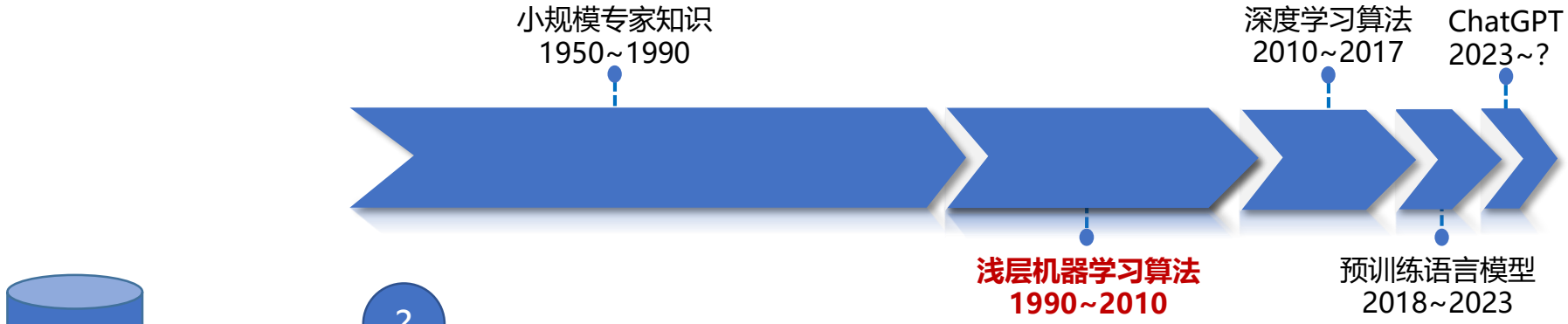
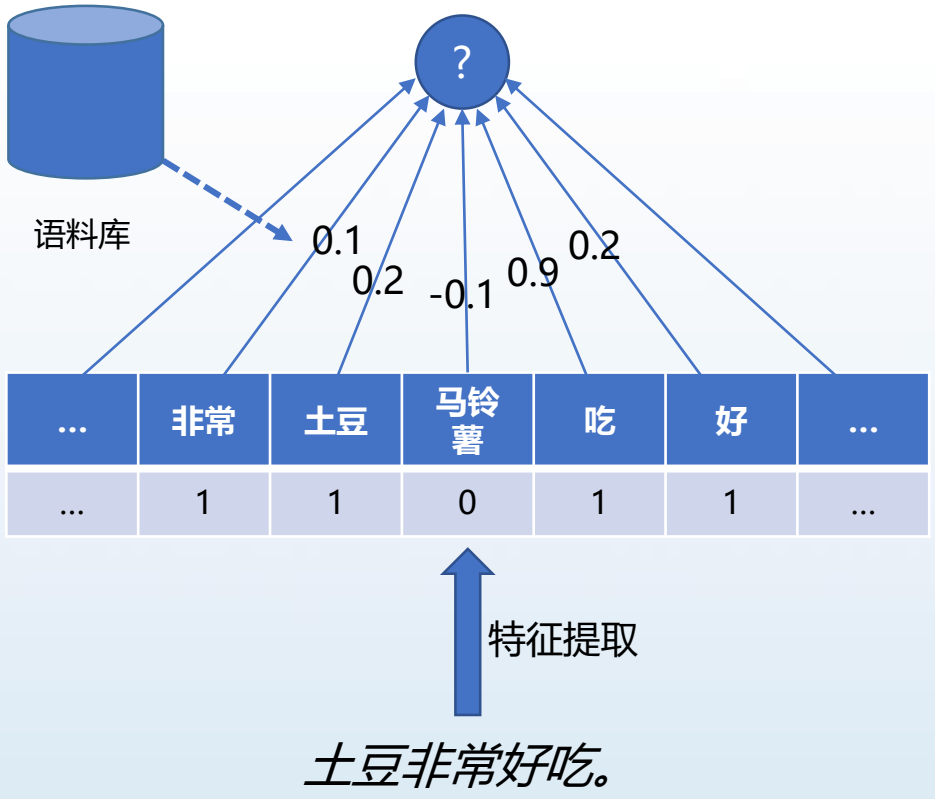
预训练词向量

杨沐昀

语言技术研究中心
哈尔滨工业大学

基于向量表示的浅层机器学习

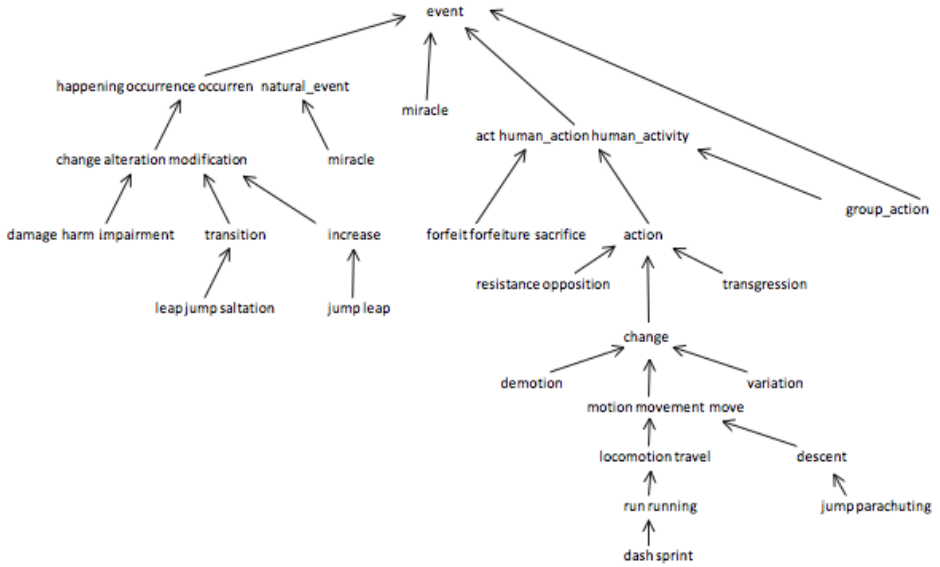


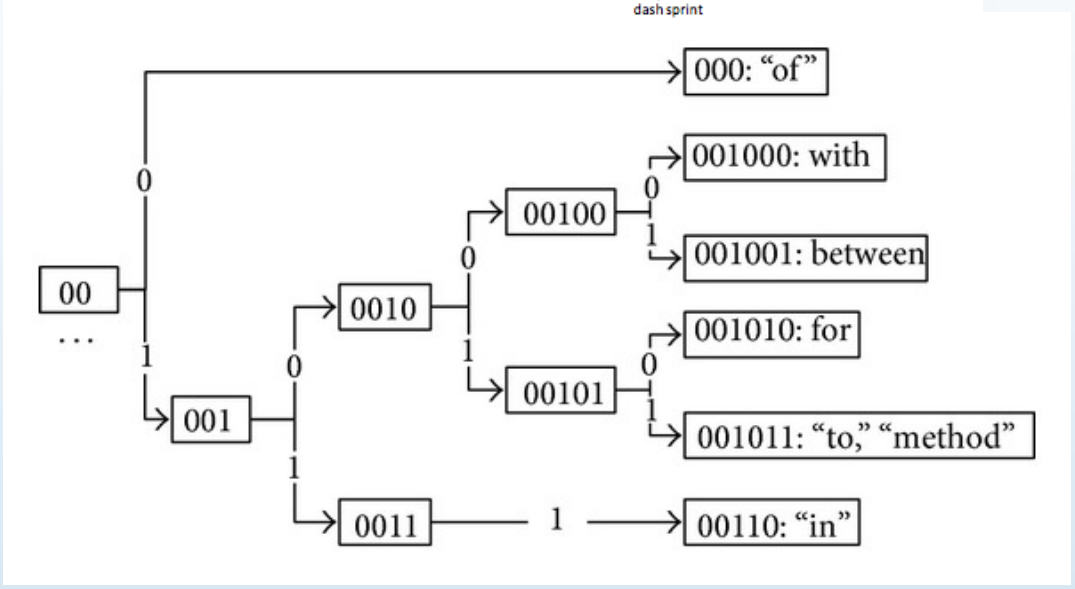


- 使用**高维、离散、稀疏**的向量表示词
  - 维度为词表大小，其中只有一位为1，其余为0
    - **土豆**：[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...]
    - **马铃薯**：[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...]
- 缺点
  - 无法处理“**多词一义**”的现象

传统解决方案

- 增加额外的特征
  - 词性特征：名词、动词、形容词
  - 前后缀特征：re-、-tion、-er
- 语义词典
  - WordNet、HowNet等
  - 如词的上位信息表示语义类别
  - 需要解决一词多义问题
  - 收录的词不全且更新慢
- 词聚类特征
  - 如Brown Clustering (Brown et al., CL 1992)
  - 潜在语义分析





