Package 'wordnet'

February 18, 2026

Title WordNet Interface

Version 0.1-18

Description An interface to WordNet using the Jawbone Java API to WordNet. WordNet (<https://wordnet.princeton.edu/>) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Please note that WordNet(R) is a registered tradename. Princeton University makes WordNet available to research and commercial users free of charge provided the terms of their license (<https://wordnet.princeton.edu/license-and-commercial-use>) are followed, and proper reference is made to the project using an appropriate citation (<https://wordnet.princeton.edu/citing-wordnet>). The WordNet database files need to be made available separately, either via package 'wordnetDicts' from <https://datacube.wu.ac.at>, installing system packages where available, or direct download from <https://wordnetcode.princeton.edu/3.0/WNdb-3.0.tar.gz>.

Imports rJava (>= 0.6-3)

Suggests wordnetDicts

Additional_repositories https://datacube.wu.ac.at

SystemRequirements Java (>= 5.0); WordNet database files (direct download: <https://wordnetcode.princeton.edu/3.0/WNdb-3.0.tar.gz>; Debian and Fedora package: wordnet)

License MIT + file LICENSE

URL https://wordnet.princeton.edu/, https://sites.google.com/site/mfwallace/jawbone

NeedsCompilation no

Author Ingo Feinerer [aut] (ORCID: <https://orcid.org/0000-0001-7656-8338>), Kurt Hornik [aut, cre] (ORCID: <https://orcid.org/0000-0003-4198-9911>), Mike Wallace [ctb, cph] (Jawbone Java WordNet API library)

Maintainer Kurt Hornik <Kurt.Hornik@R-project.org>

Repository CRAN

Date/Publication 2026-02-18 10:03:17 UTC

Contents

`getDict` *Get Default Dictionary*

Description

The package **wordnet** tries to locate a valid WordNet installation on start up by investigating the `WNHOME` environment variable and by trying default installation locations. On success it acquires a pointer to the actual WordNet dictionary and stores internally a reference to the dictionary instance. `getDict` returns this default reference.

Usage

```
getDict()
```

Details

You can manually point the package to the WordNet installation via `setDict`.

Value

A dictionary instance.

Author(s)

Ingo Feinerer

References

Fellbaum C (1998). *WordNet: An Electronic Lexical Database*. Bradford Books. https://mitpress.mit.edu/9780262561167/.

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
if(initDict())
  getDict()
```

`getDictInstance` *Get a Dictionary Instance*

Description

Returns an instance to a WordNet dictionary.

Usage

```
getDictInstance()
```

Value

A dictionary object.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
if(initDict())
  getDictInstance()
```

getFilterTypes *Get Available Filter Types*

Description

Get available filter types.

Usage

```
getFilterTypes()
```

Value

A character vector with available filter types.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
getFilterTypes()
```

getIndexTerms *Get Index Terms*

Description

Get index terms from a WordNet dictionary as specified by a filter.

Usage

```
getIndexTerms(pos, maxLimit, filter)
```

Arguments

pos	Part of speech type. Must be either "ADJECTIVE", "ADVERB", "NOUN", or "VERB".
maxLimit	Maximum number of results.
filter	A term filter (see getTermFilter).

Value

A list of index terms.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
if(initDict()) {
  filter <- getTermFilter("StartsWithFilter", "car", TRUE)
  getIndexTerms("NOUN", 5, filter)
}
```

---

getLemma *Get Index Term Lemma*

Description

Retrieve the lemma (i.e., word) of an index term.

Usage

```
getLemma(indexterm)
```

Arguments

indexterm	The index term whose lemma is returned.

Value

A character vector holding the index term lemma.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

See Also

getIndexTerms

Examples

```
if(initDict()) {
  filter <- getTermFilter("StartsWithFilter", "car", TRUE)
  terms <- getIndexTerms("NOUN", 5, filter)
  sapply(terms, getLemma)
}
```

---

getRelatedSynsets *Get Related Synsets for a Synset*

Description

Get related synsets for a given synset based on a pointer symbol.

Usage

```
getRelatedSynsets(synset, pointerSymbol)
```

Arguments

synset	Basic synset.
pointerSymbol	A symbol indicating the type of the related synsets. An overview is available at https://wordnet.princeton.edu/documentation/wnsearch3wn.

