Aspect-Based Sentiment Analysis with Python and spaCy

William Strausser, Data Analyst, Arts ISIT

THE UNIVERSITY OF BRITISH COLUMBIA
Faculty of Arts

Agenda

1. Overview of sentiment analysis / aspect-based sentiment analysis
2. Python demo of spaCy, VADER, and LA NLP
3. Tableau demo of possible end use-case

Sentiment analysis

- As defined by AWS, “sentiment analysis is the process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral.”
- Analysis usually consists of passing text to a function which returns a “polarity” score between -1 and 1 (representing negativity and positivity, respectively).
- For example:

  “I love pizza.” = 0.6369

  “I like pizza.” = 0.3612

  “I don’t like pizza.” = -0.2755
- Many different approaches can be used to conduct sentiment analysis, from simple ‘bag-of-words’ models utilizing lexical data compiled by hand, to sophisticated machine learning models trained on vast quantities of data.

Selecting a sentiment analysis model

- For best performance, sentiment analysis models can be custom-trained from the data they will be used on. However, this process is quite time-consuming and requires advanced expertise.
- Alternatively, there are several open source sentiment analysis models available requiring no custom training or other setup.
- Our criteria for selecting a model:
  1. Usable via a Python API
  2. Fast, lightweight, and easy to install/use
  3. Produces demonstrably meaningful data

Candidate models

TextBlob – Python package for performing a number of NLP tasks, including sentiment analysis.

Pattern – Another Python package with a variety of NLP and non-NLP tools, including sentiment analysis.

Valence Aware Dictionary and sEntiment Reasoner (VADER) – Long-standing pure sentiment analysis Python package specifically attuned to social media text data.

*Several other options exist on the market, but most rely on machine learning techniques and can be difficult to install/run based on hardware compatibility.

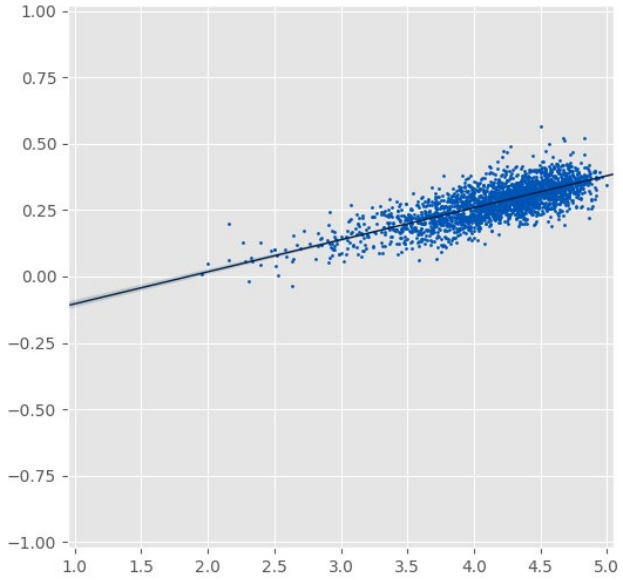
Model performance

x: Mean response to UMI 6 (“Overall, I learned a great deal from this instructor.”)

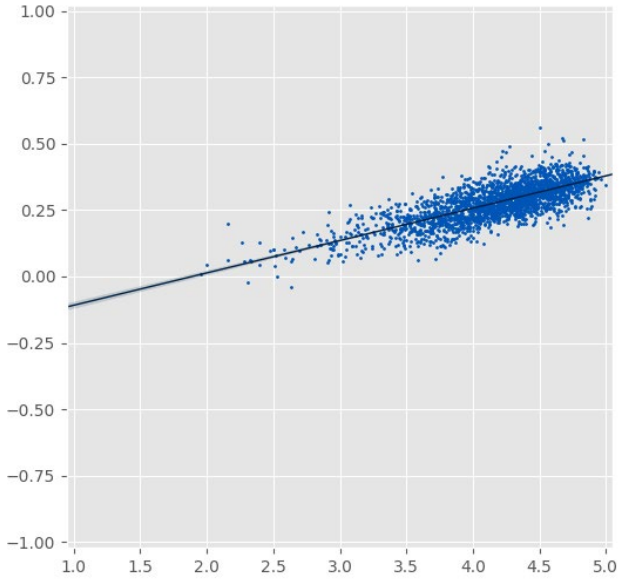
y: Mean sentiment score to open-ended question “Please comment on any aspects, positive or negative, of your instructor's teaching, attitudes to students, class atmosphere, or any other matters affecting the quality of instruction that you consider worthy of note.”

Each point represents one course taught in the Faculty of Arts from 2010-2022, excluding courses with fewer than 25 responses to either question.

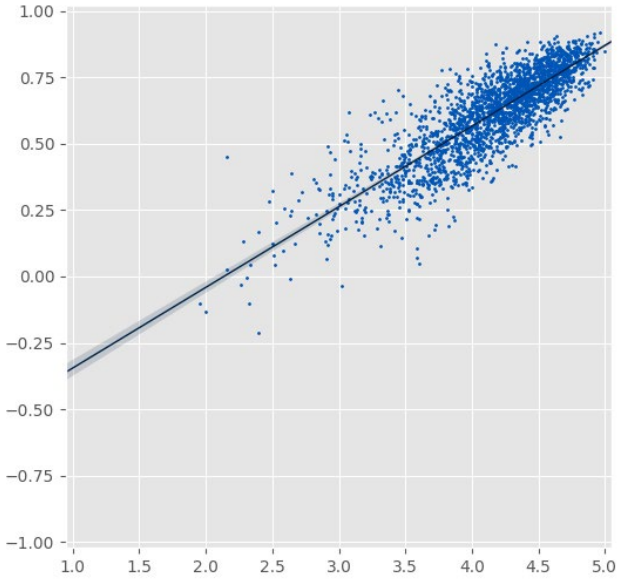
TextBlob



Pattern

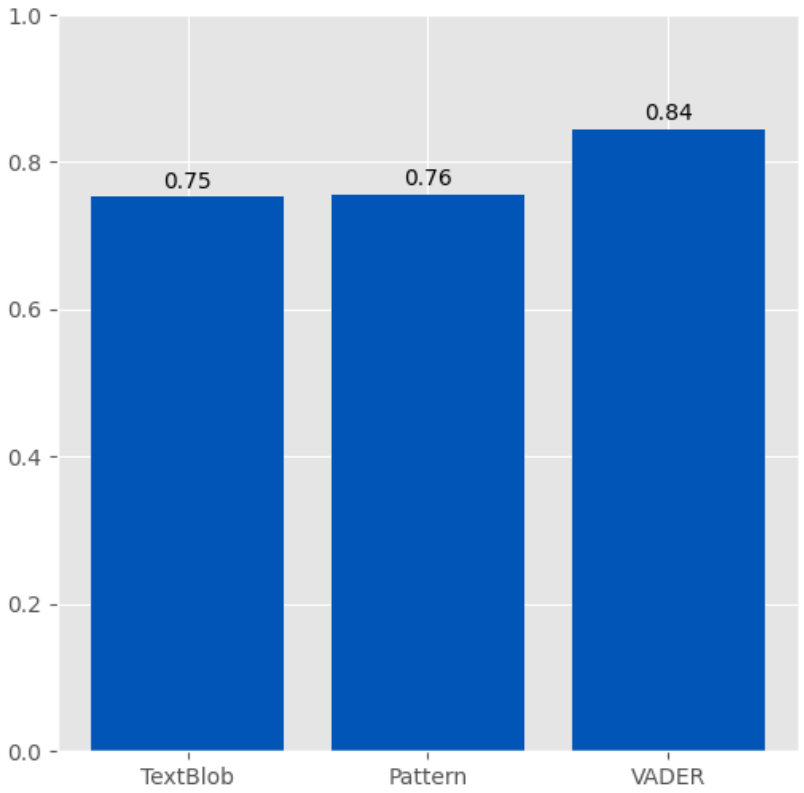


VADER



Model performance (cont.)

Calculating correlation coefficients of the previous slide's data for each model yielded the following results:



Shortcomings of basic sentiment analysis

- Semantic ambiguity: rule-based approaches, like the models introduced on the previous slides, often have difficulty with words that have multiple meanings.
  - E.g. the sentence “that person is mean” is evaluated by VADER as having a neutral sentiment score of 0, while this is clearly inaccurate to human eyes.
- Complex sentiments: texts containing complex or multiple sentiments cannot be evaluated accurately.
  - E.g. the sentence “the food was good, but the service was bad” is clearly expressing two separate sentiments, but VADER evaluates this as having a purely negative sentiment of -0.5859.
  - The rest of the presentation will focus on addressing this problem.

Aspect-based sentiment analysis (ABSA)

- Aspect-based sentiment analysis is a technique which aims to address the issue of complex sentiments mentioned in the previous slide.
- This technique separates a text into distinct ‘aspects’ towards which sentiments are expressed, then assigns scores to each aspect, rather than the entire text.

The food was great, but the service was bad.

Conventional sentiment analysis: -0.5859

Aspect-based sentiment analysis:

- Food: 0.4404
- Service: -0.5423

Existing options for ABSA

- Current open source projects for conducting ABSA mostly rely on machine learning libraries like TensorFlow and PyTorch.
- Machine learning introduces additional complexity in the form of potential hardware incompatibility, greater computational expense, and opaque calculation of results.
- In lieu of using existing ML-based software, we opted to build our own rule-based system using spaCy.

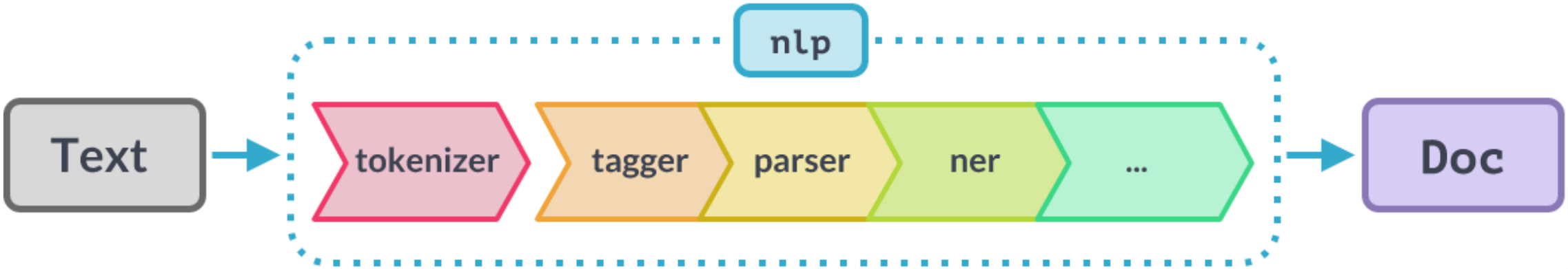
What is spaCy?

“spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python.

spaCy is designed specifically for production use and helps you build applications that process and ‘understand’ large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning.”

See spaCy’s website for a more in-depth description of spaCy and how it compares to other NLP libraries like NLTK.

spaCy's Processing Pipeline



Demo of spaCy and LA NLP

https://github.com/Arts-ISIT-LA/lava-2023-03/blob/main/lava_demo.ipynb