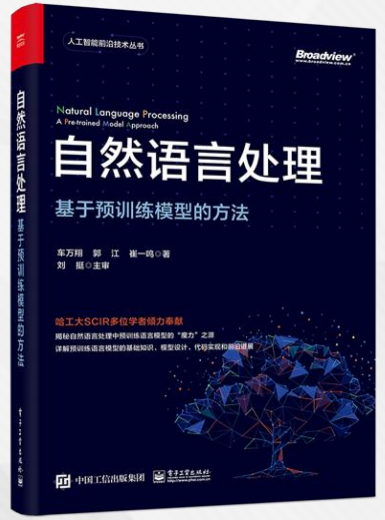



# 自然语言处理基础

车万翔、郭江、崔一鸣

社会计算与信息检索研究中心
哈尔滨工业大学

# 目 录

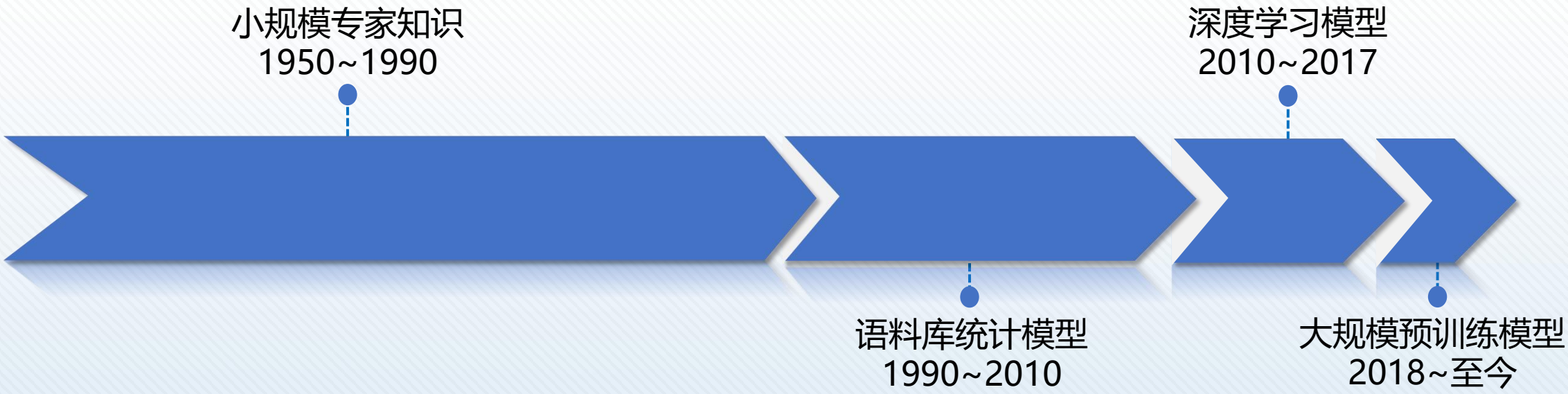
CONTENTS

# 目 录

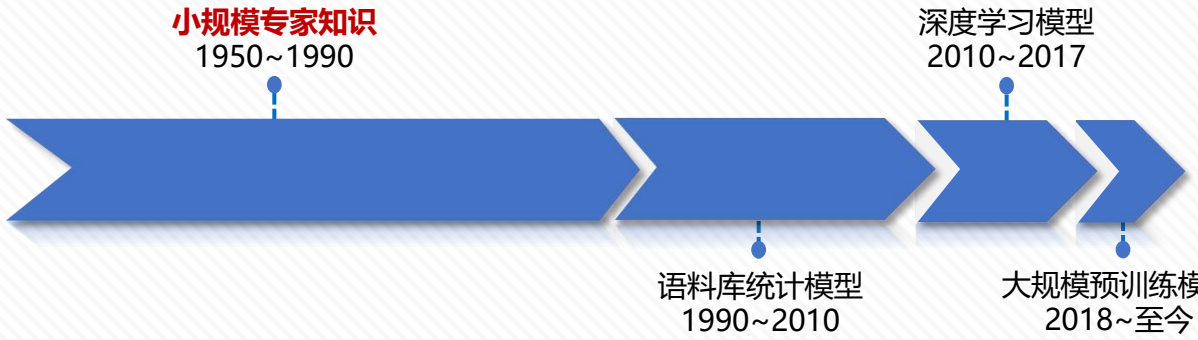
CONTENTS

# 自然语言处理技术发展历史

- 语义在计算机内部是如何**表示**的?
- 根据表示方法的不同，自然语言处理技术经历了**四次范式变迁**



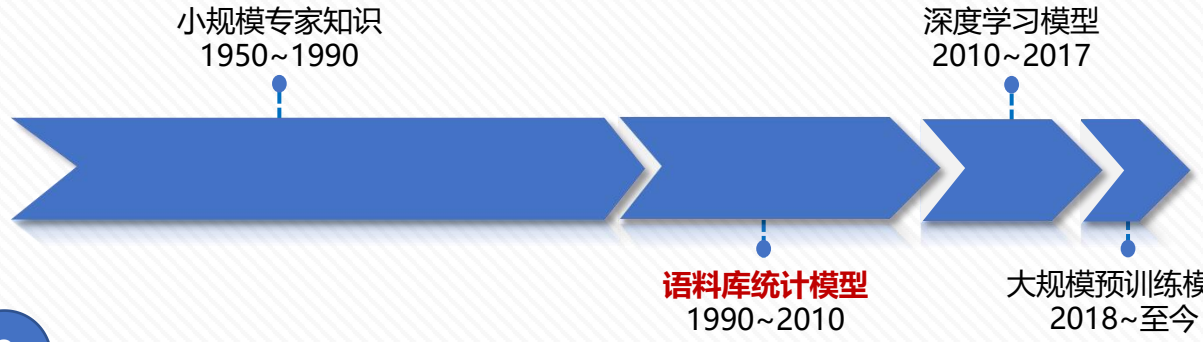
# 基于符号（字符串）表示的专家知识

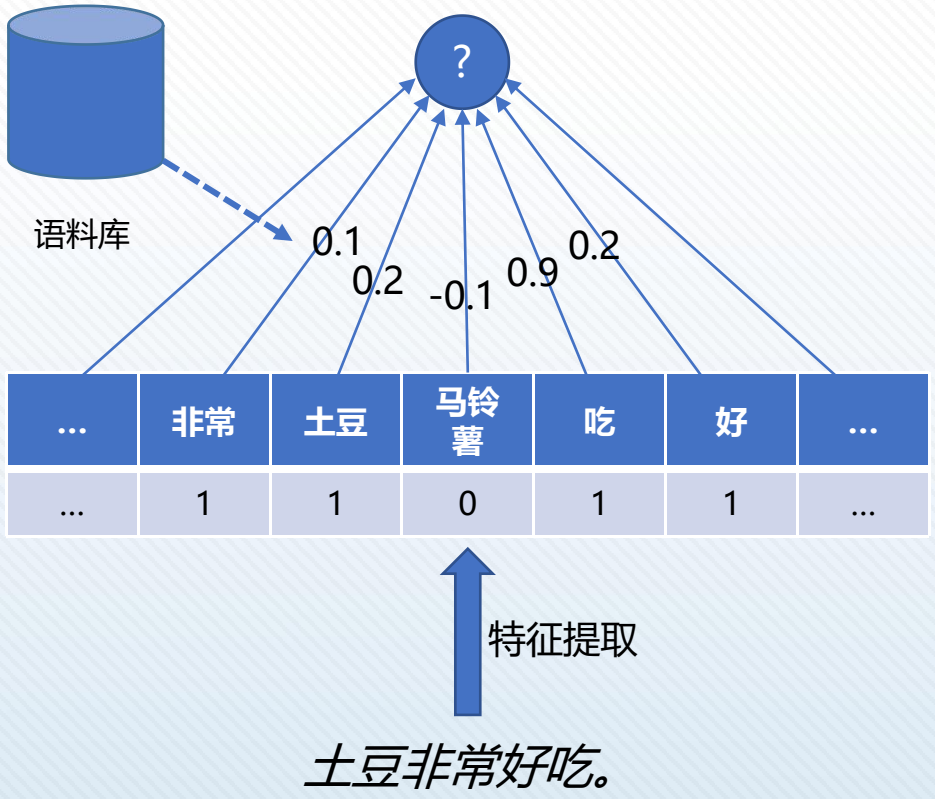



- “*土豆非常好吃。*”的情感倾向性？
  - 如果：出现褒义词“好”“喜欢”等
  - 那么：结果为褒义
