CS410 Tech Review - Kyr Nastahunin (knasta2)

NLTK for language processing.

In one of the classes I took as an undergraduate student, we had a homework where we had to classify tweets that either supported the Democratic party or the Republican party. For this homework, we used NLTK and SKLearn, and using those libraries was very straightforward. It made me think that natural language processing is actually pretty easy. All you need is a good dataset. However, after more research, I realized that the reason it was so easy is that the creators of those packages already did the majority of the work for me. All I had to do was use the very convenient APIs they provided for me. Packages like NLTK, SKLearn, and MeTa make text processing a lot more feasible for people like me, who are not scientists and don't have the capacity to implement all the algorithms and methods from scratch.

Natural Language Toolkit is a data science toolkit built for Python that allows its users to perform text classification, tokenization, stemming, tagging, parsing, and semantic reasoning. It has comprehensive documentation and is a very popular choice for Natural Language Processing projects. The very first example on the main page of the toolkit shows how in just 4 lines of code, you can parse a sentence, build a semantic tree of that sentence, and display the tree on the screen. If someone were to perform the same work from scratch, it would take them hundreds of lines of code as well as a good amount of time to research all the underlying algorithms used for POS tagging, stemming, etc.

Another thing that makes NLTK a popular choice among its users is comprehensive documentation. The importance of having a well-documented library cannot be overstated, especially with Python being a dynamically typed language, which makes it even harder for users to use and learn new APIs. Additionally, NLTK works well with Pandas, NumPy and even has a wrapper for Sklearn to use its prebuild classifiers. However, despite being well documented, sometimes the documentation lacks meaningful examples and instead shows the most basic usage, which I find to be common sense and not needing documentation examples.

NLTK also provides datasets of wordnet, stopwords, punctuations, taggers, and so on for multiple languages. This allows the users to use a centralized dataset for all those features and therefore avoid inconclusive answers. Unlike MeTa, NLTK is available for any version of python and still gets updates from time to time. It is an open-source project and depends mainly on the efforts of volunteers.

Let's talk a little more about what it's like to use this tool. According to devopedia.org a typical pipeline for text classification looks like the following:

1. Text tokenization
2. Removal of stop-words
3. Stemming and lemmatization
4. Constructing a bag of words
5. Performing TF-IDF weighting
6. Word embedding
7. Model building
8. Model evaluation.

NLTK's architecture is modular, so each of those steps can be performed with NLTK using various NLTK modules specified for those tasks. For example, text tokenization is performed using the tokenize module, stemming using the stem module, and so on. You can read more about NLTK's modular system on devopedia's page.

However, the tool has its disadvantages too. Despite having comprehensive documentation and being easy to use, mastering and using it to its full potential is actually hard. Additionally, some of the wrappers that NLTK uses make the library run slow and inefficient. As was mentioned, the tool has a lot of modules. However, only a few of those modules are actually used frequently, while others aren't needed for the majority of tasks.

If you're interested in NLTK, it is also helpful to know about the existence of TextBlob. TextBlob is a Python library that uses NLTK and Pattern libraries. In other words, you can consider it to be an extension of those two. The reason this library is important is that many confirm that it is easier to use and learn than NLTK while being just as powerful, or even more powerful, in some cases.

I will conclude by saying that NLTK is the most popular NLP library for a reason. Being the most popular, it has a big community that creates tutorials, answers questions, and even contributes to the open-source library. All those things are very important for someone who is only learning NLP. If you are a seasoned veteran who is pushing for the newest and most advanced technologies in your project, you might want to turn your attention toward SpaCy. It is a new library that is known for its high-performance and ready-to-use solutions. Many say this library is set to replace NLTK from its position as the most commonly used NLP library.

References

NLTK documentation - nltk.org
NLTK SciKit Wrapper Documentation - nltk.org/api/nltk.classify.scikitlearn.html
Devopedia - Natural Language Toolkit - https://devopedia.org/natural-language-toolkit
TextBlob - https://textblob.readthedocs.io/en/dev/
5 Heroic Tools for Natural Language Processing - Vladimir Fedak -
https://fedakv.medium.com/5-heroic-tools-for-natural-language-processing-7f3c1f8fc9f0