SentAlysis

Sentiment analysis of any text, text file or latest tweets for any Twitter user, hashtag or search term

Analysing text to infer the sentiment behind it can be a useful tool that is used in many industries including politics, finance and medicine to provide valuable insight to make informed business decisions. SentAlysis was designed and built to provide sentiment analysis through Deep AI of text that is input directly by the user, provided in a text file or by searching Twitter for a particular user or search term.

Sentalysis requires a text input to anylse, which can be provided by the user in a few different ways:

- User types text directly into the app.
- User provides a text file to analyse.
- User specifies a username or term to search on Twitter using the Twitter API.

Once the input is provided or retrieved, Sentalysis collates the data and sends it to the Deep AI Semantic Analysis API for analysis. The results are then gathered and displayed to the user.

Jump To Section

App Structure

Back To Top

Twitter Class

- get_tweets method

  Send request to Twitter API to retrieve tweets.

  input: search term or Twitter username provided by user, max results to retrieve

  output: dictionary of the retrieved tweets

- parse_tweets method

  Retrieved tweets are parsed and just the text content is concatenated into a single string which is returned.

  input: dictionary of the tweets retrieved by the Twitter API

  output: The text content of every tweet concatenated into a single string

Analysis Class

- analyse_text method

Send text to Deep AI to analyse and return as a list of strings for each phrase analysed.

input: text provided by user or data received from Twitter API

output: list of strings provided by Deep AI API

- collate_data method

  Take list provided by Deep AI and collate data into total phrases, totals of each response from Deep AI and return dictionary of values.

  input: list provided by analyse_text method

  output: dictionary of collated data

- view_report method

  Using the data which has been retrieved and collated, generate and display a report for the user to see.

  input: dictionary of collated data, boolean 'saved' to determine whether the report is one that's already saved - default False

  output: none, prints report to terminal

- read_file method

  Reads the contents of a given file and returns the data

  input: the path and/or filename of the file to be read

  output: the data in the file as a string, or an empty string on Exception

- save_file method

  Save analysis results to file

  input: the analysis report to be added to the already saved reports and saved

  output: the saved report

- load_saved method

  Load file of previously saved analysis results, or create the file with an empty list if the file doesn't exist

  input: none

  output: none

- view_saved method

  Display a list of the saved reports and display report that is selected by user

  input: none

  output: none

Dependencies

Back To Top

- requests: Communicating with APIs
- python_dotenv: Loading environment variables
- os: Loading environment variables, reading and writing files
- mypy: Static type hinting in Python
- flake8: Style guide enforcement
- json: Work with json data and files

Flowchart

Back To Top