excel-ngrams

Matthew Oliver

Feb 06, 2022

CONTENTS

CHAPTER

ONE

LICENSE

MIT License

Copyright (c) 2021 Matt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER

TWO

REFERENCE

2.1 excel_ngrams.console

Command-line interface.

2.2 excel_ngrams.grammer

Return dataframe of ngrams from list of words.

class excel_ngrams.grammer.**Grammer**(*terms_list*)

Class that returns n-grams from text as a list of strings.

Words are delineated by white space and punctuation. Using Spacy's NLP pipe and NLTK's ngrams function to generate ngrams within a given word length range and output them to a Pandas DataFrame for writing to an output file.

term_list

List of text as strings (one or more).

_nlp and _stopwords are shared across all instances, but is loaded by the constructor to avoid loading is in cases where it isn't needed.

combine_dataframes(*dataframes*)

Creates single multi-column dataframe.

Takes the terms and frequency values for dataframes constructed from ngrams of various lengths and combines them into a single dataframe, e.g single term and values, bigrams and values, trigrams and values, etc.

Parameters **dataframes** (*list*) – List of pd.DataFrames containing the dataframes to be merged, side by side.

Returns Single combined dataframe from list of dataframes.

Return type pd.DataFrame

df_from_terms(*ngram_tuples*)
Creates DataFrame from lists of terms and values as tuple.

Calls terms_to_columns on ngram_tuple to unpack them.

Parameters **ngram_tuples** (*list*) – list of tuple`[:obj:`tuple [str], int]. Results from get_ngrams.

Returns

Pandas DataFrame comprising a column of terms and a column of frequency values for those terms.

Return type df(pd.DataFrame)

get_ngrams(*n*, *top_n_results=250*, *stopwords=True*)
Create tuple with terms and frequency from list.

List of terms is tokenised using Spacy's NLP pipe, set to lowercase and ngrams are calculated with NLTK's ngrams function.

Parameters

- **n** (*int*) – The length of phrases to analyse.
- **top_n_results** (*int*) – The number of results to return. Default is 150.
- **stopwords** (*bool*) – flag to indicate removal of stopwords. Default is True.

Returns List of tuples containing term(s) and values.

Return type list of :obj:`tuple`[:obj:`tuple`[str, ...], int]

in_stop_words(*spacy_token_text*)
Check if word appears in stopword set.

Parameters **spacy_token_text** (*str*) – The text attribute of the Spacy token being passed to the method.

Returns Whether text is present in stopwords.

Return type bool

ngram_range(*max_n*, *n=1*, *top_n_results=250*, *stopwords=True*)
Gets ngram terms and outputs for a range of phrase lengths.

Gets ngrams from single terms as default up to desired maximum phrase length and creates Pandas DataFrame from results.

Parameters

- **max_n** (*int*) – The longest phrase length desired in output.
- **n** (*int*) – The minimum term length. Default is 1 (single term).
- **top_n_results** (*int*) – The number of rows of results to return. Default set to 150.
- **stopwords** (*bool*) – flag to indicate removal of stopwords. Default is True.

Returns

Combined dataframe of all results from various term lengths to desired maximum.

Return type pd.DataFrame

remove_escaped_chars(*text*)
Remove newline and tab chars from string list.

Parameters **text** (List of str) – Terms list to be cleaned of specific chars.

Returns

Terms list without specific chars.

Return type without_newlines(List of str)

terms_to_columns(*ngram_tuples*)
Returns term/value tuples as two lists.

Parameters **ngram_tuples** (*list*) – list of tuple`[:obj:`tuple [str], int]. Results from get_ngrams.

Returns

Terms, concatinated into single string for multi-word terms, returned as list.

value_col(list of int): Term frequencies as list. Lists are returned together as tuple containing both lists.

Return type term_col(list of str)

A project to analyse a column of text in an Excel document and return a CSV file with the most common ngrams from that text. Output file is returned to the same directory as the input file. You can choose the maximum n-gram length, and maximum number of results (rows) returned.

Words are tokenised with Spacy and ngrams are generated with NLTK.

CHAPTER

THREE

INSTALLATION

To install the Excel Ngrams Project, run this command in your terminal:

```
$ pip install excel-ngrams
```

CHAPTER

FOUR

USAGE

Excel Ngram's usage looks like:

```
$ excel-ngrams [OPTIONS]
```

-f <file-path>, **--file-path** <file-path>
: The path to the input Excel file to be parsed for words to generate ngrams.

-s <sheet-name>, **--sheet-name** <sheet-name>
: The name of the Excel sheet that contains the column of text to be analysed. By default, this is the first sheet in a document where none of the sheets have names. If any sheets are named, you must specify the one that contains the column to be analysed.

-c <column-name>, **--column-name** <column-name>
: The name of the column containing the text to be analysed for ngrams. By default, this is set to 'Keyword' (case sensitive).

-m <maximum-ngram-length>, **--max-n** <maximum-ngram-length>
: The maximum length of ngram phrase required. Each length of phrase below this number will also be returned in increments of one. For example, selecting 3 will return single word frequencies, bigrams, and trigrams. By default, this is set to 5.

-t <number-of-results>, **--top-results** <number-of-results>
: The number of rows of results to return. By default, this is 250 or all of the results if there are fewer than 250.

-w <boolean>, **--stopwords** <boolean>
: Remove stopwords from ngram analysis - true of false. By default, this is set to true.

--version
: Display the version and exit.

--help
: Display a short message and exit.

PYTHON MODULE INDEX

e

INDEX

Symbols

C

D

E

G

I

M

N

R

T