TensorFlow and Natural Language Processing

Zhiyan Jiang

1. Introduction

Natural Language Processing (NLP) is a specific domain of artificial intelligence, focusing on making machines understand human languages. It has a wide application in modern daily life. For instance, Virtual assistants are mainly NLP systems, such as Google Assistant, Cortana, and Apple Siri. NLP has fast growth in the recent years, primarily due to the benefit of the improvements to the language model architectures and the training on large text corpora. Regardless of the huge advancements, NLP remains challenging because of the highly nonlinear relationship between words and semantics, which is difficult to capture using robust numerical representations. In addition, each language has unique vocabulary, grammar, and syntax. It is difficult to find a universal solution.

There are two approaches for NLP problems in general – traditional approach and deep models. The traditional approach to NLP, also known as a statistical approach, is a sequential flow of key steps including preprocessing, feature engineering, learning, and prediction. However, this approach has several drawbacks. First, while preprocessing can reduce the number of features, it also removes useful information such as punctuation and tense information. Second, feature engineering for statistical analysis needs to be performed manually and less scalable. In addition, domain knowledge may be required. Last but not least, essential external resources are not always freely available. For example, representing queries by vectors requires a large vocabulary dataset, which may not be available. On the other hand, deep models learn rich features from raw data instead of using limited human-engineered features, and performs feature engineering and task learning simultaneously. Deep learning models can encompass significantly more features than human. Typical deep models include Convolution Neural Network (CNN), Recurrent Neutral Network (RNN), and Long Short-Term-Memory (LSTM) networks. CNN can learn from two-dimensional data without loss of spatial information, and RNN allows the model to exhibit temporal dynamic behavior, such as sequence of texts. LSTM is an extension of RNN to incorporate long-term memory.

TensorFlow is an emerging tool to facilitate application of neural networks on NLP problems. Its value has been proved by numerous successful cases. For example, Twitter implemented TensorFlow to rank tweets by importance for a given user, instead of in the reverse chronological order. E-commerce platform Carousell used TensorFlow to provide recommendations for customers. The purpose of this technology review is to investigate what TensorFlow is, and how it helps solve NLP problems.

2. Natural Language Processing tasks

The tasks of NLP are summarized as below:

- Tokenization: divide a text body into atomic units, such as words.
- Part-of-Speech (PoS) tagging: assign words to their corresponding role in a corpus.
- Word-sense disambiguation (WSD): identify the meaning of words.
- Text classification: classify a text fragment into one of several predefined classes.
- Sentiment analysis: label text in a dataset based on the degree positivity.
- Named Entity Recognition (NER): extract entities from text, such as dates, addresses, people names.
- Keyword extraction: find the most meaningful words within a text. The words are used as tags.
- Text summarization: extract the most meaningful group of words within a text.

- Question Answering (QA): given a text fragment and a question, extract a piece of text as an answer.
- Language generation: After trained with text corpora, predict the subsequent text.
- Machine translation (MT): transform an expression from a source language to a target language.

The first eight tasks are categorized as “Analysis” tasks, while the remaining three are “Generation” tasks. It is noted that multiple NLP tasks may be coordinated to build a single system.

3. Introduction of TensorFlow and its application on NLP

TensorFlow is a free and open-source software library designed for machine learning and artificial intelligence. The name derives from tensors, i.e., multidimensional data arrays. With applicability on multiple tasks, it is particularly designed for establishing deep neural networks. TensorFlow was first developed by the Google Brain team for internal research and production, and became open public since 2015. The updated version 2.0 was released in 2019. TensorFlow has been successfully deployed in business.

TensorFlow has a major interface with Python. Other programming languages of JavaScript, C++, and Java are also supported. In addition, it has language binding packages for C#, Haskell, MATLAB, R, Scale, etc. TensorFlow is available on mainstream operating systems, such as 64-bit Linux, macOS, Windows, online notebook (Colaboratory), and even mobile platforms (using TFLite). TensorFlow is compatible with Numpy data structures (NDarrays), which provides an interface with existing mathematic libraries. In terms of hardware, TensorFlow can operate on CPUs, GPUs, and tensor-processing units (TPUs).

Typical features of TensorFlow include:

- Automatic differentiation: automation of calculation of the gradient with respect to each parameter.
- Eager execution: TensorFlow has an “eager execution” mode, in which operations are evaluated immediately.
- Distributed: TensorFlow provides APIs for distributed computation across multiple devices.
- Losses: TensorFlow provides a set of loss functions, such as mean squared error and binary cross entropy.
- Metrics: common metrics for performance evaluation are provided, including accuracy, precision, recall, among others.
- Optimizers: TensorFlow offers a set of optimizers for training neutral networks, such as ADAM, ADAGRAD, and Stochastic Gradient Descent.

Among the numerous modules, the following ones have direct relation with NLP:

- TensorFlow Extended (TFX). This module enables end-to-end production. Its components include loading, validating, and transforming data, tuning, training, and evaluation.
- TensorFlow Recommenders (TFRS) – a library for building recommendation system models. It helps build and evaluate flexible recommendation retrieval models, incorporate item, user, and context information into recommendation models, and train multi-task models that jointly optimize multiple recommendation objects.
- TensorFlow Graphics – a library providing graphic functions to visualize machine learning models.
- TensorFlow Cloud – facilitate integration of local code/modeling into Google Cloud

- TensorFlow Model Optimization – a suite of tools for optimizing ML models through reducing latency and inference cost for cloud and edge devices, deploying models to edge devices with restrictions, and enabling execution on existing hardware or accelerators
- TensorFlow Probability (TFP) – A library about probabilistic models and deep learning. It includes a rich selection of probability distributions and bijectors, tools to build probabilistic models, variational inference and Markov chain Monte Carlo, and optimizers such as Nelder-Mead, BFGS, and SGLD.
- TensorFlow Decision Forests (TF-DF) – A library to train, execute, and interpret tree-based models, such as Random Forests and Gradient Boosted Trees.
- TensorFlow Hub - a repository of trained ML models for direct deployment or further tunning.

4. Conclusions

Natural Language Processing is an active and challenging domain. TensorFlow is an artificial intelligence tool that can utilize neural networks to fulfill the NLP tasks effectively.

5. References

1. TensorFlow official website (www.tensorflow.org), accessed on Oct.26, 2022.
2. G, Thushan. Natural Language Processing with TensorFlow. Packt Publishing, 2018. ISBN: 1-78847-831-2.
3. Microsoft Learn, Introduction to natural language processing with TensorFlow (https://learn.microsoft.com/en-us/training/modules/intro-natural-language-processing-tensorflow/), accessed on Oct. 26, 2022.