>, <-er> or <-ir>) then
        V = [O]
    else
        V = [o]
    end if
else
    Go to Algorithm 7
end if
```

Algorithm 7

```
if (Word is a homograph of the type 7) then
    if (P-1 = P_PESS_SU, P_PESS_O_1 or CS) or (P+1 = P_PESS_O_1, CONTR or HN) or (P-1 or P-2 = <não>, <nunca>, <ainda> or <já>) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 8
end if
```

Algorithm 8

```
if (Word is a homograph of the type 8) then
    if (P-1 = <eu>, P_PESS_O_1 or CS) or (P-1 or P-2 = <não> or <nunca>) or (P+1 = <fora>, P_PESS_O_1, CONTR or HN) then
        V = [O]
    else
        V = [o]
    end if
else
    Go to Algorithm 9
end if
```

Algorithm 9

```
if (Word is a homograph of the type 9) then
    if (P+1, P+2 or P+3 = <oeste>) or (P-1 = <vento>) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 10
end if
```

Algorithm 10

```
if (Word is a homograph of the type 10) then
    if (P-1 = <não> or <já>) or (P-1 or P-2 = <ainda> or <nunca>) or (P-1, P-2 or P-3 = <tu>) or (P+1 = HN, A_IND or P_PESS_O_1) then
        V = [e]
    else
        V = [E]
    end if
else
    Go to Algorithm 11
end if
```

Algorithm 11

```
if (Word is a homograph of the type 11) then
    if (P+1 = <ti>, <mim> or <si>, HN, P_PESS_SU or P_PESS_O_1) or (P-1 = P_PESS_SU or P_PESS_O_1 e P+1 = A_IND) or (P-1, P-2 or P-3 = VERB or VERB_IRR) or (P-1 = nc or P_PESS_SU e P+1 or P+2 = nc) then
        V = [o]
    else if (P-1 = P_PESS_SU, P_PESS_O_1 or CS) or (P-1 or P-2 = <não> or <nunca>) or ((P-1 or P-2 = <que> or <ainda>) e (P+1 = A_IND)) or (P+1 = PREPO, CONTR or P_PESS_O_1) then
        V = [O]
    else
        V = [o]
    end if
else
    Go to Algorithm 12
end if
```

Algorithm 12

```
if (Word is a homograph of the type 12) then
    if (P-1 = <da>, <das>, <na>, <nas>, <pela>, <pelas> or <em>) or (P-1 or P-2 = <a>, <uma>, <mesma>, <ortra>, <de>, <por>, P_DEM, P_POSS or CONTR) or (P+1 = CONTR) then
        V = [O]
    else if (P-1 = <toda>) or (P-1 ends by <-mente>) or (P-1 or P-2 = nc) then
        V = [o]
    else
        V = [O]
    end if
else
    Go to Algorithm 13
end if
```

Algorithm 13

```
if (Word is a homograph of the type 13) then
    if (The homograph is inside the BC_forma_o) or (WN_forma_o is on F0) or (P-1 = <uma> and the word is <corte>) or (P-1 = <um> and the word is <molho> or <soco>) then
        V = [o]
    else if (P-1 or P-2 = <a>, <uma>, <esta>, <qualquer>, P_IND, P_DEM, P_POSS, CONTR or PREPO) or (P+1 or P+2 = ad) or (The homograph is inside the BC_forma_O) then
        V = [O]
    else
        V = [O]
    end if
else
    Go to Algorithm 14
end if
```

Algorithm 14

```
if (Word is a homograph of the type 14) then
    if (The homograph is inside the BC_cerca_e) or (WN_cerca_e is on F0) or (P+2 or P+3 = NUM) then
        V = [e]
    else if (P-1 = <uma>, <a>, CONTR or PREPO) or (P+2 = <madeira>, <arame>, <espinhos>) or (<saltar> or <levantar> is on F0) or (P+1 = ad) then
        V = [e]
    else if (P-1 or P-2 = <que>, <não>, <ainda>, <já> or <também>) or (P-1 = <ele>, <ela> or P_PESS_O_1) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 15
end if
```

Algorithm 15

