Sentiment Tagging and Keywords Extraction of Hotel Reviews

Songwei Feng (songwei3), Xinrui Ying (xinrui2), Pengyu Chen (pengyu2)

1. Introduction

This project is inspired by `FindLike` hotel search system created by kavita Ganesan. This system allows users to search for hotels by specifying keyword queries such as 'clean', 'safe location', 'good breakfast' to express preferences. Given a set of keywords, the system returns a rank list of all the hotels in the city specified by the user, based on how well the keywords are matched with hotels' reviews. Furthermore, the system also provides information like opinion summaries and buzz words together with each hotel. Thus, this system would help those who are looking for hotels make informed decisions regarding their preferences. The hotel ranking of this system is based on the mining of hotel reviews on TripAdvisor; however, it lacks the sentiment computing and tagging and does not have some useful functionalities such as keywords extraction of hotel reviews.

Thus, we aim to extend the functionalities of FindLike system from two aspects: 1) sentiment polarity tagging, and 2) key words extraction. All these extensions are meant to provide more information about the hotel which users may be interested in and help them make informed decisions.

2. An Overview of Submitted Files

- `pos.csv`, `neg.csv`: a list of positive/negative opinion words.
- `review_spider`: a folder that contains a set of python files for crawling hotel review data from TripAdvisor.com
- `hotel_reviews.csv` a csv file that contains reviews of Chicago hotels crawled from TripAdvisor.com
- `review_analyzer.py`: the main program for sentiment polarity tagging and keywords mining.

3. Details of review_analyzer.py

First of all, we define a vocabulary of opinion words and auxiliary words for keywords extraction and sentiment tagging of reviews (**pos_adj**, etc.) The implementation of sentiment tagging is achieved in the function `sentiment_tag` by two steps. Given a review, we first check the composition of adjectives of it, and if the proportion of negative adjectives takes over a threshold we set, it's then considered a negative review. If it doesn't, then the program will use `SentimentIntensityAnalyzer` from NLTK to make decisions based on the given text. We come up with this idea because during the experiment we found that people who write reviews do not prefer to use negative words, leading the NLTK analyzer prone to output a positive label, even for many negative reviews.

Function `keyword_mining` and `quick_keyword_mining` are what we use to mine keywords from reviews. `keyword_mining` is the major function we use for keywords mining. It takes a string of review and outputs a list of keywords. The key idea of this function is regular expression matching. In practice, the performance of currently available keywords extraction is barely satisfactory, but on the other hand, we noticed that the keywords of hotel reviews do follow some patterns (adj. + noun, etc.) and primarily use a small set of words. Thus, we manually define a vocabulary of informative words and perform regular expression pattern matching directly. `quick_keyword_mining` is a function mostly based on `rake_nltk` to extract keywords not detected by our self-defined dictionary.

