Tech Review

Author: Mochammad Dikra Prasetya (mdp9@illinois.edu)
Course: CS410 Fall 2020

Objective

I have years of experience in programming, but nearly zero experience in Python nor NLP (Natural Language Processing) before CS410. For this tech review, I am writing down my journey on getting going with the available Text Information System tools out there and share my new learner perspective on dealing with the dreaded learning curve.

What I commit to cover in this review are things like:
- What was I able to learn about the tool within a few hours?
- What did I find cool about the tool?
- What did I find helpful for learning the tool?
- What did I wish the tool had to help me learn better?
- What did I wish CS410 had to help me learn these tools better?

I chose **MeTA** as the first tool to dig with as I learned this in the course's MPs, so it should be a good starting point. The second tool I picked would be **NLTK** as it seems to be the most popular tool for NLP to date. The next thing I am interested in checking is **TensorFlow NLP**, since my limited knowledge says that Tensorflow is the coolest tool for machine learning stuff at the moment.

I'm learning each tool by following their guidance for beginners. At the end of this tech review, I am giving my 2 cents on how I would suggest new learners learn the tools smoothly.

Again, disclaimer: the writings below are my personal thoughts when quick-diving through these tools.

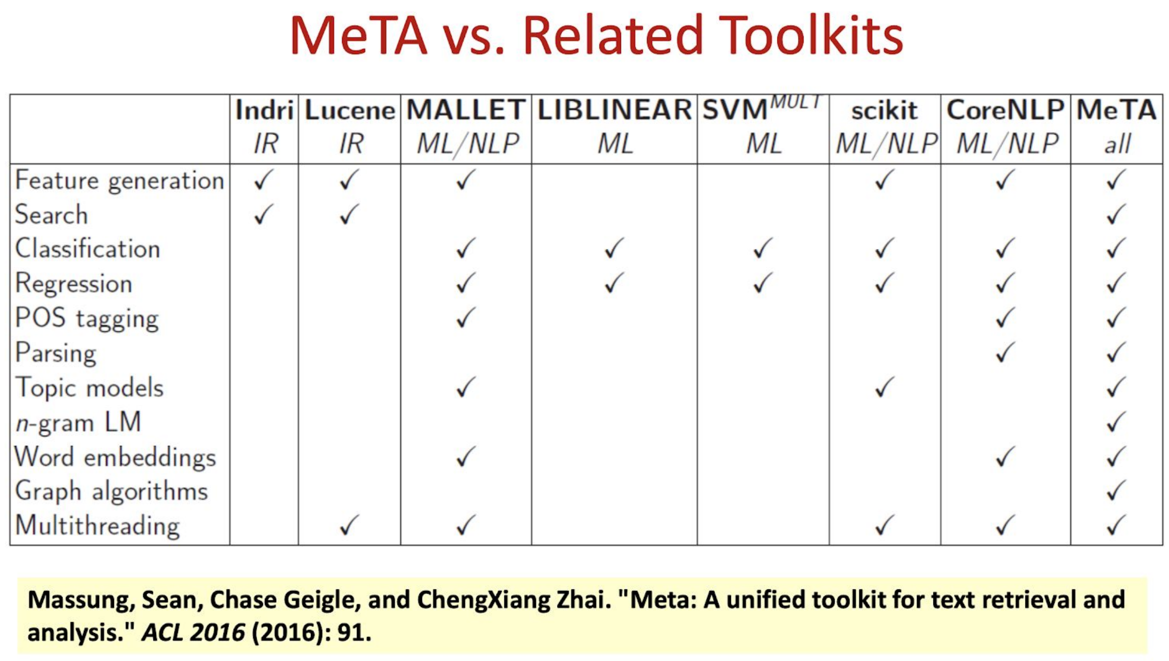
MeTa Toolkit / MeTAPy

Overview

MeTA (ModErn Text Analysis) toolkit was built by the team in University of Illinois Urbana-Champaign with the main motivation of facilitating education and research in Text Information System. Their tutorial covers most of the applied features that are discussed in UIUC CS410. The tool has a homepage: https://meta-toolkit.org/.

