WebInfo Lab2

PB18000239 何灏迪

PB18000221 袁一玮

实验背景

本次实验要求完成实体关系的抽取。实验给出6400条训练集以及1600条测试集，每条数据由一个句子和其实体关系标签以及其中的两个实体组成。

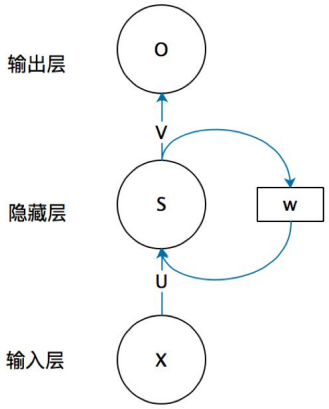
实体关系包括：

```
    "Cause-Effect": 0,
    "Component-Whole": 1,
    "Entity-Destination": 2,
    "Product-Producer": 3,
    "Entity-Origin": 4,
    "Member-Collection": 5,
    "Message-Topic": 6,
    "Content-Container": 7,
    "Instrument-Agency": 8,
    "Other": 9
```

实验中，我们将训练集分为6000+400的训练集与验证集，在验证集上进行本地的测试。在此基础上，我们进行了许多尝试，其中一些尝试获得了不错的效果，另外一些则由于考虑不够充分最终没有得到更好的结果而被舍弃。下面首先介绍最终实现的方法，再介绍过程中进行的许多尝试。

实验原理

最终实现的方法中，我们使用了**Bi-RNN**（双向循环神经网络）进行训练。

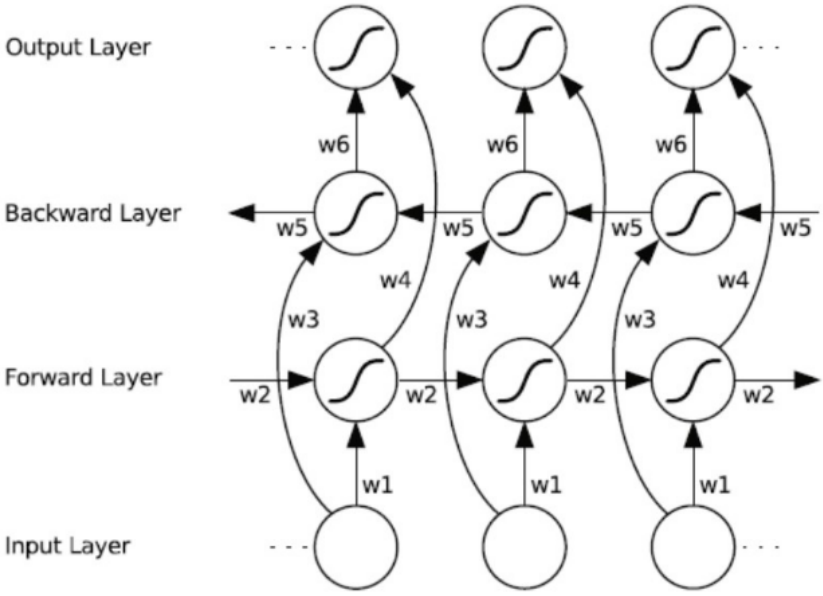


$$O_t = g(V \cdot S_t)$$
$$S_t = f(U \cdot X_t + W \cdot S_{t-1})$$

St的值不仅仅取决于Xt，还取决于St-1

上图为普通**RNN**的基本结构。相比起普通的神经网络，该结构神经网络的输出将与之前状态中神经网络中的结果相关。由此，该网络可以对序列信息进行分析，并生成最终的预测结果。相比**LSTM**，该网络的结构更加简单，没有复杂的门控结构。

