Week 13 Notebook: Analyzing Sentiment

Name: Muhammad Rakibul Islam

Date:

Class:

What is Sentiment Analysis?

- opinion mining
- The author's attitude toward the subject (tone).
- Measuring emotion -->
    - how much?
    - directed toward?
    - negation?

Opinion / emotion are equated for the purpose of measuring in sentiment analysis. There is work by linguists on how to measure sarcasm, refining negation measures, etc.

3 Elements of Sentiment

Opinion / Emotion

Polarity

```
- positive (+)
- negative (-)
- sometimes though not often_ neutral (=)
```

Multi-class

```
- joy
- surprise
- anger
- love
```

Quantitative

```
- Likert scale
- rating
- numerical grading, number of likes, endorsements, etc.
```

Subject

```
- What is discussed?
    - Book, movie, song, product, service, teacher...
- Can be mixed
    -  ie. I like the main character of the novel, but the plot was slow and the dialogue was
- In other words, the level of granularity at which the sentiment is leveled becomes part of
```

Opinion Holder or Entity

```
- Who holds the opinion?
- What do we know about the opinion holder?
- How much does it matter?
```

Sentiment analysis does not work well with null values.

How is Sentiment Analysis used

Social listening is a job category in which people are paid to use data such as Amazon reviews, Twitter hashtags, Rate My Professor reviews, etc, and identify: what is discussed (including granularity), how is it being discussed (opinion equated to sentiment / emotion), and by whom is it being discussed. Sentiment analysis is performed on single sentences, single words, collections of words, social media datasets, but also blog posts, online forums, and the news.

It is meant to *enrich* an assessment of a brand and opinions held about it. The same tools have also been used for social science research, and literary studies. For example, Matthew Jocker's analysis of *forms of the novel* using the Syuzhet Package (an R package described here: https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html).

Question for reflection: How does "sentiment analysis" work in a rhetorical analysis?

Consider how we approach questions of "sentiment" (arguably, tone) in a close reading "rhetorial analysis" and what the slippages might be between that form of close reading analysis of sentiment and the "operationalized" reading that measures language's sentimental or emotional weight.

RESPONSE:

In a close reading rhetorical analysis, we can approach questions of or classify or detect sentiment from various insights like the emotion conneceted to a word (whether is is positive, negative, happy, sad, anger etc.), usage of imagery, metaphors and other figure of speech that can project a certain emotion etc. Also, it is important to understand that this is being by a human reader who's understanding of language far exceeds any machine as the reader themselves can feel the tones and emotions being used in the text.

On the other hand "operationalized" or "computational" sentiment analysis first of all we lose subjectivity itself. Because algorithms are quite literally just numbers and calculation adding weights and proabibilities to text, the human sense is lost.

Also machines, while better at perform calculations and faster, can very easily miss things like irony or metaphors or sarcasm and take such text purely in their rinsed out literary form.

Close reading is also highly adaptable because as a human reader I can for example understand the difference between how language will be used in a work of objective journalism vs satirical articles while a machine will be needed to be specifically trained with two different models altogether to be used on such a case.

However, at the same time, it is important to mention that when it will come to large amounts of texts that might not be possible to analyse through close reading, operationalized or computational reading can be handy, although it still needs training, evaluation and model improvement for the best outcome.

Question for reflection: Consider what it means to bring a metric and method designed for consumer research into the humanities? What are the potential advantages or disadvantages of doing so?

You might consider as a starting off point that many scholars, especially Marxist scholars, of text have pointed to the tensions between market value of texts (consumer interest, purchasing, annd production) and the *literary tastes* of "high fiction." Or, one might consider the tensions between mass market production of books and reader reception theory in conversation with the marketing and reception or "literariness" of small press literary production.

RESPONSE:

First of all I think what we need to understand is that bringing a metric and method designed for consumer research into the humanities should be taken as an additional set of tool that can assist not as a replacement for techniques that already exist within the humanities.

Most of the methods designed for consumer research consists of advanced data-driven quantitative tools for analysis which can complement the existing qualitative methodologies in the humanities to provide new insights and perspectives, particularly in the case of literary production and consumption.

Also it can help produce interdisciplinary conversations combining the humanities with quantitative economics as well as psychology (consumer behavior) to expand existing conversations.

However, drawing from Marxists theories, without critically examining the models being used there is the danger of capitalizing on and commodifying literature which can lead to diminishing values in the quality of literature from a humanities perspective. Certain literary work which are vital but not seen as a high commercial value can be neglected and abandoned even when it adds important literary value.

Additionally, drawing on ideas from the previous answer, all machines are doing is quantifying things in a manner that can enable it to perform calculations which can end up losing context as well as the subjectivity of what is being studied.

Our weekly readings, particularly the paper by DA, associates with the questions asked. However, I felt that DA was taking on a more critiquing role to the use of technology in computation literary studies which although is important to understand, we should equally highlight the benefits. What we also have to make a special note of is that advance computational methods that are currently being used are still a fairly new technology and while there is a lot to work on, using feedback to iteratively improve our models can make them more useful in addition to already existing technqiues in the humanities.

```
import nltk
import pandas as pd
import numpy as np
import sklearn
```

Please download the following dataset to complete this assignment: IMDB_sample.cvs. Then use the following code snippet to upload the file to your Google Colab working space.

