Snowball

作者：李一飞 时间：2019-07-23

本文讨论了一种利用很少的seed set，通过迭代，从文本中提取关系的技术。DIPRE主要是做这个工作。本文提出的snowball系统是在这个基础上，引入了一些新策略以及评估标准，并用30万份报纸中的数据做了实验。

1 Introduction

本文的方法是建立在DIPRE基础上的一个改进。本部分首先介绍了一下DIPRE。文章主要以<organizaiton, location>格式的tuple作为例子讨论。DIPRE首先需要若干个元组作为seed，然后按照这些seed去数据集中匹配关系（pattern），用这些匹配到的关系去查询新的<o, l>对，添加到seed set中，再用新的seed去匹配新的pattern，不断迭代，直到效果不再有明显提升，或者已经获取了一定数量的tuple为止。

本文的contribution：

- 提出了一种生成pattern和提取tuple的技术
- 提出了评估pattern和tuple的策略
- 提供了评价方法和指标定义

2 The Snowball System

2.1 Generating Patterns

这里在DIPRE上进行了一点改进，规定o和l必须是命名实体，且是同一个tag下的命名实体。

snowball中pattern是一个五元组：<left, tag1, middle, tag2, right>。其中tag1和tag2是打好标签的命名实体，l，m和r是带权重的表示上下文的向量。用五元组与包含tag1和tag2的文本相匹配，从上下文创建出三个向量l_s，m_s和r_s。每个向量包含了一个权重，代表当前的向量在上下文中出现的频率。

定义了一个Match函数：

Definition 2 *The degree of match* $Match(t_P, t_S)$ *between two 5-tuples* $t_P =< l_P, t_1, m_P, t_2, r_P >$ *(with tags* t_1 *and* t_2*) and* $t_S =< l_S, t'_1, m_S, t'_2, r_S >$ *(with tags* t'_1 *and* t'_2*) is defined as:*

$$Match(t_P, t_S) = \begin{cases} l_P \cdot l_S + m_P \cdot m_S + r_P \cdot r_S & if\ the\ tags\ match \\ 0 & otherwise \end{cases}$$

具体原理是snowball为每个与seed中的tag匹配的字符串生成一个五元组，然后跑一个**singlepass聚类算法**，计算它们之间的匹配度并定义一个相似度阈值τ_{sim}。这些五元组聚类的left，middle，right的重心由$\bar{l_s}$，$\bar{m_s}$，$\bar{r_s}$表示。这三个重心加上原来的两个tag构成了一个pattern$< \bar{l_s}, t_1, \bar{m_s}, t_2, \bar{r_s} >$。

2.2 Generating Tuples

生成tuple的算法如下：

```
sub GenerateTuples(Patterns)
  foreach text_segment in corpus
(1) {< o, ℓ >, < l_s, t_1, m_s, t_2, r_s >} =
      = CreateOccurrence(text_segment);
    T_C = < o, ℓ >;
    Sim_Best = 0;
    foreach p in Patterns
(2)   sim = Match(< l_s, t_1, m_s, t_2, r_s >, p);
      if (sim ≥ τ_sim)
(3)     UpdatePatternSelectivity(p, T_C);
        if(sim ≥ Sim_Best)
          Sim_Best = sim;
          P_Best = p;
    if(Sim_Best ≥ τ_sim)
      CandidateTuples[T_C].Patterns[P_Best] =
        = Sim_Best;
  return CandidateTuples;
```

Figure 4: Algorithm for extracting new tuples using a set of patterns.

先用seed set中的<o, l>对从文本中提取出tag匹配的五元组，再遍历已提取出来的pattern集合，与其进行匹配，如果出现相似度大于阈值的tuple，则更新当前pattern的selectivity并更新SimBest。遍历结束之后若最好的匹配度满足最低阈值，就将此tuple放进候选tuple中，同时赋以与其匹配度最高的pattern。

2.3 Evaluating Patterns and Tuples

文章举了一个反例，比如<{}, *ORGANIZATION, <",", 1>, LOCATION, {}*>这种元组（两边都是空格，英语中很常见的表达，比如"Microsoft, Redmond"），会出现很多错误的匹配，因此提出了可信度评估的概念，丢弃掉那些可信度低的。tuple的置信度是由pattern的selectivity和数量决定的。如果一个tuple是由几个高选择性的pattern生成的，则会具有较高的可信度。

首先，筛掉所有含有匹配的tuple个数小于τ_{sup}的pattern。然后在上面生成tuple的算法执行步骤(3)时更新pattern的selectivity和数量。如果检查$t =< o, l >$时有一个先前生成的$t^{'} =< o, l^{'} >$存在，则比较l和$l^{'}$，若相同则判定为positive，否则判定为negative。最终这个tuple的得分Conf(P)是所有positive和negative匹配数之和中positive所占的比例。

