全国大学生生物医学工程创新设计竞赛

预赛文档

[作品名称：中文电子病历命名实体识别和关系抽取]

**[作品ID号:** **B7060]**

**[参赛组别：命题项目组 医疗人工智能类]**

目录

[**1. 项目概述** 3](#_Toc107505038)

[**1.1项目目标** 3](#_Toc107505039)

[**1.2需求分析** 4](#_Toc107505040)

[**2. 项目设计方案** 4](#_Toc107505041)

[**2.1 数据预处理** 4](#_Toc107505042)

[**2.2 项目模型解读** 5](#_Toc107505043)

[**2.2.1基于GlobalPointer的命名实体识别** 7](#_Toc107505044)

[**2.2.2基于PURE\_RE的关系抽取** 9](#_Toc107505045)

[**2.3技术亮点** 12](#_Toc107505046)

[**2.3.1 对抗训练(FGM)** 12](#_Toc107505047)

[**2.3.2 R-Drop** 13](#_Toc107505048)

[**3. 项目技术指标与测试** 14](#_Toc107505049)

[**3.1 F1指标测评** 14](#_Toc107505050)

[**3.2 模型运行速度测评** 16](#_Toc107505051)

[**4. 作品展示** 17](#_Toc107505052)

[**参考文献** 17](#_Toc107505053)

**1. 项目概述**

随着电子病历的应用范围越来越广，如何对其中的医疗信息进行挖掘逐渐成为近年来的研究热点。但医疗数据结构化仍处于起步阶段，更多的医疗数据仍然以自然语言形式出现。为了得到结构化的医疗数据，本项目实现了中文电子病历命名实体识别及关系抽取。作为自然语言处理核心，命名实体识别、关系抽取算法能力对医疗数据结构化至关重要。

**1.1项目目标**

基于官方提供的中文电子病历数据集，经过数据预处理，我们将该数据集分别输入命名实体识别模型和关系抽取模型进行半监督训练。通过tensorboard将损失函数、F1\_score可视化，以调整迭代次数、学习率等参数，尽可能降低损失函数，提高模型F1值，最终两曲线趋于平滑，此时预测结果接近真实结果。图1展示了中文电子病历命名实体识别和关系抽取的过程。

图图片包含 图示

描述已自动生成1 中文电子病历命名实体识别和关系抽取流程简图

**1.2需求分析**

进行电子病历标注需要具备相关专业知识及临床经验。然而，临床医生长期从事语料建设和维护的标注方案难以实现。一方面，临床医生用于语料标注的时间有限；另一方面，邀请临床医生标注语料成本更高。同时，语料标注的结果还会受到许多因素的影响，例如临床医生之间的差异，并且实体识别及关系抽取是由医生根据其专业水平做出决定，专业水平的差异会对结果造成一定的影响。即使根据相同的标注标准，也很难得到完全相同的标注结果。因此，我们希望借助计算机自然语言处理相关技术，实现中文电子病历命名实体识别以及关系抽取自动化，这对于提供可靠和客观的标注结果和减轻临床医生过于繁重的工作量十分有益。

**2. 项目设计方案**

由于数据获取和标注的困难，仍缺乏统一的标准和公共数据集，医疗命名实体识别和关系抽取在我国的研究进展相对缓慢。在样本少，实体多，实体间关系复杂的情形下，我们对数据做增强处理，对已有模型进行改进。

**2.1 数据预处理**

* 文档裁剪

在官方提供的数据集中，部分文本相对较长，超过了Bert等模型的最大输入长度。因此将原文本裁剪为几个子文本来实现数据增强，这样将获得更多的样本数据。

* 数据集修正

我们注意到官方提供的数据集在某些实体的标注上存在一定程度的标注标准不统一的问题，手动对数据集进行了修正。同时考虑到标注标准可以由模型从原始数据集中自动学习，泛化能力会更加强，所以保留了原始数据集。此时产生了原始数据集和修正数据集这两个独立的数据源。

