Homework 4: Probing BERT's Contextual Word Representations

Due February 24, 2021 (11:59 PM)

1 Introduction

In this homework, we will be working with pre-trained contextual word representations using BERT (Bidirectional Encoder Representations from Transformers), a highly influential NLP model introduced by Devlin et al. in 2018[1]. BERT and friends (e.g. ELMo[2] or Camembert[3]) use large text datasets along with unsupervised training objectives like masked word prediction to develop representations of words as they are used *in context*. These pre-trained representations have proved to be powerful features, outperforming other methods according to a number of different NLP benchmarks.

BERT's surprising effectiveness at NLP tasks like those in the commonly used GLUE benchmarks have led many people to ask: Does BERT really understand language? This homework explores one way of interrogating what BERT understands through *Probing* a pre-trained BERT model.

In the process, we'll learn how to use the popular Transformers Python library for working with pre-trained NLP models. To help you get started, this assignment's Colab notebook includes an already-implemented example of a BERT-based model for classification using the ACL topics data you annotated in Homework 1. Note that this model gets a higher accuracy score on this data then either the CNN or logistic regression models that we tried in previous homeworks: BERT tops out here with an accuracy of around 0.63 on the dev data.

The Colab notebook for this homework is located here: `https://github.com/dbamman/nlp21/blob/main/HW4/HW_4.ipynb`

2 Probing for 'Object Number'

Probing experiments in NLP involve constructing controlled cases to ask questions about what kinds of information has been encoded in internal representations learned by models like BERT. Probes are typically implemented by experimenting with simple models (we'll use logistic regression) that use pre-trained BERT representations as input; these models are trained to make classifications based on those inputs. The types of classification tasks and datasets used for probing are often designed to focus on specific linguistic phenomena.

In this assignment, we'll implement a probe to explore how well the parameters in a pre-trained BERT can do on a classification problem called "Object Number". The task is to answer the following question about a sentence S: 'Is the direct object of the main verb in sentence S singular or plural?' For example, the sentence 'I took the bus' has a singular direct object (bus), while the sentence 'she sells seashells' has a plural direct object (seashells).

[1] https://arxiv.org/abs/1810.04805
[2] https://arxiv.org/abs/1802.05365
[3] https://arxiv.org/pdf/1911.03894

This probe gives us one way of understanding how much information BERT's pre-training process has managed to encode about object numbers. In addition to probing for *how much* BERT has learned about this task, we will also probe for *which layers* have encoded the most information about object numbers.

Your job in this homework is to implement a probe for object number: for each of the 12 layers in BERT, train a classifier to predict whether the direct object of the main verb in an input sentence is a singular or plural noun. For more on this probe, see Conneau et al. (2018) Probing sentence embeddings for linguistic properties, and for more information on probing in general, see Tenney et al. (2019) BERT Rediscovers the Classical NLP Pipeline.

3 Deliverables

3.1 Deliverable 1: Complete BertLayerClassifier

A class called BertLayerClassifier has been provided for you - this class is very similar to the example BERTClassifier shown at the beginning of the Colab. Your first deliverable is to complete the indicated section of the 'forward' function in BertLayerClassifier in order to adapt this class to be able to work with any of BERT's 12 layers (as opposed to just using the last layer).

Note that when using BERT to make sentence-level classifications, the '[CLS]' token is often treated as a representation of the entire sentence. Your classifier for layer L should use only the '[CLS]' token representation from layer L for making a prediction. More information on the '[CLS]' token can be found in the **'tips'** section of the Colab.

3.2 Deliverable 2: Build classifiers for Object Number using each of BERT's 12 layers

Using your completed BertLayerClassifier, the rest of this assignment is to complete the probing experiment by implementing the following within the 'runProbes' function:

1. Create a classifier for each of BERT's 12 layers.
2. Train each classifier for 5 epochs using the provided training data for Object Number. You may want to reference the example training code used for classification with BERT earlier in the Colab.
3. Evaluate the accuracy of each classifier using the provided dev data for Object Number.

Your 'runProbes' function should return a dev accuracy for each BERT layer. You may define extra 'helper' functions that you call within 'runProbes', but please make sure that **all** of your code is written within either the 'runProbes' function or within your helper functions. Do **not** put any of your final code **outside** of functions, or we won't be able to import your code during grading. Refer to the Colab for more details and for implementation tips!

4 How to Submit

1. Download your Colab notebook as an .ipynb file (File → Download .ipynb)
2. Run the last cell provided in the Colab in order to save your dev accuracies for each BERT layer to the file 'devAccuracies.txt'.
3. download 'devAccuracies.txt'.
4. Submit 2 files, **HW_4.ipynb** and **devAccuracies.txt**, to Homework4 on Gradescope. The files must be named in this way for the Gradescope autograder.