A Stemming Algorithm for the Portuguese Language

Viviane Moreira Orengo
Christian Huyck

School of Computing Science
Middlesex University
The Burroughs, London NW4 4BT
[v.orengo, c.huyck]@mdx.ac.uk

Abstract

Stemming algorithms are traditionally used in Information Retrieval with the goal of enhancing recall, as they conflate the variant forms of a word into a common representation. This paper describes the development of a simple and effective suffix-stripping algorithm for Portuguese. The stemmer is evaluated using a method proposed by Paice [9]. The results show that it performs significantly better than the Portuguese version of the Porter algorithm.

1 Introduction

Stemming is the process of conflating the variant forms of a word into a common representation, the stem. For example, the words: "presentation", "presented", "presenting" could all be reduced to a common representation "present". This is a widely used procedure in text processing for information retrieval (IR) based on the assumption that posing a query with the term *presenting* implies an interest in documents containing the words *presentation* and *presented*.

There are several studies evaluating the validity of the stemming process for IR and they have reached contrasting conclusions: [2] examined the effects of 3 algorithms on 3 test collections and found no improvements on the retrieval performance since the number of queries with improved performance tended to equal the number with poorer performance. However, in [6] stemming improved retrieval performance by up to 35% on some collections. Finally, after exhaustive analysis [4] concludes that "some form of stemming is almost always beneficial", he found that the overall improvement ranged from 1-3% but, for many individual queries, stemming made a large difference. All those experiments were done on English collections and it seems possible that highly inflected languages such as Portuguese may benefit more from stemming.

The mistakes associated with the stemming process can be divided in two groups:

- Overstemming: when the string removed was not a suffix, but part of the stem. This can result in the conflation of unrelated words.
- Understemming: when a suffix is not removed. This will cause a failure in conflating related words.

The stemming of English seems to be a resolved problem, the Porter Stemmer [10] is a simple suffix-stripping algorithm that is based solely on rules, without exception lists or dictionary lookups, however it has proved to be as effective as more elaborated systems. Similar algorithms have been developed for other languages [3, 5, 11]. Our challenge is to design a suffix-stripping algorithm that is both simple and effective with the target of improving recall, without decreasing precision. This paper describes the development of an effective stemmer for Portuguese.

The remainder of this paper is organised as follows: section 2 describes the algorithm; section 3 comments on the biggest difficulties faced; section 4 shows the results of the tests carried out on the stemmer and finally section 5 presents the final conclusions.

2 The Algorithm

The algorithm was implemented in C and is composed by 8 steps that need to be executed in a certain order. Figure 1 shows the sequence those steps must obey:

