“技术需求”与“技术成果”项目之间关联度计算模型*

莽就完事了

马凯欣

计算机技术&2019 级

东北林业大学

中国-哈尔滨

1239977613@qq.com

团队简介

我们团队参加的是“技术需求”与“技术成果”项目之间关联度计算模型赛题，团队的名字叫莽就完事了。团队中只有我一个人，自然我也就是队长了。我的名字叫马凯欣，来自东北林业大学，目前是计算机技术专业研究生一年级在读，专业方向为自然语言处理。我以前没有参加过大数据与人工智能的相关比赛，这次完全是头一次，CCF BDCI 的许多赛题也让我挑不知道选哪个好。但要说竞赛经历，那我倒是有些，本科的时候曾参加过 ACM-ICPC 竞赛，也取得过一些奖项。

摘要

本篇论文介绍了采用伪孪生网络结构的 BERT 微调模型。在数据清洗方面只使用了简单的替换。尝试了两种数据增广，但均会造成过拟合现象，由于时间有限未进一步进行尝试。在预测时使用了一种偏好处理使误差减小。经过实验对比，伪孪生 BERT 用于“技术需求”与“技术成果”项目之间关联度计算时效果优于 BERT 微调模型。

关键词

文本关联度，伪孪生网络结构，BERT 预训练

1 引言

BERT 是谷歌发布的基于双向 Transformer 的大规模预训练语言模型，该预训练模型能高效抽取文本信息并应用于各种 NLP 任务，与其他语言表示模型不同，BERT 旨在通过联合调节所有层中的上下文来预先训练深度双向表示。因此，预训练的 BERT 表示可以通过一个额外的输出层进行微调，并适用于广泛任务的先进模型的构建，比如自然语言推理，而无需针对具体任务做大幅架构修改[1]。Matthew Peters 等人对不同数据集进行微调实验，发现预训练的性能取决于预训练与目标任务的相似度[2]。Ran Wang 等人的实验中在 BERT 上堆叠一定的神经网络可以取得比 BERT 更好的效果[3]。此项任务作为自然语言推理的变形，BERT 可以很好的完成任务，因此此次比赛的最终模型主体采用 BERT。

2 方法

首先简单描述一下 BERT 进行自然语言推断时的模型，如图 1[1]。

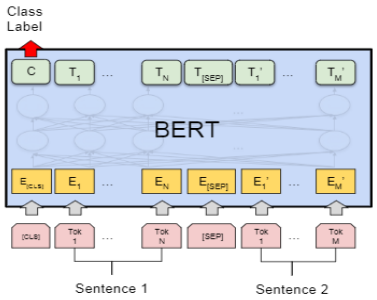

图 1：BERT 进行自然语言推断时的模型

2.1 BERT

BERT 在预训练阶段输入是将句子加入[CLS]和[SEP]两个特殊字符，采用 WordPiece 方法进行分割[4]，并加入段嵌入与位置嵌入，每个序列的第一个标记始终是特殊分类嵌入[CLS]，该特殊标记对应的最终输出被用作分类任务中该序列的总表示。

2.2 尝试过的模型

2.2.1 标题与内容拼接的孪生 BERT 模型

首先将数据集中对应标题与内容拼接成一段长文本。该模型是一个孪生网络的结构，使用了两个共享权重的 BERT 模型，分别将拼接后的技术成果长文本与技术需求长文本输入到 BERT 中，将两个[CLS]分别取出后做差，最后传入到一个全连接层进行分类，模型如图 2。

