词表示

CS2916 大语言模型

饮水思源 爱国荣校

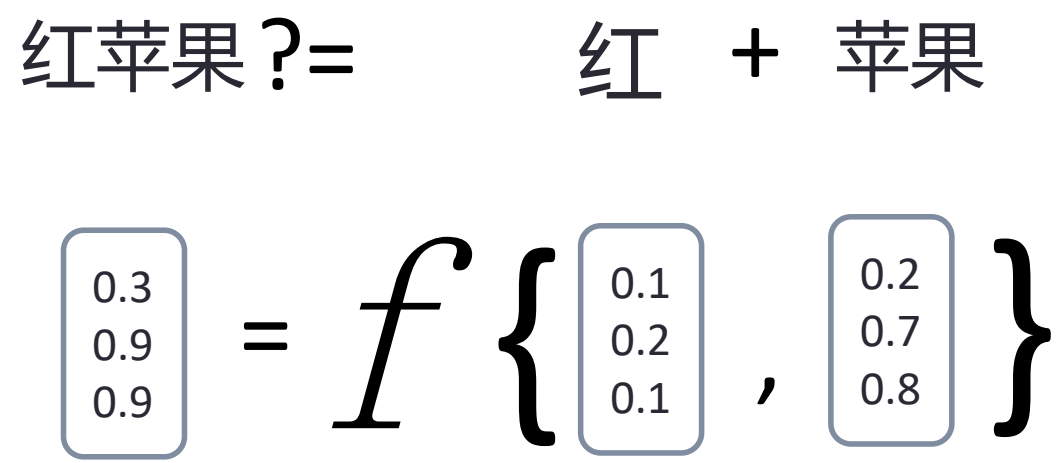
https://plms.ai/teaching/index.html

什么是词表示?

“苹果” ⟶ [0.1, 0.3. 0, 4]

为什么要学习词表示?

- 容易进行数学运算
  - 假如我们想计算“红苹果”的含义

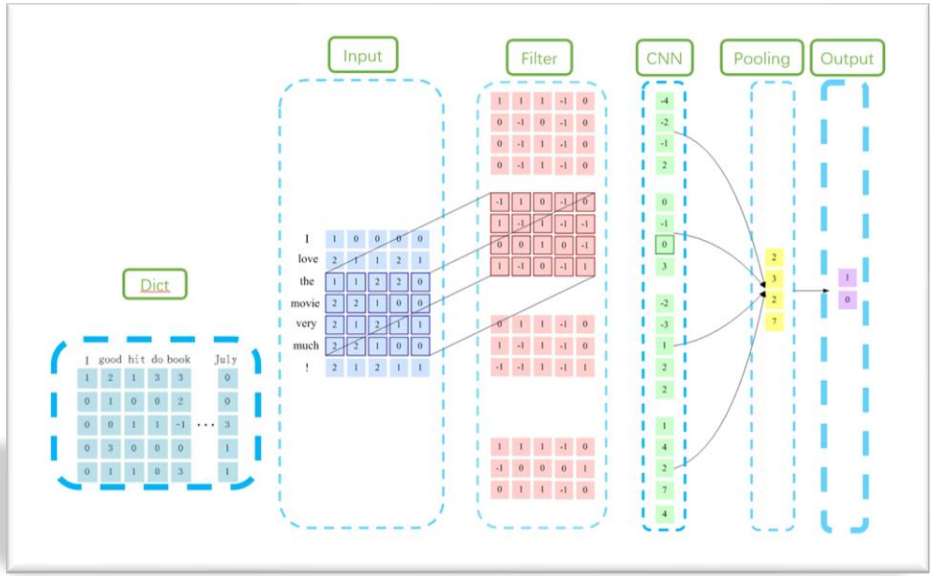


为什么要学习词表示?

- 容易进行数学运算
  - 假如我们想计算“红苹果”的含义

红苹果 ?= 红 + 苹果

$$\begin{bmatrix}0.3\\0.9\\0.9\end{bmatrix} = f\left\{\begin{bmatrix}0.1\\0.2\\0.1\end{bmatrix}, \begin{bmatrix}0.2\\0.7\\0.8\end{bmatrix}\right\}$$



词表示学习

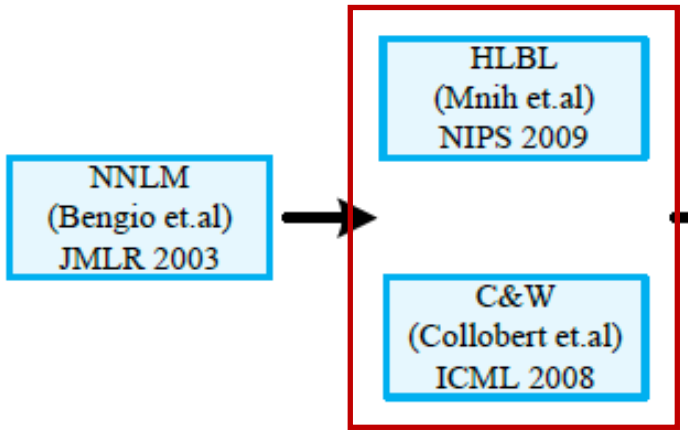
NNLM
(Bengio et.al)
JMLR 2003

词表示学习

- NNLM: 引入神经网络框架来训练语言模型

关注语言模型、词表示学习两个任务；抛出难题：如何提高NNLM训练效率

词表示学习



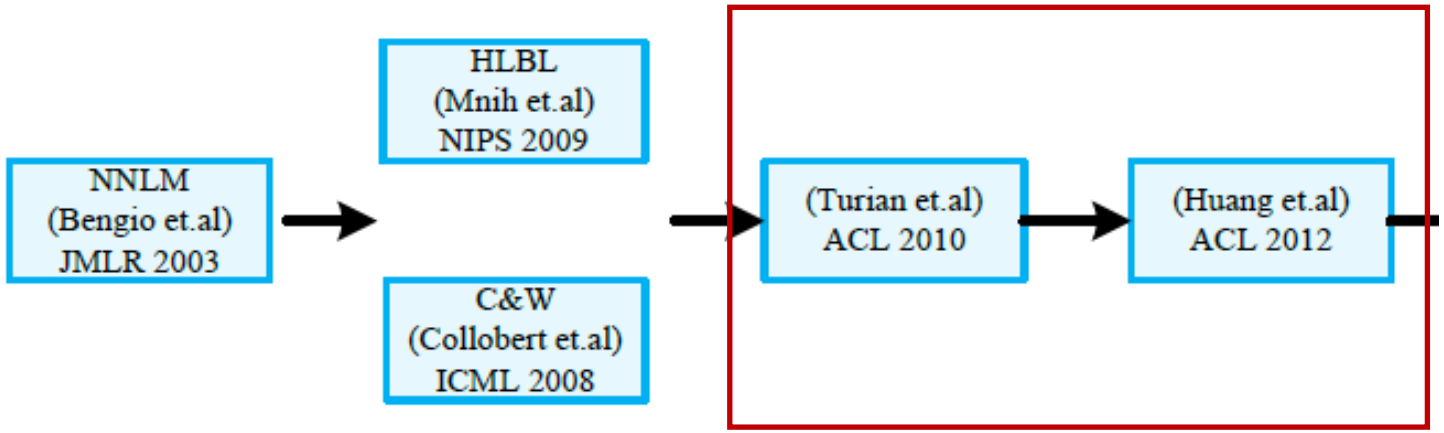
一般说来，基于对比形式的loss会让模型损失语言模型的特性。在基于人类反馈的大语言模型对齐中，这个现象被称为“**对齐税**”

- HLBL: 介绍了一种高效训练NNLM的方法
- C&W: 通过区分正、负样本来学习单词表征 $\sum_{s\in\mathcal{S}}\sum_{w\in\mathcal{D}}\max(0,\,1-f(s)+f(s^w))$