0-7695-1192-9/01 $10.00 © 2001 IEEE

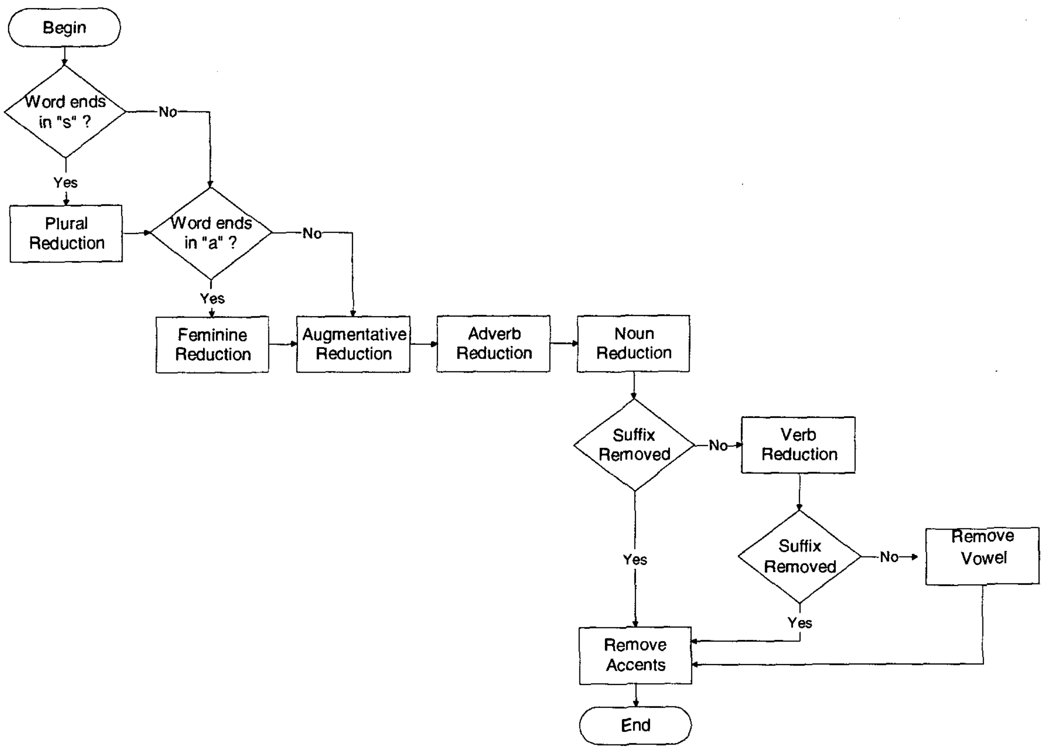

Figure 1. Sequence of steps for the Portuguese Stemmer

Each step has a set of rules, the rules in the steps are examined in sequence and only one rule in a step can apply. The longest possible suffix is always removed first because of the order of the rules within a step, e.g. the plural suffix *–es* should be tested before the suffix *–s*. The rules were created based on the most common Portuguese suffixes presented in [1,7]. The validity of such rules was checked using the evaluation methods presented in section 4. At the moment, the Portuguese Stemmer contains 199 rules, please refer to the Appendix for the complete list.

Each rule states:

- The suffix to be removed;
- The minimum length of the stem: this is to avoid removing a suffix when the stem is too short. This measure varies for each suffix, and the values we set by observing lists of words ending in the given suffix. Although there is no linguistic support for this procedure it reduces overstemming errors.
- A replacement suffix to be appended to the stem, if applicable;
- A list of exceptions: for nearly all rules we defined, there were exceptions, so we added exception lists for each rule. Such lists were constructed with the aid of a vocabulary of 32,000 Portuguese words freely available at [8]. Tests with the stemmer have shown that exceptions list reduce overstemming mistakes by 5%.

An example of a rule is:

```
"inho", 3, "", {"caminho", "carinho",
"cominho", "golfinho", "padrinho",
"sobrinho", "vizinho"}
```

Where *"inho"* is a suffix that denotes diminutive, 3 is the minimum size for the stem, which prevents words like *"linho"* (linen) from being stemmed and the words between brackets are the exceptions for this rule, that is, they end in the suffix but they are not diminutives. All other words that end in *–inho* and that are longer than 6 characters will be stemmed. There is no replacement suffix in this rule.

Note that the stems do not have to be linguistically meaningful, since they are used to index a database of documents and are not presented to the user. They need however, to capture the word meaning without losing too much detail.

Step 1: Plural Reduction

With rare exceptions, the plural forms in Portuguese end in *–s*. However, not all words ending in *–s* denote plural, e.g. *lápis*, (pencil). This step consists basically in removing the final "*s*" of the words that are not listed as exceptions. Yet sometimes a few extra modifications are needed e.g. words ending in *–ns* should have that suffix replaced by "*m*" like in *bons* → *bom*.

Step 2: Feminine Reduction

All nouns and adjectives in Portuguese have a gender. This step consists in transforming feminine forms to their corresponding masculine. Only words ending in –a are tested in this step but not all of them are converted, just the ones ending in the most common suffixes, e.g. *chinesa* → *chinês*.

