法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

自动机

主讲人： 史兴

07/28/2017

提纲

- Seq2Seq 模型可视化
  - Word Embedding 可视化
  - Attention 可视化
- 自动机
  - (Weighted) Finite State Acceptor
  - (Weighted) Finite State Transducer
  - Chinglish 生成
  - Noisy Channel Model & HMM
  - EM 算法

Seq2Seq 模型可视化

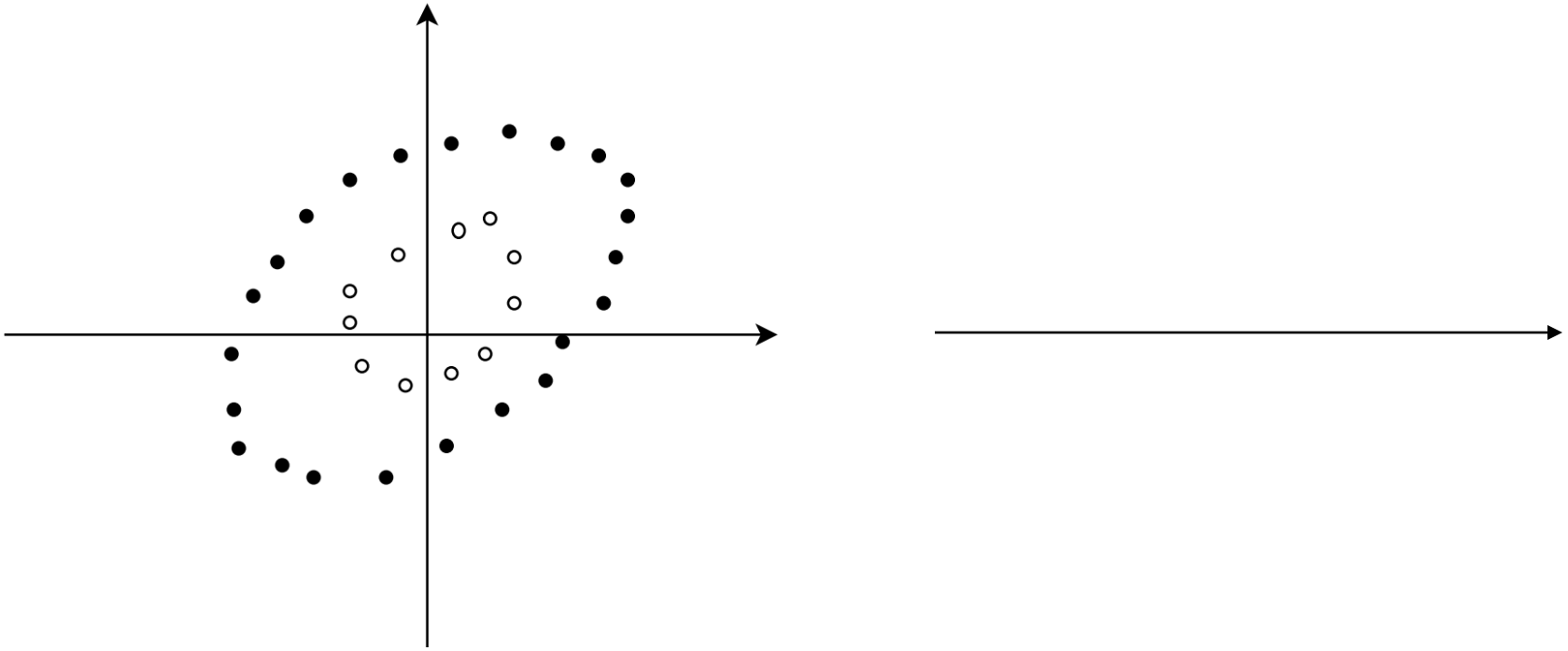
- Word Embedding 可视化
  - $x_1, \ldots, x_n; x_i \in R^d$
    - 人可以直接观察的维度：2 or 3
    - 降维；保留相似性
  - PCA: Principal component analysis
  - t-SNE: t-Distributed Stochastic Neighbor Embedding

Seq2Seq 模型可视化

- PCA: Principal Component Analysis
  - $x_1, \ldots, x_n;\ x_i \in R^d$
  - 将高维数据投射到2维空间中：
    - 第一维 (first component)
    - $t_i[1] = \mathrm{x_i} * \mathrm{w_1}\ ; \mathrm{w_1} \in R^d\ and\ |w_1| = 1$
    - 第二维 (second component)
    - $t_i[2] = \mathrm{x_i} * \mathrm{w_2}\ ; \mathrm{w_2} \in R^d\ and\ |w_2| = 1$
    - 如何求解$\mathrm{w_1}, w_2$？
      - 最大化 $t_i[1]$ 的方差 (如何可视化课程的学员？)
      - $argmax_{|w_1|=1} \sum (x_i w_1)^2$

Seq2Seq 模型可视化

- PCA: Principal Component Analysis
  - $x_1, \ldots, x_n;\ x_i \in R^2 \rightarrow t_1, \ldots, t_n; t_i \in R$

