6_TextAnalytics

November 8, 2024

1 Analyzing the book Dracula published by Project Gutenberg

Project Gutenberg can be found here: www.gutenberg.org

We used two different methods of sentiment analysis:

* The vader model
* Affection score calculated with NLTK opinion lexicon

As seen in the resulting graph, the affection line has much highher amplitude and variance compared to vader. Using a longer rolling average causes the values of the vader model to shift entirely onto the positive side, while the very negatives of the affection score are kept. Both of these methods seem to have various advantages and disadvantages, but if the output of the affection score had been calculted as a fraction instead, the scales of the two method may have been more comparable.

The up and down results on both of the models seem to mostly match. There are some exceptions, for example the drop in the lexicon model around the line 6000. Vader does not react to those lines almost at all.

[5]:
```
import nltk
from nltk import tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from urllib.request import urlopen
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from nltk.corpus import opinion_lexicon
import string
```

[9]:
```
# These packages need to be run only once
# nltk.download('vader_lexicon')
# nltk.download('opinion_lexicon')
```

[13]:
```
text = urlopen("https://www.gutenberg.org/cache/epub/345/pg345.txt").read().
 ↪decode('utf-8')

# Cleaning the text
text = text.lower()
text = text.translate(str.maketrans('','',string.punctuation))
text = text.splitlines()
```

```
# Filter non-empty lines
lines = []
for line in text:
    strippedLine = line.strip()
    if strippedLine != '':
        lines.append(strippedLine)

# Selecting the important lines
# Start at the title DRACULA and end at THE END
lines = lines [66:-329]

# Testing the lines
print("Beginning:\n")
for line in lines[:5]:
    print(line)

print("\n")

print("End:\n")
for line in lines[-3:]:
    print(line)
```

```
Beginning:

dracula
chapter i
jonathan harker's journal
kept in shorthand
3 may bistritzleft munich at 835 p m on 1st may arriving at


End:

know what a brave and gallant woman his mother is already he knows her
sweetness and loving care later on he will understand how some men so
loved her that they did dare much for her sake"
```

[15]:
```
# Creating the sentiment analysis using both of the models
sid = SentimentIntensityAnalyzer()

scores_vader = []
affections = []

for line in lines:
    # vader
    scores_vader.append(sid.polarity_scores(line))
```

```
    #lexicon
    affection = 0
    for word in line.split(' '):
        if word in opinion_lexicon.positive():
            affection += 1
        elif word in opinion_lexicon.negative():
            affection -= 1
    affections.append({'affection': affection})

scores_vader = pd.DataFrame(scores_vader)
affections = pd.DataFrame(affections)

print(len(scores_vader))
```

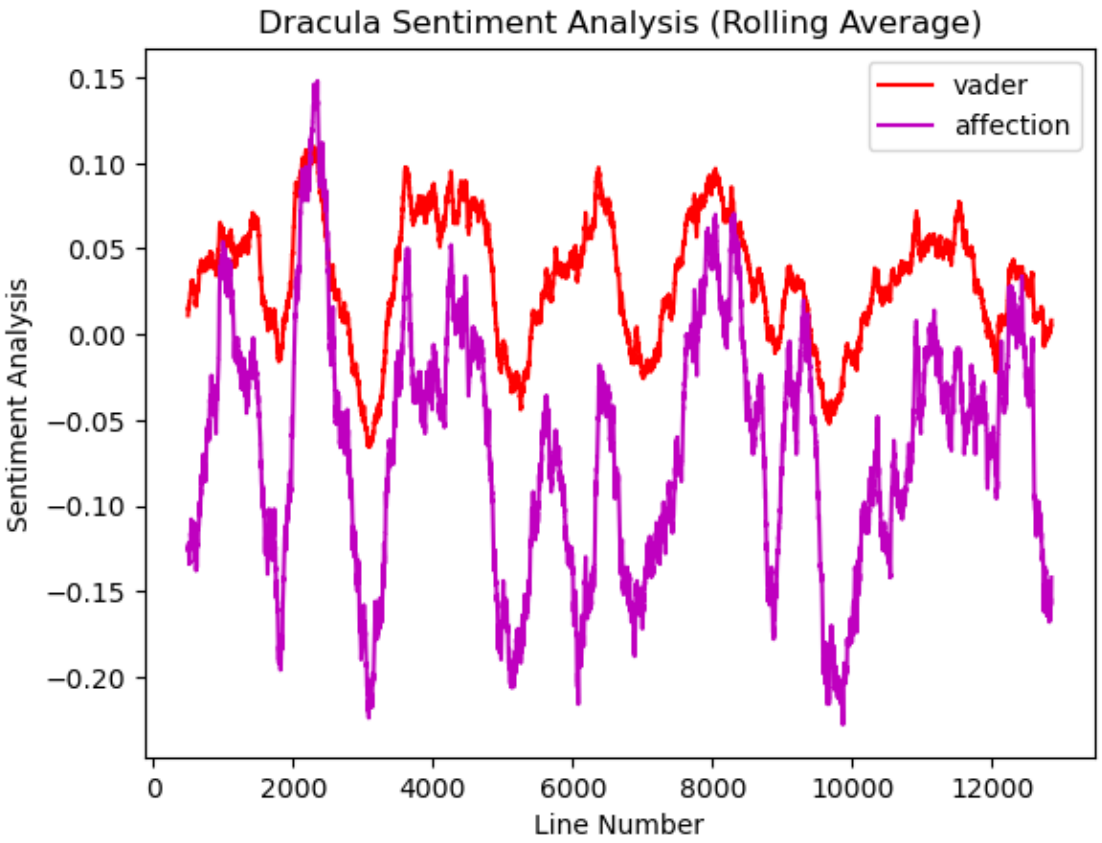
```
12848
```

[17]:
```
# Calculating rolling averages
rolling_avg_vader = scores_vader.compound.rolling(500).mean()
rolling_avg_affection = affections.affection.rolling(500).mean()

# Creating a display graph for the data
plt.plot(rolling_avg_vader, 'r-', label="vader")
plt.plot(rolling_avg_affection, 'm-', label="affection")

plt.title("Dracula Sentiment Analysis (Rolling Average)")
plt.xlabel("Line Number")
plt.ylabel("Sentiment Analysis")
plt.legend()

plt.show()
```


Dracula Sentiment Analysis (Rolling Average)
vader
affection
Sentiment Analysis
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
2000
4000
6000
8000
10000
12000
Line Number