**基于深度学习的自然语言处理**

**第7次实验报告**

**一、实验名称**

实现命名实体识别和关系抽取的整个流程

**二、实验日期**

2020.12.19 – 2021.1.1

**三、实验目的**

（1）熟悉命名实体识别的流程

（2）熟悉关系抽取的流程

**四、实验数据**

提供的开源的实验数据

**五、实验步骤及结果分析**

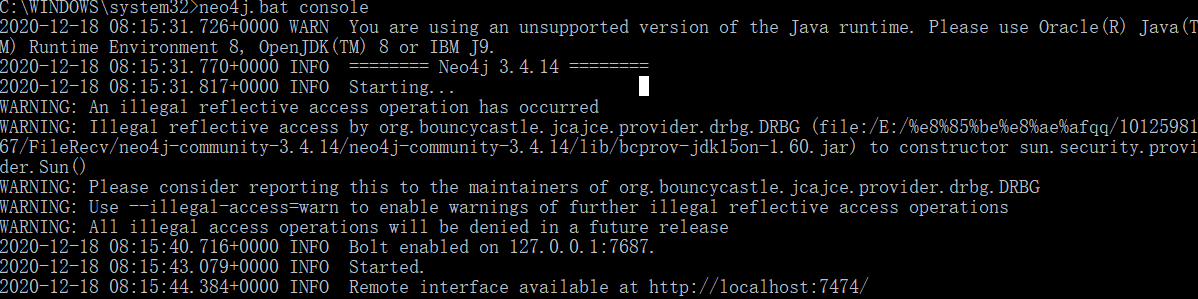
5.1 安装基本环境：

确保安装好python3和Neo4j（任意版本），安装一系列pip依赖： cd至项目根目录，运行 sudo pip3 install -r requirement.txt

5.2 导入数据：

将hudong\_pedia.csv导入neo4j：开启neo4j，进入neo4j控制台。将hudong\_pedia.csv放入neo4j安装目录下的/import目录。

neo4j.bat console

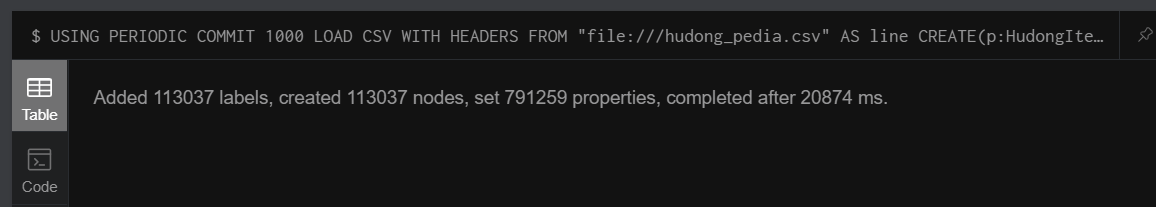


我的NEO4J地址E:\腾讯qq\1012598167\FileRecv\neo4j-community-3.4.14\neo4j-community-3.4.14\import

