Package ‘ptstem’

January 2, 2019

Type Package

Title Stemming Algorithms for the Portuguese Language

Version 0.0.4

Description Wraps a collection of stemming algorithms for the Portuguese Language.

URL https://github.com/dfalbel/optstem

License MIT + file LICENSE

LazyData TRUE

RoxygenNote 5.0.1

Encoding UTF-8

Imports dplyr, hunspell, magrittr, rslp, SnowballC, stringr, tidyr, tokenizers

Suggests testthat, covr, plyr, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Daniel Falbel [aut, cre]

Maintainer Daniel Falbel <dfalbel@gmail.com>

Repository CRAN

Date/Publication 2019-01-02 14:40:02 UTC

R topics documented:

  complete_stems ........................................ 2
  extract_words ......................................... 2
  overstemming_index .................................. 3
  performance ........................................... 3
  ptstem_words ......................................... 4
  stem_hunspell ....................................... 5
  stem_modified_hunspell ............................... 5
  stem_porter ......................................... 6
extract_words

description

extract_words extracts all words from a character string of texts.

usage

extract_words(texts)

arguments

texts character vector of texts

note

it uses the regex \b[:alpha:]\+\b to extract words.
overstemming_index  

**Overstemming Index (OI)**

**Description**

It calculates the proportion of unrelated words that were combined.

**Usage**

```r
overstemming_index(words, stems)
```

**Arguments**

- `words` is a data frame containing a column `word` and a column `group` so the function can identify groups of words.
- `stems` is a character vector with the stemming result for each word.

---

**performance**

**Performance**

**Description**

Performance

**Usage**

```r
performance(stemmers = c("rslp", "hunspell", "porter", "modified-hunspell"))
```

**Arguments**

- `stemmers` is a character vector with names of stemming algorithms. In the near future, functions will also be accepted.

**Value**

A data frame with the following measures calculated for each stemmer:

- UI: Understemming Index
- OI: Overstemming Index

**Examples**

```r
## Not run: perf <- performance()
```
ptstem_words  

Stem Words

Description

Stem a character vector of words using the selected algorithm.

Usage

ptstem_words(words, algorithm = "rslp", complete = T, ...)

ptstem(texts, algorithm = "rslp", n_char = 3, complete = T, ignore = NULL, ...)

Arguments

- words, texts: character vector of words.
- algorithm: string with the name of the algorithm to be used. One of "hunspell", "rslp", "porter" and modified-hunspell.
- complete: whether to complete words or not i.e. change all words with the same stem by the word that appears the most with that stem.
- ...: other arguments passed to the algorithms.
- n_char: minimum number of characters of words to be stemmed. Not used by ptstem_words.
- ignore: vector of words and regex's to ignore. Words are wrapped around stringr::fixed() for words like 'banana' don't get excluded when you ignore 'ana'. Also elements are considered a regex when they contain at least one punctuation symbol.

Details

You can choose whether to complete words or not using the complete argument. By default all algorithms are completing stems. For hunspell, it's better to always use complete = TRUE since even when using complete = FALSE it will complete words.

Complete finds the stem that appears the most in the full corpus. That's why it should not be used when you are stemming in parallel.

Examples

words <- c("balões", "aviões", "avião", "gostou", "gosto", "gostaram")
ptstem_words(words, "hunspell")
ptstem_words(words)
ptstem_words(words, algorithm = "porter", complete = FALSE)

texts <- c("coma frutas pois elas fazem bem para a saúde.", "não coma doces, eles fazem mal para os dentes.")
ptstem(texts, "hunspell")
ptstem(texts, n_char = 5)
stem_hunspell

Stemming using Hunspell

Description

This function uses Hunspell Stemmer to stem a vector of words. It uses the (Portuguese Brazilian) dictionary by default, and unlike hunspell::hunspell_stem it returns only one stem per word.

Usage

stem_hunspell(words, complete = TRUE)

Arguments

words character vector of words to be stemmed
complete whether words must be completed or not (T)

Details

As hunspell_stem can return a list of stems for each word, the function takes the stems that appears the most in the vector for each word.

Examples

words <- c("balões", "aviões", "avião", "gostou", "gosto", "gostaram")
ptstem:::stem_hunspell(words)

stem_modified_hunspell

Stemming with small modification of Hunspell

Description

This function uses Hunspell Stemmer to stem a vector of words. It uses the (Portuguese Brazilian) dictionary by default, and unlike hunspell::hunspell_stem it returns only one stem per word.

Usage

stem_modified_hunspell(words, complete = TRUE)
Arguments

words character vector of words to be stemmed
complete whether words must be completed or not (T)

Details

Then it uses the rslp stemmer in the hunspell stemmed result.

As hunspell_stem can return a list of stems for each word, the function takes the stems that appears the most in the vector for each word.

Examples

```r
words <- c("balões", "aviões", "avião", "gostou", "gosto", "gostaram")
ptstem:::stem_modified_hunspell(words)
```

---

Description

This function uses the Porter's algorithm to stem a vector of words. By default, the Porter's algorithm leaves words cutted. As this makes reading stemmed texts very difficult, this function provides an option to complete the stemmed words. By default it completes with the most used word in the text that has the same stem.

Usage

```r
stem_porter(words, complete = TRUE)
```

Arguments

words character vector of words to be stemmed
complete whether words must be completed or not (T)

```r
words <- c("balões", "aviões", "avião", "gostou", "gosto", "gostaram")
ptstem:::stem_porter(words)
```
**stem_rslp**

**Stemming using RSLP**

**Description**

This function uses the RSLP algorithm to stem a vector of words. By default, the RSLP algorithm leaves words cutted. As this makes reading stemmed texts very difficult, this function provides an option to complete the stemmed words. By default it completes with the most used word in the text that has the same stem.

**Usage**

```r
stem_rslp(wordsL complete = TRUE)
```

**Arguments**

- `words`: character vector of words to be stemmed
- `complete`: wheter words must be completed or not (T)

**References**


**Examples**

```r
words <- c("balões", "aviões", "avião", "gostou", "gosto", "gostaram")
ptstem:::stem_rslp(words)
```

---

**understemming_index**

**Understemming Index (UI)**

**Description**

It calculates the proportion of related words that had different stems.

**Usage**

```r
understemming_index(words, stems)
```

**Arguments**

- `words`: is a data.frame containing a column word a a column group so the function can identify groups of words.
- `stems`: is a character vector with the stemming result for each word
unify_stems

Description

Hunspell can suggest a list of stems for a word. This function tries to aggregate all stems into one. Consider the following:

Usage

unify_stems(words, stems)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>character vector of words</td>
</tr>
<tr>
<td>stems</td>
<td>character vector of stems</td>
</tr>
</tbody>
</table>

Details

a c(1,2) b c(2,3) c c(3)

You want that a, b and c to have the same stem.
Index

complete_stems, 2
extract_words, 2
overstemming_index, 3
performance, 3
ptstem (ptstem_words), 4
ptstem_words, 4
stem_hunspell, 5
stem_modified_hunspell, 5
stem_porter, 6
stem_rslp, 7
understemming_index, 7
unify_stems, 8