语言模型为主

词表示学习为主

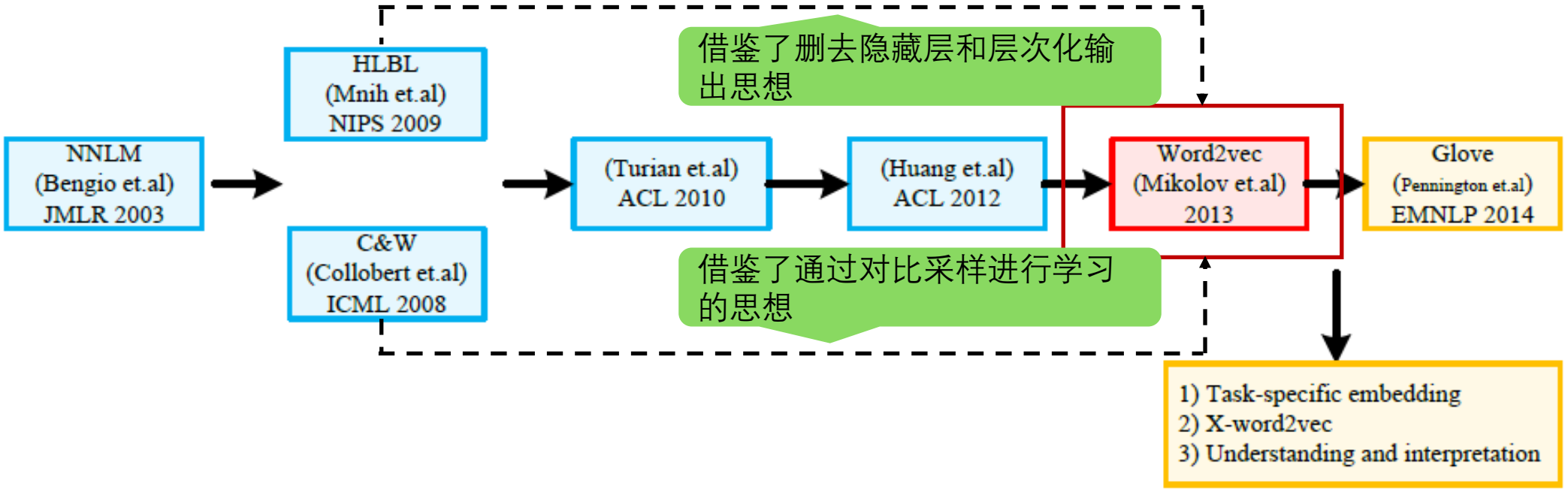
词表示学习



- Turian et. Al: 在几种典型的NLP任务中评估几种典型的词表示
  - Chunking, NER,
- Huang et. Al: 探究一词多义

关心词表示学习和评估

词表示学习



- Word2vec: 提出两种非常有效地训练词表示的方法
  - Negative sampling
  - Hierarchical Softmax

Word2vec

- 一个学习词表示“极度”高效的框架
- 设计一个只专注学习词向量网络架构和训练方法，其它都删减
- **设计哲学：提高训练效率，可以“吃”很多数据，可以兑现“算力”的优势**

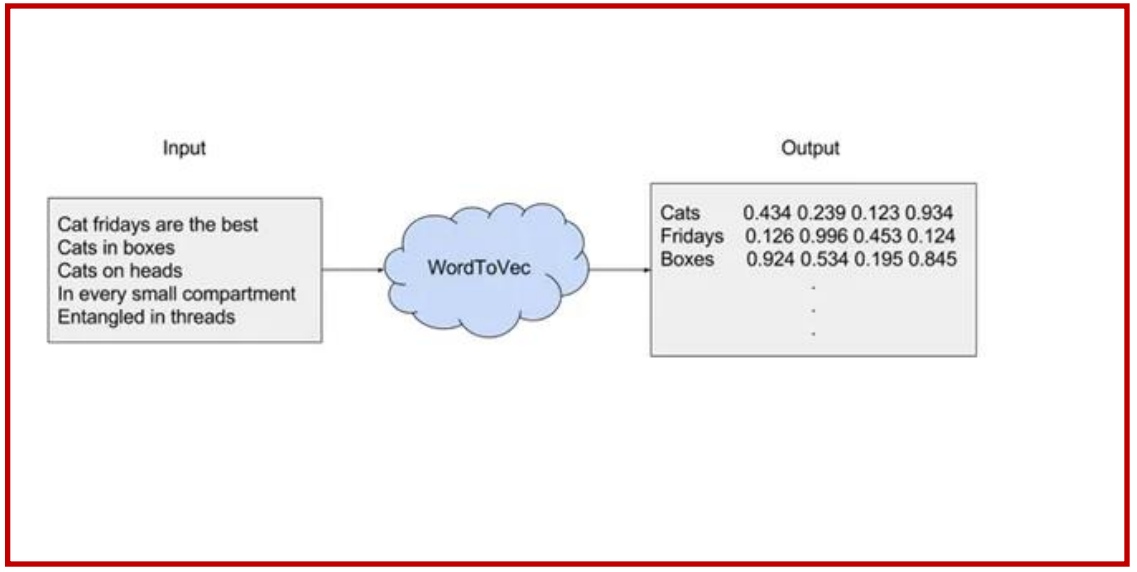
Distributed representations of words and phrases and their compositionality T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean Neural information processing systems	42840	2013
Efficient estimation of word representations in vector space T Mikolov, K Chen, G Corrado, J Dean arXiv preprint arXiv:1301.3781	40435	2013

Word2vec

- 一个学习词表示“极度”高效的框架
- 设计一个只专注学习词向量网络架构和训练方法，其它都删减
- **设计哲学：提高训练效率，可以“吃”很多数据，可以兑现“算力”的优势**

Distributed representations of words and phrases and their compositionality 42840 2013
T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean
Neural information processing systems

Efficient estimation of word representations in vector space 40435 2013
T Mikolov, K Chen, G Corrado, J Dean
arXiv preprint arXiv:1301.3781



与“France”距离相近的词语

spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

Word2vec

- 一个学习词表示“极度”高效的框架
- 设计一个只专注学习词向量网络架构和训练方法，其它都删减
- **设计哲学：提高训练效率，可以“吃”很多数据，可以兑现“算力”的优势**

Distributed representations of words and phrases and their compositionality T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean Neural information processing systems	42840	2013
Efficient estimation of word representations in vector space T Mikolov, K Chen, G Corrado, J Dean arXiv preprint arXiv:1301.3781	40435	2013

“The Bitter Lesson”

Rich Sutton
强化学习之父

The biggest lesson that can be read from 70 years of AI research is that general methods that **leverage computation** are ultimately the most effective, and by a large margin

We want AI agents that can **discover like we can**, not which contain what we have discovered. Building in our discoveries only makes it harder to see how the discovering process can be done.

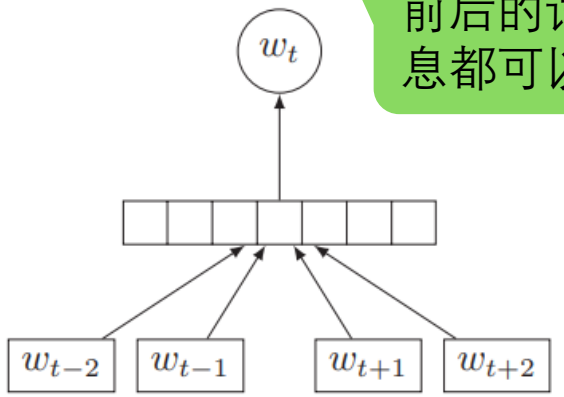
- AI 研究人员常常试图在自身智能体中构建知识，
- 从短期看，这通常是有帮助的，能够令研究人员满意，
- 但从长远看，这会令研究人员停滞不前，甚至抑制进一步发展，
- 突破性进展最终可能会通过一种相反的方法——基于以大规模计算为基础的搜索和学习

