Web-Lab2

小组成员

PB18111791 雷雨轩

PB18111793 裴启智

实验目的

本实验要求以给定的英文文本数据集为基础，实现一个信息抽取系统。

实验内容

关系抽取

朴素贝叶斯

Info

- 朴素贝叶斯是一种构建分类器的简单方法。该分类器模型会给问题实例分配用特征值表示的类标签，类标签取自有限集合。所有朴素贝叶斯分类器都假定样本每个特征与其他特征都不相关。
- 对于某些类型的概率模型，在监督式学习的样本集中能获取得非常好的分类效果。在许多实际应用中，朴素贝叶斯模型参数估计使用最大似然估计方法；换而言之，在不用到贝叶斯概率或者任何贝叶斯模型的情况下，朴素贝叶斯模型也能奏效。
- 朴素贝叶斯分类器的一个优势在于只需要根据少量的训练数据估计出必要的参数（变量的均值和方差）。由于变量独立假设，只需要估计各个变量的方法，而不需要确定整个协方差矩阵。

Design

- 使用Python的sklearn库所提供的多项式分布贝叶斯

  适用于服从多项分布的特征数据。

  ```
  class sklearn.naive_bayes.MultinomialNB(alpha=1.0, fit_prior=True,
  class_prior=None)
  ```

  `alpha`:先验平滑因子，默认等于1，当等于1时表示拉普拉斯平滑。

  `fit_prior`:是否去学习类的先验概率，默认是True

  `class_prior`:各个类别的先验概率，如果没有指定，则模型会根据数据自动学习， 每个类别的先验概率相同，等于类标记总个数N分之一。

Implementation

- 首先将训练集和测试集转化为词向量，并计算其`tf-idf`矩阵

  ```
  tf_idf_vec = TfidfVectorizer()
  train_vec = tf_idf_vec.fit_transform(train_sentences)
  test_vec = tf_idf_vec.transform(test_sentences)
  ```

- 训练朴素贝叶斯分类器，实际运行发现将参数alpha设置为0.7左右分类效果较好（设为1会将许多测试句子都分类为Other）

```
classifier = MultinomialNB(alpha = 0.7)
classifier.fit(train_vec, train_relations)
```

- 使用分类器对测试集进行分类（预测）

```
test_result = classifier.predict(test_vec)
```

Result

Filename	ACC-Relation	ACC-NER
裴启智-PB18111793-7.txt	0.376875	0.0

可以看到不加额外处理的朴素贝叶斯方法分类效果一般

Simple Transformers(BERT)

