**自然语言处理报告**

1. **训练模型的主要技术**

首先，第一个关键技术是 Transformer。它在NLP各个任务中都取得了优异的性能，它是预训练语言模型的核心网络。给定一句话或是一个段落作为输入，首先将输入序列中各个词转换为其对应的词向量，同时加上每一个词的位置向量，体现词在序列的位置。然后将这些词向量输入到多层Transformer网络中，通过自注意力（self-attention）机制来学习词与词之间的关系，编码其上下文信息，再通过一个前馈网络经过非线性变化，输出综合了上下文特征的各个词的向量表示。每一层Transformer网络主要由Multi-head self-attention层（多头自注意力机制）和前馈网络层两个子层构成。Multi-head self-attention会并行地执行多个不同参数的 self-attention，并将各个 self-attention的结果拼接作为后续网络的输入，self-attention机制会在后面中做详细介绍。此后，我们得到了蕴含当前上下文信息的各个词的表示，然后网络会将其输入到前馈网络层以计算非线性层次的特征。

在每一层Transformer网络中，会将残差连接（residual connection）把自注意力机制前或者前馈神经网络之前的向量引入进来，以增强自注意力机制或者前馈网络的输出结果向量。并且还做一个layer normalization，也就是通过归一化把同层的各个节点的多维向量映射到一个区间里面，这样各层节点的向量在一个区间里面。这两个操作加入在每个子层后，可更加平滑地训练深层次网络。

Transformer可以用于编码，也可以用于解码。所谓解码就是根据一个句子的输入得到一个预想的结果，比如机器翻译（输入源语言句子，输出目标语言句子），或者阅读理解（输入文档和问题，输出答案）。解码时，已经解码出来的词要做一个自注意力机制，之后和编码得到的隐状态的序列再做一个注意力机制。这样可以做 N 层，然后通过一个线性层映射到词表的大小的一个向量。每个向量代表一个词表词的输出可能性，经过一个softmax层得到每个词的输出概率。

然后是self-attention机制，以一个head作为示例。假定当前输入包含三个词，给定其输入词向量或是其上一层Transformer网络的输出，将其通过三组线性变换，转换得到三组queries、keys和values向量。Query和key向量用来计算两两词之间的得分，也就是其依赖关系，这个得分会同其对应的value向量做加权和，以得到每个词综合上下文信息的表示。给定当前第一个词的query向量，其首先同各个词的 key 向量通过点积操作得到这两个词的得分，这些得分用来表示这两个词的依赖或是相关程度。这些得分之后会根据query等向量的维度做一定比例的缩放，并将这些得分通过softmax操作做归一化。之后，各个得分会同其相对应的value向量相乘得到针对第一个词加权的各个value向量，这些加权的value向量最终相加以得到当前第一个词的上下文表示。

在得到第一个词的上下文表示后，给定第二个词的query向量，我们会重复之前的操作，计算当前query向量同各个词key向量的得分，对这些得分做 softmax归一化处理，并将这些得分同其对应的value向量做加权和，以得到其编码上下文信息的表示。