Value

A list of synsets.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

See Also

getSynsets

Examples

```
if(initDict()) {
  filter <- getTermFilter("ExactMatchFilter", "hot", TRUE)
  terms <- getIndexTerms("ADJECTIVE", 5, filter)
  synsets <- getSynsets(terms[[1]])
  related <- getRelatedSynsets(synsets[[1]], "!")
  sapply(related, getWord)
}
```

`getSynonyms` *Get Synonyms for an Index Term*

Description

Get synonyms for a given index term.

Usage

```
getSynonyms(indexterm)
```

Arguments

`indexterm`	The input index term.

Value

A character vector holding the synonyms for the given index term.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

See Also

`getIndexTerms`

Examples

```
if(initDict()) {
  filter <- getTermFilter("ExactMatchFilter", "company", TRUE)
  terms <- getIndexTerms("NOUN", 5, filter)
  getSynonyms(terms[[1]])
}
```

`getSynsets` *Get Synsets for an Index Term*

Description

Get synsets for a given index term.

Usage

```
getSynsets(indexterm)
```

Arguments

`indexterm`	The input index term.

Value

A list of synsets.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

See Also

`getIndexTerms`

Examples

```
if(initDict()) {
  filter <- getTermFilter("ExactMatchFilter", "hot", TRUE)
  terms <- getIndexTerms("ADJECTIVE", 5, filter)
  getSynsets(terms[[1]])
}
```

getTermFilter *Get a Term Filter*

Description

Get a term filter.

Usage

```
getTermFilter(type, word, ignoreCase)
```

Arguments

type	Filter type. Available filters are "ContainsFilter", "EndsWithFilter", "ExactMatchFilter", "RegexFilter", "SoundFilter", "StartsWithFilter", and "WildcardFilter". Can also be a unique abbreviation of an available filter name.
word	Term to be matched.
ignoreCase	Indicates whether lower and upper case are distinguished.

Value

A term filter.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
if(initDict())
  getTermFilter("StartsWithFilter", "car", TRUE)
```

getWord *Get Synset Word*

Description

Get the words in a synset.

Usage

```
getWord(synset)
```

Arguments

synset	The synset whose words are returned.

Value

A character vector holding the words.

Author(s)

Ingo Feinerer

References

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

See Also

getSynsets

Examples

```
if(initDict()) {
  filter <- getTermFilter("ExactMatchFilter", "hot", TRUE)
  terms <- getIndexTerms("ADJECTIVE", 5, filter)
  synsets <- getSynsets(terms[[1]])
  related <- getRelatedSynsets(synsets[[1]], "!")
  sapply(related, getWord)
}
```

`initDict` *Initialize Dictionary*

Description

Initializes the WordNet dictionary using the Jawbone Java API to WordNet.

Usage

```
initDict(pathData = "")
```

Arguments

`pathData`	Path to the WordNet data files.

Details

In case the user supplied path is invalid the function tries to find the installation itself by investigating the `WNHOME` environment variable and by trying default installation locations.

Value

A logical value indicating whether a valid WordNet installation has been found.

Author(s)

Ingo Feinerer

References

Fellbaum C (1998). *WordNet: An Electronic Lexical Database*. Bradford Books. https://mitpress.mit.edu/9780262561167/.

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
## Not run: initDict("/usr/local/WordNet-3.0/dict")
```

setDict *Set Default Dictionary*

Description

The package **wordnet** tries to locate a valid WordNet installation on start up by investigating the WNHOME environment variable and by trying default installation locations. On success it acquires a pointer to the actual WordNet dictionary and stores internally a reference to the dictionary instance. However, if this procedure does not work automatically in your environment, you can provide the path to the WordNet installation and set the internal default reference via this function.

Usage

```
setDict(pathData)
```

Arguments

pathData	Path to the WordNet data files.

Value

A dictionary instance.

Author(s)

Ingo Feinerer

References

Fellbaum C (1998). *WordNet: An Electronic Lexical Database*. Bradford Books. https://mitpress.mit.edu/9780262561167/.

Wallace M (2007). *Jawbone Java WordNet API*. https://sites.google.com/site/mfwallace/jawbone.

Examples

```
## Not run: setDict("/usr/local/WordNet-3.0/dict")
```

synonyms *Get Synonyms for a Word*

Description

Get synonyms for a given word.

Usage

```
synonyms(word, pos)
```

Arguments

word	The input word.
pos	Part of speech type. Must be either "ADJECTIVE", "ADVERB", "NOUN", or "VERB".

Value

A character vector holding the synonyms for the given word.

Author(s)

Ingo Feinerer

See Also

getSynonyms

Examples

```
if(initDict())
  synonyms("company", "NOUN")
```

Index