  - 如果：出现“不”
  - 那么：结果倾向性取反

- 优点
  - 符合人类的直觉
  - 可解释、可干预性好
- 缺点
  - 知识完备性不足
  - 需要专家构建和维护
  - 不便于计算

# 基于向量表示的统计模型

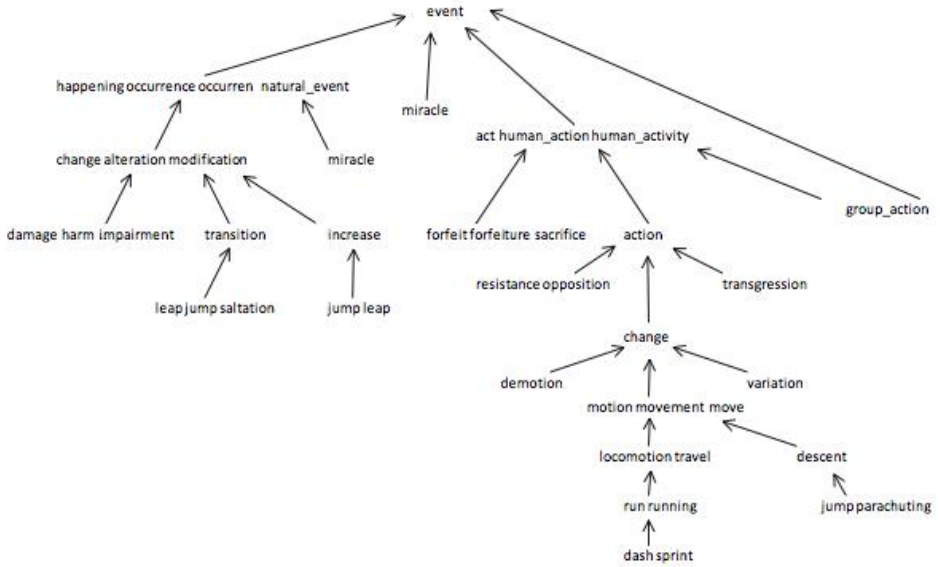


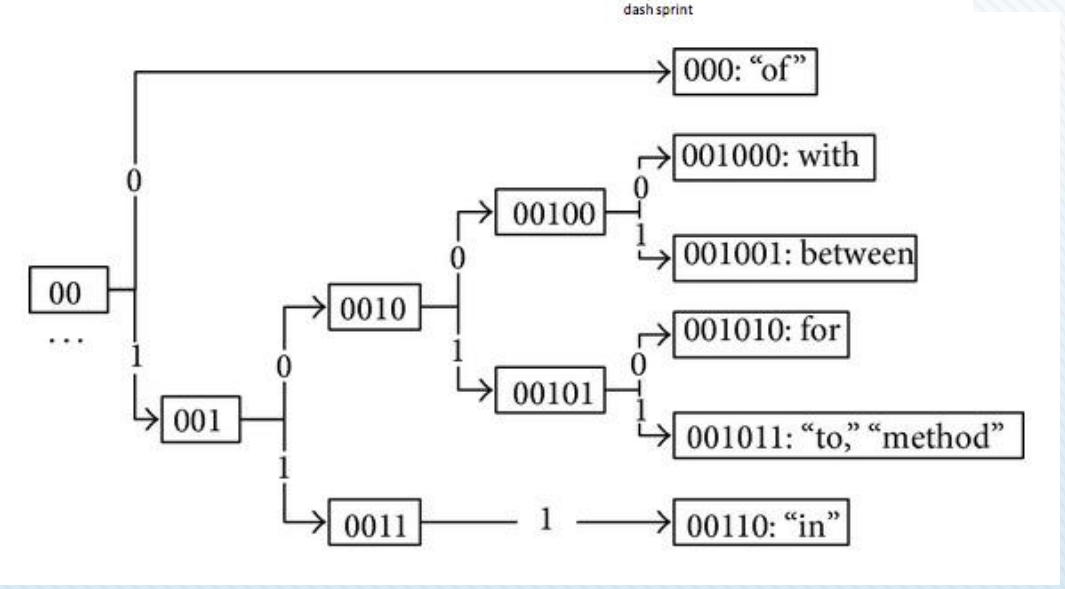




- 使用**高维、离散、稀疏**的向量表示词
  - 维度为词表大小，其中只有一位为1，其余为0
    - **土豆**：[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...]
    - **马铃薯**：[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...]
- 缺点
  - 严重的**数据稀疏**问题
  - 无法处理“**多词一义**”的现象

