**IST-736 Text Mining**

**Homework-1**

Sharat Sripada (vssripad@syr.edu)

### Introduction

This week’s homework is playing the role of a consultant at a public relations firm who is a task to evaluate the sentiment toward ‘AI’ (Artificial Intelligence) on social media platforms like Facebook or Twitter.

Further, since scraping through data on such platforms can yield significant scale, we are required to use automated means (via sdk, API etc.) to fetch data and evaluate tools for sentiment analysis. Having known a few libraries, we will focus our analysis in comparing the following specific methods or tools:

* NLTK’s built-in sentiment analysis tool
* TextBlob – Simple Python library that offers API to NLP related tasks like sentiment analysis

The data source would be twitter, and we would specifically collect a sample of tweets from Twitter related to topic or string *#ai/#AI*.

### Data source and collection

Data or tweets would be gathered from Tweeter using the Python tweepy package/library. The code shows having to use consumer\_key, consumer\_secret, an access\_token and access\_token\_secret to setup an OAuthHandler and api handle (row 13).

Further, since we are merely looking to compare the variants and possibly study their functional behavior we will work on a sample of the dataset. We are able to limit (via tweepy api knobs) to 100 most recent tweets at the time/date of writing this document.

The auth parameters are setup as below:

Graphical user interface, text, application

Description automatically generated

Using filters like search string, date, and number of tweets to retrieve, a sample of tweets tagged with word *#ai* is stored in a python list *tweets = []*:

Graphical user interface, text

Description automatically generated

Examining the raw data, we see tweets comprise of several redundant stop-words and other metadata like URLs, emojis etc. which may not necessarily have an impact on sentiment.

Trimming for brevity would likely save us runtime but this was not considered.

### Evaluation

This section will explore and bring out differences between the sentiment analysis techniques outlined at the outset. The process will broadly encompass iterating over each tweet (remember we have 100 tweets in total), running sentiment analysis, translating scores correspondingly to categorical values – Positive, Negative or Neutral sentiment and store all this in a *Pandas* data-frame so we can easily do some visualizations.

#### Method-1 - NLTK’s built-in sentiment analysis tool

Using the NLTK library and the code below we iterate each tweet applying the *SentimentIntensityAnalyzer()* function and extract polarity scores as seen on row-12:

Graphical user interface, text, application, email

Description automatically generated

**Fig:** Code showing the NLTK’s sentiment functionality

The resulting data-frame is shown here:

Table

Description automatically generated

**Fig:** Data-frame comprising Tweet and sentiment

Finally using the matplotlib pyplot and sns libraries we will group by sentiment and visualize the data through a bar-plot:

Chart, bar chart

Description automatically generated

**Fig:** Bar-plot showing sentiment distribution for all tweets

**NOTE:**

A common function was written to make this plot:

import matplotlib.pyplot as plt

import seaborn as sns

def plot\_bar\_graph(title, label\_groups):

plt.figure(figsize=(8,5))

ax = sns.barplot(x="Sentiment", y="counts", data=label\_groups)

ax.set\_xticklabels(ax.get\_xticklabels(), rotation=0)

ax.set\_title(label="%s" %title)

plt.show()

#### Method-2 – TextBlob based sentiment analysis tool

A similar method was followed using the TextBlob python API for NLP analytics. A score is extracted per tweet as shown on row-7 and then translated to fit in data-frame:

Graphical user interface, text, application, email

Description automatically generated

**Fig:** Code showing the TextBlob sentiment functionality

A screenshot of a computer

Description automatically generated with medium confidence

**Fig:** Data-frame comprising Tweet and sentiment

Correspondingly, grouping by sentiment and calling the *plot\_bar\_graph()* function the following bar-plot

Chart, bar chart

Description automatically generated

**Fig:** Bar-plot showing sentiment distribution for all tweets

### Conclusion

Comparing the results of the two methods for sentiment analysis based on the bar-chart visualization, we can summarize as follows:

|  |  |  |
| --- | --- | --- |
| **Sentiment/Method** | NLTK | Textblob |
| Negative | <10 | <10 |
| Neutral | 30 | 40-50 |
| Positive | >60 | 45-50 |

Notice that the NLTK library classified >60% of the tweets as positive while 30% as neutral. Meanwhile Textblob library has Neutral and Positive sentiment for 40-50% tweets.

Run-times were comparable but may need more data at higher scale.

Further, it would be interesting to study the NLP related algorithms at core of the process prior choosing one method over the other.