CS 410 – Technology Review: [Tools] NLTK for information extraction

A discussion on NLTK approach, design and usages. A brief comparison and analysis with its alternative approaches.

By Sameen Shaukat – shaukat2@illinois.edu

NLTK – Natural Language Toolkit is an open source library designed in Python. It's a framework that assists in processing human language data for computer interpretation. NLTK has a wide use in fields of Artificial Intelligence, natural language recognition & translation and numerous others.

NLTK consists of most common algorithms in natural language processing domain. It enables tokenization, part of speech tagging, stemming, sentiment analysis, topic mining, entity classification and semantic analysis. Basically, this library provides all algorithms required to perform generic text analysis.

NLTK Approach and Design:

NLTK is composed of sub-packages and modules. A typical processing pipeline will call modules in sequence. Python data structures are passed from one module to another. Text is first passed through sentence segmentation module which breaks text into sentences. These sentences are passed through tokenization process to generate tokenized sentences. Tokenized sentences are then passed to POS Tagging module to generate POS – Tagged sentences. These then can be used for classification and relation analysis. Below flow chart is taken from Bird, Steven, Ewan Klein, and Edward Loper. 2019. "Natural Language Processing with Python.".

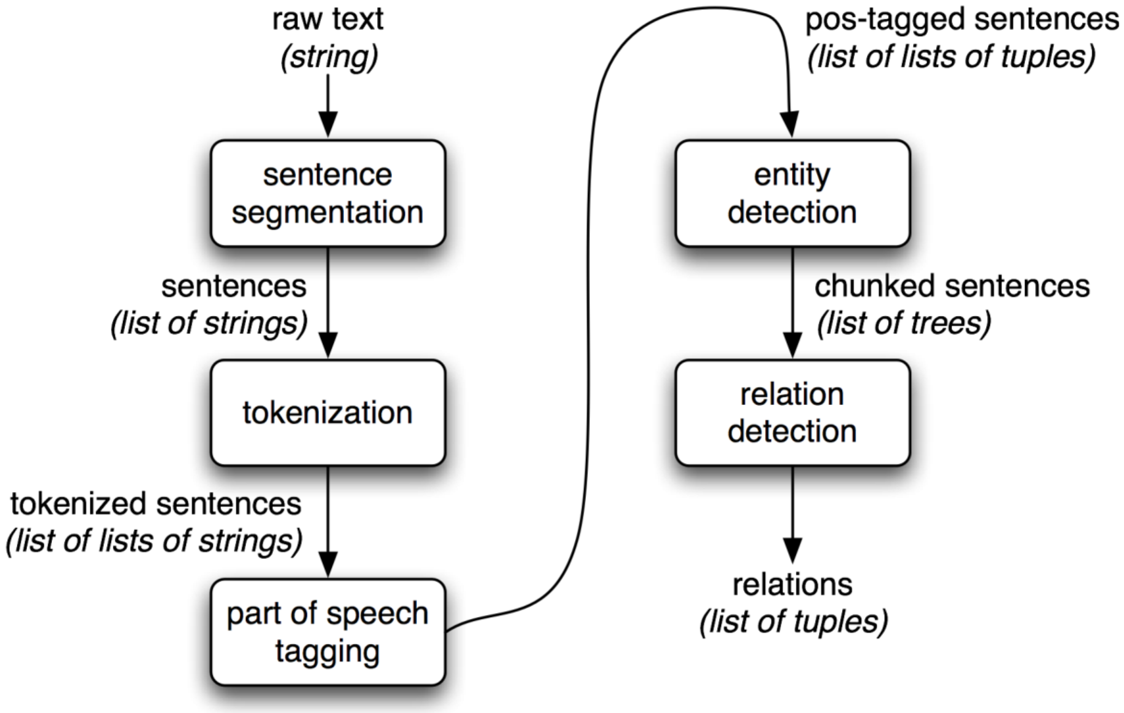
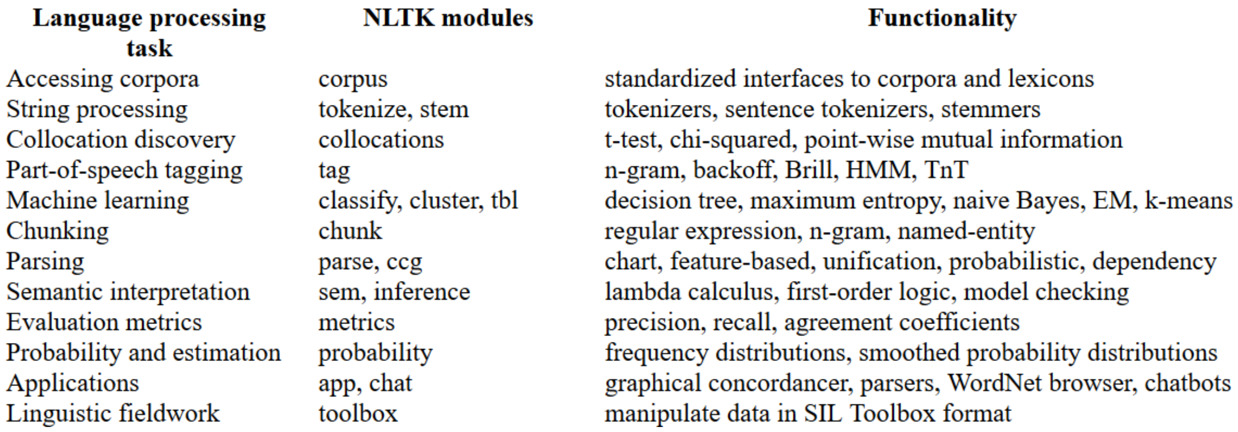