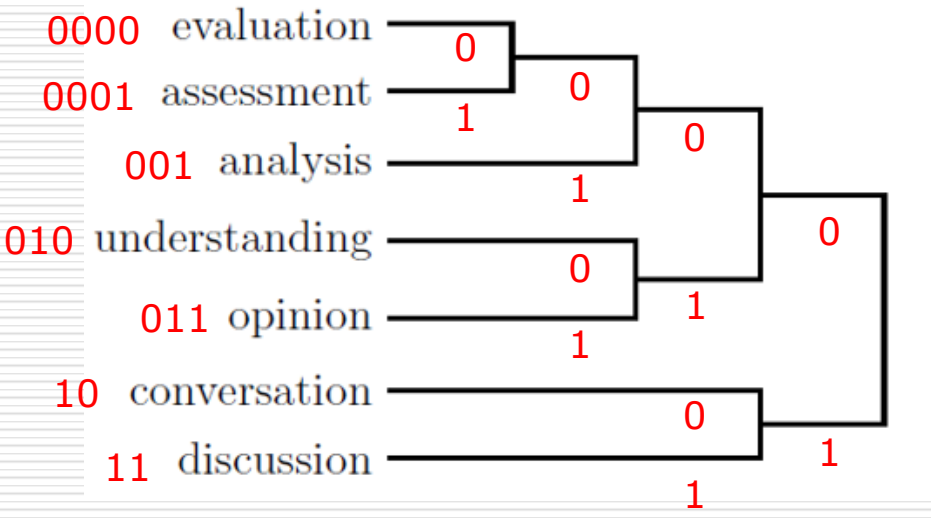
词的分布语义假设

- 分布语义假设（Distributional semantic hypothesis）
  - 词的含义可由其上下文词的分布进行表示
    - *You shall know a word by the company it keeps* -- Firth J.R. 1957

he curtains open and the moon shining in on the barely
ars and the cold , close moon " . And neither of the w
rough the night with the moon shining so brightly , it
made in the light of the moon . It all boils down , wr
surely under a crescent moon , thrilled by ice-white
sun , the seasons of the moon ? Home , alone , Jay pla
m is dazzling snow , the moon has risen full and cold
un and the temple of the moon , driving out of the hug
in the dark and now the moon rises , full and amber a
bird on the shape of the moon over the trees in front
But I could n't see the moon or the stars , only the
rning , with a sliver of moon hanging among the stars
they love the sun , the moon and the stars . None of
the light of an enormous moon . The plash of flowing w
man 's first step on the moon ; various exhibits , aer
the inevitable piece of moon rock . Housing The Airsh
oud obscured part of the moon . The Allied guns behind

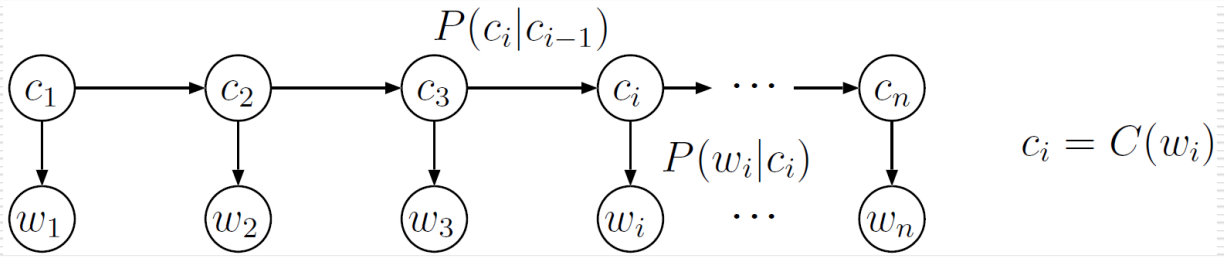
布朗聚类

- 利用上下文分布特征对词进行*层次化聚类*
- 可以用近似霍夫曼树的方式对单词进行编码
  - 前缀相似度越高的词越接近
  - 可以使用不同长度的前缀获得不同粒度的语义表示



布朗聚类获得方式

- 词类作为隐变量的二元语法模型：
  - 记文本为$w_1 w_2 \ldots w_n$，每个词所属词类$c_i = C(w_i)$
  - 每个词*只属于一个*词类



HMM，可由贝叶斯公式推导得到

- 文本概率表示为：

$$p(w_1 w_2 \ldots w_n) = \prod_{i=1}^{n} p(w|C(w_i))p(C(w_i)|C(w_{i-1}))$$

布朗聚类获得方式

- 初始状态下，每个词单独属于一个词类
- 每次选择令以下似然函数最大化的两个词类进行合并：

$$f(w,C)=\frac{1}{n}\log p(w_1 w_2 \ldots w_n)$$

$$=\frac{1}{n}\sum logp(w_i|C(w_i))p(C(w_i)|C(w_{i-1}))$$

$$=\sum_{c,c'} p(c,c')\log\frac{p(c,c')}{p(c)p(c')}-\sum_{w} p(w)\log p(w)$$

$$=I(C)-H(w)$$

相邻词类互信息　　文本信息熵

布朗聚类

- 运算效率
  - $I(C)$为相邻词类之间的互信息。每次评估似然函数只需要对$I(C)$进行计算，减少了运算复杂度
  - 整体时间复杂度为$O(|V|^5)$
  - Brown 和 Liang分别提出$O(|V|^3)$和$O(|V|m^2+n)$的算法优化
- 优点
  - 相比于独热表示，提升了语义关联性表示能力
- 缺点
  - 只利用了二元文法的前后一个词的信息
  - 离散表示限制了对细粒度语义相关性的表示

潜在语义分析 LSA

- 基于矩阵分解获得词分布表示
  - 记词为w，上下文为c，二者共现矩阵为M
  - 上下文c的选择：
    - 固定窗口的词：反映词法、句法等局部属性
    - 所在文档：反映词代表的主题信息

	c_1	c_2	c_3	...	$c_{\|C\|}$	$count(w)$
w_1	2	8	12	...	125	570
w_2	0	0	60	...	9	168
w_3	1004	987	19	...	0	3089
...	...	...	...	...	...	...
$w_{\|V\|}$	0	19	2039	...	2	5760
$count(c)$	1997	2010	4323	...	239	213985

需要降低高频词的权重

潜在语义分析 LSA

- 对矩阵M进行点互信息(PMI)变换。对于词w和上下文c：

$$PMI(w,c) = \log_2 \frac{p(w,c)}{p(w)p(c)}$$

- $p(w,c), p(w), p(c)$用极大似然法进行估计：
  - $p(w,c) = \frac{count(w,c)}{count(all)}$　　$count(w,c)$表示 w,c 共现次数
  - $p(w) = \frac{count(w)}{count(all)}$
  - $p(c) = \frac{count(c)}{count(all)}$

潜在语义分析 LSA

- 为了防止共现次数较低的词和上下文计算出现负的PMI，采用PPMI(Positive PMI)进行变换：

$$PPMI(w,c) = \max(PMI(w,c), 0)$$

	c_1	c_2	c_3	...	$c_{|C|}$
w_1	0.00	0.40	0.04	...	5.28
w_2	0.00	0.00	2.87	...	3.87
w_3	3.55	3.53	0.00	...	0.00
...	...	...	...	...	...
$w_{|V|}$	0.00	0.00	2.86	...	0.00

PPMI变换后的共现矩阵M

潜在语义分析 LSA

- 对于处理过的共现矩阵M，利用*截断奇异值分解*(Truncated Singular Value Decomposition)获取词表示：

$$M \approx U\sum \mathrm{V}^T$$

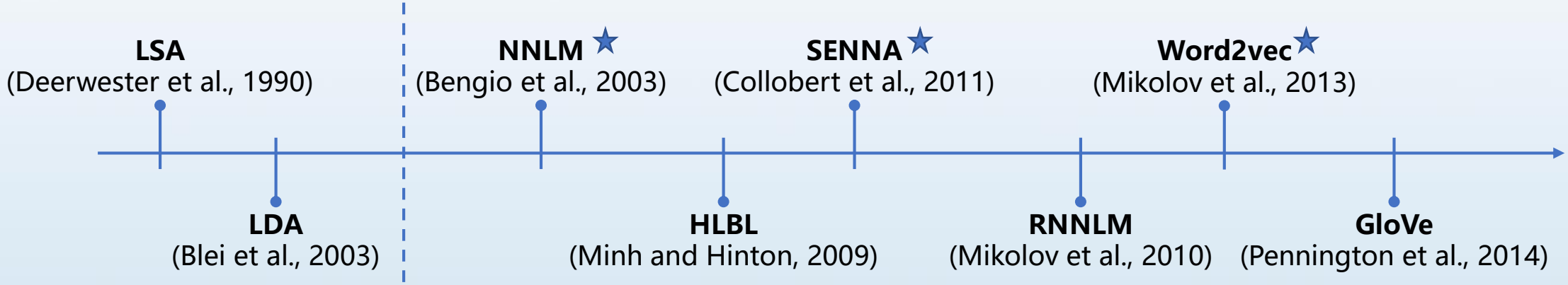
$U \in \mathrm{R}^{|V|\times d}, V \in \mathrm{R}^{d\times|C|}$为正交矩阵

避免稀疏性，反映高阶共现关系

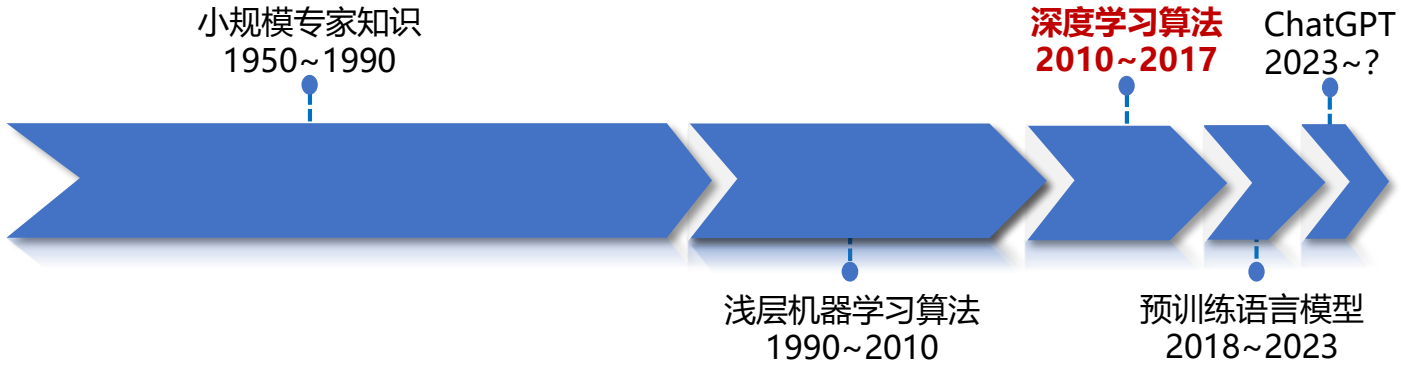
- U的每一**行**代表每个词的d维向量表示
  - U的各列之间正交（词表示每一维度正交）
    - 词表示的每一维度表达了一种***独立***的潜在语义
  - 类似的$\sum \mathrm{V}^T$的每一**列**也可以作为向量表示

