语言计算

——以词表示为例

王厚峰

wanghf@pku.edu.cn

主要内容

- 概述
- 词的表示
- 词表示的统计计算方法
- 词表示的训练方法

从人工智能谈起

- Alpha Go产生了广泛的影响

2017-5

2016-3

IBM Deep Blue(1997) ->IBM Waston DeepQA(2011)

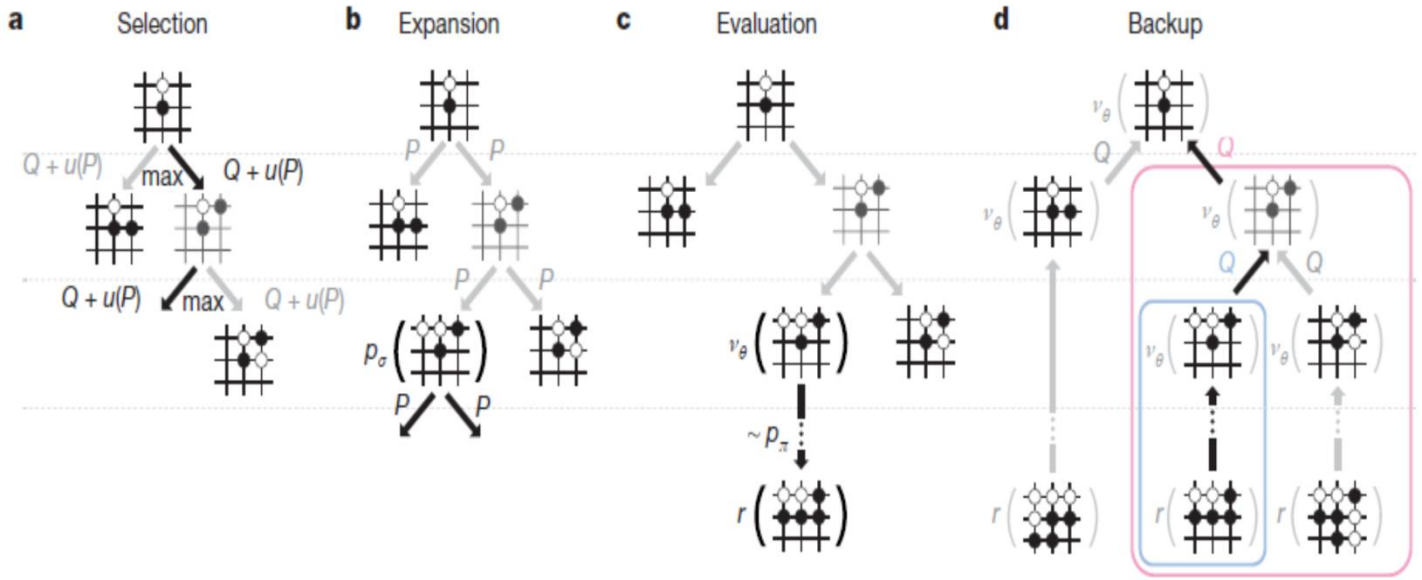
Ken:连胜纪录保持者； Brad:最高奖金得主

Siri人机语音交互（2011年）

围棋为什么会引起轰动

- 围棋太难
  - 搜索空间太大：19*19棋盘约需250^{150}搜索（汉诺塔64块盘，移动2^{64},每秒移动一次，需$5*10^{11}$年）
- 蒙特卡洛搜索树(MCTS)中嵌入深度神经网络
  - 目的：减少搜索空间，寻找近似解

计算智能



AlphaGo vs IBM Waston Deep QA

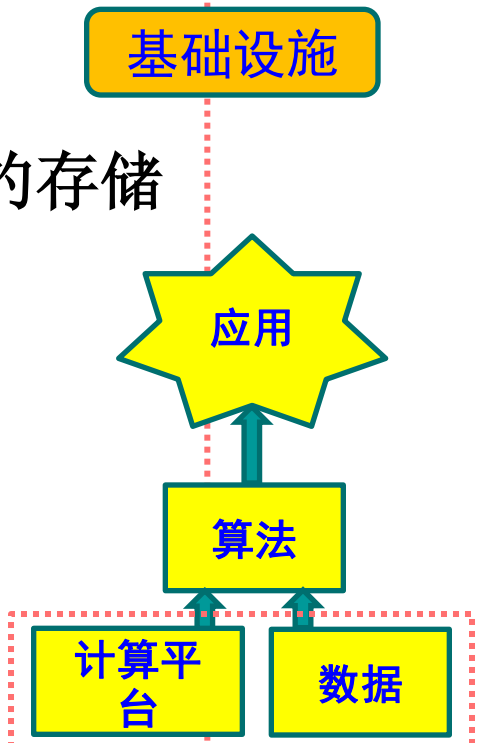
- 庞大的知识体系
  - 字典、百科全书、网页主题分类、宗教典籍、小说、戏剧和其他资料
- 强大的知识库搜索
  - 使用数以百计的算法，来搜索问题的候选答案、并对每个答案进行评估打分，同时为每个候选答案收集其他支持材料和证据

计算智能

- 多层面的语言分析功能
  - 词法分析，句法分析，问题分类，实体识别，语义分析等
- 答案生成能力
  - 衔接而连贯的语言表达

AI再度兴起的3大助推器

- 强大的计算能力
  - 高性能、分布式、云平台
  - 智能问题需要高性能的计算+大容量的存储
    - AlphaGo的状态空间搜索
    - DeepQA 的知识库搜索
- 大数据
  - 规律（或知识）就在数据之中
  - 网络化与数字化造就了大数据
  - 评测的推动加强了数据的产生



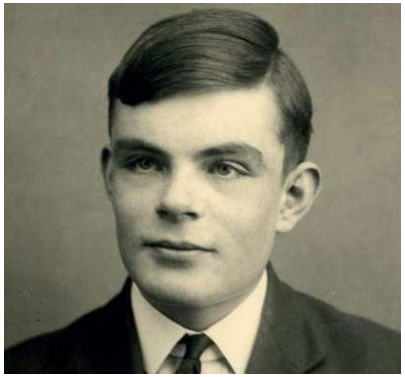
- 以深度学习为代表的新方法、新技术
  - 面对多层神经网的问题，深度学习有新的突破
  - 数据+评测推动了技术发展

究竟什么是人工智能

计算机器与智能(Computing Machinery and Intelligence)，1950

❖ 人工智能

- 图灵测试（Turing Test）：



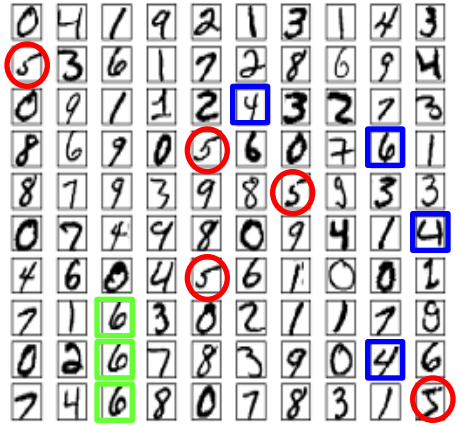
1912-1954

  - 直观、易懂
  - 缺乏形式描述、没有客观标准
- 以自然语言理解为测试手段：对话
  - NLU 的重要性显而易见

计算智能->感知智能->认知智能

❖ 计算智能

- 通过强大的计算能力解决复杂计算问题
  - 经典的博弈（回溯/搜索）——NP难
  - 并行/近似



❖ 感知智能

- 手写数字识别
- 语音识别：噪音/口音/语速(吞/添音)

❖ 认知智能—自然语言处理

- 机器翻译
- 问答与知识服务
- 聊天机器人

模糊(不确定)
隐含(不直接)
知识与推理

语言计算的不同层次

- **词层面**
  - 汉语切词
    - 南京市长江大桥 => 南京 市长 江大桥/南京市 长江 大桥
  - 词性（汉语 把：量词+介词+名词+动词）
  - 词义（汉语 打：打的+打电话+打球+打人+打酱油+…）
  - 新词新语（打酱油）、缩略（人大）…
- **句层面**
  - 句子成份（主、谓、宾）、成份依赖
  - 词/短语的角色

老鼠咬死猎人的狗

咬死猎人的狗

咬死猎人的狗又在咬老虎

- **句间与篇章**
  - 省略、指代（具体指代：我你他它；抽象指代：这那）
    - 我自来是如是，()从会吃饮食时便吃药，()到今未断()。()请了多少名医，()修方配药，()皆不见效
  - 句间关系（因果、转折、并列…）

语言之外还有很多知识

❖ 语言处理需要的知识永远不完备

- 社会在发展、知识在更新
  - 词典量不足：本身无法穷尽所有词汇
  - 词的很多知识没有刻画（以“树”为例）
    - 作名词/动词使用
    - 木本植物
    - 可以植、栽、锯、砍、挖、烧
    - 有杆、枝、叶…
    - 可以用于绿化
    - 可以防护沙尘/遮太阳
    - 可以攀爬
    - 可以由鸟筑巢
    - 可以造纸
    - 可以用于做家具，可以建房…

语法义
概念义
情感义
联想义
社会义
……

符号（词）：树

知识是联想和推理的基础

语言智能超出语言本身

- **歧义问题**
  - 四川省十三届人大常委会举行第一次会议任命汪洋为四川省交通运输厅厅长
  - 汪洋在四川甘孜调研（全国政协主席）
  - 汪洋组织创作了《祝福》《早春二月》《林家铺子》《骆驼祥子》《小花》等众多经典电影，让北影成了一块金字招牌
  - 一片汪洋都不见，知向谁边
  - 能穿多少就穿多少（夏天去海南，冬天去东北）
- **深层语义问题（超出字面意）：**
  - 超出字面义。如：金融风暴，民族的脊梁，蓝色的海洋/红色的海洋
- **特定的知识**
  - 该来的没有来
  - 不该走的走了

内含有大量知识超越了语言本身

语言计算？知识计算？

- **重新认识语言计算问题**
  - 语言计算要解决的不仅仅是语言问题
  - 再看Bar-Hillel的批评： real world knowledge is needed
    - 著名的歧义例子 the box was in the pen
    - 例子的本质：知识问题：知识足够吗？如何使用知识
- **语言计算为什么难？**
  - 人的要求高？计算机不能收纳所有人的知识
    - 任何人几乎都能指出自己能轻松解决而 NLP不能解决的问题
      - 计算机为什么这么笨？
  - 求解问题所需的知识还没有充分了解（或掌握）
    - 计算机能读懂《红楼梦》吗？人都没有完全弄清楚（红学会继续的使命）

语言可计算吗?

- **回顾《计算机不能做什么——人工智能的极限》**
  （生活读书新知三联书店. 1986）
  - 译者序（马希文）的观点（三个方面）
    - 计算机求解问题的先决条件是问题必需形式化
      - 语言是否可以形式化
      - 语言如何形式化
      - 如果语言与外界有千丝万缕的联系，是否要对整个世界形式化
    - 即便问题可以形式化，是否可以计算（存在算法）
      - 已有例子表明，可形式化的问题未必可计算（刁番图方程）

系数、指数和常量均为整数 $a_1x_1^{b_1} + a_2x_2^{b_2} + \ldots + a_nx_n^{b_n} = c$

      - 没有算法判断任意给定的多项式方程是否有整数解（ Hilbert: Problem 10: Diophantus Equation）
    - 即便有算法，其复杂度如何控制在合理的范围内（围棋）

主要内容

- 概述
- 词的表示
- 词表示的统计计算方法
- 词表示的训练方法

语言计算之语言表示

语言表示

- 语言计算的前提是语言表示
  - 词汇表示
  - 句子表示
  - …

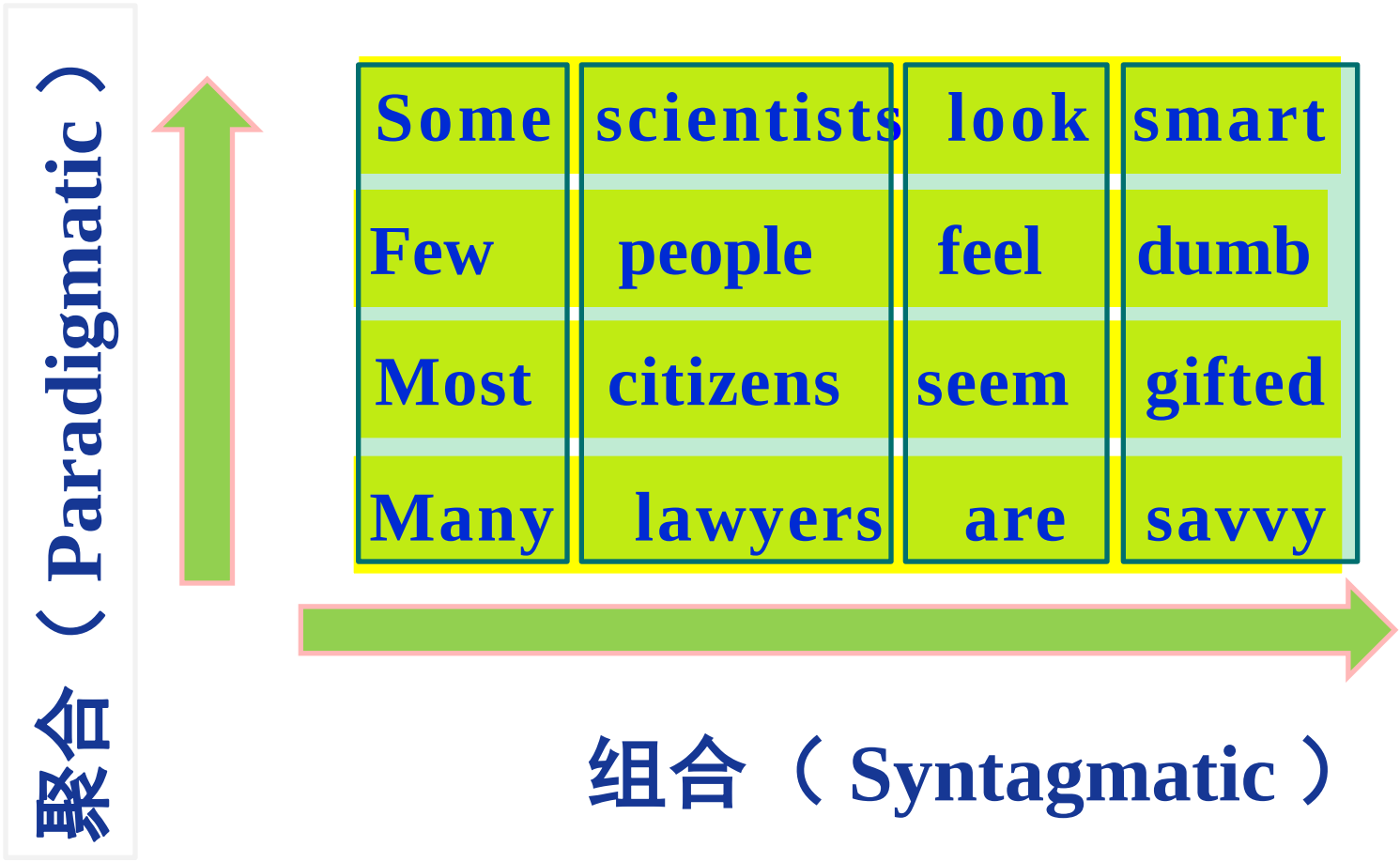
句子的形式化表示

- 文法：用于分析句子结构
  - **N.Chomsky的文法体系（Type-2：上下文无关文法CFG）**
  - 带概率的上下文无关文法 **P-CFG**
  - …
- 语言模型的统计表示（ **n-gram** ）
  - 用于度量生成结果的合理性
  - $P(w_n|w_{n-1}w_{n-2}\cdots w_1)$

词及其特征

- **语言学上词的功能或意义**
  - Syntagmatic relations(组合关系：由句法体现)
    通过与其它词之间的搭配组合体现词义 (co-occurrences)
  - Paradigmatic relations（聚合关系：由概念关系体现）
    通过词之间的排斥关系体现 (substitutions)
- **词义定义:Webster dictionary**
  - the idea that is represented by a word, phrase, etc.
  - the idea that a person wants to express by using words, signs, etc.
  - the idea that is expressed in a work of writing, art, etc.