Figure 1: NLTK Basic Flow Chart

NLTK is built in modular mode. This package is further divided into sub-packages and modules. Following table taken from Bird, Steven, Ewan Klein, and Edward Loper. 2019. "Natural Language Processing with Python." explains the modules of NLTK.



Language processing task	NLTK modules	Functionality
Accessing corpora	corpus	standardized interfaces to corpora and lexicons
String processing	tokenize, stem	tokenizers, sentence tokenizers, stemmers
Collocation discovery	collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	tag	n-gram, backoff, Brill, HMM, TnT
Machine learning	classify, cluster, tbl	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	chunk	regular expression, n-gram, named-entity
Parsing	parse, ccg	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	sem, inference	lambda calculus, first-order logic, model checking
Evaluation metrics	metrics	precision, recall, agreement coefficients
Probability and estimation	probability	frequency distributions, smoothed probability distributions
Applications	app, chat	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	toolbox	manipulate data in SIL Toolbox format

Table 1: NLTK Modules

NLTK Features and Usage:

- NLTK provides easy-to-implement interfacing for 50 corpora and linguistics sources.
- NLTK has useful algorithms for pre-processing of text and analysis of 50 corpora.
- NLTK is suitable for translators, educators, researchers, and industrial applications and accessible in multiple operating systems.
- NLTK library also contain a lot of NLP datasets that can be used for practice or combined for generating models.

Alternative Approaches to NLTK:

There are many NLP libraries in use today. However, NLTK is the most usable. Other mentionable libraries for natural language processing are

1. spaCy
2. Gensim
3. Stanford CoreNLP
4. TextBlob

spaCy

5. NLTK is slow, whereas spaCy is said to be the faster substitute.

- NLTK is a string processing library that is each function inputs and outputs a string. Whereas, spaCy is built on an object-oriented approach. Each function returns objects.
- Each library utilizes either time (NLTK) or space (spaCy) to improve performance.

Gensim

Gensim is has latest computational efforts and algorithms in NLP while NLTK packages standard algorithms and datasets. Gensim is used for topic and vector space modeling, document similarity generally heavier and industry level tasks. It supports deep learning.

Stanford CoreNLP

Stanford CoreNLP includes part-of-speech (POS) tagging, entity recognition, pattern learning, parsing, etc. This library is written in Java but Python wrappers are publicly available. Many organizations in market use Stanford CoreNLP for market ready implementations. This library provides fast, accurate, and multiple major languages support.

TextBlob

TextBlob provides an interface to NLTK. This is used for processing textual data and provide mainly all types of operations in the form of API.

Conclusion

NLTK is more academic than industrial level package, which is why it is made part of many university courses curriculum. It gives an easy, extendible outline intended for assignments and class demonstrations as it is heavily and sufficiently documented, open source and provides easier implementation.