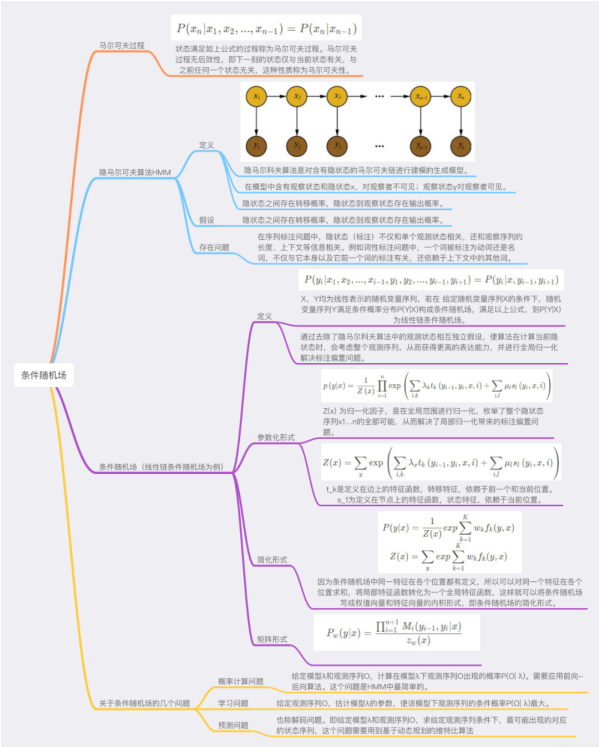
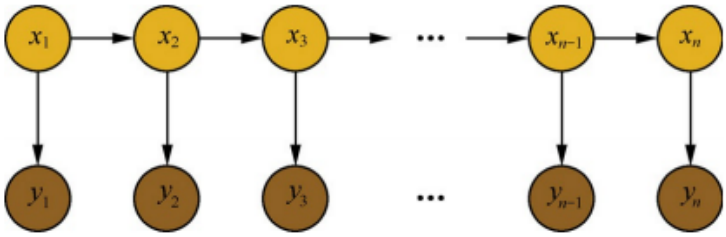
NLP学习算法1

- 概率图模型

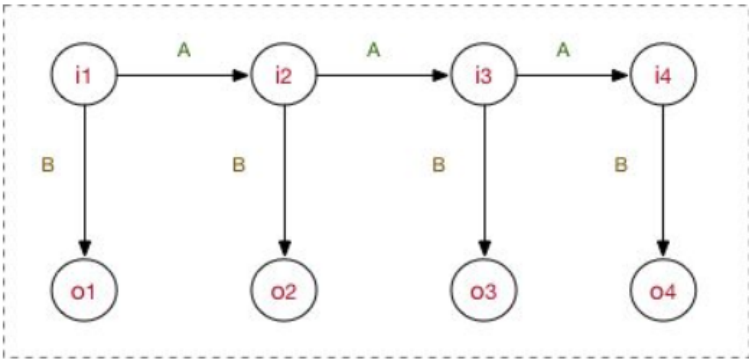


  - 什么是概率图模型?
    - 概率图模型（Probabilistic Graphical Model， PGM），简称图模型（Graphical Model，GM），是指一种用图结构来描述**多元随机变量之间条件独立性的概率模型（注意条件独立性）**，从而给研究高维空间的概率模型带来了很大的便捷性。
  - 什么是 随机场?
    - 每个位置按照某种分布随机赋予一个值 所构成 的 整体
- 马尔可夫过程 介绍篇
  - 什么是 马尔可夫过程?
    - 假设一个随机过程中，t_n 时刻的状态x_n的条件发布，只与其前一状态x_{n-1} 相关，即:
    - $P(x_n|x_1, x_2, ..., x_{n-1}) = P(x_n|x_{n-1})$
    - 则将其称为 马尔可夫过程。

- 马尔可夫过程 的核心思想 是什么?
  - 对于 马尔可夫过程 的 思想，用一句话去概括：**当前时刻状态 仅与上一时刻状态相关，与其他时刻不相关。**
  - 可以从 马尔可夫过程 图 去理解，由于 每个状态间 是以 有向直线连接，也就是 当前时刻状态 仅与上一时刻状态相关。
- 隐马尔科夫算法 篇
- 隐马尔科夫算法 是什么?
  - 隐马尔科夫算法是对含有未知参数（隐状态）的马尔可夫链进行建模的**生成模型**，如下图所示：



  - 在隐马尔科夫模型中，包含隐状态 和 观察状态，隐状态 i_i 对于观察者而言是不可见的，而观察状态 o_i 对于观察者而言是可见的。隐状态间存在转移概率，隐状态 i_i到对应的观察状态 o_i 间存在输出概率。
- 隐马尔科夫算法 中 两个序列 是什么?
  - 两序列
    - 隐藏序列：隐状态 i_i 对于观察者而言是不可见的
    - 观测序列：o_i 对于观察者而言是可见的



- 隐马尔科夫算法 中 三个矩阵 是什么?
  - 初始状态矩阵：每个标签的概率矩阵

- 发射状态矩阵：一个字变成每个标签的概率 $B = [b_{ij}]_{N \times M}$（$N$为隐藏状态集元素个数，$M$为观测集元素个数），其中$b_{ij} = P(o_t|i_t)$，（$o_t$为第i个观测节点，$i_t$为第i个隐状态节点,即所谓的观测概率（发射概率）；
- 状态转移矩阵：标签到每个标签的概率 $A = [a_{ij}]_{N \times N}$（N 表示隐藏状态集元素的个数），其中 $a_{ij} = P(i_{t+1}|i_t)$，i_t 即第i个隐状态节点，即所谓的状态转移；

- 隐马尔科夫算法 中 两个假设 是什么?
  - 齐次马尔可夫性假设：即假设隐藏的马尔科夫链在任意时刻 t 的状态只依赖于其前一时刻的状态，与其他时刻的状态及观测无关，也与时刻 t 无关
    $P(x_i|x_1, x_2, ..., x_{i-1}) = P(x_i|x_{i-1})$
  - 观测独立性假设：即假设任意时刻的观测只依赖于该时刻的马尔科夫链的状态，与其他观测及状态无关。
    $P(y_i|x_1, x_2, ..., x_{i-1}, y_1, y_2, ..., y_{i-1}, y_{i+1}, ...) = P(y_i|x_i)$