Word2vec 中的网络结构

- 连续词袋模型 (Continuous Bag-of-Words, CBOW)
  - 优化目标

$$p(w_t|c_t) = \text{softmax}\left(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t\right) = \frac{\exp(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t)}{\sum_{w'\in\mathcal{V}}\exp(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t)}$$

$$\mathbf{c}_t = \sum_{-n\le j\le n, j\neq 0} \mathbf{v}_{w_{t+j}}.$$



(a) CBOW 模型

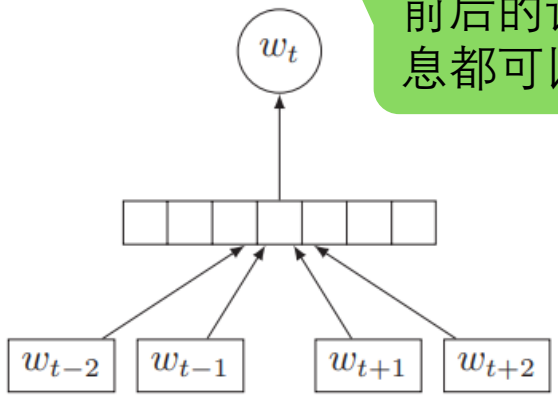
前后的词的信息都可以使用

Word2vec 中的网络结构

- 连续词袋模型 (Continuous Bag-of-Words, CBOW)
  - 优化目标

$$p(w_t|c_t) = \text{softmax}\left(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t\right) = \frac{\exp(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t)}{\sum_{w'\in\mathcal{V}}\exp(\mathbf{v'}_{w_t}^{\mathrm{T}}\mathbf{c}_t)}$$

$$\mathbf{c}_t = \sum_{-n\le j\le n, j\neq 0} \mathbf{v}_{w_{t+j}}.$$

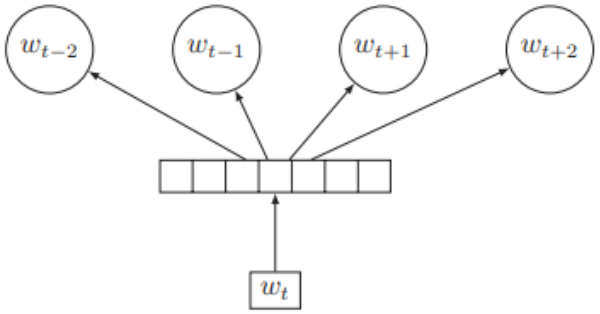


前后的词的信息都可以使用

(a) CBOW 模型

- Skip-Gram模型
  - 优化目标

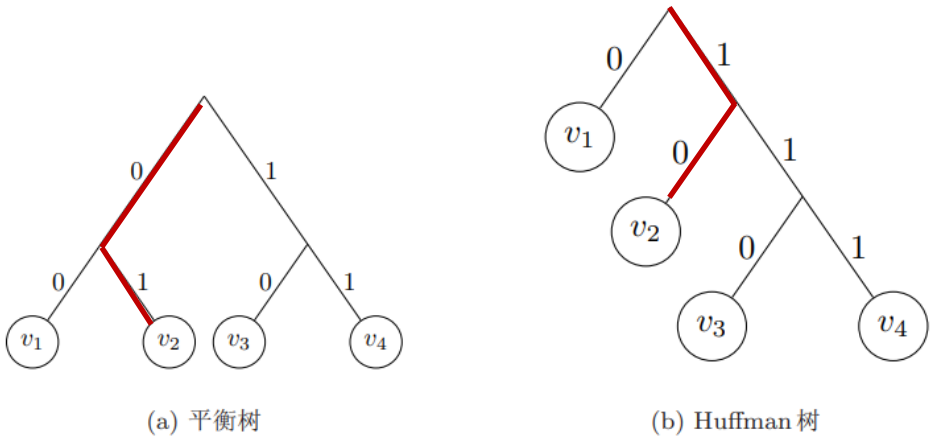
$$P(w_{t+j}|w_t) = \text{softmax}\left(\mathbf{v}_{w_t}^{\mathrm{T}}\mathbf{v}'_{w_{t+j}}\right) = \frac{\exp(\mathbf{v}_{w_t}^{\mathrm{T}}\mathbf{v}'_{w_{t+j}})}{\sum_{w'\in\mathcal{V}}\exp(\mathbf{v}_{w_t}^{\mathrm{T}}\mathbf{v}'_{w'})},$$



(b) Skip-Gram 模型

Word2vec 中的训练方法

- 层次化Softmax
  - 优化原理：条件概率估计可以转换为$\log_2|V|$个两类分类问题：
  - 构建树的方法：
    - 利用人工整理的词汇层次结构
    - 使用Huffman编码

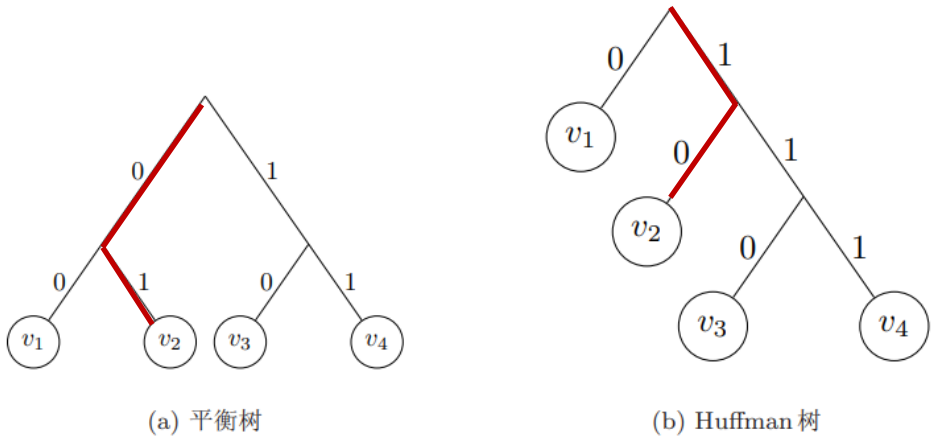


(a) 平衡树　　(b) Huffman 树

思考：假如没有层次化计算开销大在哪里？

Word2vec 中的训练方法

- 层次化Softmax
  - 优化原理：条件概率估计可以转换为$\log_2|V|$个两类分类问题：
  - 构建树的方法：
    - 利用人工整理的词汇层次结构
    - 使用Huffman编码
- 负采样 (negative sampling)
  - 优化原理：条件概率转化为多个二分类问题



(a) 平衡树　　(b) Huffman 树

$$\mathcal{L}_\theta(w_t, c_t) = -\log P(y=1|w_t, c_t) - \sum_{i=1}^{k} \log(1 - P(y=1|\tilde{w}_{t,i}, c_t))$$

$$= -\log \sigma(s(w_t, c_t; \theta)) - \sum_{i=1}^{k} \log \sigma(-s(\tilde{w}_{t,i}, c_t; \theta)).$$

Word2vec 中的 “加速技巧”

- 删除隐藏层
  - 整个网络的参数只有两个词嵌入表：输入词嵌入表和输出词嵌入表
- 使用层次化Softmax或负采样进行加速训练
- 去除低频词，对高频词进行降采样
  - 出现次数小于一个预设值词直接去除
- 动态上下文窗口大小
  - 指定一个最大窗口大小值N，对于每个词，从[1，N]中随机选取一个值n来作为本次的上下文窗口大小

Word2vec 实战

- 准备大量无监督语料（比如维基百科）
- 进行训练

Word2vec 实战

- 找语义相近的“邻居”