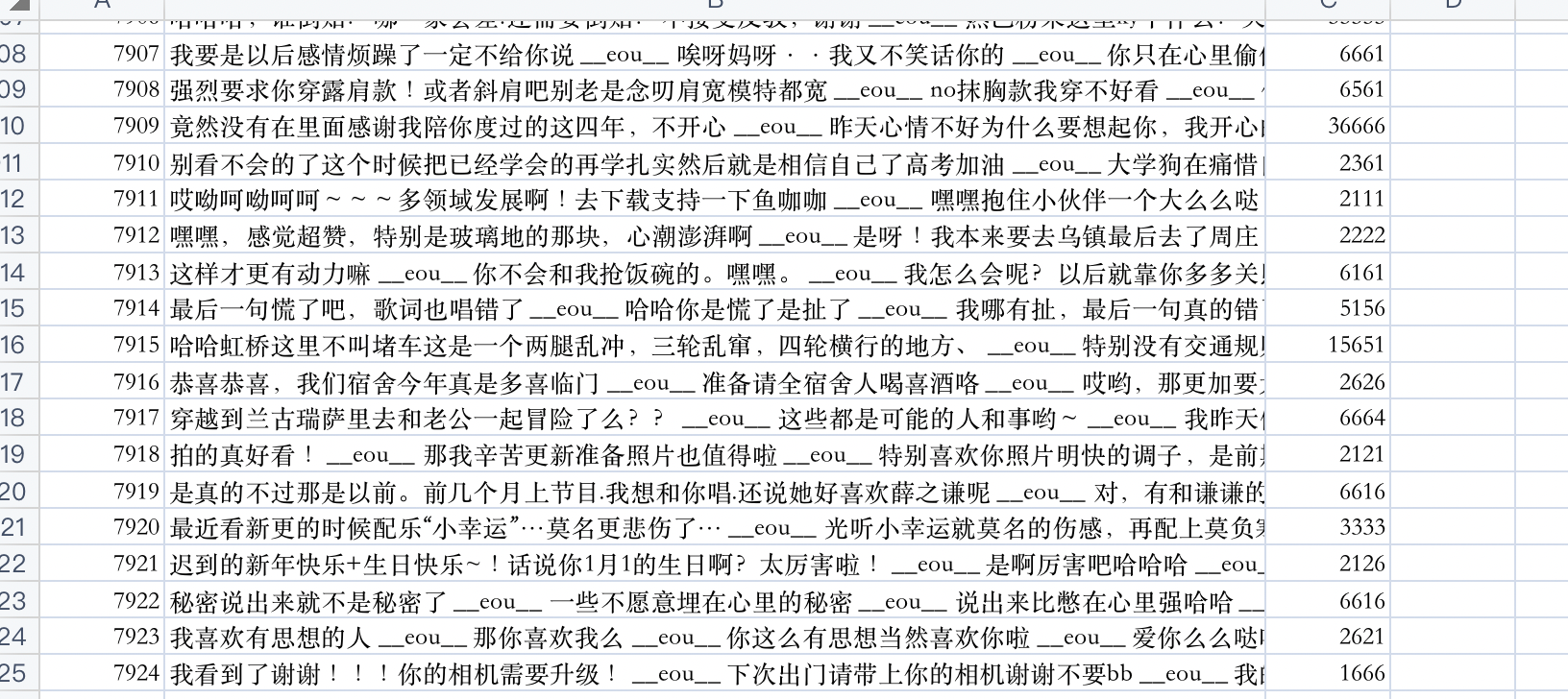
第二个关键技术是自监督学习。在预训练的模型中，AR（自回归）LM和AE（自动编码器）是最常用的自监督学习方法，其中，自回归LM旨在利用前面的词序列预测下个词的出现概率（语言模型）。自动编码器旨在对损坏的输入句子，比如遮掩了句子某个词、或者打乱了词序等，重建原始数据。通过这些自监督学习手段来学习单词的上下文相关表示。

第三个关键技术就是微调。在做具体任务时，微调旨在利用其标注样本对预训练网络的参数进行调整。以我们使用基于BERT（一种流行的预训练模型）为例来判断两个句子是否语义相同。输入是两个句子，经过BERT得到每个句子的对应编码表示，我们可以简单地用预训练模型的第一个隐节点预测分类标记判断两个句子是同义句子的概率，同时需要额外加一个线性层和softmax计算得到分类标签的分布。预测损失可以反传给BERT再对网络进行微调。当然也可以针对具体任务设计一个新网络，把预训练的结果作为其输入。