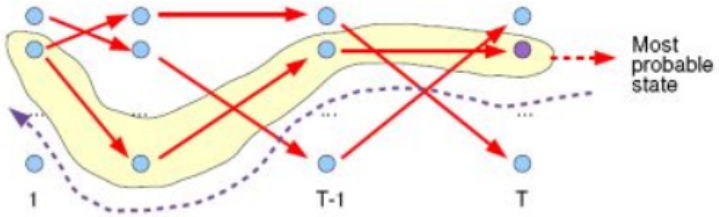
- 隐马尔科夫算法 中 工作流程 是什么?
  - 隐状态节点 i_t 是不能直接观测到的数据节点，o_t 才是能观测到的节点，并且注意箭头的指向表示了依赖生成条件关系；
  - i_t在A的指导下生成下一个隐状态节点i_{t+1}；
  - i_t在B的指导下生成依赖于该i_t的观测节点o_t；
  - 深层次理解：由于 为有向图，而且属于生成式模型，直接对联合概率分布建模
  - $P(O, I) = \sum_{t=1}^{T} P(O_t|O_{t-1}) P(I_t|O_t)$

- 隐马尔科夫算法 模型计算过程篇
  - 隐马尔科夫算法 序列概率计算过程 是什么样的?
    - 思想:
    - 如何对一条序列计算其整体的概率。即目标是计算出 $P(O|\lambda)$；

- 给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o1, o2, ..., oT)$，计算在模型λ下观测序列O出现的概率$P(O|\lambda)$
- 直接计算法（穷举搜索）
  - 由于有隐藏的状态序列 I的存在，我们是无法计算$P(O|\lambda)$的。一种常见的做法是把I边缘掉，即$P(O|\lambda) = \sum(P(O, I|\lambda))$，当然，这种计算复杂度非常高，为$O(TN^2)$
- 前向算法
  - 减少计算量的原因在于每一次计算直接引用前一个时刻的计算结果，避免重复计算，计算复杂度将为$O(T^2 * N)$，T是观测序列数量的复杂度，N是隐藏状态数量的复杂度。
- 后向算法

- 隐马尔科夫算法 学习训练过程 是什么样的?
  - 思想
    - 找出数据的分布情况，也就是模型参数的确定;
    - 已知观测序列 $O = (o1, o2, ..., oT)$，估计模型 $\lambda = (A, B, \pi)$参数，使得在该模型下观测序列概率 $P(O|\lambda)$ 最大，即用极大似然估计的方法估计参数
    - 常用方法
    - 极大似然估计：该算法在训练数据是**会 将 观测状态序列 O 和 隐状态序列 I**;
    - Baum-Welch(前向后向)：该算法在训练数据是**只会 将 观测状态序列 O**;
- 隐马尔科夫算法 序列标注（解码）过程 是什么样的?
  - 思想
    - 也就是“预测过程”，通常称为解码过程。在给定的观测序列下找出一条隐状态序列，条件是这个隐状态序列的概率是最大的那个
    - $Q_{\max} = \text{argmax}_{\text{allQ}} \frac{P(Q,O)}{P(O)}$
    - 常用方法：Viterbi算法

- Viterbi计算有向无环图的一条最大路径：



- HMM模型三个基本问题的联系？
  - 三个基本问题 存在 渐进关系。首先，要学会用前向算法和后向算法算观测序列出现的概率，然后用Baum-Welch算法求参数的时候，某些步骤是需要用到前向算法和后向算法的，计算得到参数后，我们就可以用来做预测了。因此可以看到，三个基本问题，它们是渐进的，对于做NLP的同学来说，应用HMM模型做解码任务应该是最终的目的。
- 隐马尔科夫算法 问题篇
  - 因为HMM模型其实它简化了很多问题，做了某些很强的假设，如**齐次马尔可夫性假设**和**观测独立性假设**，做了假设的好处是，**简化求解的难度**，坏处是**对真实情况的建模能力变弱**了。
  - 在序列标注问题中，隐状态（标注）不仅和单个观测状态相关，还和观察序列的长度、上下文等信息相关。例如词性标注问题中，一个词被标注为动词还是名词，不仅与它本身以及它前一个词的标注有关，还依赖于上下文中的其他词。
- 最大熵马尔科夫模型（MEMM）篇
  - 最大熵马尔科夫模型（MEMM）动机篇
    - HMM 存在 什么问题？
      - HMM中，观测节点 o_i 依赖隐藏状态节点 i_i ,也就意味着我的观测节点只依赖当前时刻的隐藏状态。但在更多的实际场景下，**观测序列是需要很多的特征来刻画的**，比如说，我在做NER时，我的标注 i_i 不仅跟当前状态 o_i 相关，而且还跟前后标注 $o_j(j \neq i)$ 相关，比如字母大小写、词性等等。
  - 最大熵马尔科夫模型（MEMM）介绍篇

- 最大熵马尔科夫模型（MEMM） 是什么样?
  - 通过 “定义特征” 的方式，学习条件概率：$P(I|O)=\prod_{t=1}^{n} P\left(i_i|i_{i-1},o_i\right),i=1,\cdots,n$
  - 并且， $P(i|i',o)$ 这个概率通过最大熵分类器建模（取名MEMM的原因）：

$$P(i|i^{'},o)=\frac{1}{Z(o,i^{'})}exp(\sum_{a})\lambda_a f_a(o,i)$$

局部归一

  - 重点来了，这是ME的内容，也是理解MEMM的关键：$Z(o,i')$ 这部分是归一化； $f_a(o,i)$ 是特征函数，具体点，这个函数是需要去定义的; λ 是特征函数的权重，这是个未知参数，需要从训练阶段学习而得。
  - 定义特征函数：

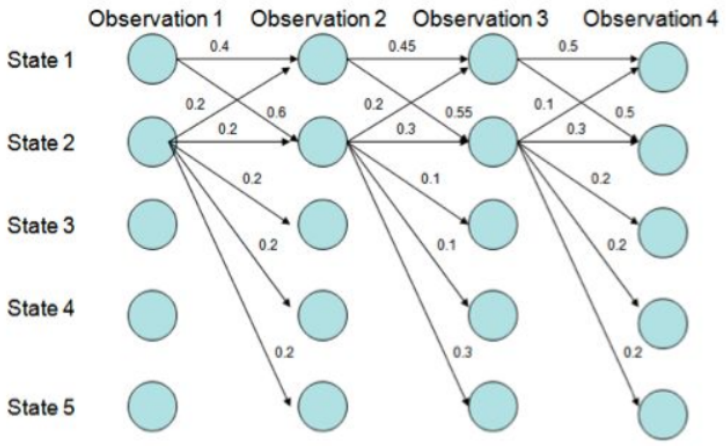
$$f_a(o,i)=\begin{cases}1 & 满足特定条件,\\ 0 & \text{other}\end{cases}$$