* 数据增强

我们将原始训练集与验证集混合，采用5折交叉验证对数据进行划分，减少了单次划分训练集和测试集方法的随机性对模型的影响，辅助后续进行参数调整。

* 数据格式转换

我们对处理后的数据进行格式转换，将现有数据集的格式转换为模型输入所需要的格式。

**2.2 项目模型解读**

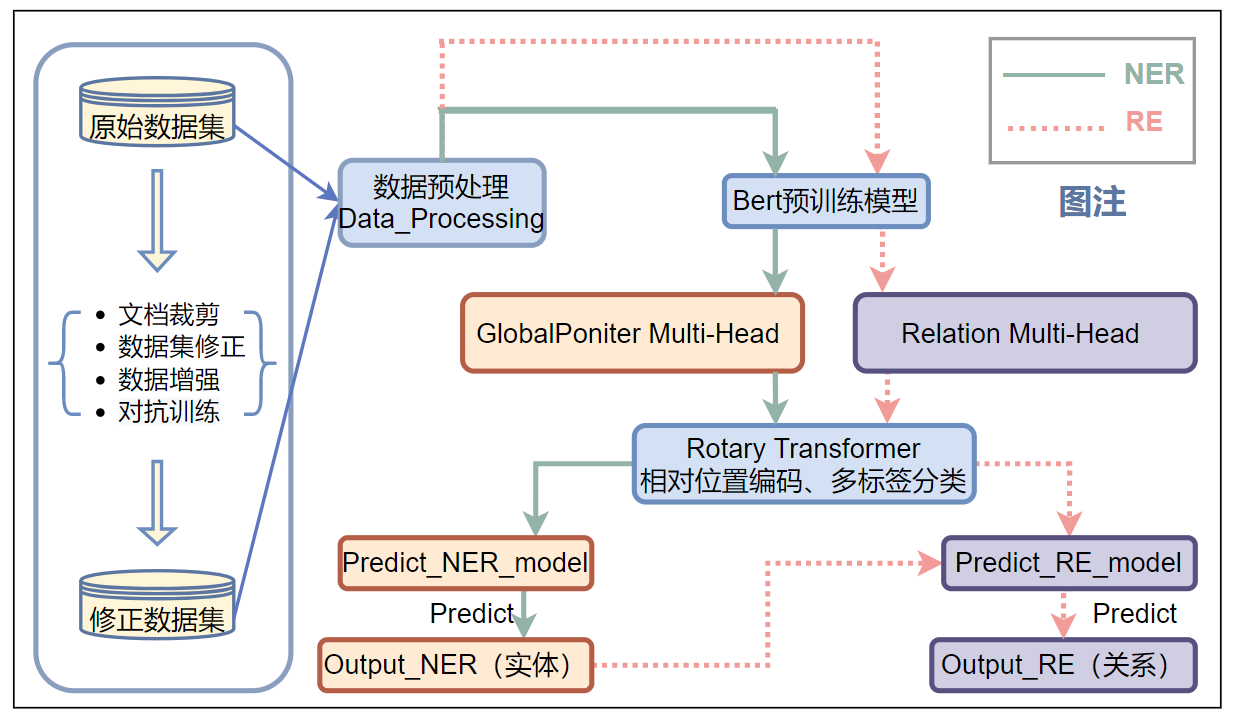
使用自然语言处理电子病历中的命名实体识别和关系抽取主要有两种方法：Pipeline方法和Joint方法。Pipeline方法要先进行实体抽取，再进行关系抽取，这种方法较为简单，并且可以优化不同的模块，然而Pipeline存在一定的传递误差。Joint方法虽然可以避免传递误差，但其方案设计困难，很难进行局部优化。所以我们采用Pipeline方式实现中文电子病历命名实体识别与关系抽取，其中，我们采用了哈工大开源的BERT模型作为预训练模型。采用GlobalPointer方法实现命名实体识别，即将实体的首部和尾部视为一个整体，该实体是当前文本序列的一个连续片段。这样避免了训练和预测时的不一致。由于该任务为多标签分类问题，在模型中将“softmax+交叉熵”推广到多标签分类问题。另外，我们采用PURE\_RE模型实现关系抽取，提出一种近似模型，加速模型的推理。图2展示了基于GlobalPointer和PURE\_RE模型的命名实体识别和关系抽取总体框架。

图2 基于GlobalPointer与PURE\_RE模型的

命名实体识别和关系抽取总体框图

其中绿色实线路线为命名实体识别路线，红色虚线路线为关系抽取路线。在经过数据预处理后，我们将数据集分别输入Bert预训练模型，分别对数据集实体与关系进行学习。由于该项目实体识别和关系抽取均为多标签分类问题，又分别使用多头注意力机制以及旋转式位置编码，实现多标签分类。至此，训练过程结束，生成对应的预测模型。使用Predict\_NER\_model对实体进行预测，使用Predict\_RE\_model且输入命名实体识别的预测结果可以得到预测的实体及实体间关系。经过数据格式转换，生成最终提交的测试集ANN文件。