分布式（Distributed）词表示：应运而生

- 分布表示的缺点
  - 训练速度慢，增加新语料库困难
  - 不易扩展到短语、句子表示
- 分布式表示直接使用低维、稠密、连续的向量表示词
  - 通过“自监督”的方法直接学习词向量
  - 也称词嵌入（Word Embedding）
- 发展历程

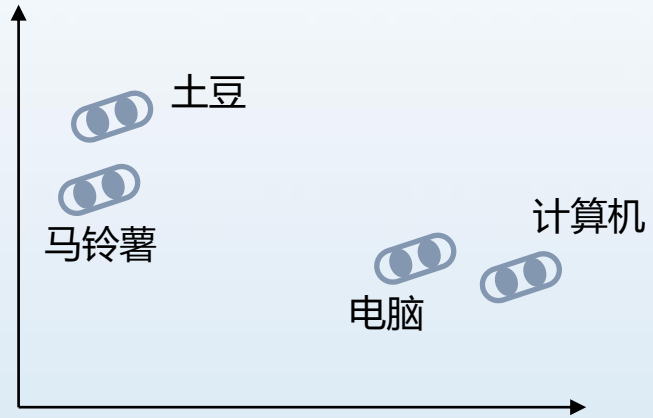


基于嵌入表示的深度学习



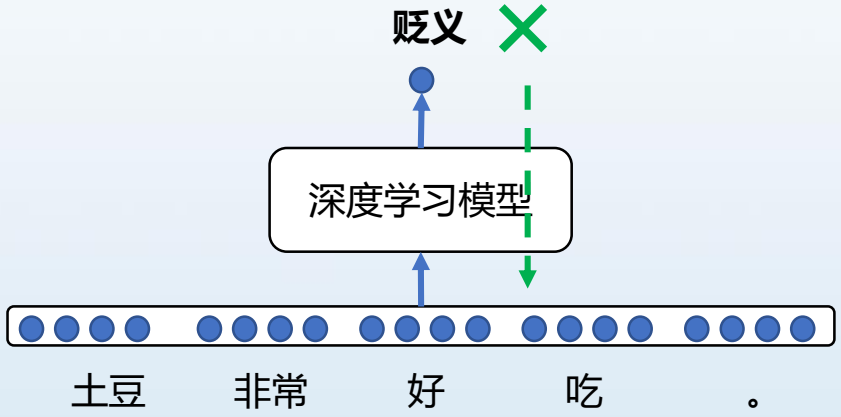
词嵌入（Word Embedding）

- 直接使用一个**低维、连续、稠密**的向量表示词（Bengio等2003）

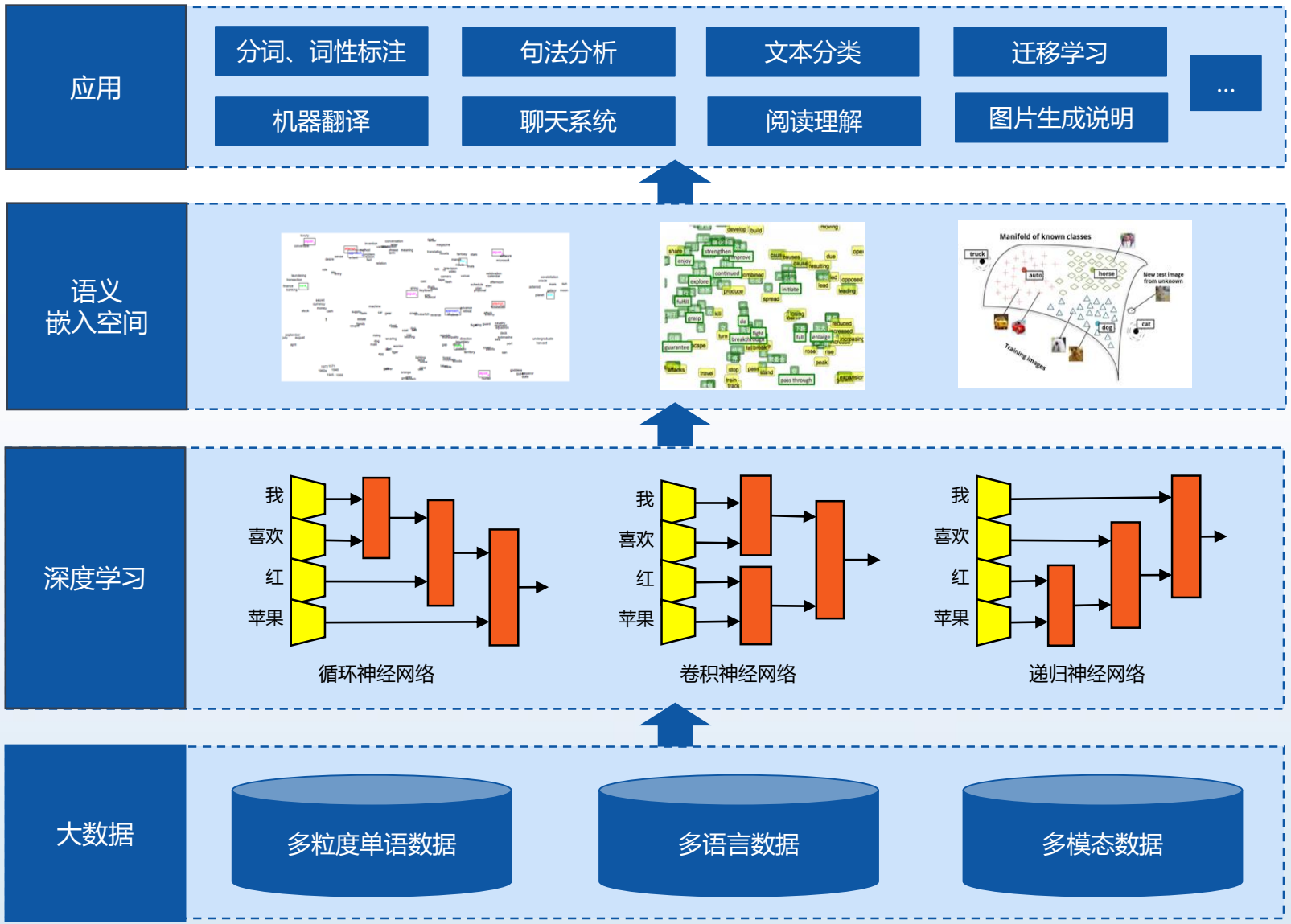


词嵌入表示的赋值方法

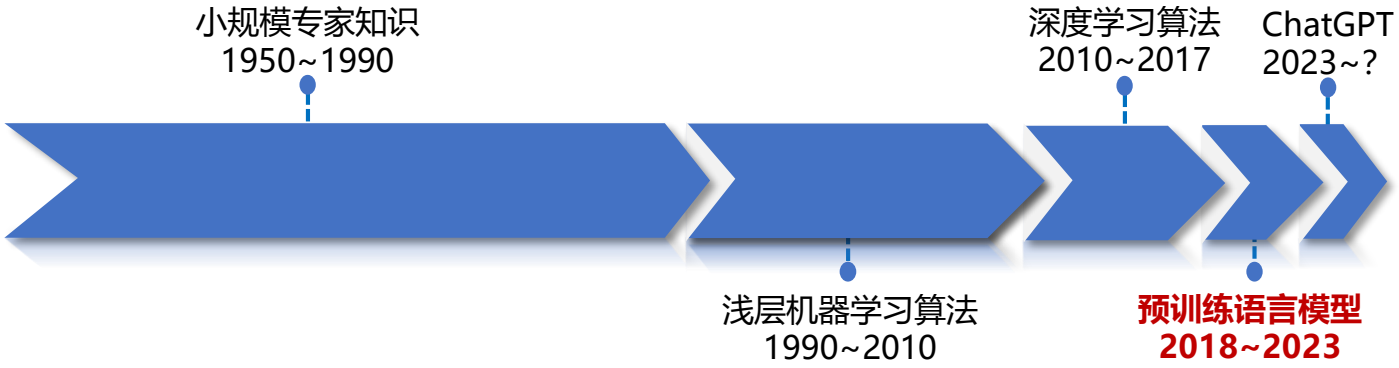
- 通过优化在**下游任务**上的表现自动学习

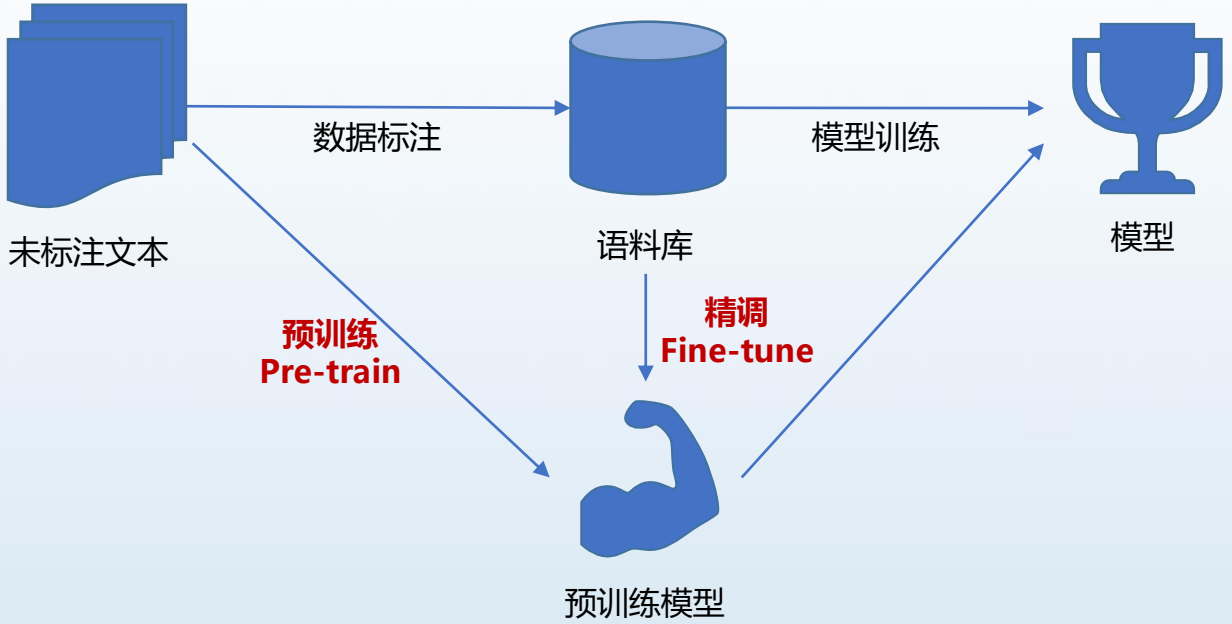


各种语言单元的统一嵌入表示



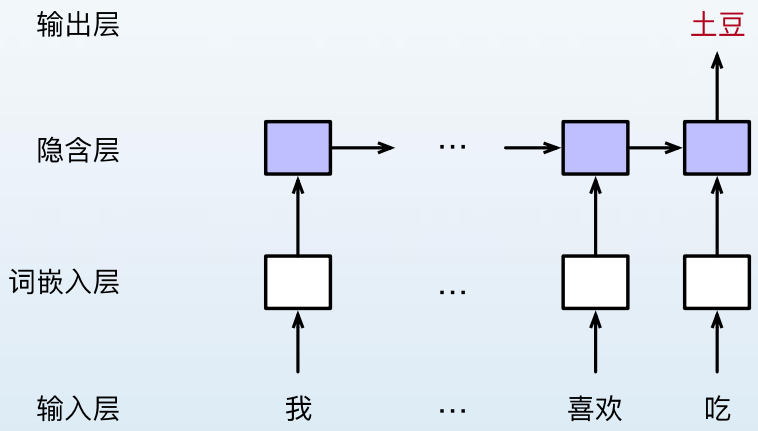
预训练语言模型


小规模专家知识
1950~1990
浅层机器学习算法
1990~2010
深度学习算法
2010~2017
ChatGPT
2023~?
预训练语言模型
2018~2023

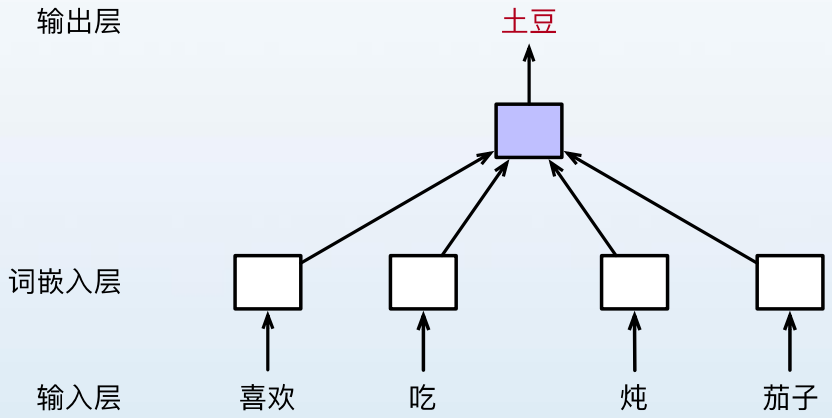

数据标注
模型训练
未标注文本
语料库
模型
预训练
Pre-train
精调
Fine-tune
预训练模型

如何利用未标注数据预训练？

- 利用语言天然的顺序性
  - *我 喜欢 吃 土豆 炖* **XX**
- 两种任务类型
  - 语言模型
    - 通过历史词序列预测**下一个词**

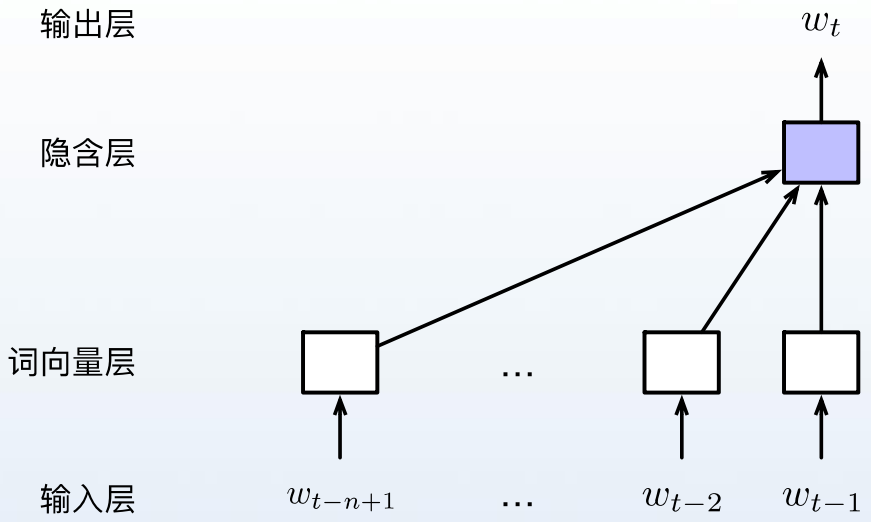


  - 完形填空
    - 通过周围的词预测**中间的词**



