Technology Review: Apache OpenNLP
CS 410 Text Information Systems
University of Illinois Urbana-Champaign
Seth Cattanach (sethac3) | October 2022

---

Among open-source toolkits in the field of natural language processing (NLP), Apache's OpenNLP project is a robust and widely supported option, particularly for multilingual applications. First released as an open-source software project in 2004 and adopted as an incubator project by the Apache Foundation in 2010[1], OpenNLP is primarily written in Java – although several wrappers extend its core functionalities to Python, as well (these wrappers are not officially supported by the Apache Foundation). As with other open-source projects, OpenNLP has widespread popularity because it is free to use, reliably maintained by both members of the Apache Foundation and the open-source community, and because it supports a wide range natural language processing applications such as tokenization, part-of-speech tagging, stemming and lemmatization, and more. OpenNLP is an alternative to other common toolkits such as NLTK, Spark NLP, and spaCy, although it struggles by comparison with more recent and advanced areas of natural language research such as deep learning.

OpenNLP is especially prevalent in applications requiring a Named Entity Recognition (NER) component. It implements a "Name Finding Training" API[2] that can be used with pre-trained models or alternatively, used to train a custom model (for common languages such as English and Spanish, the pre-trained models are wide ranging and are likely to cover the user's need; in other languages, or in technical and specific domains such as biotechnology, custom models are commonly trained to reflect unique entity sets). One of OpenNLP's biggest advantages is its multilingual support (it is frequently cited in academic research involving multilingual corpora, such as a 2016 study on robust NER techniques across five languages[3]: Basque, Dutch, German, English, and Spanish). And under the Apache 2.0 software license, it powers many commercial applications, such as Air New Zealand's "Oscar" autonomous chatbot[4].

One of the strongest criticisms of OpenNLP is its slow adaptation to more recent developments in natural language processing research, particularly its lack of support for deep learning frameworks. Part of the challenge is related to OpenNLP's underlying implementation. It was initially developed in Java, and therefore lacks natural compatibility with many deep learning frameworks developed in recent years, which are most commonly written in Python (by contrast, NLP toolkits that are more rapidly growing in popularity, such as spaCy and Spark NLP, natively support TensorFlow and PyTorch, two of the most popular deep learning frameworks). In its most recent major release, however (version 2.0.0),

---

[1] https://opennlp.apache.org/news/
[2] https://opennlp.apache.org/docs/1.9.3/manual/opennlp.html#tools.namefind.training
[3] R. Agerri, G. Rigau. *Robust multilingual Named Entity Recognition with shallow semi-supervised features, Artificial Intelligence* (2016). http://dx.doi.org/10.1016/j.artint.2016.05.003
[4] *Powered by Apache OpenNLP*. Apache Foundation. (February 2017). https://opennlp.apache.org/powered-by-opennlp.html

OpenNLP introduced support for the Open Neural Network Exchange (ONNX) runtime[5], which allows OpenNLP to call models trained using PyTorch, TensorFlow, and other ONNX-compatible deep learning frameworks. While this still does not allow deep learning model training to be conducted natively via OpenNLP APIs, it does introduce support for a format of neural networks that is rapidly growing in popularity as an intended universal standard. In contrast to OpenNLP (and in fact, most natural language toolkits), ONNX – released in 2017[6] – is relatively new. Should the popularity of ONNX grow such that the majority of neural networks developed are ONNX-compatible, this integration may effectively mitigate one of the most significant threats to future of OpenNLP.

Consideration should be given to popular OpenNLP alternatives such as NLTK, spaCy, and Spark NLP when deciding on appropriate open-source packages to use in a text processing application. Of these alternatives, OpenNLP is most closely related to the Natural Language Toolkit (NLTK), as both were released in the early 2000s[7] and feature robust support for "traditional" natural language processing techniques (such as N-gram language models). However, while OpenNLP was developed using Java, NLTK was written in Python, which has ultimately become the more popular language in natural language and machine learning applications. Similar to OpenNLP, NLTK supports a broad range of natural language processing tasks such as part-of-speech tagging, tokenization, stemming and lemmatization, language models (e.g., N-grams), and more. It also suffers from a relative lack of native integration with popular deep learning frameworks (e.g., PyTorch), but unlike OpenNLP, NLTK's Python implementation makes it a more suitable complement to these frameworks.

By contrast, the spaCy and Spark NLP projects have a comparative lack of support for traditional NLP methods (although both still support extremely common tasks such as part-of-speech tagging, tokenization, and stop-word removal), but feature deeper integrations with more novel advances in deep learning as well as unique features not found in more traditional natural language libraries like OpenNLP. Spark, for example, supports native Optical Character Recognition (OCR)[8]. While OCR is more of a computer vision task than a natural language task, its output – text strings – makes it a valuable complement for many natural language processing applications. As such, Spark NLP may be thought of as a more complete end-to-end library with broad support for all components of a text software application, whereas OpenNLP (and NLTK) are more traditionally used for specific tasks (such as tokenization or part-of-speech tagging). Other libraries are thus more commonly used alongside OpenNLP in large software applications. And the spaCy project, itself a fairly recent development (released in 2015[9]), positions itself as more of an open-source alternative for commercial, rather than academic, use[10]. This stands in contrast with both OpenNLP and NLTK, which are used more heavily in academic environments. The spaCy library features native integrations with TensorFlow and PyTorch through its custom back-end library (called "Thinc"), which means applications can implement custom

---

[5] https://opennlp.apache.org/docs/2.0.0/manual/opennlp.html#intro.models.onnx
[6] S. Shah (October 2017). *Microsoft and Facebook's open AI ecosystem gains more support*. Engadget. https://www.engadget.com/2017-10-11-microsoft-facebooks-ai-onxx-partners.html
[7] https://github.com/nltk/nltk/wiki/FAQ
[8] https://nlp.johnsnowlabs.com/docs/en/ocr
[9] M. Honnibal (February 2015). *Introducing spaCy*. Explosion AI. https://explosion.ai/blog/introducing-spacy
[10] https://spacy.io/usage/spacy-101

deep learning models such as CNNs and RNNs or utilize pre-trained models for common tasks in common languages (such as Named Entity Recognition in over 20 languages). These native integrations are comparatively lacking in the OpenNLP project and highlight why some alternatives, such as Spark NLP and spaCy, have more recently emerged as popular NLP libraries.

Though recent academic developments in natural language processing have de-emphasized the more traditional language modeling approaches implemented in the OpenNLP project, its robust and widespread support for many essential natural language tasks (and in many languages) has made Apache's OpenNLP library a widely used framework in the field of computational linguistics. Used in both academic and commercial applications under Apache's open-source license, OpenNLP remains under active development in 2022, and its newly released features improve its integration with popular deep learning frameworks using the ONNX neural network standard. And even as further research advances inevitably reprioritize which features are most desirable in the field of computational linguistics, OpenNLP's robust, multilingual support for essential NLP tasks such as tokenization and part-of-speech tagging suggests it will remain a popular framework in the years to come.