Environmental Awareness in the Arctic, Team 3

Sai Charan Akkena

Project Progress Report - II Experiment

SECTION 1

Abstract

This project aims to build a model that enables researchers to make quantitative arguments about environmental awareness and cultural change. The dataset has been taken from reddit which has the data from local news of the region for which we are analyzing the environmental changes and also from scholarly articles. We'll be using an open-sourced neural network-based technique for Natural Language Processing called BERT (Bidirectional Encoder Representations from Transformers). BERT has been pre-trained on the whole English Wikipedia and can be further fine-tuned on smaller task-specific datasets.

1. Data

1.1. Data Source

This dataset was collected from reddit, searching for articles that contain the keyword 'humanities' . The data is in the format of compressed tar.gz file which has 118,059 files inside it in the format of .txt and has an actual size of 322MB .To ensure that the program doesn't run out of memory, we will train our dataset in multiple batches.

1.2. Data Format

This data is comprised of .txt files which are in natural text language and processed with the Illinois Wikifier. The datasets are all in English except one which is in Norwegian. The data in the text files is comprised of various types of words and sentences, sentences with open and close parenthesis in the beginning and at the end, words in both wikified and non-wikified formats etc.

1.3. Programming Language, packages

Python will be used for fine-tuning and testing the BERT model on Google Colaboratory which allows us to write and execute arbitrary python code through the browser. Google Colab is a Jupyter Notebook environment that doesn't require any setup and runs entirely

on cloud. Since the dataset is huge, we'll be using our local machine's GPU and if required we'll switch to Google Cloud TPU in the future. We will be using many machine learning libraries like PyTorch, Pandas, TensorFlow, Transformers by hugging face etc.

2. Experiments

2.1. Preliminary Results

Since our data is in .txt files which contains different paragraphs with various sentences and has already been pre-processed, we combined all the .txt files into a .csv format and modified it into a .tsv file which is understandable by BERT. We performed tokenization using the Bert Tokenizer using Google Colab on our local GPU. The data has been loaded into a pre-trained BERT Model and a fine-tuned model based on our Vocabulary has been built. The synonymy test has been applied by me to the BERT by finding out the word embeddings and comparing the vector values to find out the cosine similarity between two words.

SECTION 2

1 Data statistics

Since our dataset has extracted from reddit searching for the word 'humanities', and it has already been pre-processed and wikified using the Illinois Wikifier, the wiki words are in the form of 'wiki__XXX__<wikifier_end_parts>'. These words have been separated from the dataset by removing the Wikifier Information at the beginning and also at the end and the frequency of the words has been found out. The words 'arts', 'of', 'liberal','humanities','university','social' etc., are the most frequently occurring words in the dataset whereas the words 'damn', 'reichsbank' etc., have the least occurrence in the dataset.

2. Experiments

2.1 Selected Approaches

We shall be using the Hugging Face's transformer for implementing our BERT model. It is a PyTorch implementation of Google's BERT implementation. We decided to use it due to several advantages it offers us the best results we aim for. It has the same accuracy rate as Google's BERT implementation. There are many pretrained models shared by many people which will allow us to improve our accuracy and precision while not needing us to pre-train a newmodel. It has many functions already implemented which will make our task easy.

Our approach is to infer data from the wiki words and find which words can really have an impact on training the model. So we will be extracting the wiki words, cleaning those words (i.e., removing Wikifier information at the beginning and at the end), finding the frequency of each unique words and making 2 lists from those cleaned words (List 1 – single words after removing Wikifier Information and List 2 – everything else after removing Wikifier Information), comparing the lists with the existing 'BERT vocabulary' to find the matching words from our lists, analyzing the words (i.e., identifying names of cities, universities, geographic locations, etc.,)

2.2 Implementation

This approach helps us in training BERT while finding the frequency of the words after removing Wikifier Information. It will help us decide which words are important and which are not important to the BERT Vocabulary. For example, let us consider two different sets of words with frequency greater than 100 and the other with frequency less than 5. Of these two sets, we can infer that the words which are occurring more times will be more useful to the BERT than the words which occur less time, these words which occur less time shouldn't be omitted and also be added. As our project focuses on environmental awareness using text mining, it is necessary for the BERT to analyze the text in the best way, for which data with a higher number of contexts will help.

