Package 'lexicon'

December 2, 2017

Title Lexicons for Text Analysis

Version 0.6.0

Maintainer Tyler Rinker <tyler.rinker@gmail.com>

Description A collection of lexical hash tables, dictionaries, and word lists.

Depends R (>= 3.2.2)

Imports data.table, syuzhet (>= 1.0.1)

Date 2017-12-02

License MIT + file LICENSE

LazyData TRUE

Roxygen list(wrap = FALSE)

RoxygenNote 6.0.1

BugReports https://github.com/trinker/lexicon/issues?state=open

URL https://github.com/trinker/lexicon

Collate 'available_data.R' 'common_names.R'
'discourse_markers_alemany.R' 'dodds_sentiment.R'
'freq_first_names.R' 'freq_last_names.R' 'function_words.R'
'grady_augmented.R' 'hash_emoticons.R' 'hash_grady_pos.R'
'hash_lemmas.R' 'hash_power.R' 'hash_sentiment_huliu.R'
'hash_sentiment_inquirer.R' 'utils.R'
'hash_sentiment_jockers.R' 'hash_sentiment_nrc.R'
'hash_sentiment_senticnet.R' 'hash_sentiment_sentiword.R'
'hash_sentiment_vadar.R' 'hash_strength.R' 'hash_syllable.R'
'hash_valence_shifters.R' 'key_abbreviation.R'
'key_contractions.R' 'key_grade.R' 'key_grades.R'
'key_ratings.R' 'lexicon-package.R' 'nrc_emotions.R'
'pos_action_verb.R' 'pos_adverb.R' 'pos_df_irregular_nouns.R'
'pos_df_pronouns.R' 'pos_interjections.R' 'pos_preposition.R'
'pos_unchanging_nouns.R' 'profanity_alvarez.R'
'profanity_arr_bad.R' 'profanity_banned.R' 'profanity_google.R'
'profanity_von_ahn.R' 'sw_buckley_salton.R' 'sw_dolch.R'
'sw_fry_100.R' 'sw_fry_1000.R' 'sw_fry_200.R' 'sw_fry_25.R'
'sw_jockers.R' 'sw_lucene.R' 'sw_mallet.R' 'sw_onix.R'
'sw_python.R'

Author Tyler Rinker [aut, cre]

RemoteType local

RemoteUrl C:\{ }Users\{ }Tyler\{ }GitHub\{ }lexicon

RemoteSha NA

RemoteBranch master

RemoteUsername trinker

RemoteRepo lexicon

R topics documented:

available_data 3
common_names 3
discourse_markers_alemany 4
dodds_sentiment 5
freq_first_names 5
freq_last_names 6
function_words 6
grady_augmented 7
hash_emoticons 7
hash_grady_pos 8
hash_lemmas 9
hash_power 9
hash_sentiment_huliu 10
hash_sentiment_inquirer 11
hash_sentiment_jockers 11
hash_sentiment_nrc 12
hash_sentiment_senticnet 12
hash_sentiment_sentiword 13
hash_sentiment_vadar 14
hash_strength 15
hash_syllable 15
hash_valence_shifters 16
key_abbreviation 17
key_contractions 17
key_grade 18
key_rating 18
key_sentiment_jockers 19
lexicon 19
nrc_emotions 19
pos_action_verb 20
pos_adverb 21
pos_df_irregular_nouns 21
pos_df_pronouns 22
pos_interjections 22
pos_preposition 23
pos_unchanging_nouns 23
profanity_alvarez 24
profanity_arr_bad 24
profanity_banned 25
profanity_google 25
profanity_von_ahn 26
sw_buckley_salton 26

sw_dolch 27
sw_fry_100 27
sw_fry_1000 28
sw_fry_200 28
sw_fry_25 29
sw_jockers 29
sw_lucene 30
sw_mallet 30
sw_onix 31
sw_python 31

Index **33**

`available_data` *Get Available* **lexicon** *Data*

Description

See available **lexicon** data a data.frame.

Usage

```
available_data()
```

Value

Returns a data.frame

Examples

```
available_data()
```

`common_names` *First Names (U.S.)*

Description

A dataset containing 1990 U.S. census data on first names.