# 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)



# 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

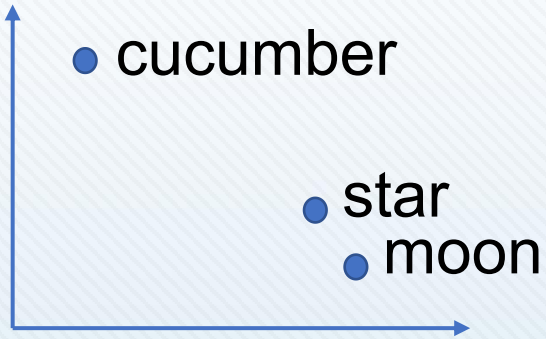
he curtains open and the moon shining in on the barely
ars and the cold , close moon " . And neither of the w
rough the night with the moon shining so brightly , it
made in the light of the moon . It all boils down , wr
surely under a crescent moon , thrilled by ice-white
sun , the seasons of the moon ? 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

# 词的分布（Distributional）表示

- 分布词向量

| | shinning | bright | trees | dark | look |
|---|---|---|---|---|---|
| moon | 38 | 45 | 2 | 27 | 12 |

- 语义相似度通过计算向量相似度获得




- 仍然存在高维、稀疏、离散的问题

# 分布表示的加权

## 降低高频词的权重

### 点互信息（Pointwise Mutual Information，PMI）

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

*我 喜欢 自然 语言 处理 。*
*我 爱 深度 学习 。*
*我 喜欢 机器 学习 。*

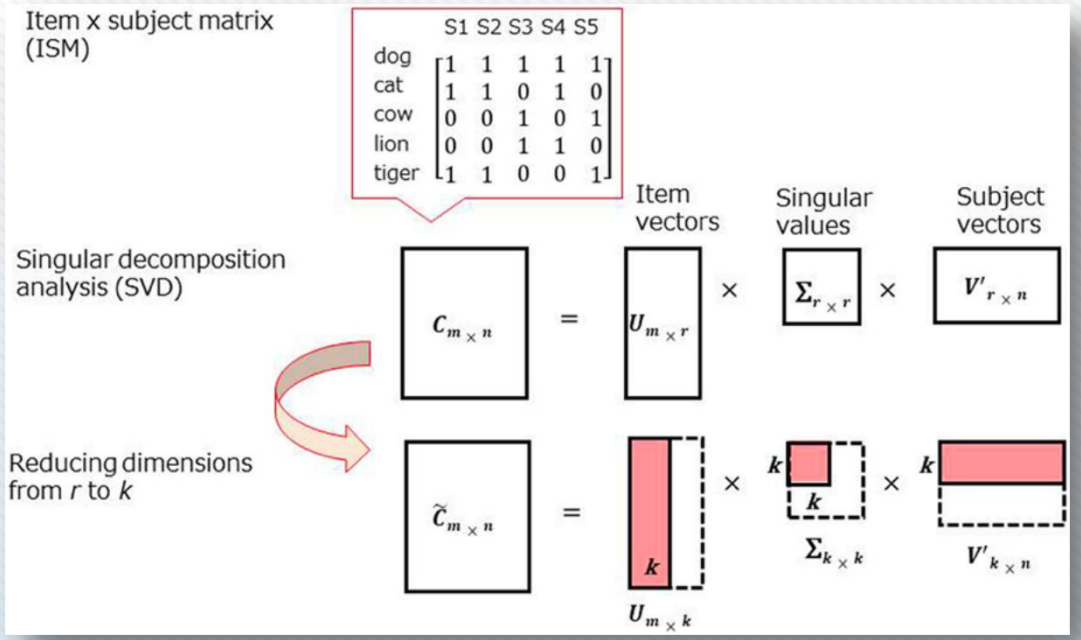
| | 我 | 喜欢 | 自然 | 语言 | 处理 | 爱 | 深度 | 学习 | 机器 | 。 |
|---|---|---|---|---|---|---|---|---|---|---|
| 我 | 0 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 3 |
| 喜欢 | 2 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 2 |
| 自然 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 语言 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 处理 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 爱 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 深度 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 学习 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 机器 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 。 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0 |

```
def pmi(M, positive=True):
    col_totals = M.sum(axis=0) # 按列求和
    row_totals = M.sum(axis=1) # 按行求和
    total = col_totals.sum() # 总频次
    expected = np.outer(row_totals, col_totals) / total # 获得每个元素的分子
    M = M / expected
    with np.errstate(divide='ignore'): # 不显示log(0)的警告:
        M = np.log(M)
    M[np.isinf(M)] = 0.0  # 将log(0)置为0
    if positive:
        M[M < 0] = 0.0
    return M
```

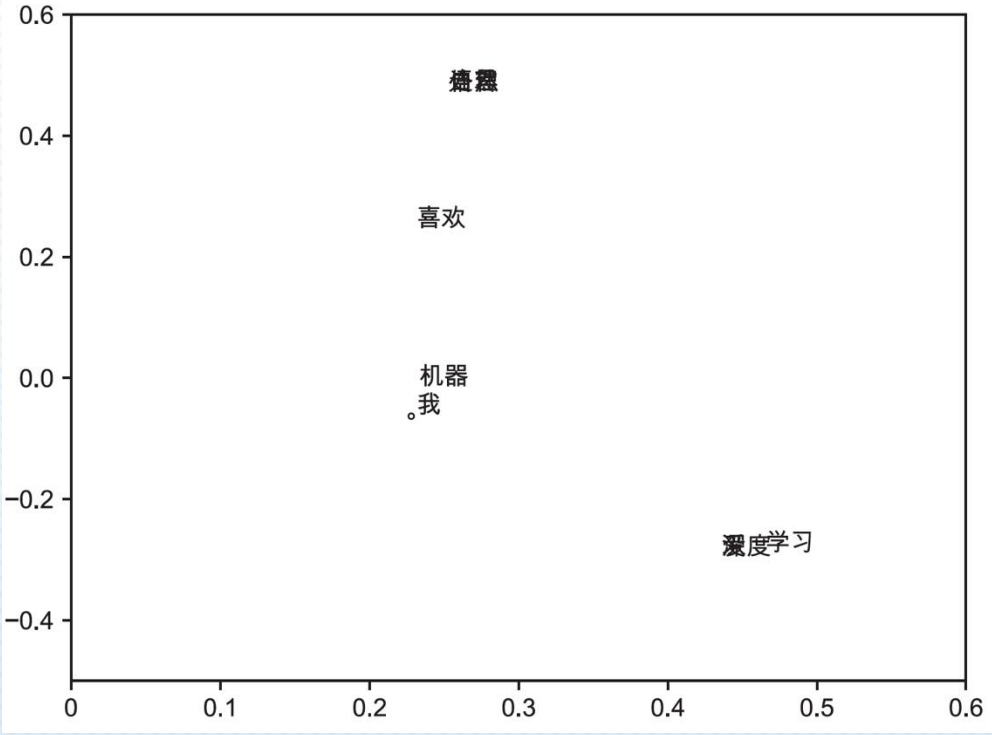
```
[[0.   0.17 0.06 0.06 0.06 0.28 0.28 0.28 0.28 0.28]
 [0.17 0.   0.43 0.43 0.43 0.   0.   0.   0.65 0.25]
 [0.06 0.43 0.   1.02 1.02 0.   0.   0.   0.   0.14]
 [0.06 0.43 1.02 0.   1.02 0.   0.   0.   0.   0.14]
 [0.06 0.43 1.02 1.02 0.   0.   0.   0.   0.   0.14]
 [0.28 0.   0.   0.   0.   0.   1.46 0.77 0.   0.36]
 [0.28 0.   0.   0.   0.   1.46 0.   0.77 0.   0.36]
 [0.42 0.09 0.   0.   0.   0.9  0.9  0.   0.9  0.  ]
 [0.28 0.65 0.   0.   0.   0.   0.   0.77 0.   0.36]
 [0.2  0.17 0.06 0.06 0.06 0.28 0.28 0.28 0.28 0.  ]]
```

