Building a Twitter Scraper and a Prototype Dictionary-based Sentiment Analyzer

Prof: Dr. Egbert
Class: ENG 678
Presenter: Rui Hu (Sherman)

In this presentation I will :

1. Introduce the topic
2. Mention the research questions
3. Talk about the methods
4. Address data and analysis
5. Point out challenges and limitations
6. Conclude the presentation

Introduction

- Sentiment analysis
  - "study of people's opinions, sentiments, emotion, and attitudes." (Liu, 2020)
- Two types of sentiment analysis
  - Dictionary-based sentiment analysis
  - Machine-learning-based sentiment analysis
- Machine-learning-based model make texts lose their linguistic underpinnings
- The purpose of the project was to create a prototype dictionary-based sentiment analyzer and investigate the texts from the perspective of corpus linguistics...

Research questions

RQ1:

What is the distribution of the ratings of the collected Tweets produced by dictionary-based and pre-trained machine-learning-based sentiment analyzer? - *I will investigate the distribution of ratings*

RQ2:

To what extent do the above-mentioned ratings correlate with each other? - *I will investigate the correlation between the ratings*

Methods

3 Python programs (~530 lines of code):

twitter_scraper.py & sentiment_analyzer.py &

main.py (provides friendly interface and customization options)

2 Ratings (from -1 to 1):

Produced by *sentiment_analyzer.py*

Sentiment: subjectivity and polarity. I will focus on polarity:

positive (rating = 1), negative (-1), neutral (0)

1 Correlation:

Between two ratings

Methods - Twitter Scraper

Table 1

Two main functions of the Twitter Scraper

Python Module: Tweepy

Function Name	Argument(s)	Feature
get_tweets()	*user screen name, tweet count*	connect to Twitter, get the Tweets, pre-process the Tweets, and write the results to a CSV file
clean_tweets()	*user screen name*	perform cleaning of the Tweets and save results as a CSV file to prepare for sentiment analysis

- In this project:
- I used 3200 Tweets collected from @abc (ABC News)

Methods - Sentiment Analyzer

Both analyzers produce ratings ranging from -1 to 1 (both inclusive)

Table 2

Three main functions of the Sentiment Analyzer

Function Name	Argument(s)	Feature
analyze_sentiment_pretrained()	*user screen name*	analyze the sentiment using the pre-trained machine learning model from a Python model called *TextBlob*
analyze_sentiment_dict_based()	*user screen name*	analyze the sentiment using the dictionary-based analyzer
sentiment_model_correlation()	*user screen name*	write out two ratings datasets to a new CSV file; create a new csv file to store the correlation data

opinion lexicons (positive words and negative words, totaling 6800 words) - also cited in ref: http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

