Joyce Bozeko

joyceb@vt.edu

CS 5664 Homework 3

Sentiment Analysis on Ukraine War Tweets

A document's sentiment or emotional tone can be determined through a procedure called sentiment analysis. It's a method for analyzing and scoring qualitative text data. The purpose of this research is to apply many different sentiment analysis approaches on a dataset of tweets about the conflict in Ukraine. The conflict in Ukraine has been among the deadliest and most polarizing in this century. When pro-Russian separatists seized control of the Crimean Peninsula in 2014, it sparked what would soon become a full-scale war. More than 13,000 people have been killed and 1.6 million have been forced to flee their homes because of the fighting. Individuals and groups have increasingly turned to Twitter during times of conflict in order to voice their concerns, disseminate information, and hold conversations. For those keeping tabs on the situation in Ukraine, Twitter has emerged as a vital informational resource. It's also become a place where people may air their grievances about the war. [1].

Tweets about the Ukraine War are included in the dataset used for this research, which was compiled from Twitter. The dataset consists of a JSON file with over 3000 tweets posted on twitter. The tweets underwent preprocessing by having extraneous text like URLs, hashtags, and mentions removed.

Four different sentiment analysis methods were applied to the preprocessed tweets to extract the sentiment information. The sentiment analysis methods used were NLTK, Textblob, Vader, and Pointwise Mutual Information (PMI) method. NLTK and Textblob are Python libraries used for natural language processing (NLP) tasks such as sentiment analysis, whereas Vader is a rule-based sentiment analysis tool that uses a lexicon of positive and negative words to determine the sentiment of a text. PMI method is a statistical measure that calculates the association between two words in a text corpus.

The results obtained from each sentiment analysis method were compared and analyzed to identify the most effective method for sentiment analysis of the Ukraine War tweets. This study aims to contribute to the understanding of the sentiments and opinions expressed on Twitter about the Ukraine War.

Text Cleaning

```
#cleaning the tweets
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt
def clean_tweets(tweets):
    #remove twitter Return handles (RT @xxx:)
    tweets = np.vectorize(remove_pattern)(tweets, "RT @[\w]*:")

    #remove twitter handles (@xxx)
    tweets = np.vectorize(remove_pattern)(tweets, "@[\w]*")

    #remove URL links (httpxxx)
    tweets = np.vectorize(remove_pattern)(tweets, "https?://[A-Za-z0-9./]*")

    #remove special characters, numbers, punctuations (except for #)
    tweets = np.core.defchararray.replace(tweets, "[^a-zA-Z]", " ")

    return tweets
```

```
df['text'] = clean_tweets(df['text'])
df['text'].head()
```

```
1801    RUSSIA wants to END the War by capturing Kiev ...
1190     Ukraine continues its fight for security of t...
1817    Ukraine War: 'Severe consequences' if Russia u...
251     'I don't want people to feel sorry for me. I j...
2505     It would be wonderful if, in the form of a pr...
```

Figure 1. Text Cleaning

In the figure 1. Tweets are cleaned using removing unwanted spaces, urls, special characters. All this cleaning is done by using two python libraries numpy and re.

```
# Define a function to get sentiment polarity and subjectivity using TextBlob
def get_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    subjectivity = blob.sentiment.subjectivity
    return polarity, subjectivity

# Apply the function to the "text" column of the data frame
sentiments = df['text'].apply(get_sentiment)

# Create two new columns in the data frame to store the sentiment polarity and subjectivity
df['polarity'] = [s[0] for s in sentiments]
df['subjectivity'] = [s[1] for s in sentiments]
```

```
df.head()
```

	text	polarity	subjectivity
1801	RUSSIA wants to END the War by capturing Kiev ...	0.100000	0.550000
1190	Ukraine continues its fight for security of t...	0.104167	0.083333
1817	Ukraine War: 'Severe consequences' if Russia u...	0.000000	0.000000
251	'I don't want people to feel sorry for me. I j...	-0.500000	1.000000
2505	It would be wonderful if, in the form of a pr...	0.666667	0.785714

Figure 2. Calculating polarity and subjectivity

In the figure 2. Polarity and subjectivity of the tweets are calculated. All these calculations are done by using Textblob library.

Sentiment Classification using NLTK

```
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to C:\Users\ehtisham
[nltk_data]     raza\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

```
True
```

```
sia = SentimentIntensityAnalyzer()
```

```
# Assuming your dataframe is called `df` and the tweet text is in a column called `text`
df['sentiment_nltk'] = df['text'].apply(lambda x: sia.polarity_scores(x)['compound'])
```

