Automatic syllabification of Portuguese

João Rodrigues, Francisco Costa, João Silva and António Branco
NLX-Natural Language and Speech Group of the
Departament of Computer Science of Universidade de Lisboa, Faculdade de Ciências

Abstract

This paper presents the first open-source implementation of an automatic process for the syllabification of Portuguese. To accomplish this, two approaches were adopted, implemented and compared: a rule-based and a machine learning one. The resulting tool is made freely available as an open-source software and as a free online linguistic service.

Keywords: automatic syllabification, rule-based, machine learning, open-source

1. Introduction

Although the syllabification of Portuguese may be regarded as a trivial task, we haven't found any scientific inquiry followed by a algorithmic solution and by its implementation available. We address these gaps with this paper.

We start by discussing syllabification, we continue by presenting the methods used to make this process automatic, the evaluation results of the implemented solution and we finalize with the concluding remarks.

2. Automatic syllabification

A syllable is a unit of spoken language consisting of a single uninterrupted pronunciation sound. Syllabification here consists in the separation of a written word in the sequences of letters representing its syllables.

Two forms of syllabic division can be accomplished:

- Translineation, where the words are divided at the end of lines, following specific rules for maintaining their correct pronunciation and comprehension.

- Phonetic base division, where common syllables division takes into account the phonetics of the words and their pronunciation.

Our objective was to implement and evaluate a Portuguese syllabifier for written words using a phonetic base division syllabification.

The need for syllabifiers arises from different fields such as: speech-recognition, text-to-speech or text readability, where some metrics as the Flesch-Kincaid one (Flesch, 1981) need to be fed by the outcome of a sillabifier. Though it is possible to find a couple of publications on the development of syllabifiers for Portuguese (Gouveia et al., 2000)(Oliveira et al., 2005), we could not find any automatic syllabifier available.

The tool we implemented receives a string as input, representing a word, and outputs a string with the syllibification's divisions marked with the extra character "|".

3. Methods

3.1 Rule-based

The development of the syllabifier using a rule-based algorithm took into account linguistic knowledge. It was supported with the specification provided in (Mateus et al., 2003).

The algorithm performs the following operations for each word:

1) Replacement of sub-string "qu" by "ŕ", "wh" by "ẃ" and "gu" by "ṕ" before a vowel.

2) Mark as candidate syllabification point every position in a word that occurs after a vowel.

3) Place consonants at the end of words with a possible syllabification.

4) Remove a candidate syllabification when it occurs with diphthongs taking into account exceptions (e.g. names with “-idade” derived from adjectives).

5) Separate consonants sequences except in "muta cum liquida" and in digraphs “nh”, “lh” and “ch”.

6) Restore the syllabification mark where “u” and “i” don't perform a diphthong with a vowel and before “nh”, “r”, “l”, “z” or nasal in the syllable end.

7) Mark as syllabification “ŕ” and “ṕ” if previous syllable has a diacritic.

8) Replace “ŕ” by “qu”, “ṕ” by “gu” and “ẃ” by “wh”.

3.2 Machine-learning methods

The machine learning method used n-grams as features to train a Naive Bayes Classifier and a Decision Tree Classifier implemented using NLTK (Bird et al., 2009). These automatic classifiers were trained and tested using a set of characteristics (features) with a corresponding binary class. The features used were the resulting grams from the slice of each word between each letter and storing the n-characters (n = 3) to its right and left, and also the concatenation of the first left and right character.

For example, from the word “palavra”, six instances of features are created since the word can be sliced in six places: p|a|l|a|v|r|a. For each of these instances, the features are extracted and in the training stage a class is assigned to the set, taking into the account the correct syllabification provided in the lexical database Porlex (Gomes et al., 2003). As an example of greater detail of the features, on the third word “slice”, between “l” and “a” the resulting instance of features is:

l3=**pal**, l2=**al**, l1=**a**, center=**la**, r1=**a**, r2=**av**, r3=**avr**

and looking at the Porlex we know that no syllabification occurs between these two letters, so the class gets the value Negative in the training phase. For the forth slice, between “a” and “v” the features are:

l3=**ala**, l2=**la**, l1=**a**, center=**av**, r1=**v**, r2=**vr**, r3=**vra**

and since a syllabification occurs, the class for these feature instance is assigned with the Positive value.