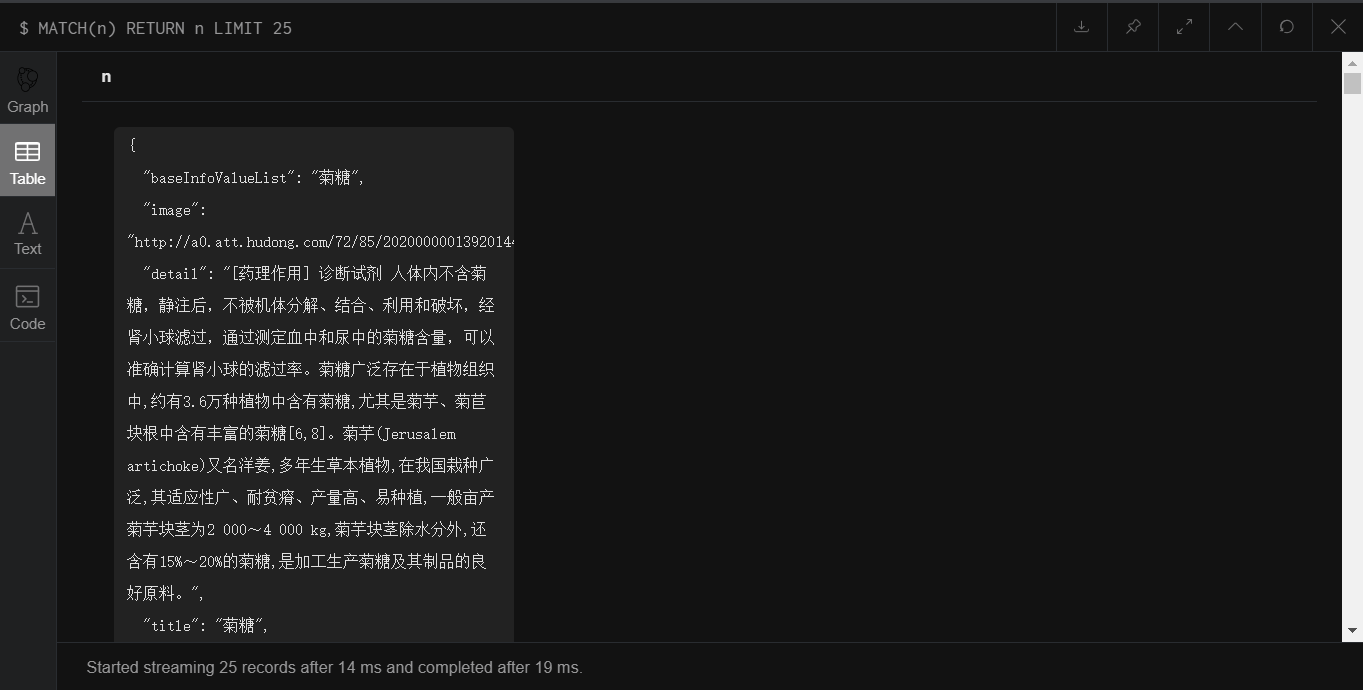
我的NEO4J内容备份E:\neo4jcsv

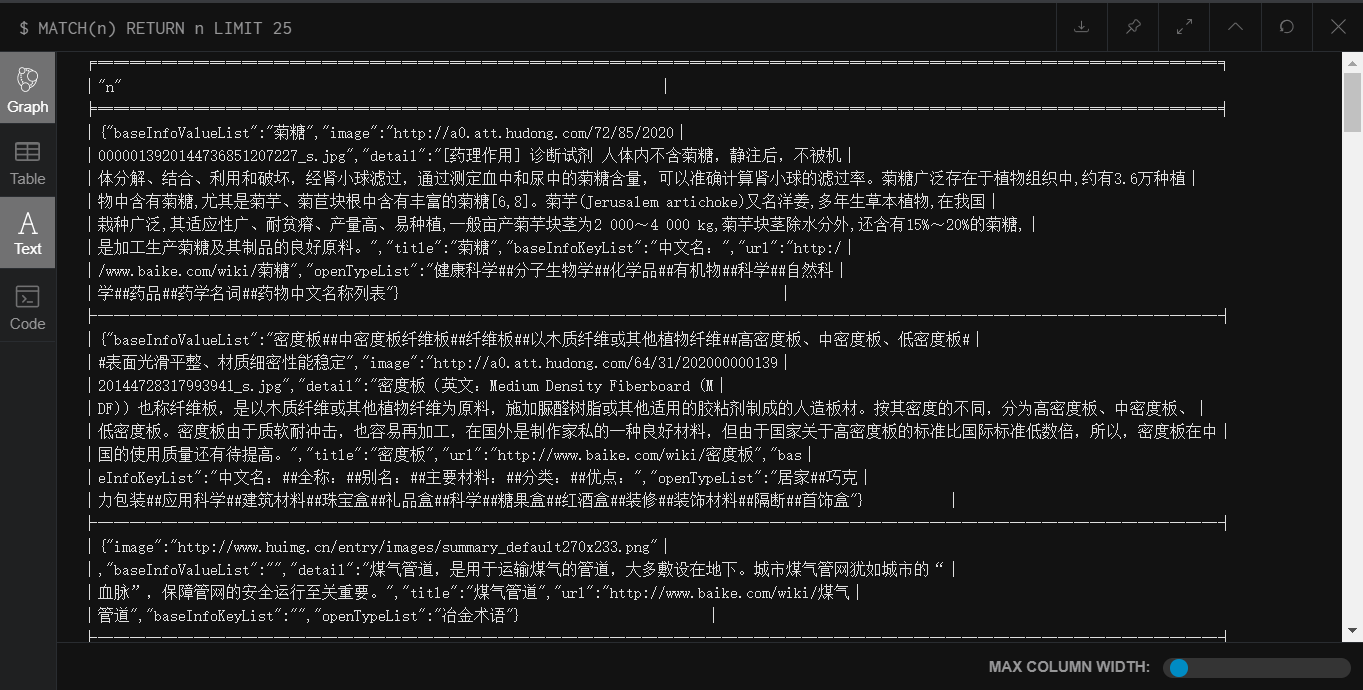
过程性教程，代码见：blog.csdn.net/weixin\_41104835/article/details/89213327

