PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 26/08

A Verbal Stemmer for the Brazilian Portuguese Language

Akeo Tanabe

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 26/08 **ISSN: 0103-9741**
Editor: Prof. Carlos José Pereira de Lucena **August,2008**

A Verbal Stemmer for the Brazilian Portuguese Language

Akeo Tanabe[1]

[1]LES-Laboratório de Engenharia de Software (PUC-Rio)

akeo@les.inf.puc-rio.br

Abstract. This paper describes a Verbal Stemmer for the Brazilian Portuguese idiom, with recognition of verbal forms playing a main role in this process. After a word has been recognized as an accurate verbal form, the process outputs the corresponding infinitive form of the verb. The infinitive form is the correct stem for the verbal form. Portuguese verbal recognition is a hard task for a computer, because the idiom is very rich in inflected verbal forms, with many irregularities. A painstaking task of constructing a knowledge base for the recognition of Brazilian Portuguese verbs by computer has been accomplished. With more than 4,000 verbs it has a broad coverage when considering words currently in use.

Keywords: verbal stemmer; verbal form recognition; Brazilian Portuguese language

Resumo. Este artigo descreve o processo de descobrir a raiz de uma flexão verbal para o português praticado no Brasil. Uma palavra sendo reconhecida como uma flexão verbal válida, o mecanismo fornece, como raiz, o infinitivo do verbo. A mecânica de reconhecimento de uma flexão verbal é a parte mais importante do processo. A língua portuguesa é muito rica em flexões verbais e apresenta muitas irregularidades, o que dificulta o seu reconhecimento por computador. Mas os instrumentos para esta tarefa já estão concluídos e o processo reconhece mais de 4.000 verbos diferentes, conferindo boa cobertura para textos comuns.

Palavras-chave: raiz de flexão verbal; reconhecimento de forma verbal; Língua Portuguesa do Brasil

* This work has been sponsored by the LES- Laboratório de Engenharia de Software (PUC-Rio)

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

1. Introduction

The Orengo and Huyck[2001] paper, presented at the 8th International Symposium of String Processing and Information Retrieval (SPIRE 2001), sets a milestone in the stemming process, being the first specific for the Portuguese language. The Viera and Virgil [2007] paper did a profound multi-lingual analysis of algorithms used in stemming process. They concluded that the Orengo and Huyck algorithm needs improvement. The process of just stripping off the suffix part of a word, even when considering exception rules, is not an adequate approach. Word category awareness is a must.

The Portuguese language is very rich in verbal inflected forms, with many irregularities. Considering just this category, more than two hundred verbal endings are needed to take account of all this diversity(see Appendix 2). However, Orengo and Huyck's [2001] process uses a hundred word endings for the entire universe of Portuguese words. To the best of our knowledge, there is no tool dealing with verbal inflected forms.

This paper presents a program for computer recognition of Brazilian Portuguese verbal forms and its offspring: the verbal stemmer. The verbal stemmer is a unique tool, because it puts ordinary consulting process upside-down: starting from a word, after recognizing it as a valid verbal form, returns the corresponding infinitive form.

Providing computer expertise for verbal form recognition is a painstaking and time consuming task. Even with previous experience, the present version demanded more than a man-year of work. With more than 4,000 verbs it has a broad coverage when considering words currently in use.

The remainder of this paper is structured as follows. Section 2 presents the general concepts to describe the knowledge base architecture. Section 3 describes the recognition process, detailing its three steps. Section 4 shows, didactically, the recognition process, by means of well chosen examples. Section 5 shows how to use the "verbalstemmer" program. The last section is ended with concluding remarks. Appendixes 2 and 3 show the recognition of verbal form graphically.

2. Knowledge base architecture and concepts

2.1 General concepts

The main component for verbal forms recognition is a knowledge base composed by a collection of generation rules. Currently, the generation rule is a coded representation of a set of verbal forms, so thereafter it will be referred as "coded knowledge base". Each rule is made up of a pair (a stem and a number). The number identifies a sub-set of verbal terminations. The generation rule just states: append all allowed terminations to the stem.An ordinary regular verb has only one associated rule; irregular verbs can have several.