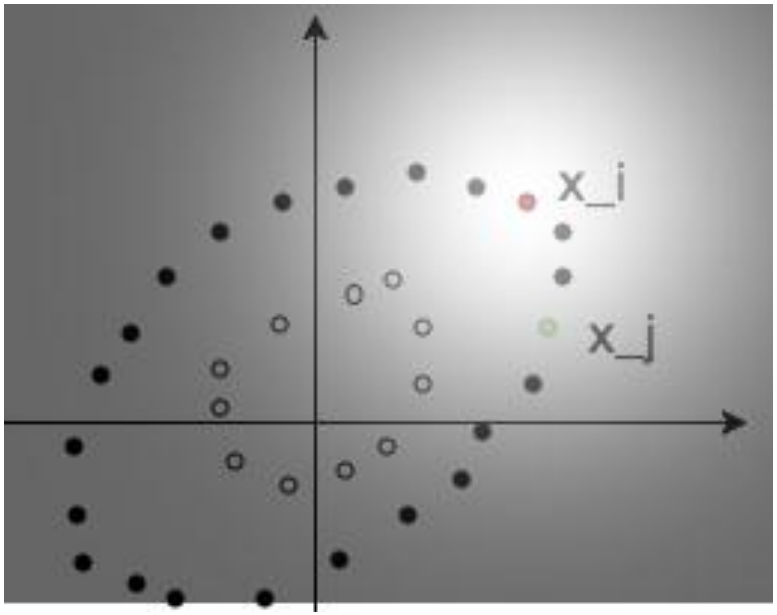


Seq2Seq 模型可视化

- PCA: Principal Component Analysis
  - $x_1, \ldots, x_n;\ x_i \in R^2 \rightarrow t_1, \ldots, t_n; t_i \in R$
  - 缺点：
    - 只在意“欧式”距离，并不在意“结构”(manifold)上的距离
  - 改进：t-SNE
    - 更在意“邻居性”：在高维中我们是邻居，在低维中还要做邻居！

Seq2Seq 模型可视化

- t-SNE: t-Distributed Stochastic Neighbor Embedding
  - 高维空间的欧式距离--> 条件概率：正比于高斯分布



$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)},$$

Seq2Seq 模型可视化

- t-SNE: t-Distributed Stochastic Neighbor Embedding
  - 高位空间，使得条件概率对称：
    - $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$
  - 低维空间的距离--> 条件概率：正比于student-t distribution
    - $y_1, \ldots, y_n; y_i \in R^{\hat{d}}, where\ \hat{d} \ll d$
    - 目标：寻找合适的 y_i

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}.$$

Seq2Seq 模型可视化

- t-SNE: t-Distributed Stochastic Neighbor Embedding
  - 目标函数：Kl散度，非对称

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

  - 大的p_{ij}（高维中的邻居），q_{ij}太小，很大的loss
  - 小的p_{ij}，q_{ij}无所谓，小的loss

Seq2Seq 模型可视化

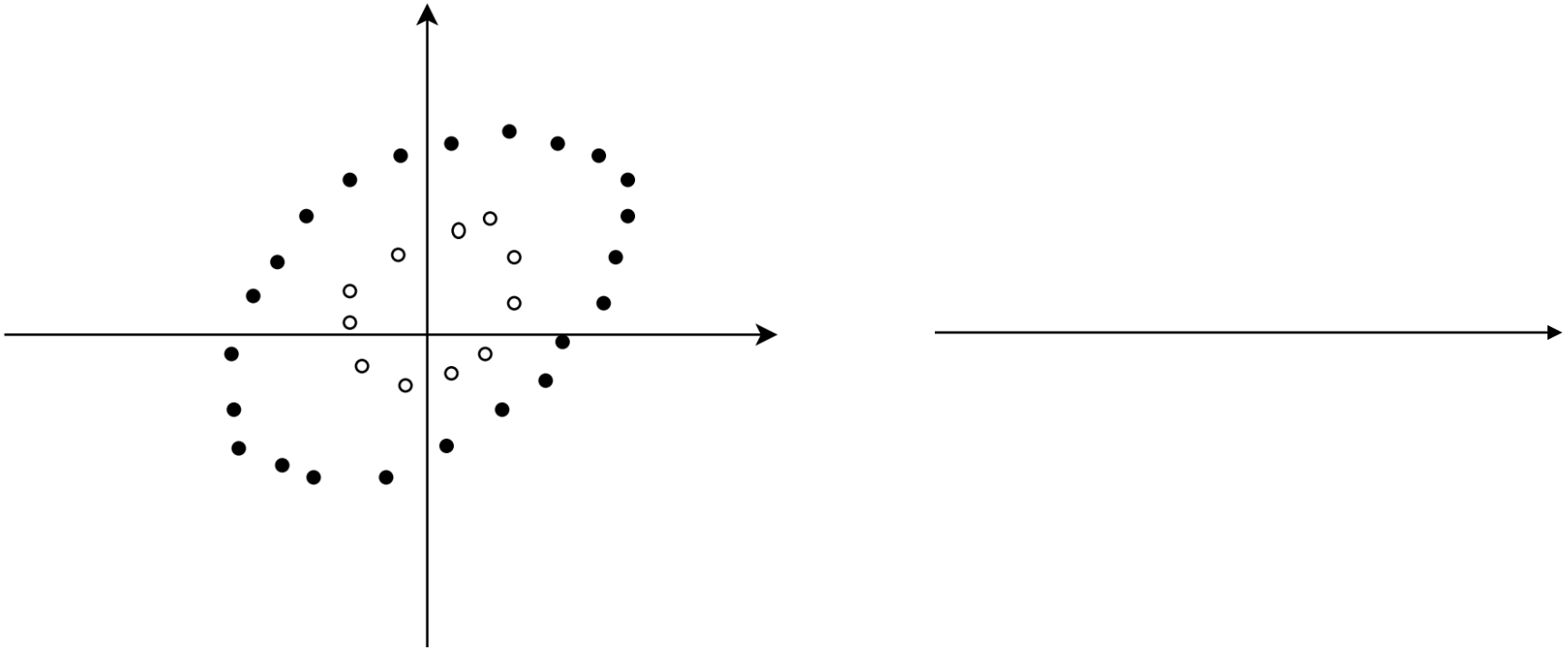
- t-SNE: t-Distributed Stochastic Neighbor Embedding
  - 目标函数：Kl散度，非对称

