Package: coreNLP (via r-universe)

October 27, 2026

Type Package

Title Wrappers Around Stanford CoreNLP Tools

Version 0.4-3

Author Taylor Arnold, Lauren Tilton

Maintainer Taylor Arnold <tarnold2@richmond.edu>

Description Provides a minimal interface for applying annotators from the 'Stanford CoreNLP' java library. Methods are provided for tasks such as tokenisation, part of speech tagging, lemmatisation, named entity recognition, coreference detection and sentiment analysis.

Imports rJava, XML

SystemRequirements Java (>= 7.0); Stanford CoreNLP <http://nlp.stanford.edu/ software/corenlp.shtml> (>= 3.5.2)

License GPL-2

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Config/pak/sysreqs default-jdk

Repository https://cran.r-universe.dev

Date/Publication 2022-08-17 12:40:09 UTC

RemoteUrl https://github.com/cran/coreNLP

RemoteRef HEAD

RemoteSha 3163b1707a002803f13d1a6cb3c560af4f8d0a06

Contents

annoEtranger.rda 2
annoHp.rda 2
annotateFile 3
annotateString 3

downloadCoreNLP 4
getCoreference 5
getDependency 5
getOpenIE 6
getParse 6
getSentiment 7
getToken 7
initCoreNLP 8
loadXMLAnnotation 9
parseAnnoXML 9
print.annotation 10
universalTagset 10

Index **11**

`annoEtranger.rda` *Annotation of first two lines of Albert Camus' L'Etranger*

Description

Parsed via the Stanford CoreNLP Java Library

Usage

```
annoEtranger
```

Format

a `annotation` object

Author(s)

Taylor Arnold, 2015-06-03

`annoHp.rda` *Annotation of first line of JK Rowling's The Philosopher's Stone*

Description

Parsed via the Stanford CoreNLP Java Library

Usage

```
annoHp
```

Format

a `annotation` object

Author(s)

Taylor Arnold, 2015-06-03

`annotateFile` *Annotate a text file*

Description

Runs the CoreNLP annotators for the text contained in a given file. The details for which annotators to run and how to run them are specified in the properties file loaded in via the `initCoreNLP` function (which must be run prior to any annotation).

Usage

```
annotateFile(file, format = c("obj", "xml", "text"), outputFile = NA,
  includeXSL = FALSE)
```

Arguments

`file`	a string giving the location of the file to be loaded.
`format`	the desired output format. Option `obj`, the default, returns an R object of class `annotation` and will likely be the desired choice for users loading the output into R. The `xml` and `text` exist primarily for saving the files on the disk.
`outputFile`	character string indicating where to put the output. If set to NA, the output will be returned by the function.
`includeXSL`	boolean. Whether the xml style sheet should be included in the output. Only used if format is `xml` and outputFile is not `NA`.

`annotateString` *Annotate a string of text*

Description

Runs the CoreNLP annotators over a given string of text. The details for which annotators to run and how to run them are specified in the properties file loaded in via the `initCoreNLP` function (which must be run prior to any annotation).

Usage

```
annotateString(text, format = c("obj", "xml", "text"), outputFile = NA,
  includeXSL = FALSE)
```

Arguments

text	a vector of strings for which an annotation is desired. Will be collapsed to length 1 using new line characters prior to the annotation.
format	the desired output format. Option obj, the default, returns an R object of class annotation and will likely be the desired choice for most users. The xml and text exist primarily for subsequently saving to disk.
outputFile	character string indicating where to put the output. If set to NA, the output will be returned by the function.
includeXSL	boolean. Whether the xml style sheet should be included in the output. Only used if format is xml and outputFile is not NA.

Examples

```
## Not run:
initCoreNLP()
sIn <- "Mother died today. Or, maybe, yesterday; I can't be sure."
annoObj <- annotateString(sIn)

## End(Not run)
```

---

downloadCoreNLP *Download java files needed for coreNLP*

Description

The coreNLP package does not supply the raw java files provided by the Stanford NLP Group as they are quite large. This function downloads the libraries for you, by default into the directory where the package was installed.

Usage

```
downloadCoreNLP(outputLoc, type = c("base", "chinese", "english", "french",
  "german", "spanish"))
```

Arguments

outputLoc	a string showing where the files are to be downloaded. If missing, will try to download files into the directory where the package was original installed.
type	type of files to download. The base backage, installed by default is required. Other jars include chinese, german, and spanish. These will be installed in addition to the base package.

Details

If you want to manually download files, simply unzip them and place in `system.file("extdata",package="coreNLP")`

Examples

```
## Not run:
downloadCoreNLP()
downloadCoreNLP(type="spanish")

## End(Not run)
```

---

getCoreference *Get Coreference*

Description

Returns a dataframe containing all coreferences detected in the text.

