Comparison of NLP Toolkits including MeTA toolkit

Nowadays, a large amount of text data is produced on the internet waiting to be further mined, such as product reviews, articles about stock markets, and so forth. NLP techniques such as sentiment analysis, classification, topic mining, clustering, and many other algorithms can enhance people's level of understanding on entities and therefore help people make actions based on the improved understandings. In this report, I will go through and compare various kinds of NLP libraries so that readers can easily choose what libraries they should use in their project in terms of language, directly supported tasks, and extensions.

MeTA toolkit

Language: Python, C++
Tasks: Feature generation, Search, Classification, Regression, POS tagging, Parsing, Topic models, n-gram LM, Word embeddings, Graph algorithms, Multithreading

MeTA's philosophy is a unified framework for analysis of text data.
Many NLP libraries usually support one of the two pillars of NLP, which is retrieval and text analysis. For example, Lucene is a famous open source project for a search engine, but researchers must choose other tools if they want to experiment other NLP tasks such as topic modeling. Using MeTA will help researchers save their time by supporting a plethora of NLP tasks in one place. MeTA toolkit's architecture is built in a way that all the NLP tasks share the same index format either inverted index or forward index based on a type of the task, which makes it easier to try different types of NLP tasks without preprocessing the same corpus. Also, it supports multithreaded indexing and its backend is implemented in C++, which means it's fast enough for researchers to try different NLP tasks efficiently. In terms of usability, MeTA supports config based execution of NLP tasks, which means no coding is required when conducting exploratory data analysis.

CoreNLP

Language: Java
Tasks: tokenization, POS tagging, lemmatization, NER, parsing, entity linking, sentiment analysis

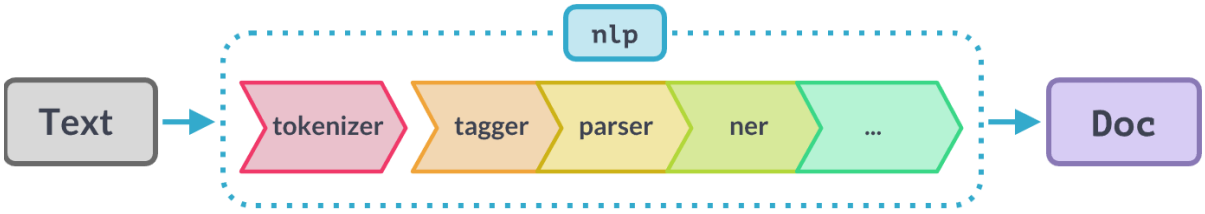
CoreNLP provides text analysis toolkits in Java language, which is quite rare. Users who want to provide NLP service based on the Java platform, CoreNLP would be a good choice. The

library supports various types of basic and advanced natural language analysis and it offers pretrained models for several languages.

spaCy

Language: Python
Tasks: Tokenization and training for more than 70 languages, POS tagging, parsing, NER, classification, multi-task learning with pre-trained transformers.

Unlike the MeTA library, spaCy is more focused on text analytics based on machine learning algorithms. This can be useful for developing and experimenting neural network based NLP algorithms. It turns text data into its internal representation object Doc and Vocab, which is shared by multiple documents to save memory.



(reference: https://spacy.io/usage/spacy-101#architecture-pipeline)
It supports extensions by implementing the Pipe class, which helps you develop your own NLP model. The trained model can be later used to predict the Doc, which is the representation of texts in the spaCy library I mentioned above.

HuggingFace Transformers

Language: Python (tensorflow, pytorch)
Tasks: classification, question answering, text generation, NER, summarization, translation, language modeling

HuggingFace transformers offer a platform where users can download and train pretrained-models, which makes users save their computing costs.
The library is strongly focused on machine learning based approaches in all of the NLP tasks, so the users should be familiar with machine learning frameworks such as tensorflow or pytorch.
The library offers an introductory guide on how to get a specific task done with the library, so users can easily start with their data to start training pretrained models to fit their needs.
Since the library offers various types of deep learning models on NLP tasks, applications that need advanced NLP tasks such as translation can utilize this library.