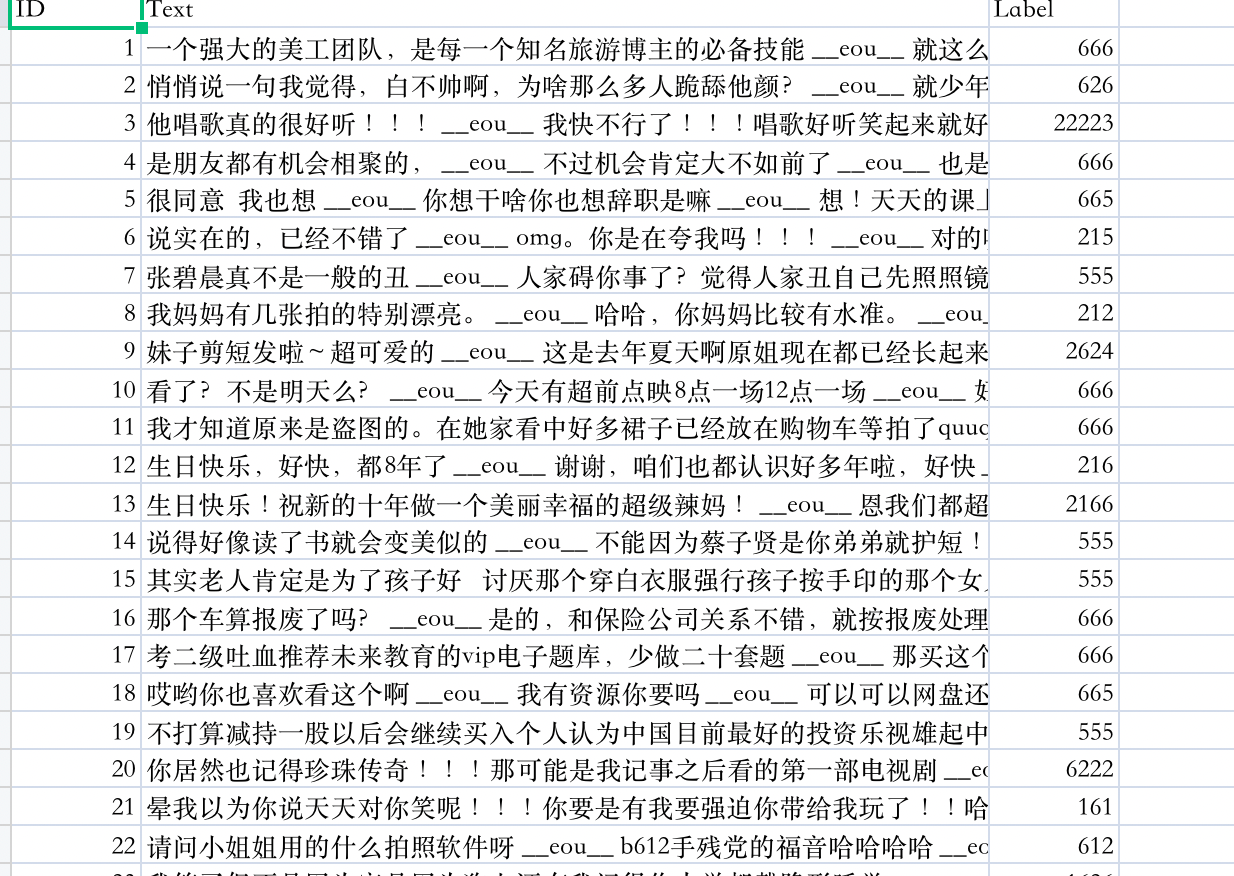
总体来讲，预训练模型发展趋势：第一，模型越来越大。比如Transformer的层数变化，从12层的Base模型到24层的Large模型。导致模型的参数越来越大，比如GPT 110 M，到GPT-2是1.5 Billion，图灵是17 Billion，而GPT-3达到了惊人的175 Billion。一般而言模型大了，其能力也会越来越强，但是训练代价确实非常大。第二，预训练方法也在不断增加，从自回归 LM，到自动编码的各种方法，以及各种多任务训练等。第三，还有从语言、多语言到多模态不断演进。最后就是模型压缩，使之能在实际应用中经济的使用，比如在手机端。这就涉及到知识蒸馏和teacher-student models，把大模型作为teacher，让一个小模型作为student来学习，接近大模型的能力，但是模型的参数减少很多。

