BIG DATA ENGINEER I

By Christan Versteeg, 500859503

Contents

1. Summary/Abstract

This document reports on a sentiment analysis project that evaluates hotel reviews using three different methods: the rule-based VADER and TextBlob models, and a supervised Naive Bayes classifier. We discuss the theoretical underpinnings of these models, describe our data collection and preparation methodology, detail the training process of the Naive Bayes classifier, and analyze its performance. We conclude with a discussion of the results and provide recommendations for future work.

2. Introduction

The document outlines the execution and results of a sentiment analysis task performed on hotel reviews. Sentiment analysis is an invaluable tool in understanding customer sentiment and can inform business decisions and strategies. This analysis is particularly pertinent to the hospitality industry, where customer satisfaction is paramount.

3. Background

The Naive Bayes classifier is a probabilistic model based on Bayes' theorem, which is particularly effective for text classification tasks due to its simplicity and speed. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool optimized for social media text. TextBlob is a Python library that offers a simple API for common natural language processing (NLP) tasks, including sentiment analysis, which it performs using a trained Naive Bayes classifier. (Korab, 2023) (Pius, 2023) (Navlani, 2020)

4. Methods

4.1. Data Collection

The dataset was obtained from the Kaggle repository, consisting of hotel reviews categorized as positive or negative. Additional reviews were scraped from TripAdvisor to augment the dataset.

```python
def setup_kaggle_reviews():
    kaggle_reviews = pd.read_csv("C:/Users/Christan/Desktop/Big-Data-Engineer/Hotel_Reviews.csv")
    kaggle_concat = pd.concat([kaggle_reviews['Positive_Review'].head(50), kaggle_reviews['Negative_Review'].head(50)], axis=0)

    num_reviews = len(kaggle_concat)

    kaggle_reviews = {
            Type: [Type.KAGGLE] * num_reviews,
            'Review': kaggle_concat,
            'TextBlob_Sentiment': [Sentiment.NULL] * num_reviews,
            'VADER_Sentiment': [Sentiment.NULL] * num_reviews,
            'Sklearn_Sentiment': [Sentiment.NULL] * num_reviews,
              }

    return kaggle_reviews
```

```python
def setup_scraped_reviews():
    vpn = Options()
    vpn.add_extension("C:/Users/Christan/Desktop/Big-Data-Engineer/VPN.crx")

    chrome = webdriver.Chrome(options=vpn, service=Service("C:/Users/Christan/Desktop/Big-Data-Engineer/chromedriver.exe"))
    chrome.get('https://www.tripadvisor.com/Hotels-g187147-Paris_Ile_de_France-Hotels.html')

    # 25 seconds of sleep to perform manual actions such as activating the VPN.
    time.sleep(25)

    soup = BeautifulSoup(chrome.page_source, 'html.parser')
    scraped_reviews = soup.find_all(class_='EcHSb')  # EcHSb is the recurring class name of the reviews on TripAdvisor.

    scraped_review_texts = [review.text for review in scraped_reviews]
    num_reviews = len(scraped_review_texts)

    scraped_reviews = {
        Type: [Type.SCRAPED] * num_reviews,
        'Review': scraped_review_texts,
        'TextBlob_Sentiment': [Sentiment.NULL] * num_reviews,
        'VADER_Sentiment': [Sentiment.NULL] * num_reviews,
        'Sklearn_Sentiment': [Sentiment.NULL] * num_reviews
    }

    chrome.quit()

    return scraped_reviews
```

4.2. Data Preparation

Reviews were preprocessed by removing non-textual elements and normalizing the text. (Small sample is taken in this code for increased compilation time, loading in all of the reviews works perfectly fine, it just takes long). All of the data is send to the database.

```
kaggle_concat = pd.concat([kaggle_reviews['Positive_Review'].head(50), kaggle_reviews['Negative_Review'].head(50)], axis=0)
```

<enum 'Type'>	Review	TextBlob_Sentiment	VADER_Sentiment	NB_Predicted_Sentiment
Type.CUSTOM	This hotel was a damn mess. The bed sheets w...	POSITIVE	NEGATIVE	NEGATIVE
Type.CUSTOM	This is your everyday average hotel, not bad, n...	POSITIVE	POSITIVE	NEGATIVE
Type.CUSTOM	Man this hotel was so damn amazing, and it was...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Only the park outside of the hotel was beautiful	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	No real complaints the hotel was great great lo...	POSITIVE	POSITIVE	NEGATIVE
Type.KAGGLE	Location was good and staff were ok It is cute ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Great location in nice surroundings the bar and ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Amazing location and building Romantic setting	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Good restaurant with modern design great chill ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	The room is spacious and bright The hotel is loc...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Good location Set in a lovely park friendly staff ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	No Positive	NEGATIVE	POSITIVE	POSITIVE
Type.KAGGLE	The room was big enough and the bed is good ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Rooms were stunningly decorated and really sp...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Style location rooms	NEUTRAL	NEUTRAL	POSITIVE
Type.KAGGLE	Comfy bed good location	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	This hotel is being renovated with great care a...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	It was very good very historic building that s w...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	This hotel is awesome I took it sincirely becaus...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Great onsite cafe Amazing building Park locatio...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	We loved the location of this hotel The fact tha...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Public areas are lovely and the room was nice b...	POSITIVE	POSITIVE	NEGATIVE
Type.KAGGLE	I liked the hotels history And for such an enorm...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Friendly staff OostPark a few yards away Goo...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	The breakfast was the only positive element of...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	The location is good You need 15min to 20min ...	POSITIVE	POSITIVE	POSITIVE
Type.KAGGLE	Bed was extremely comfy and the staff where ...	POSITIVE	POSITIVE	POSITIVE

4.3. Sentiment Analysis Implementation

VADER and TextBlob were applied to the dataset without further training, leveraging their built-in sentiment analysis capabilities. Python's NLTK library facilitated the use of VADER, while TextBlob was directly applied for sentiment evaluation. The Naive Bayes classifier was trained on the Kaggle data.

5. Results

The Naive Bayes classifier achieved an accuracy of 85-90%, while VADER and TextBlob provided fast and consistent sentiment assessments across the dataset. It general the Naive Bayes classifier was the most accurate, then VADER and afterwards TextBlob.

6. Conclusion and Recommendations

The comparative analysis showed that while the Naive Bayes classifier provided a high accuracy rate, rule-based models like VADER and TextBlob offer rapid sentiment assessment for large datasets and ease of use. For real-time analysis, VADER and TextBlob are recommended due to their simplicity and efficiency.

7. Bibliography

Korab, P. (2023, May 14). *Fine-tuning VADER Classifier with Domain-specific Lexicons*. Opgehaald van Medium: https://pub.towardsai.net/fine-tuning-vader-classifier-with-domain-specific-lexicons-1b23f6882f2

Navlani, A. (2020, September 5). *Naive Bayes Classification using Scikit-learn*. Opgehaald van Medium: https://avinashnavlani.medium.com/naive-bayes-classification-using-scikit-learn-60bc5176f868

Pius, A. (2023, November 22). *Using Python TextBlob for Text Classification*. Opgehaald van Medium: https://medium.com/chat-gpt-now-writes-all-my-articles/using-python-textblob-for-text-classification-7953014f54e6