```
if (Word is a homograph of the type 15) then
    if (P+1 = <em>, <no>, <na>, <nos>, <nas> or <fogo>) or (P-1 = <nunca>, <não>, <ainda>, <já>, <também>, <moda>, <se> or CS) or (P+1 = <ao> e P+2 = <colo>) then
        V = [E]
    else if (P-1 = <na>) or (P+1 = <a>, <uma>, <outra>, <mesma>, P_DEM or P_POSS) e (The homograph is inside the BC_pega_E) or (WN_pega_E is on F-1, F0 or F+1) then
        V = [E]
    else if (P+1 = <a>, <uma>, <outra>, <mesma>, P_DEM or P_POSS) e (The homograph is inside the BC_pega_e) or (WN_pega_e is on F-1, F0 or F+1) then
        V = [e]
    else
        V = [E]
    end if
else
    Go to Algorithm 16
end if
```

Algorithm 16

```
if (Word is a homograph of the type 16) then
    if (P+1 = <senhor>, <que>, <qual>, <tua>, <teu>, <minha>, <meu>, <sua>, <seu>) or (P-1 or P-2 = nc) or (P-1, P-2 or P-3 = VERB or VERB_IRR) then
        V = [e]
    else if (P-1 or P-2 = <o(s)>, <um>, <uns>, <esse(s)>, <este(s)>, <aquele(s)>, <nesse(s)>, <desse(s)>, <deste(s)>, <daquele(s)>) then
        V = [e]
    else if (P-1 or P-2 = <a(s)>, <uma(s)>, <essa(s)>, <esta(s)>, <aquela(s)>, <nessa(s)>, <dessa(s)>, <desta(s)>, <daquela(s)>) then
        V = [E]
    else if (P-1 or P-2 = <eu>, <tu>, <ele>, <ela>) or (P+1 = HN or A_IND) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 17
end if
```

Algorithm 17

```
if (Word is a homograph of the type 17) then
    if (WN_besta_E is on F-1, F0 or F+1) then
        V = [E]
    else if (WN_besta_e is on F-1, F0 or F+1) or (The homograph is inside the BC_besta_e) then
        V = [e]
    else
        V = [E]
    end if
else
    Go to Algorithm 18
end if
```

Algorithm 18

```
if (Word is a homograph of the type 18) then
    if (WN_sede_e is on F-1, F0 or F+1) or (The homograph is inside the BC_sede_e) then
        V = [e]
    else if (WN_sede_E is on F-1, F0 or F+1) or (The homograph is inside the BC_sede_E) then
        V = [E]
    else
        V = [E]
    end if
else
    Go to Algorithm 19
end if
```

Algorithm 19

```
if (Word is a homograph of the type 19) then
    if (WN_medo_e is on F-1, F0 or F+1) or (The homograph is inside the BC_medo_e) then
        V = [e]
    else if (WN_medo_E is on F-1, F0 or F+1) then
        V = [E]
    else
        V = [e]
    end if
else
    Go to Algorithm 20
end if
```

Algorithm 20

```
if (Word is a homograph of the type 20) then
    if (P-1 = <a> or <as>) then
        V = [E]
    else if (P-1 = <os>, <aos>, <nos>, <em>, <desses>, <destes>, <nesses>, <daqueles>, <daqueles>, <teus>, <seus>, <dos>, <cujos>, <meus>, <nestes>, <vossos>, <nossos>, <mesmos> or <esses>) then
        V = [e]
    else if (P-1 = <de> or EXPIMP) or (P+1 = HN, A_IND, P_POSS, P_DEM or P_IND) then
        V = [e]
    else
        V = [e]
    end if
else
    Go to Algorithm 21
end if
```

Algorithm 21

```
if (Word is a homograph of the type 21) then
    if (WN_cor_o is on F-1, F0 or F+1) or (The homograph is inside the BC_cor_o) then
        V = [o]
    else if (The homograph is inside the BC_cor_O) then
        V = [O]
    else
        V = [O]
    end if
else
    Go to Algorithm 22
end if
```

Algorithm 22

```
if (Word is a homograph of the type 22) then
    if (WN_lobo_o is on F-1, F0 or F+1) or (The homograph is inside the BC_lobo_o) then
        V = [o]
    else if (WN_lobo_O is on F-1, F0 or F+1) or (The homograph is inside the BC_lobo_O) then
        V = [O]
    else
        V = [o]
    end if
else
    Go to Algorithm 23
end if
```

Algorithm 23

```
if (Word is a homograph of the type 23) then
    if (WN_bola_O is on F-1, F0 or F+1) or (The homograph is inside the BC_bola_O) then
        V = [O]
    else if (WN_bola_o is on F-1, F0 or F+1) or (The homograph is inside the BC_bola_o) then
        V = [o]
    else
        V = [O]
    end if
else
    Exit
end if
```