组合 vs 聚合


聚合（Paradigmatic）
Some scientists look smart
Few people feel dumb
Most citizens seem gifted
Many lawyers are savvy
组合（Syntagmatic）

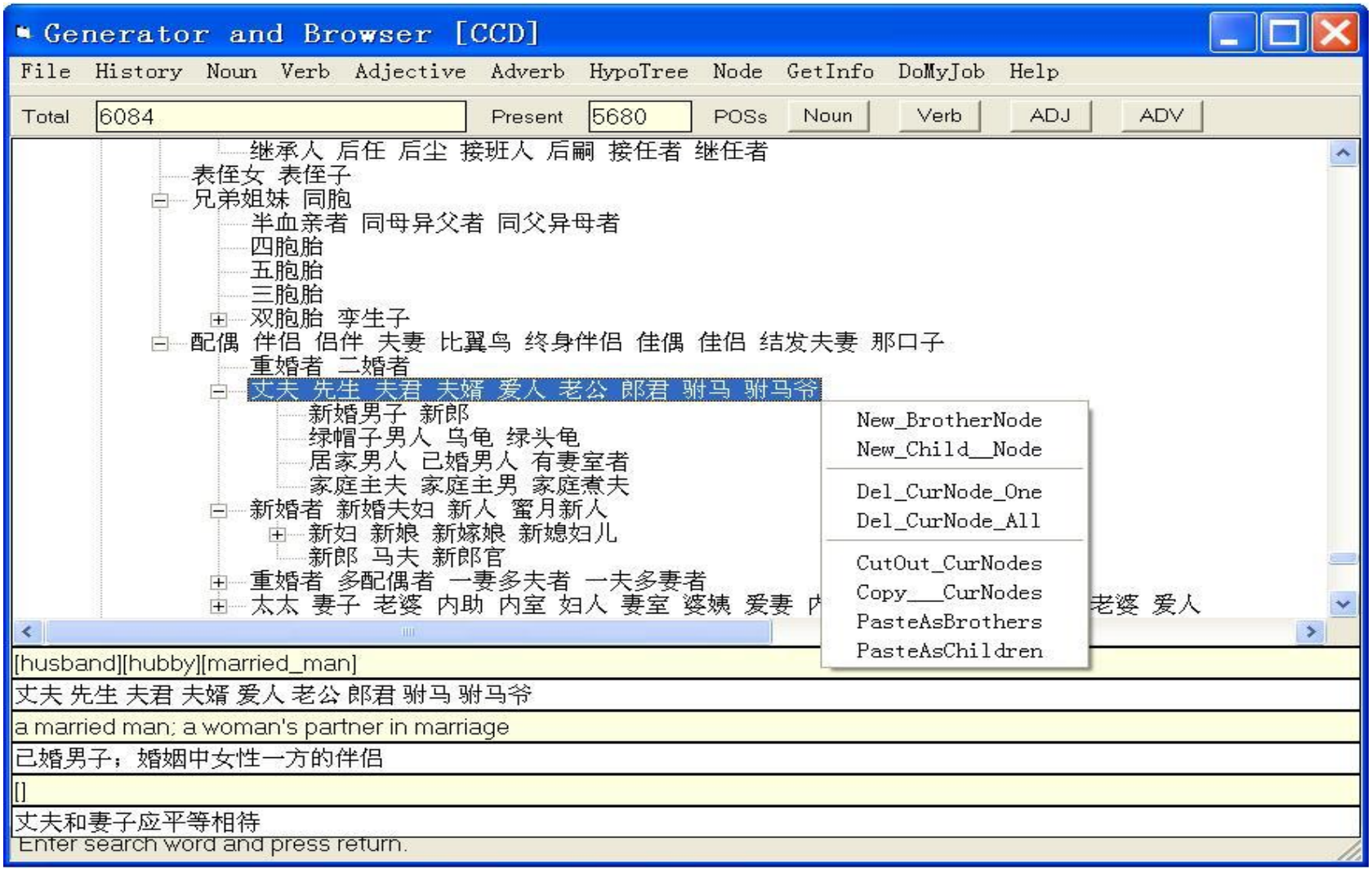
典型的组合关系

- 组合关系
  - 搭配
    - 论文 & 写，论文 & 读， 论文 & 出版
    - 钢琴 & 演奏，钢琴 & 生产/制造
  - 联想(相关)
    - **互存性：**医生 & 病人，医生 & 处方，医生 & 医院（诊所），医生 & 护士
    - **构成性：**论文 & 字，论文 & 表，论文 & 图，论文 & 标题
    - **特质性：**松树 & 常青，荷花 & 洁白，冰雪 & 寒冷，牡丹 & 富贵，石头 & 坚硬，海绵 & 柔软

典型的聚合关系

- **聚合关系（离散表示）**
  - 使用同义词集（如汉语《同义词词林》）
  - 著名的 WordNet类词典
    - 词义（ Word senses ）由同义词集定义—同义词集由具有相同意义的词构成
    - 同义词集也称为概念（Concept），概念之间连接成为网络结构，连接关系称为语义关系。典型的关系：
      - hypernymy（上-下位）
      - antonymy（同义-反义）
      - holonymy（整体-部分）

中文概念词典CCD（类Wordnet）



“先生”的意义

概念编号	Synset	Csynset	上位概念	下位概念	Definition	Cdefinition
07632177	teacher instructor	教师 教员 老师 先生 导师 …	07235322	07086332071623040720946507243767072796590729762207341176074010980741425107425180074940250752093807533674075514040755158107561151076326240763 2736	a person whose occupation is teaching	以教学为职业的人

表示职业

“先生”的意义（续）

概念编号	Synset	Csynset	上位概念	下位概念	Definition	Cdefinition
07331418	husband hubby married_man	丈夫 先生 夫君 夫婿 爱人 老公 郎君 驸马 驸马爷	07602853	07109482071959680725726073280 08	a married man; a woman’s partner in marriage	已婚男子；婚姻中女性一方的伴侣

表示身份

概念编号	Synset	Csynset	上位概念	下位概念	Definition	Cdefinition
07414666	Mister Mr.	先生 师傅 同志 大哥 老兄 老弟	07391044		a form of address for a man	对男子的一种称呼

表示关系

上述词义表示的问题

- **差异性如何表示**
  - 教师、教员、老师、先生、导师
- **新词如何表示**
  - 园丁、臭老九…
- **过于主观**
  - 不同的人（专家）构建的表示会有差异
- **需要大量的人工工作量**
  - **WordNet**从上世纪**80**年代构建，目前仍在完善中
- **很难计算词之间的相似度（或差异度）**

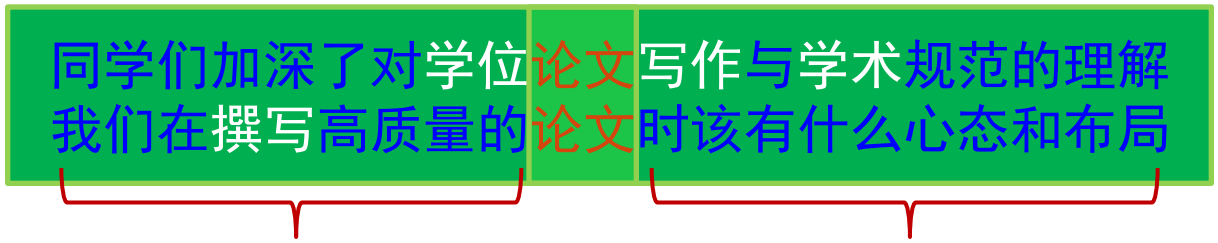
上述表示问题（续）

- 将词作为**原子符号**表示
  - 教师、教员、老师、先生、导师
- 以向量的形式表示，在高维向量空间中只有1个分量为1，其余的均为0:
  - [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
- 这种表示也称为独热（ one-hot ）向量，一个典型的问题是相似度和差异性难计算:
  - 教师 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] and
    老师 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] = ϕ

问题：①语义鸿沟②维度灾难③未知词

分布相似性

- **定义: 分布表示**
  - 一个词的词义可以通过周边词（的词义）表示

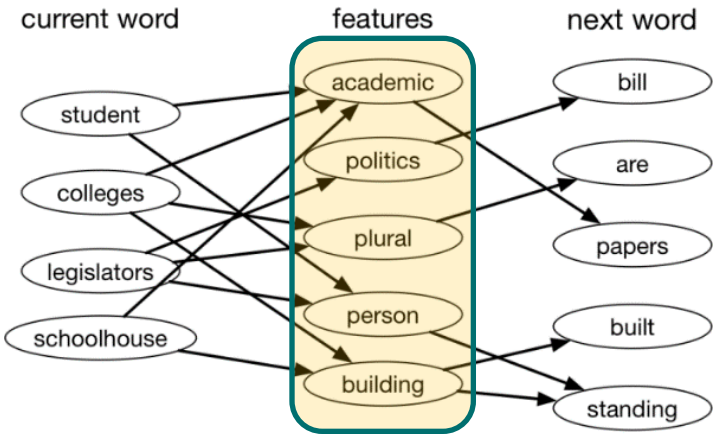


- **如何表示“论文”的含义**
  - 可以由其周围的词表示其意义（英语阅读理解的Cloze）

分布式假设：词义由上下文确定，如果两个词的上下文总相似，这两个词的含义也大致相近

分布表示

- 词之间具有相关性，这种相关性使得不同的词可以共享彼此间的信息，如：



- 即：给定词的信息分布，在表示中称之为分布表示
- 在实际情况中，并不一定能将每一维都附上一个有意义的标签（词）

词向量

以分布方式并通过向量表示词– Word Embedding

- 前面的词向量和One-hot是特殊的词向量表示
- 以神经网络学到的词表示通常就是现在的 word embedding 或词向量：
  - 利用周边词**预测当前词**，来学习词的表示
  - 以向量形式表示词 —— 词向量（word vector）

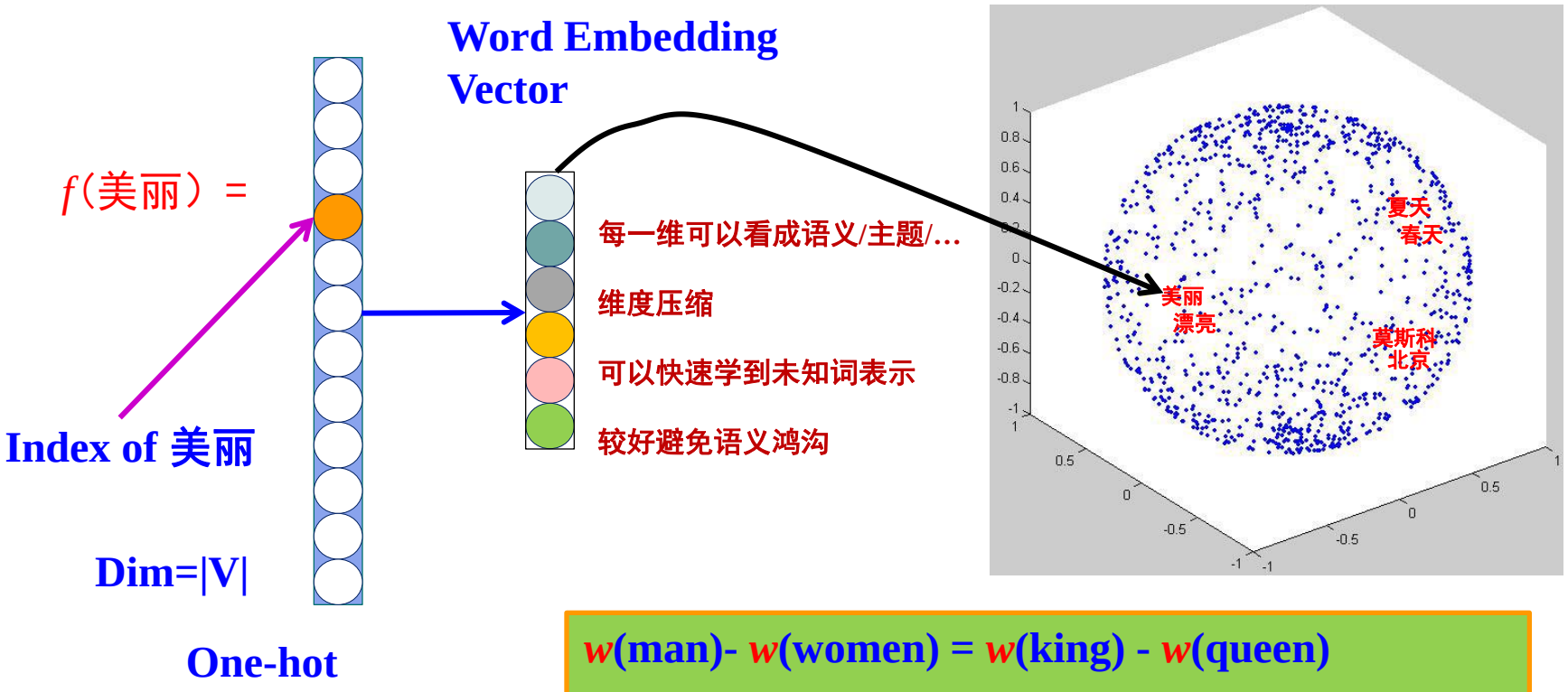
起源

- Distributed representation的思想最早出现在1986年Hinton的论文《Learning distributed representations of concepts》
- 2001年，Bengio 等人正式提出神经网络语言模型（Neural Network Language Model，NNLM）得到了词向量
- 目前已有多种学习方法（如CBOW & Skip-gram）