Usage

```
data(common_names)
```

Format

A character vector with 5493 elements

References

http://www.census.gov

`discourse_markers_alemany` *Alemany's Discourse Markers*

Description

A dataset containing discourse markers

Usage

```
data(discourse_markers_alemany)
```

Format

A data frame with 97 rows and 5 variables

Details

A dictionary of *discourse markers* from Alemany (2005). "In this lexicon, discourse markers are characterized by their structural (continuation or elaboration) and semantic (revision, cause, equality, context) meanings, and they are also associated to a morphosyntactic class (part of speech, PoS), one of adverbial (A), phrasal (P) or conjunctive (C)... Sometimes a discourse marker is **underspecified** with respect to a meaning. We encode this with a hash. This tends to happen with structural meanings, because these meanings can well be established by discursive mechanisms other than discourse markers, and the presence of the discourse marker just reinforces the relation, whichever it may be." (p. 191).

- marker. The discourse marker
- type. The semantic type (typically overlaps with `semantic` except in the special types
- structural. How the marker is used structurally
- semantic. How the marker is used semantically
- pos. Part of speech: adverbial (A), phrasal (P) or conjunctive (C)

References

Alemany, L. A. (2005). Representing discourse for automatic text summarization via shallow NLP techniques (Unpublished doctoral dissertation). Universitat de Barcelona, Barcelona.

http://www.cs.famaf.unc.edu.ar/~laura/shallowdisc4summ/tesi_electronica.pdf
http://russell.famaf.unc.edu.ar/~laura/shallowdisc4summ/discmar/#description

`dodds_sentiment` *Language Assessment by Mechanical Turk Sentiment Words*

Description

A dataset containing words, average happiness score (polarity), standard deviations, and rankings.

Usage

```
data(dodds_sentiment)
```

Format

A data frame with 10222 rows and 8 variables

Details

- word. The word.
- happiness_rank. Happiness ranking of words based on average happiness scores.
- happiness_average. Average happiness score.
- happiness_standard_deviation. Standard deviations of the happiness scores.
- twitter_rank. Twitter ranking of the word.
- google_rank. Google ranking of the word.
- nyt_rank. New York Times ranking of the word.
- lyrics_rank. lyrics ranking of the word.

References

Dodds, P.S., Harris, K.D., Kloumann, I.M., Bliss, C.A., & Danforth, C.M. (2011) Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. PLoS ONE 6(12): e26752. doi:10.1371/journal.pone.0026752

http://www.plosone.org/article/fetchSingleRepresentation.action?uri=info:doi/10.1371/journal.pone.0026752.s001

`freq_first_names` *Frequent U.S. First Names*

Description

A dataset containing frequent first names based on the 1990 U.S. census.

Usage

```
data(freq_first_names)
```

Format

A data frame with 5494 rows and 4 variables

Details

- Name. A first name
- n. The approximate frequency within the sex
- prop. The proportion within the sex
- sex. The sex corresponding to the name

References

http://names.mongabay.com

`freq_last_names` *Frequent U.S. Last Names*

Description

A dataset containing frequent last names based on the 1990 U.S. census.

Usage

```
data(freq_last_names)
```

Format

A data frame with 14,840 rows and 3 variables

Details

- Surname. A last name
- n. The approximate frequency
- prop. The proportion

References

http://names.mongabay.com

`function_words` *Function Words*

Description

A vector of function words from John and Muriel Higgins's list used for the text game ECLIPSE. The lest is augmented with additional contractions from `key_contractions`.

Usage

```
data(function_words)
```

Format

A character vector with 350 elements

References

`http://myweb.tiscali.co.uk/wordscape/museum/funcword.html`

`grady_augmented` *Augmented List of Grady Ward's English Words and Mark Kantrowitz's Names List*

Description

A dataset containing a vector of Grady Ward's English words augmented with `hash_syllable`, Mark Kantrowitz's names list, other proper nouns, and contractions.

