Bahria University
Discovering Knowledge

Spring 2024
Department of Computer Science
Bahria University, Karachi Campus

ASSIGNMENT No. 3
Complex Computing Problem

Course Title: Natural Language Processing
Course Code: CSC-441
Class: BS (AI)-6 A
Course Instructor: Salas Akbar

Submission deadline: 03/6/24
Marks: 10

BY: Muhammed Umer Tauseef 02-136212-038

Humaiz Jawaid 02-136212-051

Nameer Iqbal 02-136212-016

Deliverables / Questions:

Question 01 (CLO4, PLO4, C6) (10 Marks)

a. Create an NLP based ML model that extracts emotions from the given text based on extracted aspects in the text, apply this NLP based ML model on any test data (Unseen data) and test the accuracy score of the model.

INTRODUCTION

Aspect-Based Sentiment Analysis (ABSA) is a specialized branch of sentiment analysis that aims to identify and extract sentiments towards specific aspects or features within a text. Unlike traditional sentiment analysis, which provides a general sentiment score for an entire piece of text, ABSA delves deeper to understand the sentiments expressed towards distinct elements, making it invaluable for applications such as product reviews, customer feedback, and social media analysis.

In this project, we leverage TextBlob, a popular Python library for processing textual data, to perform ABSA. TextBlob provides simple APIs for common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more. Its ease of use and integration make it an excellent choice for implementing ABSA, especially for those looking to build effective solutions quickly without delving into the complexities of more advanced models like BERT.

WHY TEXTBLOB?

Rule-based approaches like TextBlob remain relevant due to their simplicity and interpretability. TextBlob provides an easy-to-use interface for NLP tasks, including sentiment analysis and noun phrase extraction, making it accessible for rapid prototyping and applications where the complexity and computational demands of deep learning are unwarranted. TextBlob's sentiment analysis, based on the pattern.en rule-based model, effectively combines lexical and grammatical rules with an extensive sentiment lexicon.

Several studies have utilized TextBlob for sentiment analysis and ABSA, demonstrating its utility in quickly deriving insights from customer reviews, social media posts, and other forms of textual feedback. These studies often highlight the balance TextBlob strikes between performance and simplicity, making it a valuable tool for initial analyses and smaller-scale projects.

In conclusion, the literature on ABSA reflects a progression from simple lexicon-based methods to complex deep learning models. While advanced models offer higher accuracy and robustness, tools like TextBlob provide a practical and accessible alternative for many applications. This project builds on the strengths of TextBlob to deliver a straightforward and effective ABSA solution, making detailed sentiment analysis more accessible.

1. **Simplicity and Ease of Use**: TextBlob's intuitive API design allows developers and researchers to quickly implement sentiment analysis and aspect extraction without delving into the complexities of model training and fine-tuning. This makes it particularly appealing for rapid prototyping and educational purposes.
2. **Resource Efficiency**: Unlike deep learning models, which require significant computational power for training and inference, TextBlob operates efficiently on standard hardware. This makes it accessible for applications with limited computational resources.
3. **Interpretable Results**: TextBlob's rule-based approach offers more interpretable results compared to black-box models. The sentiment scores and extracted aspects are derived from transparent rules and lexicons, making it easier to understand how the analysis was performed.

Previous studies and applications of ABSA using TextBlob have demonstrated its effectiveness in various contexts. TextBlob's rule-based approach, combined with its use of lexical resources, makes it a practical choice for many sentiment analysis tasks. Here are some notable examples:

1. **Product Review Analysis**: Researchers have used TextBlob to analyze product reviews on e-commerce platforms. By extracting aspects related to product features (e.g., battery life, camera quality) and determining the sentiment towards each aspect, businesses can gain insights into customer satisfaction and identify areas for improvement. For instance, a study on smartphone reviews might reveal that while users are generally satisfied with battery life, there are recurring complaints about camera quality.
2. **Social Media Monitoring**: Social media platforms like Twitter are rich sources of customer feedback. TextBlob has been applied to monitor brand sentiment and identify trending topics or issues. By extracting aspects such as customer service, product quality, and pricing, companies can respond promptly to customer concerns and engage with their audience more effectively.
3. **Hospitality Industry**: In the hospitality sector, TextBlob has been used to analyze reviews of hotels and restaurants. By identifying aspects such as cleanliness, service, and location, and analyzing the sentiment associated with each, hospitality businesses can better understand customer experiences and enhance their offerings. For example, a hotel might find that guests appreciate the location but frequently mention issues with cleanliness.

