Ramya Nagapudi
410 Project Documentation

Video Presentation Link:
https://mediaspace.illinois.edu/media/t/1_x2ih4ttb

Code Functionality:

1. Function **take_user_input()** reads the input the user provides, which is the keyword and location of the search query for Yelp.com.

2. Function **scrape_yelp_business(yelp_url)** scrapes the businesses that show up on the first page of Yelp's search result given user input.
    a. This is primarily used for extracting the Yelp URL for each business that Yelp highly recommends given the location and keyword. The yelp url is important to scrape the reviews for each business.

3. For each business gathered from the scraper above, function **scrape_yelp_reviews(restaurant_url)** scrapes the first page of public Yelp reviews.
    a. This is used for getting the best reviews according to Yelp so that we can analyze it in order to make a recommendation.

4. Next, the function **doc_selection(query, restaurantReviews)** keeps relevant reviews given the query = "vegan" and deletes non-relevant reviews (document selection).
    a. This is to remove reviews that don't discuss vegan food sold at the restaurant. I also chose document selection instead of document ranking because I just want all relative reviews and have no need for them to be ranked in terms of relevance.

5. Then the function **get_training_data()** is called to extract and clean a labeled twitter dataset so that it can provide a training dataset to a sentiment analysis model.
    a. This is used for classifying the sentiment of textual data (such as yelp reviews).

6. Function **get_most_positively_reviewed_business(restaurantReviews, model)** uses this model to classify each Yelp review into negative or positive sentiments. Then aggregates the sentiments for each restaurant and returns the restaurant with the most positive reviews according to the model.
    a. This is used for classifying the Yelp reviews to recommend the best vegan restaurant given the user's keyword and location.

Implementation:
Function **take_user_input():**

- Implemented using argparse to read the user input
- Formats this input into a search Yelp URL:
  "https://www.yelp.com/search?find_desc=%s&find_loc=%s" % (keyword,location)

Function **scrape_yelp_business(yelp_url)**:

- Implemented using BeautifulSoup

Ramya Nagapudi
410 Project Documentation

- Get request and beautiful soup set up: *BeautifulSoup(requests.get(yelp_url).content,'lxml')*
- I looked through Yelp's source code to identify the appropriate tags for Beautiful Soup to scrape
    - To extract the business names: *item.find('h4').get_text()*
    - To extract the businesses Yelp URL: *soup.find_all('a', {'name': restaurant_name, 'href': True})*

Function **scrape_yelp_reviews(restaurant_url)**:
- Implemented using BeautifulSoup
- Get request and beautiful soup set up: *BeautifulSoup(requests.get(restaurant_url).content, "html.parser")*
- To get 1st page of a business's reviews: *soup.find_all('span', {'class': "raw__09f24__T4Ezm"})*

Function **doc_selection(query, restaurantReviews)**:
- Set query = "vegan"
- Iterated through restaurantReviews dictionary (key = restaurantName, value = [reviews])
    - If vegan does not exist in a review then it is removed from the list of reviews for that restaurant

Function **get_training_data()**:
- learned how to implement this function using this source: https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk
- Implemented with NLTK (installed it and downloaded its twitter samples data)
- Tokenized the twitter data
- Removed stop words from the twitter data
- Converted tokens into a dictionary (key=words, value=True)
- Split dataset into training and testing data
- Used NaiveBayesClassifier to build the model and trained the training data with its .train() function

Function **get_most_positively_reviewed_business(restaurantReviews, model)**:
- Iterated through restaurantReviews dictionary (key = restaurantName, value = [reviews])
- For each review
    - It's tokenized using NLTK's word_tokenize() function
    - It's removed of stopwords using NLTL's corpus of stopwords
    - And is sent to the model provided from **get_training_data()**
- The model then classifies each review as positive or negative with the NaiveBayesClassifier's .classify() method
- Updates the tracker with the restaurant with the most positive reviews
- Outputs the restaurant name with the most positive review counts

Ramya Nagapudi
410 Project Documentation

Usage:

1. Clone the project with git clone https://github.com/ramyan2/CourseProject.git
2. cd into CourseProject and run python3 yelp_vegan.py <location> <keyword>
    a. The location can be the name of a city, state, a zip code (ex. “Chicago”)
    b. The keyword can be anything vegan related (ex. “Vegan”)
3. The script will output the recommended restaurant.

Evaluation:

The precision and recall for the document selection algorithm for the business = “Althea”

- P = relevant reviews retrieved / relevant reviews retrieved + not relevant reviews retrieved
    - 100% (query successfully removes irrelevant info)
- R = relevant reviews retrieved / relevant reviews retrieved + relevant reviews not retrieved
    - 80% (query tends to slightly overfit)

—

To evaluate the result of my project (the best vegan restaurant) I essentially manually perform the **get_most_positively_reviewed_business()** function.

Given the input location=”Chicago” and keyword=”Vegan”, the following businesses show up on the 1st page of Yelp’s search results at the moment of evaluation. I then read all the vegan related reviews on the 1st page of each business’s yelp site and determined if they were positive or negative. These are the results:

- Althea
    - Positive = 6 | Negative = 2
- Bloom Plant Based Kitchen
    - Positive = 7 | Negative = 0
- Veggie House
    - Positive = 5 | Negative = 0
- Handlebar
    - Positive = 5 | Negative = 1
- Alice & Friends’ Vegan Kitchen
    - Positive = 5 | Negative = 1
- Upton’s Breakroom
    - Positive = 8 | Negative = 1
- Kitchen 17
    - Positive = 5 | Negative = 2
- **Fancy Plants Vegan**
    - **Positive = 9 | Negative = 0**
- IM Vegan
    - Positive = 5 | Negative = 1
- The picky vegan
    - Positive = 2 | Negative = 0
- Loving Heart
    - Positive = 6 | Negative = 0

Ramya Nagapudi
410 Project Documentation

It is important to note that the list of businesses that show up on the 1st page of yelp's search results, given a keyword and location, **change every search**. Hence at the time of evaluation, the best vegan restaurant I manually determined was "Fancy Plants Vegan", which the script accurately evaluated and returned as well. Because the search results change often for the same query, the script's output will change accordingly soi the best vegan restaurant at the time will not always be "Fancy Plants Vegan".