We used hashlib package in the process of removing duplicate words after removing Wikifier Information. We used nltk package to find the frequency of the wiki words in the file. After comparing both the lists with the BERT vocabulary, analyzing the matched words from lists we used spacy package's NER (Named Entity Recognition) to identify the category of the words.

For tokenization, since we are using Hugging Face's transformers library, we will be using the provided Bert Tokenizer. It is based upon Google's Word Piece tokenizer. It already has traditional tokenization features such as lower casing which allows us to skip the pre-tokenization of the dataset. We will assign the maximum sequence length of 12 since this was the longest word length in our data. We then convert tokens to the index numbers corresponding in the BERT vocabulary.

2.3 Approach

We have several tasks to get the data which will help BERT in-text analyzing efficiently. Frist, we counted the total number of words in the entire reddit dataset and extracted all the wiki words from the dataset and placed them in a separate text file (wiki_word_dataset.txt). Then we have counted the total number of wiki words from the dataset. After this we have cleaned the wiki words, i.e. removing Wikifier Information at the beginning and end of each wiki word. After removing the Wikifier Information we will be having words that can be helpful in training the model. But we can't train BERT with all the words we got, as this might end up underfitting the model. To overcome this we found the frequency of every word after removing Wikifier Information. From the output of this task, we could see that occurrence of words is in a range from 1 to 50,000+. Among these words, words that occur more than 50 times in this dataset will have a different type of contexts that will eventually help BERT in analyzing more about that word, whereas words with occurrence less than 30 to 50 times will considerably have different contexts, this can provide more information about the word and its usage, but the words which occurred less than 20 times in this huge dataset is not a big issue, as training BERT on these words will narrow the analyzing of BERT which results in less efficiency (i.e., the error rate may be higher because of these words). After this step, we have

removed the duplicates from the cleaned words file. We then matched these unique words file with the existing BERT vocabulary (which has word count more than 30,000) and wrote them in a new separate file, so that we can know the words that need to be added into the BERT vocabulary. We then made two lists from the unique words file, List 1 will contain all the single word after removing Wikifier Information and List 2 will contain all the other set of words. We then compared these two lists with BERT vocabulary and generated two separate files. We then analyzed these two files using the spacy package, from this analysis we figured out the category each word in both of the files would fall into, this package identified the word's type (for example names of people, nationalities,cities, facilities, etc.,)

Tasks	Task Description	Student 1 name	Student 2 name	Student 3 name	Student 4 name
Task 1	Extracting the Wiki Words	Suriya Prakaash	Sai Charan		
Task 2	Cleaning the extracted word		Sai Charan		Sai Vishal
Task 3	Finding frequency	Suriya Prakaash		Siddhant Jain	
Task 4	Removal of duplicates and Comparison with Bert Vocab			Siddhant Jain	Sai Vishal
Task 5	Separating Single and Multiple Words	Suriya Prakaash			Sai Vishal
Task 6	Named Entity Recognition using Spacy		Sai Charan	Siddhant Jain	

I performed the following tasks in the project so far:

1) Combining all the text files into a format understandable by BERT and then loading the dataset set to a pre-trained BERT Model(uncased) and fine-tune the model based on our Vocabulary.

2) Perform Synonymy test on the word embeddings from the dataset and the BERT Vocabulary

3) Extracting the Wiki words, cleaning the wiki words using python by removing the Wikifier Information from the beginning and at the end and performing the text analysis on the words which are common between the reddit data set and the BERT Vocabulary.