Usage

```
getCoreference(annotation)
```

Arguments

annotation	an annotation object

Examples

```
getCoreference(annoHp)
```

---

getDependency *Get Dependencies*

Description

Returns a data frame of the coreferences of an annotation

Usage

```
getDependency(annotation, type = c("CCprocessed", "basic", "collapsed"))
```

Arguments

annotation	an annotation object
type	the class of coreference desired

Examples

```
getDependency(annoEtranger)
getDependency(annoHp)
```

getOpenIE *Get OpenIE*

Description

Returns a dataframe containing all OpenIE triples.

Usage

```
getOpenIE(annotation)
```

Arguments

annotation	an annotation object

Examples

```
getOpenIE(annoHp)
```

getParse *Get parse tree as character vector*

Description

Returns a character vector of the parse trees. Mostly use for visualization; the output of getToken will generally be more conveniant for manipulating in R.

Usage

```
getParse(annotation)
```

Arguments

annotation	an annotation object

Examples

```
getParse(annoEtranger)
```

getSentiment *Get Sentiment scores*

Description

Returns a data frame of the sentiment scores from an annotation

Usage

```
getSentiment(annotation)
```

Arguments

annotation	an annotation object

Examples

```
getSentiment(annoEtranger)
getSentiment(annoHp)
```

getToken *Get tokens as data frame*

Description

Returns a data frame of the tokens from an annotation object.

Usage

```
getToken(annotation)
```

Arguments

annotation	an annotation object

Examples

```
getToken(annoEtranger)
```

`initCoreNLP` *Initialize the CoreNLP java object*

Description

This must be run prior to calling any other CoreNLP functions. It may be called multiple times in order to specify a different parameter set, but note that if you use a different configuration during the same R session it must have a unique name.

Usage

```
initCoreNLP(libLoc, type = c("english", "english_all", "english_fast",
  "arabic", "chinese", "french", "german", "spanish"), parameterFile = NULL,
  mem = "4g")
```

Arguments

`libLoc`	a string giving the location of the CoreNLP java files. This should point to a directory which contains, for example the file "stanford-corenlp-*.jar", where "*" is the version number. If missing, the function will try to find the library in the environment variable CORENLP_HOME, and otherwise will fail.
`type`	type of model to load. Ignored if parameterFile is set.
`parameterFile`	the path to a parameter file. See the CoreNLP documentation for an extensive list of options. If missing, the package will simply specify a list of standard annotators and otherwise only use default values.
`mem`	a string giving the amount of memory to be assigned to the rJava engine. For example, "6g" assigned 6 gigabytes of memory. At least 2 gigabytes are recommended at a minimum for running the CoreNLP package. On a 32bit machine, where this is not possible, setting "1800m" may also work. This option will only have an effect the first time `initCoreNLP` is called, and also will not have an effect if the java engine is already started by a seperate process.

Examples

```
## Not run:
initCoreNLP()
sIn <- "Mother died today. Or, maybe, yesterday; I can't be sure."
annoObj <- annotateString(sIn)

## End(Not run)
```

loadXMLAnnotation *Load CoreNLP XML file*

Description

Loads a properly formated XML file output by the CoreNLP library into an annotation object in R.

Usage

```
loadXMLAnnotation(file, encoding = "unknown")
```

Arguments

file	connection or character string giving the file name to load
encoding	encoding to be assumed for input strings. It is used to mark character strings as known to be in Latin-1 or UTF-8: it is not used to re-encode the input. Passed to readLines.

parseAnnoXML *Parse annotation xml*

Description

Returns an annotation object from a character vector containing the xml. Not exported; use loadXMLAnnotation instead.

Usage

```
parseAnnoXML(xml)
```

Arguments

xml	character vector containing the xml file from an annotation

print.annotation *Print a summary of an annotation object*

Description

Print a summary of an annotation object

Usage

```
## S3 method for class 'annotation'
print(x, ...)
```

Arguments

x	an annotation object
...	other arguments. Currently unused.

Examples

```
print(annoEtranger)
```

universalTagset *Convert Penn TreeBank POS to Universal Tagset*

Description

Maps a character string of English Penn TreeBank part of speech tags into the universal tagset codes. This provides a reduced set of tags (12), and a better cross-linguist model of speech.

Usage

```
universalTagset(pennPOS)
```

Arguments

pennPOS	a character vector of penn tags to match

Examples

```
tok <- getToken(annoEtranger)
cbind(tok$POS,universalTagset(tok$POS))
```

Index

∗ **datasets**
annoEtranger.rda, 2
annoHp.rda, 2

annoEtranger *(*annoEtranger.rda*)*, 2
annoEtranger.rda, 2
annoHp *(*annoHp.rda*)*, 2
annoHp.rda, 2
annotateFile, 3
annotateString, 3

downloadCoreNLP, 4

getCoreference, 5
getDependency, 5
getOpenIE, 6
getParse, 6
getSentiment, 7
getToken, *6*, 7

initCoreNLP, 8

loadXMLAnnotation, 9

parseAnnoXML, 9
print.annotation, 10

universalTagset, 10