The main difference between the training and testing phase can be found in the class attribute for the features. While during training we provide a class to the features set, in the testing phase we expect the classifier's guess corresponding to the value of the class given a set of specific features.

4. Evaluation

For the evaluation of the syllabifier, the 27408 word lexical database was used, namely Porlex, containing the orthographic form and the corresponding syllabification for each word.

The evaluation of the tool proceeded by comparing the original syllabification in the database against the syllabification obtained from the algorithms for every word in the database. For word syllabification to be counted as correct, all of the syllables divisions needed to be correct.

The rule-base algorithm was evaluated with the full word lexical database and the machine learning methods evaluation was done with a k-fold cross validation using the full word list randomized and with a k of 10. The 10-fold cross validation consists in dividing the full word lexical database in 10 parts, iterating over each part for the evaluation of the classifier. For each iteration, the remaining parts not used for testing are used for training the classifier. The final accuracy score is obtained from the average of all the ten partial evaluations.

5. Results

With the rule based method, 80 words were incorrectly syllabified, resulting in a 99.7% accuracy over the test set.

Both machine learning methods obtained a worst result in comparison to the rule base method. Using the Decision Tree classifier a 86% accuracy was obtained. The Naive Bayes classifier, in turn, performed slight better with 90% accuracy, but still far from the rule based method score.

We thus opted for adopting the rule-based approach to implement our tool. The final version was supplemented by the list of 80 words which had been found to be wrongly syllabified, but now correctly syllabified.

6. Service Online and Distribution

The linguistic online service is available here lxsyllabifier.di.fc.ul.pt . The source code can be obtained from http://lxsyllabifier.di.fc.ul.pt/services/online_syllabifier/LX-Syllabifier.tar.gz .

7. Error analysis and final remarks

An accurate automatic Portuguese syllabifier was developed using a rule base method.

The 80 errors detected in implementation time, and afterwards solved, are related to cases where the orthography is not deterministic with respect to syllabification. The same pattern of letters corresponds to two different syllabification possibilities, and the choice between them cannot be fully determined by the word context. One example is the pair *reunião* and *feudal*. For *reunião* the target syllabification is **re|u|ni|ão**. For *feudal* the target syllabification is **feu|dal**. The sequence *eu* is ambiguous as to whether it represents a diphthong (as in *feudal*) or a sequence of two vowels (as in *reunião*). Other ambiguous sequences include *ui* (**ar|ru|i|nar** vs. **cui|dar**), *ei* (**ga|se|i|fi|car** vs **bei|jar**), *oi* (**boi|co|tar** vs. **pro|i|bir**) *ai* (**en|cai|xar** vs. **fa|is|car**), etc.

References

Mateus, Maria Helena Mira; Ana Maria Brito, Inês Duarte, Isabel Hub Faria, Sónia Frota, Gabriela Matos, Fátima Oliveira, Marina Vigário and Alina Villalva (2003). Gramática da Língua Portuguesa. 5Nd ed., Lisboa: Editorial Caminho, pp. 1037-1049.

Gouveia, Paulo, João Teixeira and Diamantino Freitas, 2000, "Divisão Silábica Automática do Texto Escrito e Falado", Actas do V PROPOR – Processamento Computacional da Língua Portuguesa Escrita e Falada, Atibaia, S. Paulo.

Oliveira, Catarina, Lurdes Castro Moutinho and António Teixeira, 2005, "On European Portuguese Automatic Syllabification", Proccedings of Interspeech 2005, pp.2933-2936.

Flesch, Rudolf. 1981. Statistical methods for rates and proportions. 2nd ed., New York: John Wiley, pp. 38-46.

Gomes, Inês and São Luís Castro, 2003. "Porlex, a lexical database in European Portuguese", Psychologica, 32, 91-108.

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.