Step 3: Adverb Reduction

This is the shortest step of all, as there is just one suffix that denotes adverbs *–mente*. Again not all words with that ending are adverbs so an exception list is needed.

Step 4: Augmentative/Diminutive Reduction

Portuguese nouns and adjectives present far more variant forms than their English counterparts. Words have augmentative, diminutive and superlative forms e.g. "small house" = *casinha*, where *–inha* is the suffix that indicates a diminutive. Those cases are treated by this step. According to [1] there are 38 of these suffixes, however some of them are obsolete therefore, in order to avoid overstemming, our algorithm uses only the most common ones that are still in common usage.

Step 5: Noun Suffix Reduction

This step tests words against 61 noun (and adjective) endings. If a suffix is removed here, steps 6 and 7 are not executed.

Step 6: Verb Suffix Reduction

Portuguese is a very rich language in terms of verbal forms, while the regular verbs in English have just 4 variations (e.g. talk, talks, talked, talking), the Portuguese regular verbs have over 50 different forms [7]. Each one has its specific suffix. The verbs can vary according to tense, person, number and mode. The structure of the verbal forms can be represented as: root + thematic vowel[1] + tense + person, e.g. *and* + *a* + *ra* + *m* (they walked). Verbal forms are reduced to their root.

[1] There are 3 classes of verbs in Portuguese according to the ending of their infinitive form: "ar", "er", "ir". Thematic Vowel the letter ("-a", "-e" and "-i") that groups verbs into categories.

Step 7: Vowel Removal

This task consists in removing the last vowel ("a", "e" or "o") of the words which have not been stemmed by steps 5 and 6, e.g. the word *menino* (boy) would not suffer any modifications by the previous steps, therefore this step will remove its final *–o*, so that it can be conflated with other variant forms such as *menina, meninice, meninão, menininho,* which will also be converted to the stem *menin*.

Step 8: Accents Removal

Removing accents is necessary because there are cases in which some variant forms of the word are accented and some are not, like in *psicólogo* (psychologist) and *psicologia* (psychology), after this step both forms would be conflated to *psicolog*. It is important that this step is done at this point and not right at the beginning of the algorithm because the presence of accents is significant for some rules e.g. *óis* → *ol* transforming *sóis* (suns) to *sol* (sun). If the rule was *ois* → *ol* instead, it would make mistakes like stemming dois (two) to *dol*.

3 Difficulties in Stemming Portuguese

Stemming Portuguese is much more troublesome than stemming English due to its more complex morphology. Below we discuss some particularly important difficulties:

- **Dealing with Exceptions:** As mentioned before one of the biggest difficulties in building this stemming algorithm was that for nearly every rule we formulated there are exceptions, e.g. *"ão"* is a commonly used suffix to denote augmentative, however not all words ending in *"ão"* are in augmentative forms. Therefore, unlike the Porter stemmer we had to use exceptions lists. If we had chosen not to use such lists, the stemmer would make overstemming errors if we kept the rule or understemming errors if we dropped it.
- **Homographs**[2]**:** There are several such cases, many involving conjugated verbs, e.g. *casais* which can mean "couples" or 2nd person plural of the verb "to marry". Our algorithm does not have information about word categories, so the different senses of those words are not distinguished. For this specific case, the stemmer assumes the first meaning and stems the word to its singular form *casal*, this is due

[2] Homographs are words which are spelled identically but which have different meanings.

to the 2nd person plural being nearly obsolete in modern Portuguese.