MeTAPy is the Python binding of MeTA, with most information could be found here: https://github.com/meta-toolkit/metapy/. There is also a demo provided separately in https://github.com/meta-toolkit/metapy-demos.

As found in https://meta-toolkit.org/data/kdd17-tutorial-short.pdf, MeTA toolkit covers full support of basic text data retrieval and analysis.



MeTA vs. Related Toolkits

	Indri	Lucene	MALLET	LIBLINEAR	SVM^{MULT}	scikit	CoreNLP	MeTA
	IR	*IR*	*ML/NLP*	*ML*	*ML*	*ML/NLP*	*ML/NLP*	*all*
Feature generation	✓	✓	✓			✓	✓	✓
Search	✓	✓						✓
Classification			✓	✓	✓	✓	✓	✓
Regression			✓	✓	✓	✓	✓	✓
POS tagging			✓				✓	✓
Parsing							✓	✓
Topic models			✓			✓		✓
n-gram LM								✓
Word embeddings			✓				✓	✓
Graph algorithms								✓
Multithreading		✓	✓			✓	✓	✓

Massung, Sean, Chase Geigle, and ChengXiang Zhai. "Meta: A unified toolkit for text retrieval and analysis." *ACL 2016* (2016): 91.

What was I able to learn about the tool within a few hours?

Their tutorials in https://github.com/meta-toolkit/metapy/tree/master/tutorials focus on the applied topics covered in UIUC CS410 - Text Information System course. There are Jupyter notebooks provided with the topics below:

1. Analyzers, tokenizers, Filters
2. Search and IR Evaluation
3. Deeper Text Analysis
4. Document Classification
5. Topic Modeling
6. KDD 2017: Basically the summary of above topics
    a. Part 1: Feature Engineering for Text Data
    b. Part 2: Information Retrieval with MeTA
    c. Part 3: Document Classification with MeTA
    d. Part 4: Topic Modeling

I found these tutorials could be completed with minimum Python familiarity, though my experience and the fundamentals I learned from UIUC CS410 really helped to understand the flow. I followed through all the tutorials within less than 3 hours and was able to learn that

MeTAPy is a really great complementary to the UIUC CS410 course to actually apply what you have learned there.

What did I find cool about the tool?

MetaPy does a good job on accomplishing the almost all basic features learned in the UIUC CS410 course, aligned with what was marketed in

What did I find helpful for learning the tool?

Following through UIUC CS410 and understanding the basic concepts really helped me. The tutorials structure was organized well with the same order as what you will learn in UIUC CS410. The Jupyter notebooks are written well with succinct explanations throughout the steps. Some of the MPs in UIUC CS410 made us implement some of the features from scratch, and I find what I have learned there has helped me to understand the tools easier.

What did I wish the tool had to help me learn better?

I also wished the MeTA toolkit had more demos with detailed logic walkthrough explanations.
The demo found in https://github.com/meta-toolkit/metapy-demos or the demos in https://meta-toolkit.org/ is not new learners-friendly.
For example, it is written that to run the demo you need to run the command

```
python search_server.py config.toml
```

However, config.toml is not provided in the repository. As someone who is new to the tool and new to Text Information System, config.toml is not very intuitively learnable even after reading https://meta-toolkit.org/search-tutorial.html. A simple starter config.toml should be helpful to help us learners get going.

Some demo links should also be updated. The demos with http://timan103.cs.illinois.edu/ domain could not be accessed at the time this Tech Review is written (November 2020).

I always find a good demo or codelab of the project is important to serve as a good “first impression” to smoothen the learning experience for new learners. It seems for now, learning the whole thing one-by-one through the MeTA toolkit tutorials in https://meta-toolkit.org/ plus knowledge in C++ would be the best way to really understand the framework.