# 分布表示的降维

- 避免稀疏性，反映高阶共现关系
  - 奇异值分解（Singular Value Decomposition，SVD）

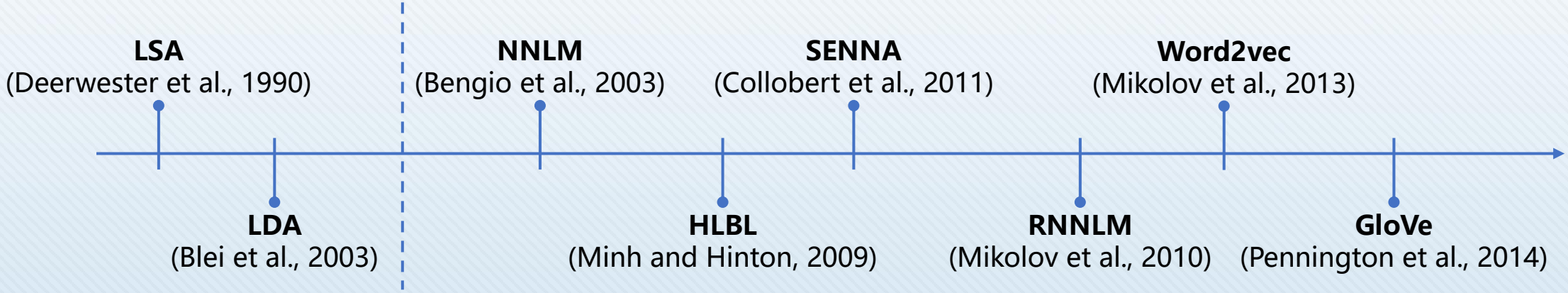



```
U, s, Vh = np.linalg.svd(M_pmi)
```



# 分布式（Distributed）词表示

- 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（第5章将详细介绍）



# 目 录

CONTENTS

# 语言模型

- 语言模型（Language Model，LM）
  - 描述一段自然语言的概率或给定上文时下一个词出现的概率
    - $P(w_1, \cdots w_l)$，$P(w_{l+1}|w_1, \cdots w_l)$
  - 以上两种定义等价（链式法则）

$$P(w_1 w_2 \cdots w_l) = P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)\cdots P(w_l|w_1 w_2 \cdots w_{l-1})$$

$$= \prod_{i=1}^{l} P(w_i|w_{1:i-1})$$

- 广泛应用于多种自然语言处理任务
  - 机器翻译（词排序）
    - $P$(the cat is small) > $P$(small the is cat)
  - 语音识别（词选择）
    - $P$(there are **four** cats) > $P$(there are **for** cats)

# 中文分词

- 词（Word）
  - 最小的能独立使用的音义结合体
  - 以汉语为代表的汉藏语系，以阿拉伯语为代表的闪-含语系中不包含明显的词之间的分隔符
- 中文分词是将中文字序列切分成一个个单独的词
- 分词的歧义
  - 如：严守一把手机关了
    - 严守一/ 把/ 手机/ 关/ 了
    - 严守/ 一把手/ 机关/ 了
    - 严守/ 一把/ 手机/ 关/ 了
    - 严守一/ 把手/ 机关/ 了
    - ……

# 子词切分（Subword）

- 以英语为代表的印欧语系语言，是否需要进行分词？
- 这些语言词形变化复杂
  - 如：*computer*、*computers*、*computing*等
- 仅用空格切分的问题
  - 数据稀疏
  - 词表过大，降低处理速度
- 子词切分
  - 将一个单词切分为若干连续的片段（子词）
  - 方法众多，基本原理相似
    - 使用尽量长且频次高的子词对单词进行切分

## 字节对编码（Byte Pair Encoding，BPE）

**Algorithm 1:** BPE 中子词词表构造算法

**Input:** 大规模生文本语料库；期望的子词词表大小 $L$

**Output:** 子词词表

将语料库中每个单词切分成字符作为子词;

用切分的子词构成初始子词词表;

**while** 子词词表小于等于 $L$ **do**

  在语料库中统计单词内相邻子词对的频次;

  选取频次最高的子词对，合并成新的子词;

  将新的子词加入子词词表;

  将语料库中不再存在的子词从子词词表中删除;

**end**

```
语料库: {'l o w e r </w>': 2, 'n e w e s t </w>': 6, 'w i d e s t </w>': 3}

初始子词词表: {'l', 'o', 'w', 'e', 'r', '</w>', 'n', 's', 't', 'i', 'd'}

合并子词词表: {'l', 'o', 'w', 'e', 'r', '</w>', 'n', 't', 'i', 'd', 'es'}

语料库: {'l o w e r </w>': 2, 'n e w es t </w>': 6, 'w i d es t </w>': 3}

合并子词词表: {'l', 'o', 'w', 'e', 'r', '</w>', 'n', 'i', 'd', 'est'}

语料库: {'l o w e r </w>': 2, 'n e w est </w>': 6, 'w i d est </w>': 3}
```

