NLP Project Report

申炳宇
5130309283
214097274@qq.com

文件说明

data/ 存储使用的特征

lib/ 用到的开源工具包

model/ 一些训练得到的 model 和对应的正确率

test/ 存储 test 时生成的特征

mytest.py 测试程序

mytrain.py 训练程序

scorer.py 评测程序

mi_process.py 用互信息进行排序

cleandata.py 对数据进行一些处理的函数

config.py 一些使用的路径常量

preprocess.py 预处理得到训练特征

使用说明

运行环境：

ntlk 3.0.0

Java version >= 1.8.0

Sklearn

训练：

python mytrain.py

默认使用 500 dependency rule, 500 production rule, 1000 word pair, 200 first last rule

测试：

python mytest.py [option] file

[opiton] : all 生成 parse tree 和 dependence tree

drule 只生成 dependence tree

prule 只生成 parse tree

none 都已生成，直接测试

最后使用的命令：

python mytest.py all test_pdtb.nosense.json

简介

Implicit discourse parsing 是指在两个连续的句子中间，在没有连接词的情况下预测两者的关系。Implicit 就是指不存在连接词的情况。像（becausr, but）等明显辨识连接关系的词对于连接关系的预测有直接的帮助作用。但是在没有连接词的情况下，一些其他的特征对于我们的预测也有很大的帮助。

我的实现是基于参考中[1]的内容进行的特征进行分类。之后根据[4]中 5.4 节的描述，增加了一个特征，这部分的内容将在特征提取这一节进行详细描述。提取特征之后，需要使用一些方法对于特征进行提取。参考[1]中的特征提取的方法，使用了互信息的方法来进行特征的选取。之后使使用最大熵的模型进行训练和分类。在增加了 first_last 词对之后，在 dev 集上的结果达到了 40.09。

下面我对具体的特征提取，特征降维和训练方法结果分别进行描述。

特征提取

在特征提取时,首先我采用了[1]中的三个特征进行提取,包括 word pair, production rules 和 dependency rules.

WORD PAIR

Word pair 就是简单的特征词信息特征，将 arg1 和 arg2 中的词进行匹配，组成一个 word pair。这个时候要注意使用词干而不是单词，因为数据中已经给出了['lemma']也就是词干部分，我们不需要继续提取。还有一个方面是一些无用的标点的去除,在统计 word pair 的时候我认为那些 word pair 如果还有标点就将这个 word_pair 去除。

PRODUCTION RULE

Production Rule 指的是根据一些方法的到的 parse tree，我们可以得到一些生成规则。在这一部分，使用 berkerly parser 可以很方便的得到 parse tree 的结果。

((S (NP (NNP BELL) (NNP INDUSTRIES) (NNP Inc.)) (VP (VB increase) (NP (PRP$ its) (NN quarterly)) (PP (TO to) (NP (CD 10) (NN cent))) (PP (IN from) (NP (NP (CD seven) (NN cent)) (NP (DT a) (NN share)))))))

如上，这是一个对应的 parse tree 的结果，从 parse tree 我们可以使用 ntlk.Tree 可以方便的得到每个 parse tree 的对应的 production rules.

DEPENDENCY RULE

Dependency rule 的获取使用了 stanford parser， 可以得到的结果如下

```
root(ROOT-0, BELL-1)
compound(Inc.-3, INDUSTRIES-2)
nsubj(BELL-1, Inc.-3)
dep(BELL-1, increase-4)
nmod:poss(quarterly-6, its-5)
dobj(increase-4, quarterly-6)
case(cent-9, to-7)
nummod(cent-9, 10-8)
nmod(increase-4, cent-9)
case(cent-12, from-10)
nummod(cent-12, seven-11)
nmod(increase-4, cent-12)
det(share-14, a-13)
dep(cent-12, share-14)
```

为了将这个格式转化为[1]中的描述的 cent<-case dep nummod 的格式，可以进行简单的转化得到。

在提取玩这三个特征之后，我参考[4]中的要求对于 First Last 词对进行了提取。文中的描述说虽然在 implicit discourse 中没有连接词，但是他们连在一起可能也是一种特征，对于结果有一定的帮助。而一些连接词起来，可能对于结果有一定的帮助，如下面的特征可能暗含了一些隐式关系。

```
arg23_we_do_not
arg121_but_they
arg23_there_be_a
```

降维

降维可以使用 MI（mutual information）来进行，根据互信息的公式，可以求出每个特征和具体分类的互信息，将互信息最大的分类当做其分类，并且按照互信息的大小进行排序选取信息量最大的特征。

使用 sklearn.metrics 中的互信息函数可以方便的得到互信息的结果。因为之前没有发现，自己实现了一下互信息的函数。互信息的公式如下：

对应于我们所要求的 p(x,y)就是 feature 在 sense 中出现的次数占 feature 和 sense 总次数的比例，p(x)为 feature 在 sense 中出现的次数除以 feature 出现的总次数， p(y)为 feature 在 sense 中出现的次数 除以 sense 出现的总次数。对 feature 和 sense 出现和未出现的次数进行加和，得到这个具体的 mutual information 的值。