而本次实验中，我们使用了双向**RNN**进行训练，其结构如下：



这允许该网络更充分地提取句子整体的信息，从而得到更准确的结果。相比起**LSTM**，该方法在实验中获得了略微更优的准确度。

在确定神经网络结构的基础上，仍需考虑一些其他问题。最主要的，如何将词语转入神经网络中进行使用。显然，使用独热码表示每一个词语将会形成极大量无用的参数空间。不难想到使用`word2vec`作为该部分的`word embedding`。

我们选取了GloVe作为我们的`word embedding`，并选取其中每个词向量为300d的表示形式。通过直接使用提前训练的语义层，相比起自行训练这个空间的参数，模型在验证集上可以获得10%~15%左右的准确率提升，可见其关键性。

此外，在句子中存在许多主观上认为无法帮助识别的词语，如形容词等。根据观察，当前的nlp工具已经可以提供比较成熟准确的词性分类工具。过程中使用了StanfordCoreNLP进行词性的分类，并剔除其中的形容词，之后获得了约3%的准确率提升。由于形容词并不影响句子整体结构的完整性，而其他词性的词语，如冠词等，可能影响句子本身的完整，怀疑对网络将产生负面的影响，这里没有进行进一步的尝试。

程序实现

程序基于Tensorflow 2.2进行实现。

程序首先对训练集和测试集进行处理，将其中的词语依次编号，并在前文提到的embedding文件中进行相关的查找，最终获得`embedding matrix`。此外，该部分将句子的长度进行了标准化，用空白词补齐为30个单词，超出部分截去。

训练时，首先载入之前预生成的`embedding matrix`， 之后进行训练。其网络结构具体如下，其中包括Dropout层防止过拟合的情况：

```
self.model = keras.Sequential([
            embedding,
            layers.GRU(64, return_sequences=True, dropout=0.5),
            layers.GRU(64, dropout=0.5),
            # WordAttention(),
            layers.Dropout(rate=0.5),
            layers.Dense(64),
            layers.Dropout(rate=0.5),
            layers.Dense(10),
            layers.Softmax()
        ])
```

经过100个epoch的训练模型在验证集上的准确率不再增长，达到65%。

在测试平台进行测试，最终得到的准确率为54.17%，较大幅度的低于验证集准确率，暂时没有发现具体的原因。

其他尝试

1. Benchmark：贝叶斯

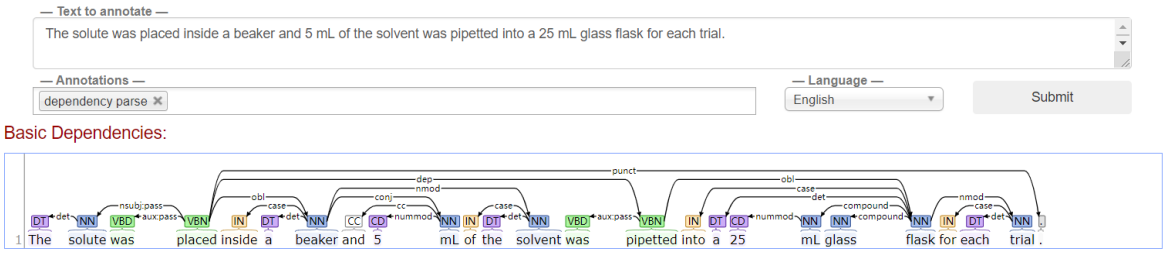
在实验最初阶段，我们首先使用贝叶斯方法进行最简单的分类尝试。该方法基于每个词语和类别共现的概率进行分析，对训练集中的所有词语进行这样的统计，得到十维的总和为1的向量。之后，将测试集中的句子中的每个词语的向量相加，取其中值最大的一维作为最终的结果。

同样如前所述使用去除形容词的策略进行优化后，在验证集上得到的准确率为44%。若不进行任何优化，该准确率为40%。

2. 利用关系树进行抽取

在实验最初的构想中，希望可以通过关系树找到 (名词) - (关联词或短语) - (名词) 的结构，并基于该结构进行网络的训练，从而可以对任意的名词对进行关系的分析。对于句子中的各个名词对，取其中得到的关系置信度最高者作为最终的结果。使用这种方法将自然而然地同时解决实体识别的问题。