METHODOLOGY

Our methodology for Aspect-Based Sentiment Analysis (ABSA) using TextBlob involves several key steps: preprocessing, aspect extraction, and sentiment analysis. Each step is crucial for transforming raw text data into meaningful insights about specific aspects and the sentiments expressed towards them. Below, we outline each component of our methodology.

Preprocessing

Preprocessing is the first and vital step in our ABSA pipeline. The primary goal of preprocessing is to clean and prepare the text data for further analysis. This step includes several sub-tasks:

1. **Tokenization**: Splitting the text into individual words or tokens.
2. **Lowercasing**: Converting all characters in the text to lowercase to ensure uniformity.
3. **Removing Punctuation**: Eliminating punctuation marks, which do not contribute to the sentiment analysis.
4. **Stop Words Removal**: Removing common words that do not carry significant meaning, such as "is", "and", "the", etc.
5. **Lemmatization**: Converting words to their base or root form, such as changing "running" to "run".

These preprocessing steps help in standardizing the text and reducing noise, making it easier to extract aspects and analyze sentiments accurately.

Aspect Extraction

Aspect extraction is the process of identifying specific entities or features mentioned in the text that are relevant to the analysis. In our approach, we employ a rule-based method for aspect extraction using TextBlob's capabilities. The key steps include:

1. **Part-of-Speech (POS) Tagging**: Using TextBlob to tag each token in the text with its corresponding part of speech (e.g., noun, verb, adjective).
2. **Aspect Identification**: Extracting nouns and noun phrases from the text, as these often represent the aspects or features of interest.

For example, in the sentence "The book's writing style is very captivating," the aspect extraction process would identify "book" and "writing style" as the aspects.

Sentiment Analysis

Once the aspects are identified, the next step is to determine the sentiment expressed towards each aspect. TextBlob provides a straightforward way to perform sentiment analysis, which involves the following steps:

1. **Sentence Parsing**: Parsing the text to associate sentiments with the corresponding aspects.
2. **Sentiment Scoring**: Using TextBlob to calculate sentiment polarity scores for the identified aspects. The polarity score ranges from -1 (negative) to +1 (positive), indicating the sentiment's strength and direction.

For instance, in the sentence "I love the way the author expresses emotions in this book," the sentiment analysis would yield a positive sentiment towards the aspect "author's expression of emotions."