  - 其中，特征函数 $f_a(o,i)$ 的 个数 可以任意制定，（$a=1,...,n$）所以总体上，MEMM的建模公式这样：

$$P(I|O)=\prod_{t=1}^{n}\frac{exp(\sum_a)\lambda_a f_a(o,i)}{Z(o,i_{i-1})},i=1,\cdots,n$$

  - 请务必注意，理解判别模型和定义特征两部分含义，这已经涉及到CRF的雏形了。
- 最大熵马尔科夫模型（MEMM） 如何解决 HMM 问题?
  - 在前面介绍 HMM 时，HMM 提出了 **观测节点 o_i 依赖隐藏状态节点** i_i 假设，该假设不合理的，针对该问题，MEMM 提出 **观测节点 i_i 依赖隐藏状态节点 o_i 以及上一时刻的隐藏节点i_{i-1}** 假设。 （HMM 和 MEMM 箭头）；
- 最大熵马尔科夫模型（MEMM）问题篇
  - 问题简述: MEMM 容易出现标注偏置问题，MEMM倾向于选择拥有更少转移的状态。

- 问题介绍



- 用Viterbi算法解码MEMM，状态1倾向于转换到状态2，同时状态2倾向于保留在状态2。 解码过程细节（需要会viterbi算法这个前提）：

P(1-> 1-> 1-> 1)= 0.4 x 0.45 x 0.5 = 0.09 ,
P(2->2->2->2)= 0.2 X 0.3 X 0.3 = 0.018,
P(1->2->1->2)= 0.6 X 0.2 X 0.5 = 0.06,
P(1->1->2->2)= 0.4 X 0.55 X 0.3 = 0.066

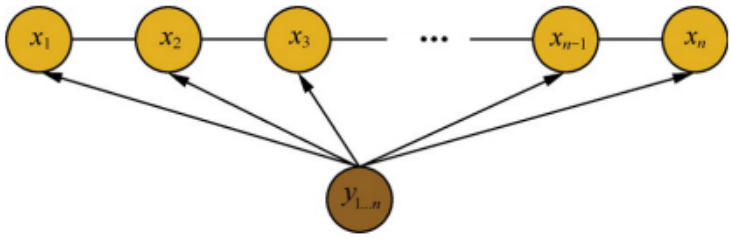
- 但是得到的最优的状态转换路径是1->1->1->1，为什么呢？因为状态2可以转换的状态比状态1要多，从而使转移概率降低,即MEMM倾向于选择拥有更少转移的状态。

P(1-> 1-> 1-> 1)= 0.4 x 0.45 x 0.5 = 0.09 ,
P(2->2->2->2)= 0.2 X 0.3 X 0.3 = 0.018,
P(1->2->1->2)= 0.6 X 0.2 X 0.5 = 0.06,
P(1->1->2->2)= 0.4 X 0.55 X 0.3 = 0.066

- 求和的作用在概率中是归一化，但是这里归一化放在了指数内部，管这叫local归一化。 来了，viterbi求解过程，是用dp的状态转移公式（MEMM的没展开，请参考CRF下面的公式），因为是局部归一化，所以MEMM的viterbi的转移公式的第二部分出现了问题，导致dp无法正确的递归到全局的最优。

- 条件随机场（CRF）篇
  - CRF 动机篇

- HMM 和 MEMM 存在什么问题?
  - HMM： 状态的转移过程中当前状态只与前一状态相关问题
  - MEMM： 标注偏置 问题
    - 解决方法：统计全局概率，在做归一化时考虑数据在全局的分布
- CRF 介绍篇
  - 什么是 CRF?
    - 设 X 与 Y 是随机变量，**P(Y|X) 是给定条件 X 的条件下 Y 的条件概率分布，若随机变量 Y 构成一个由无向图G=(V,E)表示的马尔科夫随机场。**则称 **条件概率分布P(X|Y)为条件随机场。**



  - CRF 的 主要思想是什么?
    - 统计全局概率，在做归一化时，考虑了数据在全局的分布。
    - CRF 的定义是什么?
      - 给定 $X=(x_1,x_2,...,x_n)$，$Y=(y_1,y_2,...,y_n)$ 均为线性链表示的随机变量序列，若在给随机变量序列 X 的条件下，随机变量序列 Y 的条件概率分布 $P(Y|X)$ 构成条件随机场，即满足马尔可夫性：
        $P(y_i|x_1,x_2,...,x_{i-1},y_1,y_2,...,y_{i-1},y_{i+1})=$
        $P(y_i|x,y_{i-1},y_{i+1})$则称为 P(Y|X) 为线性链条件随机场。
      - 通过去除了隐马尔科夫算法中的观测状态相互独立假设，使算法在计算当前隐状态x_i时，会考虑整个观测序列，从而获得更高的表达能力，并进行全局归一化解决标注偏置问题。
  - CRF 的 三个基本问题 是什么?
    - 概率计算问题
      - 定义：给定 观测序列 x 和 状态序列 y， 计算概率 P(y|x)
      - 公式定义：$p(y|x)=$
        $\frac{1}{Z(x)}\prod_{i=1}^{n}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1},y_i,x,i)+\sum_{i,l}\mu_l s_l(y_i,x,i)\right)$

- $Z(x)$ 为归一化因子，是在全局范围进行归一化，枚举了整个隐状态序列$x_{1...n}$的全部可能，从而解决了局部归一化带来的标注偏置问题。
- $Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda_x t_k\left(y_{i-1}, y_i, x, i\right) + \sum_{i,l} \mu_l s_l\left(y_i, x, i\right)\right)$
- t_k 为定义在边上的特征函数，转移特征，依赖于前一个和当前位置
- s_1 为定义在节点上的特征函数，状态特征，依赖于当前位置。
- 解决方法：前向计算、后向计算