- **Irregular verbs:** The current version of the stemmer is not treating irregular verbs, but surprisingly they do not seem to seriously affect our results. The tests have shown that less than 1% of the mistakes occur because of this reason.
- **Changes to the morphological root:** There are cases in which the inflection process changes the root of the word. The cases in which the change obeys orthographic rules (e.g. *ns* $\rightarrow$ *m*) are being successfully treated. As for the other cases, we are still looking for an effective way of dealing with them. At the moment, words like *emitir* (to emit) and *emissão* (emission), which are semantically related, are not being conflated, as the first is stemmed to *emit* and the later to *emis*.
- **Stemming proper names:** Proper names should not be stemmed, the problem is recognising them. A list of proper names does not present an ideal solution for two main reasons: there are infinite possibilities and some proper names are shared by names of things, e.g. *Pereira* is a common Portuguese surname but it also means "pear tree". As the Porter stemmer, the present implementation of our algorithm is stemming proper names.

4 Evaluating Our Stemmer

In order to assess the effectiveness of our stemmer we carried out a sequence of tests. Those tests used a vocabulary of 32,000 distinct word forms obtained from [8]. We tested our algorithm against the Portuguese version of the Porter stemmer [8] applied to the same vocabulary. The main differences between the two stemmers are outlined below:

- Different morphology: the Portuguese version of the Porter Algorithm is just a translation of the original English version, consequently it does not treat cases like augmentative and feminine forms since those do not occur in English. Our algorithm was exclusively designed for Portuguese, so it deals with issues that are specific to the Portuguese morphology.
- Accented characters: the Portuguese version of the Porter stemmer uses an intermediate representation for all accented characters (e.g. á is treated as a^a). Our stemmer treats all accented characters as they are.
- Exceptions: our stemmer includes al list of exceptions for each rule as we found they help avoiding overstemming errors.
- The steps composing the two algorithms are diferent.

The next sections describe the tests we carried out using the two stemmers.

4.1 Vocabulary Reduction

One of the original purposes of suffix-stripping was reducing the size of the vocabulary for indexing purposes. Porter reports a reduction of about a third of the vocabulary, using his stemmer over 10,000 different English words. The Portuguese version of the Porter stemmer reduces the vocabulary by 44 %; this is because Portuguese has far more variant forms than English. Our stemmer reduces the vocabulary by 51%.

4.2 Comparison Against Expected Output

This was the method used aid the design of our stemmer, we randomly selected 2800 words from our vocabulary and manually assigned for each one its correct stem. We then tested the calculated stems against the expected output and analysed the errors to check if new rules or exceptions were needed. However, there is a stage on the designing of a stemming algorithm in which the addition of new rules with the target of avoiding errors on a specific case causes inaccuracies in other instances. At the end of the design process our stemmer was achieving 98% precision on the training vocabulary.

We then decided to evaluate its performance using a set of words that had not been used in the training, so we selected another 1000 words and again assigned for each one its correct stem. Our algorithm calculated the right stem 96% of the times outperforming the Portuguese version of the Porter stemmer which calculated the right stem 71% of the times, for the same test set. We are aware that this was a very small test set and we intend to repeat this experiment on a larger sample.

4.3 Paice's Evaluation Method

The standard evaluation method of a stemming algorithm is to apply it to an IR test collection and calculate recall and precision to assess how the algorithm affects those measurements. However, this method does not show the specific causes of errors therefore it does not help the designers to optimise their algorithms. [9] proposes an evaluation method in which stemming is assessed against predefined groups of semantically related words. He introduced three new measurements: over and understemming index and the stemming weight.

This test requires a sample of different words partitioned into concept groups containing forms that are morphologically and semantically related to one another. The perfect stemmer should conflate all words in a group to the same stem and that stem should not occur in any other group. Section 4.3.1 describes the method and section 4.3.2 shows the results achieved by our stemmer and the Portuguese version of Porter's algorithm.

4.3.1 Method Description. For each concept group two totals are computed:

- Desired merge total (DMT), which is the number of different possible word form pairs in the particular group, and is given by the formula:

$$DMT_g = 0.5\ n_g\ (n_g - 1)$$

where n_g is the number of words in that group.