**2.2.1基于GlobalPointer的命名实体识别**

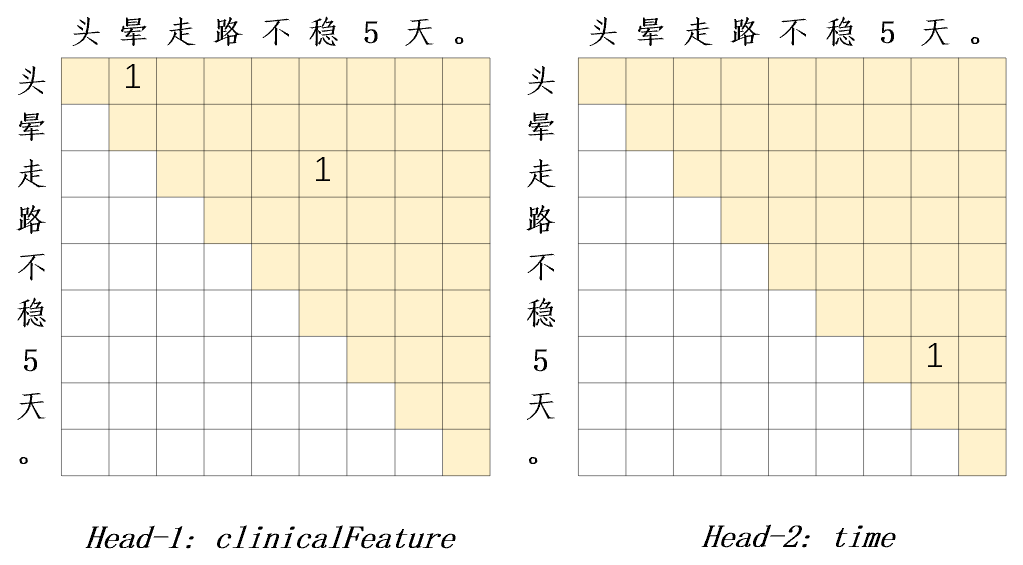
GlobalPointer利用了全局归一化的思路来进行命名实体识别（NER），可以无差别地识别嵌套实体和非嵌套实体，在非嵌套（Flat NER）[1]的情形下它能取得媲美CRF的效果。在训练的过程中,GlobalPointer不需要像CRF那样递归计算分母。在预测时不需要进行动态规划，是完全并行的，理想情况下使用GlobalPointer的模型的时间复杂度接近。使用GlobalPointer进行训练模型，不仅能达到不错预测的效果，还能节约我们模型训练的时间与预测的时间。和以往的Pointer Network的设计相比，GlobalPointer做命名实体识别时将实体的首尾作为一个整体去进行训练，所以更加具有全局观。GlobalPointer事实上就是Multi-Head Attention的一个简化版，有多少种实体就对应多少个head。如图所示：

图3 GlobalPoniter识别实体示意图

设长度为的输入经过编码后得到向量序列，通过变换 和 ,我们可以得到序列向量和，它们是识别第种类型实体所用的向量序列。此时我们可以定义:

作为从到的连续片段是一个类型为的实体的打分。也就是说，用与的内积，作为片段是类型为的实体的打分（logits），这里的指的是序列的第个到第个元素组成的连续子串。

GlobalPointer对实体的长度和跨度都不是特别敏感，如果没有相对位置信息，很容易把任意两个实体的首尾组合都当成目标预测出来。由此可以通过旋转式位置编码（RoPE）[3]显示的引入相对位置,那么实体的打分优化为:

其中为变换矩阵，通过该矩阵使其对实体长度和跨度更敏感。最终我们需要解决的就是一个多标签的分类问题。

从形式上来看，单标签分类比多标签分类要容易，就是因为单标签可以使用“Softmax+交叉熵” [2]的方法，它不会存在类别不平衡的问题，而多标签分类中的“sigmoid+交叉熵”就存在不平衡的问题。所以将“Softmax+交叉熵”推广到多标签分类上去是很有必要的。使用该方法可以自动确定打分的阈值，进而识别出真实的实体。

接下来就是损失函数的设计,利用Softmax激活并用交叉熵作为损失函数,在GlobalPointer的场景下，它可以写成:

其中:是该样本的所有类型为的实体的首尾集合是该样本的所有非实体或者类型非的实体的首尾集合。在解码过程中，所有的片段是类型为的实体。

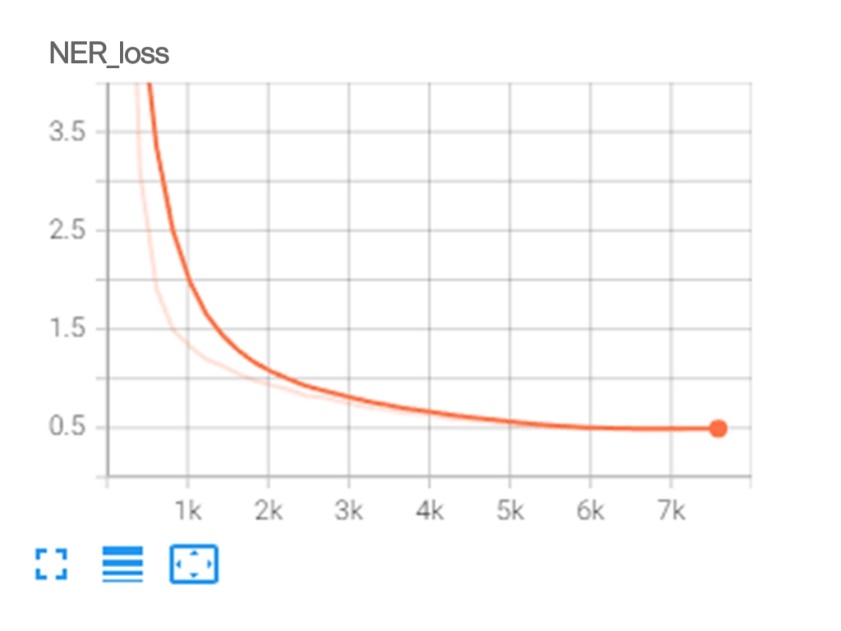
在经过每一轮的训练之后，模型的损失函数逐渐降低，损失梯度逐渐减小，我们使用Tensorboard可视化工具展现了训练的日志记录。