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

  - 大的p_{ij}（高维中的邻居），q_{ij}太小，很大的loss
  - 小的p_{ij}，q_{ij}无所谓，小的loss

Seq2Seq 模型可视化

- t-SNE: t-Distributed Stochastic Neighbor Embedding
  - $x_1, \ldots, x_n;\ x_i \in R^2 \rightarrow t_1, \ldots, t_n;\ t_i \in R$



Seq2Seq 模型可视化

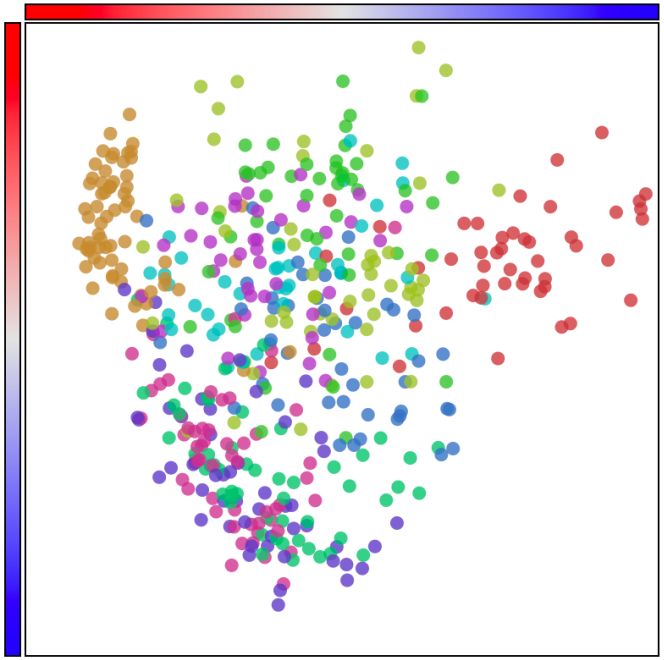
- MNIST 数据集
  - 每个图片 28*28 = 784 dimension

PCA　　t-SNE

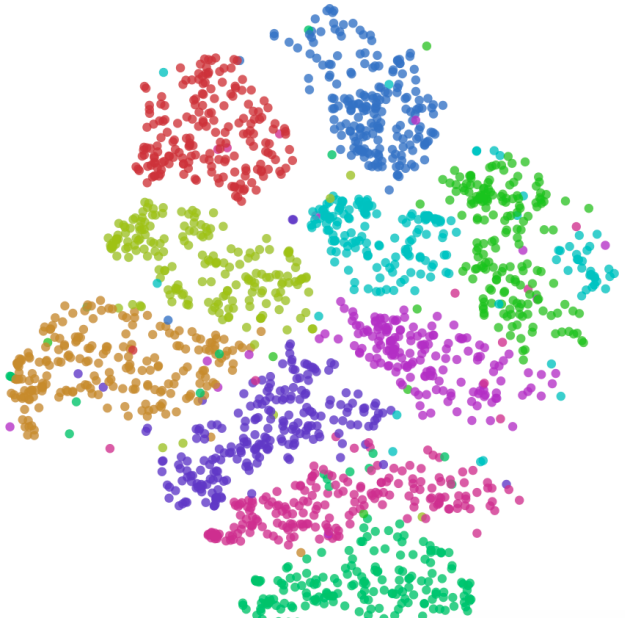
http://colah.github.io/posts/2014-10-Visualizing-MNIST/

