Python Toolkits for Text Classification

Michael Bencsik
bencsik2@illinois.edu
10/31/2022

Introduction

The purpose of this paper is to compare and contrast three popular text data and natural language processing (NLP) toolkits, TensorFlow (TF), PyTorch, and the Natural Language ToolKit (NLTK), from the perspective of someone who has never worked extensively with Machine Learning (ML) toolkits. These toolkits utilize Python 3 as one of the programming languages to implement their various libraries. This paper will cover an introduction to, base requirements for, and ease of first time use for each of the three toolkits. Any implementation and testing will be performed on a Windows 11 operating system (OS) using Python 3.10.8.

TensorFlow

TensorFlow is an Open Source Machine Learning framework that is currently on the stable version of 2.10. It was developed by Google in 2015 and is still maintained by Google (Google TFX Team). TF falls under the Apache License Version 2.0. This means that source code itself is allowed to be modified, distributed, used privately and commercially, and allows contributors to file for patents based on their contributing work as long as the license is attached to any use of the source code, other contributors are given credit for previous work, a statement is made for any significant modifications to the source code, and that the original copyright notices are included (ASF). In addition to the previously mentioned obligations, the contributors are not held responsible for the end use of the software and are not required to provide any warranty for the software.

TF was mainly written for Python and has an API stability guarantee for Python and C. The TF toolkit does contain a larger list of other language bindings such as C++, Java, Go, JavaScript, and more, giving TF more flexibility than other ML toolkits. These other bindings, however, are not maintained by Google and encourage the community to support these bindings with a recommended approach (Google Developers). Currently, TF supports Python version 3.7 through 3.10 and will work on any major, up to date OS with Python and the C++ Redistribution package (Windows only) including Windows, Linux, and M.acOS.

There are some watchouts when choosing which OS and language to use when first attempting to use TF. First, V2.10 is the last supported version for Windows Native OS. Future support of Windows is only available through Windows Subsystem for Linux or WSL2 (Google Developers). WSL2 is a compatibility layer that allows Linux systems to run on Windows without the need for dual booting or a Virtual Machine. Second, Python is now on version 3.11.0. This version was just released Oct 24, 2022 . This is not a major concern as TF will soon catch up, but it is worth noting since TF will fail to install if this version of Python is used. Third, TF encourages the use of its GPU support for building and training ML models. This is a great feature for GPUs that support this functionality, however, lots of warnings and information messages are displayed whenever the toolkit is imported without enabling GPU support. These messages cannot be disabled without disabling all TF warnings. These messages can lead to confusion and suggest that there may be issues with the toolkit. Lastly, TF recommends using Anaconda's package manager Conda to install TF's GPU support features. The Anaconda Environment is not ideal if the long term goal is to use TF in a corporate setting due to licensing fees and restrictions (Anaconda). For students and hobbyists there are no licensing fees. To avoid these fees, TF can be installed and updated manually through Python's package manager, pip. The GPU features can be installed manually through the supported GPU manufacturer's website.

TF provides installation instructions for each OS and type of installation method. In addition to the core functionality of TF, TensorFlow Text needs to be installed for text and NLP processing support ("tensorflow/text: Making text a first-class citizen in TensorFlow."). TF Text is an additional library which adds text processing features such as tokenization, n-grams, and white space parsing features to TF's existing functionality.

There is extensive documentation for TF which includes API definitions for each library, numerous datasets which are managed by a dataset library, pre-built ML models, and detailed examples of working with TF in different aspects such as text, audio, visual and more. All of these resources are available and organized for different subsets of TF such as text and NLP ("Tutorials").

PyTorch

PyTorch was originally developed from the Torch library by the Facebook AI Research lab in 2016, it is now maintained by The Linux Foundation (Moltzau). The current stable version at the time of this writing is 1.13.0 and allows nightly build versions to be downloaded for experimental features. The source code is open source, using a 3-Clause BSD license. This also allows the source code to be modified, distributed, used privately and commercially, as long as the license is attached to any use of the source code, the original copyright notices are included, and the original contributors and project names are not used for endorsing the derivatives. This license also includes a similar liability and warranty exclusion disclaimer as Apache 2.0 ("Terms").