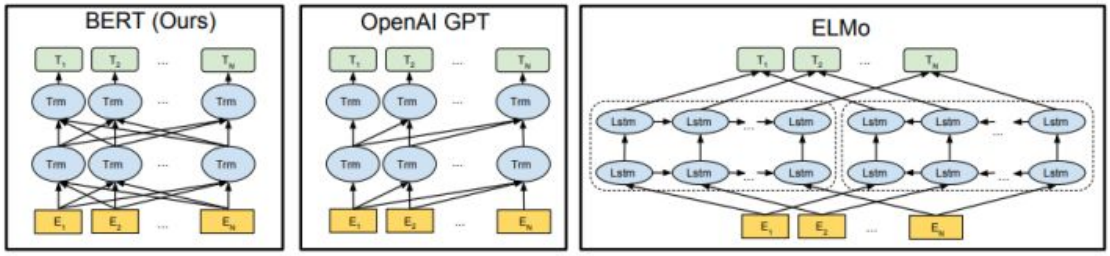
Info

Simple Transformers

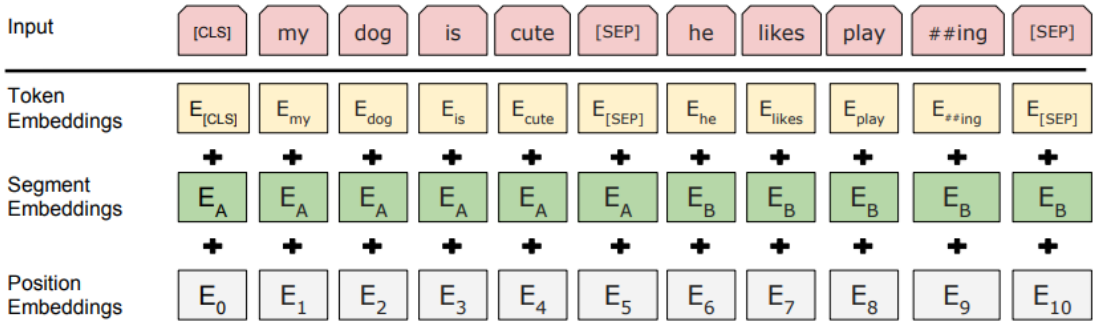
- Simple Transformers库是AI创业公司Hugging Face在Transformers库的基础上构建的。
- Simple Transformers专为需要简单快速完成某项工作而设计。不必拘泥于源代码，也不用费时费力地去弄清楚各种设置

BERT

- BERT的全称是Bidirectional Encoder Representation from Transformers，即双向Transformer的encoder，因为decoder是不能获得要预测的信息的。
- 模型的主要创新点都在pre-train方法上，即用了Masked LM和Next Sentence Prediction两种方法分别捕捉词语和句子级别的representation。
- 模型结构



- Embedding



其中：

- Token Embeddings是词向量，第一个单词是CLS标志，可以用于之后的分类任务
- Segment Embeddings用来区别两种句子，因为预训练不光做LM还要做以两个句子为输入的分类任务
- Position Embeddings是学习出来的

- 优点：
  - BERT是截至2018年10月的最新state of the art模型，通过预训练和精调横扫了11项NLP任务
  - 使用Transformer，相对rnn更加高效、能捕捉更长距离的依赖。对比起之前的预训练模型，它捕捉到的是真正意义上的bidirectional context信息。
- 缺点：
  - [MASK]标记在实际预测中不会出现，训练时用过多[MASK]影响模型表现
  - 每个batch只有15%的token被预测，所以BERT收敛得比left-to-right模型要慢（它们会预测每个token）

Design

- 基于simpletransformers.classification的ClassificationModel，将关系抽取看作一个多分类任务

Implementation

- 读入训练集和测试集
- 训练多分类器

```
model_args = ClassificationArgs(num_train_epochs=1)
    model = ClassificationModel(
        'bert',
        'bert-base-cased',
        num_labels=10,
        args=model_args,
        use_cuda=False
    )
model.train_model(train_df, output_dir='./model')
```

后续使用时只需要加载即可

```
model = ClassificationModel(
        "bert",
        "model/checkpoint-800-epoch-1",
        use_cuda=False
        )
```

- 基于model对测试集进行分类（预测）

```
test_result, raw_result = model.predict(test_sentences)
```

Result

Filename	ACC-Relation	ACC-NER
裴启智-PB18111793-8.txt	0.640625	0.0

可以看到基于BERT模型的分类器效果比朴素贝叶斯分类器好上不少

实体识别

Info

- 命名实体识别（英语：Named Entity Recognition），简称NER，是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等，以及时间、数量、货币、比例数值等文字。
- 解决问题标准流程：

  Training:

  1. 收集代表性的训练文档
  2. 为每个 token 标记命名实体(不属于任何实体就标 Others O)
  3. 设计适合该文本和类别的特征提取方法
  4. 训练一个 sequence classifier 来预测数据的 label

  Testing:

  1. 收集测试文档
  2. 运行 sequence classifier 给每个 token 做标记
  3. 输出命名实体
- 序列标注方法
  - 序列标注的方法中有多种标注方式：BIO、BIOSE、IOB、BILOU、BMEWO，其中前三种最为常见。
  - 在综合考虑各种标注方法后，本次实验选择实现相对简单但有效的**BIO**标注
    - **B** stands for '**beginning**' (signifies beginning of an Named Entity, i.e. NE)
    - **I** stands for '**inside**' (signifies that the word is inside an NE)
    - **O** stands for '**outside**' (signifies that the word is just a regular word outside of an NE)
- 命名实体识别常用方法
  - 基于词典：预先构建一个命名实体词典。
  - 基于规则：手工构造规则模板，对符合规则的实体进行识别
  - 基于统计
    - 基于分类的命名实体识别方法
    - 基于序列标注的命名实体识别方法