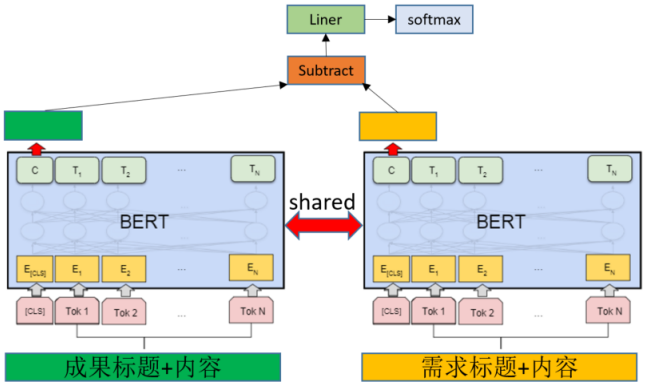

图 2：标题与内容拼接的孪生 BERT 模型

2.2.2 分别进行标题与内容关联度判别的孪生 BERT 模型

该模型是一个孪生网络的结构，使用了两个共享权重的 BERT 模型，分别将技术成果标题与技术需求标题和技术成果内容与技术需求内容输入 BERT 中，将两个[CLS]分别取出后进行拼接，最后传入到一个全连接层进行分类，模型如图 3。

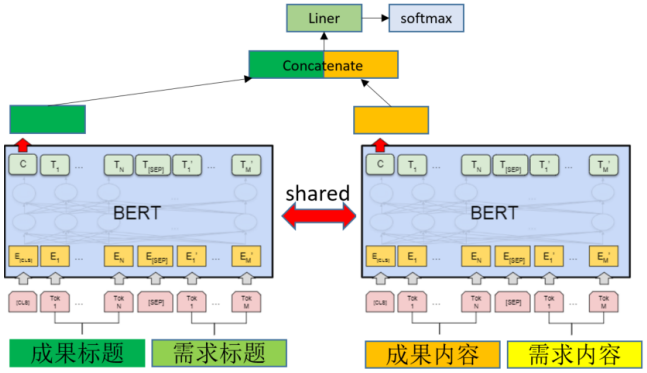

图 3：分别进行标题与内容关联度判别的孪生 BERT 模型

2.2.3 分别进行标题与内容关联度判别的伪孪生 BERT 模型

该模型是一个伪孪生网络的结构，与上述模型的区别在于两个 BERT 模型并没有进行共享权值，分别将技术成果标题与技术需求标题和技术成果内容与技术需求内容输入到对应 BERT 中，将两个[CLS]分别取出后进行拼接，最后传入到一个全连接层进行分类，模型如图 4。

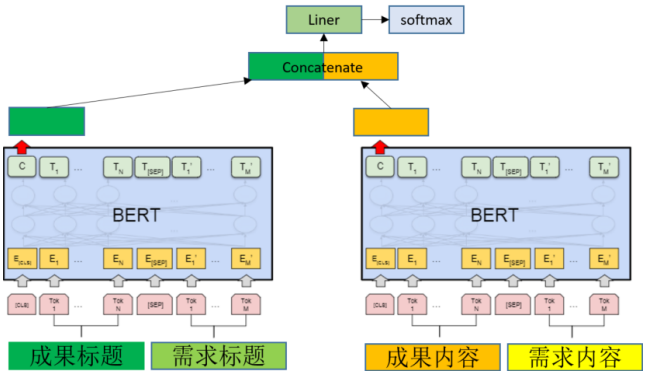

图 4：分别进行标题与内容关联度判别的伪孪生 BERT 模型

3 实验

3.1 数据清洗

经过对技术成果和技术需求的较短内容进行筛选查看，发现其中存在一定量的空白、“\n”、“未提供。”等无用信息。简单的使用对应标题对无用信息进行替换即可。

3.2 数据增广

对问题进一步化简，可以简化成两个文本之间的关联度计算。

那么 A 文本与 B 文本之间关联度，同样也是 B 文本与 A 文本之间关联度。该方法在仅取标题时可以提升成绩。当加入内容时会造成过拟合，最终未采用该方法。

那么假设 A 文本与 B 文本之间关联度为 4，A 文本与 C 文本之间关联度为 3，那么可以假定 B 文本与 C 文本之间关联度为 3，按照这个思路可以假设关联矩阵

$$R=\begin{pmatrix}1&1&1&1\\1&0&2&2\\1&2&3&3\\1&2&3&4\end{pmatrix}$$

其中 A 文本与 B 文本之间关联度为 i，A 文本与 C 文本之间关联度为 j，那么 B 文本与 C 文本之间关联度为$R_{i,j}$。此方法可增加数据 295994 条，从中按照原数据集各个关联度等级的比例从中随机取出 10000 条。该方法我认为具有一定的可