PyTorch was written with a "Python First" mentality to ensure the best user experience when used with Python's inherit functionalities (PyTorch). This allows PyTorch to be used in conjunction with other popular data libraries such as NumPy, Pandas, and Scikit. There are also two language bindings for C++ and Java, which are maintained by PyTorch. Similar to TF, PyTorch only supports Python 3.7 through 3.10, and will run on any major OS such as Windows, Linux or Mac OS as long as the Python and/or C++ capabilities are available ("pytorch").

The main watchout for PyTorch is that it also recommends using the Anaconda environment to take advantage of its features and package manager, Conda ("pytorch"). Be aware that Anaconda's license is not as open as PyTorch itself in that if used commercially for companies over 200 employees, licensing fees are required and use is restricted to licensed computers (Anaconda). If the intended use is for personal learning, the recommended installation using the Conda package installer will provide the PyTorch toolkit along with any additional dependencies. If the long term intention is to integrate Pytorch into a company's toolkits, then Python's pip package manager would be a better option to avoid extra licensing fees. This requires the user to manually install any dependencies, such as NumPy. For text and NLP, torchtext is an additional library that provides text and NLP models, transforms, vocab tools, and datasets ("pytorch/text").

PyTorch provides vast amounts of documentation on API definitions, examples, and datasets to learn and understand basic and advanced functionalities (“PyTorch documentation — PyTorch 1.13 documentation”). The documentation is also split into different ML subsets such as text, audio, and visual libraries.

NLTK

NLTK stands for Natural Language Toolkit and was originally written by Steven Bird and Edward Loper in 2001. The toolkit was built to help students learn NLP while allowing for the toolkit to be easily maintained by the creators (“FAQ · NLTK”). The current stable version of NLTK is 3.7, which drops support for Python 3.6. NLTK also falls under the same permissive license as TF, the Apache License Version 2.0 (“FAQ · NLTK”). NLTK was developed and exclusively written for Python. It currently supports Python version 3.7 through 3.10 and will run on Windows, Linux or MacOs as long as a required Python version is available (“Installing NLTK”).

There were no major watchouts when installing NLTK. Unlike TF and PyTorch, NLTK does not recommend Anaconda, but Python’s pip package manager to install.

Since NLTK was written for text and NLP, there is not an additional text library to install, but there is an additional downloader that allows for the corpora, grammars, and pre-built models to be acquired after installing the core toolkit. This requires additional setup to the pip package manager (NLTK).

NLTK provides the API definition documentation and examples (“NLTK package”), but is not as extensive or in depth as TF or PyTorch. The datasets are also presented in a long list without filtering or sorting capabilities. The examples provided are much shorter and easier to understand for those who are unfamiliar with ML or NLP.

Comparison of First Use

Overall, TF seemed to contain more information, documentation, and extended features than PyTorch or NLTK. For a beginner, it was hard to know where to start and left an overwhelming feeling of drinking through a firehose. For the installation, TF was installed using Python’s pip package manager into a clean Python 3.10.8 virtual environment using the built-in venv module (Reitz). After learning that the warning messages will always occur and that nothing was actually wrong with the installation, I followed two text classification examples (“Classify text with BERT | Text”). The examples were easy enough to follow, but once I tried to deviate from the examples I ran into many issues. The main TF data pipeline, tf.data.Dataset, was cumbersome and not intuitive to use. Data had to be loaded in a specific manner. It was much easier to use NumPy or Pandas to load and manipulate the dataset prior to using TF, than to use the built-in data features of TF. Overall, it seems that TF has a lot of advantages and advanced features. However, a toolkit this large will take time and constant use to be able to be used effectively.

PyTorch seems to be more narrowly focused on directing its usage as a ML toolkit. Its installation into the same Python virtual environment was much smoother and lacked any errors. The example for text classification was also simple to follow (PyTorch). Like TF, PyTorch uses tensors, which are similar to NumPy ndarrays, but did not have an extra dataset type. This allowed the manipulation of data to be much easier than TF. Overall, I found PyTorch to be more intuitive to work with than TF.

NLTK was the simplest toolkit to use for text and NLP. The toolkit uses Python’s built-in data types, and the examples use other Python modules to manipulate data where needed. The example followed for text classification was simpler and shorter than TF or PyTorch (NLTK). Overall, the package is limited to only text and NLP, but it was the easiest of the three to accomplish the task of creating a text classifier from start to finish. This was mostly due to the limited information and lack of distractions from being overwhelmed with which path to take.

