Lec10 序列标注

☆标注任务

△Part of speech tagging (POS) → 词性标注

2类词性标签：{ Open：表具体含义
Closed：语法功能

为后续Tasks铺垫：

- parsing (syntactic)
- translation
- Sentiment/affective tasks → 如评论adj.
- text to speech

△ Chinese word segmentation (B、I、E、S)

△ Named entity recognition (B、I、E、S、O)
(-PER、-LOC、-ORG)

△ Semantic role labeling (语义角色标注)
识别谓词及其语义角色(施动、受动等)
(-PRED、-ARG0、-ARG1) (B、I、E、S、O)

△ For POS Easiest Way:
每个词预测固定标签
⇒未考虑上下文/相邻标签间关系
⇒可采用其他方法进行POS Tagging!
⇒HMM、CRF、Neural

☆HMM Model (隐马尔可夫)

△ Model

$P(x_1 \cdots x_n, y_0 \cdots y_{n+1}) = q(y_{n+1}|y_n) \prod_t q(y_t|y_{t-1})\, e(x_t|y_t)$

$x_1 \cdots x_n$：输入Seq　$y_0 \cdots y_{n+1}$：标签Seq　$q(y_{n+1}|y_n)$：终止概率　$q(y_t|y_{t-1})$：转移概率　$e(x_t|y_t)$：发射概率

y_0: Start　y_{n+1}: Stop

0阶：>2标签关系：(High order HMM)

eg: $P(y_t|y_{t-1} \cdots y_{t-n+1}) \Rightarrow$ Similar to n-gram

△ Decoding

Stop

搜索找到 $\hat{y} = (y_1, y_2 \cdots, y_{n+1})$，使

$\hat{y} = \underset{(y_1 \cdots y_{n+1})}{\operatorname{argmax}} P(y_1 \cdots y_{n+1}|x_1 \cdots x_n) = \underset{(y_1 \cdots y_{n+1})}{\operatorname{argmax}} P(y_1 \cdots y_{n+1}, x_1 \cdots x_n)$

$= \prod_t P(y_t|y_{t-1})\, P(x_t|y_t)$　(转移+发射模型)

O(暴力枚举) → exp复杂度

Viterbi Algorithm:

$\pi(i, y_i)$：前i位置、末尾为y_i的最大概率.

$\pi(i, y_i) = \max_{y_1 \cdots y_{i-1}} P(x_1 \cdots x_i, y_1 \cdots y_i)$

$= \max_{y_{i-1}} \left[e(x_i|y_i)\, q(y_i|y_{i-1}) \right] \pi(i-1, y_{i-1})$　→只看最后一步建模

$= e(x_i|y_i) \cdot \max_{y_{i-1}} q(y_i|y_{i-1})\, \pi(i-1, y_{i-1})$　DP转移方程

↳ 前变量不含y

① Initialize: $\pi(0, \text{Start}) = 1$, for other y_0: $\pi(0, y_0) = 0$

② 第i步时：计算所有标签y_i的$\pi(i, y_i)$

③ $P(\hat{y}) = \max_{y_n} q(\text{Stop}|y_n)\, \pi(n, y_n) = \pi(n+1, \text{Stop})$

↳ 考虑到终点

复杂度：序列长n、标签数L ⇒ 算nL个π( )

⇒算每个π( )耗时O(L) ⇒ 总耗时$O(nL^2)$

△ Marginal Inference

计算 $P(x_1 \cdots x_n) = \sum_{y_1 \cdots y_n} P(x_1 \cdots x_n, y_1 \cdots y_{n+1})$

(输入序列概率)

枚举标签 ⇒ 指数复杂度

⇒ $\alpha(i, y_i)$：前 i 位置、末尾为 y_i 边的概率：

$= P(x_1 \cdots x_i, y_i)$

$= \sum_{y_1 \cdots y_{n-1}} e(x_i|y_i) q(y_i|y_{i-1}) \cdot P(x_1 \cdots x_{i-1}, y_1 \cdots y_{i-1})$

$= e(x_i|y_i) \sum_{y_{i-1}} q(y_i|y_{i-1}) \alpha(i-1, y_{i-1})$ 递推方程

① 初始化：$\alpha(0, Start) = 1$，其余 $\alpha(0, y_0) = 0$

② 第 i 步计算所有 $\alpha(i, y_i)$ → L 个标签

③ $P(x_1 \cdots x_n) = \sum_{y_n} q(Stop|y_n) \alpha(n, y_n) = \alpha(n+1, STOP)$

复杂度：$nL \cdot L \Rightarrow O(nL^2)$

△ Supervised Learning 为数据

MLE：$P(x, y) = \prod_{i=1}^{n+1} e(x_i|y_i) q(y_i|y_{i-1})$ → 转移

$= \left(\prod_{i,j \in I} q(j|i)^{c(i,j)} \right) \left(\prod_{j \in X} \prod_{i \in I} e(j|i)^{c(i,j)} \right)$

↳ 转移部分　↳ 发射部分

(概率最大)　→ 计数函数

闭式解：$e(v|u) = \frac{c(u,v)}{\sum_{v' \in X} c(u,v')}$

$q(r|o) = \frac{c(o,r)}{\sum_{r' \in I} c(o,r')}$

(稀疏性处理：用 N-grams)