词向量的特点

❖ 词嵌入

- 前面的词向量和One-hot是特殊的词向量表示



w(man)- *w*(women) = *w*(king) - *w*(queen)

w(England) - *w*(Landon) = *w*(France) - *w*(Paris)

词向量的特点

- **词向量相关的语言学特性：**
  - 词在向量空间距离相近，可实现语义相关性的任务。如，同义词检测、单词类比等
- **词向量作为静态特征：**
  - 用词向量作为模型输入，在模型训练中，只调整模型参数，不调整 输入的词向量。如，基于平均词向量的文本分类
- **将词向量作为动态初值（输入）（浅层表示模型）**
  - 使用词向量作为神经网络的初始值，模型训练过程中会调整词向量的初值，从而提升神经网络模型的优化效果。 如，基于卷积神经网络的文本分类

经典分布式表示

名称	上下文	上下文与目标词之间的建模（技术手段）
LSA/LSI	文档	矩阵
HAL	词	
GloVe	词	
Jones & Mewhort	n-gram	
Brown Clustering	词	聚类
Skip-gram	词	神经网络
CBOW	n-gram（加权）	
Order	n-gram（线性组合）	
LBL	n-gram（线性组合）	
NNLM	n-gram（非线性组合）	
C&W	n-gram（非线性组合）	

基于计算的分布式表示

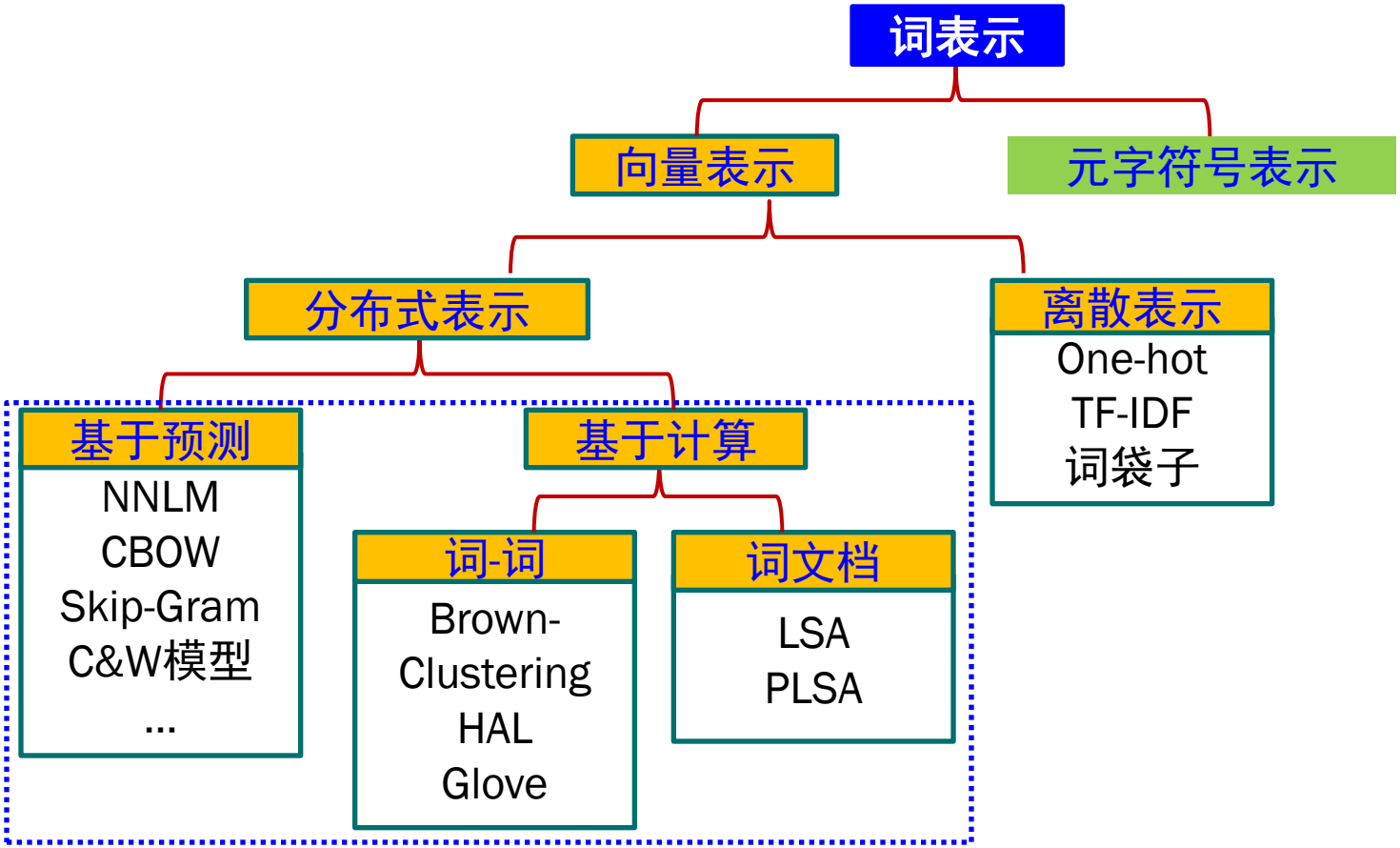
利用全部上下文或利用一定窗口内上下文词捕获语法和语义信息

局限性：空间消耗过大、稀疏等问题，需用降维法（如，SVD分解的方法）构造低维稠密向量作为词的分布式表示 (25~1000维)。与深度学习模型框架差异大。

基于预测的分布式表示
(神经网络词向量)

不计算词之间的共现频度，直接用“基于上下文词来预测当前词”或“基于当前词预测上下文词”的方法构造构造低维稠密向量作为词的分布式表示 。

经典词表示类型


词表示
向量表示
元字符号表示
分布式表示
离散表示
One-hot
TF-IDF
词袋子
基于预测
NNLM
CBOW
Skip-Gram
C&W模型
...
基于计算
词-词
Brown-
Clustering
HAL
Glove
词文档
LSA
PLSA

主要内容

- 概述
- 词的表示
- 词表示的统计计算方法
- 词表示的训练方法

基于窗口的同现矩阵

- 窗口长度=1 (常见情况是: 5 ~ 10)
  - 有如下数据:
    - I like deep learning.
    - I like NLP.
    - I enjoy flying.

count	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
Like	2	0	0	1	0	1	0	0
Enjoy	1	0	0	0	0	0	1	0
Deep	0	1	0	0	1	0	0	0
Learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

更大的窗口

整个句子全考虑

D1: dogs are animals

D2: cats are animals

D3: orchids are plants

D4: roses are plants

	animals	are	cats	dogs	orchids	plants	roses
animals		X	X	X			
are	X		X	X	X	X	X
cats	X	X					
dogs	X	X					
orchids		X				X	
plants		X			X		X
roses		X				X	

聚合关系(相似性)

聚合对

(dogs, cats)

(orchids, roses)

	animals	are	cats	dogs	orchids	plants	roses
animals		X	X	X			
are	X		X	X	X	X	X
cats	X	X					
dogs	X	X					
orchids		X				X	
plants		X			X		X
roses		X				X	

词-文档共现矩阵

- 词是否出现在文档中？
  - 矩阵A： m 行对应 m 个词， n 列对应 n 个文档

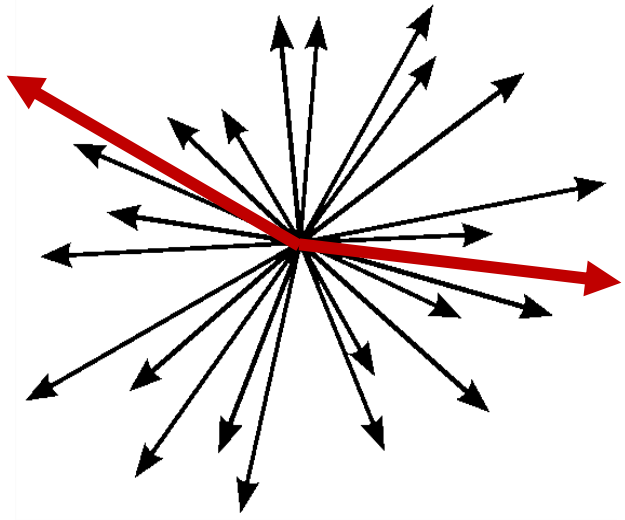
A=
9 x 8

Word\document	D1	D2	D3	D4	D5	D6	D7	D8
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
Transfer-function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1

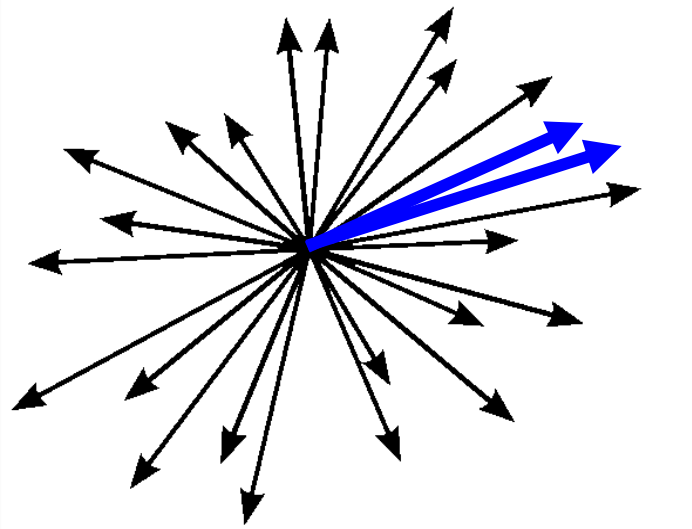
文档向量空间

- 矩阵的每个列表示一个文档：文档向量
- 相关度/相似度可以用于度量文档之间的相关/相似性

两个不相似的文档

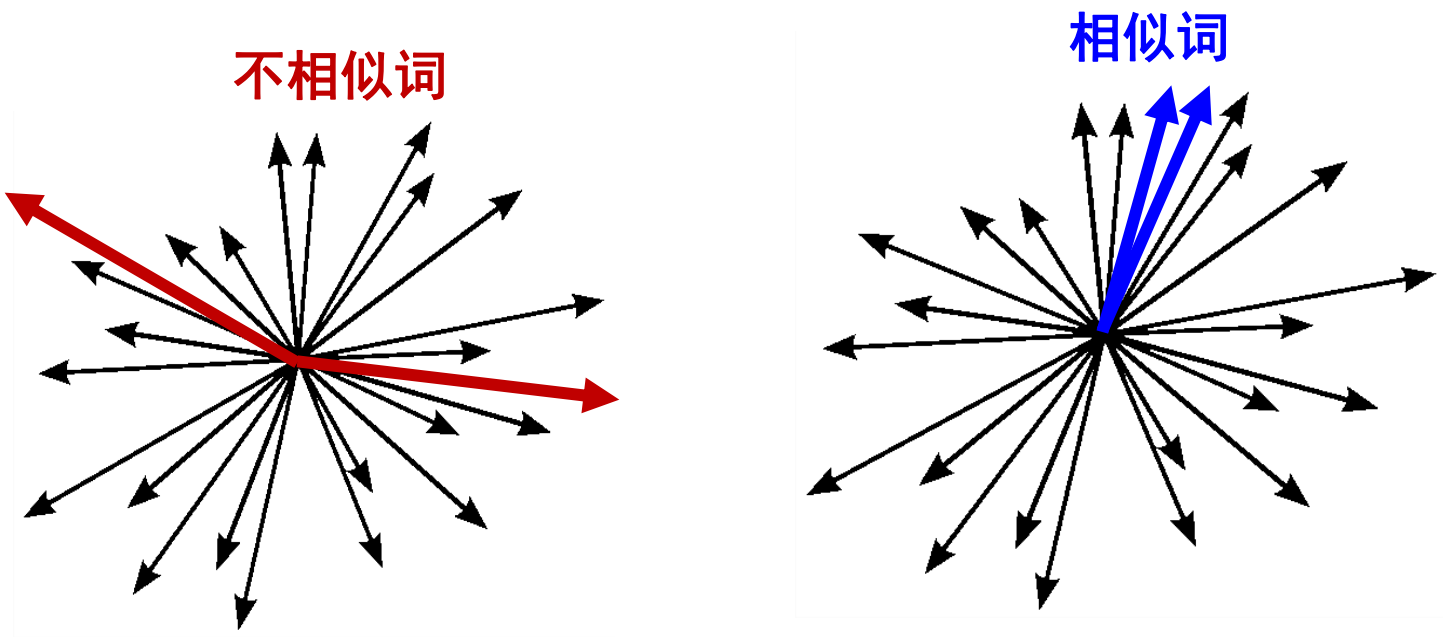


两个相似的文档



词向量空间

- 矩阵中的每一行表示一个词：词向量
- 相关度/相似度可以用于度量词之间的相关/相似性



隐性语义标引(LSI)