Usage

```
data(grady_augmented)
```

Format

A character vector with 122806 elements

Details

A dataset containing a vector of Grady Ward's English words augmented with proper nouns (U.S. States, Countries, Mark Kantrowitz's Names List, and months) and contractions. That dataset is augmented for spell checking purposes.

References

Moby Thesaurus List by Grady Ward (http://www.gutenberg.org)

`hash_emoticons` *Emoticons*

Description

A **data.table** key containing common emoticons (adapted from Popular Emoticon List).

Usage

```
data(hash_emoticons)
```

Format

A data frame with 75 rows and 2 variables

Details

- x. The graphic representation of the emoticon
- y. The meaning of the emoticon

References

http://www.lingo2word.com/lists/emoticon_listH.html

Examples

```
## Not run:
library(data.table)
hash_emoticons[c(':-(', '0;)')]

## End(Not run)
```

`hash_grady_pos` *Grady Ward's Moby Parts of Speech*

Description

A dataset containing a hash lookup of Grady Ward's parts of speech from the Moby project. The words with non-ASCII characters removed.

Usage

```
data(hash_grady_pos)
```

Format

A data frame with 250,892 rows and 5 variables

Details

- word. The word.
- pos. The part of speech; one of :`Adjective`, `Adverb`, `Conjunction`, `Definite Article`, `Interjection`, `Noun`, `Noun Phrase`, `Plural`, `Preposition`, `Pronoun`, `Verb (intransitive)`, `Verb (transitive)`, or `Verb (usu participle)`. Note that the first part of speech for a word is its primary use; all other uses are seondary.
- n_pos. The number of parts of speech associated with a word. Useful for filtering.
- space. logical. If `TRUE` the word contains a space. Useful for filtering.
- primary. logical. If `TRUE` the word is the primary part of speech used.

Source

http://icon.shef.ac.uk/Moby/mpos.html

References

Moby Thesaurus List by Grady Ward: http://icon.shef.ac.uk/Moby/mpos.html

Examples

```
## Not run:
library(data.table)

hash_grady_pos['dog']
hash_grady_pos[primary == TRUE, ]
hash_grady_pos[primary == TRUE & space == FALSE, ]

## End(Not run)
```

hash_lemmas *Lemmatization List*

Description

A dataset based on Mechura's (2016) English lemmatization list. This data set can be useful for join style lemma replacement of inflected token forms to their root lemmas. While this is not a true morphological analysis this style of lemma replacement is fast and typically still robust.

Usage

```
data(hash_lemmas)
```

Format

A data frame with 41,532 rows and 2 variables

Details

- token. An inflected token with affixes
- lemma. A base form

References

Mechura, M. B. (2016). *Lemmatization list: English (en)* [Data file]. Retrieved from http://www.lexiconista.com

hash_power *Power Lookup Key*

Description

A **data.table** containing a power lookup key.

Usage

```
data(hash_power)
```

Format

A data frame with 872 rows and 2 variables

Details

- x. A power word
- y. A positive or negative value indicating the direction of power in relation to the subject

References

http://www.wjh.harvard.edu/~inquirer/inqdict.txt

Examples

```
## Not run:
library(data.table)
hash_power[c('yield', 'admonish', 'abdicate')]

## End(Not run)
```

---

hash_sentiment_huliu *Hu Liu Polarity Lookup Table*

Description

A **data.table** dataset containing an augmented version of Hu & Liu's (2004) positive/negative word list as sentiment lookup values.

Usage

```
data(hash_sentiment_huliu)
```

Format

A data frame with 6874 rows and 2 variables

Details

- x. Words
- y. Sentiment values (+1, 0, -1.05, -1, -2), -2 indicate phrasing that is always negative (e.g., 'too much fun' and 'too much evil' both denote negative though the following word is positive and negative respectively).

References

Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. National Conference on Artificial Intelligence.

'https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html'

hash_sentiment_inquirer

Inquirer Polarity Lookup Table

Description

A **data.table** dataset containing an augmented version of General Inquirer's positive/negative word list as sentiment lookup values.