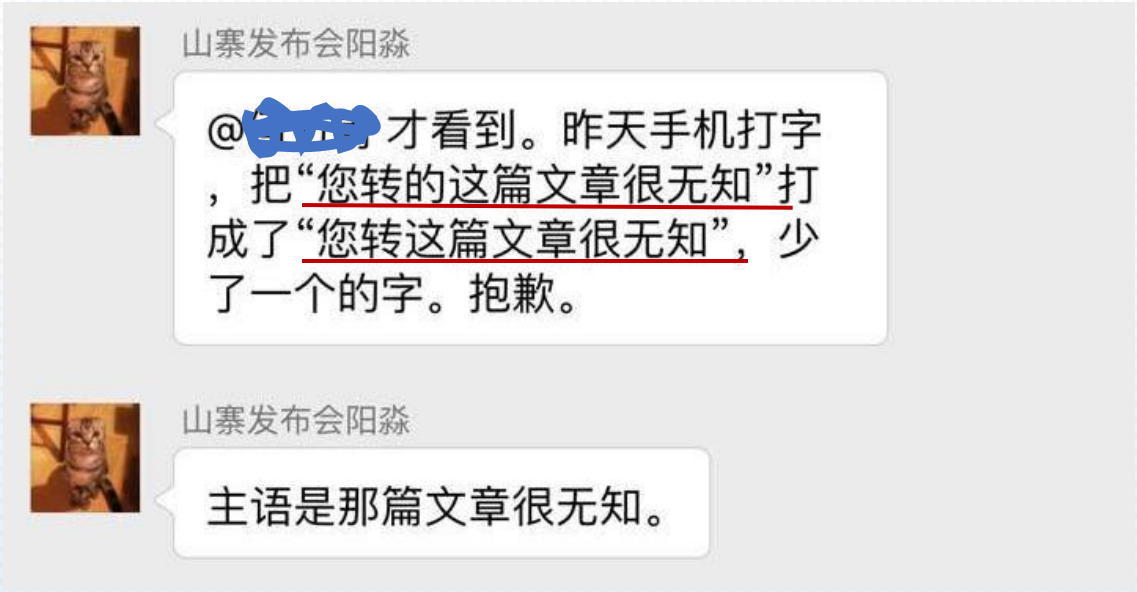
子词词表构造示例

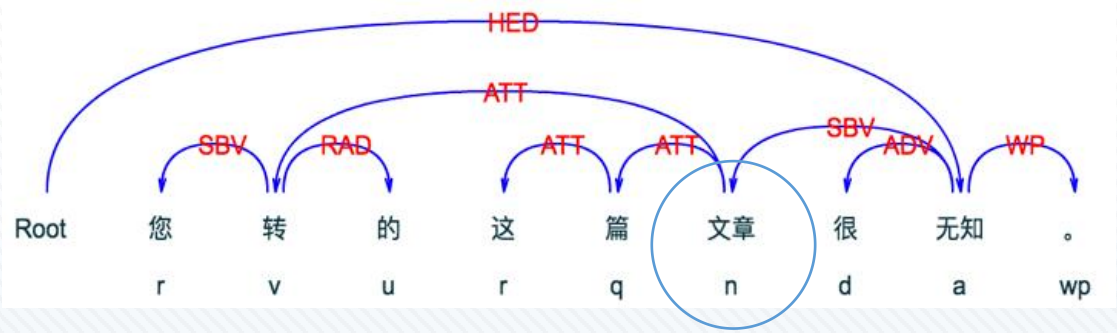
# 字节对编码

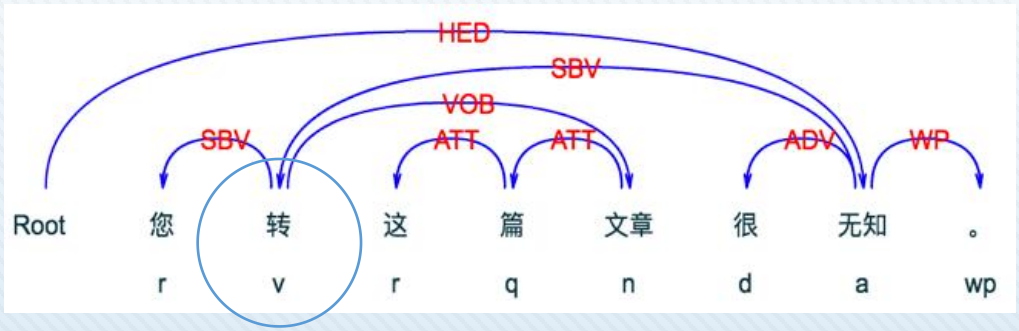
- BPE子词切分算法
  1. 将子词词表按照子词的长度由大到小进行排序
  2. 从前向后遍历子词词表，依次判断一个子词是否为单词的子串
  3. 如果是则将该单词进行切分，然后继续向后遍历子词词表
  4. 如果子词词表全部遍历结束，单词中仍然有子串没有被切分，那么这些子串一定为低频串，则使用统一的标记，如'<UNK>'进行替换
- 更多子词切分算法
  - WordPiece
  - Unigram Language Model（ULM）
  - SentencePiece
    - https://github.com/google/sentencepiece

# 句法分析

- 分析句子的句法成分，如主谓宾定状补等
- 将词序列表示的句子转换成树状结构



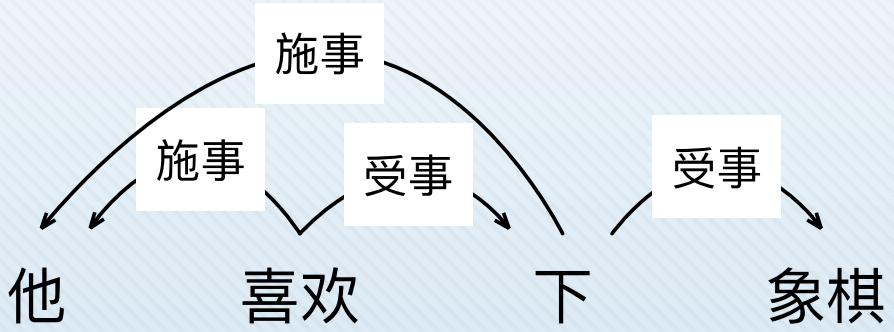

http://ltp.ai/demo.html

# 语义分析

- 词义消歧（Word Sense Disambiguation，WSD）
- 语义角色标注（Semantic Role Labeling，SRL）
  - 也称谓词论元结构（Predicate-Argument Structure）

| 输入 | 他 | 喜欢 | 下 | 象棋 | 。 |
|---|---|---|---|---|---|
| 输出 1 | 施事 | 谓词 | 受事 | | |
| 输出 2 | 施事 | | 谓词 | 受事 | |

