Tech Review for CS 410, Fall 2022
Overview of NLTK

Charles Stolz cstolz2@illinois.edu

The Natural Language Toolkit is a framework for performing natural language processing tasks in the Python programming language. It is a very popular framework with over 11,200 stars on GitHub. The toolkit includes a companion book available for free on the web called *Natural Language Processing with Python*. NLTK has a large number of corpora available which can be easily imported into a project. The total list of available corpora can be found at nltk.org/nltk_data. NLTK includes extensive documentation on potential use cases and how to interact with its datasets and models. Since it is a Python based toolkit multiple platforms are supported including Windows, Mac, and Linux. Support is available in a virtual environment, Anaconda conda environment, or standard Python 3.7 or later. Datasets can be downloaded using the nltk.downloader or nltk.download methods, this facilitates the clean downloading and loading of datasets directly into memory using a Python procedure.

The NLTK tokenize package enables text preprocessing through tokenization, a process to break down data into smaller parts called tokens. The tokenizer accepts a string as input and then outputs a Python list of substring tokens. The package also provides a feature to produce token-spans, where the input is a string and the output is a Python list of tuples where tuple[0] is the start of the token substring and tuple[1] is the end of the token substring. There are many different modules within the tokenize package such as punkt which divides the string into a list of sentences and regexp module to split the string using regular expressions.

NLTK includes a stemming package. Stemming algorithms are used to remove a word's suffix, leaving only the stem which reduces dimensionality. NLTK includes a stem module called Porter. This is a very well known technique dating back to the 1980s and introduced in the paper *An Algorithm for suffix stripping* (Porter, n.d). The Snowball Stemmer is another module offered by NLTK which is a newer version of Porter's stemmer with updated rules. Lancaster stemmer is another stemmer module available, this is a more aggressive stemmer introduced in the article *Another stemmer* (Paice, 1990). The Lancaster steemer attempts to reduce the word to the shortest stem. NLTK provides a rich set of interfaces to the most widely used stemming techniques used in industry.

POS tagging is a critical step in NLP where a tag is added to each word in a sentence to identify its part-of-speech. The tagging package in NLTK includes the Penn Treebank target for english. There are 36 part-of-speech tags used in Penn Treebank such as "CD" to indicate a cardinal number (Gimenez, n.d.). There is also an interface to the Stanford POS tagger, TnT tagger and others.

The NLTK parsing package provides interfaces to create tree structures of text. The ShiftReduceParser module creates a stack from left to right by applying a shift and reduce operation. The shift operation moves the token to the end of the stack. The reduce combines the stack elements into a tree (NLTK :: nltk.parse.shiftreduce Module, n.d.) . The RecursiveDescentParser uses recursion to parse after constructing a tree and then expands the frontier.

NLTK can be used for classification, examples include spam detection and topic modeling. The nltk.classify package provides interfaces for labeling tokens. There are several submodules available for a range of classification tasks such as decision tree module to enable assignment of a label to a token based on a tree.

NLTK includes a semantic interpretation package. This is used to represent semantic structure in first-order logic and evaluate the models recursively. The chat80 module allows for the integration of Prolog knowledge base. The logic module is built on lambda calculus. The evaluate module includes the data structures for representing the models.

The NLTK sentiment package can be used to perform sentiment analysis. There are three submodules sentiment_analyzer, util, and vader. The sentiment_analyzer module has many useful methods. The primary use case for it according to the documentation is teaching and demos. Vader, “Valence Aware Dictionary and sEntiment Reasoner”, module is a robust algorithm included in the NLTK package which can be used to mine the opinions in product reviews, news articles, and many other use cases. Vader uses words and phrases in its dictionary to help determine the sentiment of the text, punctuation is also evaluated and included in the score. The end goal is to determine if the text is negative, neutral, or positive. The approach taken by Vader is very useful in mining sentiment from social media posts. The util submodule includes useful utility methods for preparing data and timing procedures.

NLTK includes a full test suite in the nltk.test package. Unit tests can be used in automation and CI/CD to create robust software. When moving from prototyping phase to production the ability to test software and ensure code changes do not break the application is very valuable. Having a test suite directly in the toolkit is a sign of a mature framework. When performing experiments unit tests can be used to reproduce the same experiment and ensure any code changes are not going to impact the results.

In conclusion, the NLTK toolkit is a very feature rich framework with above average documentation, curated datasets and models, code examples, and a full test suite. It is an extremely valuable tool for the researcher and for teaching NLP. Since it is Python based you can interact with data directly in the REPL interpreter, within a jupyter notebook, using a script, or within a full stack application.

Works Cited

Porter, M. F. (n.d.). *An algorithm for suffix stripping*. Toronto.edu. https://www.cs.toronto.edu/~frank/csc2501/Readings/R2_Porter/Porter-1980.pdf

Paice, C. D. (1990). *Another stemmer*. Association for Computing Machinery. https://dl.acm.org/doi/abs/10.1145/101306.101310

Gimenez, J. (n.d.). *Penn Treebank Tag Set*. Retrieved November 6, 2022, from https://www.cs.upc.edu/%7Enlp/SVMTool/PennTreebank.html

NLTK :: nltk.parse.shiftreduce module. (n.d.). Retrieved November 6, 2022, from https://www.nltk.org/api/nltk.parse.shiftreduce.html