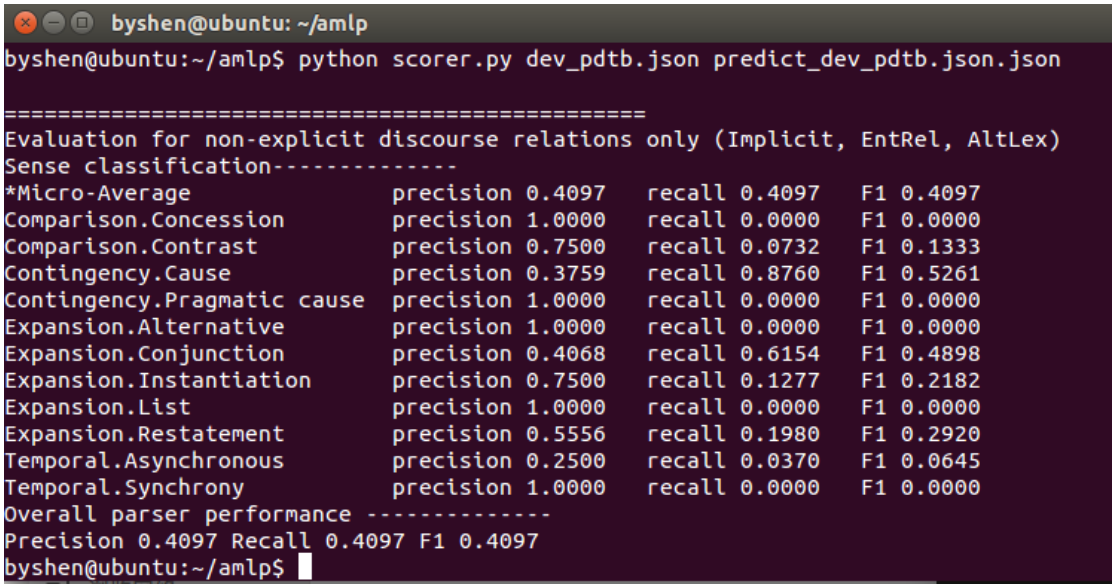
$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right),$$

训练

训练中使用了 ntlk 内置的 maxent 的分类器，只需要将数据转化为对应的格式就可以得到对应的结果。这个时候我对于特征分别进行了测试，发现在 word pairs 为 1000， dependencies rules 为 500， production rules 为 500， first_last rules 为 200 时，最后的正确率较高。训练的过程随着特征量的增加而增加，一般在 40-50min 左右。同时如果训练的循环次数太高，最后的结果可能会过拟合。最后使用循环在 20 时达到的训练结果较好。

结果

在测试集上训练的结果正确率为 40.97，截图如下：



一些提高正确率的尝试

1. 模型中的很多参数的改变对于结果还是有很大的影响的，分别对 word pair, production rule, dependency rule 和 first last pair 进行了测试，发现最后的测试结果在 500, 500, 1000, 100 时的训练的模型的测试结果较高。
2. 一些其他的特征。在使用了这四种特征之后，一直在寻找有没有其他的特征可以继续使用。在 [5] 中，使用了更多的特征如 Polarity Tags, Inquirer Tags 等特征。这些特征从语意的分析上来说比较有用，但是特征的维数的增多也可能带来像过拟合之类的麻烦。
3. 对于分类器的尝试，因为论文中使用了 GIS 最大熵分类器，我首先尝试使用了它。之后使用了 SVM 和逻辑斯蒂回归来进行了预测，但是结果都没有最大熵的好。所以最后还是使用了最大熵作为分类器。
4. 因为最后做的时间不够，没有使用 word2vec 来使用神经网络来进行分类，

从同学的结果上看，神经网络可以很轻易的达到 43 左右的正确率，不得不让人感叹深度学习对于机器学习方法的重要开创性作用。但是深度学习如果没有并行加速的话，速度相对来说较慢，在并行化之后调参之类的工作才相对好做。

总结

在本次作业中，主要的思路是[1]进行提取特征然后进行分类。在参考[4]之后选取了新的特征，显示了一些提高。最后的结果虽然勉强达到了 40.97%，但可能因为特征选取过多而导致泛化能力下降。在搜取一些 paper 之后，很多特征都是基于 word2vec 来进行神经网络的学习，并且最后的结果显示这种特征是一种新的语意的特征，对于结果的分类比较有帮助。可惜由于时间有限，未能使用神经网络的方式进行实现。同时，如果有更多好的语言特征进行挖掘，分类的结果应该有更加多的提高。

最后，感谢赵海老师一学期辛勤的指导，是我对 NLP 领域有了一定的了解。感谢助教对大作业的答疑，是我能顺利完成本次的大作业。

青山不改，绿水长流。Bug 常在，debug 长存。

如果有任何问题，还请您直接联系我的邮箱 214097274@qq.com。谢谢。

References

[1] Lin, Z., Kan, M. Y., & Ng, H. T. (2009). Recognizing implicit discourse relation-
s in the Penn Discourse Treebank.

[2] Chen, D., & Manning, C. D. (2014). A Fast and Accurate Dependency Parser using Neural Networks.

[3] Ji, Y., & Eisenstein, J. (2014). One vector is not enough: Entity-augmented distributional
semantics for discourse relations.

[4] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. (2010). A PDTB-Styled End-to-End Discourse Parser

[5] Pitler, E., Louis, A., & Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text