Package 'tidystopwords'

February 12, 2019

Type Package

Title Customizable Lists of Stopwords in 53 Languages

Version 0.9.0

Date 2019-01-23

Author Silvie Cinkova, Maciej Eder

Maintainer Maciej Eder <maciejeder@gmail.com>

Depends R (>= 2.14)

Imports dplyr, stringr

Description Functions to generate stopword lists in 53 languages, in a way consistent across all the languages supported. The generated lists are based on the morphological tagset from the Universal Dependencies.

License GPL (>= 3)

Encoding UTF-8

LazyData FALSE

LazyDataCompression TRUE

SysDataCompression TRUE

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-12 17:20:02 UTC

R topics documented:

`generate_stoplist` *Listing of stop words with control over language and part of speech.*

Description

Generate a vector of stop words in one or several languages. Parameters allow you to toggle different parts of speech as well as to add new words.

Usage

```
generate_stoplist(lang_name = NULL,
                  lang_id = NULL,
                  output_form = "vector",
                  stop_lemmas = NULL,
                  stop_forms = NULL,
                  stop_foreign_words = TRUE,
                  stop_abbreviations = TRUE,
                  stop_pronominals = TRUE,
                  stop_determiners_quantifiers = TRUE,
                  stop_conjuctions = TRUE,
                  stop_adpositions = TRUE,
                  stop_subordinating_conjunctions = TRUE,
                  stop_auxiliary_verbs = TRUE,
                  stop_interjections = TRUE,
                  stop_particles = TRUE,
                  stop_numerals = TRUE,
                  stop_symbols_crosslingual = TRUE,
                  stop_punctuation_crosslingual = TRUE,
                  custom_filter = NULL )
```

Arguments

`lang_name`	single string or a character vector. `NULL` by default. However, when both `lang_name` and `lang_id` are `NULL`, the function returns a vector of all word forms in all languages.
`lang_id`	a single string or a character vector. Call `list_supported_language_ids()` to see the supported ISO-639 language codes, if you prefer these codes to language names.
`output_form`	default `"vector"`, alternatively `"data.frame"`.
`stop_lemmas`	default `NULL`. Supply a string or a character vector. You get all word forms of each listed lemma found in the data set for your selected language(s). POS disambiguation is not possible with this argument: mind homonyms.
`stop_forms`	default `NULL`. Supply a string or a character vector. You get all word forms present in the data set for your selected language(s). POS disambiguation is not possible with this argument: mind homonyms.

The following parameters extract word forms from the data set with the relevant linguistic markup for your selected languages(s). It is often a combination of the coarse-grained universal POS tag and a set of so-called Universal Features. The same set of filters are used across all languages.

`stop_foreign_words`
: default `TRUE`.

`stop_abbreviations`
: default `TRUE`.

`stop_pronominals`
: default `TRUE`. All pronouns (e.g. "me", "this", "what") along with pronominal adjectives and adverbs (e.g. "whose", "where").

`stop_determiners_quantifiers`
: default `TRUE`. Articles, possessive pronouns, quantifiers (e.g. "all", "few", "both").

`stop_conjuctions`
: default `TRUE`. Coordinating conjunctions; e.g. "and", "but".

`stop_adpositions`
: default `TRUE`. Prepositions and postpositions; e.g. "of", "instead", "ago".

`stop_subordinating_conjunctions`
: default `TRUE`. E.g. "because".

`stop_auxiliary_verbs`
: default `TRUE`. E.g. "have", "be". Many languages mark also modal verbs ("must", "can") as auxiliary verbs, but some (e.g. Czech) do not. Check the UD documentation for your language of interest. Also, mind that when adding e.g. "have" to your stop-word vector, you are also going to get rid of all its lexical uses; e.g. in "I have two cars."

`stop_interjections`
: default `TRUE`. Exclamatory words; e.g. "wow", "ouch", but also "yes". Here the individual languages differ very much.

`stop_particles`
: default `TRUE`. Also a very heterogeneous class. Check what is in there for your language of interest. NB: the English particles in phrasal verbs are regarded as adpositions, not as particles.

`stop_numerals`
: default `TRUE`. Words denoting numbers: cardinal and ordinal numerals, as well as words that many languages normally regard as adjectives or adverbs; e.g. "double", "fourfold".

`stop_symbols_crosslingual`
: default `TRUE`. All symbols (e.g. emoji or currencies) harvested across all languages.

`stop_punctuation_crosslingual`
: default `TRUE`. All punctuation marks harvested across all languages.

`custom_filter`
: default `NULL`. If you are not happy with the predefined stopword specifications, you can write your own filter for the underlying data set. Insert an evaluating expression in quotes. It will be parsed in `dplyr::filter(multilingual_stoplist, your expression)`. If you want to use the full power of `dplyr`, e.g. grouping, use the `multilingual_stoplist` data set on its own. This argument works independently of the pre-defined filters. Mind to turn them off if you do not want them. This filter also helps you override the pre-defined language choice, should you wish to.