  考虑到实现效果以及现有工具，主要调研了基于统计的一些命名实体识别方法及可能用到的模型。
- **隐马尔可夫模型（Hidden Markov Model，HMM）**
  - NER本质上可以看成是一种序列标注问题（预测每个字的BIOES标记），在使用HMM解决NER这种序列标注问题的时候，我们所能观测到的是字组成的序列（观测序列），观测不到的是每个字对应的标注（状态序列）
  - HMM的三个要素
    - **初始状态分布**：每一个标注作为句子第一个字的标注的概率

        - **状态转移概率矩阵**就是由某一个标注转移到下一个标注的概率（设状态转移矩阵为 M ，那么若前一个词的标注为 tag_i ，则下一个词的标注为 tag_j 的概率为 M_{ij} ）
        - **观测概率矩阵**就是指在某个标注下，生成某个词的概率
    - HMM模型的训练过程对应隐马尔可夫模型的学习问题，实际上就是根据训练数据根据**最大似然**的方法估计模型的三个要素，即上文提到的初始状态分布、状态转移概率矩阵以及观测概率矩阵
    - 模型训练完毕之后，要利用训练好的模型进行解码，就是对给定的模型未见过的句子，求句子中的每个字对应的标注，针对这个解码问题，我们使用的是维特比（viterbi）算法
    - 缺陷：HMM模型中存在两个假设

        1. 输出观察值之间严格独立
        2. 状态转移过程中当前状态只与前一状态有关。

        也就是说，在命名实体识别的场景下，HMM认为观测到的句子中的每个字都是相互独立的，而且当前时刻的标注只与前一时刻的标注相关。但实际上，命名实体识别往往需要更多的特征，比如词性，词的上下文等等，同时当前时刻的标注应该与前一时刻以及后一时刻的标注都相关联。
- **条件随机场模型（CRF)**
    - 定义：

      设 X 与 Y 是随机变量，$P(Y|X)$是在**给定X的条件下Y**的条件概率分布。

      若随机变量Y构成一个由**无向图**$G=(V,E)$表示的**马尔可夫随机场**，即

      $$P(Y_v|X,Y_w,w\neq v)=P(Y_v|X,Y_w,w\sim v)$$

      对任意顶点v成立，则称条件概率分布$P(Y|X)$为条件随机场。

      ➢ **条件随机场的一般定义**

      其中：
      $w\sim v$表示在图$G=(V,E)$中**与顶点v有边连接**的所有顶点w；
      $w\neq v$表示顶点v以外的所有顶点；
      Y_v与Y_w为顶点v 与w对应的随机变量。

    - 条件随机场模型是由Lafferty在2001年提出的一种典型的判别式模型。它在观测序列的基础上对目标序列进行建模，重点解决序列化标注的问题。
    - 条件随机场模型既具有判别式模型的优点，又具有产生式模型的优点。其考虑了上下文标记间的转移概率，以序列化形式进行全局参数优化和解码的特点，解决了其他判别式模型(如最大熵马尔科夫模型)难以避免的标记偏置问题。
    - 条件随机场模型（conditional random fields ）是一种无向图模型，它是在给定需要标记的观察序列的条件下，计算整个标记序列的联合概率分布，而不是在给定当前状态条件下，定义下一个状态的状态分布。即给定观察序列O,求最佳序列S。
    - 条件随机场就通过引入自定义的特征函数，不仅可以表达观测之间的依赖，还可表示当前观测与前后多个状态之间的复杂依赖，可以有效克服HMM模型面临的问题
- **Bi-LSTM （+CRF)**
    - 除了以上两种基于概率图模型的方法，LSTM也常常被用来解决序列标注问题。和HMM、CRF不同的是，LSTM是依靠神经网络超强的非线性拟合能力，在训练时将样本通过高维空间中的复杂非线性变换，学习到从样本到标注的函数，之后使用这个函数为指定的样本预测每个token的标注。

