Package 'storywranglr'

July 23, 2025

Type Package

Title Explore Twitter Trends with the 'Storywrangler' API

Version 0.2.0

Author Christopher Belanger [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2070-5721>)

Maintainer Christopher Belanger <christopher.a.belanger@gmail.com>

Description An interface to explore trends in Twitter data using the 'Storywrangler' Application Programming Interface (API), which can be found here: <https://github.com/janeadams/storywrangler>.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, httr, jsonlite, tibble, urltools

RoxygenNote 7.1.1

URL https://github.com/chris31415926535/storywranglr

BugReports https://github.com/chris31415926535/storywranglr/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-13 14:00:02 UTC

Contents

ngrams *Explore Twitter trends with the Storywrangler ngrams API*

Description

Storywrangler's ngrams API lets you search a large historical database of Twitter data for daily usage statistics about strings of one, two, and three words (1-grams, 2-grams, and 3-grams respectively).

This function returns daily historical usage statistics for a given query over data set's entire time range.

For more details about Storywrangler, please see:

- API documentation: https://github.com/janeadams/storywrangler
- Academic paper describing uses: https://advances.sciencemag.org/content/7/29/eabe6534.full

Usage

```
ngrams(
  query,
  metric = c("rank", "freq"),
  language = "en",
  rt = c(FALSE, TRUE),
  fill_dates = FALSE
)
```

Arguments

query	Character string with the n-gram(s) to query. One, two, or three words separated by spaces will run query that string as a 1-gram, 2-gram, or 3-gram respectively. More than three space-separated words will be treated as separate queries for individual 1-grams.
metric	The measure of lexical fame to return: accepts values rank (default) and freq. *Note:* API returns both by default.
language	Two-letter code for the language to search. Defaults to en.
rt	Boolean for whether to include retweets.
fill_dates	Boolean, defaults to FALSE. The Storywrangler ngrams API only returns rows for dates when it detected any ngram usage. By default, this function passes along that data. If the parameter fill_dates is set to TRUE, this function adds rows with NA values for each day between the earliest and latest dates in the response. Note that this is closer to Storywrangler's behaviour if you download ngram statistics from the web interface.

Value

A tibble with the API query and response. If the API returns no data, this function returns a 0-row tibble.

Examples

```
## Not run:
# Query a simple 1-gram about the populatity of potatoes
result <- ngrams("potatoes")

# Query a 2-gram about the popularity of potato chips
result <- ngrams("potato chips")

# Query *four* 1-grams related to potatoes
# Note! If there are more than 3 words, they are all treated as 1-grams
result <- ngrams("potato potahto spud taters")

## End(Not run)
```

---

zipf *Explore Twitter trends with the Storywrangler zipf API*

Description

Storywrangler's ngrams API lets you search a large historical database of Twitter data for daily usage statistics about strings of one, two, and three words (1-grams, 2-grams, and 3-grams respectively).

This function will query the API for a specific date to return the rank and frequency data for its top n ngrams. Please note that queries of over 1000 ngrams will take a long time to load.

For more details about Storywrangler, please see:

- API documentation: https://github.com/janeadams/storywrangler
- Academic paper describing uses: https://advances.sciencemag.org/content/7/29/eabe6534.full

Usage

```
zipf(date, max = 100, language = "en", ngrams = c(1, 2, 3))
```

Arguments

date	The date to query, in either character "YYYY-MM-DD" or Date format.
max	The maximum number of ngrams to return. Defaults to 100.
language	The two-letter code of the language to query. Defaults to "en".
ngrams	Integer specifying the type of n-grams to return. Accepts 1, 2, and 3, and defaults to 1.

Value

A tibble with the API query and response.

Examples

```
## Not run:
# Get top English 2-grams for January 6, 2021
 result <- zipf("2021-01-06", ngrams = 2)

## End(Not run)
```

Index