回顾：前馈神经网络语言模型（FF-NNLM）

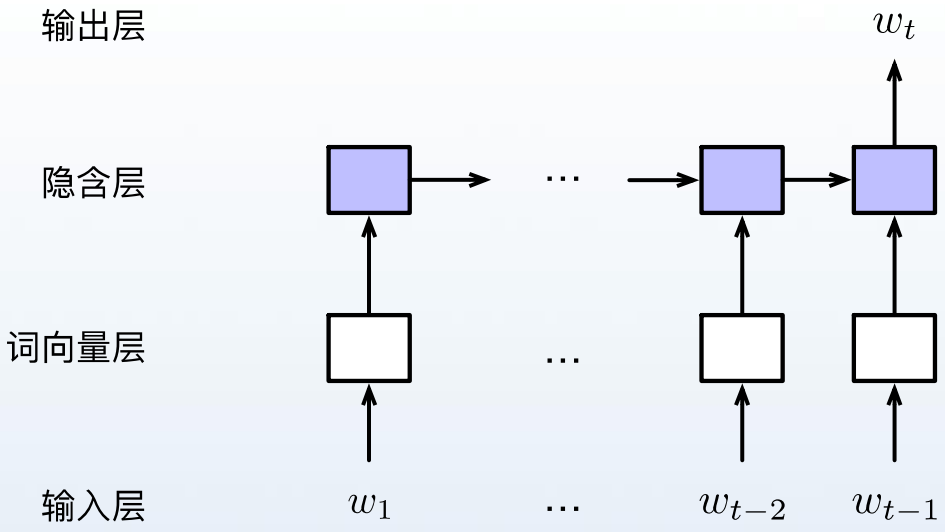
- Neural Network Language Models (Bengio et al., JMLR 2003)
  - 根据前n-1个词（历史）预测当前词，即马尔可夫假设
  - 模型结构为前馈神经网络
  - 通过查找表（Look-up Table），获得词的向量表示
    - 词向量（或词嵌入，Word Embedding）
    - 支撑图灵奖的重要工作
  - 通过梯度下降优化词向量表示



- 缺点
  - “历史”长度不可变
    - “他 喜欢 **吃** 苹果”
    - “他 **感冒** 了， 于是 下班 后 去 了 医院”

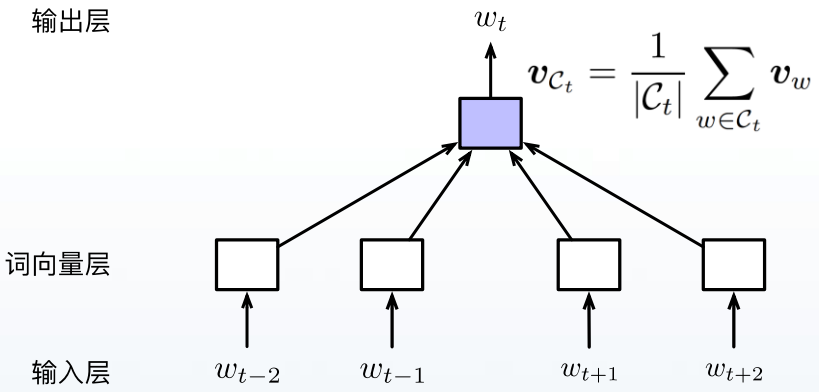
回顾：循环神经网络语言模型（RNNLM）

- Recurrent Neural Network Language Models (Mikolov et al., Interspeech, 2010)
  - 根据完整的“历史”对当前词进行预测
  - 对不定长依赖的建模能力
  - 梯度弥散/爆炸问题
    - 反向传播过程中按长度进行截断
    - 长短时记忆网络（LSTM）
- 缺点
  - “语言模型”约束
    - ：只利用了“历史”信息

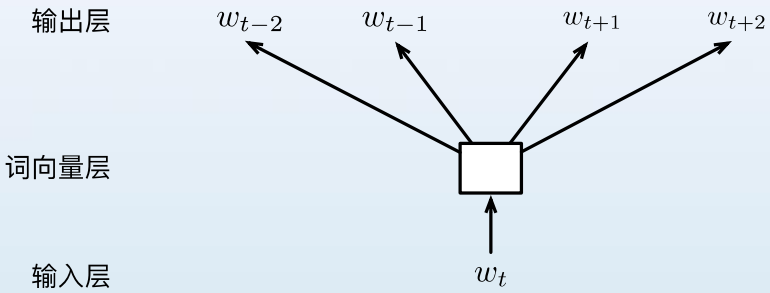


Word2vec

- https://code.google.com/archive/p/word2vec/
  - Mikolov et al., ICLR 2013
- CBOW (Continuous Bag-of-Word)
  - 根据周围词（上下文）预测中间词
  - 如何计算上下文表示：词向量取平均
- Skip-Gram
  - 根据中间词**独立地**预测周围词（上下文）
- 训练速度快
  - 可利用大规模数据
  - 弥补了模型能力的不足



