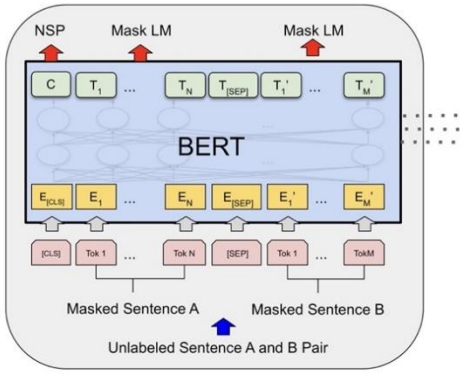
自然语言处理 PJ3 实验报告

1. 背景介绍

BERT 的全称为 Bidirectional Encoder Representation from Transformers，是由 Google 提出的用于 NLP 的预训练技术。它不像传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的 Masked Language Model（MLM）从而能生成深度的双向语言表征。

另外它是一种深度双向的、无监督的语言表示且仅使用纯文本语料库进行预训练的模型。但是如 word2vec 和 glove 是为词汇表中每个单词生成一个词向量表示，所以容易出现歧义问题。BERT 利用 MLM 进行预训练并且采用深层的双向 Transformer，这种 Transformer 其每一个 token 会 attend 到所有的 token 来构建整个模型，从而生成包含了所有上下文信息的深层双向语言表征。如下图，BERT 的主体结构是由多层 Transformer Encoder 堆叠而成。



MLM 是 BERT 能够不受单向语言模型所限制的原因就是选取 15%的 token 用 mask token 对训练序列中的 token 进行替换，然后预测出 mask 位置原有的单词。但是 mask 并不会出现在下游 fine-tuning 阶段，所以预训练阶段和微调阶段之间产生了不匹配。这个问题通过随机把语料库中 15%的单词做 mask 操作，mask 操作又包括了 80%的单词直接用 mask 替换、10%的单词直接替换成另一个新的单词、10%的单词保持不变来解决的。

Prompt 的目标是预测填充标签,使其与真实标签越接近越好。所以通过选取合适的 prompt，可以控制模型预测输出，从而一个完全无监督训练的 PLM 可以被用来解决各种各样的下游任务。

设计一个合适的 Prompt 是主要方法有 HandCraft Prompts、Continuous Prompts 和 Discrete Prompts 几类设计思路，另外基于 Prompt 的方法通常用于解决 Zero-Shot、Few-Shot 等训练样本不足的问题。

2. 实验内容

这次 PJ 使用了 BertForMaskedLM 模型实现，并且基于 OpenPrompt 实现了 Prompt Template、Verbalizer，使用了 Pytorch 作为模型框架。

1.Zero-Shot

这里以 HandCraft 的方式构建了 Prompt Template：

```
Validation accuracy: 0.510
```

在验证集上的准确率为 0.510。

2.Few-Shot Finetune

这里使用了预训练 BERT 模型，并且分别使用样本数量为 32/64/128 的训练集对模型进行了微调，在验证集上的准确率为：

train_32.txt:

```
Validation Accuracy: 0.422
```

Train_64.txt:

```
Validation Accuracy: 0.627
```

Train_128.txt:

```
Validation Accuracy: 0.773
```

3.Few-Shot Prompt

这里使用了样本数量为 32/64/128 的训练集对模型进行了基于 Prompt 的 Finetune，构建了 Mixed Template 在验证集上的准确为：

Train_32.txt:

```
Validation Accuracy: 0.610
```

train_64.txt:

```
Validation Accuracy: 0.830
```

Train_128.txt:

```
Validation Accuracy: 0.887
```

4. Template

这部分分别使用第一个实验中的 HandCraft Template 和第三个中的 Mixed Template 在 train_64.tsv 训练集上进行 Finetune，在验证集上的准确率为：

HandCraft:

```
Validation Accuracy: 0.631
```

Mixed:

```
Validation Accuracy: 0.830
```

5.Verbalizer

这里使用了 Mixed Template，然后分别使用 HandCraft Verbalizer 和 Soft Verbalizer 在 train_64.tsv 训练集上进行 Finetune，在验证集上的准确率为：

HandCraft:

```
Validation Accuracy: 0.794
```

Soft:

```
Validation Accuracy: 0.830
```

3. 总结

通过实验 2 和实验 3 的对比可得，基于 Prompt 的 Finetune 对比直接 Finetune 的性能有了显著提高，说明了 Prompt 确实有效。通过实验 4 可以发现，构建合适的 Prompt Template 对模型性能至关重要，HandCraft 是 Hard Prompt，相比 Soft Prompt 的 Mixed Template，后者能够学习到更多先验知识，因此有更好的性能。通过实验 5 可以发现，构建合适的 Verbalizer 对模型的性能也有影响，但是不如 Template 的影响大，同时，Soft Verbalizer 相比 Hard Coded 的 HandCraft Verbalizer 性能更好。

对于直接对预训练模型 Finetune 的方法，是在基于现有的观测数据对后验概率进行修正，而基于 Prompt 的 Finetune 是直接利用已有的概率模型对观测数据进行建模，因此能够直接学习到先验知识，然后根据贝叶斯公式，也就能够对后验概率进行修正所以能够有效。