Analysis of NLTK and Comparison of NLTK to Other Python NLP Libraries

by Josephine Falso

CS410 Fall 2021, University of Illinois – Urbana Champaign

This paper will provide an overview of NLTK, a leading Natural Language Processing toolkit in Python. NLTK stands for Natural Language Toolkit, a free, open-source Natural Language Processing platform. NLTK allows programmers to build Python programs that organize, analyze, and interpret human language. NLTK is available for Windows, Mac OS, and Linux. While there are many other Natural Language Processing libraries available in Python, this paper will compare and contrast NLTK with TextBlob, Gensim, and spaCy.

NLTK was created in 2001 by Steve Bird and Edward Loper of the University of Pennsylvania. Along with Ewan Klein, their book, "Natural Language Processing with Python", is an exhaustive resource on the NLTK platform. In addition to the book, there is a breath of information available on the Internet and active user forums and wikis for NLTK. NLTK comes with many corpora, toy grammars, and trained models, which can be downloaded for free and used with guided exercises and examples in the book.

NLTK allows classification, tokenization, stemming, POS tagging, parsing, semantic analysis, and many other text analysis functions. A number of these functions naturally return tree objects.

Once the text has been parsed, NLTK also has a built-in frequency counting method (FreqDist) and a method to identify the most common words (most_common). A full explanation of the uses and methods are too numerous to list here, but NLTK's usefulness spans both text analysis and text mining.

NLTK has become prolific in the field of NLP. It is used by across industry and academia by large corporations, students, researchers, and engineers; however, more emphasis is placed on its use within the academic community. It has stopword lists in twenty-two languages. For example, NLTK has its own built-in lists of stopwords (stopwords) and punctuation (string), so that the data can easily be cleaned within NLTK, and the programmer can focus his/her attention on the practice of language processing.

The creators of NLTK have continued to develop the toolkit since its inception in 2001. To appeal to students who were learning to code, the NLTK developers created a scaled-down version of NLTK in mid-2005 called NLTK-Lite. It was simpler and faster than NLTK at that time. "NLTK-Lite leveraged standard Python objects instead of custom NLP versions, so that students learning to program for the first time would be learning to program in Python with some useful libraries, rather than learning to program in NLTK."[1] "Fundamental representations were simplified, streaming tasks were implemented as iterators instead of lists to limit memory usage, and method names were shortened to be easier to read and use. The barrier to entry for contributed software was removed since there was no requirement to support the special NLTK token architecture."[2] "Once it reached version 1.0 (in mid 2009), NLTK-Lite took over the original NLTK name and became NLTK 2.0."[1] The current version of NLTK is version 3.0.0.

A much simpler Python library for NLP is TextBlob. TextBlob provides a simple API to access its methods. This library is well-suited for beginners because of the simple interface.

It offers sentiment analysis, pos-tagging, noun phrase extraction, classification, and translation. To accommodate foreign languages, TextBlob relies on Google Translate. Its simplicity and ease of use makes TextBlob a natural choice for semantic analysis. It has better memory usage and higher processing speed than NLTK. However, text modeling is primarily unsupervised, and TextBlob does not provide enough tools to be used on its own without other libraries like NLTK.[3]

Gensim is an open-source library for NLP written in Python and Cython. It is an unsupervised library for topic modelling, document indexing, and similarity retrieval. The target audience is the natural language processing and information retrieval community, unlike NLTK, which is more focused on the educational community. Like NLTK, Gensim publishes its own datasets for learning exercises and data training. Gensim is used to implement popular algorithms such as Latent Semantic Analysis**,** Latent Dirichlet Allocation, Hierarchical Dirichlet Process, and word2vec deep learning. To use Gensim, the Python libraries NumPy and SciPy must also be installed. Gensim can handle large corpora faster and more efficiently than NLTK. Gensim is also known for vector space modeling capabilities. Unfortunately, Gensim has fewer customization options than its competitors in the NLP space.[3]

SpaCy is an open-source NLP library written in Python and Cython. This library is for advanced natural language processing, and it is the fastest and most accurate syntactic parser. SpaCy is also an unsupervised library for pos-tagging, tokenization, entity recognition, word vectors, text classification and more. It uses an API to access its methods and properties governed by trained machine learning models. Implementation of spacy and access to these properties is initiated by creating pipelines. A pipeline is created by loading the models, typically a tagger, lemmatizer, parser, and entity recognizer. The components of the pipeline can be customized by the programmer.[4] While NLTK is primarily suited for the academic sector, spaCy is tailored to application developers, and spaCy should be used in a production environment. SpaCy uses much more memory than NLTK, but it returns results much faster. SpaCy currently supports over sixty-four languages.[3]

Each of the Python NLP libraries in this paper takes a different approach to space and time and targets a different user community. NLTK is slower to return results, and it is primarily used by scholars and researchers who want to explore ideas, while Gensim and spaCy are used by application developers in production environments. However, each of these libraries is an excellent choice for any project that relies on machine understanding of human languages.

References

1. Aarsen, T. (n.d.). *Natural Language Toolkit FAQ*. GitHub, Inc. Retrieved November 7, 2021, from https://github.com/nltk/nltk/wiki/FAQ
2. Bird, S. G. (2005). NLTK-Lite: Efficient Scripting for Natural Language Processing. Proceedings of the 4th International Conference on Natural Language Processing, pp.11-18. Allied Publishers.
3. Kozaczko, D. (2018, June). *8 best Python Natural Language Processing (NLP) libraries*. Sunscrapers. Retrieved November 7, 2021, from https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp-libraries/
4. spaCy. (2021). *Industrial-Strength Natural Language Processing in Python*. Retrieved November 7, 2021, from https://spacy.io/