图4 tensorboard NER模型中损失函数变化图

**2.2.2基于PURE\_RE的关系抽取**

我们发现命名实体识别和关系抽取模型在文本表示中获取的是不同的信息，如果共享他们的表示信息可能对模型的性能产生影响，而关系抽取模型的输入层融合实体信息又是至关重要的。所以我们采用Pipeline方法对病历中的实体进行关系抽取。命名实体识别和关系抽取模型的是独立训练的，关系抽取模型仅依赖命名实体识别模型来提供输入特征。如图在关系抽取阶段将实体边界和类型作为标识符加入到实体Span前后，识别出的实体是clinicalFeature，就把<cl>和</cl>插入到实体边界，同理对于time也插入<ti>和</ti>，独立的考虑每对实体之间的关系。

图示

中度可信度描述已自动生成图5 添加实体类型标识示意图

因此我们的模型存在一个缺点，我们需要对每对实体运行一次关系抽取模型，实体间具有主体和客体的关系方向性。同一文本需要进行多次编码。这极大地阻碍了我们模型的推理。为了解决这一问题。提出了一个近似模型[4]。

我们没有将标识符加入到实体Span前后，而是将实体边界和类型的标识符放入到文本之后，然后与原文对应实体共享位置向量。图中相同的颜色代表共享相同的位置向量。通过这种近似模型可以实现一次编码文本就可以判断所有实体的关系。这大大的提高了我们模型的推理速度。

图片包含 图形用户界面

描述已自动生成图6 将实体边界和类型标识符放在文本后面

此时我们也可以定义:

其中表示主体与客体的关系打分。同样使用与的内积，作为主体与客体之间是否存在关系的打分（logits），这样我们又很巧妙的将关系抽取转化成了多标签分类问题。

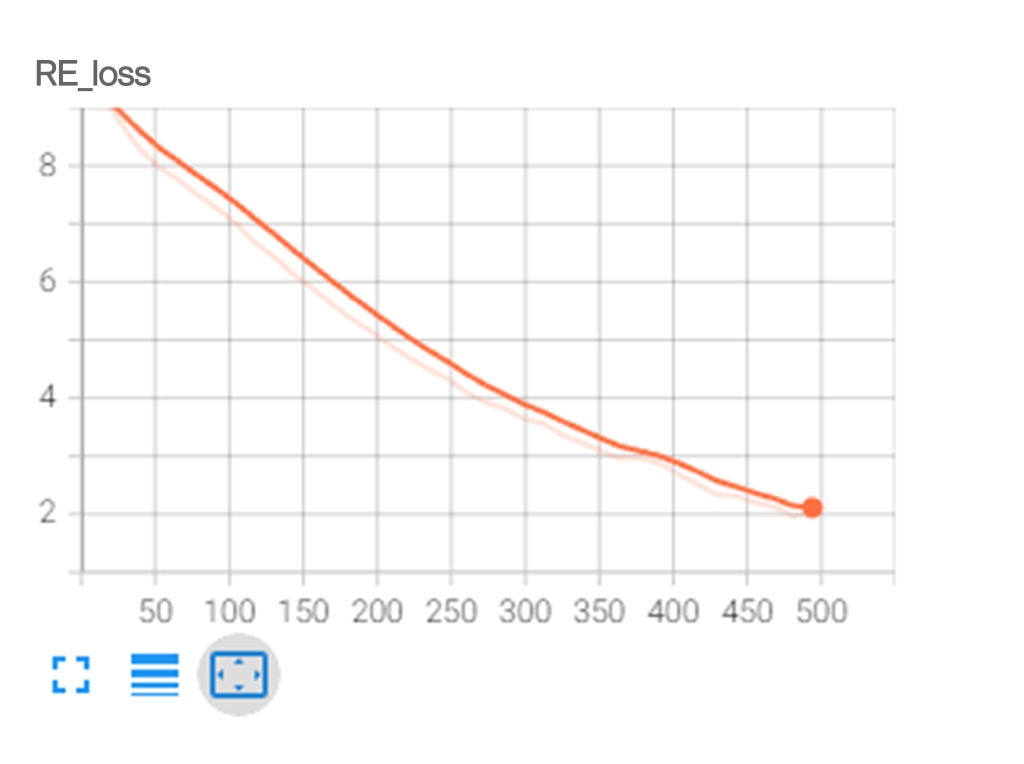
由于主体和客体存在关系方向性，且每个实体对存在不同的距离。为了防止模型将两个远距离的实体打上高分。我们同样引入一个变换矩阵，使关系抽取的打分优化为：

