

Package: rwstats (via r-universe)

August 21, 2024

Title Chinese Character Frequency in Real World

Version 0.1

Description It contains Chinese character frequency data based on news data from 2017 to 2019.

Depends R (>= 3.3.2)

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 7.0.1

Repository <https://schirp.r-universe.dev>

RemoteUrl <https://github.com/schirp/rwstats>

RemoteRef HEAD

RemoteSha 6f8a70a7f7c9b08bb919e8c29213945157571c06

Contents

| | |
|------------------------------|---|
| fiveChar | 2 |
| fourChar | 2 |
| oneChar | 3 |
| threeChar | 3 |
| twoChar | 4 |
| wordMiner.next | 4 |
| wordMiner.previous | 5 |

| | |
|--------------|----------|
| Index | 6 |
|--------------|----------|

fiveChar *Chinese Word Frequency with five Characters*

Description

This dataframe includes the usage frequency and occurrence of Chinese word

Usage

fiveChar

Format

A data frame with 3 variables:

character target character

freq occurrence of target character

pct frequency of target character

Examples

fiveChar

fourChar *Chinese Word Frequency with four Characters*

Description

This dataframe includes the usage frequency and occurrence of Chinese word

Usage

fourChar

Format

A data frame with 3 variables:

character target character

freq occurrence of target character

pct frequency of target character

Examples

fourChar

| | |
|---------|--|
| oneChar | <i>Chinese Word Frequency with one Character</i> |
|---------|--|

Description

This dataframe includes the usage frequency and occurrence of Chinese word

Usage

oneChar

Format

A data frame with 3 variables:

character target character

freq occurrence of target character

pct frequency of target character

Examples

oneChar

| | |
|-----------|---|
| threeChar | <i>Chinese Word Frequency with three Characters</i> |
|-----------|---|

Description

This dataframe includes the usage frequency and occurrence of Chinese word

Usage

threeChar

Format

A data frame with 3 variables:

character target character

freq occurrence of target character

pct frequency of target character

Examples

threeChar

twoChar

Chinese Word Frequency with two Characters

Description

This dataframe includes the usage frequency and occurrence of Chinese word

Usage

```
twoChar
```

Format

A data frame with 3 variables:

character target character

freq occurrence of target character

pct frequency of target character

Examples

```
twoChar
```

wordMiner.next

Occurrence and frequency of the next word

Description

This is a function that lists the next word appearing frequency by a given chinese character based on real world chinese character frequency Statistics table.

Usage

```
wordMiner.next(candidateWord, topN = 10)
```

Arguments

candidateWord A single Chinese character

topN The number of rows of output dataframe

Value

A dataframe containing the next word and its occurrence and frequency

Examples

```
## Not run:
wordMiner.next(strsplit(levels(twoChar$character)[988], "")[[1]][1])
wordMiner.next(strsplit(levels(twoChar$character)[988], "")[[1]][2], 20)

## End(Not run)
```

| | |
|--------------------|--|
| wordMiner.previous | <i>Occurrence and frequency of the previous word</i> |
|--------------------|--|

Description

This is a function that lists the previous word appearing frequency by a given chinese character based on real world chinese character frequency Statistics table.

Usage

```
wordMiner.previous(candidateWord, topN = 10)
```

Arguments

| | |
|---------------|--|
| candidateWord | A single Chinese character |
| topN | The number of rows of output dataframe |

Value

A dataframe containing the previous word and its occurrence and frequency

Examples

```
## Not run:
wordMiner.previous(strsplit(levels(twoChar$character)[988], "")[[1]][1])
wordMiner.previous(strsplit(levels(twoChar$character)[988], "")[[1]][2], 20)

## End(Not run)
```

Index

* datasets

fiveChar, [2](#)

fourChar, [2](#)

oneChar, [3](#)

threeChar, [3](#)

twoChar, [4](#)

fiveChar, [2](#)

fourChar, [2](#)

oneChar, [3](#)

threeChar, [3](#)

twoChar, [4](#)

wordMiner.next, [4](#)

wordMiner.previous, [5](#)