Chapter 10
High Performance Computing for Understanding Natural Language

Marija Stanojevic
Temple University, USA

Jumanah Alshehri
https://orcid.org/0000-0002-0077-7173
Temple University, USA

Zoran Obradovic
Temple University, USA

ABSTRACT

The amount of user-generated text available online is growing at an ever-increasing rate due to tremendous progress in enlarging inexpensive storage capacity, processing capabilities, and the popularity of online outlets and social networks. Learning language representation and solving tasks in an end-to-end manner, without a need for human-expert feature extraction and creation, has made models more accurate and much more complicated in the number of parameters, requiring parallelized and distributed resources high-performance computing or cloud. This chapter gives an overview of state-of-the-art natural language processing problems, algorithms, models, and libraries. Parallelized and distributed ways to solve text understanding, representation, and classification tasks are also discussed. Additionally, the importance of high-performance computing for natural language processing applications is illustrated by showing details of a few specific applications that use pre-training or self-supervised learning on large amounts of data in text understanding.

DOI: 10.4018/978-1-7998-7156-9.ch010

INTRODUCTION

The exponential data explosion requires developing practical tools for efficient and accurate pattern discovery, classification, representation, trend, and anomaly detection in large-scale high dimensional textual data (Szalay & Gray, 2006). For a decade now, IBM has been using high-performance computing (HPC) to analyze text and create intelligent machines. IBM Watson is a supercomputer that famously leveraged language analysis to win a game of Jeopardy (Hemsoth, 2011).

Advances in natural language processing (NLP) are essential for achieving real artificial intelligence. Language is considered one of the most complex human inventions and essential to human intelligence and social integration. Therefore, success in NLP is a prerequisite for fully functioning, artificially intelligent machines.

The industry is currently the largest contributor to NLP development because of its practical importance in handling large amounts of unstructured online data. Understanding public opinion through user-generated text analysis guides more informed decisions, policies, and products. Due to increased use of online social networks, forums, blogs, product reviews, and news comments, it became easy to collect an extensive amount of text needed for understanding opinions and facts about specific topics. Being able to understand those texts fully can shape politics, marketing, and many other fields.

As natural language models have become more complex in recent years, usage of HPC locally or in the cloud has become inevitable in NLP applications. Most novel NLP models are based on neural networks, which forward and backward propagation can be reduced to a vast matrix (tensor) multiplication. Therefore, Graphics Processing Unit (GPU) or Tensors Processing Unit (TPU) hardware is used for faster training. To enhance those models' speed and usability, they are mostly implemented in a distributed manner and expected to run on a high-performance parallel computing system.

Some popular libraries used in implementing and evaluating the most recent natural language models are: NLTK (Loper & Bird, 2002), Gensim (Rehurek & Sojka, 2010), SpaCy (SpaCy, 2020), TensorFlow (Abadi et al., 2016), PyTorch (Paszke et al., 2019), Keras (Chollet, 2017), scikit-learn (Pedregosa, 2011) and all of them support parallel and distributed processing, while most support GPU, and some even run on TPU hardware. Many of those frameworks are easy to learn and have complex neural networks and machine learning modules readily available for use. For those practitioners wanting to create and parallelize their algorithms in python, there is an open-source library, Dask (Dask Development Team, 2019), that natively scales python code. Also, Google has recently developed JAX (Google, 2020), which can transform any python code to allow backpropagation through it. This framework allows an additional training speed up by an innovative combination of operations and simple transformation *pmap,* making the algorithm parallelizable and easy to execute on HPC.