这里的变换矩阵可以强化两个较近实体关系的打分，而弱化两个较远实体关系的打分。

其损失函数仍然可以表示成：

其中为关系实体的集合，为非关系实体的集合。在解码过程中，所有的主体与客体都是存在关系的。

在经过每一轮的训练之后，模型的损失函数逐渐降低，损失梯度逐渐减小。

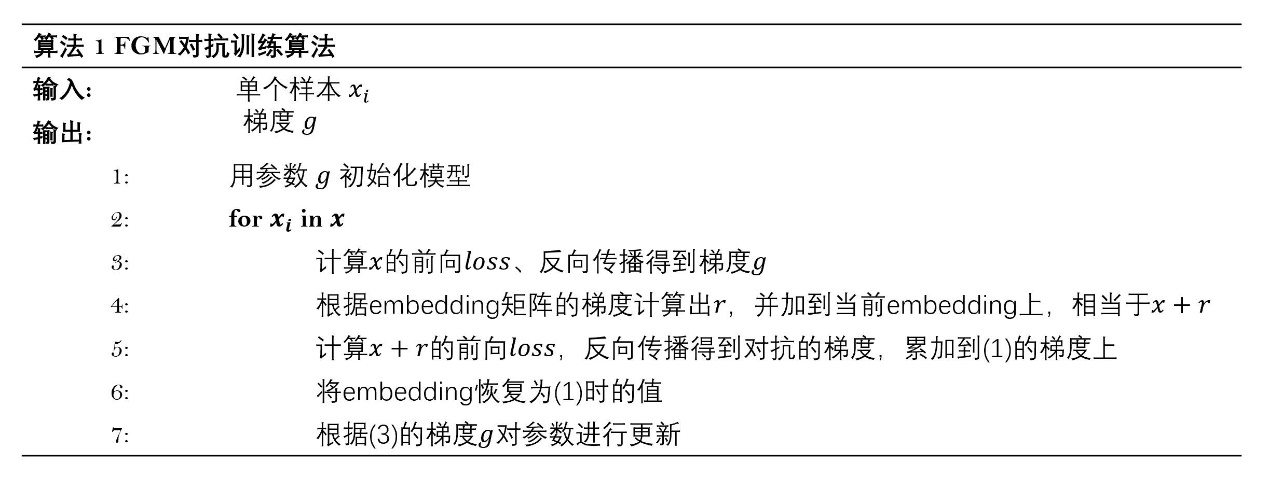
图7 Tensorboard RE模型中损失函数变化图

**2.3技术亮点**

**2.3.1 对抗训练(FGM)**

对抗训练（adversarial training）是增强神经网络鲁棒性的重要方式。在对抗训练的过程中，样本会被混合一些微小的扰动（改变很小，但是很可能造成误分类），然后使神经网络适应这种改变，从而对于对抗样本具有鲁棒性。在语言领域对抗训练既提高了鲁棒性，也提高了泛化性。总的来说，对抗训练可以统一写成如下格式：

其中代表训练集，代表输入，代表标签，、是模型参数，是单个样本的 是对抗扰动, 是扰动空间。下表为流程表。

表1 FGM算法流程表

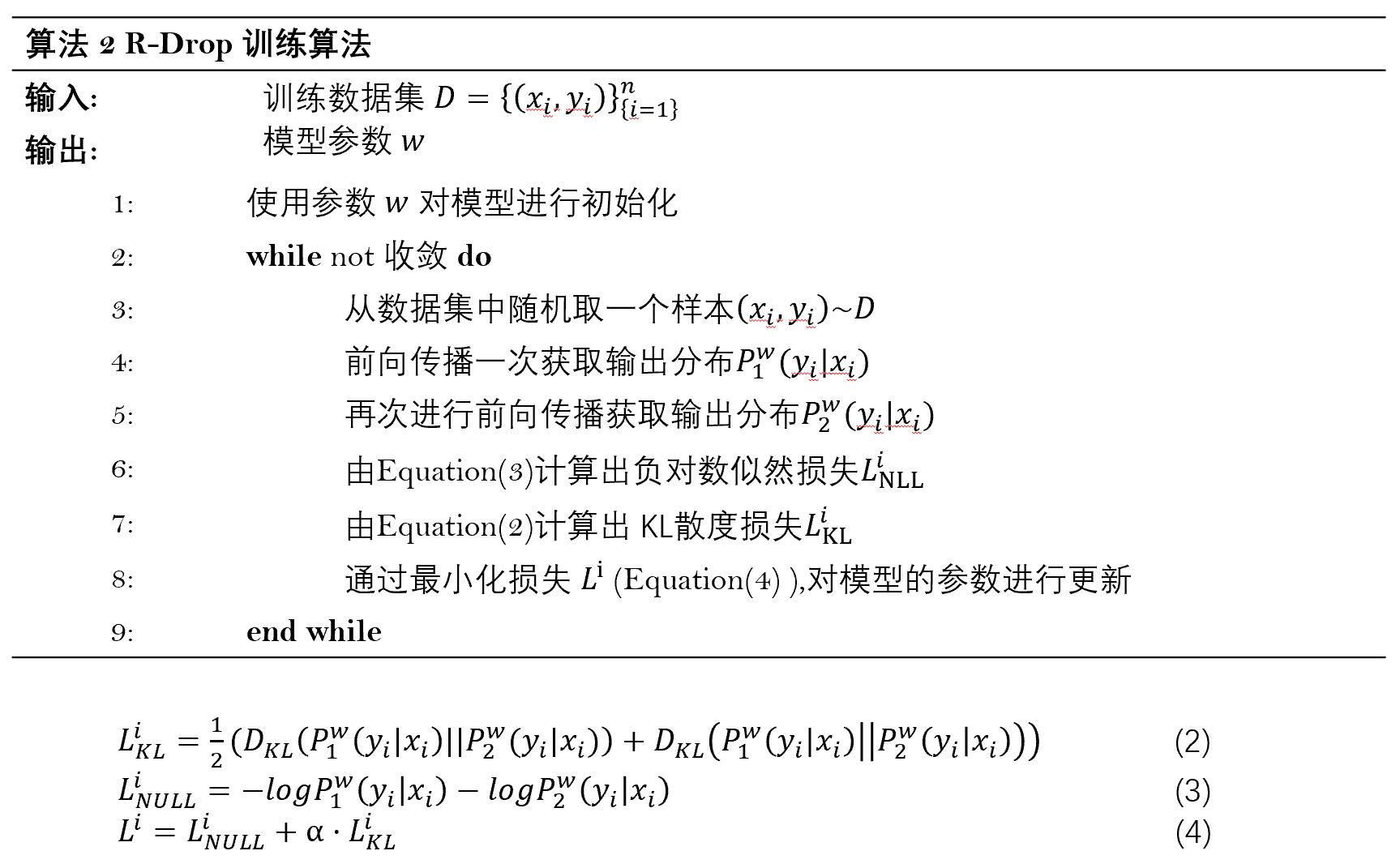
我们在模型中，向Embedding层在梯度方向添加扰动，可以使得Word Embedding的质量更好，不仅提高了置信度还避免了模型过拟合，模型的性能大大提升。