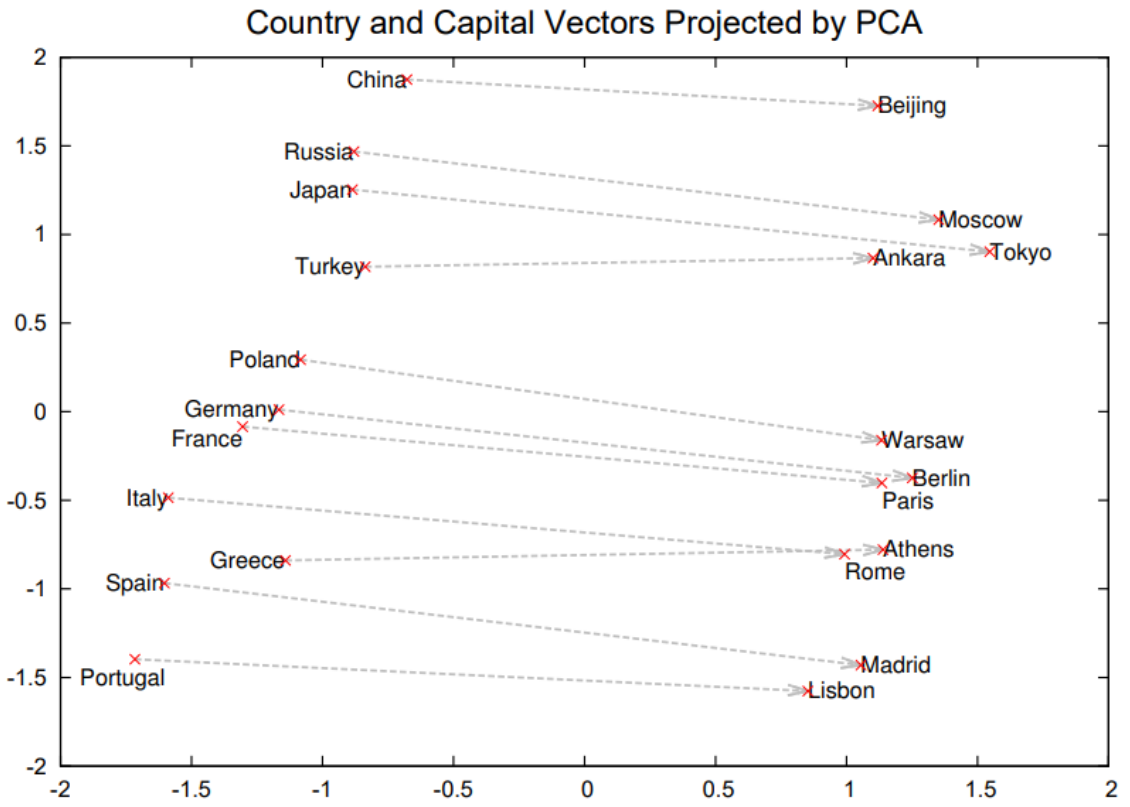
```
Query word? pidgey
pidgeot 0.891801
pidgeotto 0.885109
pidge 0.884739
pidgeon 0.787351
pok 0.781068
pikachu 0.758688
charizard 0.749403
squirtle 0.742582
beedrill 0.741579
charmeleon 0.733625
```

```
Query word? enviroment
enviromental 0.907951
environ 0.87146
enviro 0.855381
environs 0.803349
environnement 0.772682
enviromission 0.761168
realclimate 0.716746
environment 0.702706
acclimatation 0.697196
ecotourism 0.697081
```

Word2vec 实战

- 找语义相近的“邻居”
- 词的类比

```
Query triplet (A - B + C)? berlin germany france
paris 0.896462
bourges 0.768954
louveciennes 0.765569
toulouse 0.761916
valenciennes 0.760251
montpellier 0.752747
strasbourg 0.744487
meudon 0.74143
bordeaux 0.740635
pigneaux 0.736122
```

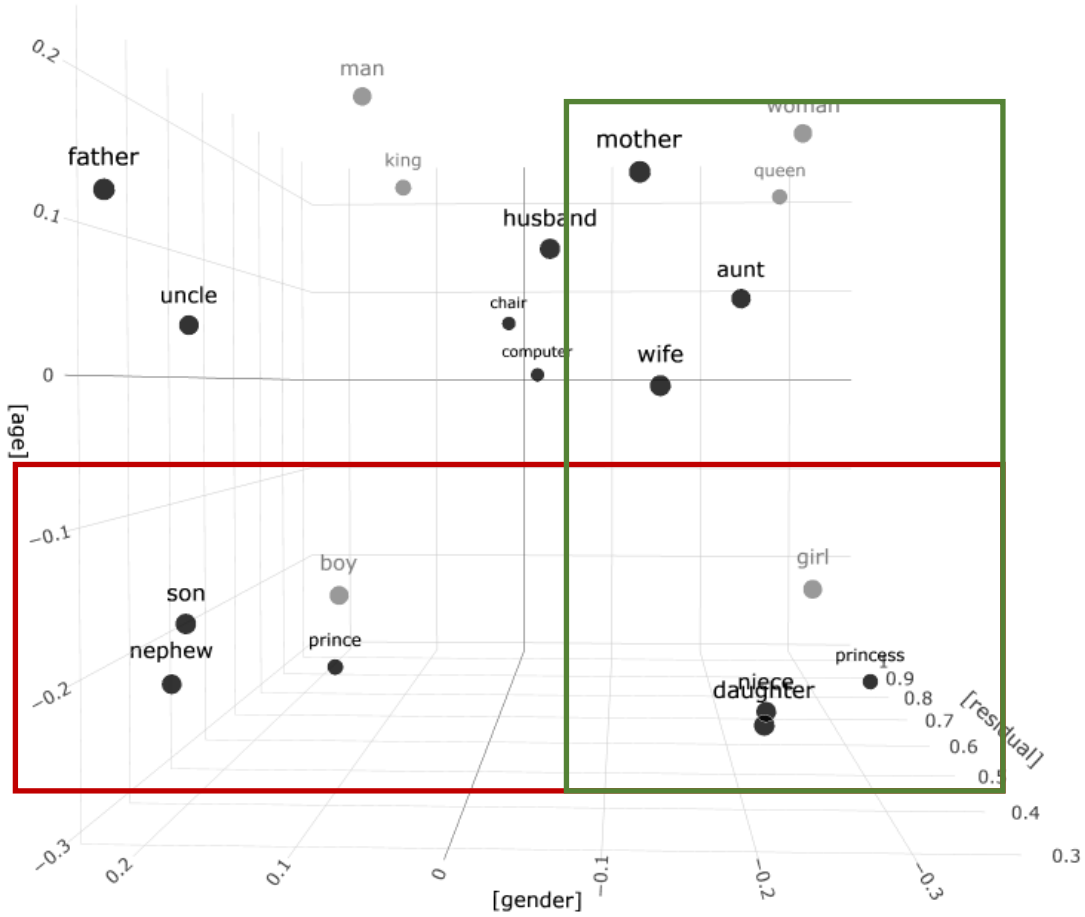


Distributed Representations of Words and Phrases and their Compositionality Mikolov et al 2013

Word2vec 实战

- 找语义相近的“邻居”
- 词的类比

```
Query triplet (A - B + C)? berlin germany france
paris 0.896462
bourges 0.768954
louveciennes 0.765569
toulouse 0.761916
valenciennes 0.760251
montpellier 0.752747
strasbourg 0.744487
meudon 0.74143
bordeaux 0.740635
pigneaux 0.736122
```



