资料来源：**[用Bi-GRU和字向量做端到端的中文关系抽取](http://www.crownpku.com/2017/08/19/%E7%94%A8Bi-GRU%E5%92%8C%E5%AD%97%E5%90%91%E9%87%8F%E5%81%9A%E7%AB%AF%E5%88%B0%E7%AB%AF%E7%9A%84%E4%B8%AD%E6%96%87%E5%85%B3%E7%B3%BB%E6%8A%BD%E5%8F%96.html)**

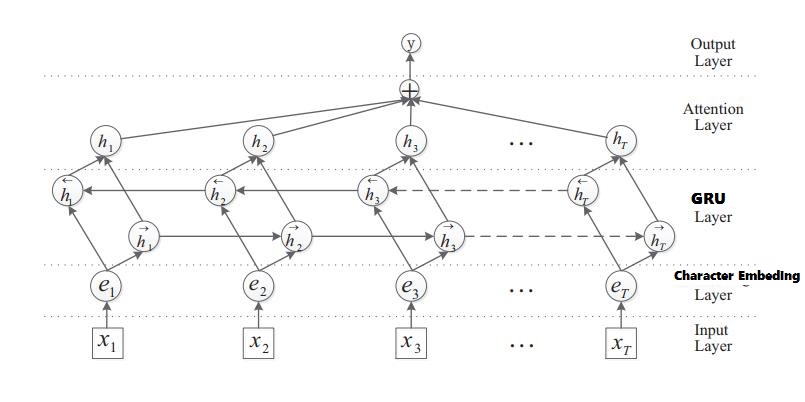
实体识别和关系抽取是例如构建知识图谱等上层自然语言处理应用的基础。关系抽取可以简单理解为一个分类问题：给定两个实体和两个实体共同出现的句子文本，判别两个实体之间的关系。

使用CNN或者双向RNN加Attention的深度学习方法被认为是现在关系抽取state of art的解决方案。已有的文献和代码，大都是针对英文语料，使用词向量作为输入进行训练。这里以实践为目的，介绍一个用双向GRU、字与句子的双重Attention模型，以天然适配中文特性的字向量(character embedding)作为输入，网络爬取数据作为训练语料构建的中文关系抽取模型。

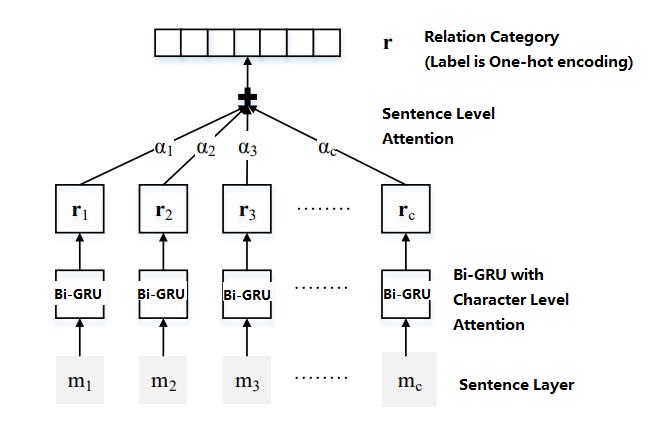
代码主要是基于清华的开源项目[thunlp/TensorFlow-NRE](https://github.com/thunlp/TensorFlow-NRE)开发，感谢！

双向GRU加Dual Attention模型

双向GRU加字级别attention的模型想法来自文章 “Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification” [Zhou et al.,2016]。这里将原文的模型结构中的LSTM改为GRU，且对句子中的每一个中文字符输入为character embedding。这样的模型对每一个句子输入做训练，加入字级别的attention。



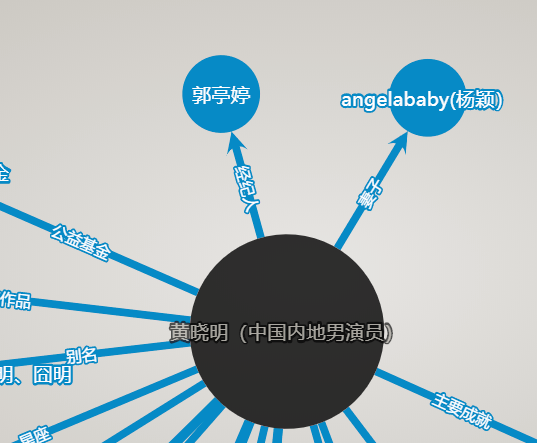
句子级别attention的想法来自文章 “Neural Relation Extraction with Selective Attention over Instances” [Lin et al.,2016]。原文模型结构图如下，这里将其中对每个句子进行encoding的CNN模块换成上面的双向GRU模型。这样的模型对每一种类别的句子输入做共同训练，加入句子级别的attention。



语料获取

中文关系抽取的公开语料比较少。我们从distant supervision的方法中获取灵感，希望可以首先找到具有确定关系的实体对，然后再去获取该实体对共同出现的语句作为正样本。负样本则从实体库中随机产生没有关系的实体对，再去获取这样实体对共同出现的语句。

对于具有确定关系的实体对，我们从[复旦知识工厂](http://kw.fudan.edu.cn/)得到，感谢他们提供的免费API！一个小问题是，相同的关系label在复旦知识工厂中可能对应着不同的标注，比如“夫妻”，抓取到的数据里有的是“丈夫”，有的是“妻子”，有的是“伉俪”等等，需要手动对齐。



得到实体对，获取实体对共同出现的语句，我们直接借助百度、搜狐、360等的新闻搜素功能，抓取实体对共同出现的新闻在搜索列表中的摘要。

我们还从一个开源项目[Roshanson/TextInfoExp](https://github.com/Roshanson/TextInfoExp)中获取了不少关系抽取语料。感谢！

模型训练

下面进入实战，Clone和运行[中文关系抽取项目](https://github.com/crownpku/Information-Extraction-Chinese/tree/master/RE_BGRU_2ATT).