值得注意的是，添加对抗训练后，训练时间会近乎成倍增常，对于FGM来说，每一次更新参数，都进行了两次后向传播。在大部分情况下，模型加上FGM能带来正向的效果，而且FGM相对其它对抗训练方法，所耗时间也较少。

**2.3.2 R-Drop**

R-Drop[5]的思想是：同一个step里面，对于同一个样本，前向传播两次，由于Dropout的存在，会得到两个不同但差异很小的概率分布，通过在原来的交叉熵损失中加入这两个分布的KL散度损失，来共同进行反向传播，参数更新。下表为R-Drop算法流程表。

表2 R-Drop训练算法

由于Dropout是典型的训练和预测不一致的方法，我们通过在模型中引入R-Drop，增加一个正则项，来强化模型对Dropout的鲁棒性，使得不同的Dropout下模型的输出基本一致，因此能降低这种不一致性。

**3. 项目技术指标与测试**

**3.1 F1指标测评**

本项目采用命名实体识别和关系抽取的通用评价指标：精确率（）、召回率（）、和,具体公式如下：

其中，对于命名实体识别而言，为正确识别的实体个数，为识别到不相关病历实体的个数，为未识别到相关病历实体的个数。对于关系抽取为正确抽取的实体关系数量，为识别到不相关实体关系的个数，为未识别到相关实体关系的个数。我们采用交叉验证的方式多次计算模型的值，并用tensorboard展示出来：

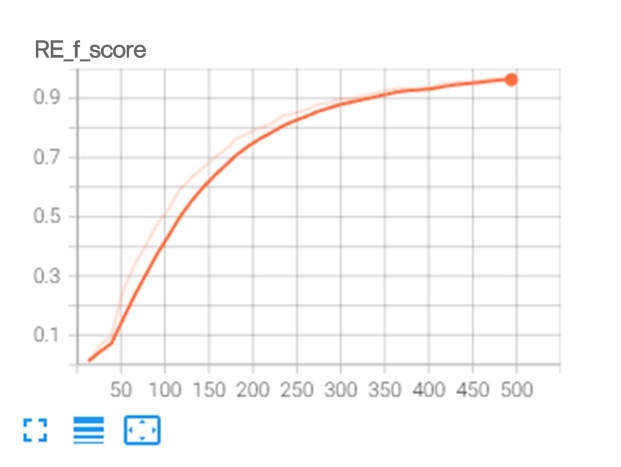
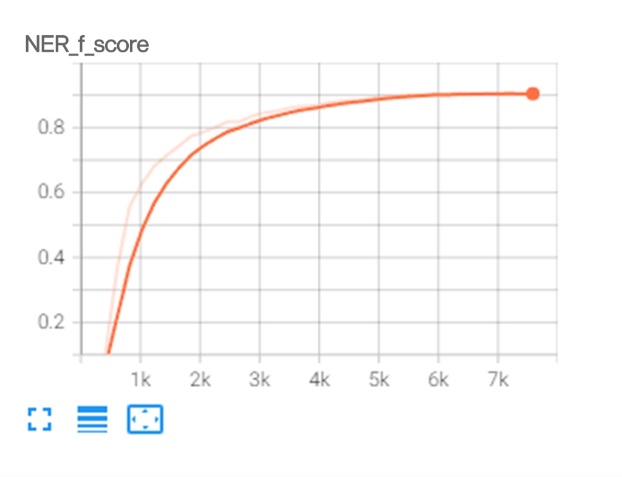
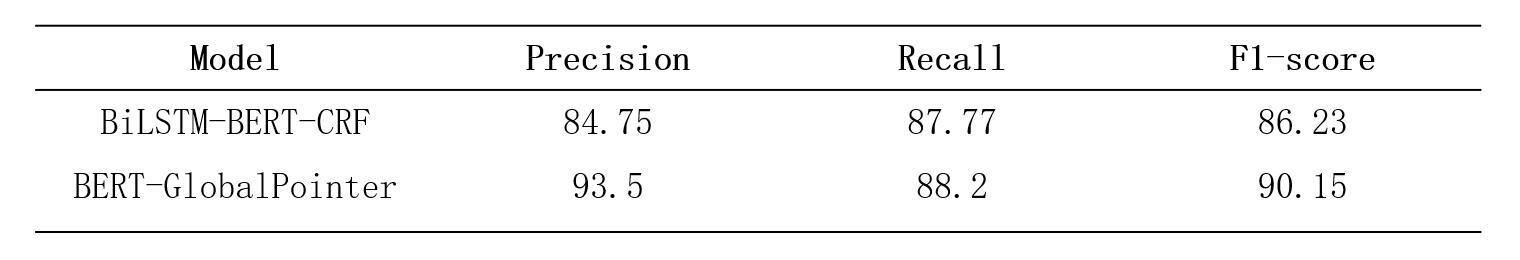