Demo from: https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/index.html

词表示质量的评估

- 内部（Intrinsic） & 外部（Extrinsic）评估
  - 内部：直接评估词向量质量
  - 外部：词向量质量对下游任务的有用程度

语言模型的评估也类似

词表示质量的内部评估

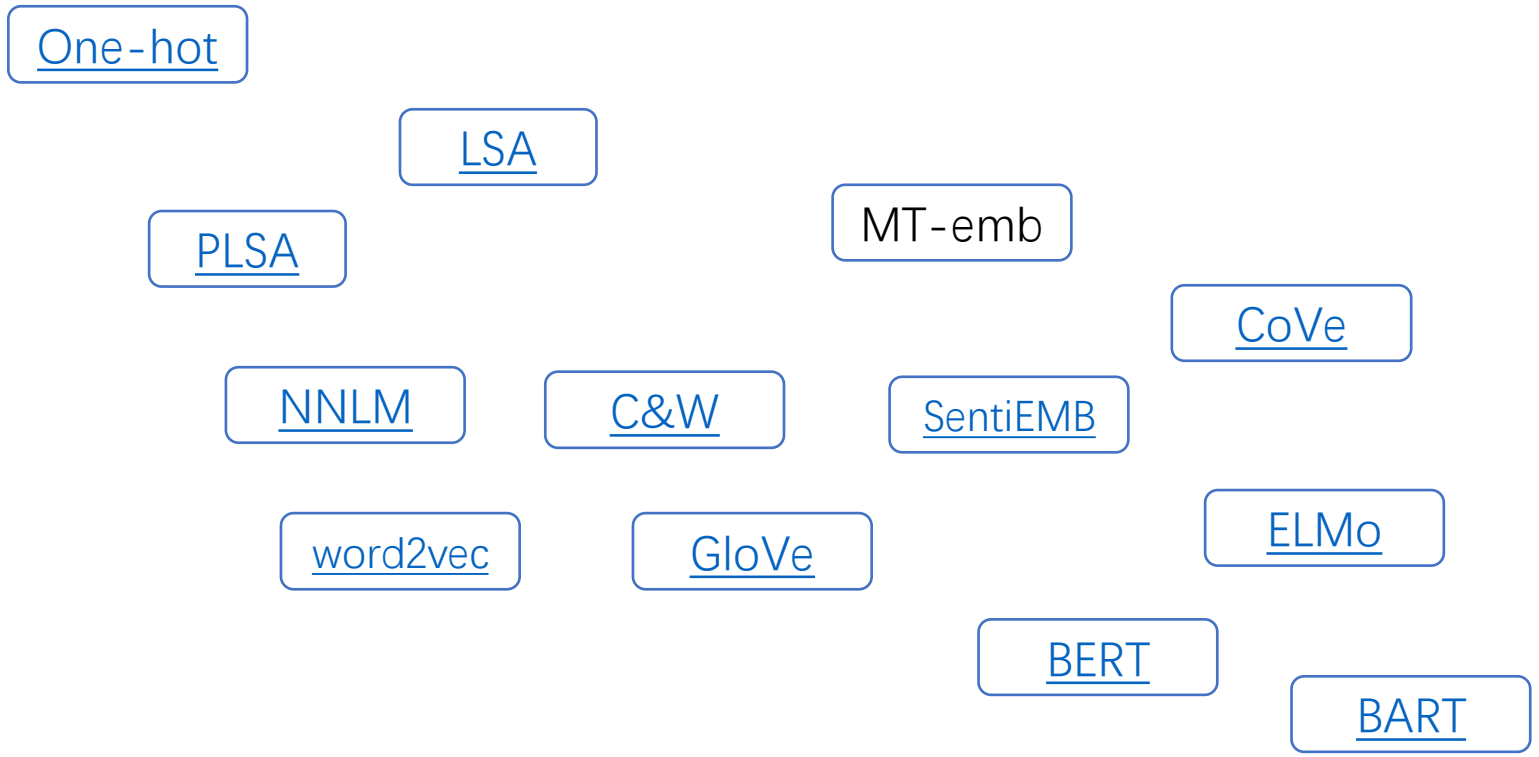
- 相关性：词向量的余弦相似性与人类相似性评估之间的相关性是什么？
- 类比：寻找x，使得“a之于b，如x之于y”。
- 分类：基于词向量创建群集(cluster)，并测量群集的纯度。
- 选择偏好：确定一个名词是否是一个动词的典型参数。

(categorization from Schnabel et al 2015)

词表示质量的外部评估

- 各种各样的NLP任务
- 比较使用词向量初始化与不使用带来的差异

词表示的学习还有很多方法!