What did I wish CS410 had to help me learn these tools better?

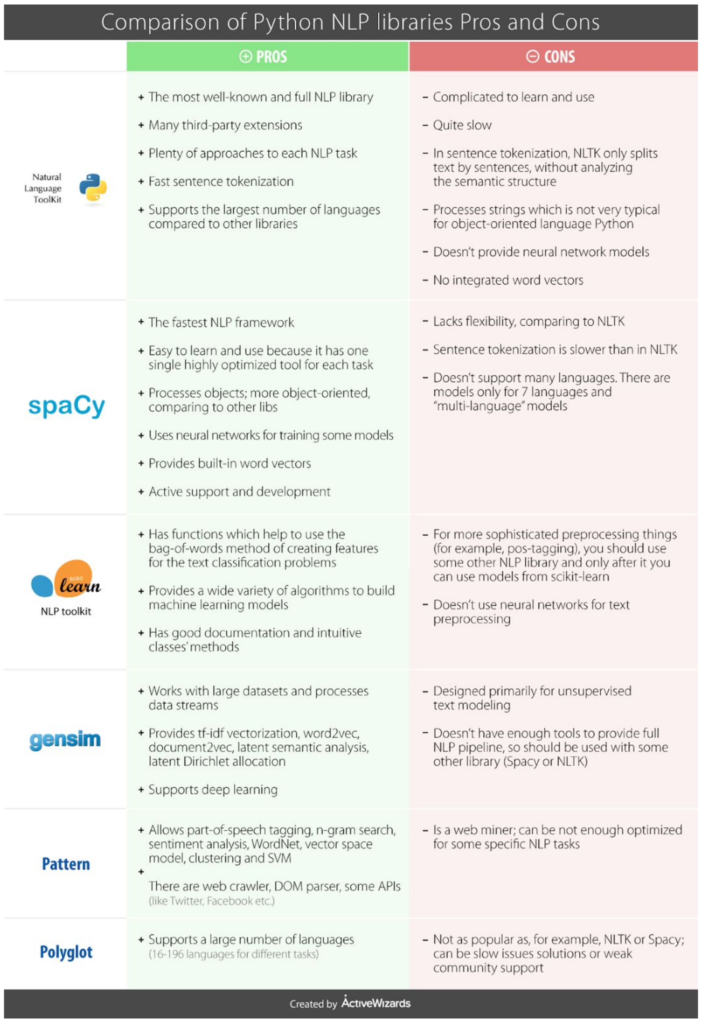
I wished the CS410 explicitly stated that MeTAPy is there to help us follow through the applicative side of the materials (maybe it has but I didn’t catch that) and explicitly ask us to go through the Jupyter notebook tutorials. A quick Jupyter notebook tutorial might help some students who were not familiar with .ipynb tutorials.

NLTK

Overview

NLTK (Natural Language Toolkit) is arguably the most popular Python library in the industry of NLP, developed since 2001. Its homepage, http://www.nltk.org/, has all of the information you need to know about NLTK.

NLTK contains full-fledged NLP features, though it is covered nicely by the below article that the tool may be most effective for academic / learning purposes due to its pros and cons: https://medium.com/activewizards-machine-learning-company/comparison-of-top-6-python-nlp-libraries-c4ce160237eb