- 学习计算问题
  - 定义：给定训练数据集估计条件随机场模型参数的问题，即条件随机场的学习问题。
  - 公式定义：利用极大似然的方法来定义我们的目标函数
  - 解决方法：随机梯度法、牛顿法、拟牛顿法、迭代尺度法这些优化方法来求解得到参数
  - 目标：解耦 模型定义，目标函数，优化方法
- 预测问题
  - 定义：给定条件随机场 P(Y|X) 和输入序列（观测序列） x ，求条件概率最大的输出序列（标记序列） y* ，即对观测序列进行标注。
  - 方法：维特比算法

- CRF 的 流程是什么?
  - 选择特征模板：抽取文本中的字符组合 or 具有其他特殊意义的标记组成特征，作为当前 token 在模板中的表示;
  - 构建特征函数：通过一组函数来完成由特征向数值转换的过程，使特征与权重对应;
  - 进行前向计算：每个状态特征函数（0-1二值特征函数）对应 L 维向量，最终状态特征函数权值的和即为该位置上激活了的状态特征函数对应的 L 维向量之和;

- 解码：利用 维特比算法 解码 出 最优标注序列

- CRF 优缺点篇
  - CRF 的 优点在哪里?
    - 为每个位置进行标注过程中可利用丰富的内部及上下文特征信息;
    - CRF模型在结合多种特征方面的存在优势;
    - 避免了标记偏置问题;
    - CRF的性能更好，对特征的融合能力更强;
  - CRF 的 缺点在哪里?
    - 训练模型的时间比ME更长，且获得的模型非常大。在一般的PC机上可能无法执行;
    - 特征的选择和优化是影响结果的关键因素。特征选择问题的好与坏，直接决定了系统性能的高低

- 对比篇
  - CRF模型 和 HMM 和 MEMM 模型 区别?
    - 相同点：MEMM、HMM、CRF 都常用于 序列标注任务;
    - 不同点:
      - 与 HMM 的区别：CRF 能够解决 HMM 因其输出独立性假设，导致其不能考虑上下文的特征，限制了特征的选择的问题;
      - 与 MEMM 的区别：MEMM 虽然能够解决 HMM 的问题，但是 MEMM 由于在每一节点都要进行归一化，所以只能找到局部的最优值，同时也带来了标记偏见的问题，即凡是训练语料中未出现的情况全都忽略掉。
      - CRF：很好的解决了这一问题，他并不在每一个节点进行归一化，而是所有特征进行全局归一化，因此可以求得全局的最优值。
  - 为什么 CRF模型 会比 HMM 被普遍使用?
    - 原因 1：CRF模型 属于 判别式模型，在 序列标注 任务上，效果优于 生成式模型;

- 原因 2：HMM 提出 齐次马尔可夫性假设 和 观测独立性假设，这两个假设过强，而 CRF 只需要满足 局部马尔可夫性就好，通过降低假设的方式，提升模型效果；

- DNN-CRF

- 命名实体识别 评价指标 是什么？
  - 命名实体识别 本质有两个任务组成：
    - 边界检测
    - 类型识别
  - 精确匹配评估

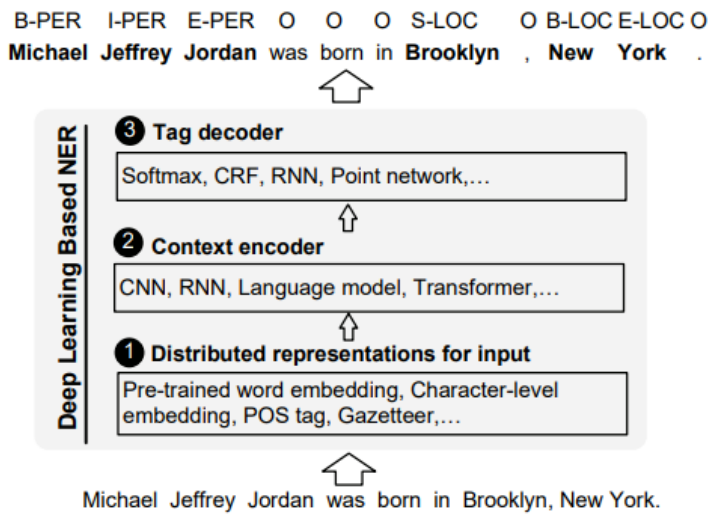
$$Precision = \frac{\#TP}{\#(TP + FP)}$$

$$Recall = \frac{\#TP}{\#(TP + FN)}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

  - 注：其中 #TP 表示识别出的实体且是正确实体的数量，#FP 表示识别出的实体却是不存在实体的数量，#FN 表示是正确的实体却没被识别出的数量。当然，除此之外，像 macro-f1、micro-f1也会用来作评估。
  - 宽松匹配评估
    - 特点：降低了匹配正确的要求
    - 类别：
      - 类别正确：一个实体被正确识别到类别，而边界与正确边界有重叠即可；
      - 边界正确：一个实体被正确识别到边界，而不管类别是什么；
- 传统的命名实体识别方法
  - 基于规则的命名实体识别方法是什么？
    - 介绍：基于特定领域或者特定语法规则来设计规则，然后利用这些规则去抽取 句子中实体
    - 缺点：依赖专家知识 构建 规则