The verbal terminations were grouped into 8 classes. Each group is a collection of terminations used by a class of verbs. Each group is divided into 7 partitions, with elements of the same length being placed together. Each group represents a set of verbal terminations, from which sub-sets are generated. Irregular verbs use these sub-sets, but only ordinary regular verbs can use the entire set.

Appendix 1 illustrates sub-sets of the third group. Each allowed terminations are underlined. Each element has a value of a power of 2 associated to it, the first one with value 1 (from left to right).

The values of the underlined terminations are added and the sum is assigned to the partition. Each number can represent at most 7 elements. Partitions containing more than 7 elements can have additional numbers. Partitions of length 2, 4 and 5 characters, of the third group, (see Appendix 1) need two integers each for their mapping.

The sub-set is represented by a list of integers, gathered from the partitions. A 3-digit number identifies this list of integers, with the most significant digit representing the group and the last two, the ordinal number of the sub-set in the group. 301 and 302 are identifiers, and 01 and 02 are sub-sets of the third group, with 300 being the first sub-set (see Appendix 1).

This coded knowledge base for Brazilian Portuguese verbal recognition comprises 93 of such sub-sets.

2.2 Some irregular verbs oddities

Some irregular verbal forms cannot be generated by adding terminations to the stem, because their own stems are valid verbal form. To cover such oddities, a complementary knowledge base with plain verbal form representation was created. This base will be referred as the plain knowledge base.

3. Recognition process

For the recognition of a word as an accepted verbal form one must look at the both knowledge bases. The plain knowledge base is considered first. If the word has a match with a registered plain form, the recognition is over. The coded knowledge base is consulted only if a match is not found.

A search in a coded knowledge base is a three-step process.

The first step - The coded knowledge base is consulted, looking for a match between a stem and the beginning of the word. As the stems are scanned sequentially, the ordinal number of the stem in the knowledge base, which matches with the word beginning, is obtained.

An ordinal number of the stem, concordant with the beginning of the word is the first information obtained. With this information all other related information can be obtained: the length of the matched stem and the numerical sub-set identification.

With the known length of the concordant stem, the word terminations and its length can be determined and used in the next step.

The second step - In the second step, the word terminations must be searched in the group. The group is represented by the most significant digit of sub-set identification. This search is made considering all the terminations of the group.

If a match is found, the process provides the ordinal number of the verbal termination in the partition searched, ending the second step.

The third step - A real look inside the sub-set is conducted in this step. The ordinal number of the termination in the partition searched marks the location where the final confirmation must be made: if the termination at that location is an allowed verbal termination (underlined termination in Appendix 1), a full match is achieved, so the word can be considered an accurate verbal form.

The last step - After a word is recognized as a valid verbal form, the "verbalstemmer" must determine the corresponding infinitive form. This information is related with the matched stem obtained in the first step. The program print a line showing the infinitive form and the accepted verbal form, ending the search for one word, and going back to repeat the process for the next words.

4. Illustrative examples

Two words, "mentais" and "mintais", are used as an illustration for the recognition process. Neither word is present in the plain knowledge base, so the search is conducted only in the coded knowledge base.

The search in the coded knowledge base is sequential, but the entire base need not be scanned, even for an unsuccessful operation. The base is alphabetically arranged, so only a specific partition needs to be scanned. In the program, the first character of the stem partitions the base: all stems beginning with a particular letter are elements of this partition.

All the numbers presented in the examples are actual values considering the instances of knowledge bases used by the "verbalstemmer" program.

4.1 Search looking for a match for "mentais"

The first character of the word, "m" drives the search to the partition of the coded knowledge base identified by this character. The character "n" identifies the next partition.

The two values of interval are the ordinals 3,352 (starting value) and 3,537 (ending value). So the sequential search must be made, at most, in 185 stem instances (instead of more than 5,000). (see Appendix 2)

The first step - The "ment" stem that is the match for "mentais" can be found at the ordinal position 3,448, after executing 96 searches. All related information can be obtained using this number as an index. The related information is the numerical sub-set identification and length of the matched stem. There are 302, stems with length of 4 characters.