Seq2Seq 模型可视化

- MNIST 数据集



PCA

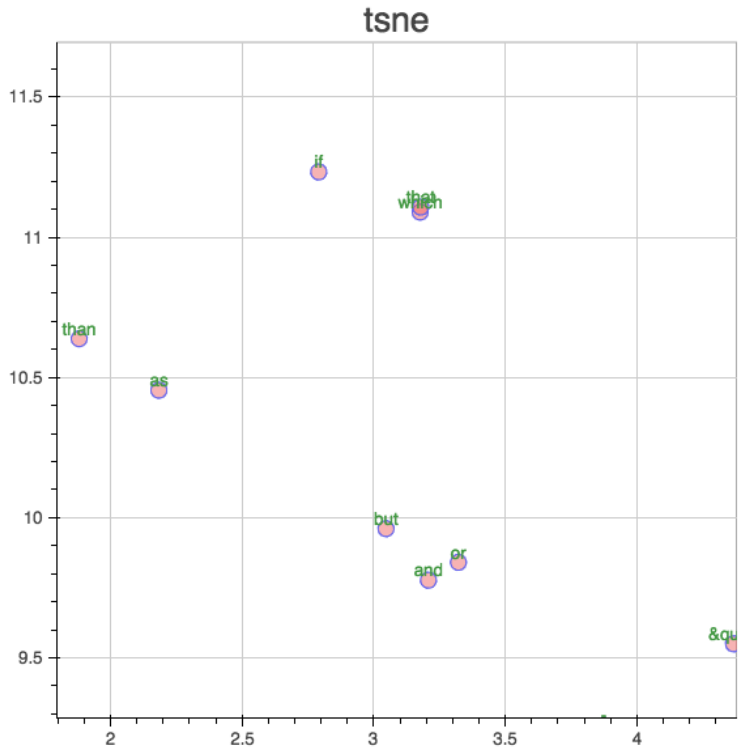


t-SNE

http://colah.github.io/posts/2014-10-Visualizing-MNIST/

Seq2Seq 模型可视化

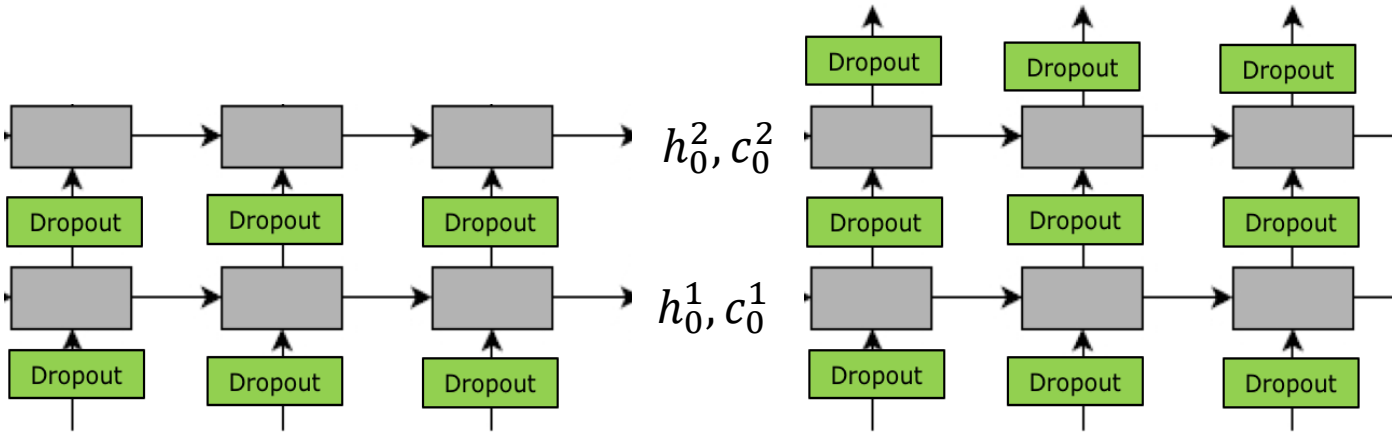
- Seq2Seq model



target input word embedding

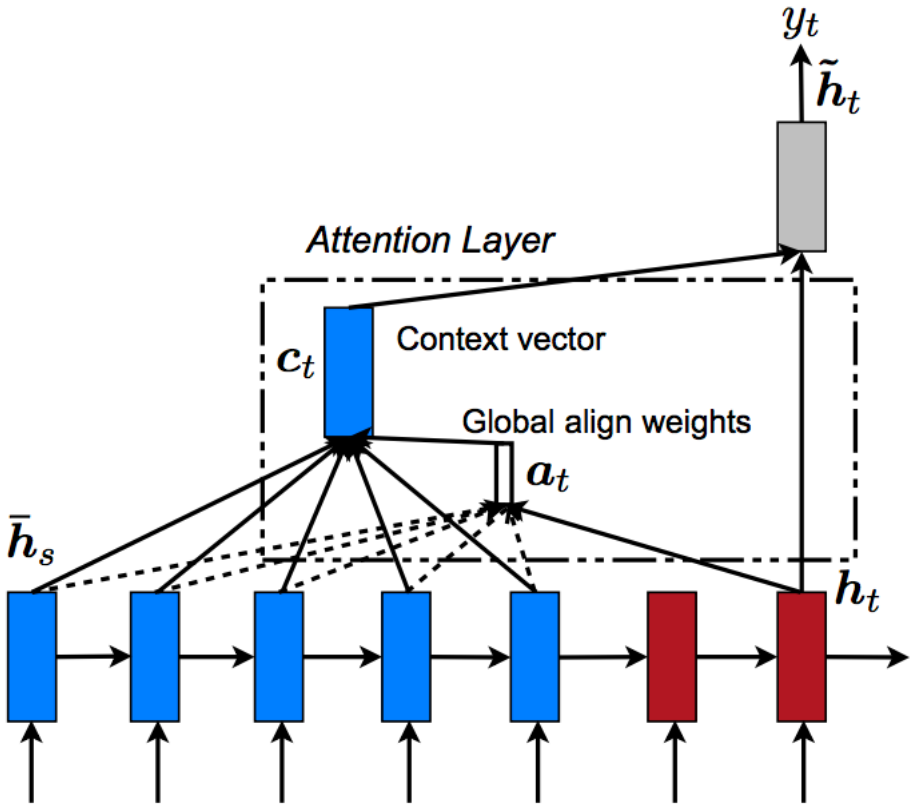
Seq2Seq 模型可视化

- Seq2Seq model
  - 作业：用t-SNE去可视化encoder的最后一步 h_0^1, c_0^1
    - 训练Eng2Fre英法翻译模型
    - 用1000个英语句子放入encoder，得到 h_0^1, c_0^1
    - 使用任意t-SNE工具进行可视化，将结果发到小象问答论坛
    - 会有“原来不过如此”的发现