- **关键思想：**并不是按词条的数目形成空间维度来表示文档
  - 而是以一种低维的空间表示
  - 每一维代表概念（由近义词集形成概念）
- **例子**
  - **Car, automobile, driver**, elephant
  - 加入查询词是 **car**， 应该能查询到 **driver** 和 **automobile**, 但不应该查到 **elephants**

通过奇异值分解（SVD）实现LSI

矩阵 A 能够奇异值分解为3个矩阵:

$$A = U\Sigma V^T$$

U 是AA^T 的正交特征向量

V^T 是A^TA 的正交特征向量

$\lambda_1 \ldots \lambda_r$ 是 AA^T 的特征值，也是 A^TA 的特征值

Σ 时 $r \times r$ 的奇异值对角矩阵. r 是 A的秩

$$\Sigma = diag(\sigma_1 \ldots \sigma_r)$$

Singular values → $\sigma_i = \sqrt{\lambda_i}$

例子

9 X 8（词-文档矩阵）

term	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
transfer function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1

该矩阵的秩为**7**，因此，需要**7**维表示

U (9x7) =

0.3996	-0.1037	0.5606	-0.3717	-0.3919	-0.3482	0.1029
0.4180	-0.0641	0.4878	0.1566	0.5771	0.1981	-0.1094
0.3464	-0.4422	-0.3997	-0.5142	0.2787	0.0102	-0.2857
0.1888	0.4615	0.0049	-0.0279	-0.2087	0.4193	-0.6629
0.3602	0.3776	-0.0914	0.1596	-0.2045	-0.3701	-0.1023
0.4075	0.3622	-0.3657	-0.2684	-0.0174	0.2711	0.5676
0.2750	0.1667	-0.1303	0.4376	0.3844	-0.3066	0.1230
0.2259	-0.3096	-0.3579	0.3127	-0.2406	-0.3122	-0.2611
0.2958	-0.4232	0.0277	0.4305	-0.3800	0.5114	0.2010

Σ (7x7) =

3.9901	0	0	0	0	0	0
0	2.2813	0	0	0	0	0
0	0	1.6705	0	0	0	0
0	0	0	1.3522	0	0	0
0	0	0	0	1.1818	0	0
0	0	0	0	0	0.6623	0
0	0	0	0	0	0	0.6487

V (8x7) =

0.2917	-0.2674	0.3883	-0.5393	0.3926	-0.2112	-0.4505
0.3399	0.4811	0.0649	-0.3760	-0.6959	-0.0421	-0.1462
0.1889	-0.0351	-0.4582	-0.5788	0.2211	0.4247	0.4346
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000
0.6838	-0.1913	-0.1609	0.2535	0.0050	-0.5229	0.3636
0.4134	0.5716	-0.0566	0.3383	0.4493	0.3198	-0.2839
0.2176	-0.5151	-0.4369	0.1694	-0.2893	0.3161	-0.5330
0.2791	-0.2591	0.6442	0.1593	-0.1648	0.5455	0.2998

$$A = U\Sigma V^T$$

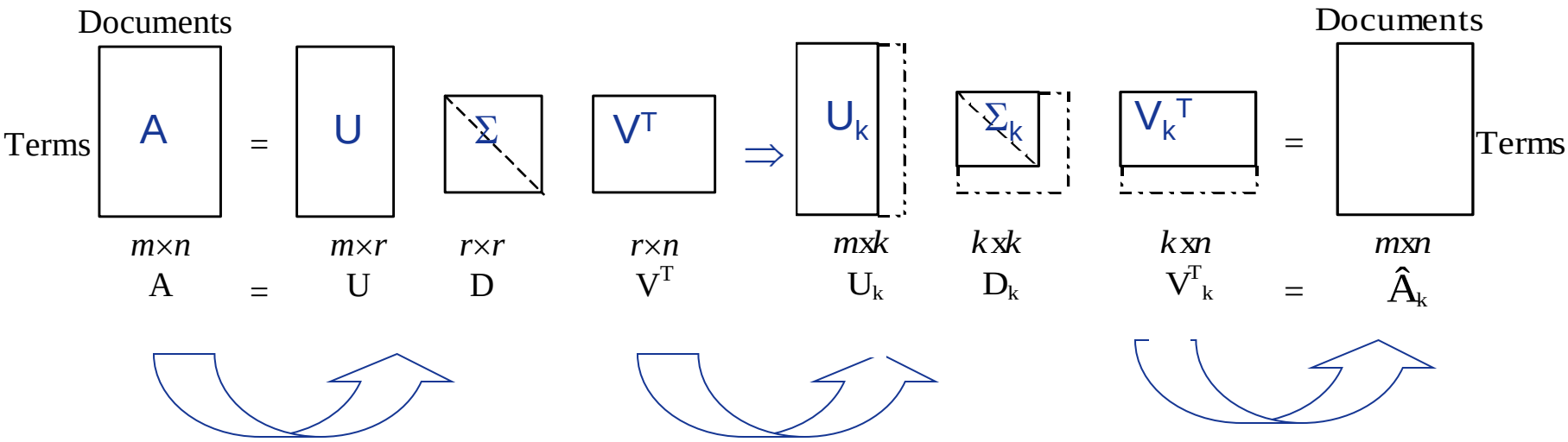
LSI 降维

- 对于对角矩阵 Σ，只选择 k 个最大的奇异值
- 保留 U 和 V^T 和对应的 k 列

类似于主成分分析

- 降维后的结果矩阵 A_k 表示为：
  - $A_k = U_k \Sigma_k V^T_k$
  - 其中 k（k < r）是概念集的大小（降维后的维度）
- 对于参数 k：
  - 足够大，以尽量拟合数据自身的特点
  - 同时也应足够小，以滤掉无关紧要的细节

再看 LSI



Singular Value Decomposition (SVD):
将原矩阵 A 分解为 3个矩阵 U, Σ 和 V

降维：去掉低阶的行和列

重构矩阵 A:
通过3个矩阵相乘近似还原原矩阵

使用新的矩阵实施查询操作

再看例子

9 X 8

term	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8
controllability	1	1	0	0	1	0	0	1
observability	1	0	0	0	1	1	0	1
realization	1	0	1	0	1	0	1	0
feedback	0	1	0	0	0	1	0	0
controller	0	1	0	0	1	1	0	0
observer	0	1	1	0	1	1	0	0
transfer function	0	0	0	0	1	1	0	0
polynomial	0	0	0	0	1	0	1	0
matrices	0	0	0	0	1	0	1	1

U (9x7) =

0.3996	-0.1037	0.5606	-0.3717	-0.3919	-0.3482	0.1029
0.4180	-0.0641	0.4878	0.1566	0.5771	0.1981	-0.1094
0.3464	-0.4422	-0.3997	-0.5142	0.2787	0.0102	-0.2857
0.1888	0.4615	0.0049	-0.0279	-0.2087	0.4193	-0.6629
0.3602	0.3776	-0.0914	0.1596	-0.2045	-0.3701	-0.1023
0.4075	0.3622	-0.3657	-0.2684	-0.0174	0.2711	0.5676
0.2750	0.1667	-0.1303	0.4376	0.3844	-0.3066	0.1230
0.2259	-0.3096	-0.3579	0.3127	-0.2406	-0.3122	-0.2611
0.2958	-0.4232	0.0277	0.4305	-0.3800	0.5114	0.2010

Σ (7x7) =

3.9901	0	0	0	0	0	0
0	2.2813	0	0	0	0	0
0	0	1.6705	0	0	0	0
0	0	0	1.3522	0	0	0
0	0	0	0	1.1818	0	0
0	0	0	0	0	0.6623	0
0	0	0	0	0	0	0.6487

V (8x7) =

0.2917	-0.2674	0.3883	-0.5393	0.3926	-0.2112	-0.4505
0.3399	0.4811	0.0649	-0.3760	-0.6959	-0.0421	-0.1462
0.1889	-0.0351	-0.4582	-0.5788	0.2211	0.4247	0.4346
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000
0.6838	-0.1913	-0.1609	0.2535	0.0050	-0.5229	0.3636
0.4134	0.5716	-0.0566	0.3383	0.4493	0.3198	-0.2839
0.2176	-0.5151	-0.4369	0.1694	-0.2893	0.3161	-0.5330
0.2791	-0.2591	0.6442	0.1593	-0.1648	0.5455	0.2998

该矩阵的秩为**7**，因此，需要**7**维表示

$$A = U\Sigma V^T$$

降维到 k = 2 的矩阵

U (9x7) =

0.3996	-0.1037	0.5606	-0.3717	-0.3919	-0.3482	0.1029
0.4180	-0.0641	0.4878	0.1566	0.5771	0.1981	-0.1094
0.3464	-0.4422	-0.3997	-0.5142	0.2787	0.0102	-0.2857
0.1888	0.4615	0.0049	-0.0279	-0.2087	0.4193	-0.6629
0.3602	0.3776	-0.0914	0.1596	-0.2045	-0.3701	-0.1023
0.4075	0.3622	-0.3657	-0.2684	-0.0174	0.2711	0.5676
0.2750	0.1667	-0.1303	0.4376	0.3844	-0.3066	0.1230
0.2259	-0.3096	-0.3579	0.3127	-0.2406	-0.3122	-0.2611
0.2958	-0.4232	0.0277	0.4305	-0.3800	0.5114	0.2010

Σ (7x7) =

3.9901	0	0	0	0	0	0
0	**2.2813**	0	0	0	0	0
0	0	1.6705	0	0	0	0
0	0	0	1.3522	0	0	0
0	0	0	0	1.1818	0	0
0	0	0	0	0	0.6623	0
0	0	0	0	0	0	0.6487

V (7x8) =

0.2917	-0.2674	0.3883	-0.5393	0.3926	-0.2112	-0.4505
0.3399	0.4811	0.0649	-0.3760	-0.6959	-0.0421	-0.1462
0.1889	-0.0351	-0.4582	-0.5788	0.2211	0.4247	0.4346
-0.0000	-0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000
0.6838	-0.1913	-0.1609	0.2535	0.0050	-0.5229	0.3636
0.4134	0.5716	-0.0566	0.3383	0.4493	0.3198	-0.2839
0.2176	-0.5151	-0.4369	0.1694	-0.2893	0.3161	-0.5330
0.2791	-0.2591	0.6442	0.1593	-0.1648	0.5455	0.2998

U2* Σ 2*V2 will be a 9x8 matrix
That approximates original matrix

U2 (9x2) =

0.3996	-0.1037
0.4180	-0.0641
0.3464	-0.4422
0.1888	0.4615
0.3602	0.3776
0.4075	0.3622
0.2750	0.1667
0.2259	-0.3096
0.2958	-0.4232

Σ 2 (2x2) =

3.9901	0
0	2.2813

V2 (8x2) =

0.2917	-0.2674
0.3399	0.4811
0.1889	-0.0351
-0.0000	-0.0000
0.6838	-0.1913
0.4134	0.5716
0.2176	-0.5151
0.2791	-0.2591

查询

查询词为： *feedback controller*,查询向量为转置

q = **[0 0 0 1 1 0 0 0 0]^T**

假定 **q** 是查询向量. 其对应的文档空间向量便为：

$$q^T*U2*inv(\Sigma 2) = D_q$$

即：

$$D_q = [\,0.1376\,,\ 0.3678]$$

为了发现最匹配的文档, 可以比较 D_q 与所有**2维**文档向量， 方向越一致，便越匹配。最简单的方法是计算余弦值 **cosine** 。以上面为例，8个余弦值分别是:

-0.3747 0.9671 0.1735 -0.9413 0.0851 0.9642 -0.7265 -0.3805

U2 (9x2) =

0.3996	-0.1037
0.4180	-0.0641
0.3464	-0.4422
0.1888	0.4615
0.3602	0.3776
0.4075	0.3622
0.2750	0.1667
0.2259	-0.3096
0.2958	-0.4232

Σ2 (2x2) =

3.9901	0
0	2.2813

V2 (8x2) =

0.2917	-0.2674
0.3399	**0.4811**
0.1889	-0.0351
-0.0000	-0.0000
0.6838	-0.1913
0.4134	**0.5716**
0.2176	-0.5151
0.2791	-0.2591

	term	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8
0	controllability	1	1	0	0	1	0	0	1
0	observability	1	0	0	0	1	1	0	1
0	realization	1	0	1	0	1	0	1	0
1	feedback	0	1	0	0	0	1	0	0
1	controller	0	1	0	0	1	1	0	0
0	observer	0	1	1	0	1	1	0	0
0	transfer function	0	0	0	0	1	1	0	0
0	polynomial	0	0	0	0	1	0	1	0
0	matrices	0	0	0	0	1	0	1	1
		-0.37	**0.967**	0.173	-0.94	0.08	**0.96**	-0.72	-0.38

LSI 的优点

- LSI 的优点：
  - 降维
  - 降噪
  - 实现相关性计算
- LSI 能较好地处理同义词，但不能处理多义词
- 问题:
  - SVD 时间复杂度高：（$O(n^3)$）
  - 一旦新文档出现，便需要再做奇异值分解

Global Vector 模型（GloVe）

Glove介绍

- 全称：**Global Vectors for word representation**
- 基于全局词频统计（**count-based & overall statistics**）的词表示
- 由**Jeffrey Pennington**和**Richard Socher & Christopher Manning**提出

基本思想：构建共现矩阵 X

- 第*i* 行第*j* 列的值为词w_i 与词w_j 的共现次数（常以次数的对数表示），共现在一定程度上表达了词间的关系
- 共现次数是在整个语料上统计的（不局限于某个句子），因此具有全局性
- 词的共现通常只考虑在一个固定窗口范围内，矩阵大小为 |V|x|V|
- 矩阵具有对称性

GloVe（续）

❖ 重要概念

- 共现矩阵 **X**： X_{ij} 表示X的第*i* 行第*j* 列的值，为词*i* 与词*j* 在语料固定窗口大小内的共现次数
  - $P_{ij}=X_{ij} / X_i$ (其中， $X_i= \sum_k X_{ik}$) 表示词 *j* 在词 *i* 的上下文中出现的概率
  - 共现概率比值： P_{ik} / P_{jk} 表示词 *k* 的意义与*i* 相近（如比值大于1）还是与 *j* 相近（如比值小于1）；当接近于1时表示都相关或都不相关。共现概率比值可以用于构建词向量