- Desired Non-merge Total (DNT), which counts the possible word pairs formed by a member and a non-member word and is given by the formula:

$$DNT_g = 0.5\ n_g\ (W - 1)$$

where W is the total number of words. By summing the DMT for all groups we obtain the GDMT (global desired merge total) and, similarly, by summing the DNT for all groups we obtain the GDNT (global desired non-merge total). After applying the stemmer to the sample, some groups still contain two or more distinct stems, incurring in understemming errors. The Unachieved Merge Total(UMT) counts the number of understemming errors for each group and is given by:

$$UMT_g = 0.5 \sum_{i=1..s} u_i\ (n_g - u_i)$$

where s is the number of distinct stems and ui is the number of instances of each stem. By summing the UMT for each group we obtain the Global Unachieved Merge Total (GUMT). The understemming index (UI) is given by: GUMT/GDMT.

After stemming, there can be cases where the same stem occurs in two or more different groups, which means there are overstemming errors. By partitioning the sample into groups that share the same stem we can calculate the Wrongly Merged Total (WMT), which counts the number of overstemming errors for each group. The formula is given by:

$$WMT_g = 0.5 \sum_{i=1..t} v_i\ (n_s - v_i)$$

where t is the number of original groups that share the same stem, n_s is the number of instances of that stem and v_i is the number of stems for each group t. By summing the WMT for each group we obtain the Global Wrongly-Merged Total (GWMT). The overstemming index (OI) is given by: GWMT/GDNT. The Stemming Weight (SW) is given by the ratio OI/UI.

Here is one example, supposing we had the following sample divided into 5 concept groups:

1. ajud (help): ajuda, ajudando, ajudinha, ajudei
2. duvid (doubt): duvido, dúvida, duvidamos, duvidem
3. chec (to check): checando, chequei, checamos, checou
4. beb (to drink): bebo, bebes, bebi, bebendo, bêbado, bebida
5. bebê (baby): bebê, bebezinho

Calculating GDMT, and GDNT, we have:

Group	DMT	DNT
1	6	32
2	6	32
3	6	32
4	15	37.5
5	1	9.5
Totals	34	143

After the stemming process we have:

1. ajud, ajud, ajud, ajud
2. duvid, duvid, duvid, duvid
3. chec, chequ, chec, chec
4. beb, beb, beb, beb, beb, beb
5. beb, beb

We have two kinds of mistake in this case: the words in group 3 were not conflated to the same stem (understemming), and words in groups 4 and 5 were stemmed to the same form (overstemming). Calculating GUMT we have:

Group	UMT
1	0
2	0
3	3
4	0
5	0
GUMT	3

Now we reorganise the sample into groups that share the same stem, in order to calculate GWMT:

1. ajud(1), ajud(1), ajud(1), ajud(1)
2. duvid(2), duvid(2), duvid(2), duvid(2)
3. chec(3), chec(3), chec(3)
4. chequ(3)
5. beb(4), beb(4), beb(4), beb(4), beb(4), beb(4), beb(5), beb, (5)

Group	WMT
ajud	0
duv	0
chec	0
chequ	0
beb	12
GWMT	12

The Understemming Index (UI) given by GUMT/GDMT is 0.088 The Overstemming Index (OI) given by GWMT/GDNT is 0.083. The Stemming Weight (SW) given by OI/UI is 1.06. Paice's method does not offer a framework to assess the values of UI and OI, however it suggests that a one stemmer will be better than another in terms of UI and worse in terms of OI. We disagree with that statement as one stemmer is better than the other if both OI and UI present lower values, that is, if it makes less overstemming and less understemming errors.

4.3.2 Comparison between the Two Stemmers. In order to compare the performance of the two Portuguese stemmers we selected another sample of 1000 words divided into 170 groups. The results obtained were:

- For the Portuguese version of the Porter stemmer:

  UI: 0.215

  OI: 2.11×10^{-4}

  SW: 9.81×10^{-4}

- For our algorithm:

  UI: 0.034

  OI: 9.85×10^{-5}

  SW: 2.89×10^{-3}

The results show that our stemmer makes less understemming errors and less overstemming errors that the Portuguese version of Porter's Algorithm, thus we can conclude that our stemmer is more efficient. Again the test was done over a small sample of words and we intend to repeat it over a larger test set.

