实体识别

目前已经实现了基本的实体识流程，可以从标注文件中生成简单模型，并预测与评分。

实体识别的框架代码已经放在 /data1/ywang/cmner/ner 目录下，所有程序已经在服务器上测试过。

依赖的包

我已经安装了如下的包：

Python 3.5 + Anaconda 4.4

目前已经安装了 anaconda 4.4 + python 3.5, 所有代码需要 Python 3.5 运行

pycrfsuite

这个是CRF的基础模型，用于测试整个流程

Tensorflow 1.1

这个可以使用基于tensorflow的深度学习算法

Jieba 0.38

“结巴”中文分词软件，用于中文分词

执行步骤

数据转换

首先进入anaconda的python 3.5环境

$ source activate py35

(py35) [ywang@localhost ner]$

进入/data1/ywang/cmner/ner 目录

$ cd /data1/ywang/cmner/ner

使用 ehost2bio.py 工具把ehost的标注文件转换成为BIO格式。可以执行 python ehost2bio.py -h 来看说明。

执行 $ python ehost2bio.py data/20170712 train.bio

这个命令读取项目目录中的所有文件，将文章分句再分词，并转换成BIO标签。每一个词给一个标签。其中data/20170712为eHOST标注项目目录。

这个工具也可以把标注文档分成训练集与测试集：

$ python ehost2bio.py data/20170712 -r 0.8 train

-r 参数是将数据按 0.8：0.2 的比例分成训练样本与测试样本，并建立 train目录，生成train/train.bio与train/test.bio

训练模型

用 train-test.py 来生成模型

$ python train-test.py -t -m creator-model train.bio

-t 参数表示训练一个新的模型
-m 参数指定模型的名称为 creator-model
train.bio 为训练集

程序会生成2个文件：creator-model.model为模型文件，creator-model.charmap为中文映射文件

预测实体

用 train-test.py 来预测实体

$ python train-test.py -m creator-model train.bio > output.bio

同上，只是省去了 -t 参数，表示预测模型而不是训练

预测的输出为bio格式，将其导入 output.bio

模型评分

用conlleval.pl来评估预测准确率

$ ./conlleval.pl < output.bio

评估模式为准确率，精准率，召回率和F1评分

目前用5篇标注的评分结果为：

训练准确度：P=81.77, R=83.49, F1=82.62

因为后几篇没有进行查体部分的标注，所以会影响到模型准确率。最后1篇标注明显比前4篇要少：

001.txt 175
002.txt 176
003.txt 247
004.txt 265
005.txt 88

后续工作

1. 训练集中需要除去有意跳过标注的部分（这个最好能够用一个标注种类来区别）
2. 特征工程
3. 使用cnn+lstm+crf模型（需要大量未标注的文档）
4. 获取更多的标注
5. 修饰分类
6. 关系分类

分句

目前使用换行符作为断句标识符，当然这不是一个最好的方法，需要加入更多的规则或者使用中文断句工作来做。

分词

加入医学字典提高分词的准确率。

Word Embedding模型

简介

Word Embedding又叫做嵌入词，与传统的One-Hot表示不同，是用一个维度较低的实数向量来表示一个词。传统的one-hot表示是将词表示为0，1向量中的一个维度，例如：“检验” = (1, 0, 0, …, 0, 0), “检查” = (0, 1, 0, …, 0, 0)。而嵌入词则表示为：“检验” = (0.8, 0.6, 0.1, …, 0.003, 0.15), “检查” = (0.7, 0.65, 0.3, …, 0.001, 0.20)，在每一个维度上都有一定的权重分配。因此，如做同义词计算就可以通过计算cosine距离下最相似的向量来获得。

2013年Tomas Mikolov发表了两篇重要的论文：“Efficient Estimation of Word Representations in Vector Space”、“Distributed Representations of Words and Phrases and their Compositionality”。

这两篇论文中提出了一个word2vec的工具包，里面包含了几种嵌入词的方法，例如 Continuous Bag of Word (CBOW) 和 Skip-gram。这些方法有两个特点。一个特点是速度快，另一个特点是得到的嵌入向量具备对比的性质，例如“国王” - “男人” = “皇后” - “女人”，因此可以获取到语义。

TensorFlow 的教程中介绍了如何训练word2vec模型，详细资料可以看https://www.tensorflow.org/tutorials/word2vec。

对于英文词嵌入模型，已经有很多的不同方法：

- Distributed Representations of Words and Phrases and their Compositionality
- Efficient Estimation of Word Representations in Vector Space
- GloVe Global Vectors forWord Representation
- Neural probabilistic language models
- Natural language processing (almost) from scratch
- Learning word embeddings efficiently with noise contrastive estimation
- A scalable hierarchical distributed language model
- Three new graphical models for statistical language modelling
- Improving word representations via global context and multiple word prototypes

