依赖

```
python 3.6
networkx
jieba
tensorflow
keras
pymysql
pandas
Levenshtein
sklearn
gensim
```

需要修改的配置

unitils.py 的第31-39,51行连接数据库的配置需要修改

将数据导入MySQL

创建名为"ccks"的mysql数据库

需要将三个文件 使用脚本"pkubase.ipynb"导成右侧相应的表

```
pkubase-triples.txt         -> pkubase
pkubase-types.txt           -> pkutype
pkubase-mention2ent.txt     -> pkuorder
```

二阶数据

pkuprop

```
create table pkuprop select prop,inverse(prop) as rprop,count(0)
from pkubase  group by prop order by count(0) desc
```

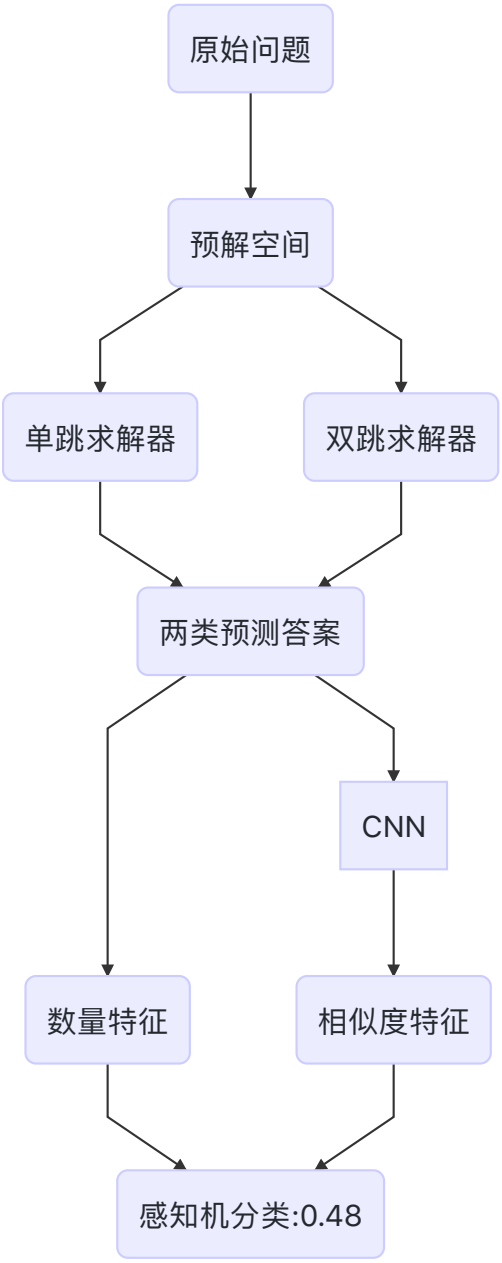
可以直接从pkuprop.sql导入

pkuvalue

```
create table pkuvalue select `value`,count(0) from pkubase  group by
`value` order by count(0) desc
```

这个计算是在Hive on Spark平台完成的.

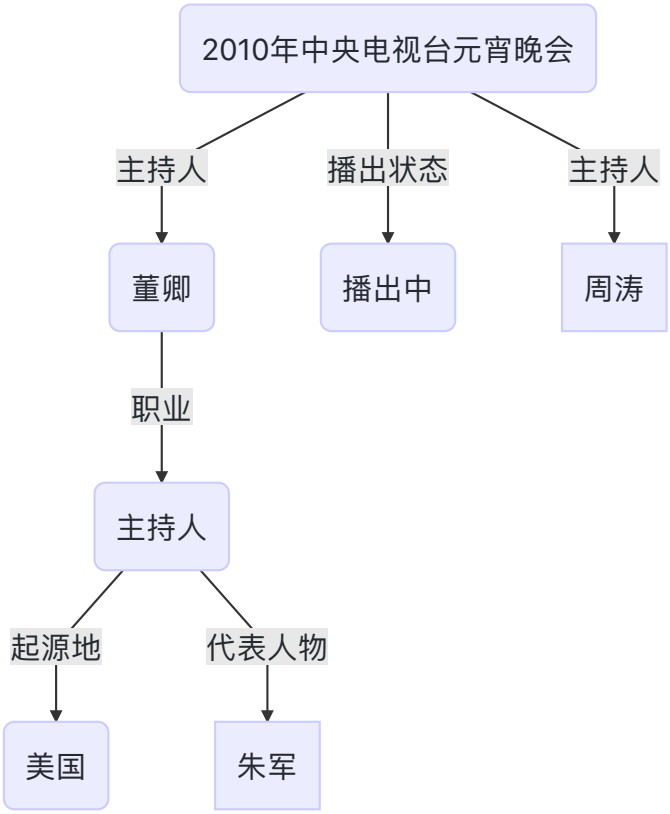
模型构造



预解空间

问句中的实体基本可以使用pkuorder进行实体连接.