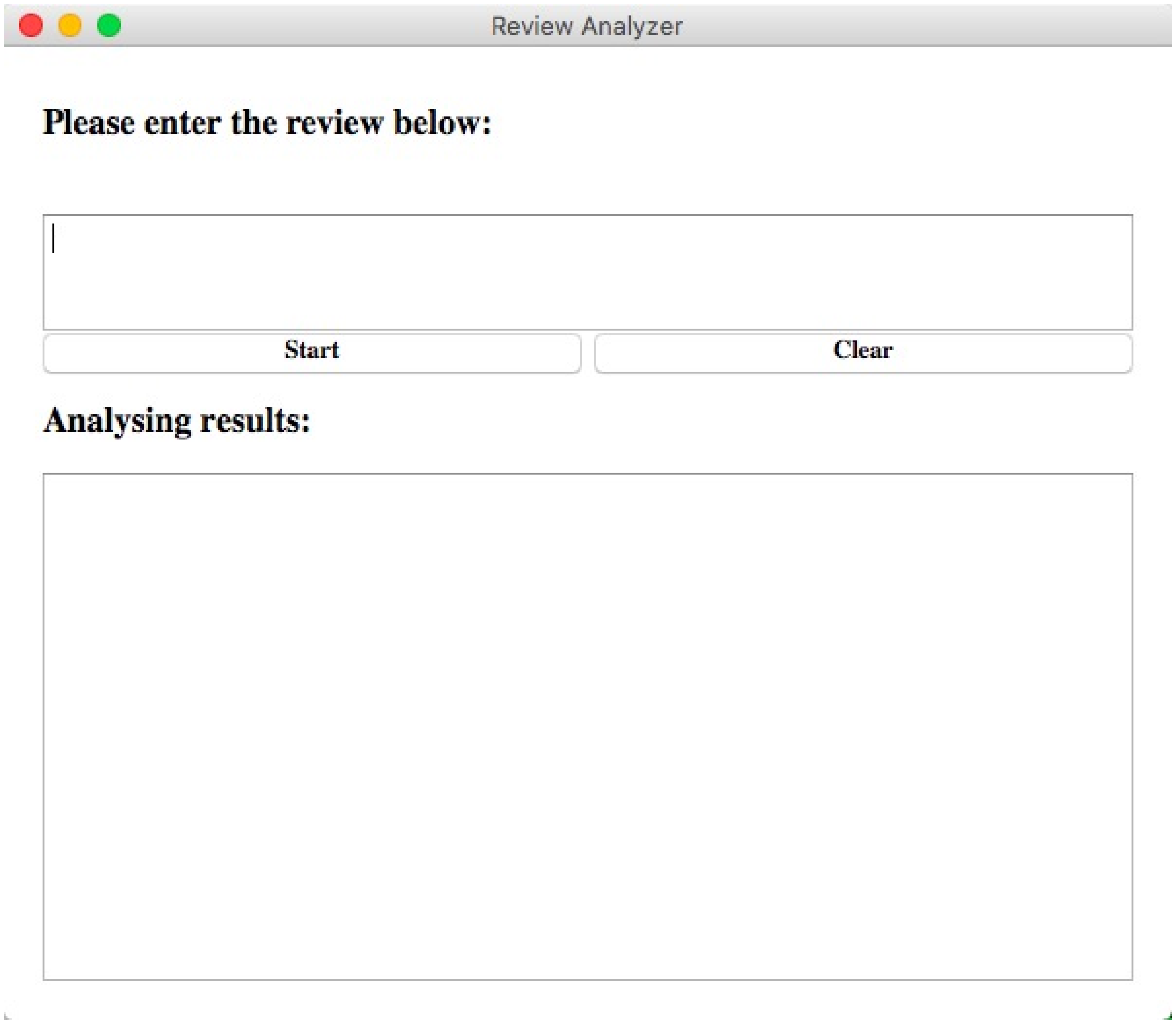
Finally, `initUI` and the followings are functions for user interface implementation.

4. The Usage of the Software

To run the program, please call `python3 review_analyzer.py` on Terminal. Note that it requires `python3` environment and `NLTK`, `rake_nltk`, and `PyQt5` packages; you can install them using pip. Once the program is invoked, a window (Figure 1) will pop up. You should paste the interested review into the text box and click `Start`. Then, the system will start analyzing the given text and output the sentiment polarity and keywords as promised. Note that this process could cost a few minutes, and if you would like to start another analysis, please paste the new text to the box directly and click `Start` again, clicking `Clear` could crash Python. You can find hotel reviews examples in the second column of `hotel_reviews.csv`. An example output is shown in Figure 2.