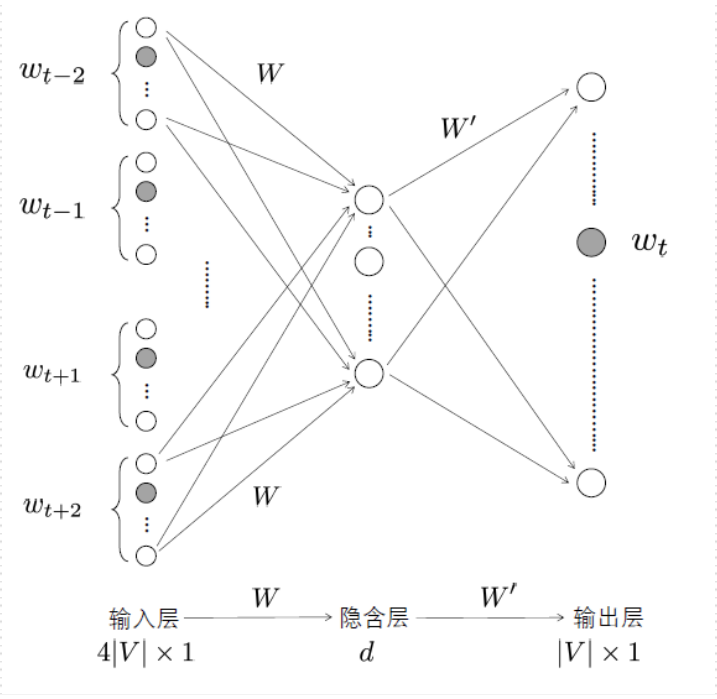
CBOW模型示意图



Skip-gram模型示意图

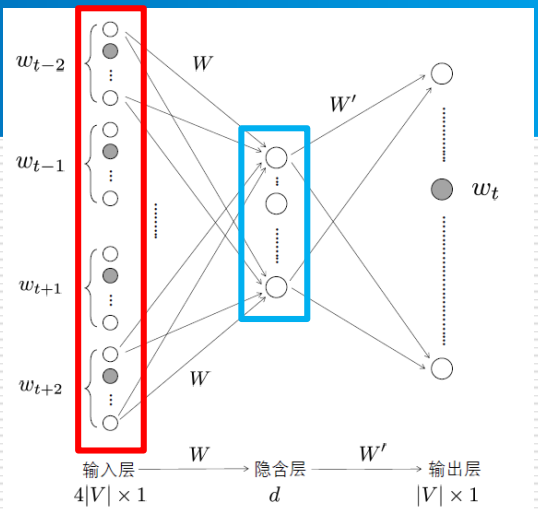
CBOW模型

- CBOW的基本思想是利用一定窗口大小内的上下文C对目标单词w_t进行预测



Eg：利用窗口大小为5的上下文C对w_t进行预测

CBOW模型



- 模型结构：
  - 输入层：
    - 记上下文单词的独热表示向量$OneHot(w_i)$
  - 隐含层：
    - 首先将输入层每个单词利用矩阵$W \in \mathrm{R}^{d \times |V|}$映射至隐含空间：

$$设\ v(w_i) = \boldsymbol{W} \cdot OneHot(w_i)$$

    - 对于多个词组成的上下文C，对所有的词向量取**平均值**作为上下文表示：

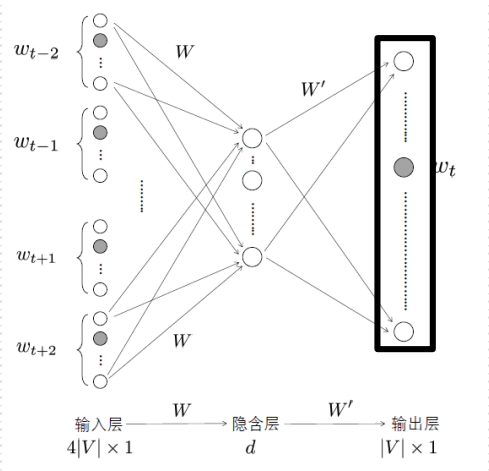
$$v(C) = \frac{1}{|C|} \sum_{w \in C} v(w)$$

CBOW模型

- 模型结构：
  - 输出层：
    - 和前馈神经网络语言模型基本一致。令$W' \in \mathrm{R}^{|V|\times d}$作为输出层权值矩阵。利用$W'$和$W$相应行列内积进行结果概率预测：

      记W'的隐含空间表示为$v'(w)$

$$p(w_t|C) = \frac{e^{<v(c),v'(w_t)>}}{\sum_{w'\in V} e^{<v(c),v'(w')>}}$$



- 最终W或W'或二者组合均可作为词向量矩阵

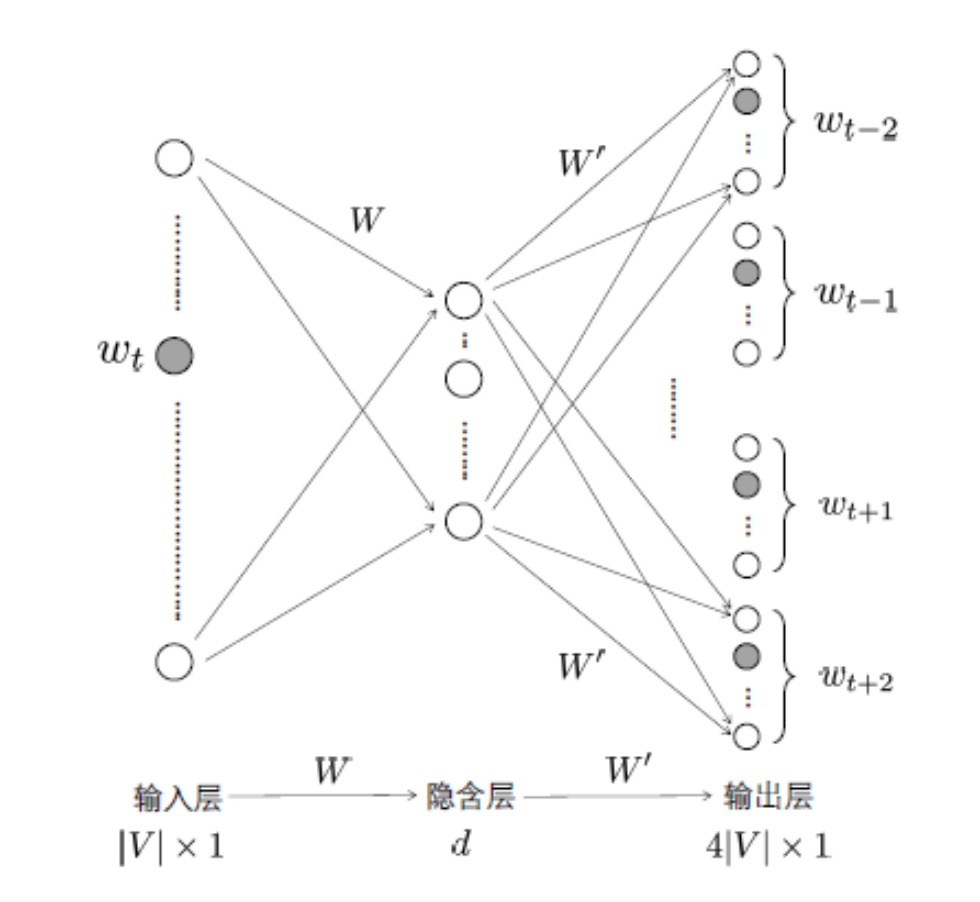
Skip-gram 模型

- CBOW模型利用上下文窗口中的词$C = \{w_{t-k}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+k}\}$计算$p(w_t|C)$

- Skip-gram对C进行了简化，每个词都可以做独立的上下文，计算概率改变为$p(w_t|w_{t+\mathrm{j}})\ (j \in \pm 1, \dots \pm k)$
  - 原论文采用了等价描述$p(w_{t+\mathrm{j}}|w_t)$，计算方法为：

$$p(w_{t+j}|w_t) = \frac{e^{<v(w_t), v\prime(w_{t+\mathrm{j}})>}}{\sum_{w\prime \in V} e^{<v(w_t), v\prime(w\prime)>}}$$

Skip-gram 模型



Word2vec 的参数估计

- 优化目标
  - 对目标词进行预测（Softmax），优化分类损失
    - CBOW

$$P(w_t|\mathcal{C}_t) = \frac{\exp(\boldsymbol{v}_{\mathcal{C}_t} \cdot \boldsymbol{v}'_{w_t})}{\sum_{w' \in \mathbb{V}} \exp(\boldsymbol{v}_{\mathcal{C}_t} \cdot \boldsymbol{v}'_{w'})}$$

    - Skip-Gram

$$P(c|w_t) = \frac{\exp(\boldsymbol{v}_{w_t} \cdot \boldsymbol{v}'_{c})}{\sum_{w' \in \mathbb{V}} \exp(\boldsymbol{v}_{w_t} \cdot \boldsymbol{v}'_{w'})}$$

    - 缺点：当词表较大且计算资源有限时，概率（归一化）计算效率较低
  - 负采样（Negative Sampling）
    - 对（词，上下文）进行二元分类，1表示在给定上下文内**共现**，0表示**不共现**
    - 与SENNA思想近似，通过“换词”构造（词，上下文）负例

$$\log \sigma(\boldsymbol{v}_{w_t} \cdot \boldsymbol{v}'_{w_{t+j}}) + \sum_{i=1}^{K} \log \sigma(-\boldsymbol{v}_{w_t} \cdot \boldsymbol{v}'_{\tilde{w}_i})$$