❖ 再看共现概率比值的意义

- 通过语料统计得到共现概率比值，可以作为标签，通过训练的方式让映射值逼近这个确定的共现概率比值——回归问题，用均方误差作为 loss，引入一个映射 f，使得三个词的词向量与共现概率比值之间满足如下关系：

$$f(v_i, v_j, \tilde{v}_k) = \frac{p_{ik}}{p_{jk}}$$

f 将向量变为标量

GloVe（续）

❖ 分析与推导

- 为了使 $f(v_i, v_j, \tilde{v}_k) = p_{ik} / p_{jk}$
- 可将左边变形为内积形式： $f(v_i, v_j, \tilde{v}_k) = f((v_i - v_j)^T \tilde{v}_k) = f((v_i^T \tilde{v}_k - v_j^T \tilde{v}_k)$
- 再进一步统一形式，变为 $f(v_i^T \tilde{v}_k - v_j^T \tilde{v}_k) = f(v_i^T \tilde{v}_k) / f(v_j^T \tilde{v}_k)$
- 满足上述式子成立的函数为 **exp**
- 于是，可以令： $\exp(v_i^T \tilde{v}_k) = p_{ik} = X_{ik} / X_i$，且 $\exp(v_j^T \tilde{v}_k) = p_{jk} = X_{jk} / X_j$
- 但上述式子有问题，因为 ***i*和*k*交换会有问题**： $\exp(v_k^T \tilde{v}_i) = p_{ki} = X_{ki} / X_k$

左边相等右边不等

- 为了避免上述问题，将上面式子两边取对数后，引入待定参数 **b**，于是有：

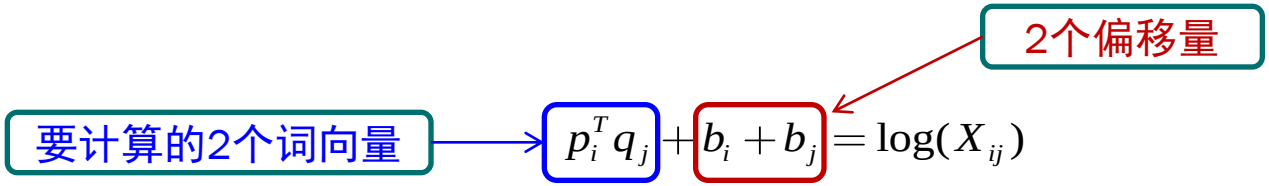
$$v_i^T \tilde{v}_k = \log X_{ik} - b_i - b_k \quad \Rightarrow \quad v_i^T \tilde{v}_k + b_i + b_k = \log X_{ik}$$

词向量的内积 + 偏置 接近于共现次数的对数（共现矩阵由共现次数的对数表示）

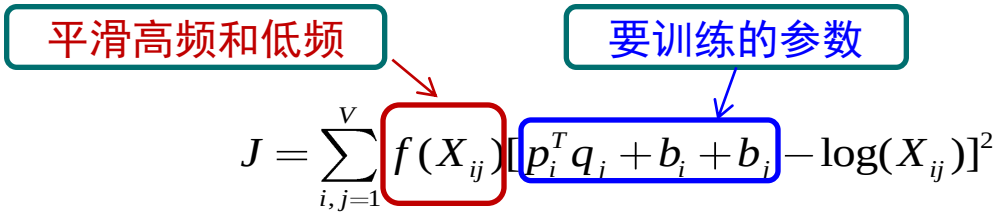
GloVe（续）

❖ 实现步骤

- 根据语料构建共现矩阵 X： X_{ij} 表示X的第*i* 行第*j* 列的值，为词 *i* 与词 *j* 在语料固定窗口大小内的带权共现和的对数
  - 权值=1/窗口内两个词 *i* 与词 *j* 的距离
- 其中：



- 损失函数



- 采用AdaGrad的梯度下降算法，对矩阵中的所有非零元素进行随机采样，学习率（learning rate）设为0.05，在vector size小于300的情况下迭代了50次，其他大小的vectors上迭代了100次

GloVe模型（续）

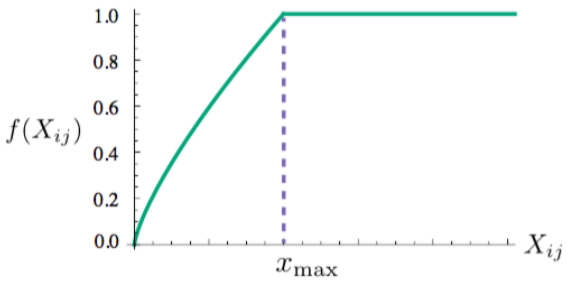
❖ 重构矩阵误差最小化项如下：

实验：GloVe略好于Word2Vec

$$\sum_{i,j\in \mathbf{V}, x_{ij}\neq 0} f(x_{ij})(\log(x_{ij}) - p_i^T q_j + b_i^{(1)} + b_i^{(2)})^2$$

其中，$\boldsymbol{p}_i$ 是词 w_i 作为目标词时的向量，$\boldsymbol{q}_j$ 是词 w_j 作为 w_i 的上下文时的词向量。$b^{(1)}$和 $b^{(2)}$是针对词表中各词的偏移向量。$f(x)$是一个加权函数，对低频的共现词对进行衰减，减少低频噪声带来的误差，定义为：

$$f(x) = \begin{cases} (x / x_{\max})^{\alpha} & if \ \ x < x_{\max} \\ 1 & others \end{cases}$$



主要内容

- 概述
- 词的表示
- 词表示的统计计算方法
- **词表示的训练方法**

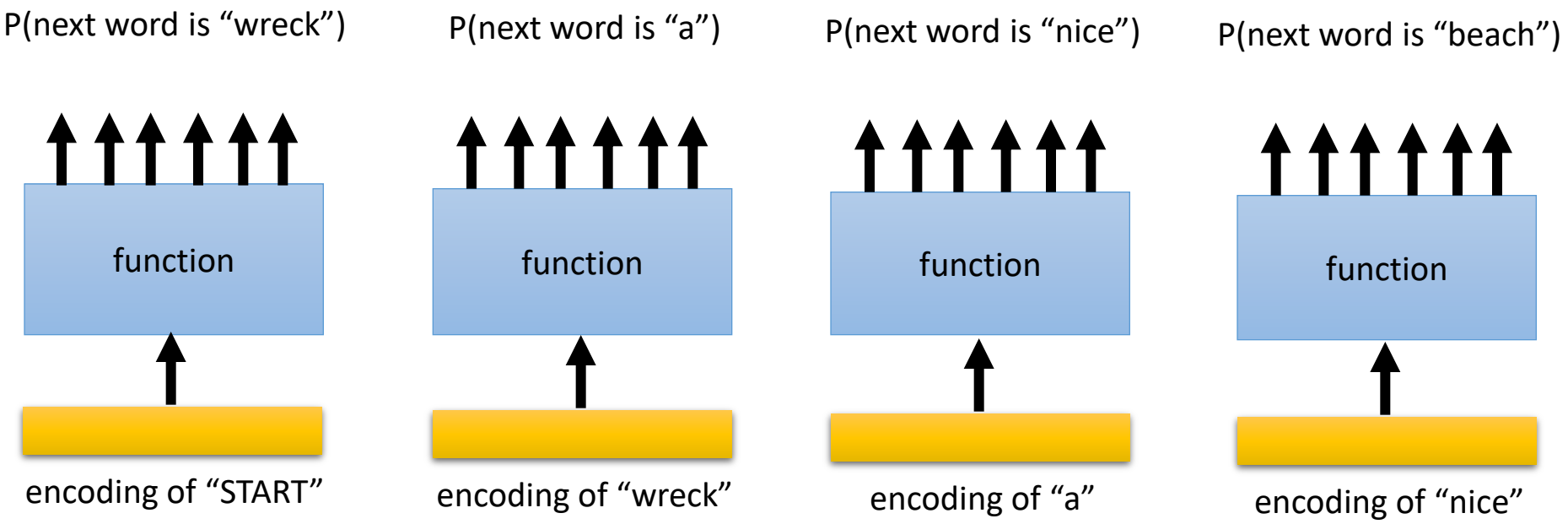
模型训练词向量

- **副产品**
  - 其他任务的中间环节（副产品）
  - 为其它任务服务
    - **Bengio 的Neural Network Language Model (NNLM)**
- **直接训练词向量**
  - 不考虑其它特殊任务
  - 直接训练获得词向量
    - **Mikolov 等的CBOW（Continuous Bag of Words）和Skip-gram 方法**

基于预测的语言模型

P("wreck a nice beach")
=P(wreck|START)P(a|wreck)P(nice|a)P(beach|nice)

P(b|a): 2-gram, 预测下一个词的概率



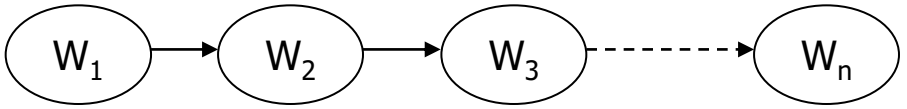
语言模型

- 什么是语言模型
  - ✓ 对语言现象进行数学抽象的一种形式化表示
  - ✓ 通过数学手段建模
- NLP常用的语言模型
  - ✓ 基于规则的模型
    - ➢ N.Chomky模型为代表的形式文法（如短语结构语法 PSG）
    - ➢ 特点：通过形式系统生成语言集合
  - ✓ 基于统计的模型
    - ➢ 主要用于判断语言的合理性：一个句子是否是合理的
      - ◘ 给出句子$s=w_1w_2\cdots w_n$，通过概率值的大小判断其合理程度
      - ◘ 例如：

P(老鼠爱大米) > P(老鼠大米爱)

N-gram 语言模型概率图表示

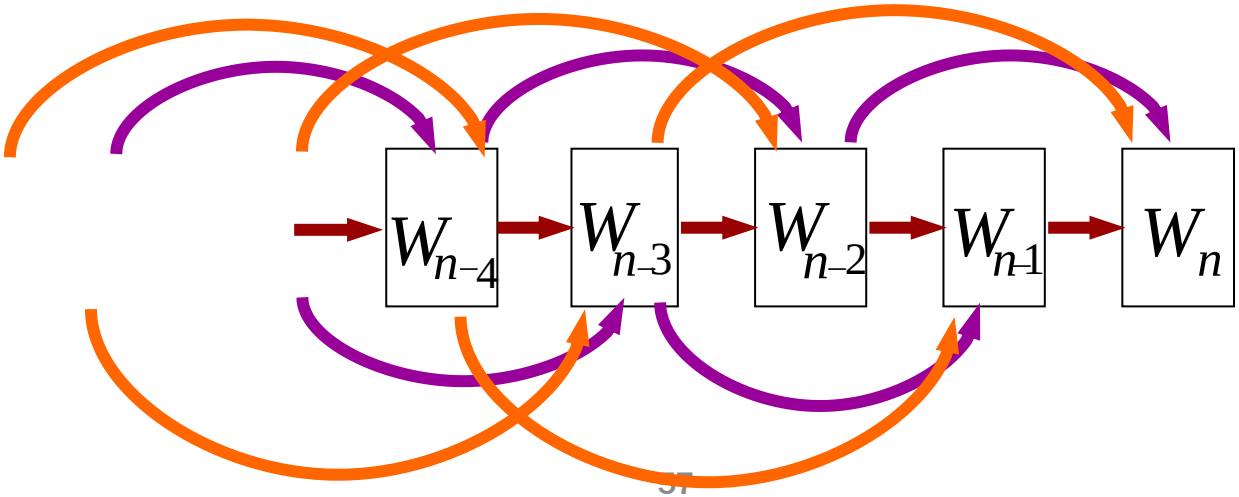
- bigram语言模型



$$P(w_1, w_2, \ldots, w_n) = P(w_1)\prod_{i=2}^{n} P(w_i \mid w_{i-1})$$

- 4-gram语言模型

$$P(w_1, w_2, \ldots, w_n) = P(w_3 \mid w_2, w_1)P(w_2 \mid w_1)P(w_1)\prod_{i=4}^{n} P(w_i \mid w_{i-1}, w_{i-2}, w_{i-3})$$



统计语言模型

- 统计语言模型引入
  - ✓ 通过概率大小度量句子的合理性
  - ✓ 链式表示形式
    - 设句子 $s = w_1 w_2 \ldots w_n$,
    - 其概率 $P(w_1 w_2 \ldots w_n) = P(w_1)\ P(w_2|w_1)\ \ldots P(w_n|w_1 \ldots w_{n-1})$
  - ✓ 条件概率的估计
    - $P(w_n|w_1 \ldots w_{n-1})$ 使用最大似然估计：

$$P(w_n \mid w_1, \ldots w_{n-1}) = \frac{C(w_1, \ldots w_{n-1}, w_n)}{C(w_1, \ldots w_{n-1})}$$

    - 计算任意条件长度概率 $P(w_i|w_1 \ldots w_{i-1})$ 的参数过多
    - 引入马尔可夫假设(Markov assumption)：假设当前词只依赖于前n-1个词—n-gram

$$P(w_1, w_2, \ldots w_N) = P(w_1)P(w_2 \mid w_1) \ldots \prod_{i=n}^{N} P(w_i \mid w_{i-(n-1)}, \ldots w_{i-1})$$

语言模型的应用价值

- 语言模型: 估计一个句子（词序列）的概率
  - 假定句子为: $w_1, w_2, w_3, \ldots, w_n$
  - 其概率表示为：$P(w_1, w_2, w_3, \ldots, w_n)$
- 应用-1: 句子生成（包括翻译）是不是人话
- 应用-2: 语音识别
  - 相同语音序列可能对应不同的词序列

recognize speech
or
wreck a nice beach

If P(recognize speech)
>P(wreck a nice beach)

Output =
“recognize speech”

基于统计的 N-gram

P("wreck a nice beach")
=P(wreck|START)P(a|wreck)
P(nice|a)P(beach|nice)