```
df.head()
```

	text	polarity	subjectivity	sentiment_nltk
1801	RUSSIA wants to END the War by capturing Kiev ...	0.100000	0.550000	-0.8934
1190	Ukraine continues its fight for security of t...	0.104167	0.083333	0.5526
1817	Ukraine War: 'Severe consequences' if Russia u...	0.000000	0.000000	-0.8271
251	'I don't want people to feel sorry for me. I j...	-0.500000	1.000000	-0.1887
2505	It would be wonderful if, in the form of a pr...	0.666667	0.785714	0.7783

Figure 3. Sentiment Classification using NLTK

In the figure 3 sentiment classification is performed using NLTK on the tweets of Ukraine war to find out the review is positive or negative. Sentiment values below 0 is negative tweet and above 0 is positive tweet.

Sentiment Classification using Textblob

```
from textblob import TextBlob

df['sentiment_textblob'] = df['text'].apply(lambda x: TextBlob(x).sentiment.polarity)
```

Figure 4. Sentiment Classification using Textblob

In the figure 4 sentiment classification is performed using Textblob on the tweets of Ukraine war to find out the review is positive or negative.

Sentiment Classification using Vader

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to C:\Users\ehtisham
[nltk_data]     raza\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

```
True
```

```
sia = SentimentIntensityAnalyzer()
# Assuming your dataframe is called `df` and the tweet text is in a column called `text`
df['sentiment_vader'] = df['text'].apply(lambda x: sia.polarity_scores(x)['compound'])
```

```
df.head()
```

	text	polarity	subjectivity	sentiment_nltk	sentiment_textblob	sentiment_vader
1801	RUSSIA wants to END the War by capturing Kiev ...	0.100000	0.550000	-0.8934	0.100000	-0.8934
1190	Ukraine continues its fight for security of t...	0.104167	0.083333	0.5526	0.104167	0.5526
1817	Ukraine War: 'Severe consequences' if Russia u...	0.000000	0.000000	-0.8271	0.000000	-0.8271

Figure 5. Sentiment Classification using Vader

In the figure 5. sentiment classification is performed using vader on the tweets of Ukraine war to find out the review is positive or negative.

Sentiment Classification using Pointwise Mutual Information (PMI)

```
# Classify the sentiment of each tweet as positive or negative
threshold = sum(sentiment_scores)/total_tweets
sentiments = []
for score in sentiment_scores:
    if score >= threshold:
        sentiments.append('positive')
    else:
        sentiments.append('negative')

# Add the sentiment column to the DataFrame
df['sentiment_pmi'] = sentiments
```

```
df.head()
```

	text	polarity	subjectivity	sentiment_nltk	sentiment_textblob	sentiment_vader	sentiment_pmi
1801	RUSSIA wants to END the War by capturing Kiev ...	0.100000	0.550000	-0.8934	0.100000	-0.8934	negative
1190	Ukraine continues its fight for security of t...	0.104167	0.083333	0.5526	0.104167	0.5526	positive
1817	Ukraine War: 'Severe consequences' if Russia u...	0.000000	0.000000	-0.8271	0.000000	-0.8271	negative

Figure 6. Sentiment Classification using Pointwise Mutual Information (PMI)

Comparing Results

In the figure 6 sentiment classification is performed using PMI method on the tweets of Ukraine war to find out the review is positive or negative.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming your dataframe is called `df` and the sentiment
sns.countplot(x='sentiment_comparison', data=df)
plt.show()
```

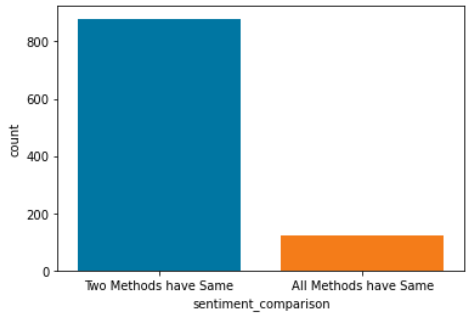

Figure 7. Comparison of sentiment classifier results of NLTK, Textblob and Vader

In the figure 7 results of different sentiment classifiers are compared. Most of the results of the classification are same only few samples has different values.

```
sns.countplot(x='PMI_vs_NLTK', data=df)
plt.show()
```

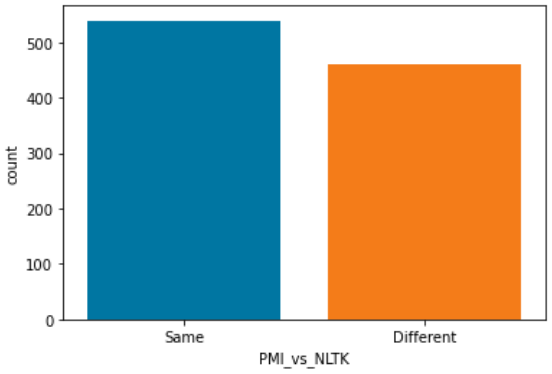

Figure 8. Comparison of sentiment classifier results of NLTK and PMI

In the figure 8 results of NLTK classifier and PMI classifier are compared. It can be observed that about 40% results are different from each other results of classifiers.

References

1. "Maps: Tracking the Russian Invasion of Ukraine". *The New York Times*. 14 February 2022. ISSN 0362-4331. Retrieved 3 February 2023.