

1. 问题与贡献

(1) 在实体集扩展 (ESE) 中, Bootstrapping 方法常常导致引入噪声实体、语义漂移和难以利用高阶实体-模式之间关系等问题。

(2) 本文提出了一种用于实体集扩展的端到端自助抽样神经网络

BootstrapNet, 该模型在编码器-解码器体系结构中对自助抽样进行建模。

1) 编码阶段, 用图注意力网络来获取实体和模式之间的一阶和高阶关系, 对其有用的信息进行编码。

2) 解码阶段, 通过 GRU 依次扩展实体, 并且输出实体的同时更新代表目标类别的隐藏状态向量。

3) 设计了一种仅用几个种子作为监督信号的学习算法。

2. 实体-模式二分图和 BootstrapNet 模型

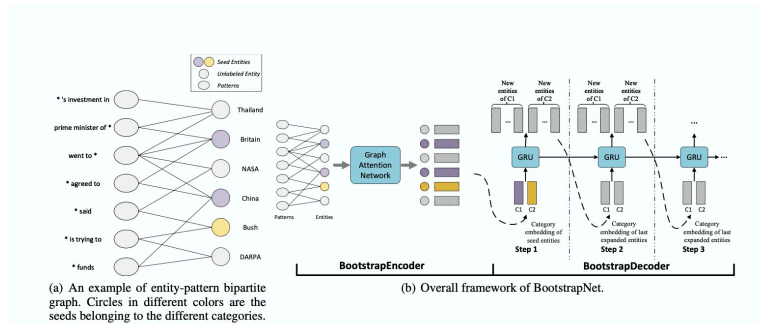


Figure 2: The input and the overall framework of BootstrapNet.

(1) 输入: 实体-模式条目

(2) 输出: 新扩展的实体类型

(3) 工作流程

1) 输入: $\langle \text{entity, context pattern} \rangle$, 如 $\langle \text{Bush, is trying to} \rangle$, 初始向量表示为 GloVe。

2) 提取实体和模式间的信息: BootstrapNet 通过 $\langle \text{entity, context pattern} \rangle$ 条目来构造实体模式二分图, 见 Figure 2(a)。通过二分图之间的多跳路径获得实体和模式之间的高阶关系。 $G = \langle V, E, L \rangle$, V 为实体节点和模式节点, E 为联系节点的边, L 是种子实体的标签。

3) BootstrapNet: 输入二分图, 更新实体的向量表示, 依次输出新的实体。

a) BootstrapEncoder: GAN, 基于图注意力网络的编码器, 对一个节点 v_i 分为两个部分, 第一部分学习 $l-1$ 层的信息, 第二部分编码相邻节点的信息。

第一部分: 输入为二分图, 用 BootstrapEncoder 获取实体和模式

$$\alpha_i^l = W_v^l s_i^{l-1} \quad (1)$$

之间的一阶和高阶关系 α_i^l ;

第二部分：整合当前节点和相邻节点的相关性信息 β_i^l ，最后将 l 层 v_i

节点的新的信息相邻表示 $s_i^l = \sigma(\alpha_i^l + \beta_i^l)$ 送到 BootstrapDecoder

$$\beta_i^l = \sum_{j \in \mathcal{N}(i)} a_{i,j}^l W_v^l s_j^{l-1} \quad (2)$$

中，以生成新的实体；

- b) BootstrapDecoder：用 GRU 实现的顺序实体生成解码器。两个阶段：类别更新和实体生成。GRU 的隐状态表示实体类别，先根据输入来更新隐状态(类别表示)，并根据隐藏层和输入的相似性来生成新实体作为输出。

第一部分：类别更新阶段，BootstrapDecoder 将扩展后的实体嵌入作为输入，并按照以下方式更新隐藏状态(也就是实体类别)：

$$\begin{aligned} z_c^t &= \sigma(W_z \cdot s_c^t + U_z \cdot h_c^{t-1}) \\ r_c^t &= \sigma(W_r \cdot s_c^t + U_r \cdot h_c^{t-1}) \\ \tilde{h}_c^t &= \sigma(W \cdot s_c^t + r_c^t \cdot U \cdot h_c^{t-1}) \\ h_c^t &= z_c^t \odot h_c^{t-1} + (1 - z_c^t) \odot \tilde{h}_c^t \end{aligned} \quad (5)$$

第二部分：实体生成阶段，根据待扩展实体和隐藏层状态的余弦相似度生成实体。在第 t 步，待扩展的实体被标注为具有最高概率的类

$$p(c|i) = \frac{g(c,i)}{\sum_{c' \in C} g(c',i)} \quad (6)$$

别， $g(c,i) = 0.5 + 0.5 * \text{sim}(h_c^t, s_i)$ ，其中 h 是更新后的隐层， s_i 是实体， sim 余弦距离； g 是将余弦距离缩放为 $[0, 1]$ 的归一化函数。对于标记为同一类的实体，只有概率最高的前 N 个实体会被扩展。