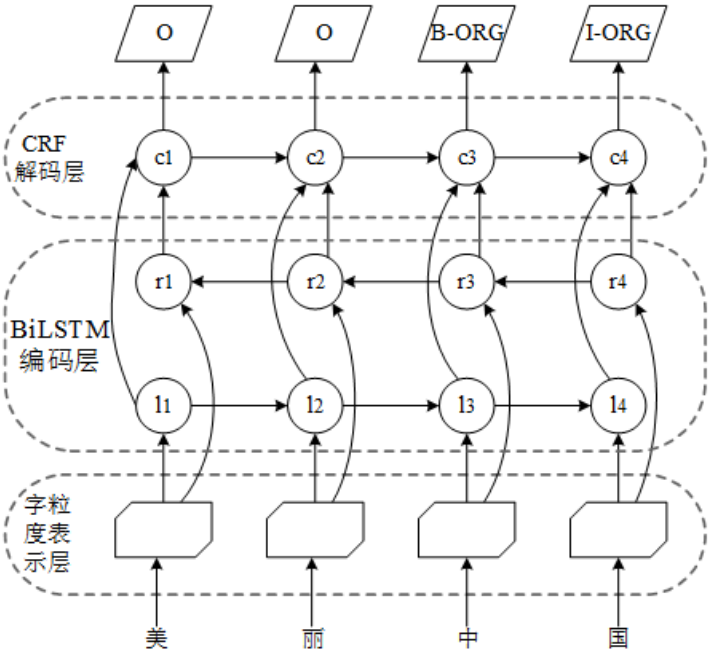
- 特点：由于特定领域的规则和不完整的词典，这类系统往往具有较高的精度和较低的召回率，无法转移到其他领域
- 基于无监督学习的命名实体识别方法是什么？
  - 主要思想：词汇、词语模式以及在大语料上计算的统计特性可以用来推断命名实体的出现
- 基于特征的监督学习的命名实体识别方法是什么？
  - 介绍：NER被转化为一个多分类问题或者序列标注问题，通过精心设计的特征，在标注语料上进行训练，从而在未知文本上识别出类似的模式实体。
- 基于深度学习的命名实体识别方法
  - 基于深度学习的命名实体识别方法 相比于 基于机器学习的命名实体识别方法的优点？
    - 受益于深度学习的非线性转换，可以从数据中学到更复杂的特征；
    - 避免大量的人工特征的构建；
    - 可以设计为端到端的结构。
  - 基于深度学习的命名实体识别方法 的 结构是怎么样？
    - 介绍：基于深度学习的命名实体识别方法 由 分布式输入层、文本编码器、解码层 组成；



  - 分布式输入层 是什么，有哪些方法？
    - 介绍：将 句子 转化为 词粒度、字符粒度、混合的表示方法作为模型输入；
    - 类别：

- 词粒度的表示：像word2vec、fasttext、glove或者SENNA等算法可以通过大规模的无监督语料学习词粒度的表示，作为NER模型的输入，在训练过程中既可以固定不变也可以微调。
- 字符粒度的表示：除了仅考虑词向量作为输入之外，也融合了字符粒度的表示作为模型的输入。字符级别的表示，相对于词向量，能够提取出子词的特征，比如前缀后缀等，也能够缓解oov（out of vocabulary）的问题。如下图所示，最常用的两种提取结构分别基于CNN与RNN模型。
- 混合表示：除了词粒度和字符粒度的表示之外，还融合了一些额外信息，比如地点信息、词汇相似度、视觉信息等等。像FLAT、BERT、XLNET等模型也被本文划分为混合表示，因为输入的时候还包含有位置等信息。

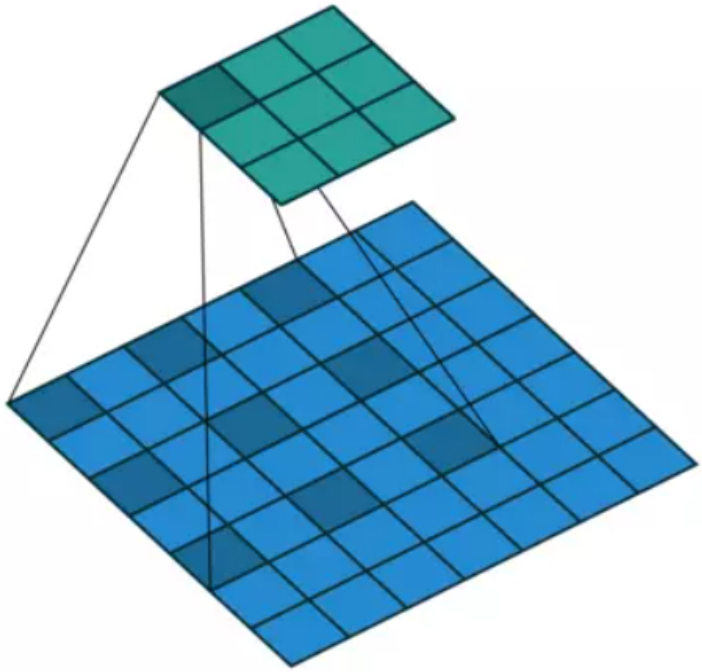
- 文本编码器篇
  - **什么是 BiLSTM-CRF?**



    - step 1：利用 字粒度表示层 将 句子中每个字 编码成一个字向量；
    - step 2：然后利用 BiLSTM 编码层 对句子中每个字的字向量，以捕获 句子中每个字的信息；
    - step 3：在 BiLSTM 后接一个 CRF 解码层，对 编码的向量进行解码，以解码出 句子最优的标注序列；
  - 为什么要用 BiLSTM?

- 由于 BiLSTM 能够捕获句子的长距离依赖信息。

- **什么是 Dilated CNN?**
  - Dilated/Atrous Convolution(中文叫做空洞卷积或者膨胀卷积) 或者是 Convolution with holes 从字面上就很好理解，是在标准的 convolution map 里注入空洞，以此来增加 reception field。相比原来的正常convolution，dilated convolution 多了一个 hyper-parameter 称之为 dilation rate 指的是kernel的间隔数量(e.g. 正常的 convolution 是 dilatation rate 1)。



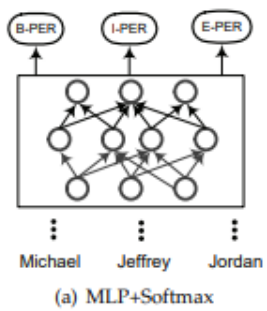
- 为什么会有 Dilated CNN?
  - CNN 特点:方便的获取局部特征，通过pad操作，可以让CNN的输出层与输入层具有相同的序列长度，从而可以使用CNN来进行序列相关的特征抽取;
  - CNN 问题:
    - 正常CNN 的 filler 作用于输入矩阵连续位置，利用 卷积 和 池化 整合多尺度的上下文信息，导致分辨率损失;
    - pooling 会损失信息，降低精度，不加则导致感受野变小，学不到全局信息;
    - 浅层的CNN只能获取局部特征，要获取全局特征或者更大的感受野，则需要比较深的CNN层，这会增加计算量也会提高过拟合的风险
- Dilated CNN 的优点?

