

Package ‘birdnik’

May 27, 2018

Type Package

Title Connector for the Wordnik API

Version 0.1.1

Author Oliver Keyes

Maintainer Oliver Keyes <ironholds@gmail.com>

Description A connector to the API for 'Wordnik' <<https://www.wordnik.com>>, a dictionary service that also provides bigram generation, word frequency data, and a whole host of other functionality.

License MIT + file LICENSE

LazyData TRUE

RoxygenNote 6.0.1

Suggests testthat

Depends httr

Date 2018-05-26

URL <https://github.com/Ironholds/birdnik>

BugReports <https://github.com/Ironholds/birdnik/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-27 13:56:32 UTC

R topics documented:

birdnik	2
get_pos	2
related_words	3
word_bigrams	3
word_definitions	4
word_frequency	5
word_pronunciations	6
word_top_example	7

Index	8
--------------	----------

birdnik	<i>Connector to the Wordnik API</i>
---------	-------------------------------------

Description

This package provides functions for the Wordnik API.

See Also

[word_definitions](#), [word_top_example](#), [related_words](#), [word_pronunciations](#), [word_frequency](#), [word_bigrams](#), and [birdnik_get](#).

get_pos	<i>Extract components from word metadata</i>
---------	--

Description

these functions allow you to extract particular components from wordnik definitions; `get_pos` the unique parts-of-speech, `get_text` the unique definitions for each word, and `get_score` the unique word scores.

Usage

```
get_pos(results)
```

```
get_text(results)
```

```
get_score(results)
```

Arguments

`results` the results of a call to [word_definitions](#).

Value

a data.frame of two columns - `word` (containing the word) and a second column of `score`, `text` or `part_of_speech` depending on the function you called.

Examples

```
## Not run:  
# Retrieve the unique parts-of-speech for 'no' and 'kings'  
no_kings <- word_definitions(key = "madeupkey", words = c("no", "kings"))  
get_pos(no_kings)  
  
## End(Not run)
```

related_words	<i>Retrieve related words</i>
---------------	-------------------------------

Description

related_words grabs terms that are in some way associated with the words argument.

Usage

```
related_words(key, words, use_canonical = FALSE, ...)
```

Arguments

key	a Wordnik API key. These can be obtained at the Wordnik developer portal .
words	a vector of words.
use_canonical	whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.
...	further arguments to pass to httr's GET.

Value

a data.frame of 3 columns: word (the original term), type (the type of relationship) and related_word.

Examples

```
## Not run:
cat_adjacent <- related_words(key = "madeupkey", words = "cat")

## End(Not run)
```

word_bigrams	<i>Retrieve word bigrams</i>
--------------	------------------------------

Description

word_bigrams grabs bigrams for whatever words you want out of the wordnik database, along with the match strength (weighted with and without factoring in word length).

Usage

```
word_bigrams(key, words, use_canonical = FALSE, min_wlmi = 0, limit = 5,
...)
```

Arguments

key	a Wordnik API key. These can be obtained at the Wordnik developer portal .
words	a vector of words.
use_canonical	whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.
min_wlmi	the minimum (word-length weighted) strength of the similarity between the bi-gram words.
limit	the maximum number of items to return for any one word.
...	further arguments to pass to httr's GET.

Value

a data.frame of 5 columns; word, first_gram, second_gram, mi (the strength of the relationship) and wlmi (the strength of the relationship, weighted by word length).

Examples

```
## Not run:

dog_associations <- word_bigrams(key = "madeupkey", words = "dog")
#mauling and fighting. But also biscuits!

## End(Not run)
```

word_definitions	<i>Retrieve definitions of a word</i>
------------------	---------------------------------------

Description

word_definitions retrieves the overall metadata for each word in words, including actual definitions.

