BERT Overview and Proposal of Data Augmentation

1 Introduction

BERT[1], standing for Bidirectional Encoder Representations from Transformers, is a transformer-based approach for language model pre-training proposed by Google AI Language. It uses the popular self-attention language model Transformer as its core, conducting bidirectional training on two main tasks: masked language modeling (MLM) and next sentence prediction (NSP). With bidirectional representations of input text, BERT can be fine-tuned to design models for large variety of language tasks, especially tasks that need the context knowledge to make decisions such as dialog state tracking and question answering tasks.

By adding a layer to BERT and utilizing different datasets can create various state-of-the-art models. However, some tasks which are not popular may do not have abundant data especially labeled ones. This review will give an overview and explanation of Transformer language model and BERT first, then introduce an initial proposal of a possible data augmentation approach for BERT training.

2 Overview and explanation

2.1 Self-attention and Transformer

Transformer is a new popular deep learning model introduced in 2017 by Google[2]. Unlike the traditional RNN (Recurrent Neural Networks) which processes word one by one in sequence, Transformer process sentences as a whole. The transformer consists of an encoder and decoder. As BERT only apply the encoder, we mainly talk about the encoder here.

Self-attention mechanism is the core of Transformer language model. In terms that easy to understand, this mechanism enables the input text to find relationships with each other and decide which one needs more "attention". For text data, the relationship talked here has nothing to do with proximity, it has to do with linguistic meaning, for example, the relationship between "She" and "Amy" in the sentence "Amy like pink color so she wants this dress". Self-attention allows the Transformer encoder takes each token's word embedding vector and output the "better" encoding of this word with context information. To do this, every layer of encoder has three main vectors including a Query vector, a Key vector and a Value vector that are trained and used to re-weight the input vectors.

2.2 BERT

BERT applies the encoder of the Transformer to read the entire sequence of words in sentences at once, the attention mechanism brought by the encoder allows BERT to learn context, therefore it is considered bidirectional. There are two main pre-training tasks in BERT.

The first is Masked LM. In this task, 15% of the words in a sequence will be masked with a [MASK] token before fed into the encoder, and the training purpose of model is to predict the masked words using context. This trick help train a deeply bidirectional model because it forces the model to learn from both left and right position to guess the masked tokens. However, using too much [Mask] tokens will cause problem when training fine-tuning tasks where no masked token needed. So BERT replace 80% words with [mask] token, and replace others with other words to cope with this problem. The second task is Next Sentence Prediction. To make BERT more proper for fine-tuning tasks like Question Answer and Natural Language Inference, NSP is added to enhance the model's ability to understand the relationship between sentences. A sentence embedding and Transformer positional embedding are added into basic token embeddings to help model distinguish the different sentences.

3 Data Augmentation Proposal

With weights and model generated from the unsupervised pre-training, BERT model can be rapidly fine-tuned on a specific downstream task. Usually there will be relatively less computation because the model already learned a lot from pre-training tasks. However, many downstream tasks heavily need human effort to annotate data for training and evaluation. Thus, data augmentation approaches are needed to solve the data dependency problems.

A method called Local Additivity based Data Augmentation (LADA)[3] was proposed to augment training data in fine-tuning BERT to conduct Name Entity Recognition (NER) task. NER[5] can classify different words in one sentence into different semantic categories (such as person, location, time), with which one can extract useful information and knowledge from text and use them for more fine-grained tasks. LADA creates an infinite amount of labeled data by doing interpolation of hidden states generated by existing data. The overall architecture of LADA is shown in Figure 1, but in practice there are more details. To maintain the context information and to reduce the noises from unrelated sentences, two kinds of interpolation are designed: Intra-LADA and Inter-LADA. Intra-LADA interpolates each tokens' hidden representation within one sentence, while Inter-LADA sample sentences based on *k*-nearest neighbors algorithm and interpolate each token in the near sentences to the center sentence. Experiments shows that LADA is rather effective.

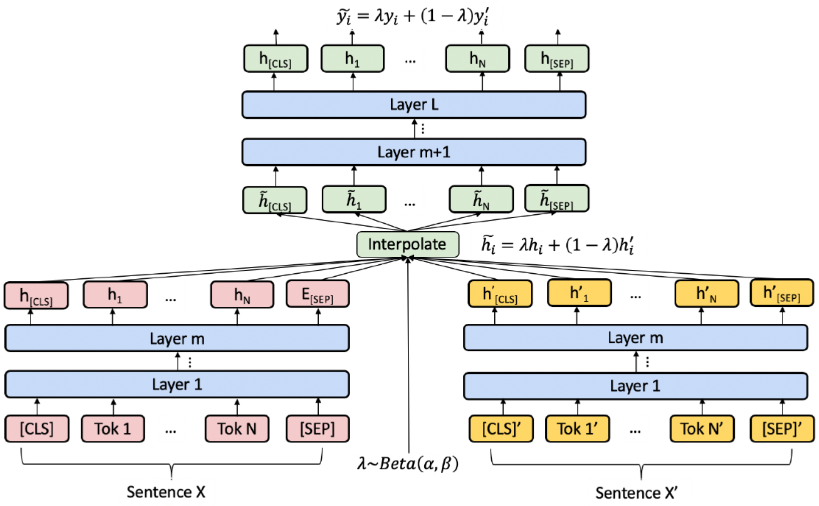

Figure 1: Overall Architecture of LADA

And from my perspective, the main concepts of LADA about data augmentation should be applied to other downstream tasks as well, and a better training result may be expected. I want to propose a data augmentation approach for Question Answer tasks.

Unlike NER tasks, which requires a classification layer that predicts the NER label for every token, the Question Answer tasks need two extra vectors denoted the beginning and the end position of the answer in a document. So instead of calculating interpolated representations of every input word, we should treat the question part and paragraph part differently. There may be two main ways to try. Firstly, it's maybe more proper to only generating pseudo labeled data for the question part. We interpolate each token's hidden representations with other tokens within one sentence but not change the representation of the answer part. On a high level it seems like we generated many similar questions with the same answer. The other way deserved trying is that we interpolate a question with other nearest questions, and in the meantime also interpolate their answers. Furthermore, I also have a hypothesis that maybe simply taking the embeddings directly from the pre-trained BERT model and do interpolations on these features is enough, because the huge number of parameters of this big model make the fine-tuning tasks computationally expensive.

4 Conclusion

BERT is no doubt one of the most important language models proposed these years, and many Natural Language Processing tasks are able to achieve state-of-the-art accuracy based on it. This review paper presents an overview and explanation of features of BERT and its underlying self-attention mechanism which is a very popular and essential mechanism in today's NLP research. And considering the necessity of data augmentation when fine-tuning BERT to different downstream tasks, I take some research about an existing data augmentation approach LADA for NER, then propose to take the intuition of LADA and modify it to be applied on other tasks such as Question Answer tasks. Although it's only my proposal without any experiments to verify, it's reasonable and worth to try in the future.

Reference

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
[2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[3] Chen, Jiaao, et al. "Local additivity based data augmentation for semi-supervised NER." arXiv preprint arXiv:2010.01677 (2020).