中文的词嵌入模型也有不少研究，因为中文里面每一个字也有一定的语义，因此主要是同时利用了字与词之间的关系，将两者结合在一起，主要方法有：

- Improve Chinese Word Embeddings by Exploiting Internal Structure
- Joint Learning of Character and Word Embeddings
- Integrating character representations into Chinese word embedding
- Multi-prototype Chinese Character Embedding

训练模型

着重阅读 Joint Learning of Character and Word Embeddings
用CWE训练词嵌入模型：

1. 从 https://github.com/Leonard-Xu/CWE 下载模型源代码
2. 在服务器上编译
3. 读入文档，进行断句与分词，转化成工具所需的格式
4. 训练词嵌入模型，并得到模型文件

特征工程

据我看过的大部分学术论文里面都说，NER的效果和所采用的词典关系很大，NER要做好，除了模型以外，字典的优化也很重要，不仅是研究问题，也是一个工程问题。

建立字典

寻找各种电子医学字典，抽取里面的词汇并到字典中的分类。例如字典中的“发烧”可能是“症状”，“阿司匹林”可能是药物，这种字典多多益善。

实现一个字典类，可以用来查找词组，并返回其在字典中的类别。

需要可以支持多个字典。如“发烧”在字典1里面的类别是“症状”，字典2里面的类别可能是”发现“， 字典3里面的类别可能是”异常“。查询”发烧“则返回[“症状”，“发现”， “异常”]

实现一个MedDict类，里面有如下函数：

load(filename) - 读如一个字典文件

lookup(word) - 查询一个词。 例如：dict.lookup(“发烧“) 返回 [“症状”，“发现”， “异常”]。如果word不在字典中则返回空的list []。

我可以调用如下代码：

```
med_dict = MedDict()
med_dict.load(“ICD10.txt”) #读入ICD10字典
med_dict.load(“ShenNei.txt”) #读如神内科的字典
dict_feature = med_dict.lookup(“发烧”) # 返回 [“症状”，“发现”， “异常”]
```

词形态获取

医学的文档里经常含有各种不同形态的词，如80/120，3.5ml等。通过正则表达式来获取一些词形的信息有助于分类。从中文英文混合的词组中提取词形特征，例如 “心率 127 次 / min ,” 的形态特征可以表示为 “C num C slash lower punc” 意思为： “中文 数字 中文 斜线 小写英文 标点”。相关的形态特征抽取可以参考 Enhancing HMM-based biomedical named entity recognition by studying special phenomena: http://www.sciencedirect.com/science/article/pii/S1532046404000838 的第3小结。

实现一个形态特征抽取器，实现函数

orth(word) - 返回当前词的形态特征，如上面的例子：

orth(“心率”) = ”C“
orth(“127”) = “num”
orth(“/”) = “slash”

更多的特征可以参考上面提到的论文

关系分类

所谓关系分类就是将句子中的实体对分类为关系类或者无。例如：

主 诉： 发现 心脏 杂音 10 月余 。

其中命名实体为：

心脏 = 身体部位
杂音 = 医学发现
10 月余 = 时间词

所有可以组成的实体对（忽略顺序）为：

（心脏，杂音）
（心脏，10月余）
（杂音，10月余）

其中标注的关系有：

（心脏，杂音）= 身体部位-病症
（杂音，10月余）= 时间-病症

未标注的：

（心脏，10月余）= 无

那么关系抽取的主要任务就是一个分类问题

目前我已经写了代码将标注中的关系抽取出来了，并转化成为了可以使用的格式

可以执行以下代码：

$ python ehost2rel.py data/20170712 train.txt

实体关系存在 train.txt 文件中

格式为：

文件名，关系ID，实体1，实体2，关系类型，实体1每个词在句中的位置，实体2每个词在句中的位置，句子

目前比较热门的关系分类使用了深度神经网络，其主要是解决了传统机器学习特征对句子形态分析的依赖。形态分析，如词形，词意等处理带来的错误大大的影响了分类效果。以下的几篇论文介绍了使用神经网络进行关系抽取的工作：

- Semantic compositionality through recursive matrix-vector spaces
- Relation Classification via Convolutional Deep Neural Network
- Distant supervision for relation extraction via piecewise convolutional neural networks
- Neural Relation Extraction with Selective Attention over Instances
- Relation Extraction: Perspective from Convolutional Neural Networks

清华大学提供的代码：

https://github.com/thunlp/NRE