$\tilde{w}_i \sim P_n(w)$（负采样分布）

注意：词与上下文分别使用不同的向量矩阵

GloVe

- GloVe: Global Vectors for Word Representation (Pennington et al., EMNLP 2014)
  - 利用“词-上下文”共现信息
    - Word2vec：局部共现，只考虑当前样本中是否共现
    - GloVe：利用全局统计信息，即共现频次
  - 利用词向量对“词-上下文”共现矩阵进行预测（或回归）
    - 构建共现矩阵：共现“强度”按照距离进行衰减

$$M_{w,c} = \sum_i \frac{1}{d_i(w,c)}$$

    - 回归目标：

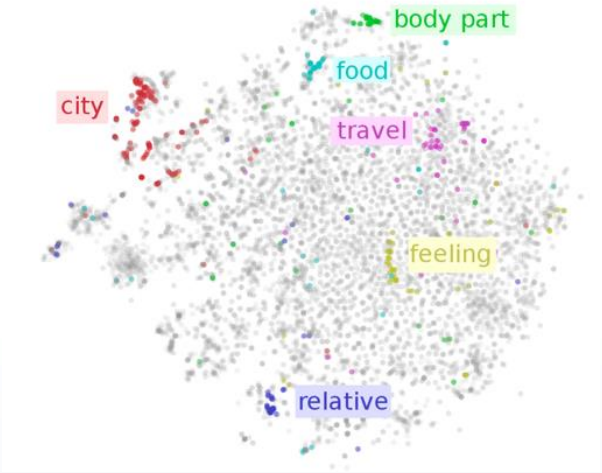
$$\boldsymbol{v}_w^\top \boldsymbol{v}'_c + b_w + b'_c = \log M_{w,c}$$

  - 参数估计

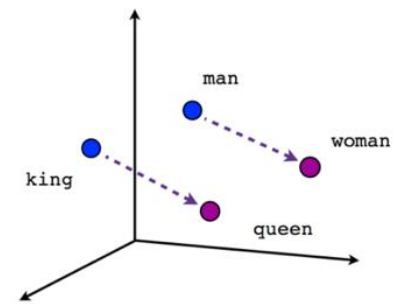
$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{M}) = \sum_{(w,c)\in\mathbb{D}} f(M_{w,c})(\boldsymbol{v}_w^\top \boldsymbol{v}'_c + b_w + b'_c - \log M_{w,c})^2$$