- 简单的LSTM的优点是能够通过双向的设置学习到观测序列（输入的字）之间的依赖，在训练过程中，LSTM能够根据目标（比如识别实体）自动提取观测序列的特征，但是缺点是无法学习到状态序列（输出的标注）之间的关系，要知道，在命名实体识别任务中，标注之间是有一定的关系的，比如B类标注（表示某实体的开头）后面不会再接一个B类标注，所以LSTM在解决NER这类序列标注任务时，虽然可以省去很繁杂的特征工程，但是也存在无法学习到标注上下文的缺点。

  相反，CRF的优点就是能对隐含状态建模，学习状态序列的特点，但它的缺点是需要手动提取序列特征。所以一般的做法是，在LSTM后面再加一层CRF，以获得两者的优点。

Design

- 调研发现 `nltk` 或者Stanford NLP已经提供了训练好的模型，能实现典型的实体识别任务，主要为Location, Person, Organization, Misc任务集。但是这与本次实验的要求不太符合，因为本次实验里实体是给定的存在一定关系的实体对，所以需要另作处理。
- 本次实验考虑采用CRF来完成命名实体识别任务。
- 当然，还可以进一步尝试 BiLSTM+CRF，BERT+CRF等，在本次实验结果的基础上应该会有进一步的效果提升。
- 考虑到Simple Transformers库已经在关系抽取任务里使用了，而且加入Bert的方法效果明显比非神经网络模型的效果好一大截。但为了尝试不同的方法，并节省实验时间，所以考虑尝试相对传统、基于CRF模型的方法来完成命名实体识别任务。此任务要换为用SimpleTransformers包也非常简单，因为数据集标签已经打好，所以只需要调用SimpleTransformers包的接口即可。

Implementation

数据预处理

- 为了得到BIO序列标注格式的数据，需对原训练集和测试集数据作处理
- 首先考虑去除停用词及标点符号。
- 考虑到所给数据里实体的稀疏性（每个句子一般仅两个实体），所以若仍按照所属关系来对实体分类的话，效果会很差，所以仅考虑统一看做一种实体。
- 预处理时除了标注实体标签外，也对每个词的词性做了标注
- 测试集也做了词性标注的处理，为了统一格式便于后续数据读入，把测试集数据里每个单词标签都打为"O"
- 预处理效果

```
The system described greatest application arrayed configuration antenna
elements
DT NN VBD JJS NN VBN NN NN NNS
O O O O O O B-1 O B-1
```

数据特征选择

- 因为是序列标注方法，所以考虑最为常见而有效的方式来为每个词标注特征，包括单词是否大写、是否小写、首字母是否大写、是否为数字，词性特征，以及该单词前、后各一个词的相应特征。

数据训练

- ```
  crf = CRF(algorithm="lbfgs",
                  c1=0.1,
                  c2=0.1,
                  max_iterations=200,
                  all_possible_transitions=True)
  ```
  ```

以上用的是sklearn实现的逻辑回归CRF模型。使用L-BFGS作为梯度下降方法。在调参后发现，正则化系数c1,c2设置为0.1， 最大迭代次数设置为200效果较好。

使用`crf.fit(train_X,train_y)`来一键训练

- 考虑使用K折交叉验证来对训练模型效果做初步评估

