姓名: 林宇辰
学号: 2200013211

Homework 3

课程名称: FNLP
2024 年 6 月 16 日

1 Dense Retriever

在这个部分我实现了一个基于稠密特征提取器的判据匹配器。我们的目标是衡量两个文本的相似度，首先用预训练模型将两个文本编码成嵌入向量，再计算这两个向量的余弦相似度，算出来的值就是两个文本的相似度。

1.1 Implementation

我使用的预训练模型是从 sentence_transformers 库中调用的 all-MiniLM-L6-v2 模型。对于每一个样本，先计算出每个 candidate 的嵌入向量，再计算出 claim 的嵌入向量，和每个 candidate 的嵌入向量计算余弦相似度，然后朴素地取前 topk 个就得到结果（注意如果 candidate 没有 topk 个，那就把所有 candidate 都当成模型的输出）。

1.2 Evaluation

使用 Recall 指标作为评估，也就是计算选出来的 top5 个 claim 在真实 claim 中的比例。基于我的实现方法在 dev 数据集上的结果是 0.520。

2 Sparse Retriever

在这个阶段我实现了一个基于稀疏特征提取器的判据匹配器（主要是基于 BM25）。BM25 改进了 TF-IDF 的方法，对每个词引入饱和函数和文档长度因子，从而能更好地建模文本向量。

2.1 Implementation

首先用朴素的方法进行 tokenization，也就是用空格划分整个句子。对编码之后的 candidate 句子集，送入 rank_bm25 实现的 BM25 抽取器进行该样本抽取器的初始化。接着把计算 claim 的嵌入再计算余弦相似度并选择 topk。

2.2 Evaluation

在 dev 数据集上的结果是 0.55。分析：使用稀疏特征抽取器的结果比使用预训练的稠密特征抽取器还要好，这是违反直觉的，原则上预训练的模型应该能更好地建模文本信息，并且稠密向量与稀疏向量相比也能更好地表示信息。究其原因，我认为最重要的是我使用的预训练模型没有在当前任务上微调过，所以存在很多与任务目标无关的冗余信息被建模在嵌入向量中了，这干扰了余弦相似度的精确性；另一方面，稀疏特征提取器是逐样本的（sample-wise），我认为这能够使稀疏特征提取器更加集中地建模一个样本的信息，从而会有更好的结果。