- 优点：利用空洞卷积能够使输入宽度随模型深度呈指数增长，达到在不影响每层网络的分辨率的情况下，降低参数数量。
- 与传统CNN层相比，空洞卷积同样对序列上下文的滑动窗口进行操作;

- **IDCNN-CRF 介绍**
  - 动机：然后 Dilated CNN 相比于 CNN 能够捕获更多信息，但是扩张的窗口捕获的信息只能学习距离为d的其他信息。
  - 操作：通过堆叠不同扩张大小的空洞卷积层，有效扩展输入宽度的大小，使其在仅使用几个空洞卷积层的情况下，学习序列的整个长度信息。

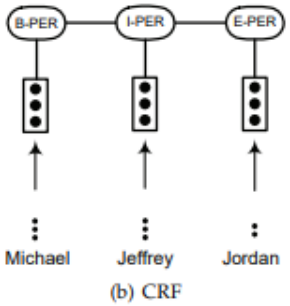
  - 注：利用四个结构相同的 Dilated CNN 拼接起来，每个 block 里面 dilated width 为 1，1，2 的三层 DCNN

- 标签解码器篇
  - 标签解码器是什么?
    - 标签解码器是NER模型的最后一个阶段。它接受上下文相关的表示作为输入，并基于输入产生相关的序列标签。
  - MLP+softmax层 介绍?
    - 介绍：使用多层感知器+Softmax层作为标签解码器层，则将序列标注任务转化为一个多分类问题。每个单词的标签都是根据上下文相关的表示独立预测的，而不考虑它的邻居标签结果。



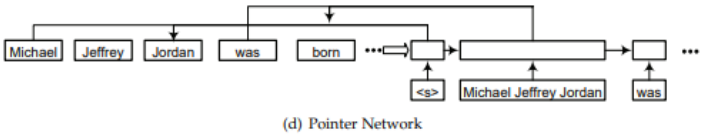
(a) MLP+Softmax

- 条件随机场CRF层 介绍?
  - 介绍：使一个以观测序列为全局条件的随机场。CRF已被广泛应用于基于特征的监督学习方法。许多基于深度学习的NER模型使用CRF层作为标签解码器，并取得了很好的精度。



(b) CRF

- 循环神经网络RNN层 介绍?
  - 介绍：RNN解码是个贪婪的过程，先计算出第一个位置的标签，然后后面每一个位置的标签都是基于前面的状态计算出标签，在标签数量比较多的时候，可以比CRF更加快速。
- 指针网路层 介绍?
  - 介绍：将序列标注问题转化为两个子问题：先分块再分类。指针网络会贪婪地从头开始找下一个块结束的位置（开始的位置很显然，第一个块的开始位置是起始点，后面的开始位置都是前面一块的结束位置的后继位置）如上图d所示，在起始块"<\s>"后一块的结束块位置是"Jordan"，这样就得到块"Michael Jeffrey Jordan"，然后将这个块进行分类确定类别，之后再继续找下一个块的结束位置，找到"was"，这样就得到一个新的块"was"，再将这个块进行分类，然后这样循环下去直到序列结束。指针网络主要就起到确定块起始位置的作用。



(d) Pointer Network

- CNN-CRF vs BiLSTM-CRF vs IDCNN-CRF?
  - CNN-CRF：对于序列标注来讲，普通CNN有一个不足，就是卷积之后，末层神经元可能只是得到了原始输入数据中一小块的信息。而对NER来讲，整个输入句子中每个字都有可能对当前位置

的标注产生影响，即所谓的长距离依赖问题。为了覆盖到全部的输入信息就需要加入更多的卷积层，导致层数越来越深，参数越来越多。而为了防止过拟合又要加入更多的Dropout之类的正则化，带来更多的超参数，整个模型变得庞大且难以训练。

- BiLSTM-CRF：因为CNN这样的劣势，对于大部分序列标注问题人们还是选择biLSTM之类的网络结构，尽可能利用网络的记忆力记住全句的信息来对当前字做标注。但这又带来另外一个问题，biLSTM本质是一个序列模型，在对GPU并行计算的利用上不如CNN那么强大。
- IDCNN-CRF:
  - 如何解决 CNN-CRF 问题：通过堆叠不同扩张大小的空洞卷积层，有效扩展输入宽度的大小，使其在仅使用几个空洞卷积层的情况下，学习序列的整个长度信息；
  - 如何解决 BiLSTM-CRF 问题：因为 IDCNN 本身是一个 CNN ，所以可以像 CNN 一样进行并行化计算；

- 为什么 DNN（深度神经网络） 后面要加 CRF?
  - DNN 做 序列标注 的 机制
    - 做法：NN 对每个 token 打标签的过程是独立进行的，不能直接利用上文已预测的标签（只能靠隐含状态传递上下文信息），进而导致预测出的标签序列可能无效
    - 存在问题：
      - B、S、I、E 顺序错乱问题
  - CRF 做 序列标注 的 机制
    - 方法：CRF是全局范围内统计归一化的条件状态转移概率矩阵，再预测出一条指定的sample的每个token的label
    - 介绍：因为CRF的特征函数的存在就是为了对given序列观察学习各种特征（n-gram，窗口），这些特征就是在限定窗口size下的各种词之间的关系。然后一般都会学到这样的一条规律（特征）：B后面接E，不会出现B。这个限定特征会使得CRF的预测结果不出现上述例子的错误。
  - DNN-CRF
    - 做法：把CRF接到LSTM后面，把LSTM在timestep上把每一个hiddenstate的tensor输入给CRF，进行句子级别的标签预测，

使得标注过程中不再是对各个 token 单独分类

- 中文领域NER
  - 中文命名实体识别 与 英文命名实体识别的区别?
    - 和英文 NER 每个单词都使用空格分隔不同，中文 NER 是基于字的表示方法，所以一般会涉及到中文分词和中文NER技术，导致 中文 NER 技术容易受到中文分词的误差的影响。
    - 那么常用的方法有哪些呢?
      - 词汇增强：在**早期**的中文NER技术中，基于字符的 NER 方法往往具有高于 基于词汇（分词后）的方法，为了提高 基于词汇方法的效果，一般会采取引入词汇信息（词汇增强）的方法；
      - 词汇/实体类型信息增强：使用特殊标记来识别句子中单词的边界，修改后的句子将由BERT直接编码
  - 词汇增强篇
    - 什么是 词汇增强?
      - 词汇增强：引入词汇信息（词汇增强）来增强 模型 识别 句子中实体的方法
    - 为什么说 「词汇增强」 方法对于中文 NER 任务有效呢?
      - 虽然**基于字符的NER系统通常好于基于词汇（经过分词）的方法**，但**基于字符的NER没有利用词汇信息**，而**词汇边界对于实体边界通常起着至关重要的作用。**
      - **如何在基于字符的NER系统中引入词汇信息**，是近年来NER的一个研究重点。本文将这种引入词汇的方法称之为「词汇增强」，以表达引入词汇信息可以增强NER性能。
      - 从另一个角度看，**由于NER标注数据资源的稀缺，BERT等预训练语言模型在一些NER任务上表现不佳。特别是在一些中文NER任务上，词汇增强的方法会好于或逼近BERT的性能。**因此，关注「词汇增强」方法在中文NER任务很有必要。
    - 词汇增强 方法有哪些?
      - Dynamic Architecture：设计一个动态框架，能够兼容词汇输入；

