Joint Bootstrapping Machines for High Confidence Relation Extraction

论文学习报告

吴锐

2019.03.05

**1、主要内容**

该论文提出了一种新的bootstrapping方法，该方法通过确保在每个循环中抽取出来的实例具有较高的置信度，来解决传统bootstrapping中存在的语义漂移问题。其主要思想在于将实体种子及模板种子结合使用，在每次迭代的过程中两者相互约束、并行扩展，并且为模板引入更高质量的相似性度量。

其主要贡献有三点：

1. 提出了一个Joint Bootstrapping Machine，提升了无噪声抽取器的所能抽取的正例数量，并显著提升了其置信度。使用实体对、模板或者两者结合作为种子进行bootstrapping，最终得到的抽取器分别叫做BREE，BRET和BREJ。
2. 引入全文本相似度（cross-context similarities），比较上下文的所有部分，较之前的“假定关系具有一致的句法表达式”，在召回率上表现的更好。
3. 实验结果表明在四种关系上，较目前的最优方法在F1上提升了13%。

**2、方法细节**

**2.1 符号与定义**

由于该论文中符号较多，首先需要介绍一下一些符号的定义。在实验之前，需要进行一定的预处理，为所有候选实体标上一个类别。所以改论文中bootstrapping算法所处理的实体均为typed entities。

对于语料库中某一句表达了x与y之间的关系的一句话，一个**模板**（template）包含三个向量：表示x前面的文本，表示x与y之间的文本，表示y后面的文本，这些向量都是简单的每个单词的embedding的sum。模板也是有类型，即需要x与y符号某一特定的类型。一个**实例**（instance）表示一个模板和一个实体对的结合，两者的类型需要一致。

Bootstrapping的第一步是从语料库中提取一组实例，使用γ来表示该组实例，i和j表示其中的某一实例。X(i)表示实例的实体对，Y(i)表示实例对应的模板。

图1中算法的输入除了γ之外，还包括实体对（和）或模板的（和）正例与负例种子，或者两者都有。这里使用来表示这个四元组。

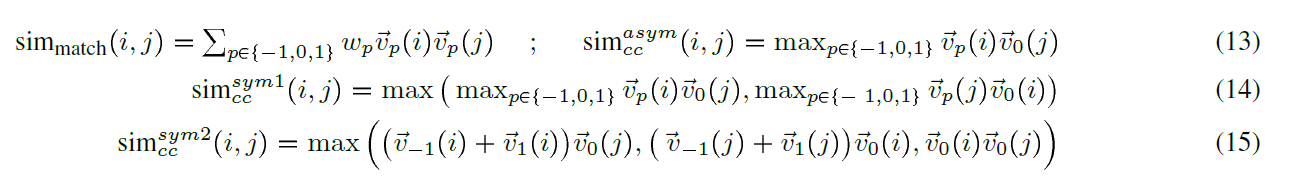
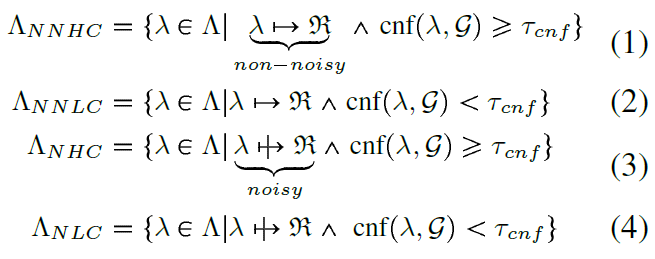


图2

有关相似度的计算，该论文对不同的相似度的度量方式进行了实验，其baseline为图2中第一行：两个实例的相似度计算为他们三个的向量之间点积的加权求和。单个实例与一组实例之间的相似度，定义为该实例与该组中所有实例的相似度的最大值。具体可以看图1中式（5）（6）。

一个抽取器可以被分成以下几类：



其中表示对应的关系（relation）。λ表示中的一个。例如，一个叫做无噪声低置信度抽取器，表示它能够体现出目标关系，但是其置信度低于某个阈值。和对于bootstrapping来说是有用的，而其余两者是没用的。

