**基于PCNN的关系抽取**

本程序的设计模式采用创建型模式。

本程序任务：给定句子集，训练一个模型，使该模型可以将每个实例中的实体对的关系分类正确。

本程序的模型: 使用改良过的CNN(卷积神经网络)来进行特征提取，完成分类任务。

1. 对数据集进行处理。

通过建立Semeval类，来对原始数据集进行处理。将句子集中的每个实例，通过

Load\_rel()函数：获取实例的实体关系对。

Load\_w2v()函数：获取实例中每个句子中的单词的词向量表示。

Parse\_sen()函数：对每个实例句子进行处理，通过调用，get\_lexical\_feature(),get\_sentence\_feature(),get\_left\_word(),get\_right\_word(),get\_pos\_feature(),得到句子的语义特征集，句子特征集，两个实体的左右两边的单词的词向量，以及实体的位置信息。

处理完之后都会保存在对应的文件之中。

1. 建立模型PCNN

模型原理CNN，在PCNN.py中创建模型类。

类PCNN包含，\_init\_()函数，init\_model\_weight(), init\_word\_emb(), forward()函数。初始化类。

在\_init\_()函数中基于pytorch模型初始化了Word\_Embedding()的参数。，以及position\_Embedding的参数。并且定义了卷积层，在卷积层之后进行maxpooling操作，最后进行SoftMAx的分类操作。

\_init\_model\_weight()是对于word\_embedding的权重矩阵进行随机初始化。

在forward()函数中，将数据集分为不同批次batch，batch\_size使128，将之前Semeval类中处理好的数据进行拼接，并且喂入到PCNN模型进行计算

1. Main()

在main()函数中定义了Semeval类的对象train\_data。并且定义PCNN对象model，将train\_data喂入model。因为使用的pytorch模型会自动调用PCNN模型中的forward函数。

最后使用nn.CrossEntropyLoss计算损失，optim.Adam(model.parameters(), lr=0.001)来进行迭代的优化参数。

4.具体运行步骤

数据预处理: python dataset/semeval.py去生成npy文件

训练, 自动保存最优模型(未设计验证集)

python main\_sem.py train

其中 参数配置位于 config.py，可以直接指定修改，如: python main\_sem.py train --batch\_size=32

模型预测的结果位于semeval文件夹，使用./test.sh使用semeval官方的脚本测试，最后结果保存于res.txt

F1分值大概可以到80-81%左右，经过fine-tuning 大概到82-83%