图8 Tensorboard NER与RE 模型的F1值变化图

我们用官方给的数据集验证了目前比较主流的命名实体识别方案, BERT-GlobalPointer得到了最优的结果，F1值为90.15%，F1值相比于BILSTM-BERT-CRF提高了3.92%，在一定程度上验证了我们的命名实体识别模型具有较好的识别性能。同时我们用官方给的数据集通过我们的关系抽取模型,得到的F1值为89.2%，不同模型在整体实体类型上的结果如下表所示。

表4 不同模型在实体识别任务上的效果对比()

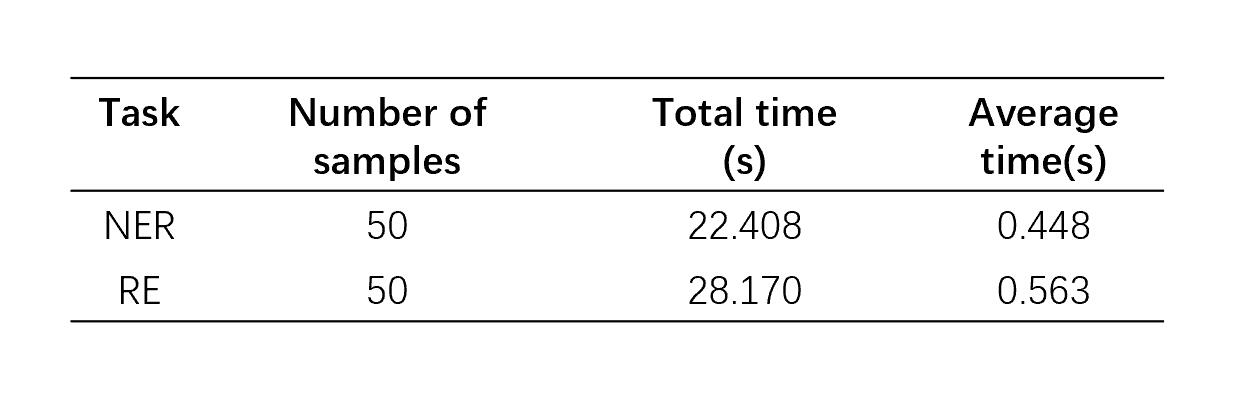
**3.2 模型运行速度测评**

硬件平台：NVIDIA Quadro RTX 5000

python环境：

1. python==3.7.11
2. torch==1.9.0
3. transformers==4.11.3
4. pytorch-lightning==1.4.7
5. tqdm==4.62.3
6. numpy==1.21.0
7. scikit-learn==0.24.2
8. setuptools==59.5.0

我们分别测得命名实体识别和关系抽取模型单条样本的运行速度如下表所示：

表5 模型预测时间（s）

**4. 作品展示**

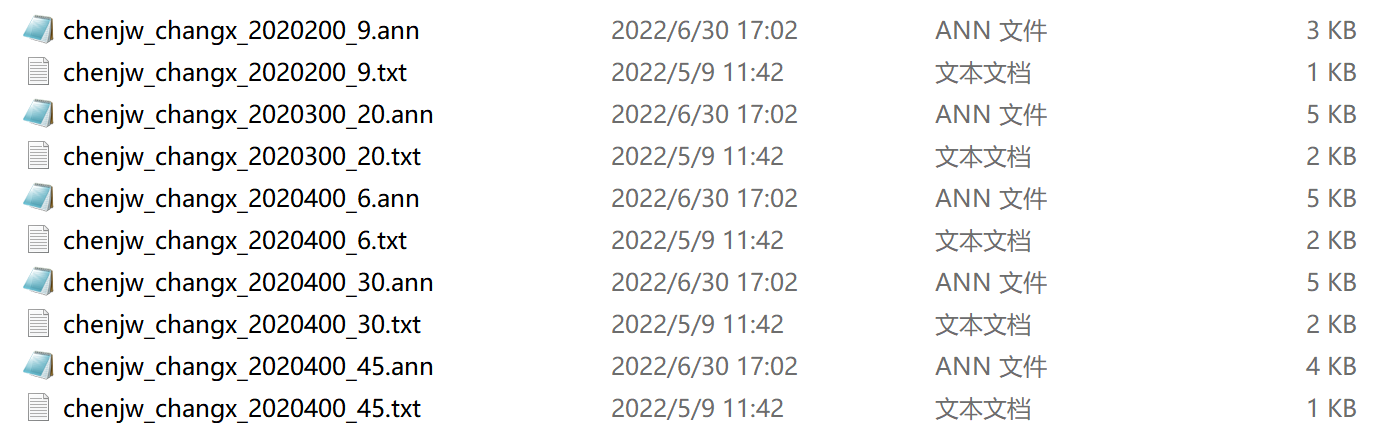
将测试集输入训练好的模型后，我们得到测试集的命名实体识别预测结果以及关系抽取预测结果。

