RoteExtractor 改进报告

作者：纽约的自行车

日期：2020 年 10 月 20 日

摘　　要

本文介绍了 RoteExtractor[1] 的改进工作。RoteExtractor 的原理可参考论文和文档[1]。

1　原有方案不足

原方案的抽取词评估方法采用有监督学习方式，训练分类模型 SVM、LR 对抽取词联合评分，选取大于阈值的抽取词。由于是有监督学习方式，故需要提供标签。用户需要提供大约 400 个正样本，然后算法随机生成 400 个负样本。同时，还需用户提供样本的特征用作模型训练。

这种方案需要用户提供的种子词数量太多，此外，样本特征需要用户根据语料仔细选择，模型才能取得较好地效果，因此不适用我们的使用场景。

2　改进方案

RoteExtractor 的基本思想和 Word2vec 相同，都是假设上下文相同的词具有相似的含义，Word2vec 用词的上下文构建词的分布式表达，而 RoteExtractor 用词的上下文构建 pattern，然后用 pattern 抽取新词，新词和原有词语义相似，达到构建领域词典目的。

RoteExtractor 认为词的上下文对词具有重要意义，那么用词的上下文对词进行评估显然是一种合理的方式。本文根据这个思想，针对原有方案的不足，提出了两个改进点：

1. pattern 的评估方式；
2. 抽取词的评估方式。

下文将词的上下文中的词称为**触发词**（trigger）。

2.1　触发词评分

为了使用触发词对 pattern 和抽取词进行评估，需要首先对触发词评分。本文使用触发词与种子词的 PMI（点互信息）值作为触发词得分。

$$score(w) = \frac{1}{N} \sum_{s \in seed} \frac{p(w,s)}{p(w)p(s)}$$

上式中 w 表示触发词，N 表示种子词数量，$seed$ 表示种子词集合，$p(w,s)$ 表示触发词和种子词在语料中共现概率，$p(w)$ 表示触发词在语料中出现概率，$p(s)$ 表示种子词在语料中出现概率。

[1] svn://10.48.60.63/Jiangxu/领域词典挖掘/RoteExtractor/文档/RoteExtractor 技术报告

2.2 pattern 评分

本文使用 pattern 包含的触发词对 pattern 进行评估。

$$score_1(pattern) = \frac{1}{N} \sum_{w \in triggers} trigger\_score(w) \tag{1}$$

上式中 N 表示 triggers 数量，triggers 表示当前 pattern 包含的触发词集合，trigger_score(w) 表示触发词 w 的分数。

Pattern 得分越高，说明其包含的触发词得分越高，而触发词得分越高说明其与种子词联系越密切，因此 pattern 得分越高表明其与种子词相关性越高。我们保留得分高的 pattern，但这将导致一个问题，保留的 pattern 之间非常相似。为了使保留的 pattern 之间具有一定的差异性，借鉴 MMR(Maximal Marginal Relevance) 算法思想，在 pattern 评估中引入 pattern 差异性评分，如下所示：

$$score_2(pattern) = \frac{1}{N} \sum_{p \in patterns} (1 - \frac{p\_triggers \cap pattern\_triggers}{p\_triggers \cup pattern\_triggers})$$

上式中 N 表示已存在的 pattern 数量，patterns 表示已存在的 pattern 集合，p_triggers 表示 p 包含的触发词集合，pattern_triggers 表示待评估 pattern 包含的触发词集合。

上式利用两个 pattern 的触发词交集与并集的比值表示相似性。结合 score_1 和 score_2，得到 pattern 最终的评估方式：

$$score(pattern) = \lambda \cdot score_1(pattern) + (1 - \lambda) \cdot score_2(pattern)$$