```
pred=cross_val_predict(estimator=crf,X=train_X,y=train_y,cv=4)
report = flat_classification_report(y_pred=pred,y_true=train_y)
```

Result

- 训练结果（K-折交叉验证）
  - 正确率`B-1`: 0.62，`I-1`: 0.26，`O`:0.86。
  - 召回率依次为 0.46，0.06，0.93

	precision	recall	f1-score	support
	1.00	1.00	1.00	7
B-1	0.62	0.46	0.53	13215
I-1	0.26	0.06	0.10	586
O	0.86	0.93	0.90	51106
accuracy			0.83	64914
macro avg	0.69	0.61	0.63	64914
weighted avg	0.81	0.83	0.81	64914

Pipeline方法

（基于前面实体识别的结果进行关系抽取）

- Pipeline方法指先抽取实体，再抽取关系。
- 这两个抽取模型的灵活性高，不需要同时标注实体和关系的数据集

关系抽取模型参考

- 论文：https://www.aclweb.org/anthology/C14-1220/
- github仓库：https://github.com/FrankWork/conv_relation

关系抽取模型简介

摘要

- 用于关系分类的最先进的方法主要是基于统计机器学习的，性能很大程度上取决于提取特征的质量。提取的特征通常源自于预先存在的NLP系统的输出，**这导致现有NLP工具的错误在特征提取任务中被不断传播且放大。**
- 该模型利用卷积神经网络提取词汇和句子级别的特征**
- 将所有单词标记(word tokens)作为输入，无需复杂预处理。
- 首先，通过查找**词嵌入(word embeddings)**将单词标记转换成向量。然后，根据给定的名词抽取出词汇级别的特征。同时使用卷积方法学习句子级别的特征。将这两个级别的特征串联形成最终提取的特征向量。最后，将这些特征输入`softmax`分类器来预测两个标记名词间的关系。实验结果表明该方法明显优于最先进的方法。

引言

- 为了识别名词对之间的关系，巧妙地从不同句法和语义结构中结合词汇和句子级别的线索是十分必要的。

  例如，在这个句子中：

  > “The [fire]e1 inside WTC was caused by exploding [fuel]e2”

  我们通常利用标记的名词和句子的意思来标识*fuel*和*fire*之间的*cause-effect*关系。
- 本文中，使用CNN来抽取关系分类中词汇和句子级别的特征。方法将所有单词标记作为输入而无须复杂处理。（例如词性标记和语法分析）。
- 首先，通过查找**词嵌入(word embeddings)**将单词标记转换成向量。然后，根据给定的名词抽取出词汇级别的特征。同时使用卷积方法学习句子级别的特征。将这两个级别的特征串联形成最终提取的特征向量。最后，将这些特征输入`softmax`分类器来预测两个标记名词间的关系。实验结果表明该方法明显优于最先进的方法。
- 可以将其视作一个多分类问题，产生不同目标函数。此外，关系分类被定义为将关系标签分配给单词对。因此有必要区分我们期望分配关系标签的单词对。**为了这个目的，利用position features（PF）来编码目标名词对的相对距离。这是使用CNN进行关系分类的第一个例子**

神经网络体系结构

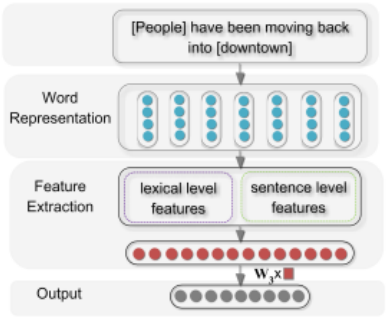


Figure 1: Architecture of the neural network used for relation classification.

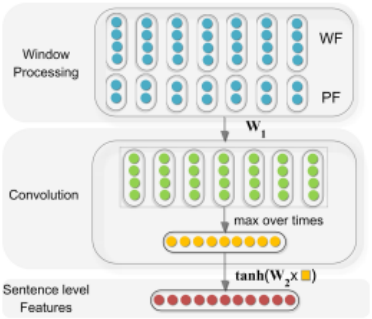


Figure 2: The framework used for extracting sentence level features.

- 图1描述了用于关系分类的神经网络体系结构，网络采用句子作为输入，并且发现特征提取的多个层次，较高的层次代表了输入的更抽象的方面。

  它主要包括3个组成部分：
  1.单词表征（Word Representation）
  2.特征提取
  3.输出