Usage

```
data(hash_sentiment_inquirer)
```

Format

A data frame with 3450 rows and 2 variables

Details

- x. Words
- y. Sentiment

References

http://www.wjh.harvard.edu/~inquirer/homecat.htm

hash_sentiment_jockers

Jockers Polarity Lookup Table

Description

A **data.table** dataset containing a modified version of Jocker's (2017) sentiment lookup table used in **syuzhet**.

Usage

```
hash_sentiment_jockers
```

Format

An object of class `data.table` (inherits from `data.frame`) with 10738 rows and 2 columns.

Details

- x. Words
- y. Sentiment values ranging between -1 and 1.

References

Jockers, M. L. (2017). Syuzhet: Extract sentiment and plot arcs from Text. Retrieved from https://github.com/mjockers/syuzhet

hash_sentiment_nrc *NRC Sentiment Polarity Table*

Description

A **data.table** dataset containing a filtered version of Mohammad & Turney', P. D.'s (2010) positive/negative word list as sentiment lookup values.

Usage

```
data(hash_sentiment_nrc)
```

Format

A data frame with 5468 rows and 2 variables

Details

- x. Words
- y. Sentiment values (+1, -1)

References

http://www.purl.com/net/lexicons

Mohammad, S. M. & Turney, P. D. (2010) Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon, In Proceeding of Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 26-34.

Examples

```
## Not run:
library(data.table)
hash_sentiment_nrc[c('happy', 'angry')]

## End(Not run)
```

hash_sentiment_senticnet *Augmented SenticNet Polarity Table*

Description

A **data.table** dataset containing an augmented version of Cambria, Poria, Bajpai,& Schuller's (2016) positive/negative word list as sentiment lookup values.

Usage

```
data(hash_sentiment_senticnet)
```

Format

A data frame with 23,633 rows and 2 variables

Details

- x. Words
- y. Sentiment values

References

Cambria, E., Poria, S., Bajpai, R. and Schuller, B. SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. In: COLING, pp. 2666-2677, Osaka (2016) `http://sentic.net/downloads`

`hash_sentiment_sentiword` *Augmented Sentiword Polarity Table*

Description

A **data.table** dataset containing an augmented version of Baccianella, Esuli and Sebastiani's (2010) positive/negative word list as sentiment lookup values. This list has be restructured to long format. A polarity value was assigned by taking the difference between the original data set's negative and positive attribution (`PosScore - NegScore`). All rows with a zero polarity were removed from the data set as well as any duplicated in the valence shifter's data set.

Usage

```
data(hash_sentiment_sentiword)
```

Format

A data frame with 20,099 rows and 2 variables

Details

- x. Words
- y. Sentiment values

References

Baccianella S., Esuli, A. and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. International Conference on Language Resources and Evaluation.

`http://sentiwordnet.isti.cnr.it/`

`hash_sentiment_vadar` *Filtered Vadar Polarity Table*

Description

A **data.table** dataset containing an filtered version of Hutto & Gilbert's (2014) positive/negative word list as sentiment lookup values.

Usage

```
data(hash_sentiment_vadar)
```

Format

A data frame with 7236 rows and 2 variables

Details

- x. Words
- y. Sentiment values

Vadar's Liscense:

References

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

https://github.com/cjhutto/vaderSentiment

hash_strength *Strength Lookup Key*

Description

A **data.table** containing a strength lookup key.

Usage

```
data(hash_strength)
```

Format

A data frame with 2085 rows and 2 variables

Details

- x. A power word
- y. A positive or negative value indicating the direction of strength in relation to the subject

References

http://www.wjh.harvard.edu/~inquirer/inqdict.txt

Examples

```
## Not run:
library(data.table)
hash_strength[c('yield', 'admonish', 'abdicate')]

## End(Not run)
```

hash_syllable *Syllable Counts*

Description

A **data.table** hash table dataset containing words and syllable counts.

Usage

```
data(hash_syllable)
```

Format

A data frame with 124603 rows and 2 variables

Details

- word. A character column of lower case words.
- syllables. The syllable counts per word.