Value

A character vector, UTF-8 encoded.

Warning

- The function stops when both `lang_name` and `lang_id` equal `NULL`. Setting a language in the custom filter does not do.
- The function stops when `lang_name` or `lang_id` contains an unsupported item. Mind that the selection is case-sensitive. The error message prints the culprit(s). If you have set both `lang_name` or `lang_id` and the check finds an error in the former, it will not go on checking the latter.
- The languages supported by the first version of this package are listed as argument default, but the actual language inventory relies on the underlying data frame. The initial checks call `list_supported_language_names()` and `list_supported_language_ids()`.
- The pre-defined linguistic filters are not mutually exclusive. Their overlap varies among languages.
- You will see a warning message when you set both `lang_name` and `lang_id`, no matter whether they represent the same language or different languages.
- When you run the function with default argument values, you will get stopwords from all supported languages mixed up and see a warning about that.
- The stoplists are fully data-driven. We have set a threshold of 10 occurrences of a combination of language, word form, lemma, POS and the Universal Features to remove obvious noise, but some noise is bound to have come through anyway. It is mainly foreign words that were given a regular POS tag (cf. the example code snippets below: the English "and" has sneaked in among the German coordinating conjunctions). Many languages are represented by balanced and large corpora of standard written texts, but some are not; e.g. based mainly on a Bible translation. Hence also their stopwords can be biased.

Author(s)

Silvie Cinková, Maciej Eder

References

The data set is based on the official release of Version 2.1 of Universal Dependencies.

http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

See Also

`list_supported_language_names`, `list_supported_language_ids`, `list_supported_pos`, `multilingual_stoplist`

Examples

```
# standard usage (might return some non-ASCII characters):
generate_stoplist(lang_name = "English")

# to get only conjunctions (11 items):
generate_stoplist(lang_name = "English",
                      stop_foreign_words = FALSE,
                      stop_abbreviations = FALSE,
                      stop_pronominals = FALSE,
                      stop_determiners_quantifiers = FALSE,
                      stop_conjuctions = TRUE,
                      stop_adpositions = FALSE,
                      stop_subordinating_conjunctions = FALSE,
                      stop_auxiliary_verbs = FALSE,
                      stop_interjections = FALSE,
                      stop_particles = FALSE,
                      stop_numerals = FALSE,
                      stop_symbols_crosslingual = FALSE,
                      stop_punctuation_crosslingual = FALSE)
```

`list_supported_language_ids` *Listing of language ids to include in stopword lists you generate by 'generate_stoplist()'.*

Description

The function gives you a character vector of supported language ids, e.g. "en", "cs", "pl".

Usage

```
list_supported_language_ids()
```

Details

The `stopwoRds` package relies on `multilingual_stoplist`, a large multilingual table with individual word forms as rows, derived from the Universal Dependencies treebanks. Each word form comes along with its lemma and part of speech, as well as with the language name and its ISO-639 code. This function gives you unique values from the `language_id` column of `multilingual_stoplist`. The current ids are a mix of different versions of ISO-639 language codes.

Value

A character vector.

Author(s)

Silvie Cinková, Maciej Eder

References

http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

See Also

list_supported_pos, list_supported_language_ids, generate_stoplist, multilingual_stoplist

---

`list_supported_language_names` *Listing of languages to include in stopword lists you generate by* `generate_stoplist()`.

Description

The function gives you a character vector of supported language names, e.g. "English".

Usage

```
list_supported_language_names()
```

Details

The `stopwoRds` package relies on `multilingual_stoplist`, a large multilingual table with individual word forms as rows, derived from the Universal Dependencies treebanks. Each word form comes along with its lemma and part of speech, as well as with the language name and its ISO-639-nnnn code. This function gives you unique values from the `language_name` column of `multilingual_stoplist`.

Value

A character vector.

Author(s)

Silvie Cinková, Maciej Eder

References

http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

See Also

list_supported_pos, list_supported_language_ids, generate_stoplist, multilingual_stoplist

---

`list_supported_pos` *Listing of parts of speech to include in stopword lists you generate by* `generate_stoplist()`.

Description

The function gives you a character vector of supported parts of speech (e.g. prepositions). They are represented by abbreviations.

Usage

```
list_supported_pos()
```

Details

The `stopwoRds` package relies on `multilingual_stoplist`, a large multilingual table with individual word forms as rows, derived from the Universal Dependencies treebanks. Each word form comes along with its lemma and part of speech, as well as with the language name and its ISO-639-nnnn code. This function gives you unique values from the `POS` column of `multilingual_stoplist`. The parts of speech (POS) are common for all supported languages ("Universal Part-of-Speech tags").

Value

A character vector.