Summary

Using these three toolkits was a great learning experience. For a beginner it was easy to see that NLTK really helps break the user into text and NLP learning in a simple manner. It also has the most limiting features, supported languages, and no GPU execution support. PyTorch is the next step up to add more ML features outside of NLP, two more available languages, more functionally from the available module methods, and supports GPU execution. Lastly, TF is the most advanced toolkit, increasing the number of support languages significantly, and provides more examples, datasets, and user contributed resources. The major downfalls to TF is the amount of dedication required and the future loss of Windows Native support.

Works Cited

Anaconda. “Contracting Hub.” 17 January 2022, https://legal.anaconda.com/policies/en/?name=contracting-hub#purchased-vs-free-offerings. Accessed 2 November 2022.

ASF. “Apache License, Version 2.0.” *The Apache Software Foundation!*, 2004, https://www.apache.org/licenses/LICENSE-2.0. Accessed 3 November 2022.

“Classify text with BERT | Text.” *TensorFlow*, 29 March 2022, https://www.tensorflow.org/text/tutorials/classify_text_with_bert. Accessed 4 November 2022.

“FAQ · NLTK.” *GitHub*, 7 April 2022, https://github.com/nltk/nltk/wiki/FAQ. Accessed 2 November 2022.

Google Developers. “Install TensorFlow with pip.” *TensorFlow*, 7 10 22, https://www.tensorflow.org/install/pip. Accessed 3 November 2022.

Google Developers. “TensorFlow version compatibility.” *TensorFlow*, 5 November 2021, https://www.tensorflow.org/guide/versions. Accessed 3 November 2022.

Google TFX Team. “Towards ML Engineering: A Brief History Of TensorFlow Extended (TFX).” *The TensorFlow Blog*, 25 September 2020, https://blog.tensorflow.org/2020/09/brief-history-of-tensorflow-extended-tfx.html. Accessed 31 October 2022.

“Installing NLTK.” *NLTK*, https://www.nltk.org/install.html. Accessed 2 November 2022.

Moltzau, Alex. “PyTorch Governance and History. PyTorch's design philosophy and… | by Alex Moltzau | Medium.” *Alex Moltzau*, 7 August 2020, https://alexmoltzau.medium.com/pytorch-governance-and-history-2e5889b79dc1. Accessed 2 November 2022.

NLTK. “Installing NLTK Data.” NLTK, https://www.nltk.org/data.html. Accessed 4 November 2022.

NLTK. “Sample usage for semantics.” *NLTK*, https://www.nltk.org/howto/semantics.html. Accessed 4 November 2022.

“NLTK package.” *NLTK*, https://www.nltk.org/api/nltk.html. Accessed 2 November 2022.

“pytorch.” *GitHub*, https://github.com/pytorch/pytorch#prerequisites. Accessed 2 November 2022.

PyTorch. “Text classification with the torchtext library — PyTorch Tutorials 1.13.0+cu117 documentation.” *PyTorch*, https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html. Accessed 4 November 2022.

PyTorch. “PyTorch Design Philosophy — PyTorch 1.13 documentation.” *PyTorch*, https://pytorch.org/docs/stable/community/design.html#principle-3-python-first-with-best-in-class-language-interoperability. Accessed 4 November 2022.

“PyTorch documentation — PyTorch 1.13 documentation.” *PyTorch*, https://pytorch.org/docs/stable/index.html. Accessed 2 November 2022.

“pytorch/text.” *GitHub*, https://github.com/pytorch/text. Accessed 2 November 2022.

Reitz, Kenneth. “12. Virtual Environments and Packages — Python 3.11.0 documentation.” *Python Docs*, https://docs.python.org/3/tutorial/venv.html#creating-virtual-environments. Accessed 4 November 2022.

“tensorflow/text: Making text a first-class citizen in TensorFlow.” *GitHub*, 19 September 2022, https://github.com/tensorflow/text#installation. Accessed 3 November 2022.

“Terms.” *Linux Foundation*, https://www.linuxfoundation.org/legal/terms. Accessed 2 November 2022.

“Tutorials.” *TensorFlow*, 15 September 2022, https://www.tensorflow.org/tutorials. Accessed 3 November 2022.