- 语义依存图（Semantic Dependency Graph）



# 信息抽取

- 信息抽取（Information Extraction，IE）
  - 从非结构化的文本中自动提取结构化信息
- 输入
  - *10月28日，AMD宣布斥资350亿美元收购FPGA芯片巨头赛灵思。这两家传了多年绯闻的芯片公司终于走到了一起。*

| 信息抽取子任务 | 抽取结果 |
| --- | --- |
| 命名实体识别 | 公司名：AMD<br>公司名：赛灵思 |
| 关系抽取 | 赛灵思 $\xrightarrow{\text{从属}}$ AMD |
| 时间表达式抽取 | *10 月 28 日* |
| 时间表达式归一化 | *10 月 28 日* → *2020 年 10 月 28 日* |
| 事件抽取 | 事件：收购<br>时间：*2020 年 10 月 28 日*<br>收购者：AMD<br>被收购者：赛灵思<br>收购金额：*350 亿美元* |

# 情感分析

- 情感分析（Sentiment Analysis）
  - 个体对外界事物的态度、观点或倾向性，如正面、负面等
  - 人自身的情绪（Emotion），如喜怒哀惧等
- 输入
  - *这款手机的屏幕很不错，性能也还可以。*

| 情感分析子任务 | 分析结果 |
| --- | --- |
| 情感分类 | 褒义 |
| 情感信息抽取 | 评价词：不错；可以<br>评价对象：屏幕；性能<br>评价搭配：屏幕 ⇔ 不错；性能 ⇔ 可以 |

# 问答系统

- 问答系统（Question Answering，QA）
  - 用户以自然语言形式描述问题，从异构数据中获得答案
- 根据数据来源的不同，问答系统可以分为4种主要的类型
  - 检索式问答系统
    - 答案来源于固定的文本语料库或互联网，系统通过查找相关文档并抽取答案完成问答
  - 知识库问答系统
    - 回答问题所需的知识以数据库等结构化形式存储，问答系统首先将问题解析为结构化的查询语句，通过查询相关知识点，并结合知识推理获取答案
  - 常问问题集问答系统
    - 通过对历史积累的常问问题集合进行检索，回答用户提出的类似问题
  - 阅读理解式问答系统
    - 通过抽取给定文档中的文本片段或生成一段答案来回答用户提出的问题

# 更多任务

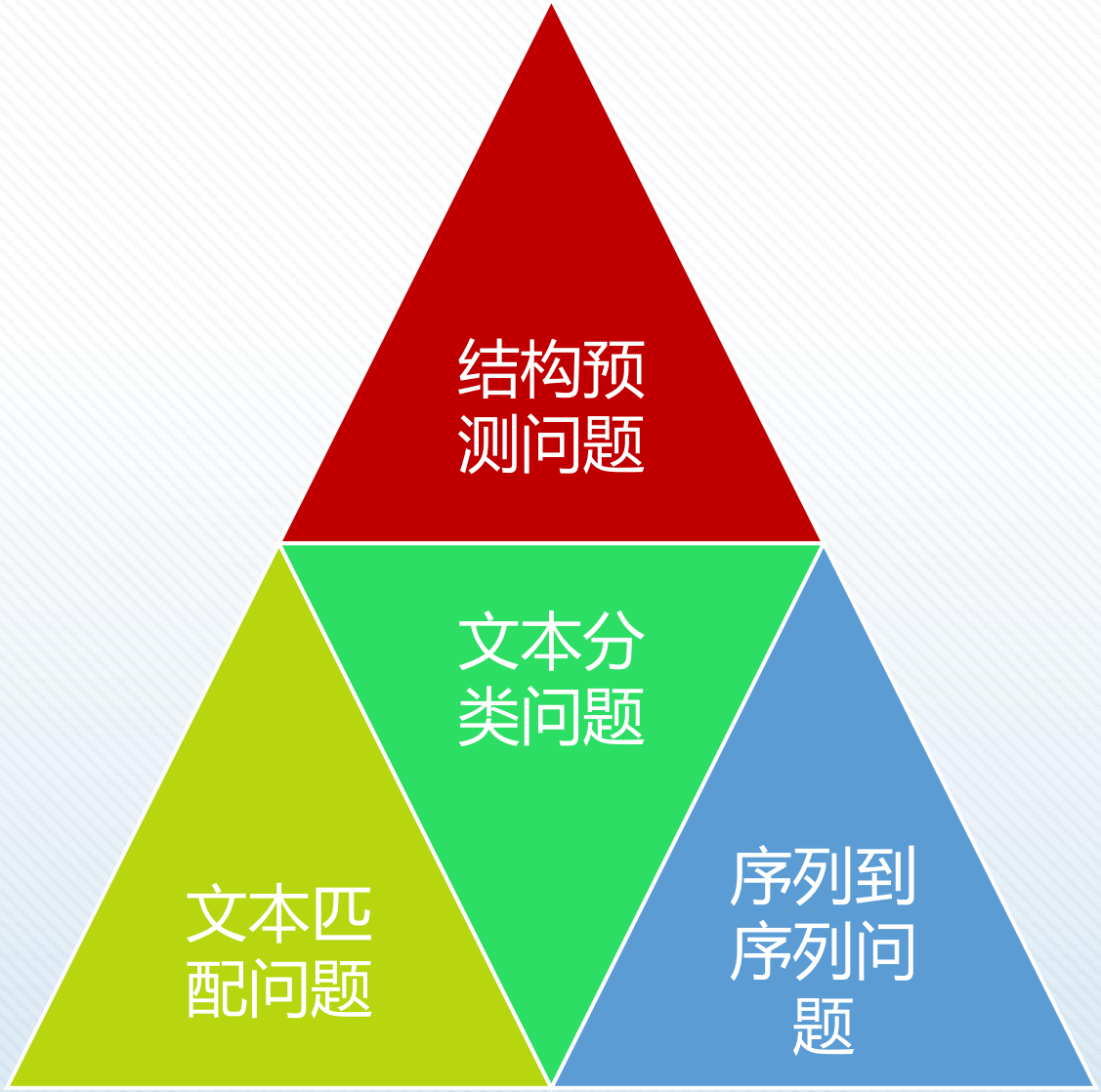
- 机器翻译（Machine Translation，MT）
- 对话系统（Dialogue System）

| | 任务型<br>Task | 聊天<br>Chat | 知识问答<br>Knowledge | 推荐<br>Recommen-dation |
|---|---|---|---|---|
| **目的** | 完成任务或动作 | 闲聊 | 知识获取 | 信息推荐 |
| **领域** | 特定域（垂类） | 开放域 | 开放域 | 特定域 |
| **以话轮数评价** | 越少越好 | 话轮越多越好 | 越少越好 | 越少越好 |
| **应用** | 虚拟个人助理 | 娱乐、情感陪护 | 客服、教育 | 个性化推荐 |
| **典型系统** | Siri、Cortana、Google Assistant、度秘 | 小冰、笨笨 | Watson、Wolfram Alpha | 阿里小蜜 |