- 如何计算 **P("wreck a nice beach")**
- 收集大规模的语料
  - 可以简化为2-*gram* 语言模型: $P(w_1, w_2, w_3, \ldots, w_n) = P(w_1|START)P(w_2|w_1) \ldots\ldots P(w_n|w_{n-1})$
  - 只需从语料中估计类似 P(beach|nice) 的概率即可:

$$P(\text{beach}|\text{nice}) = \frac{C(nice\ beach)}{C(nice)}$$

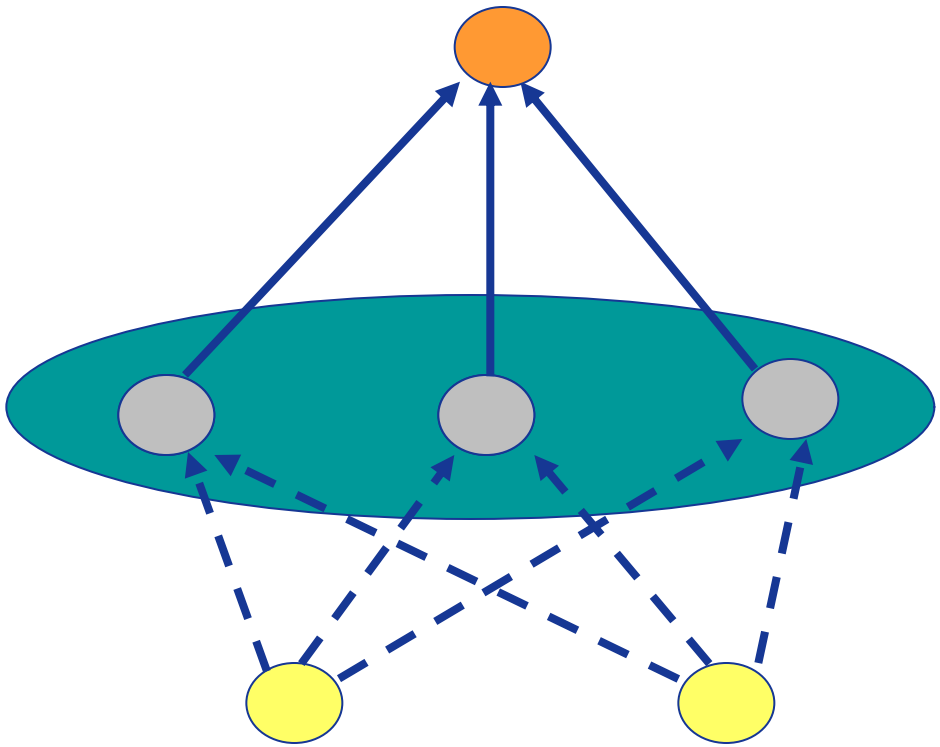
nice beach一起出现的次数

nice出现的次数

  - 也很容易推广到 3-gram, 4-gram ……

多层神经网络

- 简单的三层神经网络结构



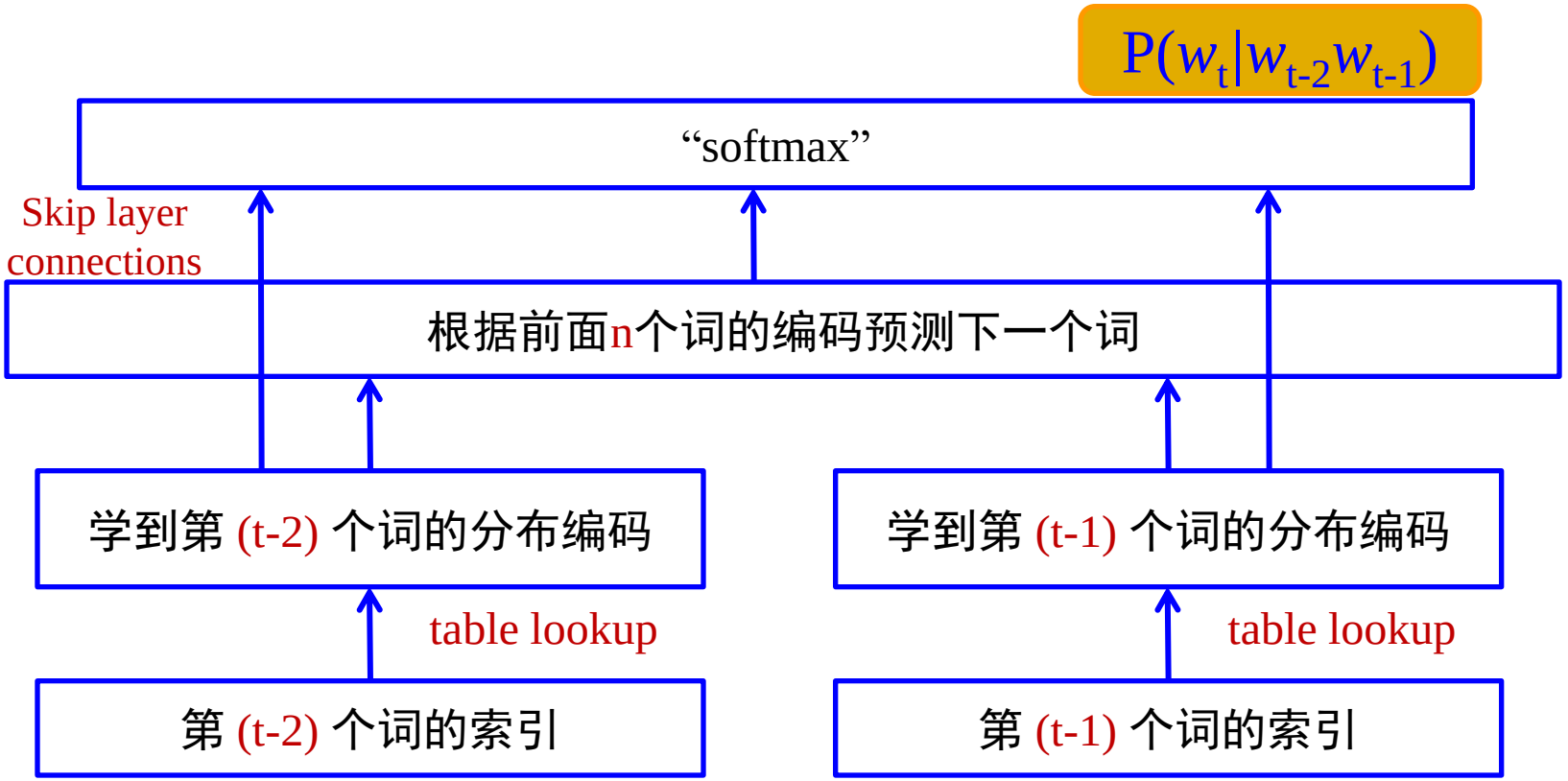
输出层

隐藏层：$H=(h_1,h_2,\ldots h_m)$

输入层：$X=(x_1,x_2,\ldots x_n)$

神经语言模型

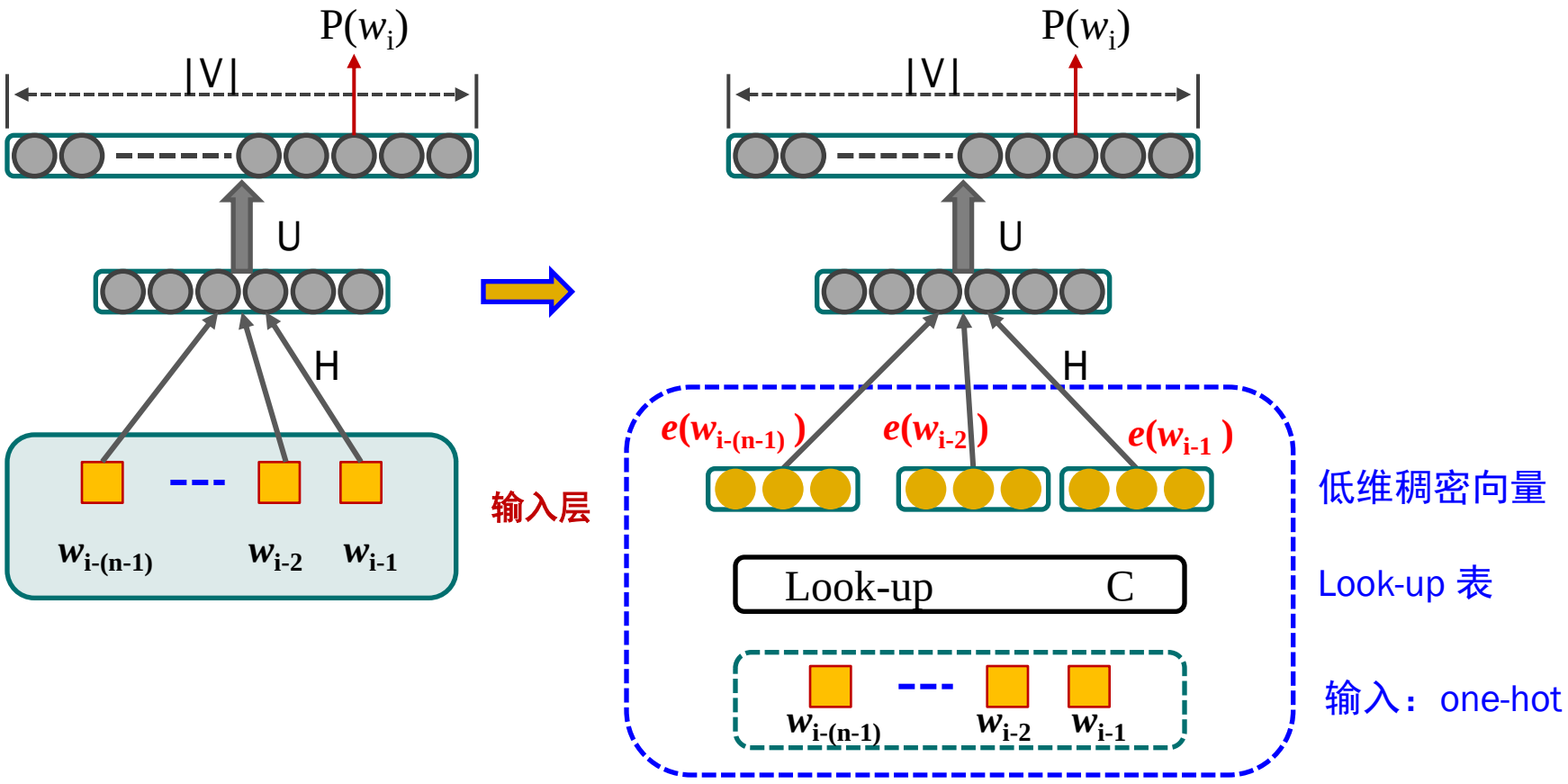
- 神经语言概率模型——神经语言模型:



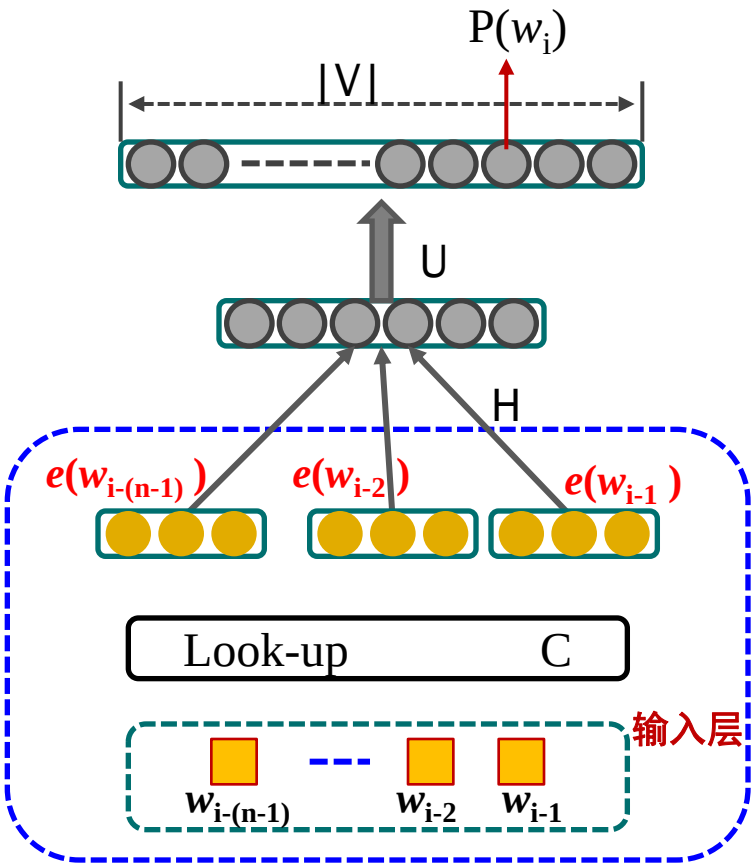
神经语言模型细解

NLM模型结构

N-gram 语言模型：$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$



NLM进一步解释



低维稠密向量

Look-up 表

输入：one-hot

Look-up 表

$$C = \begin{pmatrix} (w_1)_1 & (w_2)_1 & \cdots & (w_V)_1 \\ (w_1)_2 & (w_2)_2 & \cdots & (w_V)_2 \\ \vdots & \vdots & \ddots & \vdots \\ (w_1)_D & (w_2)_D & \cdots & (w_V)_D \end{pmatrix}$$

w_2 = [0 1 0... 0]

$e(w_2) = (w_2)_1\ (w_2)_2\ \ldots.(w_2)_D$

低维稠密向量：Look-up表 C （|D| x | V|维）投影矩阵，| V|表示词表的大小，|D| 表示词向量e的维度（一般50维以上）；各词的词向量存于C中。词w 到其词向量e(w) 的转化是从该矩阵中取出相应的列。

神经语言模型训练

- 给定了前面 K 个词，预测下一个词的问题属于多类分类问题
- **Inputs**: 前面 K 个词
- **Target**: 预测下一个词
- **Loss**: 可以看成为最大似然估计:

$$
\begin{aligned}
-log\ \mathrm{p}(\mathrm{s}) &= -log \prod_{t=1\sim T} \mathrm{p}(w_t | w_1, \ldots w_{t-1}) \\
&= -\sum_{t=1\sim T} log\ \mathrm{p}(w_t | w_1, \ldots w_{t-1}) \\
&= -\sum_{t=1\sim T} \sum_{v=1\sim V} E_{tv}\ log\ \mathrm{y}_{tv}
\end{aligned}
$$