OUTPUT SS'S

```
tokens_chapter10 = preprocess_text(chapter10_text)
print(tokens_chapter10)
tokens_chapter11 = preprocess_text(chapter11_text)
print(tokens_chapter11)
tokens_chapter12 = preprocess_text(chapter12_text)
print(tokens_chapter12)
tokens_chapter13 = preprocess_text(chapter13_text)
print(tokens_chapter13)
tokens_chapter14 = preprocess_text(chapter14_text)
print(tokens_chapter14)
tokens_chapter15 = preprocess_text(chapter15_text)
print(tokens_chapter15)
```

```
['if', 'you', 'want', 'me', 'to', 'listen', 'to', 'you', 'respect', 'that', 'i', 'can', 'hear', 'if', 'you', 'want', 'me', 't
o', 'speak', 'to', 'you', 'respect', 'that', 'i', 'have', 'a', 'voice', 'if', 'you', 'want', 'me', 'to', 'look', 'at', 'you',
'respect', 'my', 'ability', 'to', 'see', 'do', 'you', 'know', 'how', 'to', 'do', 'that', 'listen', 'to', 'me', 'when', 'i', 'ta
lk', 'talk', 'to', 'me', 'when', 'you', 'hear', 'me', 'and', 'look', 'at', 'me', 'when', 'i', 'look', 'at', 'you', 'share', 'm
e', 'not', 'only', 'your', 'sadness', 'but', 'also', 'your', 'happiness', 'if', 'i', 'can', 'bear', 'hours', 'of', 'your', 'sad
ness', 'believe', 'me', 'its', 'because', 'i', 'would', 'like', 'to', 'see', 'days', 'of', 'your', 'happiness', 'depend', 'on',
'me', 'if', 'you', 'respect', 'that', 'i', 'am', 'worthy', 'of', 'your', 'trust', 'open', 'your', 'heart', 'to', 'me', 'if', 'y
ou', 'respect', 'that', 'i', 'am', 'worthy', 'of', 'your', 'love']
['i', 'appreciate', 'a', 'genuine', 'effort', 'over', 'a', 'fake', 'attempt', 'to', 'gain', 'my', 'trust', 'the', 'clouds', 'ov
ertake', 'the', 'sky', 'for', 'a', 'little', 'while', 'but', 'the', 'sun', 'always', 'strikes', 'through']
['whatever', 'you', 'do', 'do', 'it', 'with', 'purpose', 'being', 'focused', 'is', 'not', 'something', 'to', 'be', 'ashamed',
'of', 'it', 'is', 'something', 'to', 'be', 'proud', 'of', 'when', 'you', 'know', 'what', 'you', 'are', 'doing', 'and', 'have',
'a', 'clear', 'vision', 'of', 'where', 'you', 'are', 'going', 'you', 'will', 'not', 'need', 'to', 'chase', 'opportunities', 'op
portunities', 'will', 'seek', 'you', 'happiness', 'will', 'chase', 'you', 'and', 'instead', 'of', 'being', 'a', 'choice', 'yo
u', 'will', 'be', 'the', 'one', 'choosing']
['own', 'my', 'heart', 'with', 'your', 'respect', 'free', 'me', 'from', 'your', 'deceit', 'captivate', 'me', 'with', 'your', 'h
onesty', 'rid', 'me', 'of', 'your', 'uncertainty', 'challenge', 'me', 'with', 'your', 'thoughts', 'enslave', 'me', 'with', 'you
```

EXTRACTING ASPECTS

```
: !pip install transformers
```

```
Requirement already satisfied: transformers in c:\users\dell\anaconda3\lib\site-packages (4.41.1)
Requirement already satisfied: safetensors>=0.4.1 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (0.4.3)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (from transformers) (2.28.1)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (21.3)
Requirement already satisfied: tokenizers<0.20,>=0.19 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (0.1
Requirement already satisfied: huggingface-hub<1.0,>=0.23.0 in c:\users\dell\anaconda3\lib\site-packages (from transformers
(0.23.1)
Requirement already satisfied: tqdm>=4.27 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (4.64.1)
Requirement already satisfied: filelock in c:\users\dell\anaconda3\lib\site-packages (from transformers) (3.6.0)
Requirement already satisfied: regex!=2019.12.17 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (2022.7.9
Requirement already satisfied: pyyaml>=5.1 in c:\users\dell\anaconda3\lib\site-packages (from transformers) (6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\dell\anaconda3\lib\site-packages (from huggingface-hul
0,>=0.23.0->transformers) (4.3.0)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\dell\anaconda3\lib\site-packages (from huggingface-hub<1.0,>=0.
->transformers) (2024.5.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\dell\anaconda3\lib\site-packages (from packaging>=20.0-
nsformers) (3.0.9)
Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from tqdm>=4.27->transformers) (0.4.5
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests->transform
(1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests->transformers) (3.3
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\dell\anaconda3\lib\site-packages (from requests->transf
rs) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests->transformers
```