图9 文件目录

其中，chenjw\_changx\_2020200\_9.txt测试样本预测结果如下图。

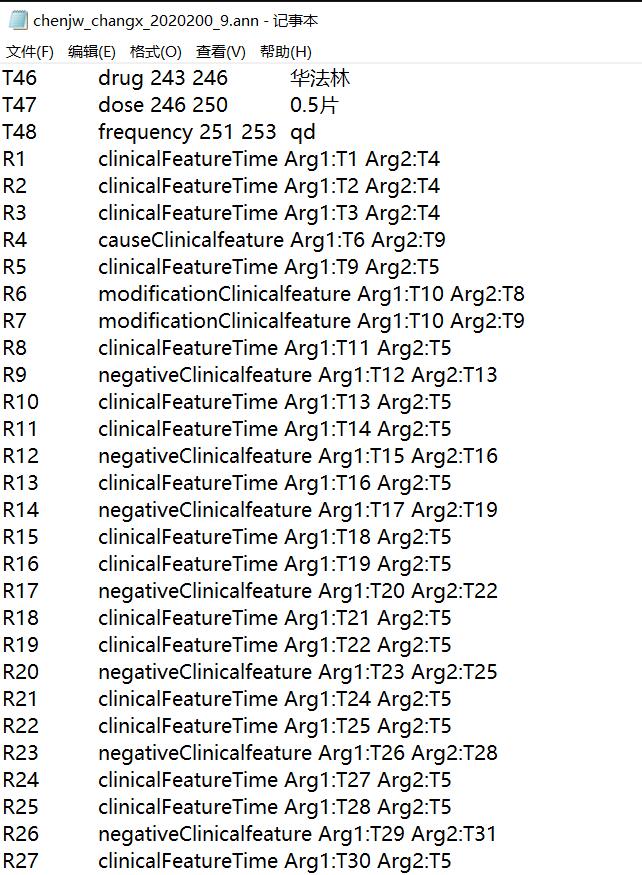
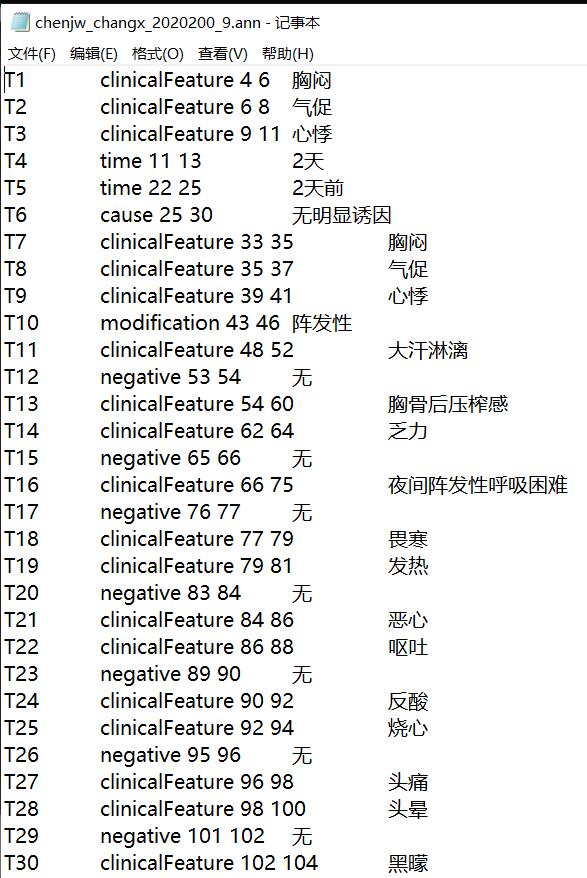


图10 部分结果展示图

**参考文献**

[1] 苏剑林. (May. 01, 2021). 《GlobalPointer：用统一的方式处理嵌套和非嵌套NER 》[Blog post]. Retrieved from <https://kexue.fm/archives/8373>

[2] 苏剑林. (Apr. 25, 2020). 《将“softmax+交叉熵”推广到多标签分类问题 》[Blog post]. Retrieved from <https://kexue.fm/archives/7359>

[3] 苏剑林. (Mar. 23, 2021). 《Transformer升级之路：2、博采众长的旋转式位置编码 》[Blog post]. Retrieved from <https://kexue.fm/archives/8265>

[4] Zhong Z, Chen D. A frustratingly easy approach for entity and relation extraction[J]. arXiv preprint arXiv:2010.12812, 2020.

[5] Wu L, Li J, Wang Y, et al. R-drop: Regularized dropout for neural networks[J]. Advances in Neural Information Processing Systems, 2021, 34: 10890-10905.