References

Counts scraped from http://www.poetrysoup.com

Examples

```
## Not run:
library(data.table)
hash_syllable[c('yield', 'hurtful', 'admonishing', 'abdicate')]

## End(Not run)
```

---

hash_valence_shifters *Valence Shifters*

Description

A **data.table** dataset containing a vector of valence shifter words that can alter a polarized word's meaning and a numeric key for negators (1), amplifiers [intensifier] (2), de-amplifiers [downtoners] (3), and adversative conjunctions (4).

Usage

```
data(hash_valence_shifters)
```

Format

A data frame with 94 rows and 2 variables

Details

Valence shifters are words that alter or intensify the meaning of the polarized words and include negators and amplifiers. Negators are, generally, adverbs that negate sentence meaning; for example the word like in the sentence, "I do like pie.", is given the opposite meaning in the sentence, "I do not like pie.", now containing the negator not. Amplifiers (intensifiers) are, generally, adverbs or adjectives that intensify sentence meaning. Using our previous example, the sentiment of the negator altered sentence, "I seriously do not like pie.", is heightened with addition of the amplifier seriously. Whereas de-amplifiers (downtoners) decrease the intensity of a polarized word as in the sentence "I barely like pie"; the word "barely" deamplifies the word like. Adversative conjunction trump the previous clause (e.g., “He’s a nice guy but not too smart.”).

- x. Valence shifter
- y. Number key value corresponding to:

Valence Shifter	**Value**
Negator	1
Amplifier (intensifier)	2
De-amplifier (downtoner)	3
Adversative Contraction	4

key_abbreviation *Common Abbreviations*

Description

A dataset containing a hash lookup of common abbreviations and their long form.

Usage

```
data(key_abbreviation)
```

Format

A data frame with 138 rows and 2 variables

Details

- abbreviation. An abbreviation
- phrase. The equivalent word/phrase

References

http://public.oed.com/how-to-use-the-oed/abbreviations

key_contractions *Contraction Conversions*

Description

A dataset containing common contractions and their expanded form.

Usage

```
data(key_contractions)
```

Format

A data frame with 70 rows and 2 variables

Details

- contraction. The contraction word
- expanded. The expanded form of the contraction

key_grade *Grades Hash*

Description

A dataset containing letter grades and corresponding semantic meaning.

A dataset containing common grades.

Usage

```
data(key_grade)
```

```
data(key_grade)
```

Format

A data frame with 15 rows and 2 variables

Details

- x. Letter grade
- y. Semantic meaning of grade

- x. The graphic representation of the grade
- y. The meaning of the grade

key_rating *Ratings Data Set*

Description

A dataset containing common ratings.

Usage

```
data(key_rating)
```

Format

A data frame with 35 rows and 2 variables

Details

- x. The graphic representation of the rating
- y. The meaning of the rating

`key_sentiment_jockers` *Jockers Sentiment Key*

Description

A dataset containing an imported version of Jocker's (2017) sentiment lookup table used in **syuzhet**.

Usage

```
key_sentiment_jockers
```

Format

An object of class `data.frame` with 10748 rows and 2 columns.

Details

- word. Words
- value. Sentiment values ranging between -1 and 1.

References

Jockers, M. L. (2017). Syuzhet: Extract sentiment and plot arcs from Text. Retrieved from https://github.com/mjockers/syuzhet

`lexicon` *Lexicons for Text Analysis*

Description

A collection of lexical hash tables, dictionaries, and word lists.

`nrc_emotions` *NRC Emotions*

Description

A **data.table** dataset containing Mohammad & Turney', P. D.'s (2010) emotions word list as a binary table.

Usage

```
data(nrc_emotions)
```

Format

A data frame with 14182 rows and 9 variables

Details

- term. A term
- anger. Counts of anger anger
- anticipation. Counts of anticipation
- disgust. Counts of disgust
- fear. Counts of fear
- joy. Counts of joy
- sadness. Counts of sadness
- surprise. Counts of surprise
- trust. Counts of trust

References

http://www.purl.com/net/lexicons

Mohammad, S. M. & Turney, P. D. (2010) Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon, In Proceeding of Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, 26-34.

`pos_action_verb` *Action Word List*

Description

A dataset containing a vector of action words. This is a subset of the Moby project: Moby Part-of-Speech.

