隐式篇章关系判别实验报告

窦骥桐

北京理工大学计算机学院 学号:1120200670

2022 年 12 月 31 日

摘要

在此次隐式篇章关系判别任务中，我们基于预训练模型的替换将测试集 Macro F1 提升到 0.6418；之后基于 prompt 方法将 macro F1 又进一步提升到了 0.6672。之后我们对实验结果和数据集进行分析，尝试了使用显式篇章关系，集成学习，双序列模型等，并不能解决向'expansion' 类严重偏移的问题，macro F1 也没有进一步提高。

1 基线模型和基于预训练的 F1 提升

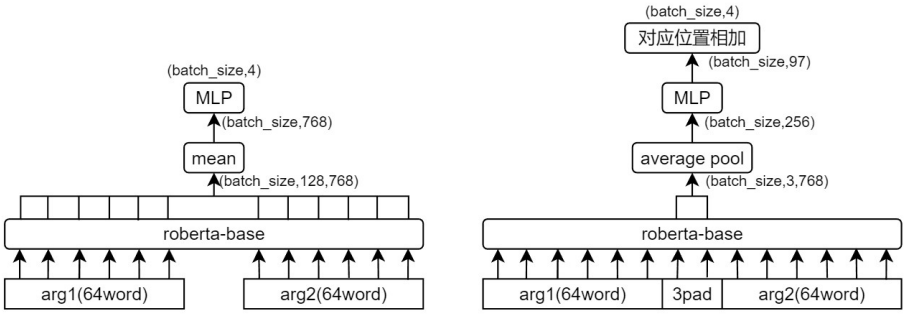

图 1: 基线模型左图；prompt 模型右图

模型结构如图 1 左图所示。我们直接对 arg1 和 arg2 进行拼接处理后输入预训练模型中得到最后一层的隐藏层表示，之后直接加权平均后接 MLP 分四类得到输出结果。

其中具体参数如下：arg1 和 arg2 截断到 64 单词长度，不足的进行补齐；预训练模型使用的是 roberta-base，当然如果使用更大的模型比如 roberta-large，效果会更好；MLP 使用两层，第一层的输入维度是 756，，输出维度是 256，后接 dropout 层；第二层输入维度是 256，输出维度是 4。实验结果如表 1 所示。

2 使用 prompt 方法的 F1 提升

2.1 数据预处理

因为我们想通过预测连接词判断样本属于哪种篇章关系，所以我们需要建立连接词和篇章关系的映射，幸运的是这种映射关系是可以被建立起来的。通过对测试数据的统计，我们发现，大多数连接词可以唯一的确定一种篇章关系；即使是那些可能得出不同篇章关系的连接词，他们也主要集中在一个篇章关系中。测试集总共有 1046 个样本，如果就将出现次数最多的篇章关系作为连接词代表的篇章关系，那么只会有 24 个样本发生错误，所以使用连接词预测篇章关系是可行的。

2.2 模型结构

模型结构如图 1 右图所示。我们在 arg1 和 arg2 中间加入 3 个单词的空白进行拼接处理后输入预训练模型中得到最后一层三个空白的隐藏层表示，之后接平均池化层和 MLP 分 97 类（97 是测试集和训练集中所有连接词的个数）。之后我们按照连接词和篇章关系的对应关系，将 97 维中与篇章关系对应的连接词位置进行加和，将 97 维输出转为 4 维输出，之所以不直接使用概率最大的连接词预测是为了防止过拟合，增加网络的泛化性。使用三个空白是因为我们发现样本中的大多数连接词是三个单词，并且实验结果也确实比只是用一个位置好。

其中具体参数如下：arg1 和 arg2 截断到 64 单词长度，不足的进行补齐；roberta 使用的是 roberta-base，当然如果使用更大的模型比如 roberta-large，效果会更好；平均池化采用（3，3）的核，MLP 就只有一层，输入 256 维，输出 97 维。实验结果如表 1 所示。

表 1: 实验数据

model	accuracy	Macro F1
baseline	0.7027	0.6418
prompt	0.7194	0.6672

3 其他提升 F1 的失败尝试

3.1 数据分析

实验发现，如果直接将显示篇章关系数据不加处理直接使用，那么基线模型和 prompt 模型的性能都会有所降低。考虑到可能是样本分布的问题，我们只选用连接词在隐式篇章关系中出现的数据进行训练，性能依然降低。并且我们还发现，显示篇章关系数据，隐式篇章关系训练数据，隐式篇章关系测试数据的标签样本分布是近似的，所以根据实验结果这三者数据和标签的联合分布应当不同，只考虑标签分布是不行的。

此外，我们还发现，无论是训练集还是测试集，预测结果都向第二类偏移，即'expansion'。虽然'expansion' 类的训练和测试数据确实基本占据了所有数据的一半，但是通过集成学习部分的实验，我们发现即使在集成学习中改变训练集的分布，降低真实'expansion' 类的比例，这种情况依然没有好转，这使我们有理由相信'expansion' 类有其自身的特殊性。

3.2 集成学习

我们训练两个 prompt 模型：一个使用所有训练数据进行训练；另一个使用第一个模型分类错误的数据进行训练。两个模型得到的最终输出直接相加取 argmax 后得到样本所属的类别。实验发现，在 prompt 模型中使用此方法并不能使模型效果进一步提升。

3.3 双序列

如果将一个 arg 输入一个编码器中得到编码结果后再进行交互，我们发现效果还不如基线模型，应该是因为使用双编码器不能在编码过程中进行交互，所以效果没有单编码器效果好。于是我们尝试调换 arg1 和 arg2 的顺序后进行拼接，将两个不同顺序的语料分别输入两个编码器后再进行交互（就是一个编码器接受 arg1+arg2，一个编码器接受 arg2+arg1，两个编码器之间参数不共享），只能说效果没有改进，该向第二类偏还向第二类偏。

4 总结

总的来说，这次实验通过更换预训练模型和 prompt 方法提升了模型的效果。但是向'expansion' 类严重偏移的问题始终没有得到很好的解决，如果这个问题

能很好的解决，模型的效果一定可以获得显著提升。

5 附录

5.1 实验环境

本次实验使用的 IDE 是 pycharm，版本为 2021.3，python 版本为 3.9，需要用到的库以及相应版本如表 7 中所示。

表 2: 第三方库以及相应版本号

	numpy	sklearn	transformers	matplotlib	torch
版本	1.23.3	0.24.2	1.2.0	3.5.1	1.10.1

5.2 数据集

使用的数据集就是老师给出的数据集，没有使用其他数据。预训练模型都是 hugging face 上的。