Tech Review: BeautifulSoup and NLTK in Sentiment Analysis of Website Data

Karl Vosatka

CS410

Prof. Cheng Zhai

11/6/2022

There are undoubtedly a large number of tools out there for efficient sentiment analysis of text data. As more data scientists begin to ask and answer questions about text data, these tools fulfill an important need for fueling sentiment analysis projects. Often, these projects entail two parts: first, the scraping of data off of websites through tools like BeautifulSoup, and second, the analysis of that text data through powerful computational linguistics libraries like NLTK in Python. BeautifulSoup and NLTK are powerful Python-based tools in collecting and parsing data for sentiment analysis of websites.

BeautifulSoup accomplishes the first step of assessing website data: acquiring it, via web scraping. Web scraping presents a logistical problem: HTML code is dense and can include a bunch of text formatted with incorrect syntax that is often referred to as “tag soup” (Browne-Anderson, 2017). BeautifulSoup takes that thorny and dense syntax and organizes it for easy access of material within a website. Essentially, it consumes data from a webpage’s HTML code and parses it into a series of tags which can be later called and searched for interesting data. For example, tags can be parsed to locate hyperlinks in navigating a website to reach user reviews or profiles. Large sections of text data under the appropriate tag can be scraped and aggregated for later analysis (Browne-Anderson, 2017). Though this may seem like a small accomplishment, BeautifulSoup’s simple and effective HTML parsing makes it a powerful and widely-used web scraping library.

BeautifulSoup has been used in a wide array of projects, in part due to its availability as open-source software. One such instance includes an art project by the New York Times where randomly pulled quotes from the full publication records of the newspaper are displayed across dozens of screens in the lobby of their building (Richardson, 2022). Another includes the application of web scraping tools to aggregate research and data about the COVID-19 pandemic for the benefit of researchers. This proved invaluable to researchers in the early days of understanding the virus (Richardson, 2022). The many applications of BeautifulSoup are impressive, and include Sentiment Analysis-relevant projects as well.

Applications for scraping a website for sentiment analysis are clear. Scraping a website’s hyperlinks via analysis of tags allows for navigation for pages of interest, including user review pages. Once these pages are reached, the contents of reviews are identified by their HTML tag and collected as separate documents for later sentiment analysis. In addition to scraping text documents, it is prudent to collect data such as explicit ratings that can later be compared to the sentiment analysis results, perhaps as part of a training set. Further details can also be gathered to track the opinion target as well as other deeper opinion analysis components. This could include opinion holder details such as their average review score across their profile or their overall number of reviews. Also, explicit or implicit consideration of review popularity might be assessed via these tags through review ratings (i.e. if a certain number of users find a review helpful, or ranking of reviews by popularity). This may be desirable in weighing the relative contribution of a review in the sentiment analysis model as desired.

Now that data has been aggregated, it is available for building a sentiment analysis through NLTK. NLTK is a Python-based language toolkit initially designed in 2002 by computer scientists at the University of Pennsylvania in part to present a simple, unified package of computational language analysis tools for instructional purposes (Loper and Bird, 2002). Sentiment analysis via NLTK can start with the parsing of text documents into a bag-of-words representation of relevant vocabulary in a document. Tools like word_tokenize can break a text document into its constituent words automatically or via user-provided regular expressions. For example, the set of words that are provided can be separated on the basis of punctuation, spaces, and tabs. The default settings of word_tokenize account for many common regex expressions such as these (Loper and Bird, 2002; Mogyorosi, 2022). Stopwords, or common words that generally are irrelevant to the specific topic of the text, can also be removed with reference to a list of stopwords. These tokens can then be combined into n-gram combinations or part-of-speech n-grams. This can be useful in more fine-grained analysis of word context. For example, in a product review, a box of granola bars can be described as having “only four” bars in it. The use of only might imply dissatisfaction, but this element would be less easily detected in a pure bag-of-words model.

NLTK also contains a variety of tools to further prune the vocabulary present to mostly content that is relevant to the opinion. One tool allows for tracking of collocations, which are useful for tracking words that often occur together (e.g. “good battery”, “boring plot”). Another allows for defining word classes for better analysis of phrases that have similar meaning with different words. There is also a tool called concordance that enables users to inspect text data for terms that might occur around a word of interest. This tool is helpful in identifying related concepts for analysis (NLTK.org, 2022; Mogyorosi, 2022).

Once all of these tools have shaped the data to pull out relevant themes, text is ready to be classified via NLTK's classifier functions. NLTK incorporates a number of different classifiers that can be used for sentiment analysis, including a Naive Bayes approach (nltk.classify.naivebayes), Support Vector Machines via scikitlearn (nltk.classify.scikitlearn), or using machine learning-based classifying methods like decision trees (nltk.classify.decisiontree; NLTK.org, 2022). These and other classifiers offer plenty of options for classifying data that's been cleaned and specified via NLTK.

NLTK and BeautifulSoup are both very impressive and helpful tools in conducting sentiment analysis of website data. BeautifulSoup provides intuitive and sentiment analysis-friendly parsing functions for extracting documents and features for using with a sentiment analysis model. NLTK can then pre-process the documents for use with a series of different classifier options in training and employing the model. These tools make sentiment analysis projects very accessible to all, and lead data scientists worldwide to answer interesting questions about text data.

Works Cited

Bowne-Anderson, H. (2017, October 13). *Web scraping & NLP in python*. DataCamp. Retrieved November 6, 2022, from https://www.datacamp.com/tutorial/web-scraping-python-nlp

Loper, E., & Bird, S. (2002). NLTK: the Natural Language Toolkit. *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics* -. https://doi.org/10.3115/1118108.1118117

Mogyorosi, M. (2022, September 1). *Sentiment analysis: First steps with Python's NLTK library*. Real Python. Retrieved November 6, 2022, from https://realpython.com/python-nltk-sentiment-analysis/#extracting-concordance-and-collocations

Richardson, L. (2022). *Beautiful soup*. Beautiful Soup: We called him Tortoise because he taught us. Retrieved November 6, 2022, from https://www.crummy.com/software/BeautifulSoup/