上式中 λ 是控制差异性的超参数。等式右边第一项表示当前 pattern 与领域的相关性，第二项表示当前 pattern 与现有 pattern 的差异性。

2.3 抽取词评估方式

抽取词评估由两部分组成，第一部分使用抽取词上下文的触发词评估，第二部分使用所有能抽取到该抽取词的 pattern 评估。

假设抽取词 e 由 pattern_A 抽取得到，那么 e 的第一部分评分等于 pattern_A 包含的触发词得分之和取平均。

$$score_1(e) = \frac{1}{N} \sum_{w \in triggers} trigger\_score(w)$$

上式中 N 表示 pattern_A 包含的触发词数量，triggers 表示 pattern_A 包含的触发词集合，trigger_score(w) 表示触发词 w 的得分。

假设抽取词 e 能被多个 pattern 抽取得到，那么 e 的第二部分评分等于这些 pattern 得分之和取平均。

$$score_2(e) = \frac{1}{N} \sum_{p \in patterns} pattern\_score(p)$$

上式中 M 表示能抽取到 e 的 pattern 数量，patterns 表示能抽取到 e 的 patterns 集合，pattern_score(p) 表示 p 的得分。抽取词 e 最终评估函数如下：

$$score(e) = \lambda \cdot score_1(e) + (1 - \lambda) \cdot score_2(e)$$

上式中 λ 是超参数。

3 实验分析

使用足球新闻作为文本数据，抽取足球俱乐部名称。俱乐部限于五大联赛，每个联赛选择一个球队作为种子词。代码和数据已上传服务器[2]。实验结果及分析详见文档《词典构建算法对比测试报告》[3]。

4 优点和不足

相比原有方案，改进方案无需用户提供大量正样本和样本特征，符合 Bootstrapping 算法风格，适用目前的使用场景。

缺点是 pattern 和抽取词的得分不是一个概率值，分值与语料库质量相关，导致难以设定阈值过滤 pattern 和抽取词。

5 与 AutoSlog 算法对比

从体育数据和景点数据的实验结果来看，RoteExtractor 相比 AutoSlog，对数据的冗余度要求更高。在数据冗余度高的语料中（如体育数据），RoteExtractor 精确度更高，但召回率相对较低。在数据稀疏的语料中（如景点数据），RoteExtractor 的召回率和精确度都很低，相比 AutoSlog 效果更差。

5.1 对语料冗余度要求分析

RoteExtractor 使用 pattern 抽取新词时，要求 pattern 的触发词在句子中都存在，而一个 pattern 具有多个触发词，所以 pattern 能匹配的句子少，这是导致其要求语料冗余度高的原因。而 AutoSlog 的 pattern 只有一个触发词和一个关系，因此在预料中能匹配的句子更多，对语料冗余度相对 RoteExtractor 较低。

5.2 召回率和精确度分析

RoteExtractor 的 pattern 具有多个触发词，同时存在这些触发词的句子才能被检索，因此对抽取词的上下文要求严格，其抽取的词与种子词相关性强，所以精确度高。AutoSlog 的 pattern 只有一个触发词和一个关系，含有该触发词的句子都能被检索，句子中与触发词具有 pattern 中指定关系的词将被抽取，因此对抽取词的限制不如 RoteExtractor 严格，其抽取的词与种子词相关性不如 RoteExtractor，但召回率高。

5.3 算法运行速度分析

限制算法运行速度的主要步骤是对检索句子的分析过程。RoteExtractor 算法对句子做分词和词性标注，AutoSlog 算法对句子做分词、词性标注和依存句法分析。由于哈工大 ltp 工具进行

[2]svn://10.48.60.63/Jiangxu/领域词典挖掘/RoteExtractor

[3]svn://10.48.60.63/Jiangxu/领域词典挖掘/RoteExtractor/文档

依存句法分析时速度慢，所以 AutoSlog 算法的运行速度相比 RoteExtractor 算法慢很多。在大语料库中更适合使用 RoteExtractor 算法。

参考文献

[1] ENRIQUE A, PABLO C, MANABU O, et al. A rote extractor with edit distance-based generalisation and multi-corpora precision calculation[C]. [S.l.]: Association for Computational Linguistics, 2006.