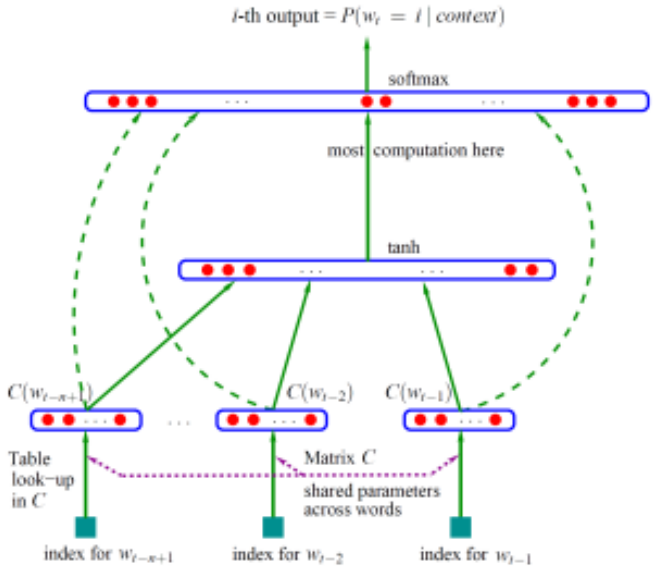
其中， E_{iv} 是第 i 个词的 one-hot 向量, y_{iv} 是第 i 个词的概率

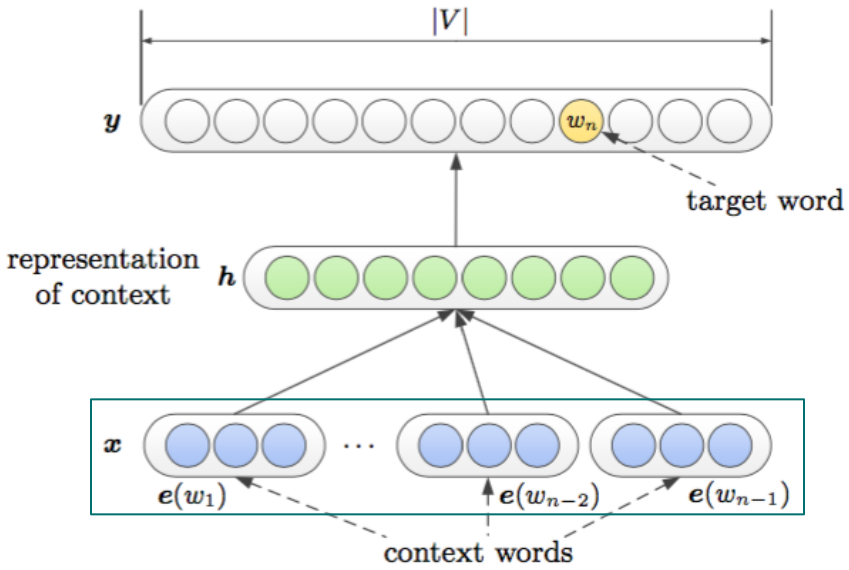
训练

- 损失函数： L
- 概率模型损失函数最常见的形式是： “负对数似然”
- 简单的随机梯度下降法优化参数：

$$\theta \leftarrow \theta - \eta * \partial L/\partial \theta$$

Bengio 的 NNLM方法





- **NNLM的基本思想：**
  - 输入层（记为 x）：用前面的 n−1个词w_{t-n+1},w_{t-n+2},w_{t-1}预测下一个词w_t，以词向量作为输入（向量首尾相连拼接作为输入，记为 x）；
  - 中间层（记为 h）：为 tanh($d + Hx$), d 是偏置项；
  - 最后层（记为 y）： $b + Wx + U$ tanh($d + Hx$)，U 是隐藏层的参数，W是直连边的权值，有助于减少迭代次数。

NNLM进一步解释

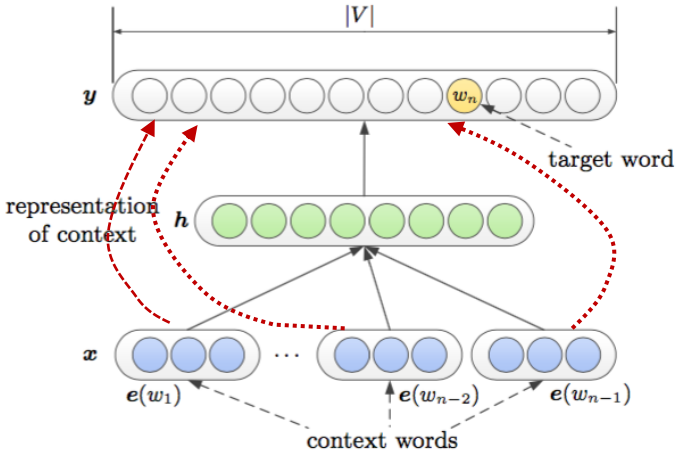
- 参数

训练参数：*C, U, W, H, b, d*

  - 训练语料：D
  - 词表为： $V = \{w_1, w_2, ..., w_{|V|}\}$
  - 输入时，每个词向量长度为 m，(n-1)个词的总长度为 m*(n-1)，也是(n-1)个拼接后的长度，对应m*(n-1)个神经元,每个词的向量通过查词向量矩阵 $C_{|V|\times m}$ 得到
  - 权重矩阵 $H_{h\times(n-1)m}$使输入进入隐层（h个隐单元）
  - 权重矩阵 $U_{|V|\times h}$ 使隐节点进入输出层（|V|个输出单元）
  - 权重矩阵 $W_{|V|\times(n-1)m}$ 使输入进入输出层（|V|个输出单元）
  - 设： $z = Wx + b + U(Hx + d)$

$$\hat{y}_i = P(w_i \mid w_{t-(n-1)}, w_{t-(n-2)}, ...w_{t-1}) = soft\max(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^{|V|} \exp z_k}$$

Bengio 方法与词向量

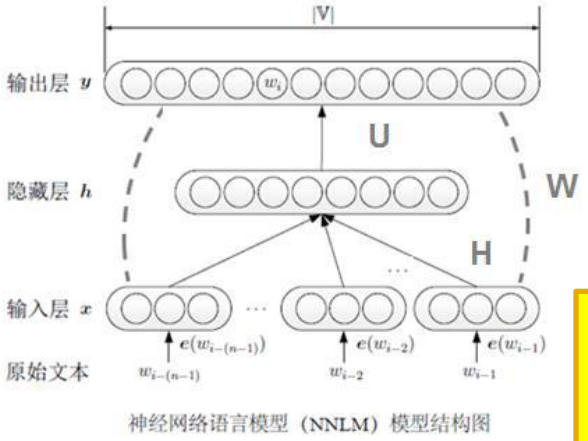


- **NNLM的基本思想：**
  - 主要是训练语言模型，词向量是副产品，服务于 NNLM的建模。
  - 输入的C矩阵需要优化，参数U矩阵（大小 $|V| \times h$）需要训练
  - 优化后的C就是对应的词向量

CBOW /skip-gram

- Mikolov等人在2013年，同时提出CBOW（Continuous Bag-of-Words）和Skip-gram模型。主要针对NNLM训练**复杂度过高的问题改进**，以更高效的方法获取词向量。

NLM



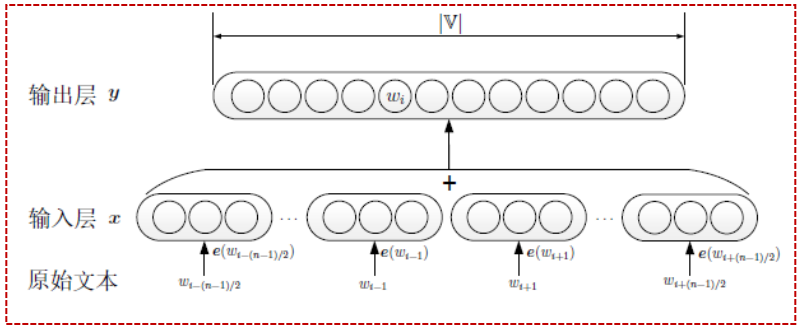
神经网络语言模型（NNLM）模型结构图

简化：

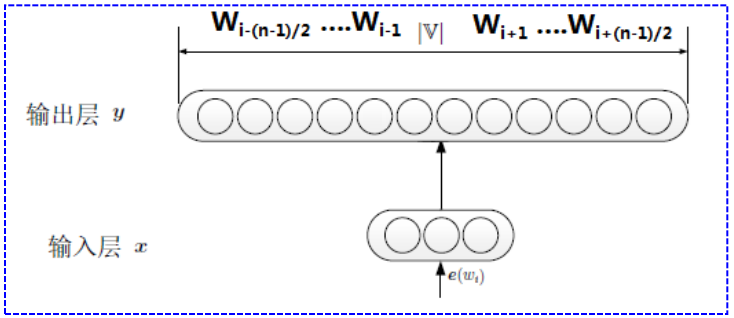
1.除去隐藏层

2.忽略词序

CBOW



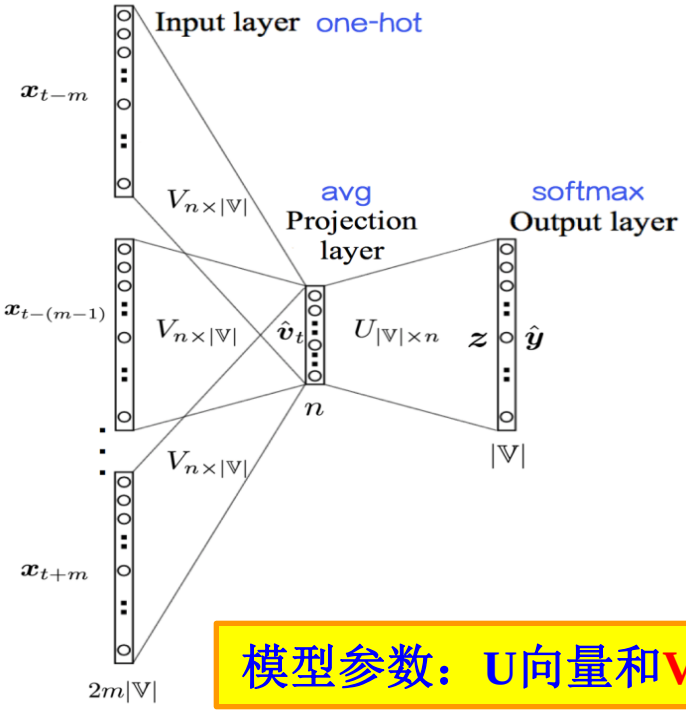
Skip-gram



CBOW(Continuous Bag-of-Words)

❖设置与符号含义：

- 词向量维度：n
- 输入：目标词w_t的左边 m个词和右边m个词（词向量的和取平均值）
- 投影层：n 个节点，上下文共 2m 个词的词向量的平均值
- 输出层：|V| 个节点。第 i个节点代表中心词是词 w_i 的概率



首先，将目标词w_t的上下文 $w_{t-m},...,w_{t-1},w_{t+1}...,w_{t+m}$

由one-hot表示的 x_{t+j} 转化为输入词向量 v_{t+j}：

$$v_{t+j}=Vx_{t+j},\quad j\in\{-m,...,m\}\setminus\{0\}$$

其次，投影。平均化输入向量 $v_{t-m},...,v_{t-1},v_{t+1}...,v_{t+m}$

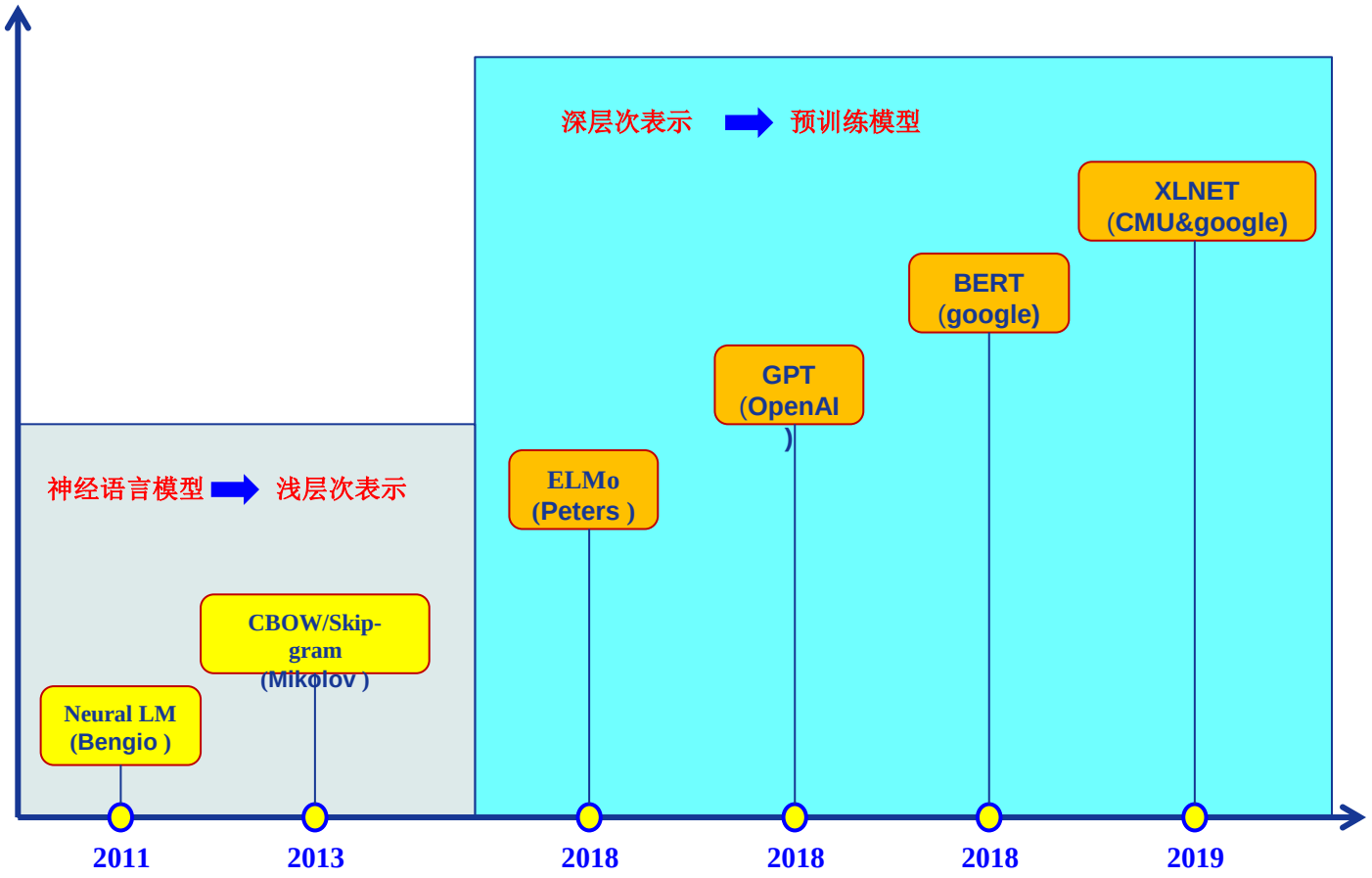
$$\hat{v}_t=\frac{1}{2m}\sum_j v_{t+j},\quad j\in\{-m,...,m\}\setminus\{0\}$$