还定义了一种RlogF置信度：$ConfRlogF(P) = Conf(P) \cdot log2(P.positive)$，并且规范化到0-1之间。

通过模式生成有效tuple的概率$Prob(P_i)$来估计元组T valid的概率：

$$Prob(T) \quad = \quad 1 - \prod_{i=0}^{|P|} (1 - Prob(P_i))$$

元组T的置信度：

$$Conf(T) = 1 - \prod_{i=0}^{|P|} (1 - (Conf(P_i) \cdot Match(C_i, P_i)))$$

为控制系统的学习率，将P的置信度设置为：

$$Conf(P) = Conf_{new}(P) \cdot W_{updt} + Conf_{old}(P) \cdot (1 - W_{updt})$$

这样，每次迭代后用于下一次迭代的种子集合是$Seed = T | Conf(T) > \tau t$。

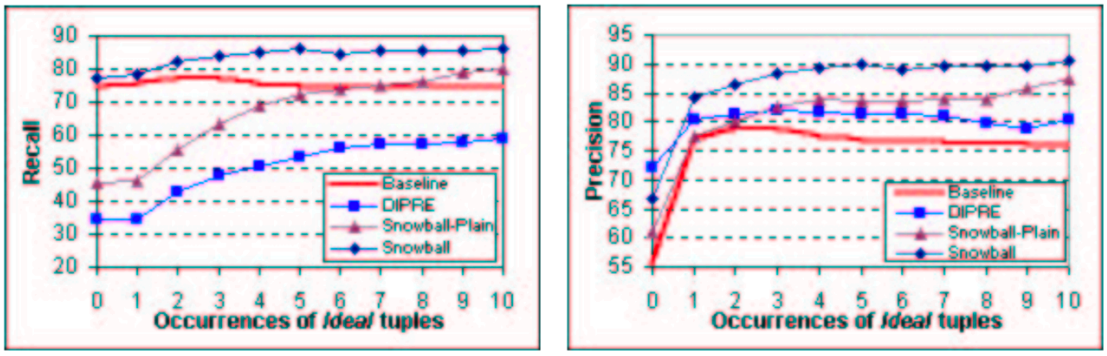
3 Evaluation Methodology and Metrics

与传统的信息提取不同，本文不在于将一个tuple的所有实例都提取出来，而是为每一个元组提取一个实例，由于元组一般都会在文字中出现多次，因此只要正确提取出一个实例就是成功的。实验在ideal集合上判断提取出来的tuple的召回率和准确率。

本节主要介绍了其对ideal数据集的处理，将o和o'进行了一些处理使其一致，并且对Recall和Precision进行了计算。还有就是提取出来元组的实际意义问题，文章规定(1)o位于美国，l给出其所在的城市或州或 (2)o位于国外，l给出其所在的城市或国家 均为正确的。

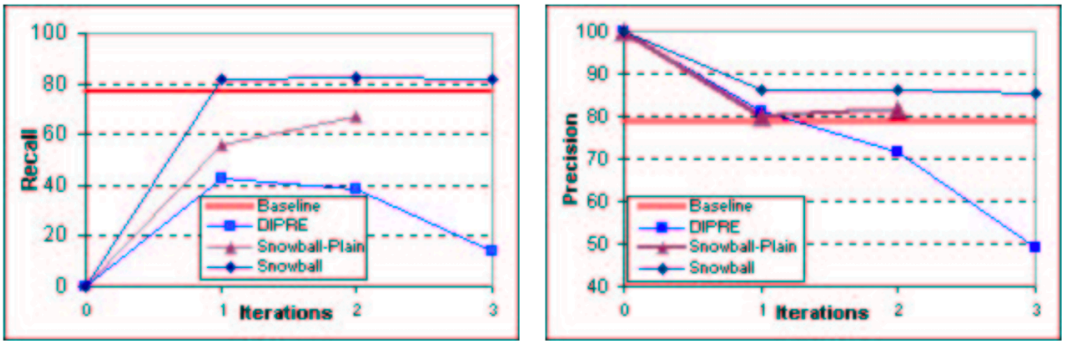
4 Experiments

实验结果：



(a) (b)

Figure 5: Recall (a) and precision (b) of *Baseline*, DIPRE, *Snowball* and *Snowball-Plain* (test collection).

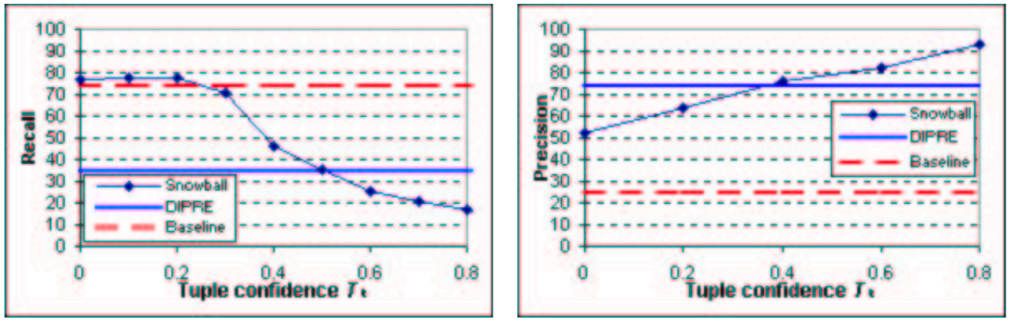


(a) (b)

Figure 6: Recall (a) and precision (b) of *Baseline*, DIPRE, *Snowball*, and *Snowball-Plain* as a function of the number of iterations (*Ideal* tuples with occurrence $\geq$ 2; test collection).

			Type of Error			
	Correct	*Incorrect*	*Location*	*Organization*	*Relationship*	P_{Ideal}
DIPRE	74	26	3	18	5	90%
Snowball (all tuples)	52	48	6	41	1	88%
Snowball ($\tau_t = 0.8$)	93	7	3	4	0	96%
Baseline	25	75	8	62	5	66%

Table 5: Manually computed precision estimate, derived from a random sample of 100 tuples from each extracted table.



(a) (b)

Figure 7: Recall (a) and *sample-based* precision (b) as a function of the threshold τ_t used for the last-step pruning of the *Snowball* tables (*Ideal* tuples with occurrence ≥ 1; test collection).