Usage

```
data(pos_action_verb)
```

Format

A character vector with 1569 elements

Details

From Grady Ward's Moby project: "This second edition is a particularly thorough revision of the original Moby Part-of-Speech. Beyond the fifteen thousand new entries, many thousand more entries have been scrutinized for correctness and modernity. This is unquestionably the largest P-O-S list in the world. Note that the many included phrases means that parsing algorithms can now tokenize in units larger than a single word, increasing both speed and accuracy."

References

`http://icon.shef.ac.uk/Moby/mpos.html`

pos_adverb *Adverb Word List*

Description

A dataset containing a vector of adverbs words. This is a subset of the Moby project: Moby Part-of-Speech.

Usage

```
data(pos_adverb)
```

Format

A list with 1 elements

Details

From Grady Ward's Moby project: "This second edition is a particularly thorough revision of the original Moby Part-of-Speech. Beyond the fifteen thousand new entries, many thousand more entries have been scrutinized for correctness and modernity. This is unquestionably the largest P-O-S list in the world. Note that the many included phrases means that parsing algorithms can now tokenize in units larger than a single word, increasing both speed and accuracy."

References

`http://icon.shef.ac.uk/Moby/mpos.html`

pos_df_irregular_nouns *Irregular Nouns Word Dataframe*

Description

A dataset containing a `data.frame` of irregular noun singular and plural forms.

Usage

```
data(pos_df_irregular_nouns)
```

Format

A data frame with 106 rows and 2 variables

Details

- singular. The singular form of the noun
- plural. The plural form of the noun

References

`http://www.esldesk.com/vocabulary/irregular-nouns`

`pos_df_pronouns` *Pronouns*

Description

A dataset containing pronouns categorized by type, singular, point_of_view, and use. Note that 'you', and 'yours' appear twice because 'you' can be singular or plural.

Usage

```
data(pos_df_pronouns)
```

Format

A data frame with 34 rows and 5 variables

Details

- pronoun. The pronoun.
- type. The pronoun type; either "personal", "reflexive", or "possessive".
- singular. logical. If `TRUE` the pronoun is singular, otherwise it's plural.
- point_of_view. The point of view; either "first", "second", or "third".

References

http://www.english-grammar-revolution.com/list-of-pronouns.html

`pos_interjections` *Interjections*

Description

A dataset containing a character vector of common interjections.

Usage

```
data(pos_interjections)
```

Format

A character vector with 139 elements

References

http://www.vidarholen.net/contents/interjections/

`pos_preposition` *Preposition Words*

Description

A dataset containing a vector of common prepositions.

Usage

```
data(pos_preposition)
```

Format

A character vector with 162 elements

`pos_unchanging_nouns` *Nouns that are the Same Plural/Singular*

Description

A dataset containing a character vector of nouns that have a single form for both singular and plural (or a singular/plural form does not exist).

Usage

```
data(pos_unchanging_nouns)
```

Format

A character vector with 95 elements

Details

These are a subset of irreguar nouns that are: plurale tantum, singularia tantum, or unchanging.

References

`https://www.vappingo.com/word-blog/101-words-that-are-both-plural-and-singular`

`profanity_alvarez` *Alejandro U. Alvarez's List of Profane Words*

Description

A dataset containing a character vector of profane words from Alejandro U. Alvarez.

Usage

```
data(profanity_alvarez)
```

Format

A character vector with 438 elements

References

https://web.archive.org/web/20130704010355/http://urbanoalvarez.es:80/blog/2008/04/04/bad-words-list/

`profanity_arr_bad` *Stackoverflow user2592414's List of Profane Words*

Description

A dataset containing a character vector of profane words from Stackoverflow user2592414.

Usage

```
data(profanity_arr_bad)
```

Format

A character vector with 343 elements

References

https://stackoverflow.com/a/17706025/1000343

`profanity_banned` *bannedwordlist.com's List of Profane Words*

Description

A dataset containing a character vector of profane words from bannedwordlist.com.