Usage

```
word_definitions(key, words, use_canonical = FALSE, source_dicts = "all",
  limit = 200, ...)
```

Arguments

key	a Wordnik API key. These can be obtained at the Wordnik developer portal .
words	a vector of words.

use_canonical	whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.
source_dicts	the dictionaries to use; options are any permutation of "ahd", "century", "wiktionary", "wordnet" and "webster". "all" (the default) searches all of them.
limit	the maximum number of items to return for any one word.
...	further arguments to pass to httr's GET.

Value

a list structure containing definitions for each word in words, along with associated metadata, which can be manipulated with [birdnik_get](#).

See Also

[birdnik_get](#) for ways to conveniently access individual chunks of the returned dataset.

Examples

```
## Not run:
cats <- word_definitions(key = "madeupkey", words = "cat")

## End(Not run)
```

word_frequency	<i>Count the per-year frequency of words</i>
----------------	--

Description

word_frequency provides, for a vector of words, the number of appearances each word made per year in the source texts Wordnik uses.

Usage

```
word_frequency(key, words, use_canonical = FALSE, start_year = 1800,
  end_year = 2012, ...)
```

Arguments

key	a Wordnik API key. These can be obtained at the Wordnik developer portal .
words	a vector of words.
use_canonical	whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.

start_year the earliest year to get frequencies for. 1800 (the earliest accepted value) by default.

end_year the latest year to get frequencies for. 2012 (the latest accepted value) by default.

... further arguments to pass to htrr's GET.

Value

a data.frame of 3 columns; word, year and frequency.

Examples

```
## Not run:
cats_versus_dogs <- word_frequency(key = "notarealkey", words = c("cat", "dog"))

## End(Not run)
```

word_pronunciations *Retrieve Pronunciations for Words*

Description

word_pronunciations grabs, well, pronunciations for a series of words.

Usage

```
word_pronunciations(key, words, use_canonical = FALSE, source_dicts = "all",
  pronunciation_type = "all", limit = 50, ...)
```

Arguments

key a Wordnik API key. These can be obtained at the [Wordnik developer portal](#).

words a vector of words.

use_canonical whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.

source_dicts the dictionaries to use; options are any permutation of "ahd", "century", "wiktionary", "wordnet" and "webster". "all" (the default) searches all of them.

pronunciation_type the type of pronunciations to return, if available; options are "ahd", "IPA", "arpa-bet" and "gcide-diacritical". "all" (the default) provides any that are available.

limit the maximum number of items to return for any one word.

... further arguments to pass to htrr's GET.

Examples

```
## Not run:
cats_vs_dogs <- word_pronunciations(key = "fakekey", words = c("cat", "dog"))

## End(Not run)
```

word_top_example	<i>Retrieve a Word's Top Example</i>
------------------	--------------------------------------

Description

word_top_example grabs, as it suggests, the top example for each entry in words.

Usage

```
word_top_example(key, words, use_canonical = FALSE, ...)
```

Arguments

key	a Wordnik API key. These can be obtained at the Wordnik developer portal .
words	a vector of words.
use_canonical	whether to use the stemmed, canonical form of the word (i.e. 'cat', for 'cats') instead of the actual word. FALSE by default. Note that for pronunciations specifically, setting it to TRUE may get a wider range of pronunciations; you should experiment.
...	further arguments to pass to httr's GET.

Value

a data.frame of 5 columns; provider, year, rating, word and example.

Examples

```
## Not run:
examples <- word_top_example(key = "madeupkey", words = c("cat", "dog", "turnip"))

## End(Not run)
```

Index

birdnik, [2](#)
birdnik-package (birdnik), [2](#)
birdnik_get, [2, 5](#)
birdnik_get (get_pos), [2](#)

get_pos, [2](#)
get_score (get_pos), [2](#)
get_text (get_pos), [2](#)

related_words, [2, 3](#)

word_bigrams, [2, 3](#)
word_definitions, [2, 4](#)
word_frequency, [2, 5](#)
word_pronunciations, [2, 6](#)
word_top_example, [2, 7](#)