NLTK vs. spaCy for Natural Language Processing

By: Preeti Amin

CS410 – Text Information Systems

NLTK is an open-source Python library that provides useful tools for natural language processing such as classification, tokenization, stemming, tagging parsing and semantic reasoning. It was developed by Steven Bird and Edward Loper at the University of Pennsylvania in 2001 mainly for research and education. spaCy is another Python open-source natural language processing library developed by Matthew Honnibol and Ines Montani in 2015 but with the goal of production use. Below are some highlights of the two libraries:

Popularity: NLTK has been around for about 2 decades and has been a popular library for natural language processing; whereas, spaCy is relatively new (2015) but has gained a large following since being released.

Performance: spaCy was written in Cython which is a superset of Python and improves code execution speed significantly by compiling Python code into C code. NLTK was written in Python which can be very slow for scalable deep NLP such as tagging and parsing.

Architecture:
NLTK follows a datatype-oriented approach where output of an algorithm is a list of strings of sentences. spaCy on the other hand, uses an object-oriented approach, where each function returns an object which allows for easier user exploration.

Language support: Though, initially spaCy only provided support for English, it now supports 19 languages including multi-language support which is especially useful for named entity recognition. Though, NLTK supports more languages, but languages supported vary by modules and algorithms. For examples, for stemming, RSLPStemmer is available for Portuguese, ISRIStemmer for Arabic, and SnowballStemmer, one of the most popular NLTK stemmers provides support for 14 languages.

Algorithms: NLTK provides various algorithms for the user to choose from, whereas spaCy tries to avoid asking the user to choose between equivalent algorithms. While NLTK's capabilities can be great for academia with the flexibility to choose specific algorithms, spaCy gets the developer going quickly with the current state-of-the-art algorithm.

Support for word vectors: spaCy has support for word vectors models like word2vec where NLTK does not. Although NLTK can be used in conjunction with Gensim to support word vectors, spaCy just makes the job easier with built in models.

Example of NER: Let's run an example of named entity recognition on a snippet of a CNBC article in both NLTK and spaCy.

```
sentence = "General Motors-backed Cruise is seeking final approval from
California to begin commercializing its robotaxi fleet in San Francisco. The
self-driving car start-up said Friday it submitted a permit for an autonomous
vehicle deployment with the California Public Utilities Commission. It is the
last of six permits needed from the CPCU and California DMV to begin charging
the public for rides."
```

I'll manually label the entities here:

```
General Motors ORG
Cruise ORG
California GPE
San Francisco GPE
Friday DATE
California Public Utilities Commission ORG
Six CARDINAL
CPCU ORG
California DMV ORG
```

Here, we run NER in NLTK using the code below

```
for sent in nltk.sent_tokenize(sentence):
  for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
     if hasattr(chunk, 'label'):
        print(chunk.label(), ' '.join(c[0] for c in chunk))
```

```
GPE California
GPE San Francisco
GPE California
ORGANIZATION Public Utilities Commission
ORGANIZATION CPCU
GPE California
```

Now, let’s run NER in spaCy

```
doc = nlp(sentence)

for ent in doc.ents:
    print(ent.text, ent.label_)
```

```
General Motors ORG
Cruise PERSON
California GPE
San Francisco GPE
Friday DATE
the California Public Utilities Commission ORG
six CARDINAL
California GPE
DMV ORG
```

From the code above, we can see that both libraries picked California, San Francisco and California (2nd time) as geopolitical entities. But spaCy also picked up Friday, six, DMV and General Motors. Though spaCy recognized the self-driving car company Cruise, it “thinks” it is a person instead of an organization. As we can see, neither result is perfect when compared with human annotation, but spaCy comes much closer.

Both NLTK and spaCy have their own place in natural language processing. Although NLTK has been a very popular library in the past and continues to be, especially within the research community, spaCy with its implementation of similarity-based models, ease of use and speed is quickly gaining steam.

References:

https://spacy.io

https://iq.opengenus.org/why-spacy-over-nltk/

Devopedia. 2019. "Natural Language Toolkit." Version 12, October 28. Accessed 2021-09-09.

https://devopedia.org/natural-language-toolkit

https://www.activestate.com/blog/natural-language-processing-nltk-vs-spacy/

https://nanonets.com/blog/named-entity-recognition-with-nltk-and-spacy/

https://www.cnbc.com/2021/11/05/gm-backed-cruise-seeks-final-approval-for-robotaxis-in-san-francisco.html