Python Natural Language Processing libraries

Bassam Alahmadi

Abstract

Natural Language Processing (NLP) is a subfield of linguistics and artificial intelligence that is concerned with the interactions between computers and human language. Usually it describes any kind of computer manipulation of natural language. NLP is subdivided into: natural language understanding (NLU) and natural language generation (NLG) by artificial intelligence, which deals with machine reading comprehension. It can be used across a wide range of applications including summarizing information, searching, conversational bots and translations between languages.

Formerly, experts could were part of natural language processing of projects that require a superior knowledge of linguistics, machine learning, and mathematics. Nowadays, developers can use ready-made tools to simplify text pre-processing. This means they can concentrate on building machine learning models through the NLP library which contains many tools created to solve NLP problems.

Introduction

Python NLP libraries are one of the most famous and contain several different kinds of components. There are many things Python has that make it a good programming language choice for NLP projects. Moreover, developers can enjoy excellent support for integration with other languages, and tools that enable developers to handle a large number of NLP-related tasks, such as topic modelling, document classification, word vectors, part-of-speech (POS) tagging, and sentiment analysis[1].

Many tools and libraries have been created to solve NLP problems; for example, Natural Language Toolkit (NLTK), Gensim, polyglot, TextBlob, CoreNLP, SpaCy, Pattern, Vocabulary… etc [2]. And It is difficult to determine which the best of these is, because for each library there are many features that assist in solving problems related to NLP, and these different in how they are used and the languages of interest. Usually, when installing Python for the Windows platform it will include the entire standard library and numerous additional components. This article provides an overview of two of these libraries [NLTK -Textbold].

NLTK

The Natural Language Toolkit (NLTK) is among the best-known and most powerful of the Python natural language processing libraries. It integrates many related lexical resources, such as WordNet, NPS Chat, Web Text Corpus, FrameNet, SemCor and many more. In addition, NLTK offers a wide variety of tools for working with text: “classification, stemming, tokenization, tagging, semantic reasoning and parsing.” It can also work with some third-party tools to enhance functionality[3]. I will explain some of these tools later.

The library was developed at the University of Pennsylvania by Steven Bird and Edward Loper (2002), and since that time it has played a key role in breakthrough NLP research. We can say that it is now common for universities around the globe to use NLTK, Python libraries, and similar tools in their courses.

This library is reasonably versatile, but we it is quite difficult to use, and can be rather slow and not matched to the demands of quick-paced production and usage. However, this issue can be overcome speeding it up with parallel processing. NLTK is considered suitable for linguists, developers, students, engineers, researchers, educators and industry users alike. Moreover, it is a free, open source, community-driven project. It is available for all operating systems such as Windows, Mac OS X, and Linux.[4]

Features of NLTK

- The text processing libraries of NLTK support working with text classification, tagging, tokenization, stemming, parsing, and semantic reasoning.
- NLTK contains a graphical illustration of data science.
- It is open source and contains over 100 corpora and lexical resources, such as question classification, open multilingual wordnet, SentiWordNet, Stopwords Corpus, SEMCOR and many more.
- Structure types, structure strings parsing, different pathways, and re-entrance.
- It is available for all operating systems such as Windows, Mac OS X, and Linux.

TextBlob

TextBlob offers a neat Application program Interface for performing common NLP tasks. It offers a friendly front-end to the Pattern and NLTK libraries and has been developed to perform different tasks to those libraries in high-level, easy-to-use interfaces without complexity. It takes advantage of leveraging native Python objects and syntax. A further advantage of TextBlob is that it supports the creation of high-level objects that combine components such as sentiment analyzer, classifier, etc.—and re-use them with minimal effort. This means a prototype can be used quickly with TextBlob, then refined later. This means that using TextBlob is a must for developers who start their journey with NLP in Python and want to make the most of their first encounter with NLTK[5].

Features of Textbold

- Easy-to-use interfaces.
- Language translation and detection powered by Google Translate.
- Smooth integration with other programming languages through its Application programs Interfaces.
- Open-source Python library for processing text-based data.
- Adds new models or languages through extensions.

Some tools

- POS Tagging

  Parts of speech Tagging assigns specific tokens to each word in the languages covered, such as noun, verb and adjective, etc.

  For example :

  Input : Everything to permit you.

  Output: [('Everything', NN),('to', TO), ('permit', VB), ('you', PRP)]

- Tokenization of Sentences

  This module breaks down each word with punctuation which you can see in the output. For example:

  Input : I buy car it is black.

  Output:[‘ I buy car’,’ it is black’]

- Stemming

  Stemming is a type of normalization for words. Normalization is a technique whereby a set of words in a sentence are converted into a sequence to shorten their lookup. For example:

  Input : Wait, waiting, waited, wats

  Output: wait, wait, wait, wait

- Porterstemmer

  Data is filtered to help with better machine training.

  For example:

  Input: I will going to Manchester.

  Output: I

  Will

  Going

  To

  Manchester

- Lemmatization

  It returns the lemma as the base form of all inflectional forms.

  For example:

  Input: studies

Output: Lemma for studies is study

- Wordnet

  Wordnet is an NLTK corpus reader, a lexical database for English. It can be used to find the meaning of words, synonyms or antonyms.

  For example:

  Input: wordnet.synsets("dog")

  Output: [Synset('dog.n.01'), Synset('frump.n.01'), Synset('dog.n.03'), Synset('cad.n.01'), Synset('frank.n.02'), Synset('pawl.n.01'), Synset('andiron.n.01'), Synset('chase.v.01')]

- Tagging Sentences

  Tagging Sentences in a broader sense refers to the addition of the labels of noun, verb, etc.

  For example:

  Input: "I have to build a good site, and She love visiting I site."

  Output: [('I', 'PRP'), ('have', 'VBP'), ('to', 'To'), ('build', 'VBN'), ('a', 'DT'), ('good', 'JJ'), ('site', 'NN'), ('and', 'CC'), ('she', 'PRP'), ('love', 'VBP'), ('visiting', 'VBG'), ('I', 'PRP$'), ('site', 'NN'), ('.', '.')].

Reference

[1] Natural Language Processing (NLP) with Python — Tutorial, , viewed 12 Sep 2020, https://medium.com/towards-artificial-intelligence/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0

[2] 8 great python libraries for Natural Language Processing, viewed 12 Sep 2020, https://www.infoworld.com/article/3519413/8-great-python-libraries-for-natural-language-processing.html

[3] Al, NLG, and Machine learning Tob 10 NLP Libraraies, viewed 12 Sep 2020, https://discover.bot/bot-talk/top-10-nlp-libraries/.

[4] NLTK 3.5 Documentation, viewed 14 Sep 2020, https://www.nltk.org/.

[5] TextBlod: Simplifiied Text Processing, viewed 15 Sep 2020, https://textblob.readthedocs.io/en/dev/.