Overview of BERT and its related models or applications on Text Classification

Introduction

Bert can be used for various NLP tasks such as text classification or sentence classification, semantic similarity between sentence pairs, question answering tasks with paragraphs, text summarization, etc. However, due to some bidirectional information retrieval features, we cannot use Bert, such as machine translation, text generator, etc. This is because we need to get information in both directions, and we need to fine-tune the BERT model for our task,

Implementation of Text Classification

First we get BERT up and running, which requires us to create a virtual environment, or we can use conda, we need a text classification dataset, we use the Yelp comment polarity dataset, and it is revealed that we use pandas to process data in csv format, Converting the data to a BERT-friendly data type, The final step before fine-tuning is to convert the data into features that BERT uses. Most of the remaining code was adapted from the Hugging Face example run_classifier.py, found here. In the first cell, we are importing the necessary packages. In the next cell, we are setting some paths for where files should be stored and where certain files can be found. We are also setting some configuration options for the BERT model. Finally, we will create the directories if they do not already exist.

Next, we will use our BinaryClassificationProcessor to load in the data, and get everything ready for the tokenization step.

Here, we are creating our BinaryClassificationProcessor and using it to load in the train examples. Then, we are setting some variables that we'll use while training the model. Next, we are loading the pretrained tokenizer by BERT. In this case, we'll be using the bert-base-cased model.

Once all the examples are converted into features, we can pickle them to disk for safekeeping (I, for one, do not want to run the processing for another one and a half hours). Next time, you can just unpickle the file to get the list of features.

Conclusion

BERT is an incredibly powerful language representation model that shows great promise in a wide variety of NLP tasks. Here, I've tried to give a basic guide to how you might use it for binary text classification.As the results show, BERT is a very effective tool for binary text classification, not to mention all the other tasks it has already been used for.

It should be noted that BERT has a large number of parameters and requires high computing resources. Model training requires a lot of time and cost. To speed up model training, other existing embedding models such as GloVe etc. can be used at the cost of accuracy. Furthermore, the use of BERT is not limited to text or sentence classification but can also be applied to advanced natural language processing applications such as next sentence prediction, question answering or named entity recognition tasks.

References:

Aryanshu Verma . (2018, June 12). Application of BERT: Binary Text Classification Retrieved November 6, 2022, from https://iq.opengenus.org/binary-text-classification-bert/

Thilina Rajapakse. (2019, June 9). A Simple Guide On Using BERT for Binary Text Classification. Retrieved November 6, 2022, from https://medium.com/swlh/a-simple-guide-on-using-bert-for-text-classification-bbf041ac8d04

Saumyab (2021 Dec 31). Text Classification using BERT and TensorFlow. Retrieved November 6, 2022, from https://www.analyticsvidhya.com/blog/2021/12/text-classification-using-bert-and-tensorflow/

Santiago González-Carvajal, Eduardo C. Garrido-Merchán (2020, May 26). Comparing BERT against traditional machine learning text classification Retrieved November 6, 2022, from https://arxiv.org/abs/2005.13012

Sumanth Prabhu, Moosa Mohamed, Hemant Misra(2021, Apr 27). Multi-class Text Classification using BERT-based Active Learning Retrieved November 6, 2022, from
https://arxiv.org/abs/2104.14289