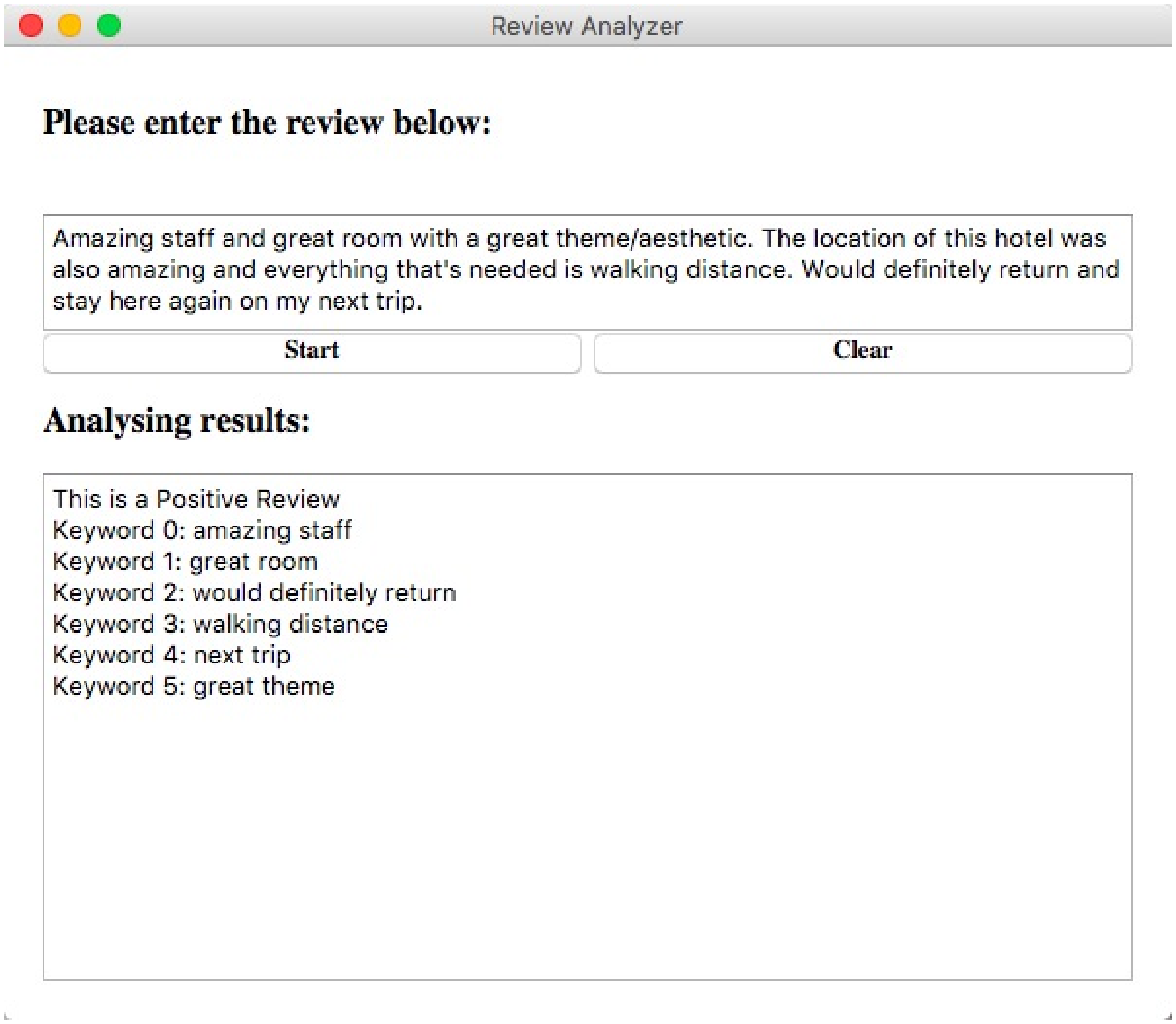
5. Contribution

Every group member has made great efforts in this projects. We met every other week and assigned tasks to do. Xinrui primarily focused on the implementation of sentiment analysis, Pengyu worked on both sentiment tagging and keywords extraction, and Songwei crawled the data, implemented the user interface, and further improved our codes.



Please enter the review below:

Analysing results:



Amazing staff and great room with a great theme/aesthetic. The location of this hotel was also amazing and everything that's needed is walking distance. Would definitely return and stay here again on my next trip.

```
This is a Positive Review
Keyword 0: amazing staff
Keyword 1: great room
Keyword 2: would definitely return
Keyword 3: walking distance
Keyword 4: next trip
Keyword 5: great theme
```