系统要求：

* Python (>=3.5) 一定要用Python 3。因为编码问题，Python 2对中文太不友好。
* TensorFlow (>=r1.0) Tensorflow几个月一个大变样，对<1.0暂不支持。不过清华Thunlp的原始代码是旧版的Tensorflow，对旧版TF仍然念念不忘的朋友们可以去参考。
* scikit-learn (>=0.18)

训练:

1.所有数据准备在 origin\_data/ 中, 包括了关系种类 (relation2id.txt), 训练数据 (train.txt), 测试数据 (test.txt) and 中文字向量 (vec.txt). 中文字向量是在中文wikipedia上训练的。训练数据和测试数据是将原始数据随机排序，然后按大概80%做训练，20%做测试的方法分开。更好的方法是再加入验证数据，但是训练一个epoch已经好长时间，就偷个懒不搞k-fold什么的了。。。

现有的数据包含以下12种关系种类：

unknown, 父母, 夫妻, 师生, 兄弟姐妹, 合作, 情侣, 祖孙, 好友, 亲戚, 同门, 上下级

2.将所有数据通过字向量整理成numpy的形式，存储在 data/

python initial.py

3.进行训练，模型会存储在 model/。并没有怎么仔细调节超参数，模型结构除了字向量的输入以外，也是完全照搬英文版的模型。感兴趣的朋友们以这个结果为baseline可以尝试调出更好的模型参数。

python train\_GRU.py

预测：

python test\_GRU.py

代码中有两个main函数。第一个main函数是用测试数据来测试准确率的（我的实验大概70%左右），第二个则是供用户输入数据进行inference。程序会在command line中提示输入”name1 name2 sentence”格式的测试数据。

一些结果

我们请来王大牛和李晓华两位同学来做角色扮演，测试了一些例子。

可以看到模型给出的结果还是蛮有趣的，有一些错误，但也是“情有可原”的合理。

回顾整个过程，对于中文关系提取这个比较复杂的任务，我们的方法完全没有构造n-gram、词性、依存句法等复杂特征，使用深度学习模型，甚至不需要中文分词，只依靠字向量和大量训练语料就得到了一个效果不错的端到端模型。

INFO:tensorflow:Restoring parameters from ./model/ATT\_GRU\_model-9000

reading word embedding data...

reading relation to id

实体1: 李晓华

实体2: 王大牛

李晓华和她的丈夫王大牛前日一起去英国旅行了。

关系是:

No.1: 夫妻, Probability is 0.996217

No.2: 父母, Probability is 0.00193673

No.3: 兄弟姐妹, Probability is 0.00128172

实体1: 李晓华

实体2: 王大牛

李晓华和她的高中同学王大牛两个人前日一起去英国旅行。

关系是:

No.1: 好友, Probability is 0.526823

No.2: 兄弟姐妹, Probability is 0.177491

No.3: 夫妻, Probability is 0.132977

实体1: 李晓华

实体2: 王大牛

王大牛命令李晓华在周末前完成这份代码。

关系是:

No.1: 上下级, Probability is 0.965674

No.2: 亲戚, Probability is 0.0185355

No.3: 父母, Probability is 0.00953698

实体1: 李晓华

实体2: 王大牛

王大牛非常疼爱他的孙女李晓华小朋友。

关系是:

No.1: 祖孙, Probability is 0.785542

No.2: 好友, Probability is 0.0829895

No.3: 同门, Probability is 0.0728216

实体1: 李晓华

实体2: 王大牛

谈起曾经一起求学的日子，王大牛非常怀念他的师妹李晓华。

关系是:

No.1: 师生, Probability is 0.735982

No.2: 同门, Probability is 0.159495

No.3: 兄弟姐妹, Probability is 0.0440367

实体1: 李晓华

实体2: 王大牛

王大牛对于他的学生李晓华做出的成果非常骄傲！

关系是:

No.1: 师生, Probability is 0.994964

No.2: 父母, Probability is 0.00460191

No.3: 夫妻, Probability is 0.000108601

实体1: 李晓华

实体2: 王大牛

王大牛和李晓华是从小一起长大的好哥们

关系是:

No.1: 兄弟姐妹, Probability is 0.852632

No.2: 亲戚, Probability is 0.0477967

No.3: 好友, Probability is 0.0433101

实体1: 李晓华

实体2: 王大牛

王大牛的表舅叫李晓华的二妈为大姐

关系是:

No.1: 亲戚, Probability is 0.766272

No.2: 父母, Probability is 0.162108

No.3: 兄弟姐妹, Probability is 0.0623203

实体1: 李晓华

实体2: 王大牛

这篇论文是王大牛负责编程，李晓华负责写作的。

关系是:

No.1: 合作, Probability is 0.907599

No.2: unknown, Probability is 0.082604

No.3: 上下级, Probability is 0.00730342

实体1: 李晓华

实体2: 王大牛

王大牛和李晓华为谁是论文的第一作者争得头破血流。

关系是:

No.1: 合作, Probability is 0.819008

No.2: 上下级, Probability is 0.116768

No.3: 师生, Probability is 0.044831