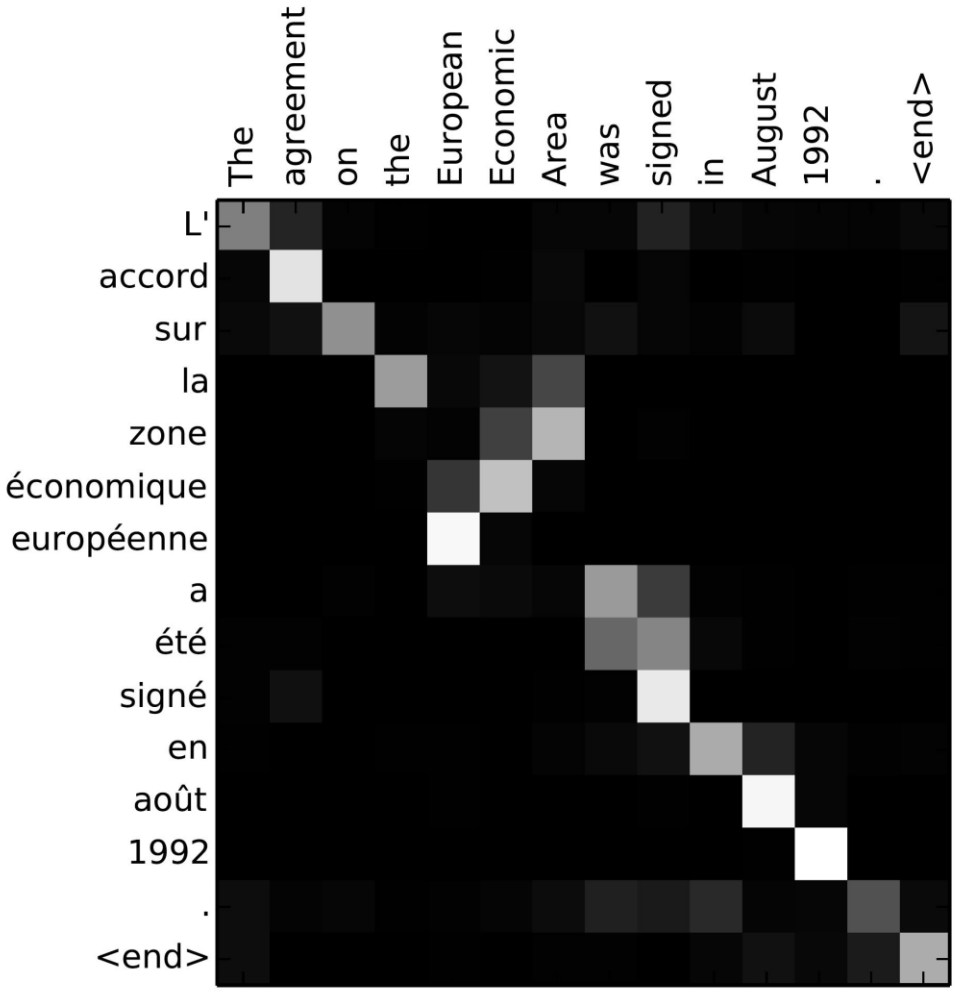
Seq2Seq 模型可视化

- Attention 可视化
  - 直接查看a_t



Seq2Seq 模型可视化

- Attention 可视化
  - 英法翻译

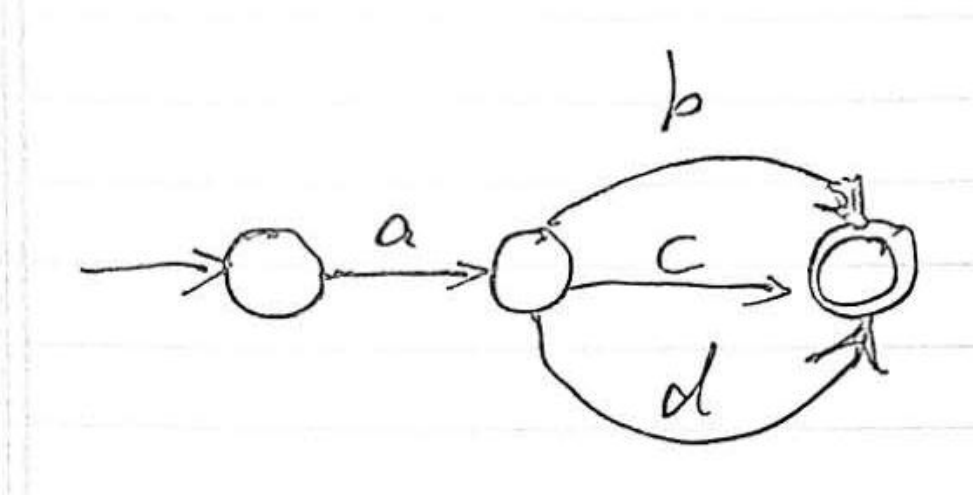


自动机

- Finite State Acceptor
  - $\sum$ 字典
  - S 有限个数的状态
  - s_0 初始状态
  - F 结束状态 (可以有多个)
  - $\delta: S \times \Sigma \rightarrow S$ 状态转移函数

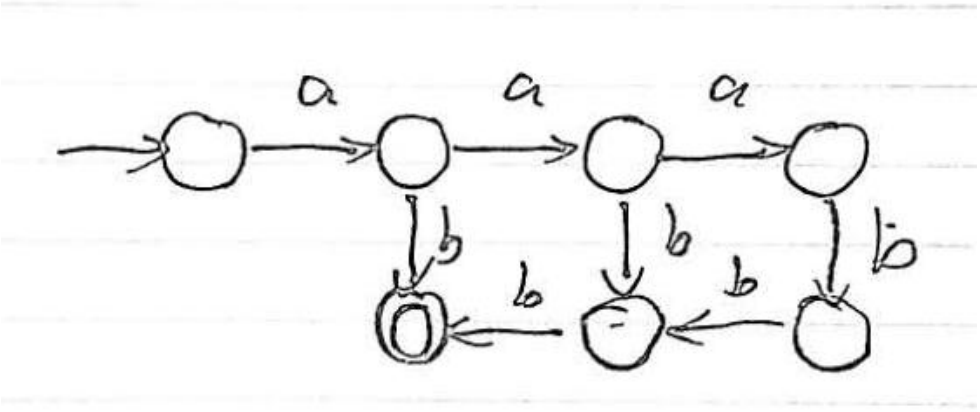
自动机

- Finite State Acceptor
  - 接受 {ab, ac, ad} 的 FSA
  - 文件: abcd.fsa
  - carmel -Ok 3 abcd.fsa



自动机

- Finite State Acceptor
  - 接受$a^n b^n (\mathrm{n} \leq 3)$的 FSA
  - 文件: anbn.fsa
  - carmel -Ok 3 anbn.fsa



  - 接受$a^n b^n$的FSA?

自动机

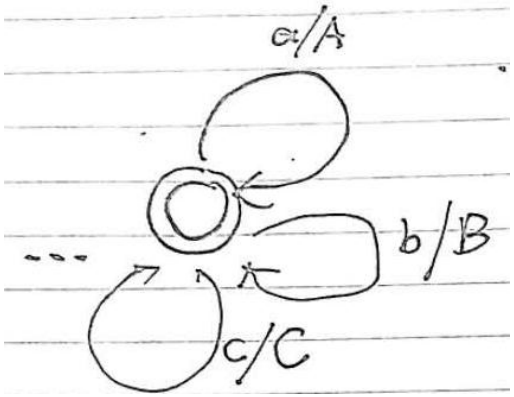
- Finite State Acceptor
  - 使用carmel将两个fsa取交集
    - carmel anbn.fsa abcd.fsa

自动机

- Finite State Transducer
  - $\sum$ 输入字典
  - Γ 输出字典
  - S 有限个数的状态
  - s_0 初始状态
  - F 结束状态 (可以有多个)
  - $\delta: S \times \Sigma \rightarrow S$ 状态转移函数
  - $\omega: S \times \Sigma \rightarrow \Gamma$ 输出函数

自动机

- Finite State Transducer
  - 将小写字母变成大写字母



- echo "l i k e" | carmel -sliOk 1 capitalize.fst
- echo “L I K E" | carmel -sriIk 1 capitalize.fst

自动机

- Finite State Transducer
  - 将abcd.fsa和capitialize.fst串行
    - carmel -kO 3 abcd.fsa capitalize.fst
  - 将anbn.fsa和capitialize.fst串行
    - carmel -kO 3 anbn.fsa capitalize.fst