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
# import dataset (a sample from IMDB)
movies = 'IMDB_sample.csv'
```

```
# convert the IMDB comma separated values file into a pandas dataframe
movies_df = pd.read_csv(movies)
```

```
# Let's look at how the dataframe is organized by displaying the first 5 entries.

movies_df.head()
```

	Unnamed: 0	review	label
0	18245	This short spoof can be found on Elite's Mille...	0
1	19801	A singularly unfunny musical comedy that artif...	0
2	3222	An excellent series, masterfully acted and dir...	1
3	6010	The master of movie spectacle Cecil B. De Mill...	1
4	16355	I was gifted with this movie as it had such a ...	0

Ok, so when we're looking at this dataframe, what do we see? Each row has a unique identifier, which begins with 0. There is also an unnamed label. This number is probably either the number of the review or the number of characters, but we don't know b/c it's not labeled. Then, the text of the review can be found in the "review" column. Finally, the "label" colum holds a series of 0s and 1s. These 0s and 1s represent *polarity* data. That is to say that it is a binary distinction between "positive" (1) and "negative" (0) reviews. The word "label" is confusing, because "reviews" is, technically also a label. Try not to get hung up there. But for now, the "label" label is where we know that the "sentiment" measure is held.

```
# What if we wanted to know how many "positive" and "negative" reviews are in the data
# We'd call the name of the dataframe, we'd name the column that we want to perform an
# action we'd like to perform. Here, value_counts will count the total number of each
# It will also tell you what "data type" you are counting. If you want to eventually c
# you would want an even distribution. So, this is a layer of exploratory work before

movies_df.label.value_counts()
```

```
0    3782
1    3719
Name: label, dtype: int64
```

```
# So, you know the numbers... but maybe it's hard to tell whether or not the two grou
# Another way to look at your data would be to turn those label values into proportion
# a mathematical equation. This says, take the label column from the movies dataframe
# unique label. Then, divide those numbers by the total number of rows in the movies_

movies_df.label.value_counts()/len(movies_df)
```

```
0    0.504199
1    0.495801
Name: label, dtype: float64
```

```
# Look back at the dataframe. What is the degree of granularity at which each review i
# Do you have any thoughts about whether or not the length of a review relates to the
# be found in side? Well, one thing you might want to do is to figure out how long the
# We do this by cfreating a pandas series. We're taking the reviews column from the mo
# the data type as a string (str), and then we are calculating the length of each.

length_reviews = movies_df.review.str.len()
```

```
# The output when we do this is called a "series."
type(length_reviews)
```

```
pandas.core.series.Series
```

```
# We can use the "max" function to search the dataframe and then to find how many cha

# THAT IS A VERY LONG REVIEW!!
max(length_reviews)
```

```
10321
```

```
# Same thing for the shortest review.
min(length_reviews)
```

```
52
```

If we believe that the length of the review is a valuable feature that we want to continue working with, we could append this information to the dataframe. If you want to continue learning outside of class, this is a good challenge problem to work on. See if you can take this new data about each review and add it as an additional "feature" in your dataset.

Exploring Data in Detail

Levels of granularity

Measuring "sentiment" this way is highly dependent on how closely you want to look at the text.

- Document level?
- Sentence level?
- Aspect level? (relating sentiment to a direct referent, even within a single statement)

Types of Sentiment Analysis

In general, there are two kinds of sentiment analysis.

Rule or Lexicon-based

In this approach, algorithms match the words in the lexicon to the dataset and eitheer sums the whole or averages them, depending on what function you choose. The result is a combinatory values. In other words, the reviews above are measured in 1/0 because they are a "net positive" or "net negative"--and the algorithm assignes the value based on where the total text ends up on a scale.

- List of words
- Balance score --> nice: +2, good: +1, miserable: -4, happy: +3
- Relies on a hand-crafted set of valence scores as dictionaries / lexicons.
- Fails at some tasks because different words have different valences in different contexts.
- - Polarity of words may change with the topic
- - These changes can't be reflected easily in the dictionary
- - Can work fast and is less computationally resource intensive

Example:

Today, was, a, good, day. 0, 0, 0, 1, 0 --> 1 - positive

Automatic / Machine Learning

- Modeled as a classification problem
- Using a dataset with "known sentiment" we need to predict the sentence of a dataset with unknown sentiment.
- relies on labeled historical data
- is resource intensive (uses lots of a computer's reources to train models)
- can be "powerful" (ie. goes fast and changes flexibly, depending on how it is deployed)

Unsupported Cell Type. Double-Click to inspect/edit the content.

```
from textblob import TextBlob
text = "Today was a good day."
```

```
my_valence = TextBlob(text)
my_valence.sentiment
```

```
    Sentiment(polarity=0.7, subjectivity=0.6000000000000001)
```

THe sentiment function in TextBlob returns a tuple (that means data in pairs). Sentiment has 2 components, a *polarity* value, and a *subjectivity* value. Polarity is measured on a scale of -1 (negative) to 1 (positive) with 0 as a neutral value. Subjectivity, however, is measured in a range from 0 to 1. Measures the calculated degree to which a value may be accurately assessed at the assigned polarity value.

```
# The datatype becomes a specific thing, a textblob object that has NLP processing pe
# Sentiment is one part of the processing that we're just calling from the new textbl
type(my_valence)
```

```
    textblob.blob.TextBlob
```

```
twocities = "It was the best of times, it was the worst of times,it was the age of wi
```

```
citiesblob = TextBlob(twocities)
```

```
citiesblob.sentiment
```

```
    Sentiment(polarity=0.022916666666666658, subjectivity=0.5895833333333332)
```

```
# Create a text string. Turn it into a TextBlob object. Call out the sentiment measur
tomcabin = "Late in the afternoon of a chilly day in February, two gentlemen were sit
tomblob = TextBlob(tomcabin)
tomblob.sentiment
```

```
    Sentiment(polarity=-0.04444444444444443, subjectivity=0.4305555555555555)
```

Questions?

Tutorials and resources for future study:

```
* https://pythonspot.com/python-sentiment-analysis/
* https://github.com/nltk/nltk/wiki/Sentiment-Analysis
```