##### ****导入节点HudingItem数据****









##### ****导入节点NewNode数据****

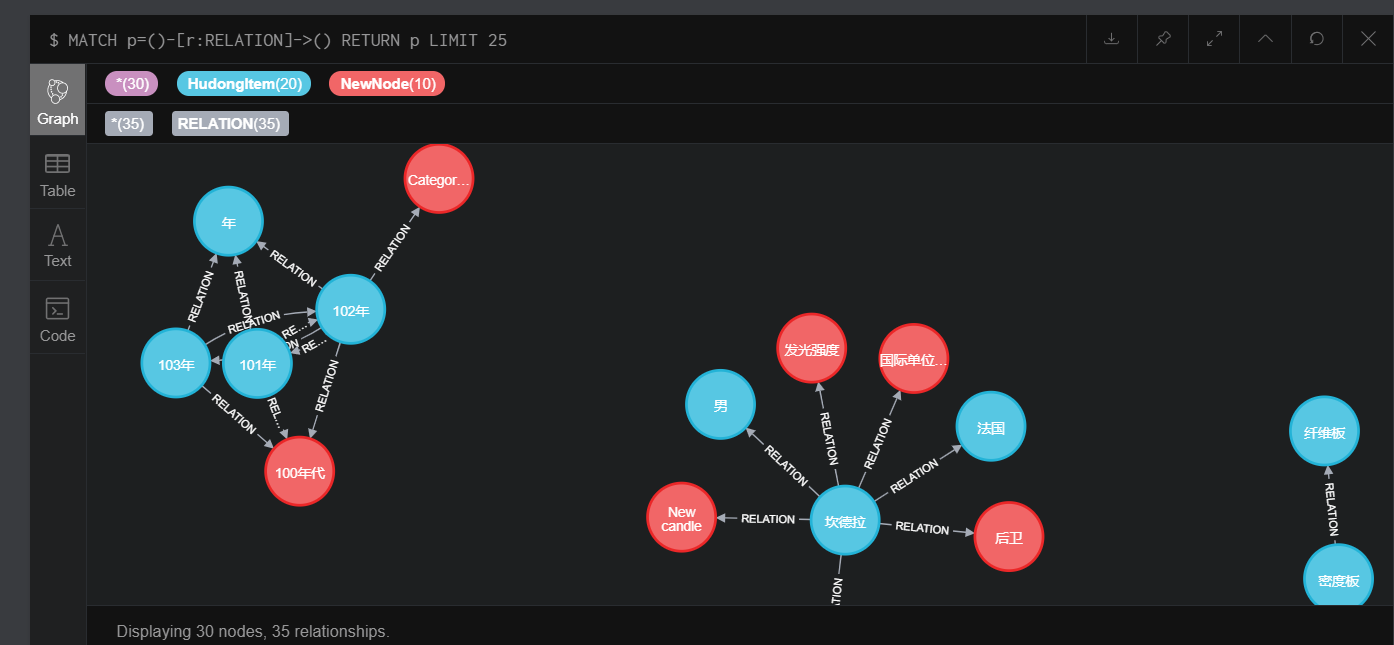
导入后点击Node Labels中的New Node



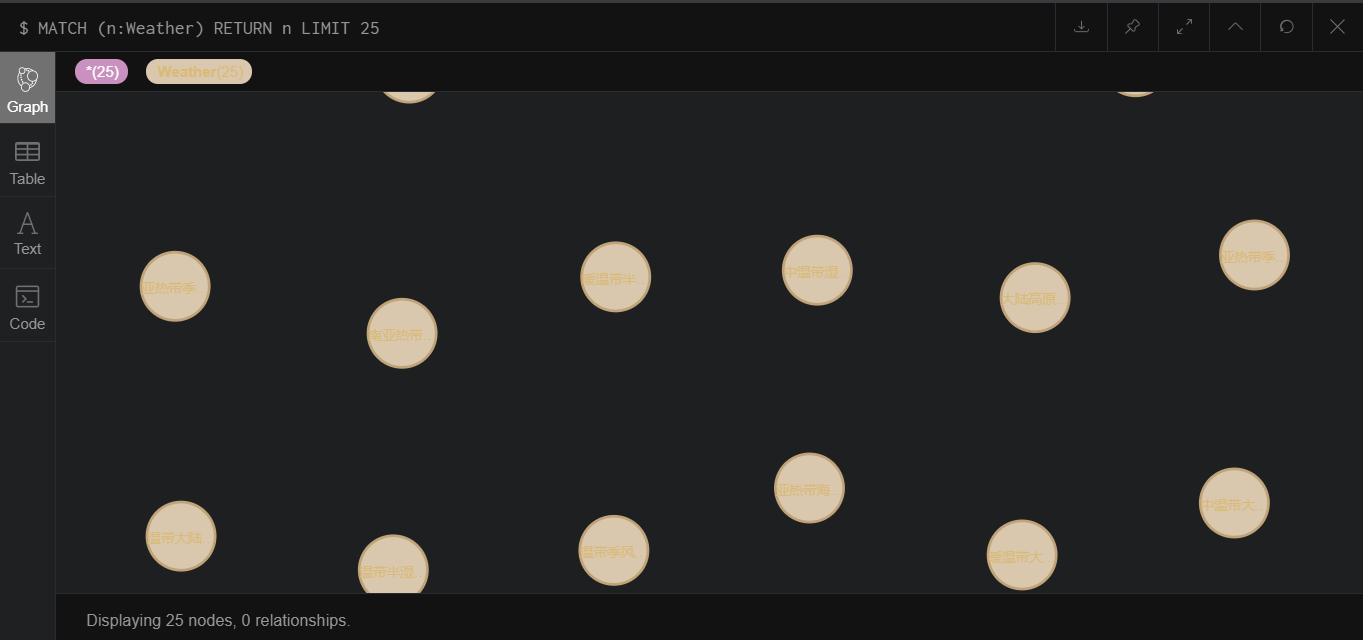
##### ****导入hudongItem节点和NewNode节点之间的关系RELATION****