让我们尝试对这些方法分个类

学习词表示的方法

- 符号的（Symbolic）
  - One-hot Vector

- 分布式（Distributed）
  - 实值向量

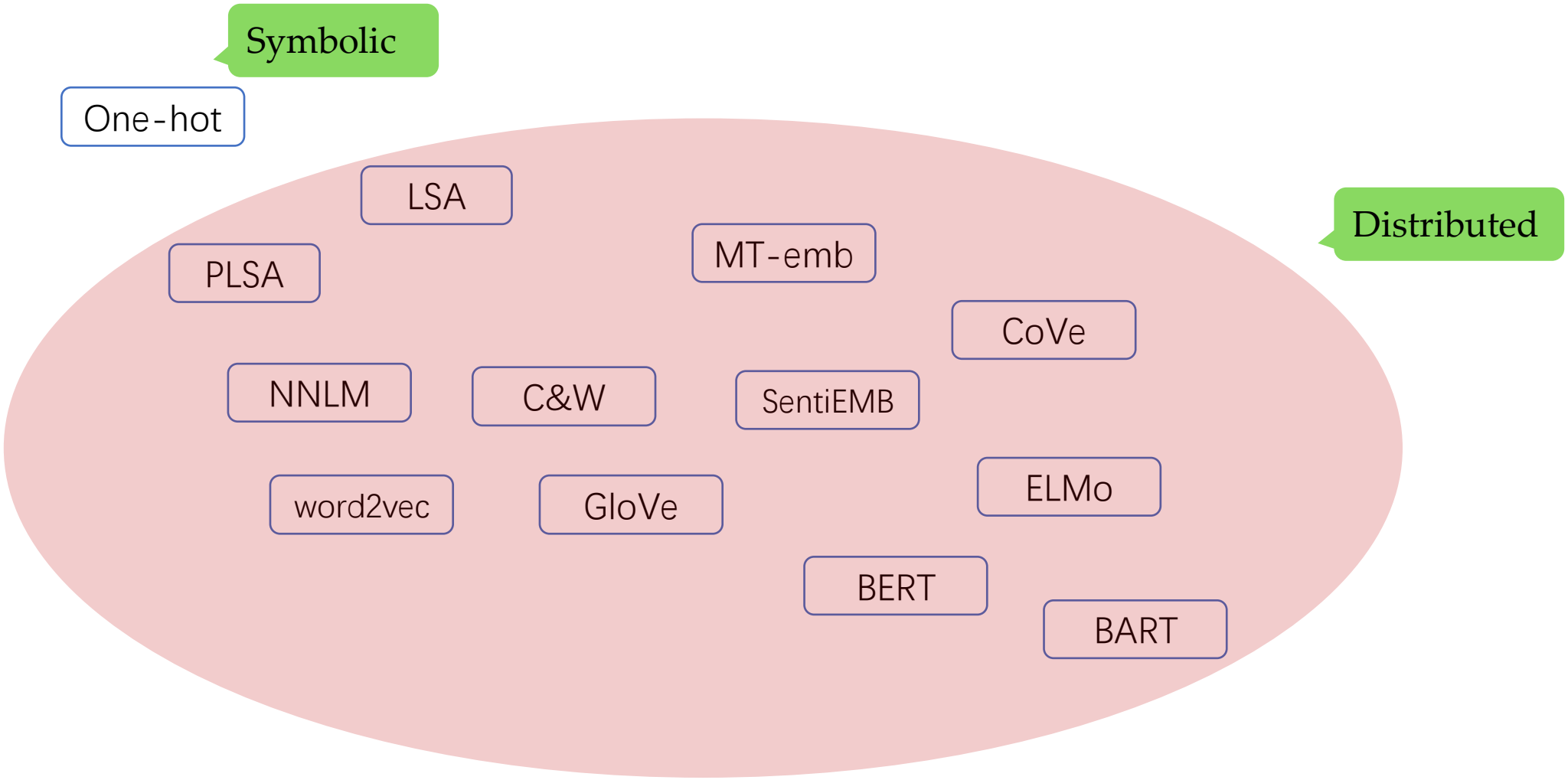
0
1
0
0
0

→ 可解释

0.1
0.3
0.5
0.1
0.9

→ 不可解释

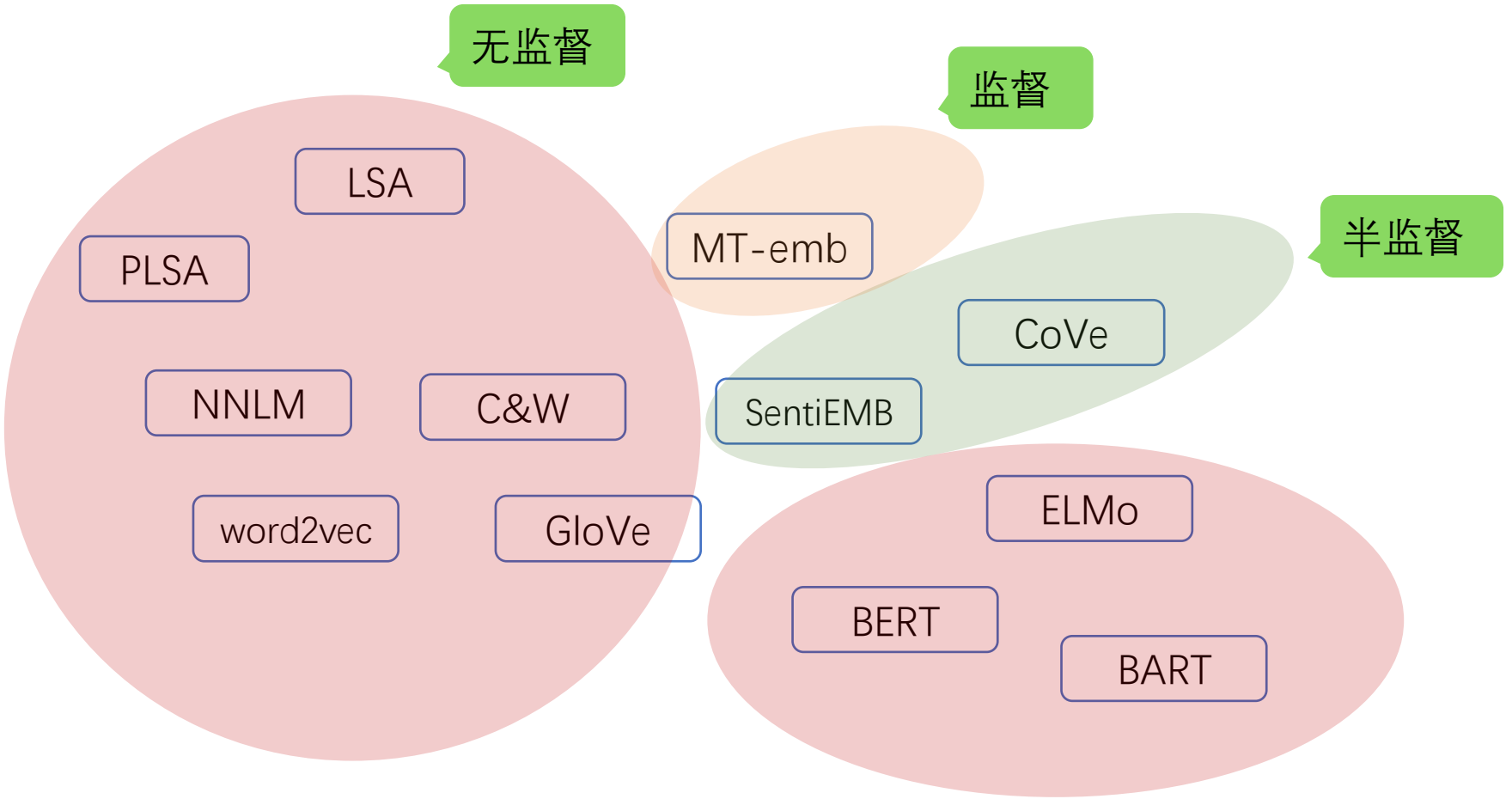
学习词表示的方法


Symbolic
One-hot
LSA
PLSA
MT-emb
Distributed
CoVe
NNLM
C&W
SentiEMB
word2vec
GloVe
ELMo
BERT
BART