在 a) 步骤的等式(2)中， W 是规范化权重，用于汇总信息时的过滤噪

$$\begin{aligned} a_{i,j}^l &= \frac{\exp(e_{i,j}^l)}{\sum_{k \in \mathcal{N}(i)} \exp(e_{i,k}^l)} \\ e_{i,j}^l &= f(W_a^l h_i^l, W_a^l h_j^l) \end{aligned} \quad (4)$$

声，对于节点 v_i 用自关注打分的形式计算权重，

3. 模型的学习问题

种子实体是第一步作为输入提供的，没有别的监督来评估 decoder 的连续输出，所以多数半监督学习算法不适用于该模型。

为解决这个问题，使用多视图算法优化 BootstrapNet。首先构建一个 BootstrapTeacher 的辅助神经网络，该网络和 BootstrapNet 共享相同的编码器，但是它将编码器直接连接到多层感知机（MLP）分类器，从而对每个实体进行分类。

BootstrapTeacher 对非顺序扩展过程进行建模，分类结果仅取决于各个实体的表示形式；BootstrapNet 对顺序扩展过程建模，考虑当前扩展对连续扩展的影响。

Algorithm 1 Optimization Algorithm

Input: A bipartite graph G , and seed entities with category labels (L, y_L)

Output: Expanded entities for each category

- 1: Construct BootstrapTeacher with the BootstrapEncoder followed by an MLP classifier
 - 2: Pre-train BootstrapTeacher according to Eq. (9)
 - 3: **while** convergence criteria not met **do**
 - 4: Copy Encoder's parameters in BootstrapTeacher to BootstrapNet
 - 5: Annotate unlabeled entities with BootstrapTeacher
 - 6: Update BootstrapNet with Eq. (10)
 - 7: Expand new entities with labels using BootstrapNet
 - 8: Update BootstrapTeacher with Eq. (9)
 - 9: **end while**
 - 10: Expand new entities using the learned BootstrapNet.
-

4. 实验结果

使用 ontoNotes 和 CoNLL 两个数据集。实验证明，Encoder 和 Decoder 对模型最终性能有很大的贡献。

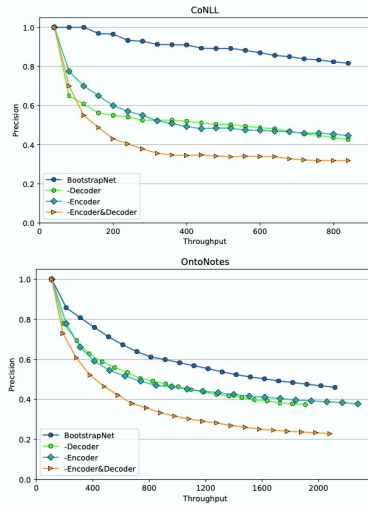


Figure 4: An ablation study of the BootstrapNet.

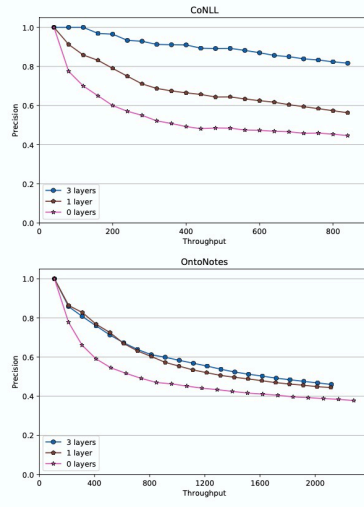


Figure 5: The performance of BootstrapNet with different numbers of layers in the BootstrapEncoder.

论文阅读笔记二

Gazetteer-Enhanced Attentive Neural Networks for Named Entity Recognition

1. 解决的问题与贡献

(1) 问题：缺乏训练数据会导致性能下降，本文提出的方法适用于缺乏训练数据的场景。

解决思路：本文提出基于 gazetteer 词典的增强型注意力神经网络，从 gazetteer 词典中学习实体指称的名称知识，用 ANN 显式地对实体的上下文关系进行建模。