样本权重

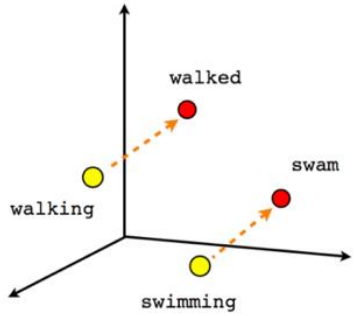
词向量的评价与应用



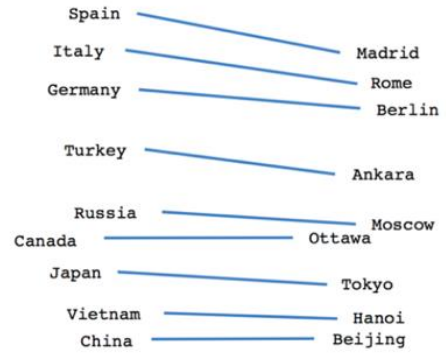
词义相似度计算



Male-Female

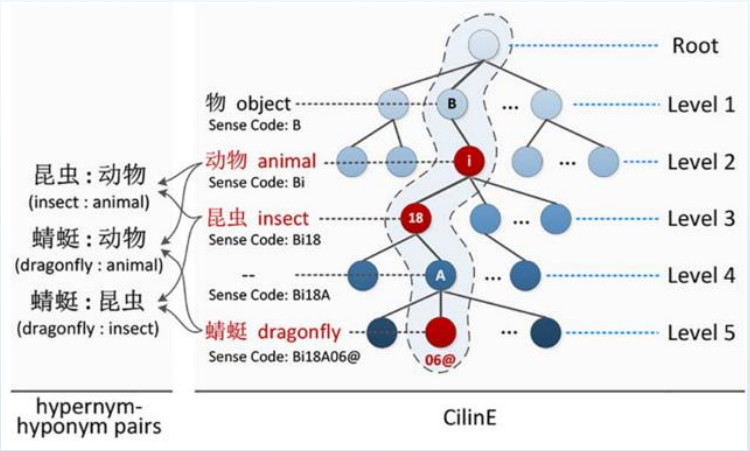


Verb tense

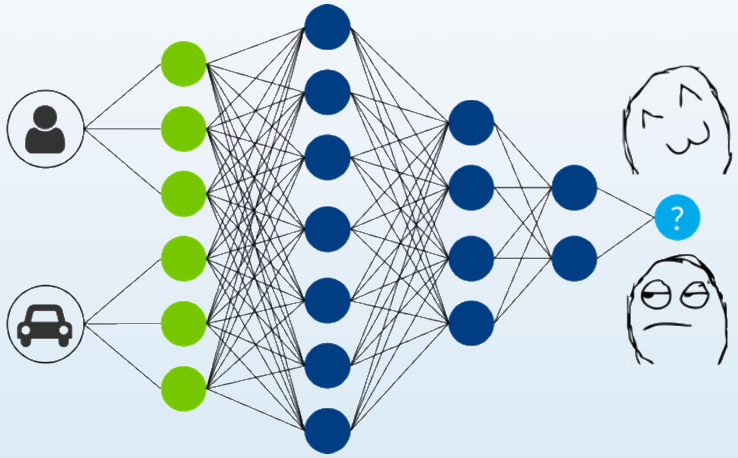


Country-Capital

词类比关系计算



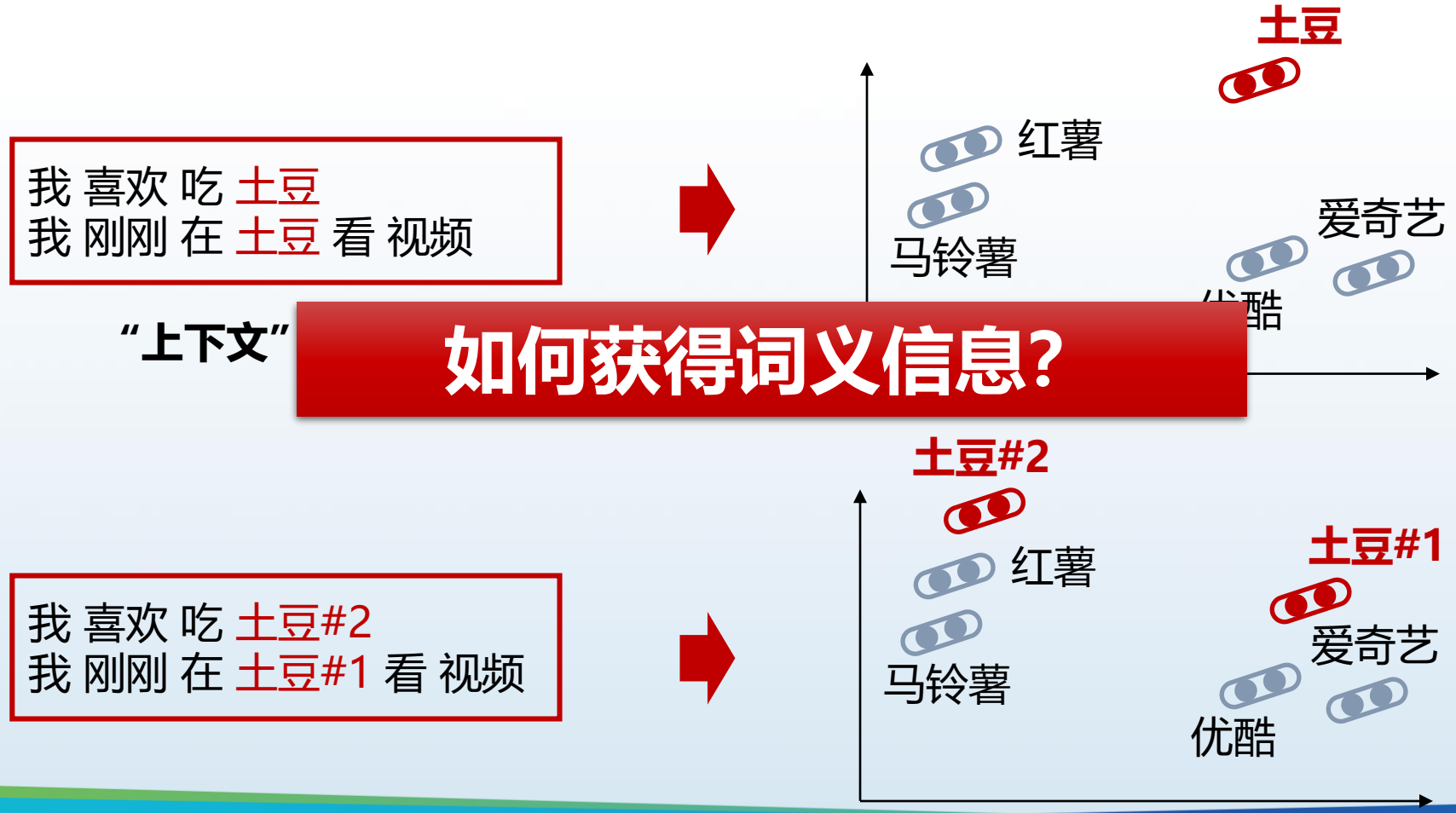
知识图谱补全



推荐系统

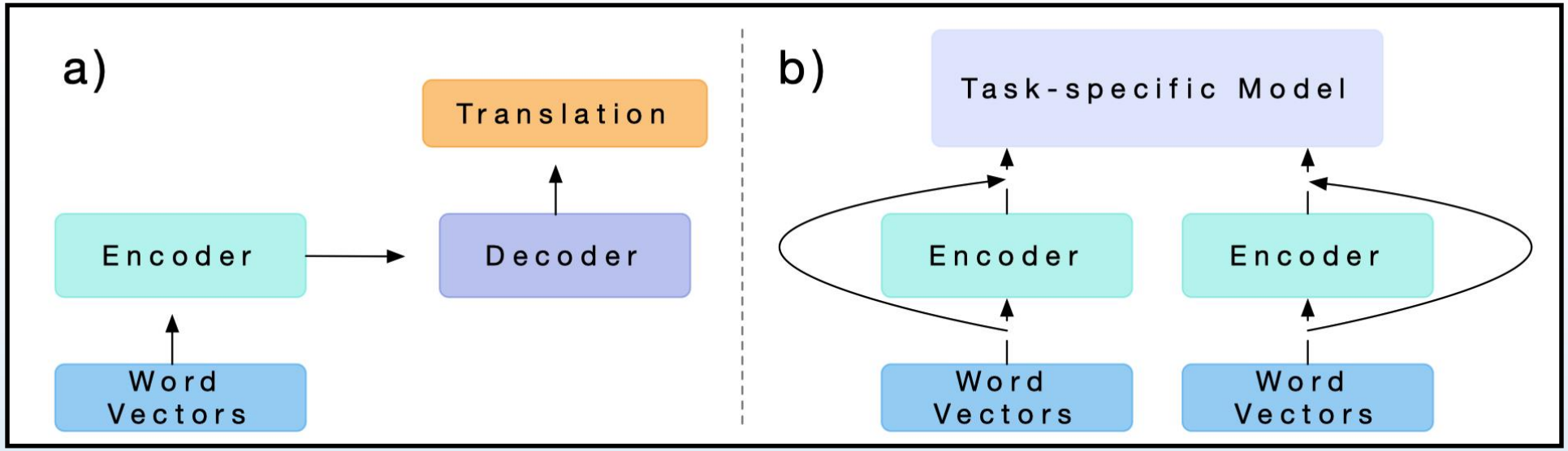
静态词向量不足：一词多义现象

- 静态词向量假设一个词由**唯一**的词向量表示
  - 无法处理一词多义现象



词向量——从静态到动态

- CoVe (Contextualized Word Vectors)
  - 提出使用上下文相关的文本表示，即每个token的向量表示不唯一
  - 主要思想：将神经机器翻译（NMT）的表示迁移到通用NLP任务上



基于语言模型的动态词向量预训练

CoVe存在的问题

训练依赖于双语平行语料

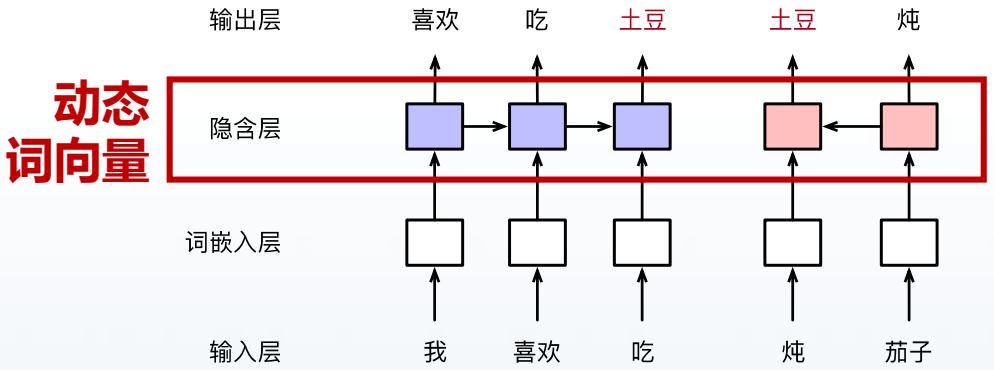
- 训练神经机器翻译模型需要双语平行语料，获取难度较高
- 相比单语语料，覆盖的领域也相对优先，通用性一般

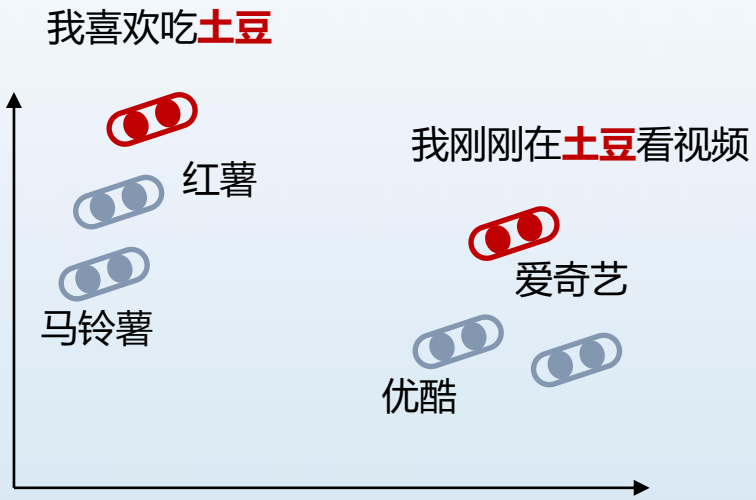
单独使用效果一般，性价比不高

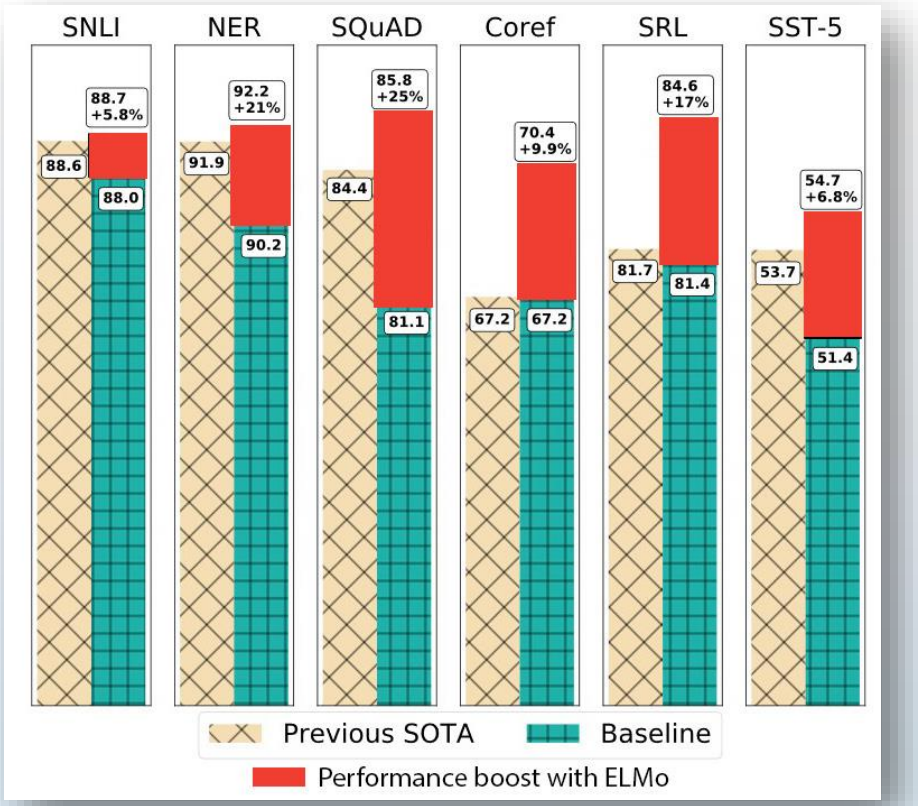
- 实验结果表明单独使用CoVe的效果一般
- 需要搭配传统静态词向量才能获得较为显著的性能提升