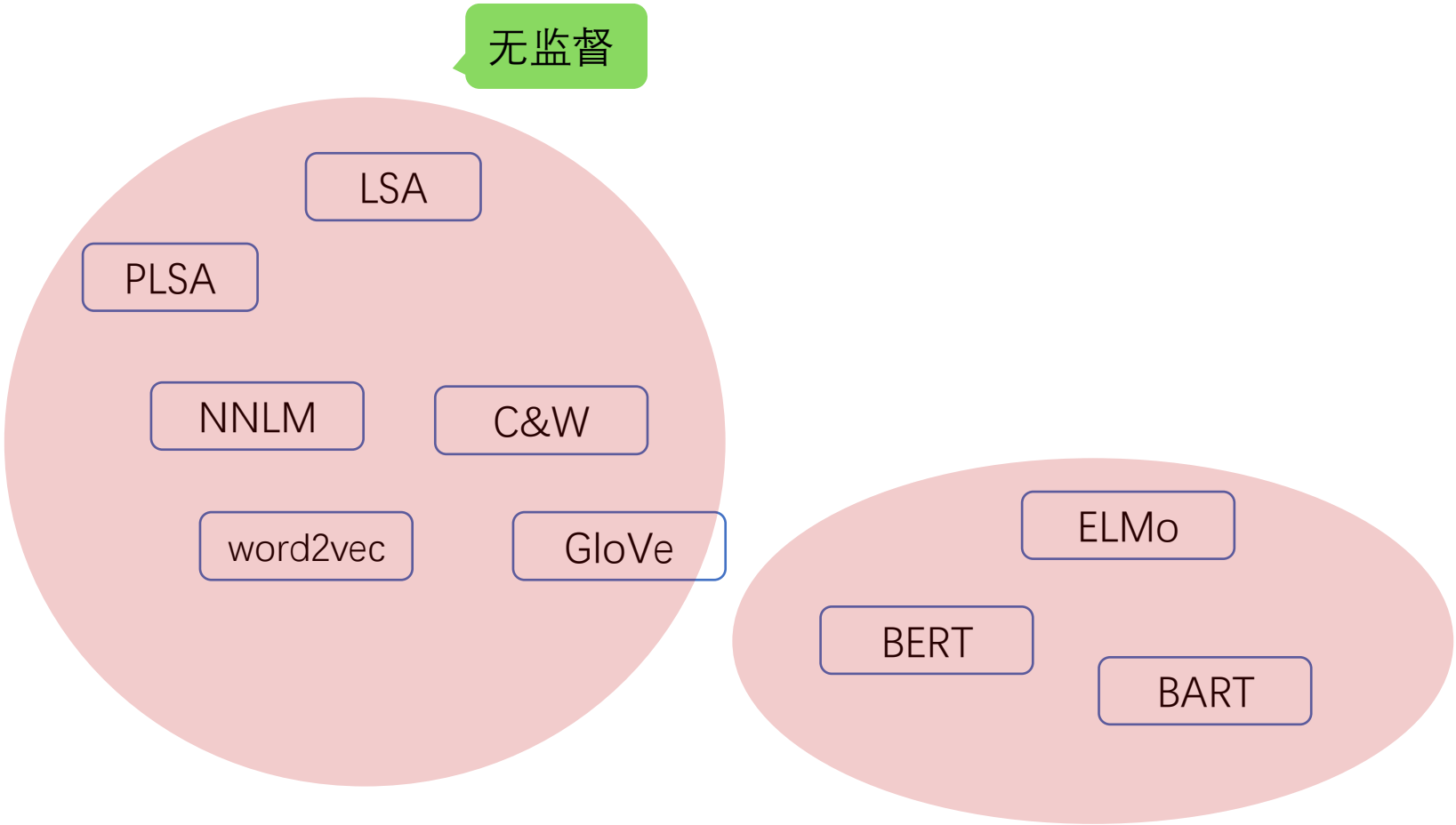
学习词表示的方法

- 监督学习
- 非监督学习
- 半监督学习

学习词表示的方法

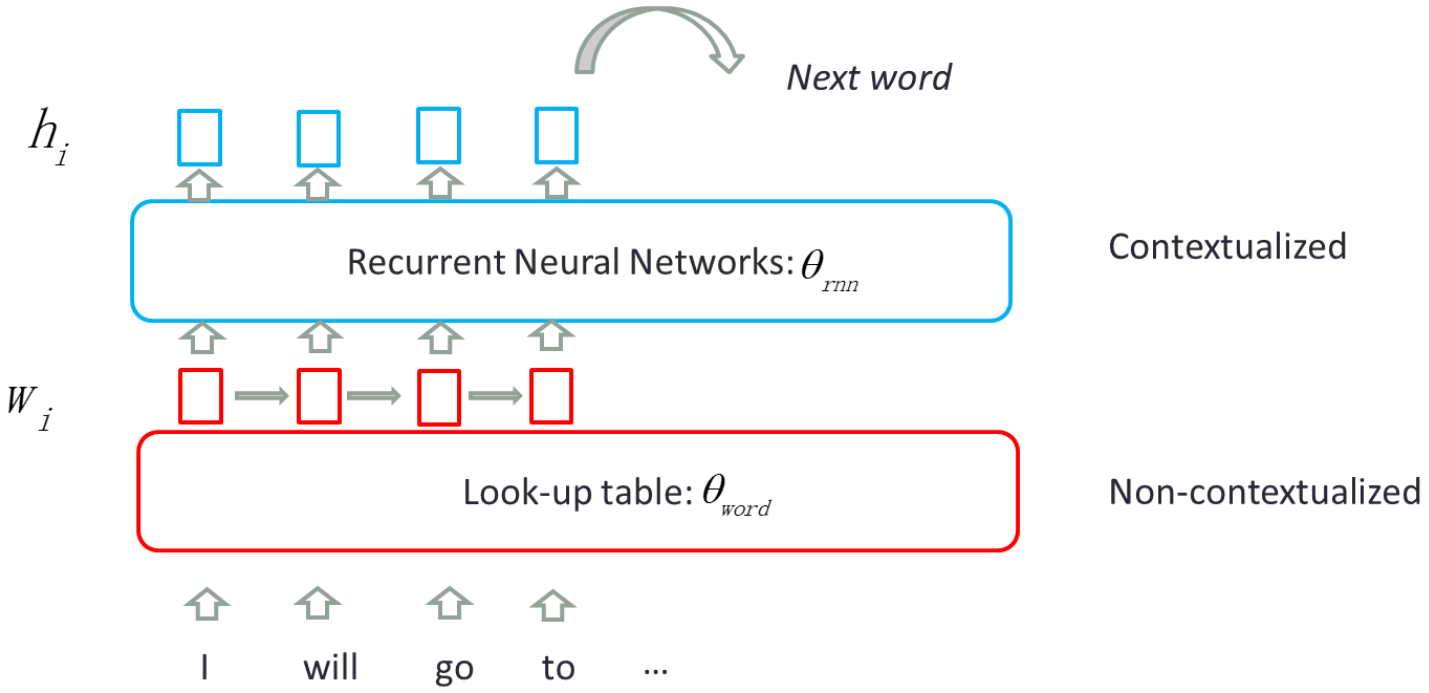

无监督
监督
半监督
LSA
PLSA
NNLM
C&W
word2vec
GloVe
MT-emb
CoVe
SentiEMB
ELMo
BERT
BART

学习词表示的方法

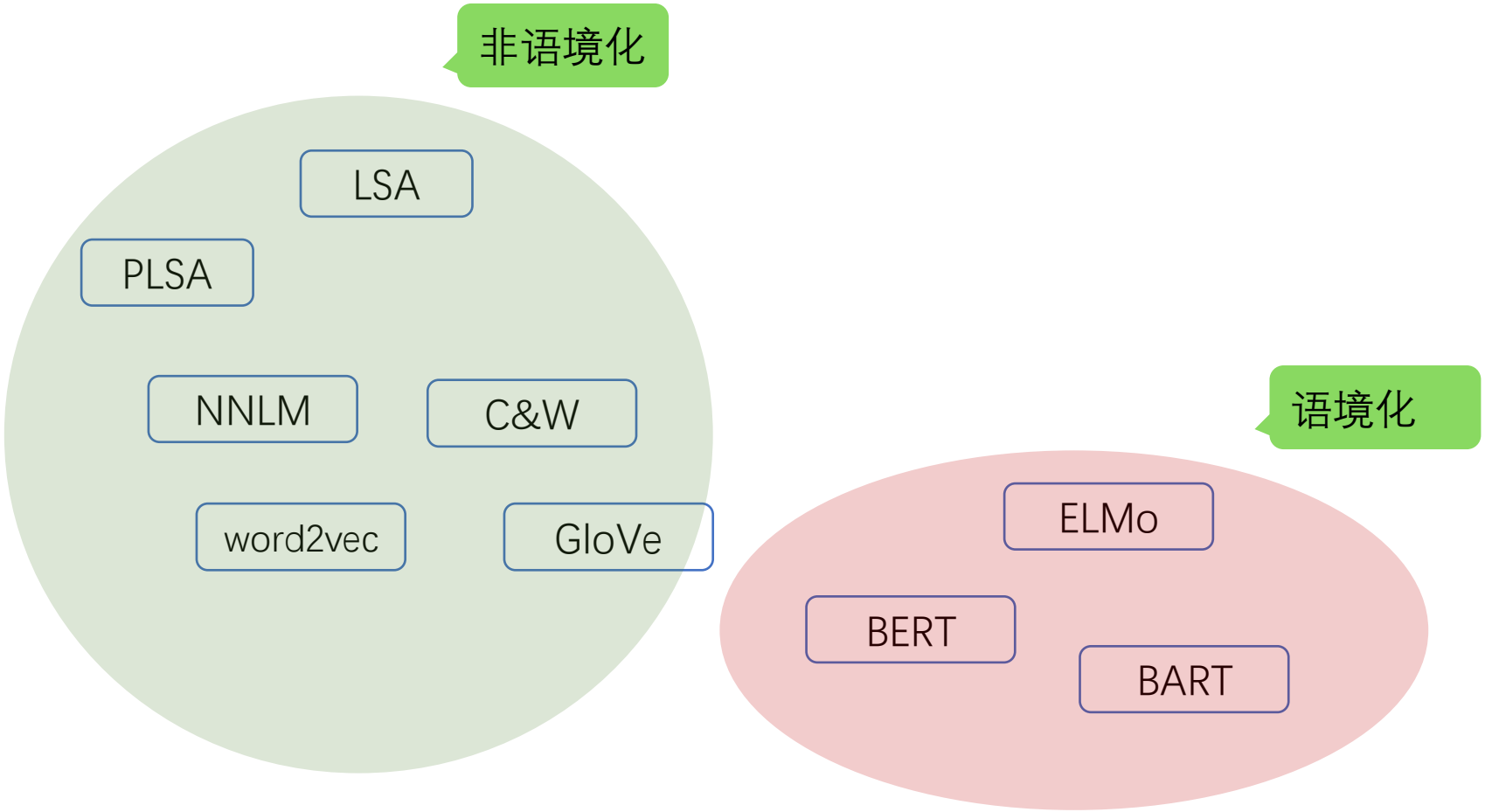


学习词表示的方法

- 非语境化的（non-contextualized)
  - Context-independent
- 语境化的（contextualized)
  - Context-dependent



学习词表示的方法



学习词表示的方法

- 基于计数的 (Count-based)
  - Count the number of co-occurrences of word/context, with rows as word, columns as contexts
  - Maybe weight with pointwise mutual information
  - Maybe reduce dimensions using SVD

假设我们有以下三篇简短的文档：

1. 文档1: "The cat is on the table."
2. 文档2: "The dog is under the table."
3. 文档3: "The cat and the dog are friends."