# 目 录

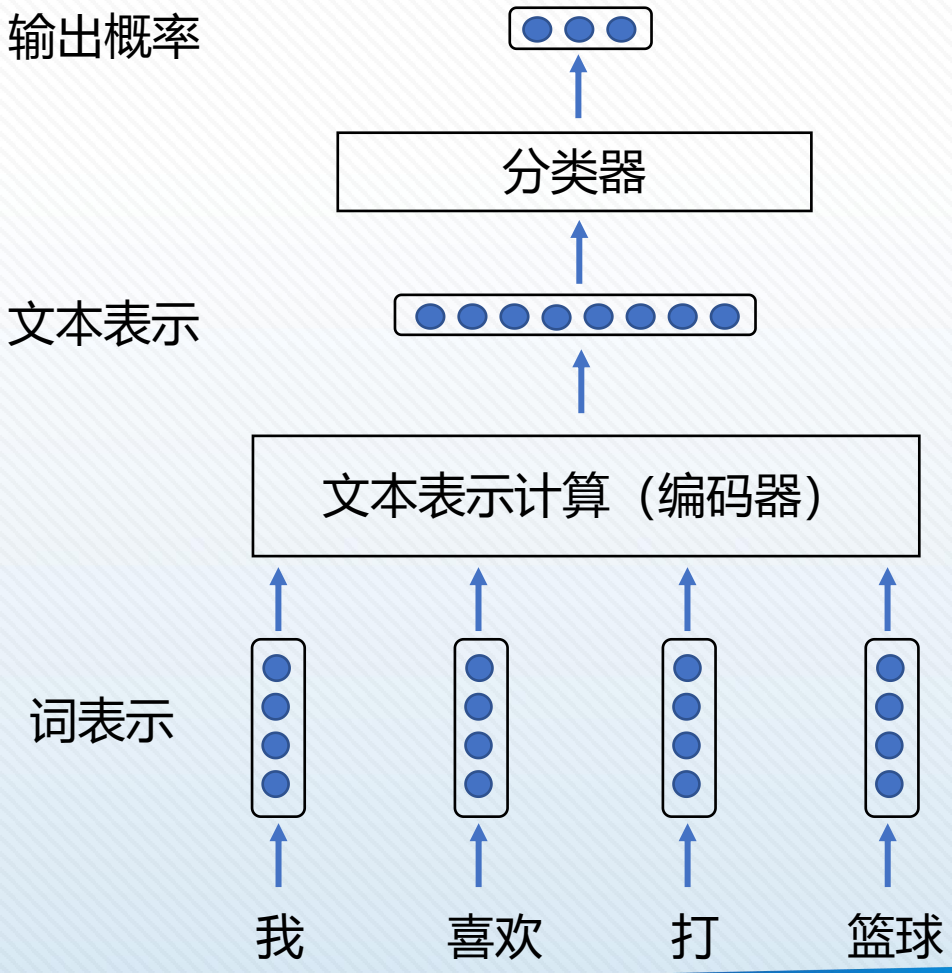
CONTENTS

# 归结为四个基本问题



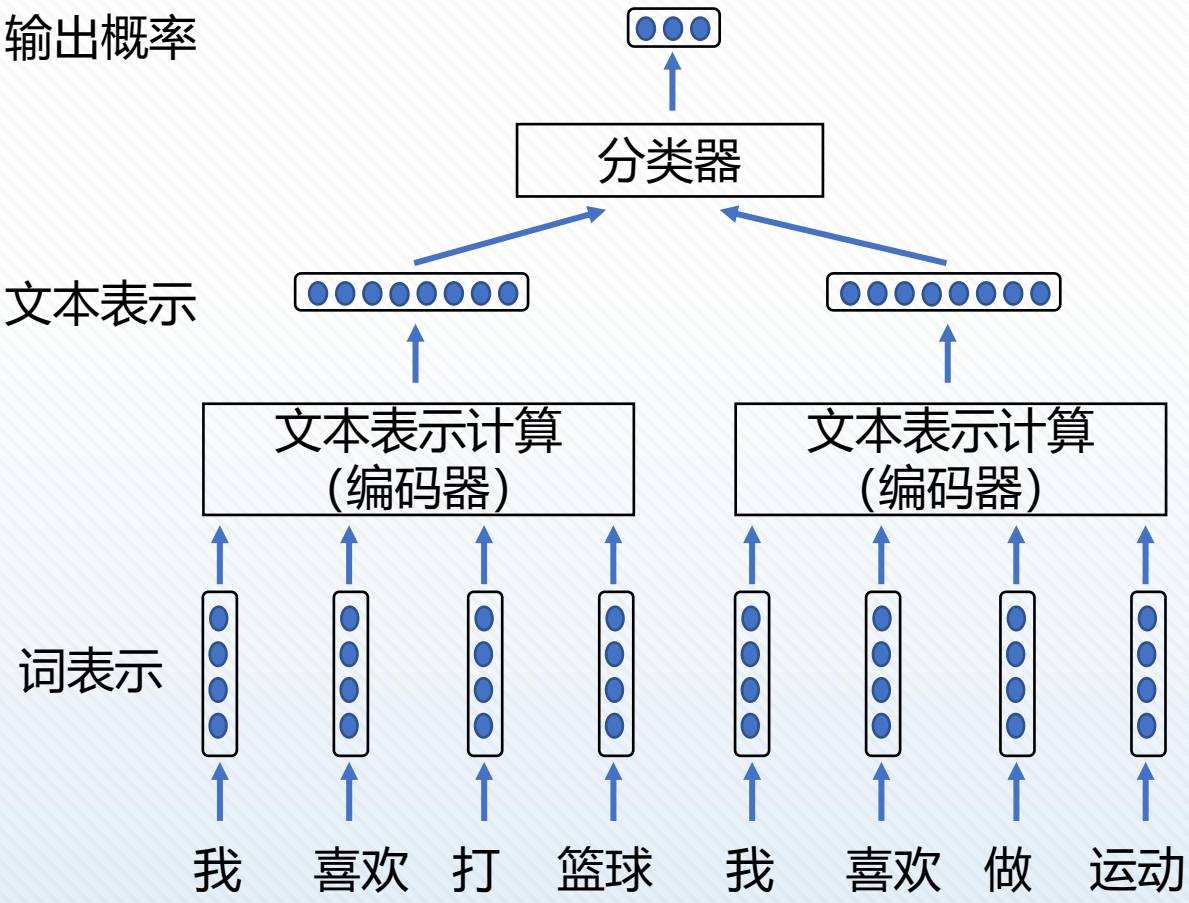
# 文本分类问题

- 将输入文本映射为所属类别（预定义的封闭集合）
- 最简单最基本的自然语言处理问题
- 应用场景
  - 垃圾邮件过滤、情感分类等
- 很多问题可以转化为文本分类问题