- Adaptive Embedding：基于词汇信息，构建自适应Embedding；

- 什么是 Dynamic Architecture?
  - Dynamic Architecture 就是 需要设计相应结构以融入词汇信息。
  - 常用方法有哪些?
    - Lattice LSTM 、FLAT。。。
  - **什么是 Lattice LSTM，存在什么问题?**
    - 介绍：Lattice LSTM 通过词汇信息（词典）匹配一个句子时，可以获得一个类似Lattice（晶体、格子）的结构。融合了词汇信息到原生的LSTM

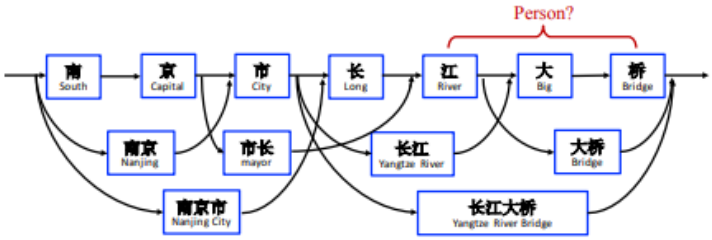


Figure 1: Word character lattice.

- Lattice是一个有向无环图，词汇的开始和结束字符决定了其位置。Lattice LSTM结构则融合了词汇信息到原生的LSTM中：

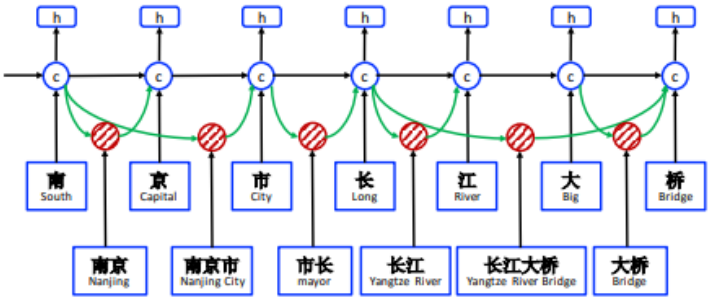
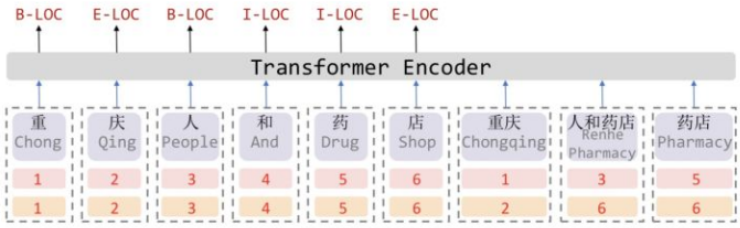


Figure 2: Lattice LSTM structure.

- 如上图所示，Lattice LSTM引入了一个word cell结构，对于当前的字符，融合以该字符结束的所有word信息，如对于「桥」融合了「长江大桥」和「大桥」的信息。对于每一个字符，Lattice LSTM采取注意力机制去融合个数可变的word cell单元，其主要的数学形式化表达为：

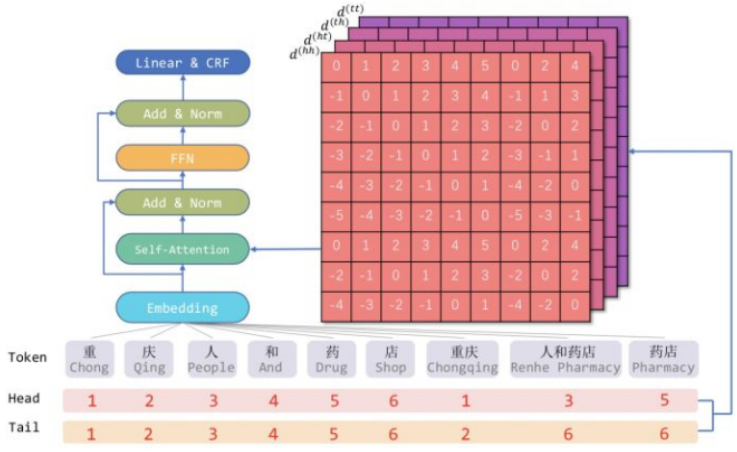
$$\mathbf{c}_j^c = \sum_{b \in \{b' | w_{b',j}^d \in \mathbb{D}\}} \boldsymbol{\alpha}_{b,j}^c \odot \mathbf{c}_{b,j}^w + \boldsymbol{\alpha}_j^c \odot \widetilde{\mathbf{c}}_j^c$$

- 存在问题：
  - Lattice LSTM 的提出，将词汇信息引入，有效提升了NER性能；但其也存在一些缺点：
  - 计算性能低下，不能batch并行化。究其原因主要是每个字符之间的增加word cell（看作节点）数目不一致；
  - 信息损失：
    - 1）每个字符只能获取以它为结尾的词汇信息，对于其之前的词汇信息也没有持续记忆。如对于「大」，并无法获得'inside'的「长江大桥」信息。
    - 2）由于RNN特性，采取BiLSTM时其前向和后向的词汇信息不能共享。
  - 可迁移性差：只适配于LSTM，不具备向其他网络迁移的特性。
- **什么是 FLAT，存在什么问题?**
  - 动机一：Lattice-LSTM 和 LR-CNN 问题
    - 这些模型采取的RNN和CNN结构无法捕捉长距离依赖；
    - 动态的Lattice结构也不能充分进行GPU并行；
  - 动机二：CGN 和 LGN 问题
    - 采取的图网络虽然可以捕捉对于NER任务至关重要的顺序结构，但这两者之间的gap是不可忽略的；
    - 这类图网络通常需要RNN作为底层编码器来捕捉顺序性，通常需要复杂的模型结构
  - 思路：
    - Transformer：采取全连接的自注意力机制可以很好捕捉长距离依赖，由于自注意力机制对位置是无偏的，因此Transformer引入位置向量来保持位置信息。
    - FLAT 亮点：对于每一个字符和词汇都构建两个head position encoding 和 tail position encoding。例如，字符[药]可以匹配词汇[人和药店]和[药店]。



