shizen$_{g}engoDocumentation$

Release 0.1.5

Raoul Biagioni

Sep 01, 2020

Contents

`shizen_gengo` is a Python library for simplifying common hands on NLP tasks.

CHAPTER 1

Contents:

1.1 About

`shizen_gengo` is a Python library for simplifying common hands on NLP tasks.

1.2 Installation

Create a Virtual Environment (Recommended)

With Conda:

```
$ conda create --name gengo python=3.7
$ source activate gengo
(gengo) $
```

Pip Install

```
(gengo) $ pip install shizen-gengo
```

1.3 API

1.3.1 Explore

Functions to search for text in a pandas dataframe column.

Explore Utils

search(df[, df_col, tok])	Search dataframe column and return rows that contain the specified token.
check_missing_values(df)	Returns a dataframe with missing values count for all columns sorted in descending order.

search

search (*df*, *df_col=''*, *tok=''*)

Search dataframe column and return rows that contain the specified token.

Parameters

- **df** – dataframe
- **df_col** – string
- **tok** – string

Returns dataframe

Example

```
>>> from shizen_gengo.explore import explore_utils
>>> df.col_name = explore_utils.search(df, 'col_name', 'keyword')
```

check_missing_values

check_missing_values (*df*)

Returns a dataframe with missing values count for all columns sorted in descending order.

Parameters **df** – dataframe

Returns dataframe

1.3.2 Preprocess Dataframe

Functions to modify a pandas dataframe e.g. rename columns, to standardise column headers. or to fill missing values with a string.

Preprocess Dataframe Utils

rename_column(df_col_names, before, after)	Rename column.
standardise_column_headers(df_col_names[, ...])	Make dataframe headers lowercase and replace spaces by underscores.
fill_missing(df_col[, val])	Fill missing values with string of choice.

rename_column

rename_column (*df_col_names*, *before*, *after*)

Rename column.

Parameters

- **df_col_names** – dataframe column names <class 'pandas.core.indexes.base.Index'>
- **before** – string
- **after** – string

Returns dataframe column names <class 'pandas.core.indexes.base.Index'>

Example

```
>>> from shizen_gengo.preprocess_dataframe import dataframe_utils
>>> df.columns = dataframe_utils.rename_column(df.columns, 'Description2',
↪'Description')
```

standardise_column_headers

standardise_column_headers(*df_col_names*, *before=None*, *after=None*)

Make dataframe headers lowercase and replace spaces by underscores.

The function allows to specify custom replacements.

For example: *Group - Primary* can be changed to *group_primary* by calling

df.columns = utils.standardise_column_headers(df.columns, before='_-_', after='_')

Parameters

- **df_col_names** – dataframe column names <class 'pandas.core.indexes.base.Index'>
- **before** – string
- **after** – string

Returns dataframe column names <class 'pandas.core.indexes.base.Index'>

fill_missing

fill_missing(*df_col*, *val='MISSING'*)

Fill missing values with string of choice. Default is "MISSING".

The function first replaces cells with an empty string and/or cells with only spaces with *np.nan*.

Parameters

- **df_col** – a single dataframe column
- **val** – string

Returns a single dataframe column

1.3.3 Preprocess Text

Functions to clean text in a pandas dataframe column.

Preprocess Text Utils

remove_newline_chars(df_col)	Remove new line and/or carriage return from dataframe column.
remove_digits(df_col)	Remove digits.
remove_non_char(df_col)	Remove non-alphabetic tokens: `[#<>=.,;:$&*|?'"-()%]`
custom_replace(df_col[, change_from, change_to])	Replace tokens.
remove_url(df_col)	Remove hyperlink / url.
remove_email(df_col)	Remove email address.
remove_consecutive_spaces(df_col)	Remove consecutive white spaces.
remove_stopwords(df_col)	Remove stopwords.
remove_accented_chars(df_col)	Remove accented characters.
remove_punctuation(df_col)	Remove punctuation.
remove_repeating_letters(df_col)	Remove repeating letters with a minimum threshold of 2.
clean_text(df_col)	Function that combines all text pre-processing tasks.

remove_newline_chars

remove_newline_chars(*df_col*)

Remove new line and/or carriage return from dataframe column.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

Example

```
>>> from shizen_gengo.preprocess_text import text_utils
>>> df.col_name = text_utils.remove_newline_chars(df.col_name)
```

remove_digits

remove_digits(*df_col*)

Remove digits.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_non_char

remove_non_char(*df_col*)

Remove non-alphabetic tokens: `[#<>=.,;:$&*|?'"-()%]`

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

custom_replace

custom_replace(*df_col*, *change_from=''*, *change_to=''*)

Replace tokens.

Parameters

- **df_col** – a single dataframe column <class 'pandas.core.series.Series'>
- **change_from** – string
- **change_to** – string

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_url

remove_url(*df_col*)

Remove hyperlink / url.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_email

remove_email(*df_col*)

Remove email address.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_consecutive_spaces

remove_consecutive_spaces(*df_col*)

Remove consecutive white spaces.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_stopwords

remove_stopwords(*df_col*)

Remove stopwords.

Also removes carriage return *r* and line break *n* characters.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_accented_chars

remove_accented_chars(*df_col*)
Remove accented characters.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_punctuation

remove_punctuation(*df_col*)
Remove punctuation.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

remove_repeating_letters

remove_repeating_letters(*df_col*)
Remove repeating letters with a minimum threshold of 2.

The threshold prevents repeated letters in names e.g. Aaron to be preserved.

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

clean_text

clean_text(*df_col*)
Function that combines all text pre-processing tasks.

- remove accented_chars
- remove punctuation
- remove repeating_letters
- remove newline_chars
- remove digits
- remove non_char
- remove url
- remove email
- remove consecutive_spaces
- remove stopwords

Parameters **df_col** – a single dataframe column <class 'pandas.core.series.Series'>

Returns a single dataframe column <class 'pandas.core.series.Series'>

CHAPTER 2

Changelog

- v 0.1.5 add clean_text function to perform all pre-process text tasks in one go.
- v 0.1.4 minor bug fix (remove print statement).
- v 0.1.3 improve function to remove new line and carriage return characters.
- v 0.1.2 further development of new or improvement of existing functions and docstring.
- v 0.1.1 pre-release.

Python Module Index

e

p

Index