Package ‘Rstem’

September 19, 2011

Project Omegahat

Version 0.4-1

Title Interface to Snowball implementation of Porter’s word stemming algorithm.

Description An R interface to the C code that implements Porter’s word stemming algorithm for collapsing words to a common root to aid comparison of texts. There is code to for different languages (i.e. danish, dutch, english, finnish, french, german,norwegian, portuguese, russian, spanish, swedish). However,these may not be applicable if the words require UTF encoding. This is extensible by allowing different routines to be specified to create the C routines used in the stemming,permitting debugging, profiling, pool management, caching, etc.

Author Duncan Temple Lang <duncan@wald.ucdavis.edu>

Maintainer Duncan Temple Lang <duncan@wald.ucdavis.edu>

Acknowledgements The stemming code is taken directly from Dr Martin Porter’s distribution, available at http://snowball.tartarus.org and is generated via Snowball. The Snowball code is distributed under the BSD license.

Note This package is setup to support stemming for languages other than English and to dynamically fetch code for languages from the Snowball web site. The inst/scripts/download script fetches the code for languages supported by Snowball and includes it in the distribution. Support for Unicode is needed in R.

Repository CRAN

Date/Publication 2011-08-19 05:25:59

R topics documented:

getStemLanguages 2
wordStem 3

Index 5

`getStemLanguages` *Query the languages supported in this package*

Description

This dynamically determines the names of the languages for which stemming is supported by this package. This is controlled when the package is created (not installed) by downloading the stemming algorithms for the different languages.

This language support requires more support for Unicode and more complex text than simple strings.

Usage

```
getStemLanguages()
```

Details

This queries the C code for the list of languages that were compiled when the package was installed which in turn is determined by the code that was included in the distributed package itself.

Value

A character vector giving the names of the languages.

Author(s)

Duncan Temple Lang <duncan@wald.ucdavis.edu>

References

See `http://snowball.tartarus.org/`

See Also

`wordStem` `inst/scripts/download` in the source of the Rstem package.

Examples

```
getStemLanguages()
```

wordStem *Get the common root/stem of words*

Description

This function computes the stems of each of the given words in the vector. This reduces a word to its base component, making it easier to compare words like win, winning, winner. See http://snowball.tartarus.org/ for more information about the concept and algorithms for stemming.

Usage

```
wordStem(words, language = character(), warnTested = FALSE)
```

Arguments

`words` a character vector of words whose stems are to be computed.

`language` the name of a recognized language for the package. This should either be a single string which is an element in the vector returned by `getStemLanguages`, or alternatively a character vector of length 3 giving the names of the routines for creating and closing a Snowball `SN_env` environment and performing the stem (in that order). See the example below.

`warnTested` an option to control whether a warning is issued about languages which have not been explicitly tested as part of the unit testing of the code. For the most part, one can ignore these warnings and so they are turned off. In the future, we might consider controlling this with a global option, but for now we suppress the warnings by default.

Details

This uses Dr. Martin Porter's stemming algorithm and the interface generated by Snowball http://snowball.tartarus.org/.

Value

A character vector with as many elements as there are in the input vector with the corresponding elements being the stem of the word.

Author(s)

Duncan Temple Lang <duncan@wald.ucdavis.edu>

References

See http://snowball.tartarus.org/

Examples

```
    # Simple example
    # "win"    "win"    "winner"
  wordStem(c("win", "winning", 'winner'))


   # test the supplied vocabulary.
  testWords = readLines(system.file("words", "english", "voc.txt", package = "Rstem"))
  validate = readLines(system.file("words", "english", "output.txt", package = "Rstem"))

 ## Not run:
  # Read the test words directly from the snowball site over the Web
  testWords = readLines(url("http://snowball.tartarus.org/english/voc.txt"))

 ## End(Not run)


  testOut = wordStem(testWords)
  all(validate == testOut)

   # Specify the language from one of the built-in languages.
  testOut = wordStem(testWords, "english")
  all(validate == testOut)

   # To illustrate using the dynamic lookup of symbols that allows one
   # to easily add new languages or create and close environment
   # routines (for example, to manage pools if this were an efficiency
   # issue!)
  testOut = wordStem(testWords, c("testDynCreate", "testDynClose", "testDynStem"))
```

Index

∗Topic **IO**
- getStemLanguages, 2
- wordStem, 3

∗Topic **utilities**
- getStemLanguages, 2
- wordStem, 3

getStemLanguages, 2, *3*

wordStem, *2*, 3