在课程PPT中提到使用依存树进行抽取。我们使用的`StanfordCoreNLP`提供了依存树、语法树的接口（这也是我们选择该库的原因）。但在实际的使用过程中发现，依存树的准确率较低（误识别率大>10%），语法树准确率稍好，最简单的词性分析则具有较高的准确率。基于实际的使用情况，我们认为使用依存树进行分析实际不太可靠。而如果使用语法树分析，需要考虑变化多样的句式，其中涉及非常复杂的模式匹配设计，在尝试后选择放弃。



上图为依存树分析错误的典型例子之一：and应该连接前后两个句子，而非"beaker"与"5 mL of the solvent"。

3. Others的分类问题

在验证集上进行实验，发现在所有类别中，`Others`的分类的准确率仅为20%，召回率也同样远小于其他各类别。

由于`Others`可能并不具有显著的的类别特征，对其进行分类可能无法达到很好的准确率。具体思考分类过程，考虑改为九分类任务，并将`Others`在训练过程中的标签改为`(0,0,0,0,0,0,0,0,0)`，在最终验证/测试时使用置信度阈值来划分`Others`类。由于这样的设定，无法再使用`crossEntropy`进行计算，改使用`squaredError`来作为loss函数。

经实际的实验，这样操作的效果实际上并不优于原策略，在不同的阈值下测量结果，在结果最佳时也仅与原准确率几乎一样。考虑原因，既可能是由于loss函数改变后效果不如原先，也可能是这样的分类策略实际上已经被之前的网络学习到，没有办法在同样的网络结构上仅仅通过这样的人为策略来提高准确率，要得到更好的准确率可能需要在网络结构本身上做改进。

4. 句子尾部截去

该部分进行的操作如下：

通过找到训练集中两个实体中位置较后者，将在其之后的所有词语截去作为一种数据增强手段。

该操作基于这样的考虑：句子中用于描述两个实体的关系的词语通常出现在较后实体之前。此时，其后的所有单词就属于无效的干扰信息。通过进行截取，可以去除数据中的无效部分，设想中应该与去除形容词具有相似的效果。

这样的操作同时也是对之前无法抽取 实体对+连接短语 结构的一种弥补，希望通过这种方式能将训练集中实体对的信息使用起来，辅助后续的识别。设想中，在测试时对同一个句子按其中各个名词的位置进行多种不同的裁剪，取其中得到最高置信度的标签作为句子整体的识别结果。

在验证集上，我们进行了较简单的实验：验证集上同样按groundtruth进行裁剪，之后识别。

经过这样的数据增强操作，其在验证集上的结果可以显著提升至75%，是之前所有方法都无法达到的。

但是在测试集上，由于没有groundtruth，必须尝试多种裁剪策略，经过综合，最终的准确率仅为51%，落后于原方法的54%。这说明训练出来的网络具有比较好的"在实体对已知的情况下获得实体关系"的能力，但并没有很好的分辨实体对的能力。

这种方法或许仍可以进一步优化，如在训练集中加入一些不正确的实体对关系（即对应的错误裁剪方法）作为负面样本，但是在本次实验中限于时间的关系无法进一步探索。

5. Word Attention

该方法希望能使用注意力机制优化网络的分类效果。在二分类任务（如影评是正面/负面）中，该方法的使用具有显著的效果。

在本次实验中同样进行了类似的尝试，使用Word Attention + RNN，但最终没有显示出很好的结果。通过对一些特定案例的分析，我们发现Word Attention并没有起到想象中的效果。如`Cause-Effect`关系中，`cause`这个词本身理应获得很高的注意力权值，但实际上没有出现。

最终在验证集上，其效果稍逊于原先的`Bi-RNN`方法。

实验效果

在测试平台提交得到最高准确率为54%。