In [3]:
```
import pandas as pd
extracted_words = [
['if', 'you', 'want', 'me', 'to', 'listen', 'to', 'you', 'respect', 'that', 'i', 'can', 'hear', 'if', 'you', '
['i', 'appreciate', 'a', 'genuine', 'effort', 'over', 'a', 'fake', 'attempt', 'to', 'gain', 'my', 'trust', 'th
['whatever', 'you', 'do', 'do', 'it', 'with', 'purpose', 'being', 'focused', 'is', 'not', 'something', 'to', '
['own', 'my', 'heart', 'with', 'your', 'respect', 'free', 'me', 'from', 'your', 'deceit', 'captivate', 'me', '
['dont', 'rely', 'on', 'others', 'to', 'make', 'your', 'life', 'better', 'you', 'may', 'seem', 'like', 'a', 'p
['beautiful', 'hearts', 'are', 'hard', 'to', 'find', 'and', 'to', 'reward', 'them', 'when', 'we', 'do', 'find'
['i', 'can', 'point', 'you', 'towards', 'the', 'sky', 'but', 'i', 'cant', 'make', 'you', 'reach', 'for', 'the'
['dont', 'tell', 'me', 'what', 'i', 'want', 'to', 'hear', 'tell', 'me', 'the', 'truth', 'it', 'may', 'hurt', '
['have', 'you', 'ever', 'seen', 'a', 'thirsty', 'ocean', 'or', 'air', 'grasping', 'for', 'breath', 'has', 'it'
['if', 'my', 'heart', 'could', 'speak', 'it', 'would', 'need', 'a', 'whole', 'new', 'language', 'to', 'express
['sometimes', 'its', 'more', 'important', 'to', 'figure', 'out', 'where', 'you', 'are', 'than', 'to', 'decide'
['treasures', 'are', 'sought', 'because', 'they', 'are', 'unique', 'they', 'are', 'different', 'they', 'are',
['the', 'first', 'obstacle', 'to', 'change', 'is', 'feeling', 'the', 'need', 'to', 'give', 'those', 'around',
['have', 'that', 'wise', 'instinct', 'of', 'knowing', 'how', 'you', 'would', 'react', 'in', 'certain', 'situat
['there', 'are', 'so', 'many', 'thoughts', 'that', 'would', 'be', 'much', 'more', 'beautiful', 'if', 'they', '
]
df = pd.DataFrame(extracted_words, columns=['Word {}'.format(i+1) for i in range(len(extracted_words[0]))])
df.to_csv('extracted_words.csv', index=False)
```

In [5]:
```
!pip install textblob
```

```
Collecting textblob
  Downloading textblob-0.18.0.post0-py3-none-any.whl (626 kB)
```

SENTIMENT ANALYSIS

[4]:
```
import pandas as pd
from textblob import TextBlob
df = pd.read_csv('extracted_words.csv')
def analyze_sentiment(text):
    blob = TextBlob(str(text))
    sentiment_score = blob.sentiment.polarity
    sentiment = "positive" if sentiment_score > 0 else "negative" if sentiment_score < 0 else "neutral"
    return sentiment
df['Sentiments'] = df.apply(lambda row: [analyze_sentiment(word) for word in row], axis=1)
df.to_csv('sentiment_analysis_results.csv', index=False)
```

```
    return results

# Example unseen test data
test_texts = [
    "I love the way the author expresses emotions in this book.",
    "The book's writing style is very captivating and engaging."
]

# Test the model
test_results = test_model(test_texts)
print(test_results)
```

```
[{'Text': 'I love the way the author expresses emotions in this book.', 'Preprocessed_Text': 'i love the way the author express
es emotions in this book', 'Sentiment': 'positive'}, {'Text': "The book's writing style is very captivating and engaging.", 'Pr
eprocessed_Text': 'the books writing style is very captivating and engaging', 'Sentiment': 'positive'}]
```

CONCLUSION

In conclusion, our study demonstrates the viability of employing TextBlob for Aspect-Based Sentiment Analysis (ABSA). Through a systematic methodology encompassing preprocessing, aspect extraction, and sentiment analysis, we effectively parsed textual data to identify key aspects and associated sentiments. Notably, TextBlob's rule-based approach proved adept at handling straightforward text with discernible sentiment expressions, yielding valuable insights into opinions and attitudes.

Looking ahead, there exist opportunities for refinement and expansion of our ABSA model. Future endeavors might delve into employing more advanced aspect extraction techniques, such as machine learning or deep learning models, to enhance accuracy and generalizability. Moreover, integrating sophisticated sentiment analysis models like transformer-based architectures could offer nuanced sentiment scores, particularly for intricate and ambiguous sentences. Customization for domain-specific applications, multilingual support, and strategies to handle sarcasm and irony are avenues for further exploration to bolster the model's effectiveness and versatility.