Comparison of Python NLP libraries Pros and Cons

	⊕ PROS	⊖ CONS
Natural Language ToolKit	+ The most well-known and full NLP library + Many third-party extensions + Plenty of approaches to each NLP task + Fast sentence tokenization + Supports the largest number of languages compared to other libraries	- Complicated to learn and use - Quite slow - In sentence tokenization, NLTK only splits text by sentences, without analyzing the semantic structure - Processes strings which is not very typical for object-oriented language Python - Doesn't provide neural network models - No integrated word vectors
spaCy	+ The fastest NLP framework + Easy to learn and use because it has one single highly optimized tool for each task + Processes objects; more object-oriented, comparing to other libs + Uses neural networks for training some models + Provides built-in word vectors + Active support and development	- Lacks flexibility, comparing to NLTK - Sentence tokenization is slower than in NLTK - Doesn't support many languages. There are models only for 7 languages and "multi-language" models
scikit learn NLP toolkit	+ Has functions which help to use the bag-of-words method of creating features for the text classification problems + Provides a wide variety of algorithms to build machine learning models + Has good documentation and intuitive classes' methods	- For more sophisticated preprocessing things (for example, pos-tagging), you should use some other NLP library and only after it you can use models from scikit-learn - Doesn't use neural networks for text preprocessing
gensim	+ Works with large datasets and processes data streams + Provides tf-idf vectorization, word2vec, document2vec, latent semantic analysis, latent Dirichlet allocation + Supports deep learning	- Designed primarily for unsupervised text modeling - Doesn't have enough tools to provide full NLP pipeline, so should be used with some other library (Spacy or NLTK)
Pattern	+ Allows part-of-speech tagging, n-gram search, sentiment analysis, WordNet, vector space model, clustering and SVM + There are web crawler, DOM parser, some APIs (like Twitter, Facebook etc.)	- Is a web miner; can be not enough optimized for some specific NLP tasks
Polyglot	+ Supports a large number of languages (16-196 languages for different tasks)	- Not as popular as, for example, NLTK or Spacy; can be slow issues solutions or weak community support

Created by ActiveWizards

What could I learn about the tool in a few hours?

Its book, http://www.nltk.org/book/, is an ultimate resource you could read to learn from zero about the tool. The book assumes the reader does not know anything about Python and NLP, so I had a great experience learning the tool without problem considering my near zero experience in Python.

The book covers the below topics:

1. Language Processing and Python
2. Accessing Text Corpora and Lexical Resources
3. Processing Raw Text
4. Writing Structured Programs
5. Categorizing and Tagging Words
6. Learning to Classify Text
7. Extracting Information from Text
8. Analyzing Sentence Structure
9. Building Feature Based Grammars
10. Analyzing the Meaning of Sentences
11. Managing Linguistic Data

It took me more than 3 hours to read and digest these topics as the tool is NLP-heavy compared to what I have learned in UIUC CS410. However, since NLTK is widely used in the industry, it is easy to find other learning resources when a more practical explanation is needed.

What did I find cool about the tool?

Needless to say, NLTK is a strong tool for NLP tasks and covers comprehensive features that some tools may not have. The tool also has been around since 2001, have tons of tutorials out there, and known to be widely used in the industry. These reasons made me feel that NLTK is the safe choice as the tool to deep-dive in if I want to embark into the NLP industry.

What did I find helpful for learning the tool?

The plethora of learning resources really gave me a great experience on learning the tool. There are many examples out there that give you the image of what NLTK is capable of AND kindly covers step-by-step how to achieve that.

I also found their official book easy to read and helped me to understand what it can do without having to actually do the codelab by myself. I am particularly amazed by its Python-newbie friendliness that saves me from going back and forth trying to understand what the written syntax actually does.

Even though CS410 does not focus on the NLP part, I found that the knowledge I learned from the course had helped me to easily navigate on NLTK and look for what I need to develop a

Text Information System by myself. I also found that learning MeTAPy as instructed by the course had prepared me to know what to expect with the NLTK library.

What did I wish the tool had to help me learn better?

So far I am satisfied with the availability of tutorials and demos, either the official one or the ones that came from its community.

What did I wish CS410 had to help me learn these tools better?

I wished the course had a session where the Professor introduces the students with the popular tools in the industry and discusses their pros and cons. I guess the tech review was provided for that purpose, but an opinion directly from the Professor would be interesting to hear.