5 Conclusion

This paper presented the development of a stemming algorithm for Portuguese. The algorithm is simple yet highly effective, it is based on a set of steps composed by a collection of rules. Each rule specifies the suffix to be removed; the minimum length allowed for the stem; a replacement suffix, if necessary, and a list of exceptions.

The stemmer was evaluated using 3 different methods, including the one proposed by Paice [9] and outperformed the Portuguese version of the Porter stemmer in all tests. Future work will include using the Portuguese stemmer on an Information Retrieval system to assess its impact over recall and precision.

References

[1] Cunha, C. and Lindley-Cintra, L. **Nova Gramática do Português Contemporâneo**. ed. Nova Fronteira, Rio de Janeiro, 1985, 719 p.

[2] Harman, D. How effective is suffixing? In: **Journal of the American Society for Information Science** 42(1), 1991, p. 7-15.

[3] Honrado, A. et al. A Word Stemming Algorithms for the Spanish Language. In: Proceedings of **the String Processing and Information Retrieval Conference (SPIRE)**, A Coruña, Spain, Sept. 27-29, 2000.

[4] Hull, D.A. Stemming Algorithms: A Case Study for Detailed Evaluation. In: **Journal of the American Society for Information Science** 47(1), 1996, p. 70-84.

[5] Kraaij, W. and Pohlmann, R. **Porter's Stemming Algorithm for Dutch**. In L.G.M. Noordman and W.A.M. de Vroomen, editors, *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pp. 167-180, Tilburg, 1994. http://www-uilots.let.uu.nl/uplift/stinfon.ps

[6] Krovetz, R. Viewing morphology as an inference process. In: **ACM SIGIR Conference on Research and Development in Information Retrieval**, 1993, pp. 191-202.

[7] Macambira, J.R. **A Estrututa Morfo-Sintática do Português.** Ed. Pioneira, São Paulo, 1999.

[8] http://open.muscat.com

[9] Paice, C.D. An Evaluation Method for Stemming Algorithms. In: **ACM SIGIR Conference on Research and Development in Information Retrieval**, 1994, pp. 42-50.

[10] Porter, M.F. An Algorithm for Suffix Stripping. **Program,** 14(3), 130-137.

[11] Wechsler, M.; Sheridan, P.; Schäuble, P. Multi-Language Text Indexing for Internet Retrieval. Proceedings of **the 5th RIAO Conference Computer Assisted Information Searching on the Internet**. Montreal-Canada, June, 1997.

Appendix – Suffix Stripping Rules

Those are the stemming rules. The exceptions were omitted due to limited space.

Plural Reduction Rules

Suffix to Remove	Minimum Stem Size	Replacement	Example
"ns"	1	"m"	bons → bom
"ões"	3	"ão"	balões → balão
"ães"	1	"ão"	capitães → capitão
"ais"	1	"al"	normais → normal
"éis"	2	"el"	papéis → papel
"eis"	2	"el"	amáveis → amável
"óis"	2	"ol"	lençóis → lençol
"is"	2	"il"	barris → barril
"les"	3	"l"	males → mal
"res"	3	"r"	mares → mar
"s"	2		casas → casa

Feminine Reduction Rules

Suffix to Remove	Minimum Stem Size	Replacement	Example
"ona"	3	"ão"	chefona → chefão
"ã"	2	"ão"	vilã → vilão
"ora"	3	"or"	professora → professor
"na"	4	"no"	americana → americano
"inha"	3	"inho"	chilena → chileno
"esa"	3	"ês"	sozinho → sozinha
"osa"	3	"oso"	inglesa → inglês
"íaca"	3	"íaco"	famosa → famoso
"ica"	3	"ico"	maníaca → maníaco
"ada"	2	"ado"	prática → prático
"ida"	3	"ido"	mantida → mantido
"ída"	3	"ido"	cansada → cansado
"ima"	3	"imo"	prima → primo
"iva"	3	"ivo"	passiva → passivo
"eira"	3	"eiro"	primeira → primeiro

