Some useful Open source Python libraries for NLP

In this course, we primarily studied techniques for Text Retrieval and Text Analytics. Natural Language Processing is at the heart of both Text Retrieval and Text Analytics. In today's world, where it is becoming increasingly important to extract information from text, there are various libraries that are available to Data scientists.

These libraries make the basic tasks for Text pre-processing much simpler, so developers can concentrate on creating advanced analytical applications through machine learning models.

In this article, I wanted to look at two of the most common libraries available in Python for Natural Langauge Processing –

- Natural Language Toolkit (NLTK)
- Spacy
- FlashText
- Fuzzywuzzy

NLTK (Natural Language Toolkit)

NLTK is an open source, community driven free project. It is one of the most widely used NLP library. It provides libraries for performing the following –

- Classification
- Tokenization
- Stemming
- Tagging
- Parsing
- Semantic Reasoning
- Support for 10+ languages for most of the above tasks

Additionally, there are various third party extensions available for developers to use.

Couple of points to note about NLTK –

- Performance tends to be slow
- No integrated word vectors

Documentation

Installation

```
pip install -U nltk
pip install -U numpy
```

spaCy

spaCy is one of the fastest (industrial-strength) NLP framework which excels at large-scale information extraction. The library supports the following –

- Classification
- POS tagging
- Tokenization
- Tagging
- Parsing
- Entity recognition
- Syntax-driven sentence segmentation
- Pre-trained word vectors
- Deep learning integration

Point to note about spaCy

- Not as flexible as NLTK

Installation

```
pip install -U spacy
```

Documentation

FlashText

To perform NLP on large text, there's often need to replace or extract keywords. FlashText module, based upon the FlashText algorithm, is an excellent library to perform such tasks. Read more about this here.

Installation instructions

```
$ pip install flashtext
```

Example:

Extract keywords

```
from flashtext import KeywordProcessor
keyword_processor = KeywordProcessor()
# keyword_processor.add_keyword(<unclean name>, <standardised name>)
keyword_processor.add_keyword('Big Apple', 'New York')
keyword_processor.add_keyword('Bay Area')
keywords_found = keyword_processor.extract_keywords('I love Big Apple and Bay Area.')
keywords_found
['New York', 'Bay Area']
```

Replace keywords

```
keyword_processor.add_keyword('New Delhi', 'NCR region')
new_sentence = keyword_processor.replace_keywords('I love Big Apple and new delhi.')
new_sentence
'I love New York and NCR region.'
```

Official Documentation here -

Fuzzywuzzy

This is a very useful library, used by thousands of developer for string matching. This is very handy to evaluate string comparison ratios, tokens ratios etc..

Installation:

```
$ pip install fuzzywuzzy
```

Example:

```
from fuzzywuzzy import fuzz
from fuzzywuzzy import process

# Simple Ratio
fuzz.ratio("this is a test", "this is a test!")
97
# Partial Ratio
fuzz.partial_ratio("this is a test", "this is a test!")
 100
```

More interesting examples can be found at their GitHub repo.