WORD DENSITY ANALYSIS

STEPS TO RUN :

1. Ensure that the package is downloaded and run in the virtual environment to make sure all the modules run as required.

   For vitual env:

   ```
   $pip install virtualenv

   $pip version-----will give you the version installed

   $cd my_project_folder/

   $vitrtualenv my_project

   $source virtualenv/bin/activate
   ```

2. Install the required packages

   ```
   $pip install nltk

   $pip install BeautifulSoup
   ```

   OR

   ```
     $sudo apt-get install BeautifulSoup
     $sudo apt-get install nltk
   ```

3. Then to install other nltk files do

   ```
   $python
   >>>import nltk
   >>>nltk.download()
   Then a window pops up click on download.
   This downloads all the required nltk files.
   ```

4. To run the file :

   ```
   $python main.py [URL]
   ```

   The URL you need to scrape should be passed as a command line argument.
   Eg: `$ python main.py http://wkbn.com/2017/07/14/spokeswoman-jimmy-carter-out-of-hospital-after-rehydration/amp/`

DESCRITION

Running the main.py will follow this flow:

- Fetch the URL
- Download the page
- Extract the parts of the page using BeautifulSoup
  -Title
  -Meta keywords
  -Header
  -Content
  (While getting the content of the page, the method just gets the content excluding the Title, Meta keyword and header. It also recursively traverses the DOM structure to get all the descendants of the current node and also checks the word vs tag density.
- Processing the content and Headers
  -After the headers and content is fetched, they are tokenized using nltk.word_tokenize
  -All the words are then stemmed to their based version using the nltk SnowBallStemmer
  -All the punctuations are removed
  -All the Stop words are removed
  -All the words are then Part of speech tagged using nltk.pos_tag
  -And only the Nouns and Adjectives are fetched from the list as important words because nouns and Adjectives are the words that best describe the core topic better.
- Then for each word in the list frequency of its occurrence in the page is calculated and populated in a dictionary
- This dictionary is then sorted based on frequencies in the reverse order to get the words that are most frequent in the page.
- Get top 20 words from the content and top 10 words from the headers.
- The words in the headers are more important as they are emphasized so they are given 3 times more weight as compared to a word in the content.
- Then the header list and the content list is merged and sorted in reverse order of their frequencies.
- Top 5 words with maximum frequencies are returned as the topics that best describe the contents of the page.

EXAMPLE

URL: https://www.amazon.com/Cuisinart-CPT-122-Compact-2-Slice-Toaster/dp/B009GQ034C/ref=sr_1_1?s=kitchen&ie=UTF8&qid=1431620315&sr=1-1&keywords=toaster