Adverb Reduction Rule

Suffix to Remove	Minimum Stem Size	Replacement	Example
"mente"	4		felizmente → feliz

Augmentative/Dimunutive Reduction Rules

Suffix to Remove	Min	Replacement	Example
"díssimo"	5		cansadíssimo → cansad
"abilíssimo"	5		amabilíssimo → ama
"íssimo"	3		fortíssimo → fort
"érrimo"	4		chiquérrimo → chiqu
"zinho"	2		pezinho → pe
"quinho"	4	"c"	maluquinho → maluc
"uinho"	4		amiguinho → amig
"adinho"	3		cansadinho →cansad
"inho"	3		carrinho → carr
"alhão"	4		grandalhão→ grand
"uça"	4		dentuça → dent
"aço"	4		ricaço→ ric
"adão"	4		Casadão→ cans
"ázio"	3		corpázio→ corp
"arraz"	4		pratarraz→ prat
"arra"	3		bocarra → boc
"zão"	2		calorzão→ calor
"ão"	3		meninão → menin

Noun Reduction Rules

Suffix to Remove	Min Size	Repl	Example
"encialista"	4		existencialista → exist
"alista"	5		minimalista → minim
"agem"	3		contagem → cont
"iamento"	4		genrenciamento → gerenc
"amento"	3		monitoramento →monit
"imento"	3		nascimento → nasc
"alizado"	4		comercializado→comerci
"atizado"	4		traumatizado → traum
"izado"	5		alfabetizado→alfabet
"ativo"	4		associativo → associ
"tivo"	4		contraceptivo→contracep
"ivo"	4		esportivo→esport
"ado"	2		abalado → abal
"ido"	3		impedido → imped
"ador"	3		ralador → ral
"edor"	3		entendedor→entend
"idor"	4		cumpridor→cumpr
"atória"	5		obrigatória→obrig
"or"	2		produtor→produt
"abilidade"	5		comparabilidade→compar
"icionista"	4		abolicionista→abol
"cionista"	5		intervencionista→interven
"ional"	4		profissional→profiss
"ência"	3		referência→refer
"ância"	4		repugnância→repugn
"edouro"	3		abatedouro→abat
"queiro"	3	"c"	fofoqueiro→fofoc
"eiro"	3		brasileiro→brasil
"oso"	3		gostoso→gost
"alizaç"	5		comercializaç→comerci
"ismo"	3		consumismo→consum
"izaç"	5		concretizaç→concret
"aç"	3		alegaç→aleg
"iç"	3		aboliç→abol
"ário"	3		anedotário→anedot
"ério"	6		ministério→minist
"ês"	4		chinês → chin
"eza"	3		beleza→ bel
"ez"	4		rigidez→rigid
"esco"	4		parentesco→parent
"ante"	2		ocupante→ocup
"ástico"	4		bombástico→bomb
"ático"	3		problemático→problem
"ico"	4		polêmico→polêm
"ividade"	5		produtividade→produt
"idade"	5		profundidade→profund
"oria"	4		aposentadoria→aposentad

"encial"	5		existencial→exist
"ista"	4		artista→art
"quice"	4	"c"	maluquice→maluc
"ice"	4		chatice→ chat
"íaco"	3		demoníaco→demon
"ente"	4		decorrente→decorr
"inal"	3		criminal→crim
"ano"	4		americano→ americ
"ável"	2		amável→ am
"ível"	5		combustível→combust
"ura"	4		cobertura→cobert
"ual"	3		consensual→consens
"ial"	3		mundial→mund
"al"	4		experimental→experiment

