因果向量

篇章关系向量

训练多种关系的向量表示

改进负采样

1. 随机采样的问题
    - 可能会采样到因果词对
    - 采样的句对太简单，没有区分度
2. 做法如下

- 预训练Denoising Auto-encoder（DAE）
- 随机选取因果句对$<C,E>$ ，进行单词的随机删除或者替换，得到$<C_{noise},E_{noise}>$
- 将 $DAE_{<C,E>}$ 记为S_{real} , $DAE_{<C_{noise},E_{noise}>}$ 记为S_{fake} ，generator(G)以S_{fake}为输入，生成负样本$G(S_{fake})$, 生成的负样本更接近真实的负样本,能够提供更多信息
- discriminator(D): ascending

$$log\ D(S_{real}) + log\ (1 - D(G(S_{fake}))) \tag{1}$$

- generator(G): descending

$$log\ (1 - D(G(S_{fake}))) \tag{2}$$

用处

- copa
- Yahoo-QA
- contingency

Why-QA

数据集: Yahoo QA数据集

通过"What causes ..." 以及 "What is the result of ..."抽取了3031个question,每个question有至少四个候选答案,五折交叉验证

难点

可能也存在数据集太小的问题

做法

- 原因结果二分类+句对分类模型
- 加入额外特征，因果向量

因果推理

数据集：COPA

1. dev、test set各500个, 例子为:

```
1. My body cast a shadow over the grass. What was the CAUSE of this?
       The sun was rising.
       The grass was cut.
2. The woman repaired her faucet. What was the CAUSE of this?
       The faucet was leaky.
       The faucet was turned off.

```

形式为句对建模问题, 但缺少训练集,只有测试集和验证集.

1. 句子较短,一般不超过10个词,句式较为简单,基本为主谓宾结构. domain偏向日常生活,如上例.
2. 候选为两个,随机选择的accuracy为0.5
3. 因果关系较多的存在于词级别, 如<sun, shadow> <leaky, repaired>
4. **state-of-art的做法**: 使用10TB的语料抽取关系对,在单词的层面计算词对的PMI值,对句对进行因果打分,选取打分高的候选;因为语料很大,基本能够cover 测试集,所以取得了较好的结果
5. 目前还没有基于神经网络的做法

难点

- 没有训练集,只能通过人工抽取的数据进行训练,所以语料domain不匹配的问题可能存在.
- 由于偏向日常生活,所以使用传记,小说类型的语料可能取得好一点的结果

词表cover情况

1. 测试集

- 验证集词表:原因词1377个, 结果词1307个.
- 因果向量词表包含:原因词1222个,结果词1072个.

2. 验证集

- 验证集词表:原因词1437个, 结果词1334个.
- 因果向量词表包含:原因词1266个,结果词1130个.

CNN分类器行不通的原因

- domain不一样，验证集和测试集的分布跟抽取的数据不一样
- 词对没有共同出现

做法

- 放低对pattern精度的要求,抽取更多的语料,提升覆盖度，训练因果向量
- sharp公布的数据集有80w+个句对,缺少功能词.但是在这个问题中功能词并不是很重要,可以把这部分数据利用起来训练因果向量
- 获取更多语料，提升cover情况下，CNN建模词组表示，计算词组attention
- PMI打分为baseline，用预训练的句对模型，应用通用词向量concat因果向量的特征，如计算句对的max，min，average，对两个候选做rank

隐式篇章

数据集：PDTB

relation	train	dev	test
Comparison	1942	197	152
Contingency	3342	295	279
Expansion	7004	671	574
Temporal	760	64	85

难点

1. 语料为Wall Street Journal,与从英文wiki的抽取的关系的domain可能不一样
2. 数据集太小,容易过拟合,训练过程很不稳定
3. 各论文并没有明确的motivation，主体思路为尝试不同的attention,例如: **为了更好的建模句子表示及句对信息交互,本文提出了一种新的attention机制,并获得了state-of-art结果**
4. 最简单的CNN句对模型（不加attention）便能取得较好结果，然后加component便会过拟合
5. 关系的线索词,重合度很大,同一个句对可能属于多种关系.