1. **实践**

本次任务采用bert预训练模型进行训练，训练集如下所示，每行有N个句子，每个句子对应一个情绪，共有六种情绪。任务的目标是预测对话最后一句话的情绪，并在验证集上加以验证。



验证集



首先对数据集进行数据预处理

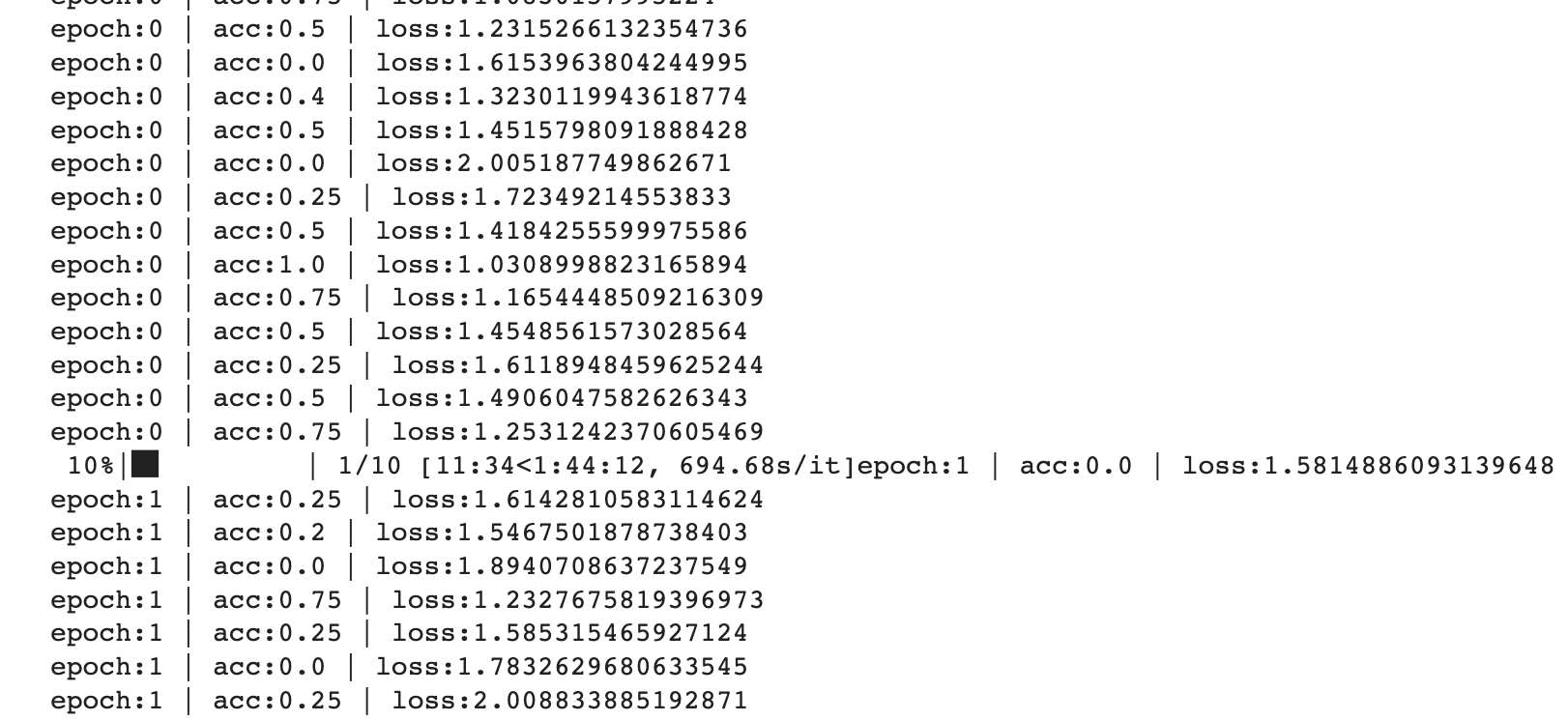


训练数据



测试及验证

  
共10个epoch，部分训练结果如下



将结果写成csv文件

