Package ‘healthforum’

October 3, 2019

Type Package

Title Scrape Patient Forum Data

Version 0.1.0

Description Scrape data from Patient Forum <https://patient.info/forums> by entering urls. It will return a data frame containing text, user names, like counts, reply counts, etc.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports rvest, magrittr, xml2, purrr, tokenizers, stringr

Depends R (>= 3.5.0)

RoxygenNote 6.1.1

Suggests testthat (>= 2.1.0)

NeedsCompilation no

Author Lingshu Hu [aut, cre] (<https://orcid.org/0000-0003-0304-882X>),
Michael W. Kearney [ctb] (<https://orcid.org/0000-0002-0730-4694>)

Maintainer Lingshu Hu <lingshu.hu@hotmail.com>

Repository CRAN

Date/Publication 2019-10-03 08:20:02 UTC

R topics documented:

count_medical_terms *Count medical glossaries*

Description

Count the number and ratio of medical glossaries used in the text.

Usage

```
count_medical_terms(text)
```

Arguments

`text`	Input data. Should be character vector or data frame with character variable of interest named "text".

Details

The medical glossary dictionary was obtained from Aristotelis P. <https://github.com/glutanimate/wordlist-medicalterms-en>. It is based on two prominent medical dictionary projects: OpenMedSpel by R. Robinson of e-MedTools (Version 2.0.0, released 2014-01-21) <http://www.e-medtools.com/openmedspel.html> and MTH-Med-Spel-Chek by Rajasekharan N. of MT-Herald (released 2014-04-02) <http://mtherald.com/free-medical-spell-checker-for-microsoft-word-custom-dictionary/>.

Value

A data frame containing the number of medical words, the number of total words, and the ratio of medical words to total words.

Examples

```
## create a character vector
medical_text <- c(
"No, it isn't possible to predict anything before the result of your biopsy is received.",
"Thank you for the nice reply! Very thoughtful answer that did ease my fears!",
"Can't help regards the meds.  Just want to give support.")

## get the medical glossaries counts from a character vector
count_medical_terms(text = medical_text)

## creat a data frame with a character vector named "text"
df <- data.frame(
  id = c(1, 2, 3),
  text = c("No, it isn't possible to predict anything before the result
           of your biopsy is received.",
          "Thank you for the nice reply! Very thoughtful answer that
           did ease my fears!",
          "Can't help regards the meds.  Just want to give support."),
  stringsAsFactors = FALSE
)

## get the medical glossaries counts from a data frame with "text" variable
count_medical_terms(df)
```

medical_words *The English medical glossary dictionary*

Description

A dataset containing two prominent medical dictionary projects OpenMedSpel by R. Robinson of e-MedTools (Version 2.0.0, released 2014-01-21) <http://www.e-medtools.com/openmedspel.html> and MTH-Med-Spel-Chek by Rajasekharan N. of MT-Herald (released 2014-04-02) <http://mtherald.com/free-medical-spell-checker-for-microsoft-word-custom-dictionary/>.

Usage

```
medical_words
```

Format

A data frame with 94959 rows and 2 variables:

word medical words

value weight of the medical words ...

Source

https://github.com/glutanimate/wordlist-medicalterms-en

scrape_groups_by_category *Scrape groups by category*

Description

Get posts and all the replies to the posts from groups of a category by entering category name or URL

Usage

```
scrape_groups_by_category(cat, post_number_per_group = NULL, ...)
```

Arguments

`cat` The category name (lower case, replace space with -) or category URL

`post_number_per_group`
The number of random posts to scrape per group. Default is NULL, which means scrape the total number of posts in each group

`...` optional arguments to FUN.

Value

A data frame

Examples

```
## Get the posts data of groups whose names starting with the letter "a" and "z"
scrape_groups_by_category(cat = "health-promotion", post_number_per_group = 1)
cat_url = "https://patient.info/forums/categories/health-promotion-17"
scrape_groups_by_category(cat = cat_url, post_number_per_group = 1)
```

scrape_groups_by_initial_letter

Scrape groups by initial letter

Description

Get posts and all the replies to the posts from groups by entering the initial letter of group names

Usage

```
scrape_groups_by_initial_letter(index, post_number_per_group = NULL, ...)
```

Arguments

`index` The initial letter of groups. Can be one letter or a vector of letters.

`post_number_per_group`
The number of random posts to scrape per group. Default is NULL, which means scrape the total number of posts in each group

`...` optional arguments to FUN.

Value

A data frame

Examples

```
## Get the posts data of groups whose names starting with the letter "a" and "z"
scrape_groups_by_initial_letter(index = "x", post_number_per_group = 1)
```

---

scrape_one_group *Scrape one group*

Description

Get all the posts containing all the replies to those posts from one group by entering its url.

Usage

```
scrape_one_group(group_url, random_post_number = NULL,
  random_seed = NULL, ...)
```

Arguments

`group_url` URL to the page to scrape.

`random_post_number` The number of random posts to scrape. Default is NULL, which means scrape the total number of posts.

`random_seed` A random number used to set the random seed to reproduce the work.

`...` optional arguments to FUN.

Value

A data frame

Examples

```
## get the data of 5 random posts from the group "Angiotensin II Receptor Blockers"
group_url = "https://patient.info/forums/discuss/browse/angiotensin-ii-receptor-blockers-3037"
scrape_one_group(group_url = group_url, random_post_number = 5)
```

scrape_one_post *Scrape one initial post*

Description

Get the data from one initial post by entering its url

Usage

```
scrape_one_post(url, From = 1L, To = Inf, get_user_info = TRUE)
```

Arguments

url	URL to the post to scrape
From	The starting page number. Default is the first page
To	The ending page number. Default (Inf) is the last page (so all pages)
get_user_info	Get users' profile information. It includes the date of joining the forum and the total posts they have sent. The default is TRUE.

Value

A data frame

Examples

```
## get two pages of data from the post titled "Can Gastritis be cured?"
post_url = "https://patient.info/forums/discuss/can-gastritis-be-cured--613999"
scrape_one_post(url = post_url, From = 1, To = 2)
```

scrape_user_posts *Scrape a user's posts*

Description

Get all posts/replies one user has sent by his/her user name

Usage

```
scrape_user_posts(user_profile_url, type = c("both", "replies",
  "topic_post"))
```

Arguments

`user_profile_url`
The URL of a user's profile page.

`type`
Choose a type of posts to be scraped. It includes "replies", i.e., a user's replies to others' posts, or "topic_post", i.e., a user's initial posts. The default is to get "both"

Value

A data frame

Examples

```
user_profile_url <- "https://patient.info/forums/profiles/utgh4k33-1264038"
scrape_user_posts(user_profile_url = user_profile_url, type = "both")
```

Index