关系抽取

1. 书写规则,对四种关系进行了抽取,数量如下

关系类型	抽取数量	pattern数量
contingency	330713	很多
expansion	31590	9个
temporal	34072	8个
comparison	82036	11个

2. 可能的问题

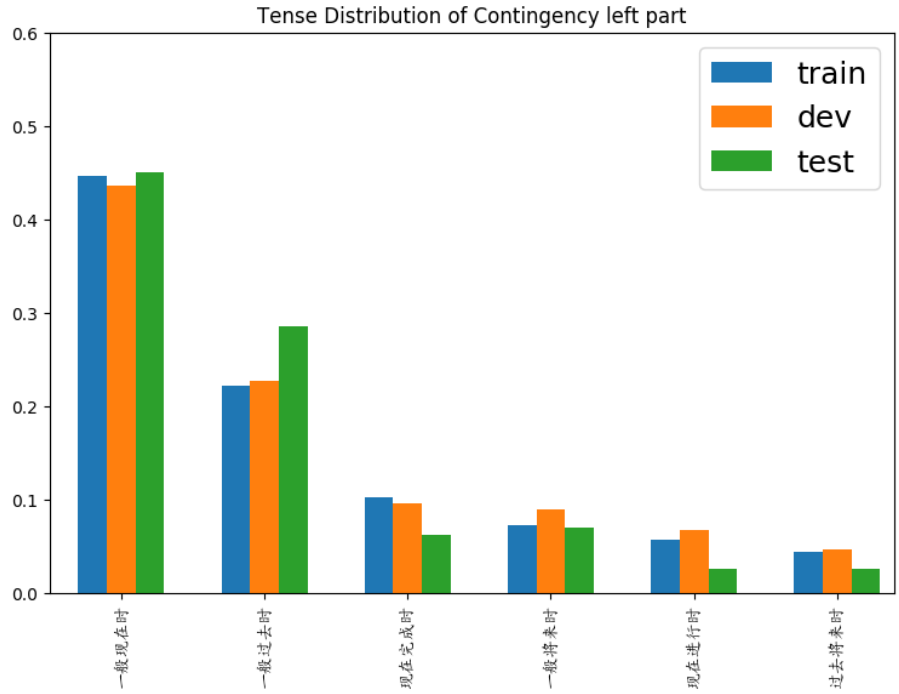
- 各关系的线索词集较小，可能存在对应的PDTB关系cover不够的问题
- 由于要确保pattern的质量，抽取的数量较少

3. 词表cover情况
    - **验证集**: 原因2195 结果1935 **因果向量cover**: 原因1958 结果1661
    - **测试集**: 原因1761 结果1651 **因果向量cover**: 原因1539 结果1445