**导入实体和属性之间的关系，添加到关系RELATION中**

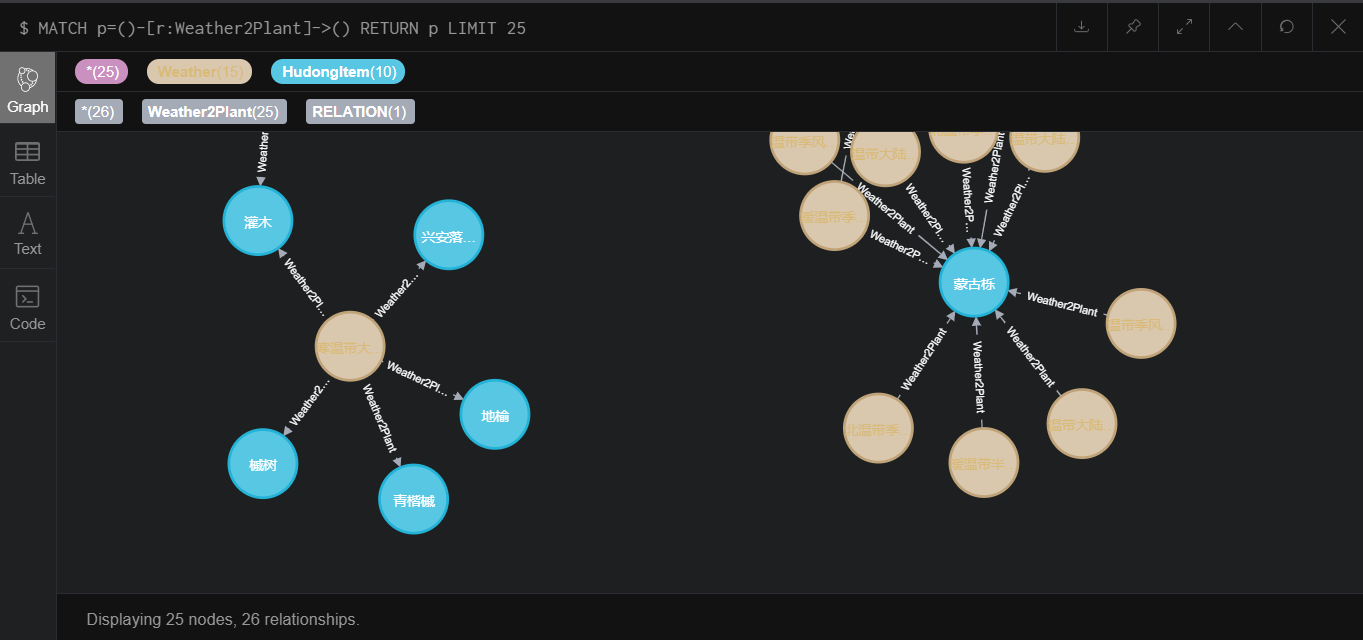
**点击Relation Types中的Relation**



##### 导入节点Weather数据

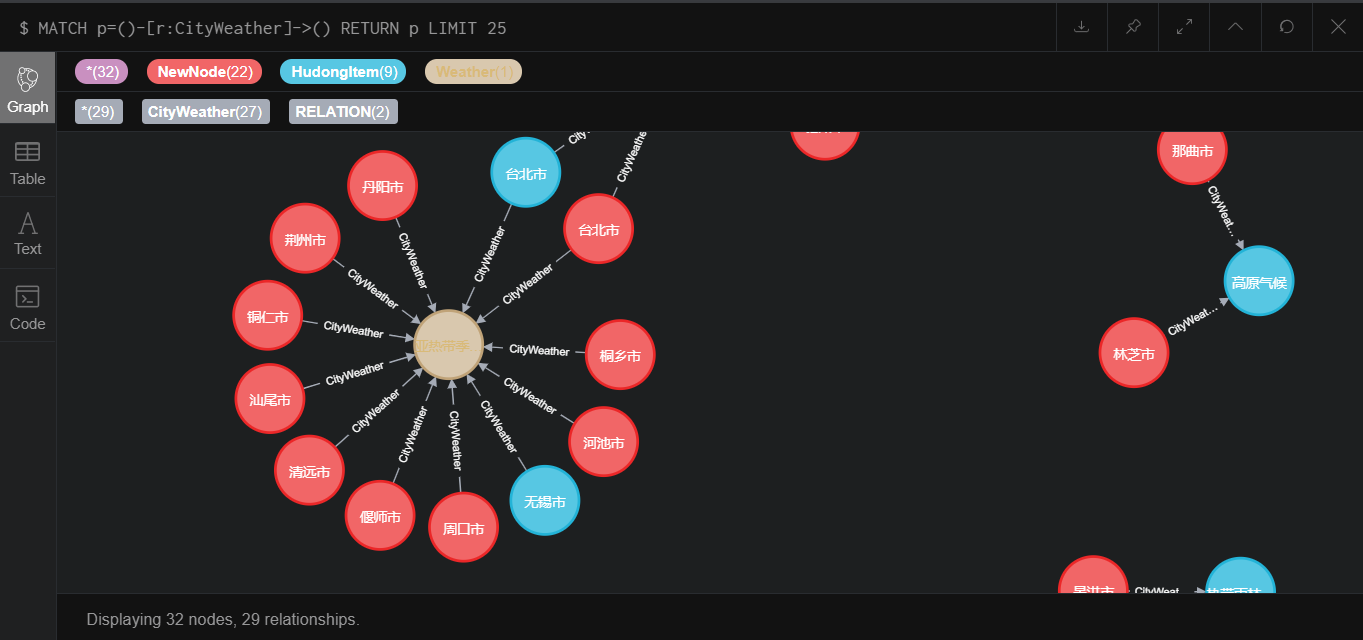