Usage

```
data(profanity_banned)
```

Format

A character vector with 77 elements

References

http://www.bannedwordlist.com

`profanity_google` *Google's List of Profane Words*

Description

A dataset containing a character vector of profane words from Google's "what do you love" project, compiled by Jamie Wilkinson.

Usage

```
data(profanity_google)
```

Format

A character vector with 451 elements

References

https://gist.github.com/jamiew/1112488

`profanity_von_ahn` *Luis von Ahn's List of Profane Words*

Description

A dataset containing a character vector of profane words from Luis von Ahn's research group.

Usage

```
data(profanity_von_ahn)
```

Format

A character vector with 1384 elements

References

`http://www.cs.cmu.edu/~biglou/resources`

`sw_buckley_salton` *Buckley & Salton Stopword List*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_buckley_salton)
```

Format

A character vector with 546 elements

Details

From Onix Text Retrieval Toolkit API Reference: "This stopword list was built by Gerard Salton and Chris Buckley for the experimental SMART information retrieval system at Cornell University. This stopword list is generally considered to be on the larger side and so when it is used, some implementations edit it so that it is better suited for a given domain and audience while others use this stopword list as it stands."

Note

Reduced from the original 571 words to 546.

References

`http://www.lextek.com/manuals/onix/stopwords2.html`

sw_dolch *Leveled Dolch List of 220 Common Words*

Description

Edward William Dolch's list of 220 Most Commonly Used Words by reading level.

Usage

```
data(sw_dolch)
```

Format

A character vector with 220 elements

Details

Dolch's Word List made up 50-75% of all printed text in 1936.

- Word. The word
- Level. The reading level of the word

References

Dolch, E. W. (1936). A basic sight vocabulary. Elementary School Journal, 36, 456-460.

sw_fry_100 *Fry's 100 Most Commonly Used English Words*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_fry_100)
```

Format

A character vector with 100 elements

Details

Fry's Word List: The first 25 make up about one-third of all printed material in English. The first 100 make up about one-half of all printed material in English. The first 300 make up about 65% of all printed material in English.

References

Fry, E. B. (1997). Fry 1000 instant words. Lincolnwood, IL: Contemporary Books.

`sw_fry_1000` *Fry's 1000 Most Commonly Used English Words*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_fry_1000)
```

Format

A character vector with 1000 elements

Details

Fry's 1000 Word List makes up 90% of all printed text.

References

Fry, E. B. (1997). Fry 1000 instant words. Lincolnwood, IL: Contemporary Books.

`sw_fry_200` *Fry's 200 Most Commonly Used English Words*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_fry_200)
```

Format

A character vector with 200 elements

Details

Fry's Word List: The first 25 make up about one-third of all printed material in English. The first 100 make up about one-half of all printed material in English. The first 300 make up about 65% of all printed material in English.

References

Fry, E. B. (1997). Fry 1000 instant words. Lincolnwood, IL: Contemporary Books.

`sw_fry_25` *Fry's 25 Most Commonly Used English Words*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_fry_25)
```

Format

A character vector with 25 elements

Details

Fry's Word List: The first 25 make up about one-third of all printed material in English. The first 100 make up about one-half of all printed material in English. The first 300 make up about 65% of all printed material in English.

References

Fry, E. B. (1997). Fry 1000 instant words. Lincolnwood, IL: Contemporary Books.

`sw_jockers` *Matthew Jocker's Expanded Topic Modeling Stopword List*

Description

A dataset containing a character vector of Jocker's stopwords he used for topic modeling. He later resorted to eliminating everything but nouns: `http://www.matthewjockers.net/2013/04/12/secret-recipe-for-topic-modeling-themes/`.

Usage

```
data(sw_jockers)
```

Format

A character vector with 5,902 elements

References

`http://www.matthewjockers.net/materials/uwm-2013`

sw_lucene *Lucene Stopword List*

Description

A dataset containing a character vector of Lucene's stopwords used in StopAnalyzer.ENGLISH_STOP_WORDS_SE.

Usage

```
data(sw_lucene)
```

Format

A character vector with 33 elements

Details

References

http://lucene.apache.org/core/4_0_0/analyzers-common/org/apache/lucene/analysis/core/StopFilter.html

sw_mallet *MALLET Stopword List*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_mallet)
```

Format

A character vector with 523 elements

Details

From MAchine Learning for LanguagE Toolkit

References

`http://mallet.cs.umass.edu`

sw_onix *Onix Text Retrieval Toolkit Stopword List 1*

Description

A stopword list containing a character vector of stopwords.