- 系统无需复杂的句法或语义预处理，输入是带有两个标记名词的句子。然后通过查找词嵌入将单词标记转换为向量。接下来，分别提取词汇和句子级的特征，然后直接连接形成最终的特征向量。最后，为了计算每个关系的置信度，特征向量被输入`softmax`分类器。分类器的输出是一个向量，其维数等于预定义关系类型的数量。每个维度的值是对应关系的置信度得分。

单词表征

- 在单词表征部分中，通过查找词嵌入将每个输入单词标记变换为向量。Collobert等人汇报，从大量未标记数据中学习的词嵌入要比随机初始化词嵌入更令人满意。在关系分类中，我们首先应当使用大量未标注数据集中精力学习有判别能力的词嵌入，它拥有更多句法和语义信息。不信的是，训练

词嵌入总是花费很长时间。然而，有很多训练好的词嵌入是可以免费使用的。我们的实验直接使用Turian等人提供的词嵌入。

词汇级特征

- 词汇级特征是决定关系的重要线索。传统的词汇级别特征主要包括名词本身，名词对以及实体间词序列的类型，其质量很大程度上取决于现有NLP工具产生的结果。
- **取而代之，本文使用词嵌入作为基本特征的来源。选择标记名词的词嵌入及其上下文标记，**所有这些特征串联到词汇级特征向量 **1。**
- 表1展示了选择的word embeddings，与句中标记名词相关。

Features	Remark
L1	Noun 1
L2	Noun 2
L3	Left and right tokens of noun 1
L4	Left and right tokens of noun 2
L5	WordNet hypernyms of nouns

Table 1: Lexical level features.

句子级特征

- 如上所述，所有标记都被表示成词向量，已被证明与人类对词相似性的判断有很好的相关性。尽管取得了成功，单个词向量模型是严重受限的，因为它们不能捕捉长距离特征和语义合成性，这是自然语言的重要品质，它使人能够理解更长表达的含义。
- 这里提出一个`max-pooled CNN`以提供句子级别的表示并且自动抽取句子级别的特征.
- 图2展示了句子级别的特征抽取框架。在`Window Processing`部分，每个标记进一步被表示为`词特征(WF)`和`位置特征(PF)`。然后，向量通过`卷积`部分。最后，我们通过`非线性变换`得到了句子级别的特征。

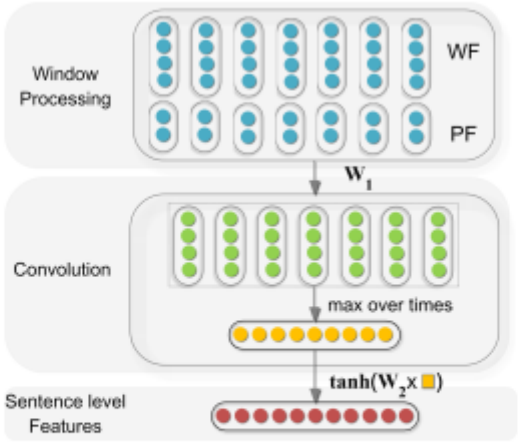


Figure 2: The framework used for extracting sentence level features.

Word Features 词特征

分布假设理论（Harris，1954）指出，在相同语境中出现的词语往往具有相似的含义。为了捕获这个特征，`WF`结合了词的向量表示以及其在上下文中的向量表示。假设我们有下列单词序列。

$$S:[People]_0\ have_1\ been_2\ moving_3\ back_4\ into_5\ [downtown]_6$$

被标记的名词与标签yy关联，yy定义了被标记名词对包含的关系类型。每个单词也与word embeddings的索引相关联。句子SS中的单词标记被表示为向量列表$x_0, x_1, x_2, x_3, x_4, x_5, x_6$,其中$x_i$对应于句子中第i个单词的word embedding。

要使用w的上下文大小，我们将w大小的向量窗口组合成一个更丰富的特征。例如，当w=3,句子S中第三个单词`"moving"`的`WF`被表达为$[x2, x3, x4]$,相似地，考虑到整个句子，`WF`可以作如下表示：

$$\{[x_s, x_0, x_1], [x_0, x_1, x_2], \cdots, [x_5, x_6, x_e]\}$$

> x_s和x_e是分别对应于句子开头和结尾的特殊word embedding。

Position Features 位置特征

- 关系分类是复杂的任务。传统上，结构特征（例如，名词间的最短依赖路径）用于解决该问题。显然，只通过`WF`无法捕获这类结构信息。有必要指明哪个输入标记(tokens)是句子的目标名词。为此，建议将`PF`用于关系分类。本位中，`PF`是当前单词与w1w1和w2w2的相对距离。例如，句子S中`"moving"`与`"people"`和`"downtown"`的相对距离分别为为3和-3。
- 在论文的方法中，相对距离也被映射到维向量dede(超参数），这个向量是随机初始化的。然后，获得当前单词与w1和w2的相对距离相关的距离向量d1和d2，以及PF=[d1,d2]
- 结合`WF`和`PF`，单词被表示为[WF,PF]T，随后被输入算法的卷积部分。