##### 导入Weather节点和HudongItem节点（植物）的关系Weather2Plant

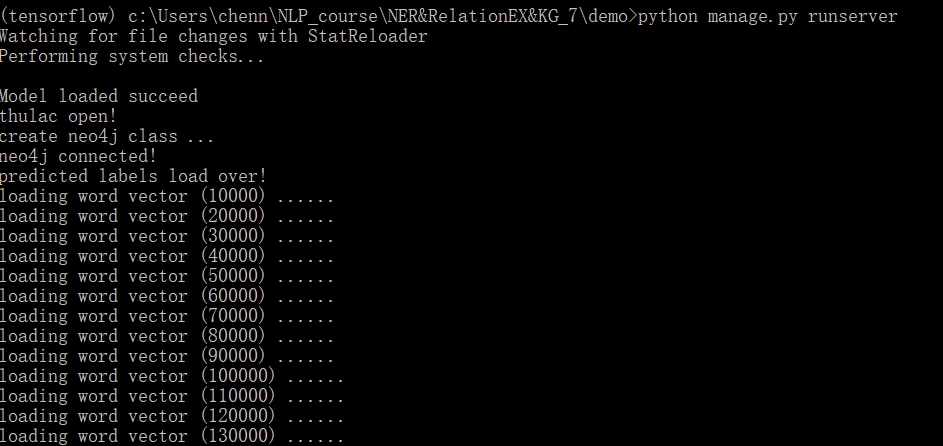


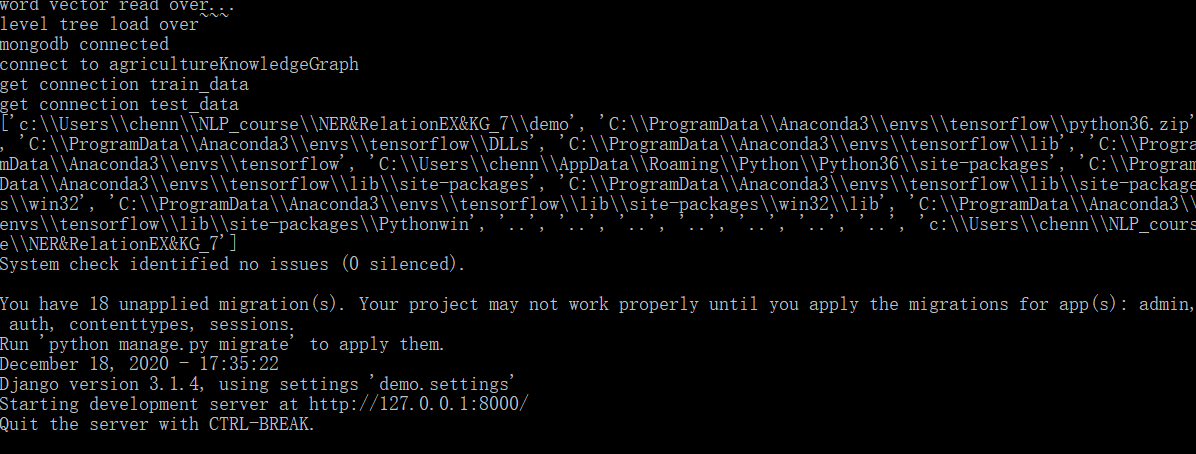
##### 导入城市（city）和weather节点之间的关系



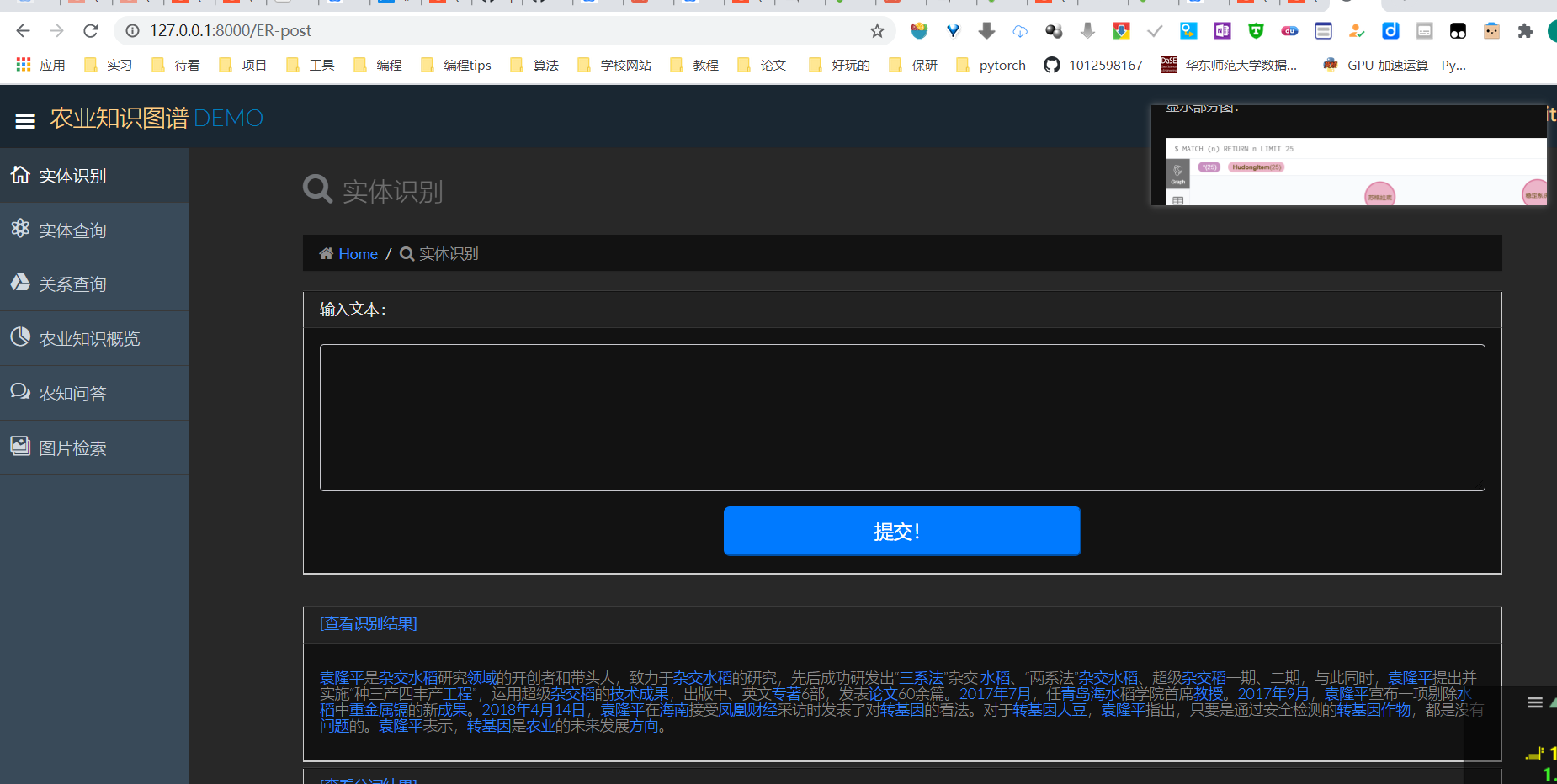


#### 启动django服务





进入前端，输入一段文字并测试效果



点击实体展示超链接



5.3下载词向量模型

（如果只是为了运行项目，步骤2可以不做，预测结果已经离线处理好了）<http://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.zh.zip>将wiki.zh.bin放入 KNN\_predict 目录 。