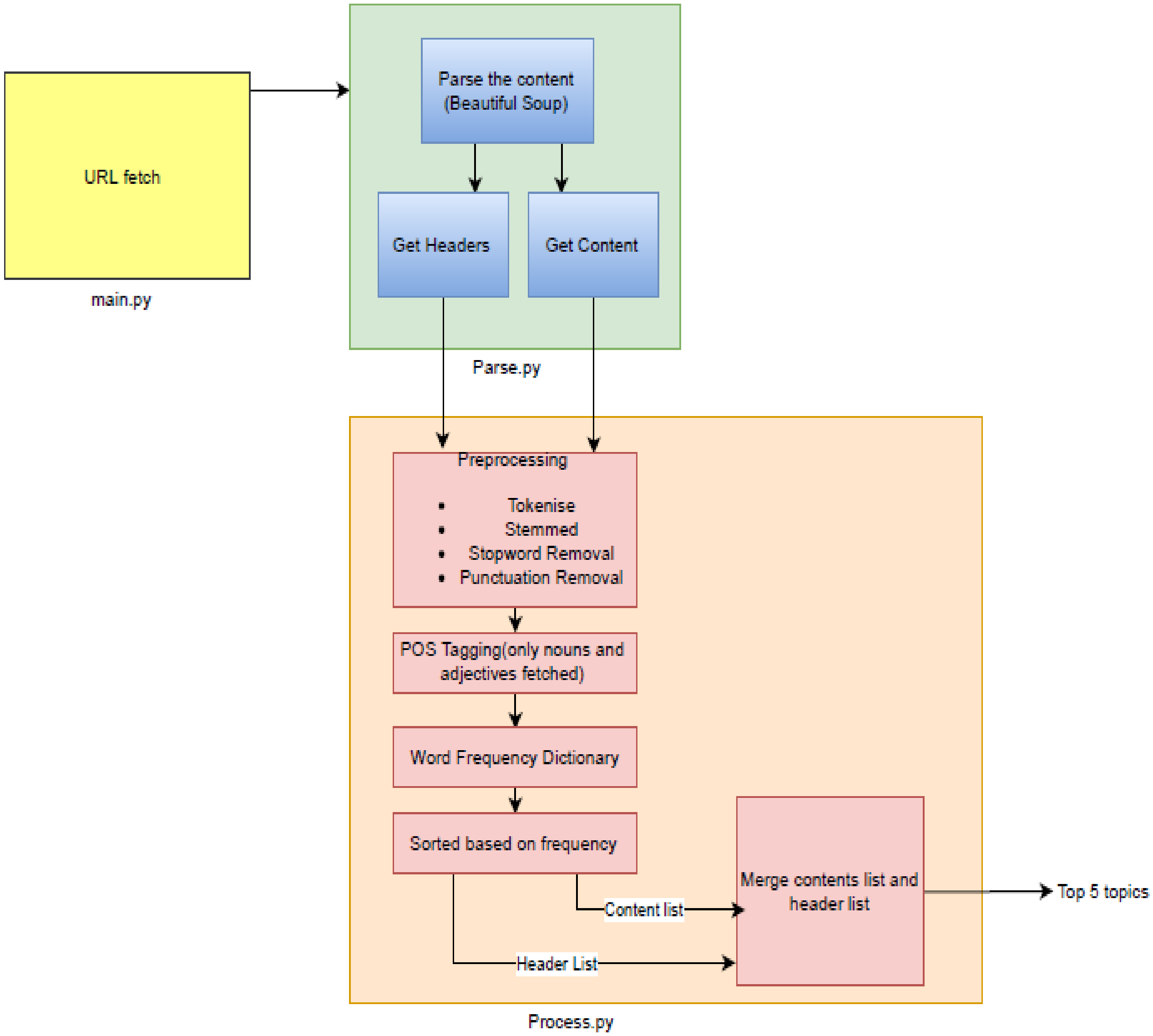
Topics: Toaster, Bread, Toast, Cuisinart, Slice, Stainless, Compact

URL: http://thehill.com/homenews/administration/342976-8-new-trump-moments-that-have-people-talking

Topics: Trump, President, Interview, macron, Insurance, hitler, Investigation

DESIGN

- main.py- Taking the input and call the various functions that process the input and prints the topics that best describe the article
- Parse.py- Parses the contents of the page using BeautifulSoup and gets the heading and content for Process.py
- Process.py- Text Processing is done here-Stop word Removal, Stemming, POS Tagging, Word Frequency Dictionary creation, Sorting and Merging of Headers and Content.


URL fetch
main.py
Parse the content (Beautiful Soup)
Get Headers
Get Content
Parse.py
Preprocessing
• Tokenise
• Stemmed
• Stopword Removal
• Punctuation Removal
POS Tagging(only nouns and adjectives fetched)
Word Frequency Dictionary
Sorted based on frequency
Content list
Header List
Merge contents list and header list
Top 5 topics
Process.py

ERROR HANDLING

1. Ensure that the package is downloaded and run in the virtual environment to make sure all the modules run as required.

   For vitual env:

   ```
   $pip install virtualenv
   $pip version-----will give you the version installed
   $cd my_project_folder/
   $vitrtualenv my_project
   $source virtualenv/bin/activate
   ```

   Then please install all the necessary packages like nltk and beautifulsoup.

2. While connecting to the URL, you may face an error in fetching the content, so this is handled in the script using a try except block.
3. If we are unable to get content from the URL, that is handled by the try except block in Parse.py
4. There may be errors in parsing the content using BeautifulSoup this is also handled in the try except block of Parse.py
5. After parsing if there is no title or content for that page that is handled in the processing.