卷积

- Word Representaion方法可以通过窗口中的向量组合来捕获上下文信息。但是，它只会在句子中的每个单词周围产生局部特征。在关系分类中，用目标名词标记的用于输入的句子，仅对应于关系类型而不是预测每个单词的标签。因此，可能有必要使用所有局部特征并预测全局关系。

句子级别特征向量

为了学习更复杂的特征，设计了一个非线性层，且选择双曲线`tanh`作为激活函数。`tanh`中的一个有用的属性是**它的导数可以用函数值本身来表示**:

$$\frac{d}{dx}tanhx = 1 - tanh^2x$$

它的优点是在反向传播训练过程中使得梯度的计算变得容易。形式上，非线性变换可以被写作:

$$g = tanh(W_2m)$$

W2∈Rn2×n1是线性变换矩阵，其中n2(超参数)是隐藏层2的大小。相比于m∈Rn1×1, g∈Rn2×1可以被认为是更高级别的特征（句子级别）。

输出

自动学到的词汇和句子级特征被串联成一个单独的向量f=[l,g]。为了计算每个关系的置信度,特征f∈Rn3×1(n3等于n2加上词汇级别特征的维数)，被输入到`softmax`分类器。

$o = W_3 f$

反向传播训练

- 这里提出的基于DNN的关系分类方法可以被表示为5元组θ=)θ=(X,N,W1,W2,W3)(N代表WordNet上位词的词嵌入）。
- 在本文中，认为每个输入的句子是独立的。给出输入样例s,带有参数θ的网络输出向量o,其中第i个部分oi包含了关系i的得分。为了获得条件概率p(i|x,θ),将对所有关系类型使用`softmax`操作:

$$p(i \mid x, \theta) = \frac{e^{o_i}}{\sum_{k=1}^{n_4} e^{o^k}}$$

- 给出所有（假设T）训练示例(x(i);y(i)),可以写出参数的对数似然：

$$J(\theta) = \sum_{i=1}^{T} \log p(y^{(i)} \mid x^{(i)}, \theta)$$

- 为了计算参数θ,我们使用简单优化方法SGD来最大化对数似然J(θ).N,W1,W2,W3是随机初始化的，X使用Word Embeddings初始化。由于这些参数在神经网络的不同层，我们实现后向传播算法：通过网络使用差异化链规则，迭代的选择样例(x,y)并且应用以下更新规则，直到word embedding层reached

$$\theta \leftarrow \theta + \lambda \frac{\partial \log p(y|x,\theta)}{\partial \theta}$$

结果

Filename	ACC-Relation	ACC-NER
裴启智-PB18111793-6.txt	0.410625	0.3984375

对比基于该模型的单纯的关系抽取（实体对参考原数据集标注出来）的结果

Filename	ACC-Relation	ACC-NER
裴启智-PB18111793-2.txt	0.643125	0.0

可以看到Pipeline方法得到的综合结果并不理想，分析可知Pipeline方法存在如下缺点

1. 误差积累：实体抽取的错误会影响下一步关系抽取的性能。
2. 实体冗余：由于先对抽取的实体进行两两配对，然后再进行关系分类，没有关系的候选实体对所带来的冗余信息，会提升错误率、增加计算复杂度。
3. 交互缺失：忽略了这两个任务之间的内在联系和依赖关系。

参考文献

- https://zhuanlan.zhihu.com/p/147537898
- https://zhuanlan.zhihu.com/p/61227299?tdsourcetag=s_pctim_aiomsg
- https://blog.csdn.net/asialee_bird/article/details/85936784
- https://lab.datafountain.cn/forum?id=146&tab=first
- https://blog.csdn.net/weixin_42001089/article/details/97657149
- https://github.com/ThilinaRajapakse/simpletransformers
- https://zhuanlan.zhihu.com/p/46652512
- https://zhuanlan.zhihu.com/p/77868938
- https://fishwinwin.top/2019/10/11/Zeng-2014-note/