模型：模型包括 Inner-Region Encoder、Attentive Context Encoder、Utterance Encoder 三个部分。

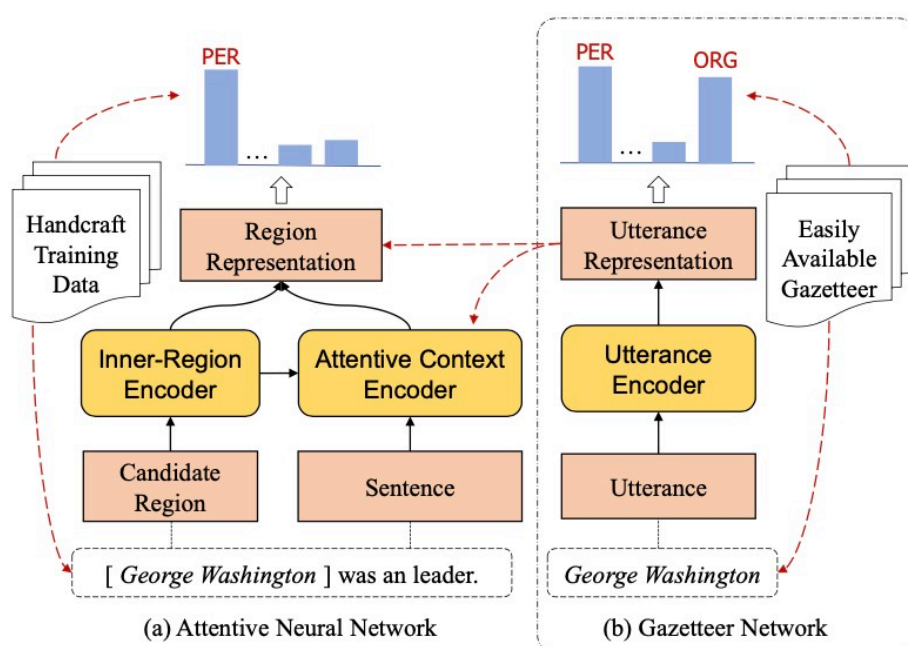


Figure 1: The overall architecture of GEANN. The candidate region is “George Washington”, which literally could be a person or an organization (university).

step0: 给定句子，ANN 首先将单词映射为词向量 x_i ，再用 BiLSTM 获取单词的上下文表示形式 h_i^A 。

Step1: 用 Inner-Region encoder 获取 name knowledge，考虑单词的边界和内部信息 s_{ij}

$$s_{ij} = \text{MLP}([h_i^A; \frac{1}{j-i+1} \sum_{k=i}^j h_k^A; h_j^A]), \quad (1)$$

Step2: 用 Attentive Context Encoder 获取 context knowledge：用 inner-region encoder 得到的向量对其他单词做 attention：

$$\begin{aligned} \mathbf{c}_{ij} &= \sum_{k=1}^{i-1} \alpha_{ijk} \mathbf{h}_k^A + \sum_{k=j+1}^n \alpha_{ijk} \mathbf{h}_k^A \\ \alpha_{ijk} &= \frac{\exp(e_{ijk})}{\sum_{t=1}^{i-1} \exp(e_{ijt}) + \sum_{t=j+1}^n \exp(e_{ijt})} \end{aligned} \quad (2)$$

where

$$e_{ijk} = \tanh(\mathbf{s}_{ij}^T \mathbf{A} \mathbf{h}_k^A + \mathbf{W} \mathbf{h}_k^A + b) \quad (3)$$

Step 3: Utterance Encoder 语料编码器，对给定语料 $\mathbf{u} = \{\mathbf{u}_i, \dots, \mathbf{u}_j\}$ ，首先用 Utterance Encoder 学习 \mathbf{u} 的向量表示，然后用 \mathbf{u} 计算该词为某有效实体的概率 \mathbf{O} :

$$\mathbf{O}_u^G = s(\mathbf{W} \mathbf{u} + \mathbf{b}). \quad (4)$$

然后通过多标签、多类别交叉熵损失函数对网络进行训练:

$$\mathcal{L}^G(\theta) = \sum_{\mathbf{u} \in G} [g'_u \log \mathbf{O}_u^G + (1 - g'_u) \log(1 - \mathbf{O}_u^G)] \quad (5)$$

实验结果: 在 ACE2005 数据集上，该模型较 BaseLine 有较大的提升，并且也能在 Bert 模型的基础上进行 Fine Tune。

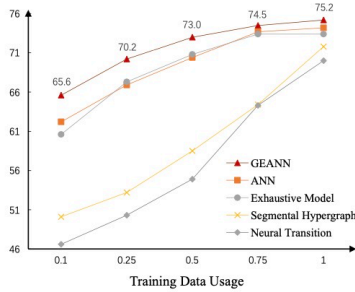


Figure 2: F1-scores on ACE2005 when training data size varies.

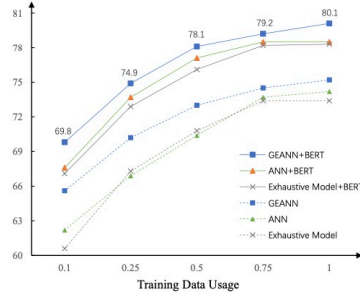


Figure 3: F1-scores of models with BERT on ACE2005 when training data size varies.