Author(s)

Silvie Cinkova, Maciej Eder

References

http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

See Also

list_supported_language_names, list_supported_language_ids, generate_stoplist, multilingual_stoplist

`multilingual_stoplist` *Multilingual stoplist*

Description

This dataset contains a dataframe with individual word forms in rows. You can control the part of speech and various frequency counts of your desired stop-word list.

Format

A data frame with 213,526 observations on the following 10 variables, encoded in UTF-8.

`language_id` a character vector

`language_name` a character vector

`POS` a character vector

`UFeat` a character vector

`lemma` a character vector

`word_form` a character vector

`freq_formUFeatPOS` an integer vector

`freq_formPOS` an integer vector

`freq_form` an integer vector

`prop_wformpos_wform` an integer vector

Details

This data frame has been derived from an official release of the Universal Dependencies (UD) treebanks. Treebanks are text corpora with linguistic annotation. The UD syntactic annotation follows principles of dependency syntax. The annotation encompasses for each text token:

- relevant morphological categories;
- lemma (the vocabulary form; e.g. active present infinitive in verbs)
- a reference to its syntactically governing word in the clause; e.g. "house" governs "old" in "old house".
- the type of the syntactic dependency between the word and its governing word; e.g. "attribute".

.

The morphological categories in UD are divided into two groups: the coarse-grained, cross-linguistically universal part-of-speech tags, and the Universal Features.

`language_id` - a mix of different versions of ISO-639 language codes as they were used in the source corpora;

`language_name` - the English name of the language, starting with capital letter;

`POS` - Universal part-of-speech tag;

UFeat - Universal Features tags separated by |; this is more fine-grained information derived from the national morphological tagsets that is too specific to apply to all languages. However, even these tags are cross-linguistically coordinated to ensure maximum cross-linguistic uniformity in markup.

lemma - the basic (vocabulary) word form;

word_form;

freq_formUFeatPOS - frequency of the given combination of word form, Universal Features and Universal Part of Speech. All frequencies are valid within the given language. Cross-lingual homonyms are kept separate (e.g. the preposition "in" existent in several Germanic languages is counted for each language separately, despite identical morphological markup));

freq_formPOS - frequency of the given combination of word form and Universal Part of Speech for the given language;

freq_form - frequency of the given word form for the given language. (Mind that this number does not come from the addition of the former two!);

prop_wforpos_wform - proportion of the given Universal Part of Speech in the given word form in the given language. This can be helpful when you decide whether to include a word in your list that is homonymous - considering the proportions gives you a hint on how much content you lose vs. how much noise you preserve. Remember to check the UD documentation of the individual corpora to assess whether their corpora are comparable to your texts, though!

Source

The data set is based on the official release of Version 2.1 of Universal Dependencies stored in the LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Czech Republic, http://hdl.handle.net/11234/1-2515.

References

To perform more sophisticated searches in this data frame, check the UD documentation. We have included all parts of speech except X (unknown POS). You can e.g. add to your stoplist all nouns above or below a given frequency threshold. http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

Examples

```
data(multilingual_stoplist)
print(multilingual_stoplist)

# this gives you the following results:

#   Observations: 213,526Variables: 10
#   $ language_id         <chr> "af", "af", "af", "af", "af", "af", "af"...
#   $ language_name       <chr> "Afrikaans", "Afrikaans", "Afrikaans", ...
#   $ POS                 <chr> "ADJ", "ADJ", "ADJ", "ADJ", "ADJ", "ADJ"...
#   $ UFeat               <chr> "AdjType=Attr|Case=Nom|Degree=Cmp", "Ad"..
```

```
#   $ lemma               <chr> "goed", "addisioneel", "afloop", "agbaa...
#   $ word_form           <chr> "beter", "addisionele", "afgelope", "ag...
#   $ freq_formUFeatPOS   <int> 12, 19, 23, 12, 117, 11, 22, 13, 17, 11...
#   $ freq_formPOS        <int> 13, 19, 23, 12, 117, 11, 22, 13, 21, 11...
#   $ freq_form           <int> 13, 19, 23, 12, 117, 11, 22, 13, 21, 11...
#   $ prop_wformpos_wform <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
```

`tidystopwords` *Customisable Lists of Stopwords in 53 Languages*

Description

The idea behind this package is to give the user control over the stop-word selection.

Details

The idea behind this package is to give the user control over the stop-word selection. The core `generate_stoplist` function relies on `multilingual_stopwords`, a large data frame derived from the current release of the Universal Dependencies Treebanks. We have included all languages whose corpora totalled above 10,000 tokens – large enough to cover all common closed-class words, such as prepositions, conjunctions, and auxiliary verbs. The data comes encoded in UTF-8.

Author(s)

Silvie Cinková, Maciej Eder

References

The data set is based on the official release of Version 2.1 of Universal Dependencies.

http://universaldependencies.org

Nivre, Joakim; Agić, Željko; Ahrenberg, Lars; et al., 2017, Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

Index