Verb Reduction Rules			
Suffix to Remove	**Min Size**	**Repl**	**Example**
"aríamo"	2		cantaríamo → cant
"ássemo"	2		cantássemo → cant
"eríamo"	2		beberíamo → beb
"êssemo"	2		bebêssemo → beb
"iríamo"	3		partiríamo → part
"íssemo"	3		partíssemo →part
"áramo"	2		cantáramo→ cant
"árei"	2		cantárei→ cant
"aremo"	2		cantaremo → cant
"ariam"	2		cantariam→ cant
"aríei"	2		cantaríei→ cant
"ássei"	2		cantássei→ cant
"assem"	2		cantassem→ cant
"ávamo"	2		cantávamo→ cant
"êramo"	3		bebêramo→ beb
"eremo"	3		beberemo→ beb
"eriam"	3		beberiam→ beb
"eríei"	3		beberíei→ beb
"êssei"	3		bebêssei→ beb
"essem"	3		bebessem→ beb
"íramo"	3		partiríamo→ part
"iremo"	3		partiremo→ part
"iriam"	3		partiriam→ part
"iríei"	3		partiríei→ part
"íssei"	3		partíssei→ part
"issem"	3		partissem→ part
"ando"	2		cantando→ cant
"endo"	3		bebendo→ beb
"indo"	3		partindo→ part
"ondo"	3		propondo→ prop
"aram"	2		cantaram→ cant
"arde"	2		cantarde → cant
"arei"	2		cantarei→ cant
"arem"	2		cantarem→ cant
"aria"	2		cantaria→ cant
"armo"	2		cantarmo→ cant
"asse"	2		cantasse → cant
"aste"	2		cantaste→ cant
"avam"	2		cantavam→ cant
"ávei"	2		cantávei→ cant
"eram"	3		beberam→ beb
"erde"	3		beberde→ beb
"erei"	3		beberei→ beb
"êrei"	3		bebêrei→ beb
"erem"	3		beberem→ beb
"eria"	3		beberia→ beb
"ermo"	3		bebermo→ beb
"esse"	3		bebesse→ beb
"este"	3		bebeste→ beb
"íamo"	3		bebíamo→ beb
"iram"	3		partiram→ part
"íram"	3		concluíram→ conclu
"irde"	2		partirde→ part
"irei"	3		partírei→ part
"irem"	3		partirem→ part
"iria"	3		partiria→ part
"irmo"	3		partirmo→ part
"isse"	3		partisse→ part
"iste"	4		partiste→ part
"amo"	2		cantamo→ cant
"ara"	2		cantara→ cant
"ará"	2		cantará→ cant
"are"	2		cantare→ cant
"ava"	2		cantava→ cant
"emo"	2		cantemo→ cant
"era"	3		bebera→ beb
"erá"	3		beberá→ beb
"ere"	3		bebere→ beb
"iam"	3		bebiam→ beb
"íei"	3		bebíei→ beb
"imo"	3		partimo→ part
"ira"	3		partira→ part
"irá"	3		partirá→ part
"ire"	3		partire→ part
"omo"	3		compomo→ comp
"ai"	2		cantai→ cant
"am"	2		cantam→ cant
"ear"	4		barbear→barb
"ar"	2		cantar→ cant
"uei"	3		cheguei → cheg
"ei"	3		cantei→ cant
"em"	2		cantem→ cant
"er"	2		beber→ beb
"eu"	3		bebeu→ beb
"ia"	3		bebia→ beb
"ir"	3		partir→ part
"iu"	3		partiu→ part
"ou",	3		chegou→ cheg
"i"	3		Bebi→ beb

Vowel Removal Rules			
Suffix to Remove	**Minimum Stem Size**	**Replacement**	**Example**
"a"	3		menina → menin
"e"	3		grande → grand
"o"	3		menino → menin