最后，计算输出概率值：$z=U\hat{v}_t$

$$\hat{y}_i=P(w_i\mid w_{t-m},...,w_{t-1},w_{t+1},...,w_{t+m})=soft\max(z_i)=soft\max(u_i^T\hat{v}_t)$$

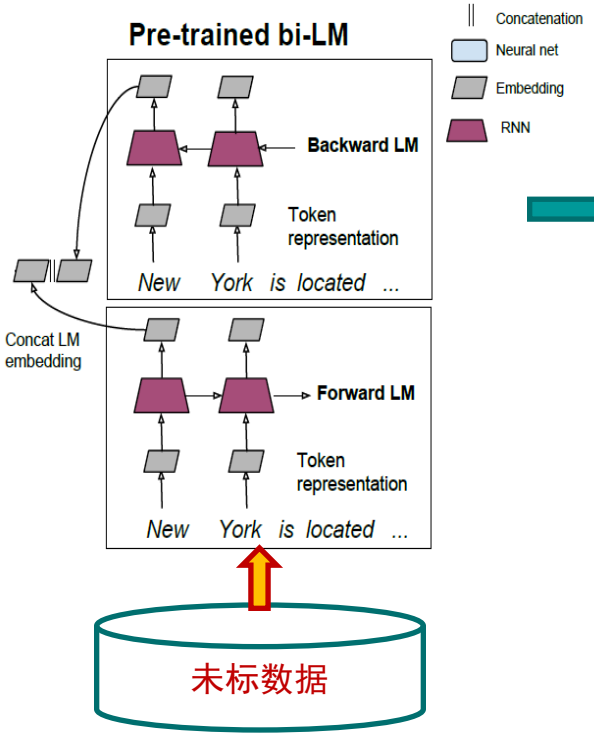
模型参数：U向量和V向量（随机初始化，训练结束后，V为词向量矩阵）

预训练语言模型与词的表示

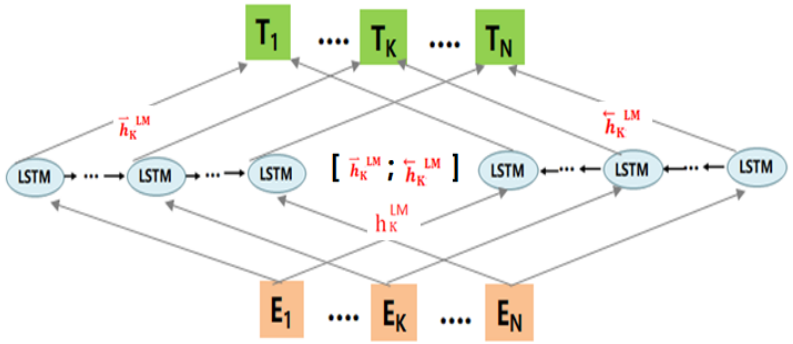


EMLo

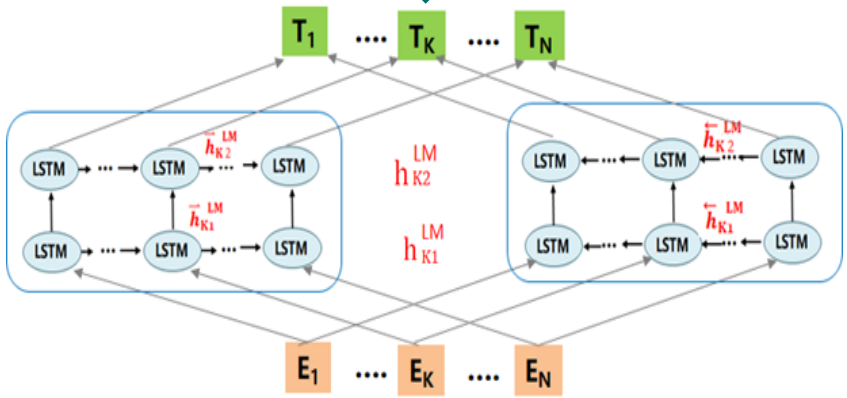
双向语言模型（源任务）



浅层语言模型：1层



深层语言模型：2层

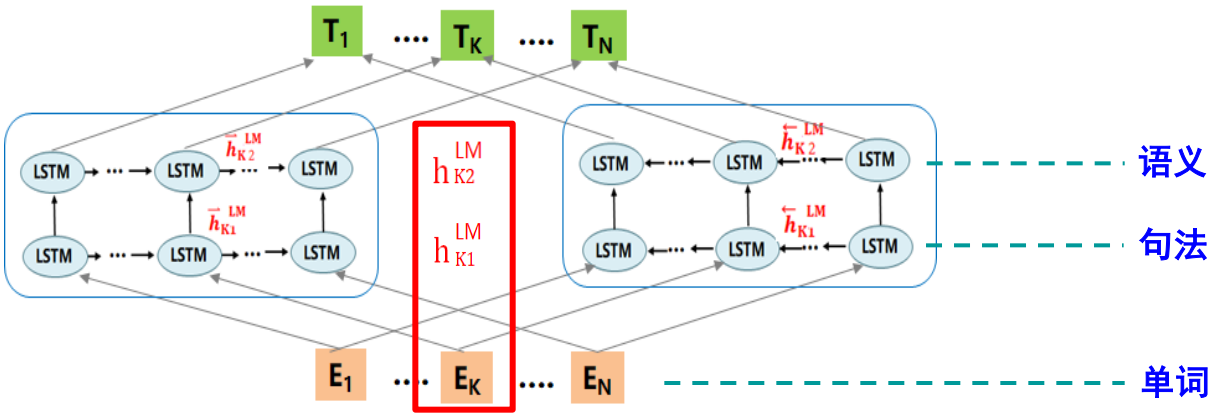


EMLo学到了什么

查词义

↓

计算词义



Glove训练出的play大致表达“运动”含义，可能因为训练数据中表示体育的语料更多

Play（多义词：运动/音乐）

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	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 .

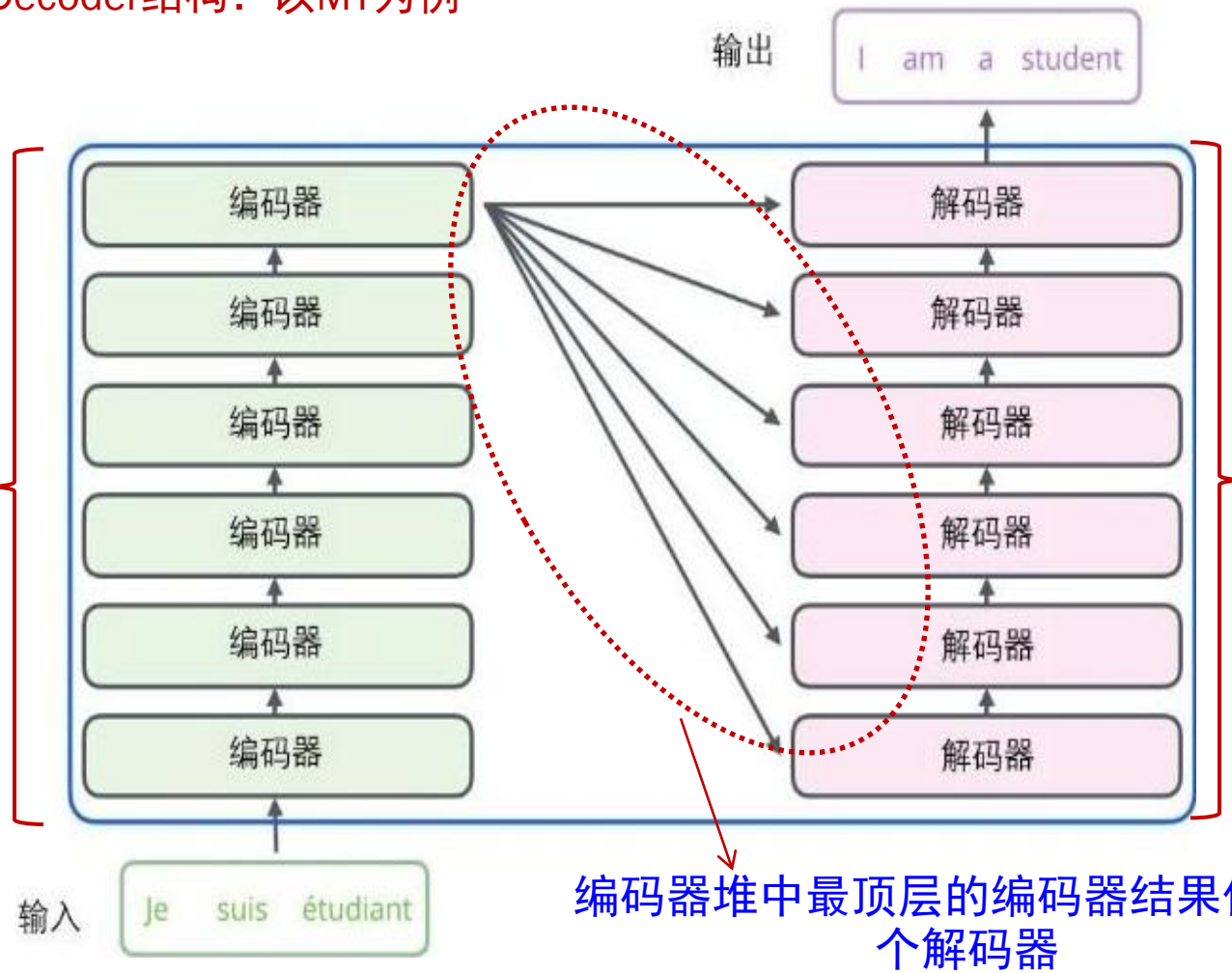
运动

音乐/演出

根据上下文获得词的信息（动态性）

Transformer的构架

Encoder-Decoder结构：以MT为例



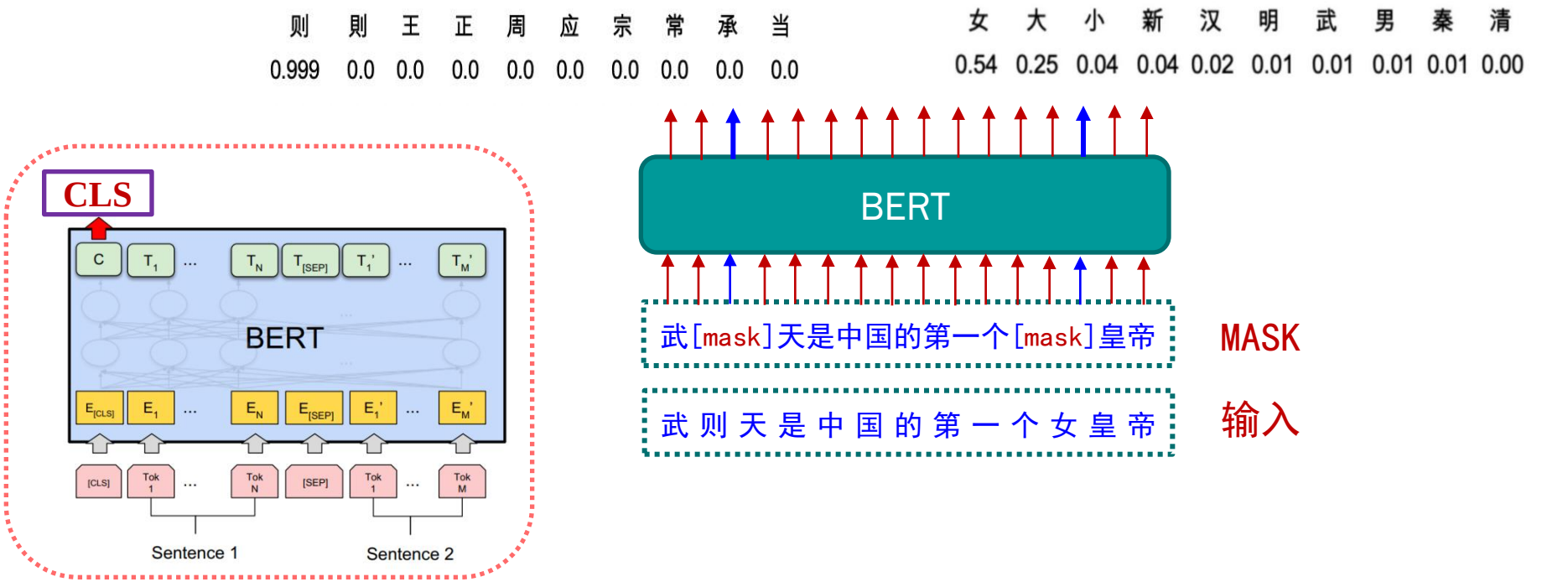
N(=6)个结构相同的编码器堆叠而成

M(=6)个结构相同的解码器堆叠而成

编码器堆中最顶层的编码器结果传给每个解码器

BERT 掩码语言模型MLM

输入端遮掩住某些位置的词，在输出端预测相应位置的词



Thanks