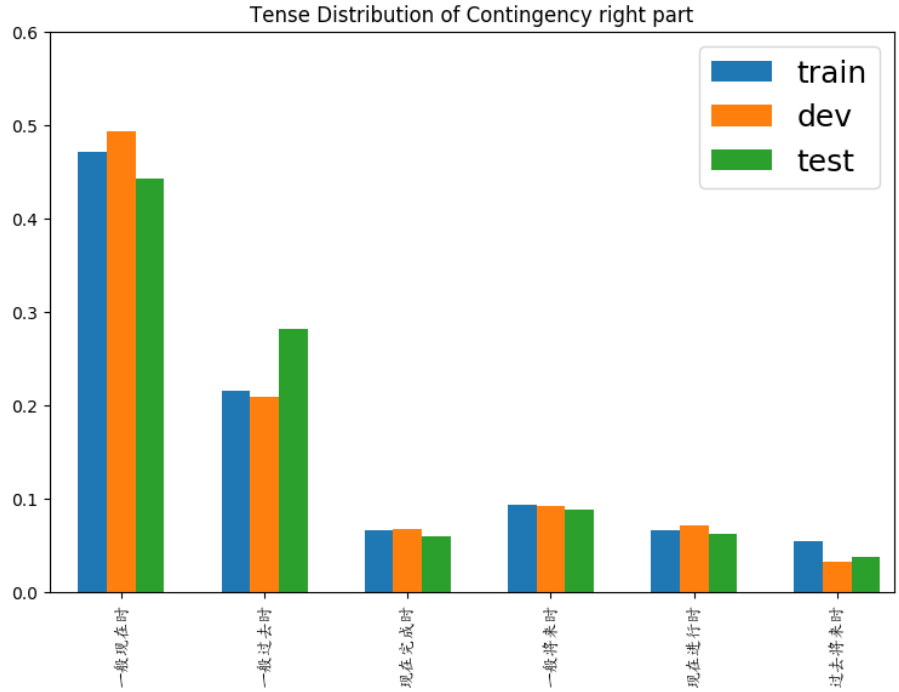
时态特征

Contingency关系,取最常见7种时态

- 原因句中时态分布:



- 结果句中时态分布



做过的尝试

1. 多任务：区分内容词，功能词
2. 加外部数据
3. 单任务分类：$Classifier_{arg1}$ + $Classifier_{arg2}$ + $Classifier_{<arg1,arg2>}$

数据集太小

可能的方向

- 如果$e^{'}$跟随e一起出现的概率大于其单独出现的概率，则e很可能是$e^{'} $的原因

$$p(e^{'}|e) > p(e^{'}) \rightarrow \frac{p(e^{'},e)}{p(e)p(e^{'})} > 1 \tag{3}$$

只有contingency有这个性质，其他关系不适用，使用因果向量作为特征,学习contingency关系

$ p(e^{'}|e)>p(e)\rightarrow\frac{p(e^{'},e)}{p(e)p(e^{'})}>1 $

- 句式特征：不考虑长期依赖，摒弃无关的词，使用cnn+self-attention学习重要的句式特征
- 频繁模式特征

通用词向量

负采样的问题

- 负采样是根据词频的，是静态的；而训练过程中，参数是动态变化的
- 没有观察到的高频词跟target word不相关
- 低频词仍然后可能包含重要信息
- 没法根据上下文针对性地选择有信息的负样本

adapter采样的问题

- 根据rank，选择排在前面的unobserved words. <火山，爆发>是一个频繁词对。选择基于rank选择“火山”的负样本，很可能选择到“喷发”

可能的方向

对窗口上下文加噪声，使用GAN根据synthetic上下文 生成负样本

Pattern表示

因果词对回溯语料

目的：保证precision的情况下，提升 recall，充分利用low precision线索词

做法：

现有high precision 线索（动）词$Verb_{hp}$ 、low precision 线索（动）词$Verb_{lp}$ 、因果向量 $causalEmbed_{hp}$ 、评估数据$Test_{word}$

1. 根据$causalEmbed_{hp}$找出因果词对（或者手工寻找），进而找出短语对
2. 短语对放回语料中，寻找共现的上下文(pattern)，对这些pattern打分
3. 选取打分高的pattern，抽取短语对，训练因果向量，希望其于$causalEmbed_{hp}$结果一样
4. **寻找数据集，实验验证**

细节：

- 因果词对选取

需要人工选择或设阈值选取，不够精确，覆盖度较低

- 短语对匹配

频繁模式匹配，词干替换，词性替换，

- 对pattern打分

使用这些pattern抽取短语对（或者抽取SVO），用Skip-thoughts（或 因果数据）对短语对进行打分，将短语对的打分转化为pattern的打分

- pattern 的权重

权重总和设个超参，按打分分配权重；先预计算权重表，训练时直接查表获取pattern的权重，权重在训练的时候调整

- 约束条件

添加语法约束，如短语对间的距离、词性约束，dependency依赖关系

- 损失

衡量两个矩阵或分布的差异性，KL，MMD

用什么损失函数