This finding shows that "ais" is the termination to be searched in the third group (because of the digit 3 of sub-set identification 302) and in the sub-set partition with terminations of length 3.

The second step - The search for "ais" in the third group in the partition with 3 characters is successful. In this search, the program is aware that the termination "ais" occurs as the first element of this partition.

The third step - To decide if "mentais" is an accurate verbal form, a last search is made, now in the sub-set of verbal terminations with identification 302. But the first element of the third partition is not underlined. So, "mentais" is not a verbal form.

4.2 Search looking for a match for "mintais"

All the steps are exactly analogous to the preceding case. In this case however the findings are as follows.

The first step - In the interval of starting index of 3,352 and ending index of 3,538, the stem "mint" can be found at the 3,481th position. The corresponding sub-set of verbal terminations is 301, driving the search to the third group, looking for the word ending "ais".

The second step - As in the preceding case, "ais" must be searched in the third group, in the same partition. So the answer is the same, saying that this termination is the first of this partition.

The third step - To decide if "mintais" is an accurate verbal form, a last search is made, now in the sub-set of verbal terminations with identification 301, looking if the first element of the third partition is underlined. As the answer is true, "mintais" is an accurate verbal form.

The infinitive form - As "mintais" was considered a valid verbal form, the last task of the "verbalstemmer" is to obtain the corresponding infinitive form.

Using the number 3,481 as an index in the related information repository, a number with value 2,661 is obtained. This value is the ordinal number of the infinitive form associated with the stem "mint". Using this last information as an index, an infinitive form "mentir" can be rescued, finishing a successful search.

After printing "`mentir#mintais`" as output, the program continues the iteration with the next word to be searched.

5. Operating guide for "verbal stemmer"

5.1 Syntax

The program can be run in an MS/DOS window of the Microsoft XP platform.

The program is called by a DOS command of the form:

```
verbalstemmer   <options> <input file>
```

where:

<options> - an option is specified by a hyphen followed by a character.

-r with this specification the program is transformed into a verbal form filter.

The "r" is an order to remove the verbal forms.

-u with this specification the program outputs a message showing how to use it

Without option specification the program acts as a verbal form recognizer.

<input file>

Any plain text files can be an input file for the program. As the program is case sensitive, and the text information in the knowledge base is always stored in lower case, it is advisable that the text be transformed into lower case text before being submitted to it.

<output file>

The full output of the program is directed to the "standard output" of a C language program. Since the word "mintais" is a valid verbal form and "mentir" is its infinitive form, the "verbalstemmer" outputs:

```
mentir#mintais
```

5.2 The program as a filter

With **(-r)** option specification, the program acts as a filter. The paragraph structure is preserved, but delimiter characters (mainly punctuation characters) are stripped off. The words that are not verbal forms are outputted in the original order, with a blank character in between.

6. Conclusion

The verbal stemmer only recognizes accepted forms of verbs included in the knowledge bases. For recognized verbal form, the program answers with the correct infinitive form.

The infinitive verbs "poder" and "podar" share some equal verbal forms (for instance, "pode" and "podem"). Some of these common verbal forms were included as plain verbal forms of verb "poder". As plain verbal forms have precedence over coded verbal forms, these are considered as having "poder" as the infinitive form.

Another known case of distinct verbs sharing identical verbal forms is of "ir" and "ser" (for instance "fui", "foi"). The program associates these verbal forms to the infinitive "ir", the first verb found in the knowledge base (because alphabetical order).

To maintain a vocabulary is a never-ending task. There will always be missing verbs and mistakes in programs, returning incorrect answers.

Please notify the author about such errors and of missing verbs so that the program can be continuously updated.

Source code of "verbalstemmer" can be obtained at URL address xxxxx.