能性，但由于训练时间过长、提交次数有限，尝试过的参数均会造成过拟合现象。最终模型中未对数据进行数据增广。

3.3 最终模型

经过一定量的实验对比最终模型确定为分别进行标题与内容关联度判别的伪孪生 BERT 模型。其中进行技术成果标题与技术需求标题关联度计算的 BERT 采用谷歌开源的 BERT-base[5]；进行技术成果内容与技术需求内容关联度计算的 BERT 采用哈工大讯飞联合实验室发布基于全词覆盖的 BERT-WWM[6,7]。该预训练由于采用了全词覆盖，在多数情况下可以取得更好的效果[6]。在第一个进行技术成果标题与技术需求标题关联度计算的 BERT 中输入最大长度 MaxLenT 设置为 128，两个标题拼接最大长度也没有超过 128 个字，同时这样可以减少训练时间和显存需求，标题拼接后的长度所占全部长度的百分比如图 5 所示；在第二个进行技术成果内容与技术需求内容关联度计算的 BERT-WWM 中输入最大长度 MaxLenC 设置为 512，尽可能多的读取数据内容。两个 BERT 都采用 12layers, 768hidden states, 12heads 版本，该模型采用 7 折交叉验证，其中 batch size 取 16，epoch 取 8，并在训练时保存较好的模型权值[8]，初始学习率设置成 5e-5，后续学习率设置成 1e-5。

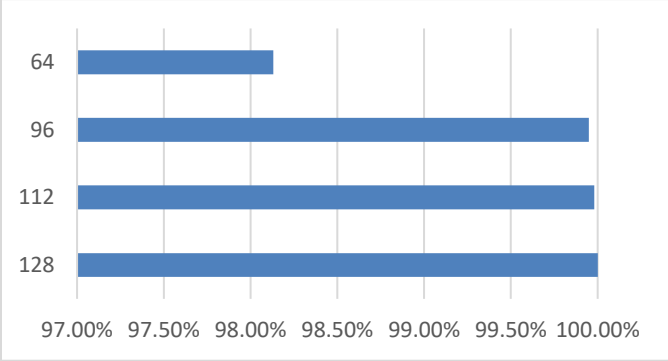

图 5：拼接后的长度所占全部长度的百分比

将多次训练结果的预测概率求取平均是加强泛化的好方法[9]。但是通过观察测评指标

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|pred_i - y_i| \tag{1}$$

$$Score = \frac{1}{1+MAE} \tag{2}$$

其中$pred_i$为预测样本，y_i为真实样本，这个差值越小越好。那么当我的模型判断关联度为 1 和 2 的概率非常接近时，我更希望它能输出 2。所以当模型无法判别时，我更希望它的输出能够偏向 2 和 3，故需要将概率进行偏好处理，新的概率如公式(3)所示。

$$\begin{cases} p_i' = (1-\alpha)p_i & i = 1,4 \\ p_i' = \alpha p_i & i = 2,3 \end{cases} \tag{3}$$

如图 6 为一次实验结果，经实验发现将α确定在 0.52 至 0.525 之间成绩较好。最终我将α取值为 0.52。

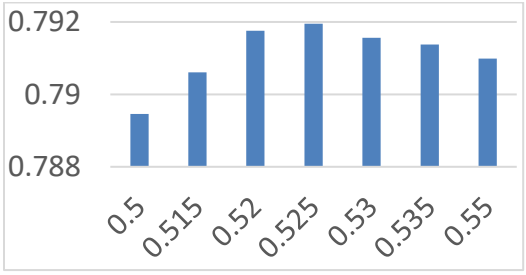

图 6：在一次实验中α取不同结果所得分数

3.4 模型对比

表 1：多种模型成绩对比

模型	初赛成绩	复赛成绩
BERT-base[10]	0.78585178	0.79595751
BERT+GRU	0.78927016	/
BERT+BiGRU	0.78907849	/
RoBERTa-base	0.78077936	/
孪生 BERT-1	0.78604090	0.79607499
孪生 BERT-2	0.78509617	0.79843128
BERT+数据增广-1	/	0.80163449
BERT+数据增广-2	/	0.77996242
BERT+数据增广-3	/	0.79548806
BERT-T128C512	0.79079902	0.79866767
BERT-WWM-T128C512	0.79099053	0.80008900
最终模型	**0.79175758**	**0.80642748**