自动机

- Weighted Finite State Transducer
  - Chinglish生成
    - 英文--> 音标
      - echo '"hello" # "word"' | carmel -sliEOk 1 eword-epron3.fst.wb
    - 音标-->拼音声母韵母
      - echo 'ae k s eh p t' | carmel -sliOEk 1 phrase_derivation.fst.wb
    - 拼音声母韵母→拼音
      - echo 'w o # sh i # sh ui'|carmel -sliEOk 1 pinyin-if-pinyin-q.fst.wb
    - 拼音→ 汉字
      - $echo '"de"' | carmel -sliOEk 5 pinyin-q-to-chinese.fst.wb

自动机

- Weighted Finite State Transducer
  - Chinglish生成
    - 全部串联起来
      - echo '"i" "s" "i" # "is" # "awesome" ' | carmel -sliOEk 1 eword-epron3.fst.wb phrase_derivation.fst.wb pinyin-if-pinyin-q.fst.wb pinyin-q-to-chinese.fst.wb
    - 微信公众号：“外语立刻说”

Noisy Channel Model

- Part-of-speech Tagging
  - I love you => PRON VERB PRON
  - 监督方法：
    - (句子，POS标注)都是已知的
    - MLE: max P(POS|sentence)
  - 非监督方法：
    - 只有句子已知，并不知道对应的POS标注
    - MLE: max $P(sentence) = \sum_{all\ possible\ POS} p(sentence|POS_i)P(POS_i)$

Noisy Channel Model

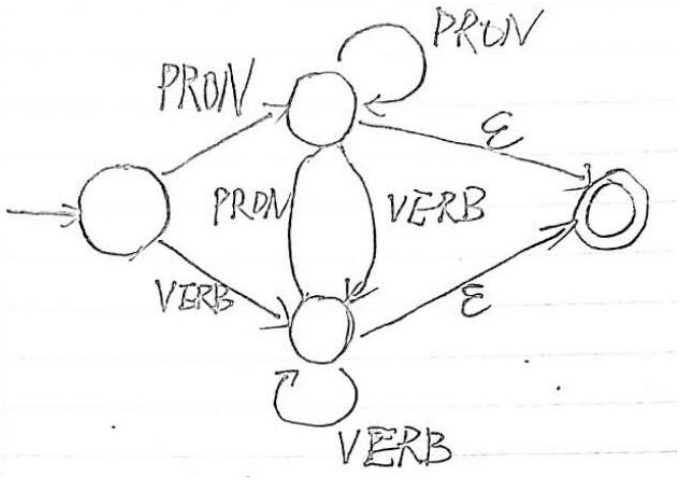
- Part-of-speech Tagging
  - I like you => PRON VERB PRON
  - 监督方法：
    - (句子，POS标注) 都是已知的
    - MLE: max P(POS|sentence)
  - 非监督方法：
    - 只有句子已知，并不知道对应的POS标注
    - max $P(sentence) = \sum_{all\ possible\ POS} p(sentence|POS_i)P(POS_i)$
    - EM 算法

Noisy Channel Model

- $P(sentence) = \sum_{all\ possible\ POS} p(sentence|POS_i)P(POS_i)$
- 生成故事(generative story)
  - Prior 先验
    - 根据$P(POS_i)$生成最可能的满足语法的词性序列
  - Likelihood 似然概率 / noisy channel
    - 将(POS_i)中的词性一一替换成单词

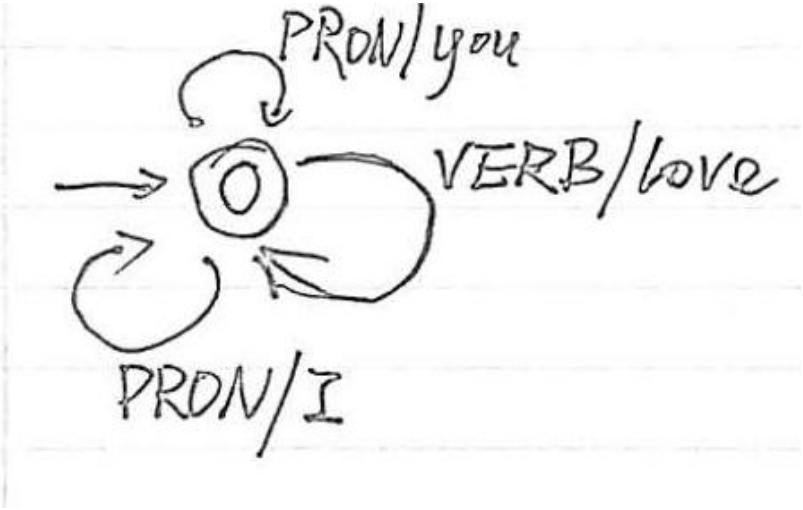
Noisy Channel Model

- 生成故事(generative story)
  - Prior 先验
    - 根据$P(POS_i)$生成最可能的满足语法的词性序列
    - 用bigram模型来表示先验 --> 用一个FSA来表示
    - tagging.fsa



Noisy Channel Model

- 生成故事(generative story)
  - Likelihood 似然概率 / noisy channel
    - 将(POS_i)中的词性一一替换成单词
    - tagging.fst