Usage

```
data(sw_onix)
```

Format

A character vector with 404 elements

Details

From Onix Text Retrieval Toolkit API Reference: "This stopword list is probably the most widely used stopword list. It covers a wide number of stopwords without getting too aggressive and including too many words which a user might search upon."

Note

Reduced from the original 429 words to 404.

References

`http://www.lextek.com/manuals/onix/stopwords1.html`

sw_python *Python Stopword List*

Description

A dataset containing a character vector of Python's stopwords.

Usage

```
data(sw_python)
```

Format

A character vector with 174 elements

Details

Copyright (c) 2014, Alireza Savand, Contributors All rights reserved.

References

`https://pypi.python.org/pypi/stop-words`

Index

∗Topic **datasets**
common_names, 3
discourse_markers_alemany, 4
dodds_sentiment, 5
freq_first_names, 5
freq_last_names, 6
function_words, 6
grady_augmented, 7
hash_emoticons, 7
hash_grady_pos, 8
hash_lemmas, 9
hash_power, 9
hash_sentiment_huliu, 10
hash_sentiment_inquirer, 11
hash_sentiment_jockers, 11
hash_sentiment_nrc, 12
hash_sentiment_senticnet, 12
hash_sentiment_sentiword, 13
hash_sentiment_vadar, 14
hash_strength, 15
hash_syllable, 15
hash_valence_shifters, 16
key_abbreviation, 17
key_contractions, 17
key_grade, 18
key_rating, 18
key_sentiment_jockers, 19
nrc_emotions, 19
pos_action_verb, 20
pos_adverb, 21
pos_df_irregular_nouns, 21
pos_df_pronouns, 22
pos_interjections, 22
pos_preposition, 23
pos_unchanging_nouns, 23
profanity_alvarez, 24
profanity_arr_bad, 24
profanity_banned, 25
profanity_google, 25
profanity_von_ahn, 26
sw_buckley_salton, 26
sw_dolch, 27
sw_fry_100, 27
sw_fry_1000, 28
sw_fry_200, 28
sw_fry_25, 29
sw_jockers, 29
sw_lucene, 30
sw_mallet, 30
sw_onix, 31
sw_python, 31

available_data, 3

common_names, 3

discourse_markers_alemany, 4
dodds_sentiment, 5

freq_first_names, 5
freq_last_names, 6
function_words, 6

grady_augmented, 7

hash_emoticons, 7
hash_grady_pos, 8
hash_lemmas, 9
hash_power, 9
hash_sentiment_huliu, 10
hash_sentiment_inquirer, 11
hash_sentiment_jockers, 11
hash_sentiment_nrc, 12
hash_sentiment_senticnet, 12
hash_sentiment_sentiword, 13
hash_sentiment_vadar, 14
hash_strength, 15
hash_syllable, *7*, 15
hash_valence_shifters, 16

key_abbreviation, 17
key_contractions, *6*, 17
key_grade, 18
key_rating, 18
key_sentiment_jockers, 19

lexicon, 19
lexicon-package (lexicon), 19

nrc_emotions, 19

package-lexicon (lexicon), 19
pos_action_verb, 20
pos_adverb, 21
pos_df_irregular_nouns, 21
pos_df_pronouns, 22
pos_interjections, 22
pos_preposition, 23
pos_unchanging_nouns, 23
profanity_alvarez, 24
profanity_arr_bad, 24
profanity_banned, 25
profanity_google, 25
profanity_von_ahn, 26

sw_buckley_salton, 26
sw_dolch, 27
sw_fry_100, 27
sw_fry_1000, 28
sw_fry_200, 28
sw_fry_25, 29
sw_jockers, 29
sw_lucene, 30
sw_mallet, 30
sw_onix, 31
sw_python, 31