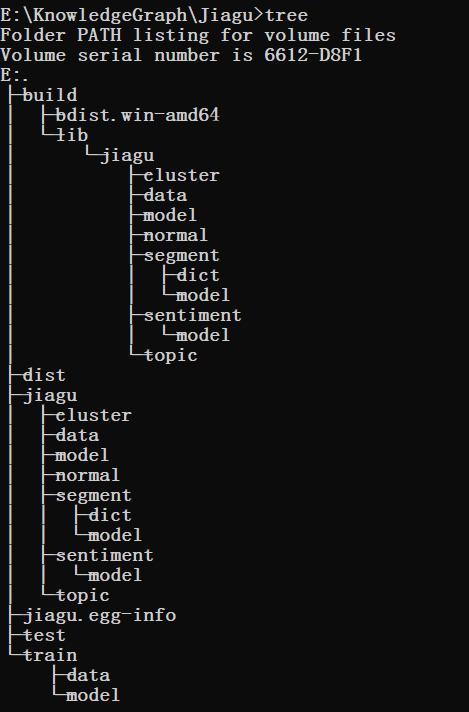
同志们，试了三个模型，DeepDive和OpenRE的配置都非常复杂，而且要求Linux系统，我不打算再装系统、配环境了。

我个人的工作没有做好，没有很好地协调团队解决问题，导致现在还没完成项目，我对此负责。原因客观与主观都有，就不解释了。

这个Jiagu模型，配置起来比较简单。图1是它的项目组成。

Figure 1 项目结构图



Jiagu文件夹下面是工程代码，test文件夹下是用来测试的几个工具，train文件夹是供自己训练数据用的，dist文件夹是部署时产生的。

介绍一下文件夹train的结构，perceptron.py是对data中的数据进行训练和测试的。需要训练的数据放在train.txt中，测试的数据放在test.txt中。如果对机器学习有基本的了解，很显然这是常规的机器学习训练数据的文件结构。在对train.txt的数据训练以后，需要使用test.txt中的数据对其进行测试，看训练效果如何。

train.txt与test.txt本质上是一样的，都是人工打过标签的数据，算法在使用train.txt学习之后，会得到一个模型，将这个模型对test.txt进行测试，检查模型的检测结果与人工标记的匹配程度，可以得到本模型的一系列参数：准确率(Accuracy)、平均准确率(Average Per-class Accuracy)、精确率-召回率(Precision-Recall)与对数损失函数(Log-loss)等。

我粗略浏览了一下样例中train.txt的数据量，一共2357096行，每一行一个字符含标签。我们的数据如果要人工打标签的话，一定非常费时费力。因此，权宜之计是想用那个数据挑一部分出来，先把系统搞出来，我再把流程详细地走一遍。对数据打标签的事情，想麻烦周余来帮我，刘宏鑫用现有的数据搞可视化。

我测试了，训练代码可以直接跑，只要标签打好，就可以上我们自己的数据。我还想再挣扎一下DeepDive和OpenRE，虽然我们不用两者，但写论文时说不定可以用实验结果来个对比啥的。

大二暑假时，我们已经用数据python爬取数据，spark处理过了。粗略一点，分词就当做是知识（关系）实体的来源（虽然用Jiagu未涉及到）。搭建完图谱、給出“产品使用说明”、写论文，咱们就可以交差了。

以上。