**2.2 Boottrapping方法：BREX**

为了描述方便，此处使用item来表示实体对、模板或者两者兼有。

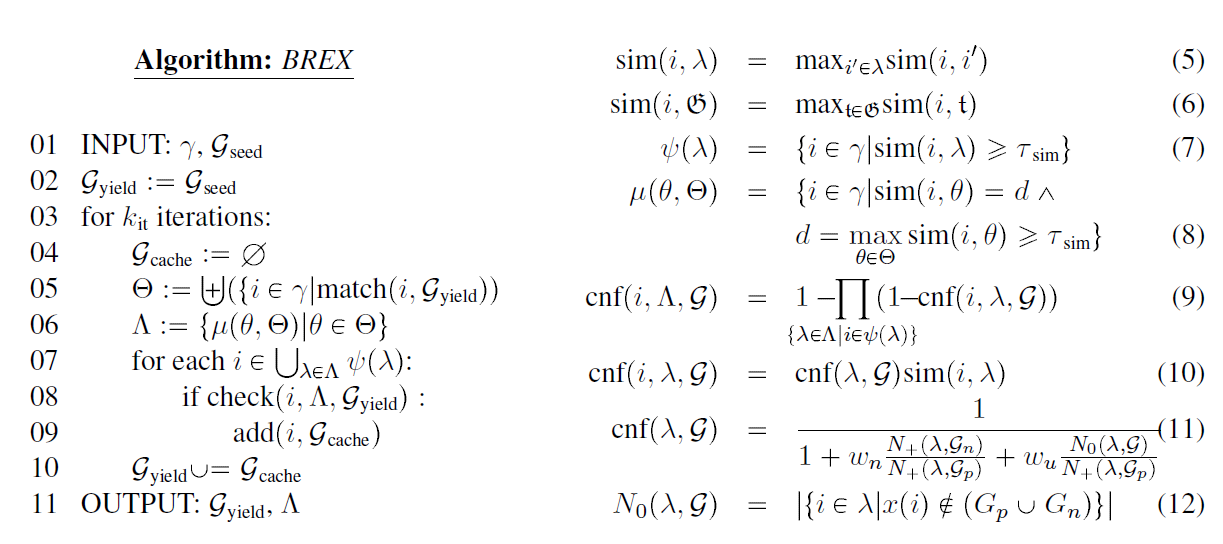


图1

图1左侧是该论文中bootstrapping算法的全过程。

输入为γ以及，上面已经描述过其来源及组成。表示多轮循环BREX抽取出来的items，表示每一轮循环抽取出来的items。

该算法的设计平衡了保证高的召回率的元素与保证高的精确率的元素。

算法通过三步来得到一个高的召回率：

（1）05行，收集γ中所有与某一个种子相似的实例，这个相似的计算对于不同的BREX是不同的。如下图：

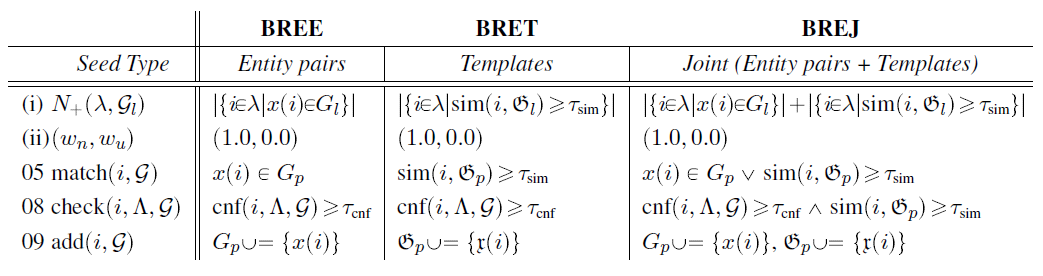


图3

接下来所有收集到的实例会被分组。（文中未提及分组的方式）；

（2）06行，添加γ中所有与第一步中得到的任一实例相似度高于的实例，并且每个实例只能被添加到相似度最高的组内，参见定义（8）；该步内的得到的每一组实例，文中称之为一个抽取器（extractor）

（3）07行是第三步，收集γ中所有与第二步中得到的任一实例相似度大于的实例，参见定义（7）。

接下来，在08行，在将实例的信息添加到之前，对每一个实例做一个严格的检查，以达到较高的精确率。该检查的核心函数在于式（9）。可以看到，式9其实可以理解为针对γ中每一个实例，对应到式10的值都越大越好。式10表达了一个抽取器λ对于一个实例的置信度（confidence），定义是λ的整体的可靠度（与实例i无关）乘以实例i与λ的相似度（式10的第二个因子）。后者避免了抽取器中的实例与语料库中的实例差距过大，即使抽取器本身可靠度很高。

那么式10中的第一个因子是如何表示抽取器的可靠程度呢？其主要是通过计算一个比率： ，具体见图3中的（i）式。如果这个比率接近于0，则很有可能正例的抽取比负例的抽取要多；将该算法简化（），则其在cnf函数中的结果接近于1，其可靠性的度量并无折扣。但另一方面，如果该比率较大，则表示负例抽取的个数可能比正例的个数要大，则会导致cnf函数接近于0，其被分类为一个无噪声低置信度的分类器（non-noisy-low-confidence extractor）。由于缺少标注数据，评分机制难以区分有噪声和无噪声的抽取器。因此，该论文考虑通过能否提取更多的正例与更少的反例来判别一个抽取器。这种可靠性的统计是在第二步的结果上进行计算的，

最后，所有通过检查的实例被加入到cache当中，然后内循环结束，cache被合并到yield中。然后新的一轮循环又开始了。

一个内循环的过程的图例如下：