1.其中 BERT-base、BERT+GRU、BERT+BiGRU、RoBERTa-base、BERT+数据增广-1、BERT+数据增广-2、BERT+数据增广-3 模型中输入均只有技术成果标题与技术需求标题，MaxLenT 为 128，其余超参数与最终模型中基本相同。

2.孪生 BERT-1 模型为标题与内容拼接的孪生 BERT 模型，MaxLen 为 512，其余超参数与最终模型中基本相同。

3.孪生 BERT-2 模型为分别进行标题与内容关联度判别的孪生 BERT 模型，MaxLen 为 512，其余超参数与最终模型中基本相同。

4.BERT+数据增广-1 模型中，数据增广采用第一种方式。

5.BERT+数据增广-2 模型中，数据增广采用第二种方式，且取全部增广数据。

6.BERT+数据增广-3 模型中，数据增广采用第二种方式，但按照原数据集各个关联度等级的比例从中随机取出。

7.BERT-T128C512 模型中 BERT 均采用谷歌发布的 BERT-base，其余超参数与最终模型中相同。

8.BERT-WWM-T128C512 模型中 BERT 均采用采用哈工大讯飞联合实验室发布的 BERT-WWM，其余超参数与最终模型中相同。

4 结论

该模型按照思路变化称其为优化后的伪孪生 BERT，其实可以简单的认为它是将两个 BERT 的输出结果进行拼接，然后进行四分类。模型并不复杂但结果却在我意料之外。我个人认为 BERT-WWM 预训练相比于 BERT 预训练对中文效果应该更好，而得到这样的结果，可能的原因是两个预训练在训练时使用的语料库不同[5,7]，标题部分中专业名词比重较大且短小，BERT 对此比较敏感，而 BERT-WWM 对常规文本比较敏感。当然这个成绩中也有预测偏好处理的功劳。

致谢

感谢 DataFountain 提供公平、公正的比赛平台；感谢主办方中国计算机协会主办此次比赛；感谢河南八六三软件股份有限公司提供的赛题；以及感谢其他各方单位的支持；感谢刘美玲老师的指导。感谢华南理工大学 Chevalier 同学在知乎上分享的 BaseLine。

参考

[1] JDevlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[2] Peters, Matthew E, Ruder, Sebastian, Smith, Noah A. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. https://arxiv.org/abs/1903.05987

[3] Wang R , Su H , Wang C , et al. To Tune or Not To Tune? How About the Best of Both Worlds?. https://arxiv.org/abs/1907.05338?context=cs

[4] Wu Y , Schuster M , Chen Z , et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. https://arxiv.org/abs/1609.08144

[5] Google, bert, https://github.com/google-research/bert

[6] Cui Y , Che W , Liu T , et al. Pre-Training with Whole Word Masking for Chinese BERT. https://arxiv.org/abs/1906.08101

[7] 哈工大讯飞联合实验室, 中文预训练 BERT-wwm, https://github.com/ymcui/Chinese-BERT-wwm

[8] Huang G , Li Y , Pleiss G , et al. Snapshot Ensembles: Train 1, get M for free. https://arxiv.org/abs/1704.00109

[9] Izmailov P , Podoprikhin D , Garipov T , et al. Averaging Weights Leads to Wider Optima and Better Generalization. https://arxiv.org/abs/1803.05407

[10] Chevalier,“技术需求”与“技术成果”项目之间关联度计算模型 TOP10 baseline, https://zhuanlan.zhihu.com/p/82737301