Flat-Lattice Transformer

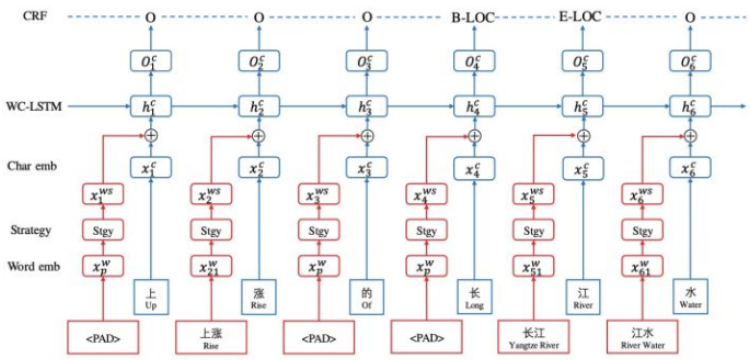
- 因此，我们可以将Lattice结构展平，将其从一个有向无环图展平为一个平面的Flat-Lattice Transformer结构，由多个span构成：每个字符的head和tail是相同的，每个词汇的head和tail是skipped的。



FLAT结构

- 什么是 Adaptive Embedding 范式?
  - Adaptive Embedding 范式 就是 在embedding层对于词汇信息进行**自适应**，后面通常接入LSTM+CRF和其他通用网络，这种范式与模型无关，具备可迁移性。
  - 常用方法有哪些?
    - WC-LSTM、 Multi-digraph、 Simple-Lexicon...
  - **什么是 WC-LSTM ，存在什么问题?**
    - 动机：**Lattice LSTM中每个字符只能获取以它为结尾的词汇数量是动态的、不固定**的，这也是导致Lattice LSTM**不能batch并行化**的原因。
    - 思路:
      - WC-LSTM为改进这一问题，采取**Words Encoding Strategy**，将每个字符为结尾的词汇信息进行固定编码表示，即每一个字符引入的词汇表征是静态的、固定的，如果没有对应的词汇则用代替，从而可以进行batch并行化。

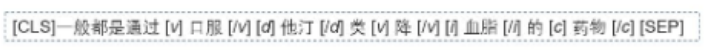
- 这四种 encoding 策略分别为：最短词汇信息、最长词汇信息、average、self-attenion。以「average」为例：即将当前字符引入所有相关的词汇信息对应的词向量进行平均。



WC-LSTM

- 存在问题：WC-LSTM仍然**存在信息损失问题**，无法获得'inside'的词汇信息，不能充分利用词汇信息。虽然是Adaptive Embedding范式，但WC-LSTM仍然采取LSTM进行编码，建模能力有限、存在效率问题。

- 词汇/实体类型信息增强篇
  - 什么是 词汇/实体类型信息增强?
    - 词汇/实体类型信息增强：使用特殊标记来识别句子中单词的边界，修改后的句子将由BERT直接编码
  - 为什么说「词汇/实体类型信息增强」方法对于中文 NER 任务有效呢?
    - 词汇增强方法 在NER任务中的表现令人印象深刻，但最近已经证明，添加词汇信息可以显著提高下游性能。然而，**没有任何工作在不引入额外结构的情况下将单词信息纳入BERT。**
  - 词汇/实体类型信息增强 方法有哪些?
    - LEX-BERT
  - 什么是 LEX-BERT ?
    - **LEX-BERT V1**
    - 方法：Lex BERT的第一个版本通过在单词的左右两侧插入特殊标记来识别句子中单词的 span（跨度、范围）。特殊标记不仅可以标记单词的起始位置和结束位置，还可以为句子提供实体类型信息

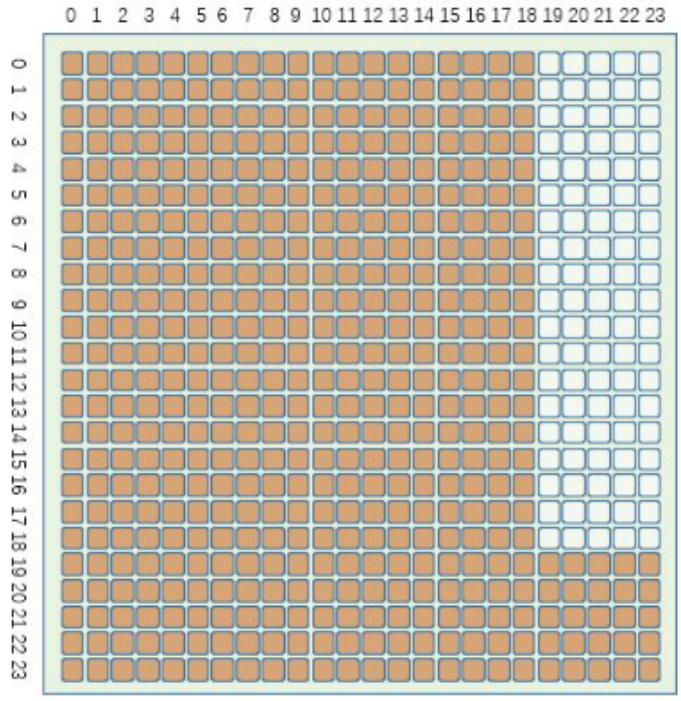


(a) Lex-BERT V1

- 作用：
  - 首先，如果我们的字典带有实体类型信息，我们可以通过标记将其插入到句子中。这里是医学相关动词的缩写，表示药物，表示检查索引，表示医学概念术语。
  - 其次，它们表示词汇集合中单词的开始或结束。这里开始标记的格式是[x]，结束标记的格式是[/x]，其中x可以是v、d、i等
- **LEX-BERT V2**
- 方法：对于在句子中加宽的单词，我们没有在句子中单词的周围插入起始和结束标记，而是在句子的末尾附加一个标记[x]。请注意，我们将标记的位置嵌入与单词的起始标记绑定：

$$P(w_{start}) = P([x]),$$

- 修改了BERT的注意矩阵，如图2所示。句子中的文字标记只会互相注意，而不会注意标记。相反，markertoken可以处理输入序列中的所有标记。



以上内容整理于 幕布文档