预解空间生成方法是:
对可能的实体,去数据库搜索它的子节点或者父节点作为第一跳节点,然后搜索第一跳节点的子节点作为第二跳节点.不同实体入口的图是分别存储的.
例如对于问题 seq = "董卿主持的正在播出的节目是"
我们把 seq[i:j] (0<i < j-1< len(seq))作为关键字取搜索数据库,首先会发现"<董卿>"这个实体,然后将董卿作为三元组(SPO)中S和O分别取搜寻节点,并用network中的有向图存储起来,然后再以这些新加入的节点作为S取搜索O,这样以"<董卿>"作为根节点的一棵预解树就完成了.



这个例子解释了为什么在第一跳时要使用两个方向.
预解树通常不止一颗,对这个问题来说"<节目>"也会是一颗预解树.

生成预解空间

调用unitils模块中的get_ans方法可以获得问题的预解树.

需要表

```
三元组数据   : pkubase
实体链接     : pkuorder
子节点数据   : pkuvalue
```

```
from unitils import get_ans
import re
question =
open("task4coqa_test.questions","rb").read().decode("utf8").split("\
r\n")
final_test_cache = []
for num,ss in enumerate(question,0):
    ss = re.sub("(q\d{1,4}:|? )","",ss)
    print("-"*130)
    print(num,ss)
    ans = get_ans(ss,10)
    final_test_cache.append((num,ans))
```

或者使用缓存文件

```
final_test_cache_07_19_17_35.bin
```

数量特征 相似度特征

训练模型

使用train_model.ipynb即可训练模型,至少需要pkuprop这张表.

双跳问题和数量特征

以双跳问题seq = "张柏芝、谢霆锋合作的动作电影里的服装是谁设计的"为例

构造特征

首先我们会得到以张柏芝,谢霆锋为入口的预解树.在每棵预解树上的形如(S1-P1-O-P2-S2)的子树,其中S1 = "<张柏芝>" or "<谢霆锋>".我们使用如下特征,计算这些(seq,S1+P1+P2)的jaccard距离,(seq,S1+P1+P2+O+S2)的公共字符个数hint,(seq,S1+P1+P2)的编辑距离Les,而最重要的是我们引入了一个叫做placeholder的定义,即问句中"是谁,是什么，哪里，何时，哪个，有什么"这类提示词.观察发现,这类提示词附近往往有着提示答案的重要信息,对这个问题来说，"设计"，就是一个很重要的信息,那么子树(S1-P1-O-P2-S2)的P2如果包含设计,那么它的评分应该很高.具体来说我们为P2在句子中找一个子串,使得该字串能与P2的头部或者尾部重合,然后将字串距离placeholder的距离作为特征.而这些特征会作为之后感知机的数量特征输入.

节点合并

然后我们得到了一系列(S1-P1-O-P2-S2,jaccard,hint,les,placeholder)的待选答案及相关信息,(我们将S2作为可能的答案节点).但注意到,题目问的是张柏芝,谢霆锋共同主演的电影,那么"<无极>"作为桥接节点O,实际上会和作为S1的"<张柏芝>","<谢霆锋>"相连.遇到这种桥接节点O和S2相同的情况,我们会合并这两个待选答案,重新计算相关特征

剪枝

由于之前的实体链接是用的穷举匹配的方法,到了这一步实际上待选答案的规模是非常大的,如果在这上面训练分类器,会直接导致类别不均衡.受到决策树模型的启发,我们使用启发式的法则进行剪枝.以某个或几个指标的均值mean作为基准,去搜索某个倍率a,使得大于a*mean的答案小于 80.这样我们控制了训练的正反例比率

```
# 搜索能返回 合理规模 的 候选 答案集
    for sup in range(1,30,1):
        index = ans[:,3] > 0.1*sup*ans[:,3].mean()
        if sum(index)< 80 and sum(index)>0:
            tmp = ans[index]
            break
```

单跳问题

单跳问题与双跳求解方式类似.由于有了节点合并,形如北京大学出了哪些文学家之类的问题,也被视为一个单跳问题.

相似度特征

使用纯粹的数量特征使得我们实际上丢失了大量信息,因此有必要为最后的感知机输入新的特征,这里是用的CNN来衡量候选答案的路径(S1-P1-O-P2),(S1-P1)与问句seq之间的相似度.

```
原始数据 :this_feature_07_20_09_39.bin
训练数据 :train_x_07_20_09_40.bin
标准化   :scaler = scaler_07_18_18_30.bin
模型文件 :sim = final_cnn.bin
```

训练数据

每个问题用单跳,双跳,去求解都会得到答案和相关路径path,将(path,seq)做为X,答案的F1作为Y进行训练.

感知机分类

运行 感知机分类.ipynb即可 生成result_ss.txt文件.中间结果nnn文件,total_test_feature,用于最终融合.

```
模型文件:final_model_07_18.bin
输入特征:test_fff_feature_07_20_09_38.bin
```