References

1. VIERA, A.F.G. & VIRGIL, J.(2007) "Uma revisão dos algoritmos de radicalização em língua portuguesa" Information Research, Vol 12, nº 3, april 2007, Paper 315. Available at http://InformationR.net/ir/12-3/paper315.html
2. ORENGO, V.M. & HUYCK, C.(2001) "A Stemming Algorithm for the Portuguese Language" String Processing and Information Retrieval(SPIRE 2001) Proceedings. 8th International Symposium 13-15 Nov 2001 - pp. 186-193, Chile, 2001
3. BARBOSA, O. (1961) "Dicionário de Verbos da Língua Portuguesa" EDIOURO - Editora TECNOPRINT Ltda. Rio de Janeiro, 1961.
4. BORBA, F.S. (1991). "Dicionário Gramatical de Verbos do Português Contemporâneo do Brasil" Editora Unesp, 2ª ed. 1991
5. FERNANDES, F. (1959) "Dicionário de Verbos e Regimes". Editora Globo, 4ª ed. Porto Alegre, 1959.
6. HOLANDA, A. B. (1999) "Novo Dicionário da Língua Portuguesa". NOVA FRONTEIRA, 3ª ed. Rio de Janeiro, 1999.
7. TANABE, A (1986) "Um algoritmo para conjugação de verbos irregulares da Língua Portuguesa". XIX Congresso Nacional de Informática, agosto 1986, Rio de Janeiro, pp. 483-485, Anais do XIX Congresso Vol I.

Appendix 1

- ***Sample of sub set of verbal endings of the 3rd Group***

a	e	i	o	í

am	as	em	es	ia	ir	is
iu	ís					

ais	iam	ias	ido	ira	irá	

amos	imos	indo	iram	iras	irei	irem
ires	iria	irás	irão	isse	iste	íeis

irdes	ireis	iriam	irias	irmos	issem	isses
istes	íamos	íreis				

iremos	iríeis	íramos	ísseis			

iríamos	íssemos					

302(6, 124,1, 62, 126,127, 127,7, 15, 0)

a	e	i	o	í

am	as	em	es	ia	ir	is
iu	ís					

ais	iam	ias	ido	ira	irá	

amos	imos	indo	iram	iras	irei	irem
ires	iria	irás	irão	isse	iste	íeis

irdes	ireis	iriam	irias	irmos	issem	isses
istes	íamos	íreis				

iremos	iríeis	íramos	ísseis			

iríamos	íssemos					

301(9, 3,0, 1, 1,0, 0,0, 0, 0)

**** accepted verbal endings are underlined**

Sample of registered entries in Knowledge Base of Verbs:

```
 sentir(sent   302, sint  301)
mentir(ment   302, mint  301)
```

Appendix 2
Samples of verbal knowledge base

Regular verbs

```
definir(defin  300)
programar(program  100)
vender(vend  200)
```

Phonetically regular verbs

```
abraçar(abraç  102,abrac  101)
interligar(interlig  102,interligu  101)
restringir(restring  302,restrinj  301)
significar(signific  102,signifiqu  101)
tecer(tec  202,teç  201)
```

Irregular verbs

```
aprazer(apraz  227,aprouv  205)
dizer(d  305,dig  201,diss  205,diz  228)
estar(est 108,est 501,estej 203,estiv 205)
fazer(f  105,faç  201,faz  228,fiz  208)
haver(h 401,haj 203,hav 211,houv 205)
ir(f 601,v 602)
mentir(ment 302,mint 301)
prazer(praz 230,prouv 231)
querer(queir 203,quer 209,quer 215,quis 208)
saber(sab 209,sab 216,saib 203,soub 205)
sentir(sent 302,sint 301)
ser(f 601,s 212,s 600,sej 203)
trazer(tr  105,trag  201,traz  229,troux  205)
ver(v  232,v  314,v  701,vej  201)
```

Numbers of Brazilian Portuguese Verbs

There are a total of 12,216 verbs, estimated in the decade of the 1990s, with nearly 6,000 in current use (reference [4])

Knowledge Bases of Brazilian Portuguese Verbs in numbers

- 8 groups of verbal terminations with 355 elements, with repetition
- 215 distinct verbal terminations
- 93 patterns of verbal terminations (regular and irregular paradigms)
- 4,029 verbs - 5,282 stems
- 2,901 regular verbs
- 1,128 irregular verbs