ELMo

- Embeddings from Language Models，AI2 2017
  - 基于LSTM的**语言模型**

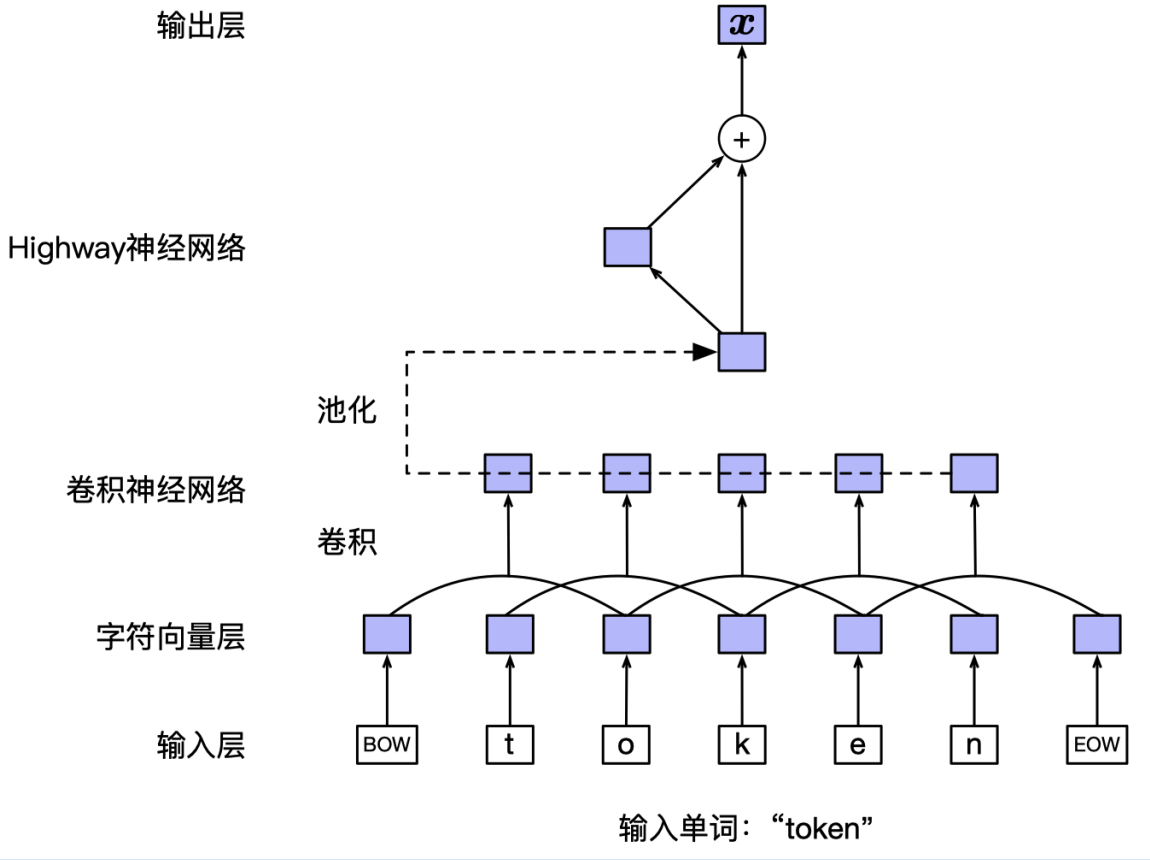






基于语言模型的动态词向量预训练

- ELMo模型使用字符的CNN模型表示词
  - 具有泛化作用



基于语言模型的动态词向量预训练

- 双向语言模型BiLM
  - 从前向（从左到右）和后向（从右到左）两个方向同时建立语言模型

- 前向语言模型

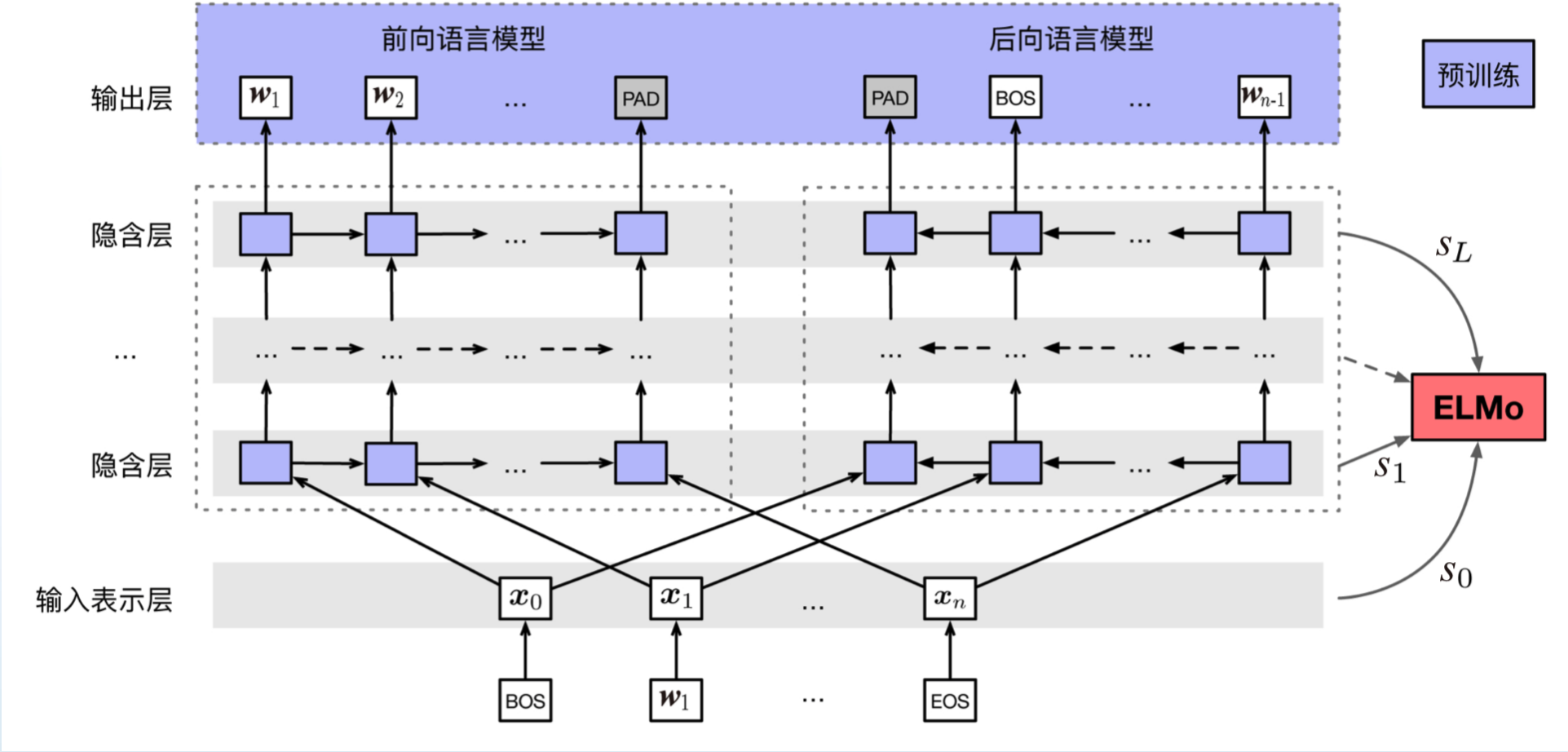
$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^{n} P(w_t | \boldsymbol{x}_{1:t-1}; \overrightarrow{\boldsymbol{\theta}}^{\text{lstm}}, \boldsymbol{\theta}^{\text{out}})$$

- 后向语言模型

$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^{n} P(w_t | \boldsymbol{x}_{t+1:n}; \overleftarrow{\boldsymbol{\theta}}^{\text{lstm}}, \boldsymbol{\theta}^{\text{out}})$$

基于语言模型的动态词向量预训练

双向语言模型BiLM



基于语言模型的动态词向量预训练

- ELMo词向量
  - ELMo采取对不同层次的向量表示进行加权平均的机制，为不同的下游任务提供更多的组合自由度

$$\mathbb{R}_t = \{\boldsymbol{x}_t, \boldsymbol{h}_{t,j} | j = 1, \cdots, L\}$$

$$\text{ELMo}_t = f(\mathbb{R}_t, \Psi) = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{\text{task}} \boldsymbol{h}_{t,j}$$

  - ELMo特点
    - **动态（上下文相关）**：词的ELMo向量表示由其当前上下文决定；
    - **鲁棒（Robust）**：ELMo向量表示使用字符级输入，对于未登录词具有强鲁棒性；
    - **层次**：ELMo词向量由深度预训练模型中各个层次的向量表示进行组合，为下游任务提供了较大的使用自由度。

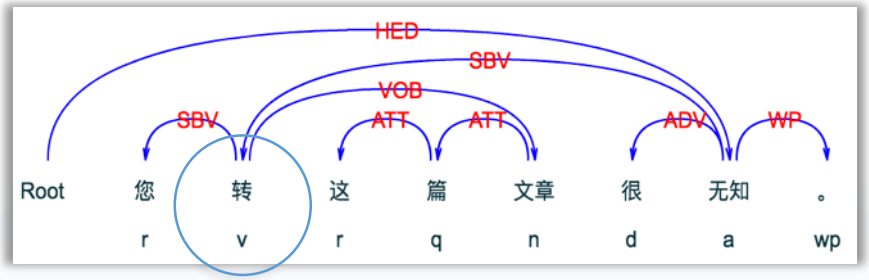
基于语言模型的动态词向量预训练

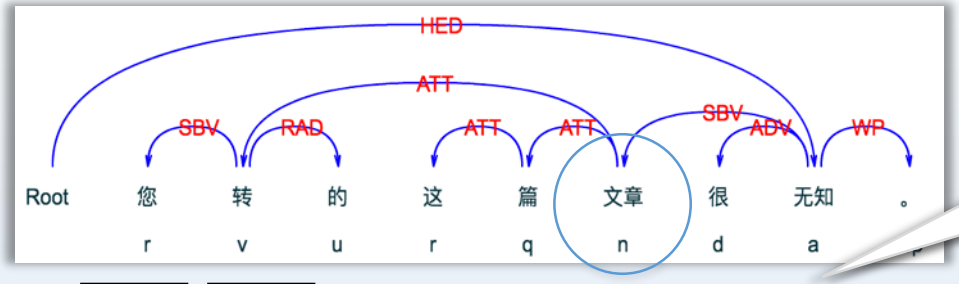
- 应用与评价
  - 上下文相关的词义相似性检索
    - ELMo相比GloVe（静态词向量）在词义消歧和近邻分析任务上都有比较好的表现

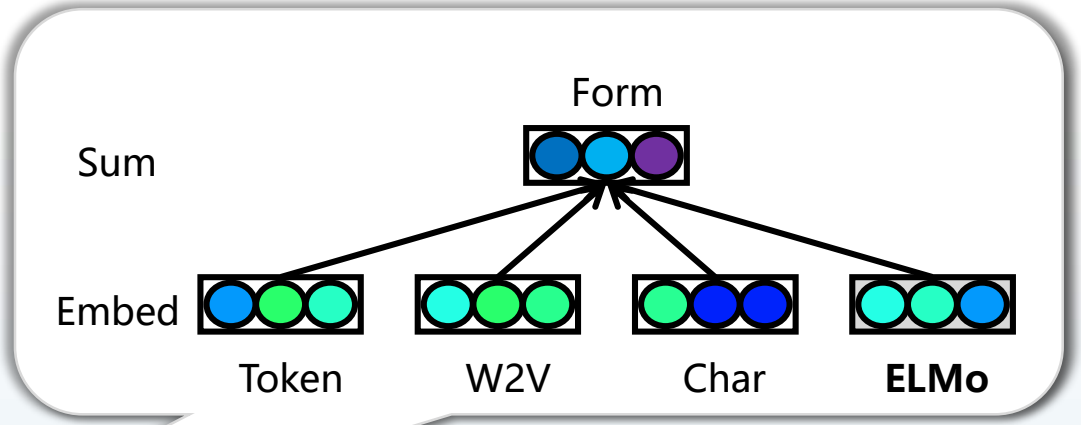
模型	词	近邻
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
ELMo	Chico Ruiz made a spectacular play on Alusik's grounder ···	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play
	Olivia De Havilland signed to do a Broadway play for Garson ···	··· they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement

基于ELMo的应用示例

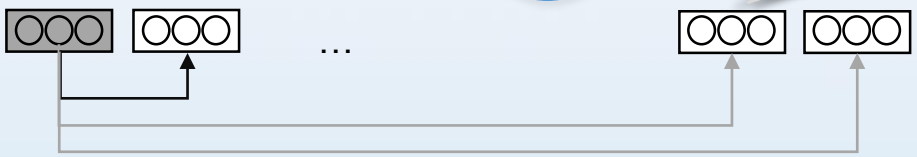
- 依存句法分析 (Che et al., CoNLL 2018[SCIR])







http://ltp.ai/



分类

谢 谢!

语言技术紫丁香

微信扫描二维码，关注我的公众号