TensorFlow NLP

Overview

TensorFlow is a popular end-to-end open source machine learning platform. It is widely used and has a great community. TensorFlow is developed by Google Brain and being used internally in Google, released as open source since 2015. Most information could be found on its homepage, https://www.tensorflow.org/. As a complete machine learning platform, TensorFlow provides machine learning tools for NLP purposes.

What was I able to learn about the tool within a few hours?

TensorFlow organizes their text processing tutorials in https://www.tensorflow.org/tutorials with the structure below:

1. Word embeddings
2. Word2Vec
3. Text classification with an RNN
4. Text generation with an RNN
5. Neural machine translation with attention
6. Image captioning
7. Transformer model for language understanding
8. Fine tuning BERT

However, I found that their new “NLP Zero to Hero” series: https://goo.gle/nlp-z2h is easier to digest. It is meant to introduce TensorFlow NLP to learners with no experience required, structured as:

1. Natural Language Processing - Tokenization
2. Sequencing - Turning sentences into data
3. Training a model to recognize sentiment in text

4. ML with Recurrent Neural Networks
5. Long Short-Term for NLP
6. Training an AI to create poetry

Trying to learn this in a few hours convinces me that TensorFlow is indeed a machine learning platform that could be used for advanced tasks, including for NLP purposes for that matter. Although, I found that learning TensorFlow NLP may be a quite steep learning curve if you don't approach it starting with learning the basic TensorFlow and its Keras API.

What did I find cool about the tool?

It is a strong machine learning platform and its NLP support focuses on the applied NLP using machine learning. The official tutorial showcases how effortless it could be to achieve what was thought to be hard (i.e. sentiment analysis).

I found it very cool that the TensorFlow is developed by Google Brain and being used internally in Google. From that factor, you may bet that even without much community involvement, the TensorFlow may be developed in awesomely unthinkable ways accordingly with how Google Brain grows.

What did I find helpful for learning the tool?

TensorFlow NLP is used widely and has many tutorials and demos covered by the community. The official tutorial is followable though might be a bit challenging for new learners as there are many advanced terminologies involved which I found may be overwhelming.

I found the machine learning for NLP introduction covered in UIUC CS410 helpful to navigate throughout the TensorFlow learning process.

What did I wish the tool had to help me learn better?

The tutorial expects you to take the tutorials from the beginning, where you get familiar with the TensorFlow basics first before playing around with TensorFlow NLP. I was hoping to be able to find a TensorFlow NLP quick start that does not need you to start from the general TensorFlow.

The tutorials also assumed you are familiar with Machine Learning terms, which I wished they had helped give us new learners a chance with pointers or concise explanations in the tutorial.

What did I wish CS410 had to help me learn these tools better?

Same as mentioned in the NLTK section, I wished the course had a session where the Professor introduces the students with the popular tools in the industry and discusses their pros and cons.

Conclusion: a suggestion to new learners

The UIUC CS410 is a great course that has been organized well to prepare you to navigate through the world of Text Information System and NLP. I found what I learned so far makes the learning process of new tools--at least the ones that are covered in this tech review--more effective. I would suggest the new learners finish through all the materials in the course first before diving through the toolkit (i.e, for tech review purposes). I found it helpful to know the fundamentals on Information Retrieval, Information Extraction, and Natural Language Processing beforehand to handle these tools and understand what it may be used for.

Python seems to be the language to master if you want to thrive in the industry, considering most tools are well-supported on Python. Though later, C++ may still be the first choice when performance becomes a concern.

So far I found linear algebra is the only statistical knowledge prerequisite for understanding the terms in the discussed tools. Though of course doing a refresh in calculus would really help smoothen the learning process.

After conducting this tech review, my personal one sentence takes for each toolkit would be:

- MeTA is indeed a great complement for the CS410 course that helps students consolidate their understanding.
- NLTK is a great tool with an abundance of learning resources to accomplish comprehensive NLP tasks.
- TensorFlow NLP gives you flexibility on cool machine learning experimentation for NLP purposes.