△ Unsupervised Learning

(↳ 只有 x 无 y^* → 寻找隐性规律

→ EM (Baum-Welch Algorithm) For HMM

○ E 步：算分布参数下 Y 分布 → exp 级

⇒ 期望计数：

$c(S) = E_{P(y_1 \cdots y_{n+1} | x_1 \cdots x_n)} [count(S | x_1 \cdots x_n, y_1 \cdots y_{n+1})]$

⇒ 描述序列下标为 S 期望出现次数

$c(NN) = \sum_i P(y_i = NN | x_1 \cdots x_n)$

$c(NN \to VB) = \sum_i P(y_i = NN, y_{i+1} = VB | x_1 \cdots x_n)$

$c(NN \to apple) = \sum_{i: x_i = apple} P(y_i = NN | x_1 \cdots x_n)$

⇒ 利用 Forward-Backward Algorithm (α, β)

○ M 步：计算新参数 (e、q)

期望：$E_{Q(y_1 \cdots y_{n+1})} [\log P(x_1 \cdots x_n, y_1 \cdots y_{n+1})]$

↳ 待更新参数

(极大似然) → e_{ML}、q_{ML}

☆ 除 EM 外：也可用梯度下降求解

$P(x_1 \cdots x_n)$：

计算边际概率 $\alpha(n+1, Stop)$ (前向算法)

再在计算图上反向传播 → e、q 参数导对

→ 与前一版向完全一致

△ MEMM (最大熵 Markov)

Background: HMM 可解决相邻 label 关系估计，解决上下文关系

变化过程：

HMM: $P(x_{1:n}, y_{1:n}) = \prod_t P(y_t|y_{t-1}) P(x_t|y_t)$（转移 / 发射）

↳ generative model

MEMM: $P(y_{1:n}|x_{1:n}) = \prod_t P(y_t|y_{t-1}, x_t)$ （从上一标签&该步输入评判）

↳ discriminative model

$\frac{1}{Z(y_{t-1}, x_t)} \exp(S(y_{t-1}, y_t, x_t))$ → 匹配得分后就归一化

$= W^T f(y_{t-1}, y_t, x_t)$　　可分解为 $S_e(y_t, x_t) + S_q(y_t, y_{t-1})$

label bias problem

(最终模型)

MEMM: $P(y_{1:n}|x_{1:n}) = \prod_t P(y_t|y_{t-1}, x_{1:n})$

↳ $\frac{1}{Z(y_{t-1}, x_{1:n})} \exp(s(y_t, y_{t-1}, x_{1:n}))$ → label bias

(但解决了上下文问题)

Sol: 不考虑概率而是得分

△ CRF

$P(y_{1:n}|x_{1:n}) = \frac{1}{Z(x_{1:n})} \prod_t \exp(s(y_{t-1}, y_t, x_{1:n}))$

（全局归一化；取决于上一标签&全局输入）

o Decoding

找到 $y^* = \underset{y_1 \dots y_n}{\arg\max} \frac{1}{Z(x_{1:n})} \prod_{t=1}^{n+1} \exp(S(y_{t-1}, y_t, x_{1:n}))$

$= \underset{y_1 \dots y_n}{\arg\max} \sum_{t=1}^{n+1} s(y_{t-1}, y_t, x_{1:n})$

⇒ Viterbi 算法求 max

$\pi(i, y_i) = \max_{y_{i-1}} [s(y_{i-1}, y_i, x_{1:n}) + \pi(i-1, y_{i-1})]$

$(\pi(0, Start) = 0)$ → 初始得分

o 监督学习

1、负对数条件似然 L 最小化

① Forward Algorithm 计算 $Z(x_{1:n})$ ⇒ 算出 $L = -\log P(\vec{y}^* | \vec{x})$

② 梯度下降

{ 梯度包含期望计数 c(S) → 前向后向算法
{ auto-differentiation

2、最小化边际损失 → SSVM

$L_{SSVM} = \max_{y_{1:n}} (s(y_{1:n}) + \Delta(y_{1:n}, y^*_{1:n}) - s(y^*_{1:n}))$

↳ 二者损失，≥ 0 (相似程度)

→ 考虑的是 decision boundary 而非整体分布

↳ 即：不注重状态的概率分布，只注重标签评分

→ 如果 Δ 可按位置分解 → L_{SSVM} 可用 Viterbi 求解

优化: { subgradient
{ quandratic programming

o 无监督学习

直接通过 y_i 发射出 $\hat{x}_i$ (Decoder)　→ Encoder 为 CRF

$L = -\log P(\hat{x}_{1:n}|x_{1:n})$

$= \sum_{y_{1:n}} P(y_{1:n}|x_{1:n}) P(\hat{x}_{1:n}|y_{1:n})$

（编码似然；解码似然）

$= \frac{1}{Z} \prod \exp(s(\ ,\ ))$

⇒ 可用前向算法计算 & 梯度下降优化

* Neural Method

△ 直接用 Static embeddings 给每个 Label ⇒ 2 problems

△ RNN → 上、下文结合上文

Bidirectional RNN → 上下文 ✓

△ Transformer ✓

△ Neural CRF：神经方法计算 CRF potentials

(如发射得分)

For all:

o Inference: { 不用 CRF: neural softmax: 通过量独立预测

用 CRF: Viterbi

o Learning: 为 CRF 和似 { 负对数条件似然

边际推断