参考链接

<https://github.com/qq547276542/Agriculture_KnowledgeGraph>

https://blog.csdn.net/weixin\_41104835/article/details/89213327

**六、实验感想**

实体关系抽取一般流程

场景假设通过枚举所有实体对之间是否有关系建模，所以需要产生所有实体对之间的关系数据(有关系与无关系)。 正样本与负样本：例子：句子：实体1xxx实体2,xxx实体3。由上述句子可以生成实体1-实体2，实体1-实体3，实体2-实体3，共三个备选样本，假设实体1-实体2为一个正样本(即这两个样本之间有关系)，那么剩余的两种组合即为负样本。当然，这种样本制作方式会带来很高的正负样本比。 数据分析：通过数据分析，可以得到，两个有关系的实体有71%可能在一个句子中，有21%的可能分布在相邻的两个句子中。实体含有的信息：实体名字与其表示的向量，实体位置， 数据样本生成：如果通过常规的句子切割方式，一来诸如半角符号的存在，无法判断是小数点还是句子结束符。二来通过这样的方式生成的样本长度均值在150左右。本方案通过将【实体1xxx实体2】这样的结构作为一条样本，但是，考虑到上下文的重要性，又在这样的样本前后追加20个字符【xx实体1xxx实体2xx】。最后的文本长度控制在100左右。中间文本超阈值范围的不计。为了再次降低正负样本比例，清除无关实体构成的句子。 特征制作：词性特征：将字的所属词的词性进行embedding.词边界特征：通过BIOE标记法获得的标记进行embedding.标记实体所在位置。实体位置标记1，其它字标志为0。由于疾病最后一个字往往比较重要，所以将标注方式换为B(Begin),I(Inside),O(Outside),E(End),S(Single)。多了E和S。实体对之间的距离作为关系。 采用词向量：中文的分词困难，特别是医疗领域含有大量的专业词汇。对于稀少的专有名词的训练很艰难。大量的未登录词，词的覆盖率只有55+%，而字的覆盖率达到90+% 词向量的选择：Word2Vec获得的词向量本次实验效果优于Glove.但是由于负样本的上下文环境多种多样，所以引进了动态词向量。动态字词向量含有上下文的语义信息。由于 Word2Vec 和 Glove 学习得到的词向量是固定不变的，即一个字只有一种词向量，显然不适合用于多义词。而 ELMo 算法使用了深度双向语言模型 (biLM)，只训练语言模型，而词向量是在输入句子实时获得的，因此词向量与上下文信息密切相关，可以较好地区分歧义。ELMo 是一种动态词向量算法，在大型的语料库里训练一个 biLSTM (双向LSTM模型)语言模型。下游任务需要获取单词词向量的时候，将整个被向量化的句子输入biLSTM（在 ELMo 中使用了 CNN-BIG-LSTM 生成的词向量），利用 biLSTM 的输出作为单词的词向量，包含了上下文信息。可以理解成，biLSTM 是一个函数，函数的输入是一个句子，输出是句子中单词的词向量。biLSTM 中不同层得到的词向量侧重点不同，输入层采用的 CNN-BIG-LSTM 词向量可以比较好编码词性信息，第 1 层 LSTM 可以比较好编码句法信息，第 2 层 LSTM 可以比较好编码单词语义信息。通过多层词向量的融合得到最终词向量，最终词向量可以兼顾多种不同层次的信息。 Loss的设置：由于负样本的文本模式复杂多样，无规律可循，所以，对于负样本作为一个类别去预测就会很难提取这个类别的共性特征。因此，将10+1类的预测改为10类的预测，然后设置阈值来区分是不是负样本，这样提高了召回率。预测阶段，设t是模型正负样本的判断阈偵：1、如果有预测得分人于等于t，则选取最大得分对应的类别为预测类别；2、如果预测得分均小0，则该样本为负类。 图谱的功能查询某一个实体的属性与对应的关系实体。比如疾病与其对应相关的药物，对应的检查方法。关于药物也可以推理出它的用量用法属性。