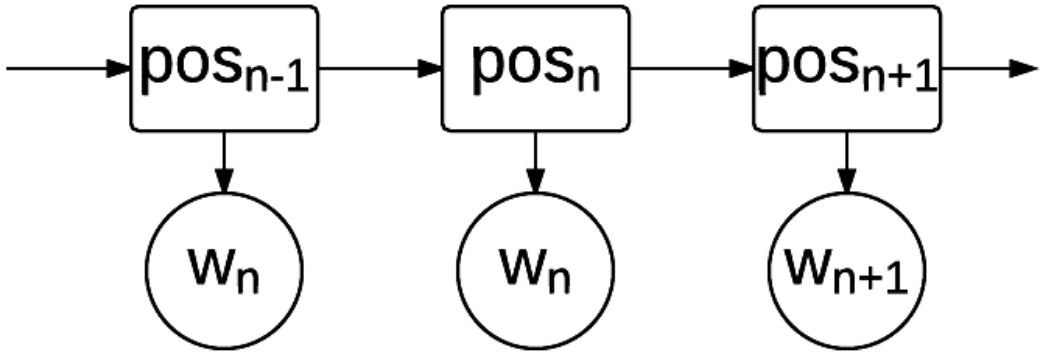


Noisy Channel Model

- 将tagging.fsa和tagging.fst串联起来，就构成了noisy channel model
- 如何估算tagging.fsa和tagging.fst中的参数
  - 利用训练数据 tagging.data (只包含有句子)
  - carmel 可以调用Em算法对其进行训练
  - carmel --train-cascade -HJ tagging.data tagging.fsa tagging.fst
- 使用Viterbi算法来得到最佳POS序列
  - echo '"I" "like" "you" "."' | $CARMEL -qbsriWIEk 1 tagging.fsa.trained.noe tagging.fst.trained

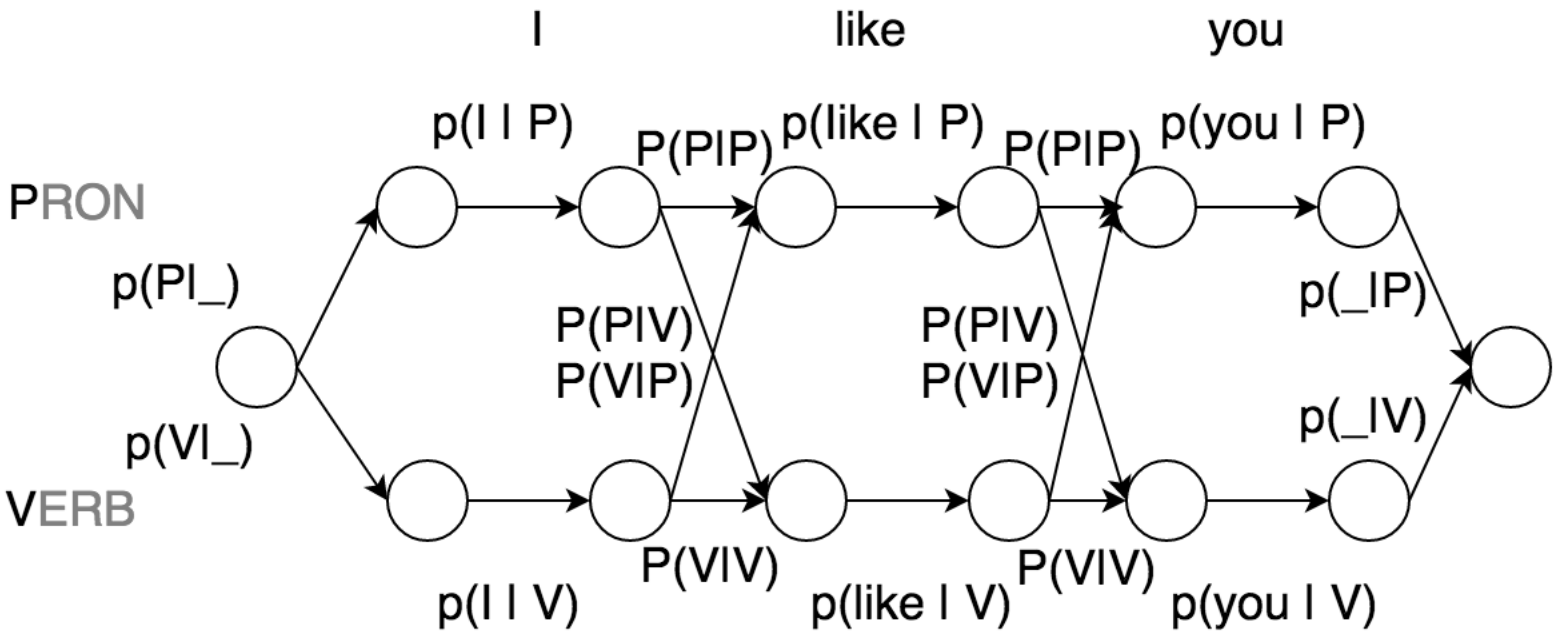
Noisy Channel Model

- Hidden Markov Model
  - 状态转移概率 == FSA
  - 输出概率 == FST
  - HMM == FSA + FST



Noisy Channel Model

- Noisy Channel Model和HMM
  - 统一用lattice来表示



联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop