An Analysis on LingPipe and NLTK Toolkit on Sentiment Analysis

The purpose of this text review is to analyze two popular toolkits for text sentiment analysis: LingPipe and NLTK. A brief introduction to its capabilities and a descriptive explanation of implementing the toolkit will be provided.

Introduction

One of the most popular open-source tools for classifying text and analyzing sentiment is LingPipe and NLTK. A classifier is an algorithm for assigning categories to text, for example, assigning the language to a sentence or determining whether a Twitter tweet has a positive, negative, or neutral sentiment. LingPipe is a Java toolkit for NLP-oriented applications that was created by Breck Baldwin & Bob Carpenter who own Alias-i, Inc. In 2003, Alias-i, Inc. announced the release of LingPipe 1.0, its suite of linguistic tools, for research and commercial use. Similarly, NLTK is a suite of libraries and programs written in Python for symbolic and statistical natural language processing (NLP). This program was created by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania and is a leading Python library for Natural Language Processing today. An analysis on LingPipe and NLTK can help users determine which toolkit best meets their needs for sentiment classification.

What is Sentiment Analysis?

Sentiment analysis or opinion mining is a text mining technique in machine learning and natural language processing (NLP) to decipher and analyze sentiment from text to derive insights such as determining whether an article or author contains or holds a positive, negative, or even neutral perspective. Analysis of sentiment from text data can provide additional insights because it contains subjective and rich opinions not available in other forms of data. Using these insights, researchers will be able to gain a more comprehensive understanding of users' behavior, preferences, and opinions, which will prove extremely valuable for the development of decision-making and business intelligence, which will enable organizations to make more informed decisions. Governments are also implementing sentiment analysis to achieve greater transparency, to engage citizens, and even to determine the response of citizens to the ongoing fight against COVID-19. A view of sentiment allows governments and policymakers to identify widespread societal and epidemiological issues before they spiral out of control.

What are the Challenges with Identifying Sentiment?

Companies depend on sentiment analysis to gain a deeper understanding of the consumer mindset. Although sentiment analysis offers significant benefits, there are challenges, such as the inability of computer algorithms to determine sentiment and which objects within a text are the subjects of which sentiments simply because computers lack emotions. Consider, for example, the word 'nice', which conveys a positive sentiment when applied to a particular product. However, a sarcastic comment can also be construed from this statement. The machine must understand context and intention to distinguish between the two. For a computer, the context of a subject matters because computers are incapable of learning context unless something is explicitly stated. In contrast, humans can recognize the underlying meaning of a statement.

Toolkits Available for Sentiment Analysis

For a machine to perform such an analysis, it would need to be equipped with machine learning techniques such as text analytics, natural language processing, and named entity recognition. A viable sentiment analysis algorithm requires identifying and capturing basic opinion representations, as noted by Professor ChengXiang Zhai: the opinion holder, opinion target, and opinion context (this includes enriched opinions such as sentiment). Next, sentiment classification is required to understand opinions, which may require breaking down comments, paragraphs, or documents into smaller fragments (Zhai). Customer feedback, for example, often holds multiple ideas or opinions, so analyzing the overall sentiment of reviews, tweets, documents, and so on, may result in a variety of sentiments. Sentiment classification can be accomplished by leveraging toolkits for processing text using computational linguistics. The two most used toolkits for sentiment analysis will be examined in this study: LingPipe and NLKT. This paper aims to compare the toolkits for movie reviews to assist users in making an informed decision regarding which toolkit is most appropriate for their needs. In addition, it provides an unbiased comparison of the performance of each toolkit.

LingPipe

LingPipe is a free and open-source toolkit implemented in Java. It is designed for processing text data using computational algorithms to perform tasks such as identifying names, organizations, and locations of things. It is mostly used for classifying Twitter search results into categories and optimizing spelling queries. The LingPipe language classification framework can also perform two type of classification tasks: it can separate subjective and objective statements, as well as differentiate positive and negative statements.

LingPipe's site provides an example of conducting sentiment classification using movie reviews. The dataset contains reviews, and labels were created with an improved rating-extraction system by using a pre-annotated classification of the reviews as boolean (positive and negative) and scalar data representation (objective and subjective sentences). In this example, a logistic regression classifier is applied to build the LingPipe program to classify sentiment. The program reads the training directory location, trains a classifier on the training data, then evaluates the classifier on the test data. The resulting classifier can judge whether a whole film review is positive or negative (as defined by the data set curators). The code begins by setting the number of tests and number of correct answers counters to zero. Then, as each review is processed, the number of tests is incremented. Then the classifier is used to produce a classification for a review string on a single line. Following that, the most appropriate category is extracted from the classification as the result category. If the result category matches the test category, the number of correct classifications is incremented. A second form of sentiment analysis, namely determining if a sentence is "objective" or "subjective" (again, as defined by the database curators). It follows much the same pattern as the last example, with a slightly different data format: the addition of the classifier evaluation framework and a step to compile the model to a file for later use. The advantage of the evaluation framework is that it cannot only tell right from wrong, but also distinguish several shades of grey. This extracts the subjective sentences and returns them as a string. The example creates a classification using the filtered input. Finally, the example add a case to the evaluator. This illustrates how the evaluator may be used without an embedded classifier -- cases are just added in terms of the first-best answer and the response classification.

Natural Language Toolkit (NLTK)

LingPipe offers a way for extracting deeper insights from text documents and many other methods of developing a classifier to provide a variety of sentiment analysis. Another comparable toolkit is the Natural Language Toolkit (NLTK) which is a toolkit built for working with NLP in Python. The NLTK

library offers several tools for manipulating and analyzing linguistic data. Text classifiers are among its advanced characteristics, which can be used for various classifications, including sentiment analysis. NLTK provides a package called “SentimentIntensityAnalyzer” from “nltk.sentiment.vader,” where VADER (Palaparthi) stands for Valence Aware Dictionary and Sentiment Reasoner. It is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments. VADER uses a combination of a sentiment lexicon, a list of lexical features (i.e., words) which are labelled according to their semantic orientation as either positive or negative. In addition to providing information about the positivity and negativity scores, VADER also makes it possible to determine levels of positive or negative sentiment.

Using movie reviews, a supervised learning model was developed using a training data set to train the model. The first steps can involve tokenizing the data, normalizing, and cleaning the data or removing nulls and stop words (i.e., words such as, the, but, etc.). In this example, the sentiment analyzer was developed using the movie review corpus with a random forest algorithm which is a meta-estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy (the same as the concept of bagging) and control over-fitting. The results showed movies classified between positive, negative, or neutral sentiment based on scores derived from reviews.

Conclusion

LingPipe and NLTK are immensely powerful toolkits for performing sentiment evaluations. Both toolkits provide libraries for opinion mining and sentiment classification. The sentiment analysis performed by LingPipe resembles that of general text classification, though. According to the website, LingPipe does not offer any available models and relies on a classification framework to extract features, requiring users to create their own classifier. In contrast, NLTK supplies built-in classifiers - this means that since the classifier is pre-trained, results can be obtained more quickly than in many other analyzers. However, VADER is best suited to language used in social media, like short sentences with some slang and abbreviations. It is less accurate when rating longer, structured sentences, but it’s often a helpful starting point. Deciding which toolkit to use is dependent on a user’s use case and the goals that the user is trying to accomplish.

References:

Adamjee, Uzair. “Introduction to Sentiment Analysis Using Python NTK Library.” Python.plainenglish.io, 16 Jul. 2020, https://python.plainenglish.io/introduction-to-sentiment-analysis-using-python-nltk-library-f00c227ef56e

Calderon, Pio. “Vader Sentiment Analysis Explained.” *Medium*, Medium, 31 Mar. 2018, https://medium.com/@piocalderon/vader-sentiment-analysis-explained-f1c4f9101cd9#:~:text=VADER%20(Valence%20Aware%20Dictionary%20for,intensity%20(strength)%20of%20emotion.

Zhai, ChengXiang. “Opinion Mining and Sentiment Analysis: Motivation” Coursera, https://www.coursera.org/learn/cs-410/lecture/o93Yl/11-5-opinion-mining-and-sentiment-analysis-motivation.

“Natural Language Toolkit.” NLTK, 25 Mar. 2022, https://www.nltk.org/

“Sentiment Tutorial.” alias-i.com, http://www.alias-i.com/lingpipe/index.html

“What is LingPipe.” alias-i.com, http://www.alias-i.com/lingpipe/index.html