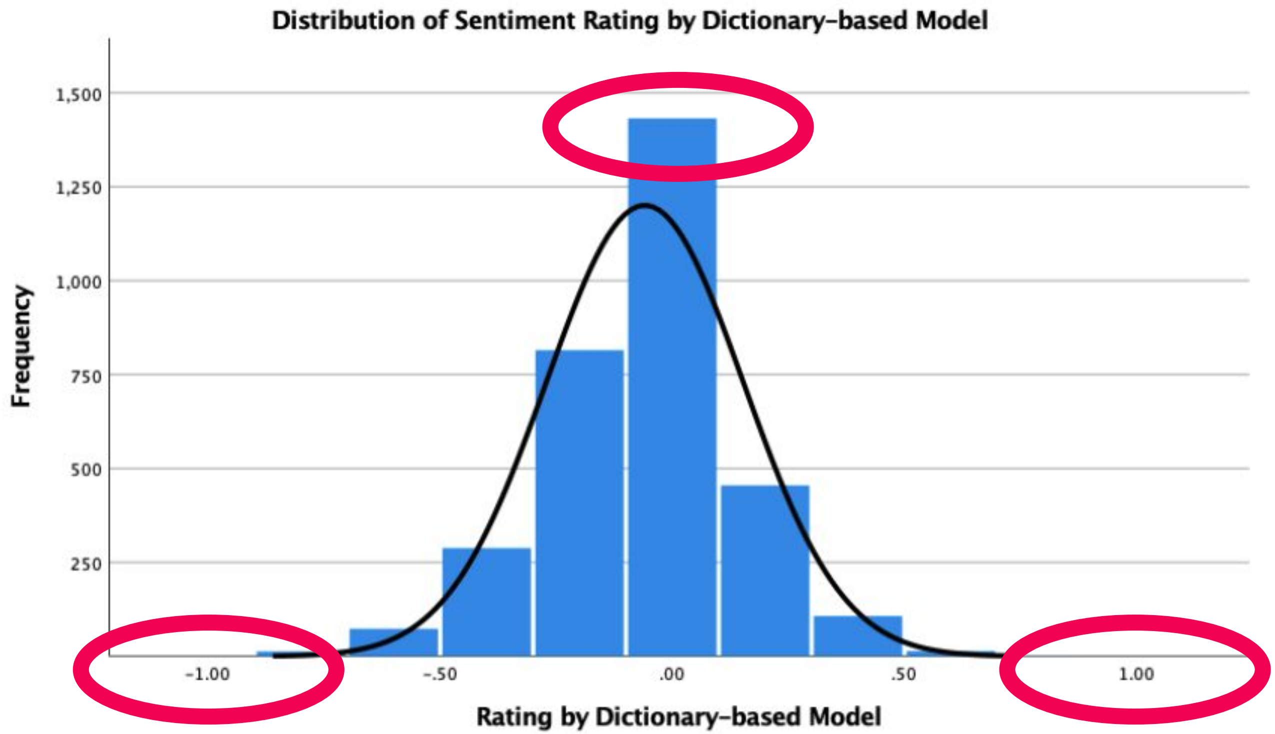
Data and Analysis

Table 3

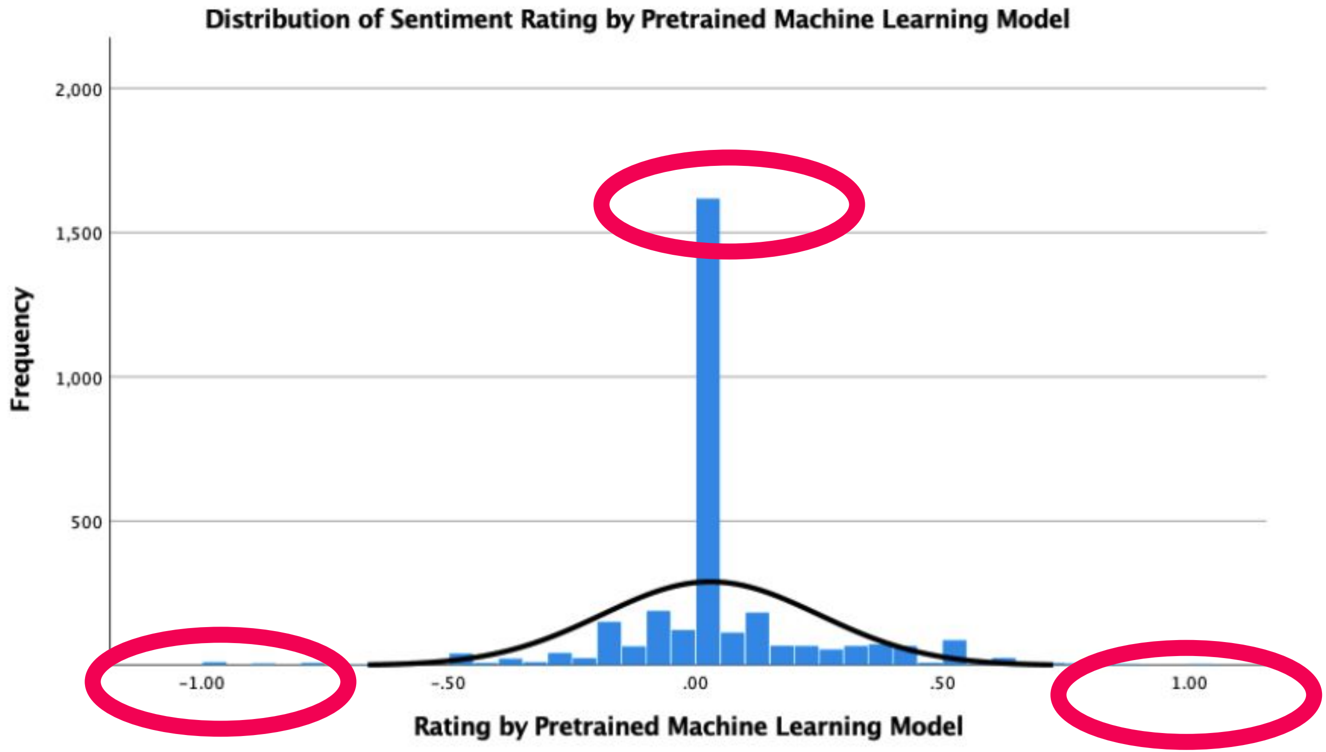
Summary of the Tweets collected by the Twitter Scraper (sorted by month)

	March	April	Total
Count of Tweets	611	2,589	3,200
Favorite Count	179,509	846,867	1,026,376
Average Favorite Count	294	327	321
Retweet Count	46,553	247,417	293,970
Average Retweet Count	76	96	92
Text Length	82,349	342,900	425,249
Average Text Length	135	132	133

Data and Analysis - RQ1



Data and Analysis - RQ1



Data and Analysis - RQ1

Table 5

Frequency table of neutral sentiment rating (rating = 0) (N = 3200)

