Paraphrase Identification

Zhe Han 1401214342 iampkuhz@gmail.com

2015年1月9日

Outline

- 算法流程
 - 特征选择
 - 特征效果
 - 分类器说明
- 分析总结
 - 实验效果
 - 分析

Process

• 流程

- 提取句子 lexical, semantic features (precision, recall, normal form/lemma form)
 - lexical: 相同单词, 相同单词对, 编辑距离, 最大公共子串,
 - semantic: 相似/不同的 POS tagger 分布, N/V 单词相似度, 命名实体相似度分布, dependency relation 类型分布, dependency relation pair 相似度
 - 所有判断单词是否相似都采用 WordNet::Synset.Similarity
- 将特征转换为 weka 要求的文件格式
- 调用 weka 的系统分类器训练模型
 - 在开发集 (扩展到 400 维) 上测试效果, 选取最好的组合分类器 (voting) 作为最终的分类器
- 使用选定的组合分类器在扩展的训练集 (train + dev) 训练模型
- 在测试集上跑分类结果

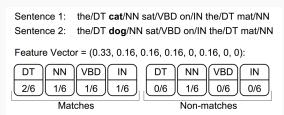
- 实现了下面三篇文章的大多数特征
- 利用第三篇的思想: 多个分类器投票 (+0.4% compared with SVM only)

paper

- (75.6%)Using dependency-based features to take the "para-farce" out of paraphrase
- (75.0%)Using machine translation evaluation techniques to determine sentence-level semantic equivalence
- (76.6%)Paraphrase identification on the basis of supervised machine learning techniques

- common words
 - common words precision: 统计第一句话里重复单词的频率
 - common words recall: 统计第二句话里重复单词的频率
- Position-independent word error rate(precision, recall)
 - 和 common words 正好相反, 统计每句话里不同单词的概率
- proper name(4 dimension, precision, recall)
 - 对句子做命名实体识别 (stanford ner), 对于 4 类, 分别求每句话中每 一类的单词的重复出现的概率

POS distribute vector



- skip-gram common words
 - 句子的任意两个单词组成的单词对 (距离小于等于 4), 考虑另一句话中是否有相同配对 (距离小于等于 4), 计算重复单词对数
- noun/verb similarity
 - 对句子先做词性标注, 然后统计其中名词和动词的重复率

- dependency relation distribution (88 dimension)
 - 类似于 POS distribute vector。Stanford parser 一共有 44 种关系, 前
 44 维表示相同关系的分布频率, 后 44 维表示不同关系的分布频率。
- dependency relation lemma word
 - 使用 lemmatization 之后的句子, 分析 relation pair 的相似概率

feature list

feature	accuracy	feature	accuracy
common words	69.9%	common lemma words	72.6%
edit distance	69.8%	lemma sentence edit distance	70.4%
skip-gram common words	70.3%	skip-gram common lemma words	70.9%
longest common subsequence	70.2%	longest common lemma subsequence	72.3%
noun/verb similarity	69.7%	noun/verb lemma similarity	69.7%
proper name(4 dimension)	73.9%	proper lemma name(4 dimension)	73.5%
dependency relation pair	69.1%	dependency relation lemma pair	68.6%
dependency relation distribution	70.7%		
(88 dimension)			
dependency relation lemma distribution	69.8%		
(88 dimension)			
dependency relation distribution)	70.7%		
dependency relation lemma word	70.0%		
POS distribute vector(72 dimension)	73.2%		
POS distribute lemma vector(72 dimension)	72.05%		
Position-independent word error rate(PER)	70.7%		
Position-independent lemma word error rate(PER)	73.4%		

Classifier

• accuracy on single classifier

classifier	accuracy	
SVM	76.8%	
LibLINEAR	76.6%	
SPegasos	76%	
SimpleLogistic	75.6%	
VotedPerceptron	74.7%	
J48	68.4%	
KNN	67.9%	
NaiveBayes	67.4%	
RBFNetwork	66.9%	

Summary

- 优点
 - Accuracy 高
 - 高于三篇参考文章, 只略低于 statte-of-art(77.4%)
 - trick 很少, 不需要交叉验证找参数
 - 只采用 weka 的默认分类函数, 没有训练特别参数
 - 特征覆盖面广
 - lexical, POS tag, NER, dependency, word semantic(WordNet), ...
 - 提升了求解速度
 - HashMap 存储 POStagged sentence, lemmatized sentence, POStagged lemmatized sentence. 一次计算, 之后直接调用
 - 不加 dependency feature, 程序在十几秒内得到结果; 加入 dependency feature, 在 8 分钟左右

Summary

- 缺点
 - 特征可能冗余 (不漂亮)
 - 特征太多. 将所有正确的特征加入, 没有实验是否冗余
 - 写了一个算法, 对所有多维的特征, 遍历 (取/不取), 时间有限没有跑
 - 句子级别的语义信息抽取很不好
 - 没有真正的提取到句子的信息...
 - Socher(2011NIPS): Dynamic pooling and unfolding recursive autoencoders for paraphrase detection
 - wordvec + R(ecusive)NN + dynamic polling: 4 类特征, 76.8%

方法概述

paper

paper

- Using dependency-based features to take the "para-farce" out of paraphrase
- Using machine translation evaluation techniques to determine sentence-level semantic equivalence
- Paraphrase identification on the basis of supervised machine learning techniques
- vector machines for paraphrase identification and corpus construction