首先，我们创建一个词项-文档矩阵，如下所示：

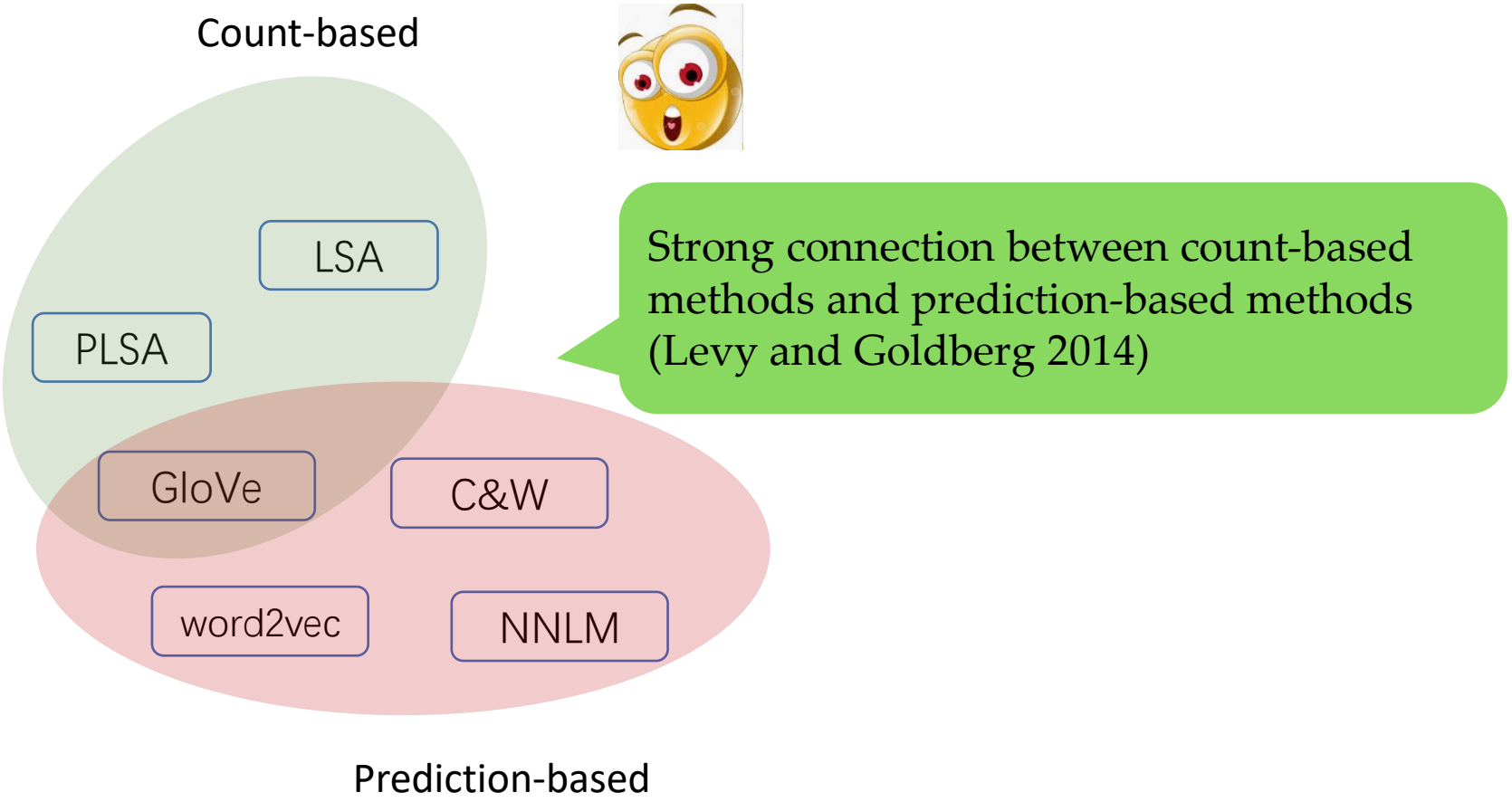
	文档1	文档2	文档3
cat	1	0	1
dog	0	1	1
table	1	1	0
friends	0	0	1

学习词表示的方法

- 基于计数的 (Count-based)
  - Count the number of co-occurrences of word/context, with rows as word, columns as contexts
  - Maybe weight with pointwise mutual information
  - Maybe reduce dimensions using SVD

- 基于预测的（Prediction-based)
  - try to predict the words within a neural network

学习词表示的方法


Count-based
LSA
PLSA
GloVe
C&W
word2vec
NNLM
Prediction-based
Strong connection between count-based methods and prediction-based methods (Levy and Goldberg 2014)

什么时候使用“non-contextualized”？

- Limited computation resources
- Fast training/quickly evaluate your models
- No off-the-shelf BERT models
- Huge domain shift
- Best of both worlds

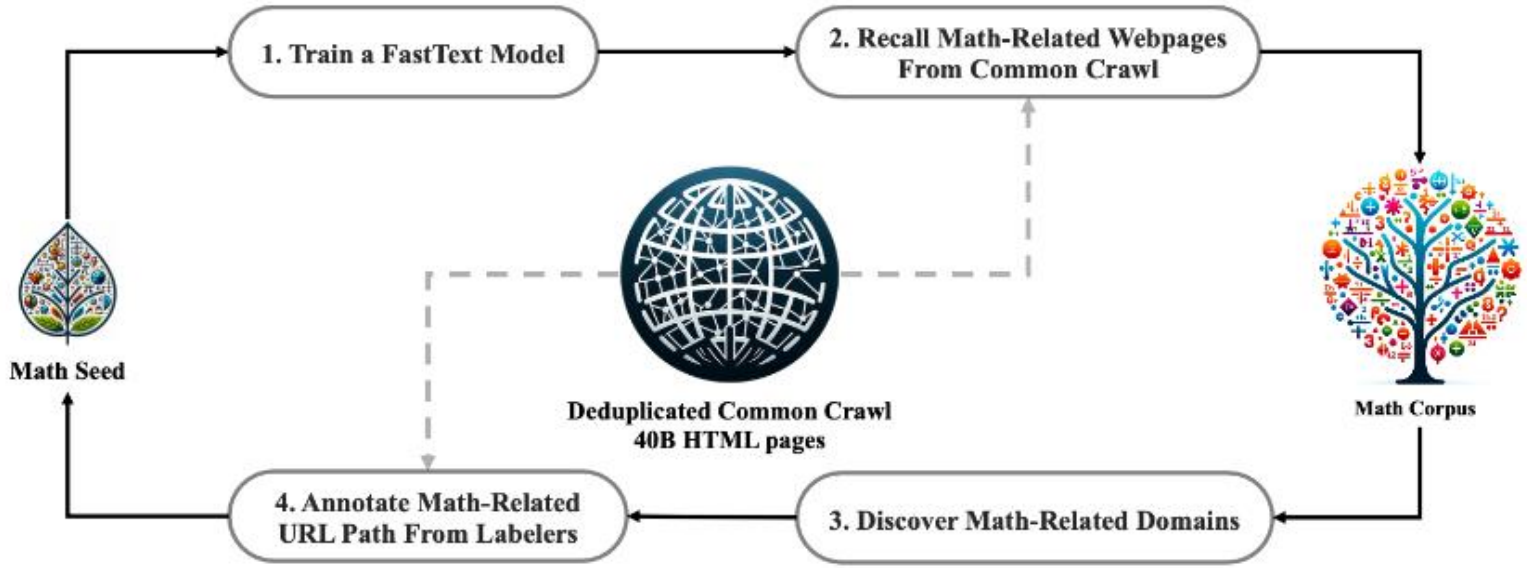


Image from: https://github.com/deepseek-ai/DeepSeek-Math

什么时候使用“contextualized”？

- Rich in GPUs
- Care the SOTA result
- Don’ t care the training time
- Off-the-shelf BERT models
- Few training samples/Low-resource