	Dictionary-based Model (*rating_db*)	Pre-trained Machine Learning Model (*rating_pt*)
Count of neutral sentiment	1,432	1,542
Portion of neutral sentiment	44.8%	48.2%
Total tweets	3200	3200

- it is safe to rudimentarily summarize that the dictionary-based sentiment analyzer has reasonable accuracy in distinguishing **neutral sentiment**

Data and Analysis - RQ2 - Produced by Python

Positive correlation

Table 6

Correlation matrix of dictionary-based sentiment rating and pre-trained machine-learning-based sentiment rating (N = 3200)

⁉️

	r	r^2	adjusted r^2	95% CI	p-value *
Pearson's Correlations	.357	.128	.127	[.33, .39]	7.462×10^{-97}

Note: CI stands for confidence interval; * p-value shows statistical significance, $p < .001$

Challenges and limitations

1. Scraping the Twitter corpus; max Tweet limit (200 -> 3200 by using cursor and pagination)
2. More factors should be considered:
   a. Negation words (negative influence on rating)
   b. Punctuations
   c. Emoticons
   d. Emojis
3. Other News Twitter accounts (Fox, CNN, CBC, CNBC) and other registers (personal Tweets by famous people)

Conclusion (brief)

Built a Twitter scraper and a prototype dictionary-based sentiment analyzer; applied the model pre-trained machine-learning-based sentiment analyzer

I would not declare the dictionary-based sentiment analyzer flawless -> $r = .357$

Shed some light on future corpus linguistics research and sentiment dictionary building

“You can do hard things!”

💪💪💪💪

Dr. Jesse Egbert

Thank you! ☺

Any questions?

Python Modules (8+ main modules)

1) tweepy - Twitter scraper

2) pandas (pd) - enhanced dataframe

3) os (operating system) - for folder and file

4) datetime - current date and time

5) textblob (TextBlob) - pretrained machine learning sentiment analyzer

6) numpy (np) - enhanced scientific calculation

7) sklearn - 'MaxAbsScaler' from 'sklearn.preprocessing' - used to scale/normalize the data

8) pingouin (pg) - contains many statistic models (in this program we use this to count Pearson's r)

….

References

Biber, D., Egbert, J., & Davies, M. (2015). Exploring the composition of the searchable web: A corpus-based taxonomy of web registers. Corpora, 10(1), 11-45. https://doi.org/10.3366/cor.2015.0065

Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2013). *Combining strengths, emotions and polarities for boosting Twitter sentiment analysis*. Paper presented at the Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, Chicago, Illinois. https://doi.org/10.1145/2502069.2502071

Everitt, B., & Skrondal, A. (2010). *The Cambridge dictionary of statistics*. Cambridge, UK: Cambridge University Press.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hu, M., & Liu, B. (2004). *Mining and summarizing customer reviews*. Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA. https://doi.org/10.1145/1014052.1014073

IBM Corp. (2020). *IBM SPSS Statistics* (Version 27.0) [Computer software]. Retrieved from https://www.ibm.com/products/spss-statistics

Liu, B., & Cambridge University Press. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions (Second Edition)*. New York: Cambridge University Press.

Loria, S. (2018). TextBlob Documentation. *Release 0.15, 2*.

O'Brien, S. F., & Yi, Q. L. (2016). How do I interpret a confidence interval?. *Transfusion*, *56*(7), 1680–1683.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Roesslein, J. (2020). Tweepy: Twitter for Python. https://github.com/tweepy/tweepy.

The Pandas Development Team. (2020, February). *pandas-dev/pandas: Pandas*. https://doi.org/10.5281/zenodo.3509134

Vallat, R. (2018). Pingouin: statistics in Python. *Journal of Open Source Software*, 3(31), 1026, https://doi.org/10.21105/joss.01026

Supplementary Slides

(on Demand)

Rating of my prototype sentiment analyzer

Sentiment rating = Positive score + Negative score

(future improvement: by adding more factors/variables)

one positive word equals 1 point;

one negative word equals -1 point.

Range: infinite $\rightarrow$ [-1, 1]

Normalize: Maximum Absolute Value Method (Pedregosa et al., 2011)

$$Normalized\ sentiment\ rating = \frac{Sentiment\ rating}{Absolute\ value\ of\ the\ maximum\ in\ the\ column}$$

Table 4
Descriptive statistics of the ratings produced by sentiment analyzers (N = 3200)

	Sentiment Analyzer Model	
Statistics	**Dictionary-based Model (*rating_db*)**	**Pre-trained Machine Learning Model (*rating_pt*)**
N	3200	3200
Mean	-.059	.030
Median	.000	.000
Mode	.000	.000
Min	-1.000	-1.000
Max	.800	1.000
Midpoint	-.100	.000
Range	1.800	2.000
Std. Deviation	.213	.220
Variance	.045	.048
Skewness	-.242	.139
Kurtosis	.933	4.551