# 文本匹配问题

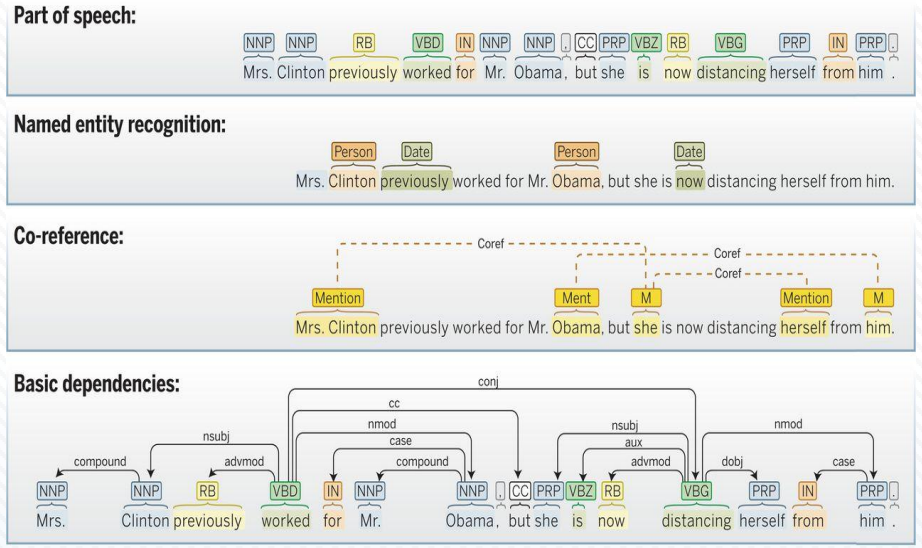
- 判断两段文本之间的匹配关系
  - 如复述关系、蕴含关系等
- 解决方案
  - 双塔结构
    - 两段文本分别通过两个模型映射为向量，然后判断两个向量之间的匹配关系
  - 单塔结构
    - 将两段文本直接拼接，然后进行匹配关系分类




双塔结构示意图

# 结构预测问题

- 输出类别之间具有较强的相互关联性
- 自然语言处理的本质问题
- 三种典型的结构预测问题
  - 序列标注
    - 为输入文本序列中的每个词标注相应的标签
    - 如词性标注：他/PN 喜欢/VV 下/VV 象棋/NN
  - 序列分割
    - 在文本序列中切分出子序列
    - 可以转化为序列标注问题
  - 图结构生成
    - 输入自然语言，输出以图表示的结构




Julia Hirschberg and Christopher D. Manning. Advances in Natural Language Processing. **Science** 2015

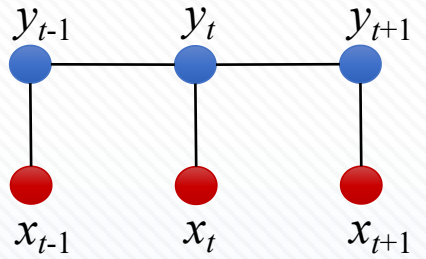
Science AAAS

| 输入 | 我 | 爱 | 北 | 京 | 天 | 安 | 门 | 。 |
|---|---|---|---|---|---|---|---|---|
| 分词输出 | B | B | B | I | B | I | I | B |
| 命名实体输出 | O | O | B-LOC | I-LOC | I-LOC | I-LOC | I-LOC | O |

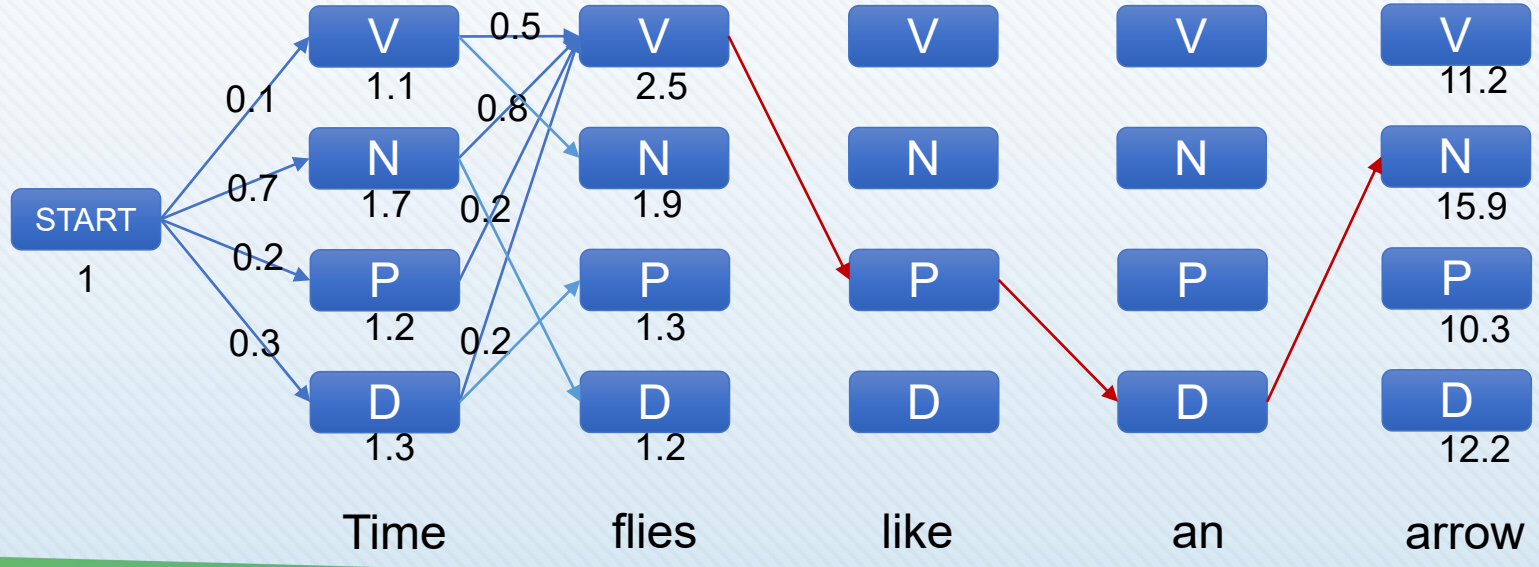
# 结构预测问题：序列标注

- 经典的序列标注模型
  - 条件随机场（Conditional Random Field，CRF）

CRF $$P(y_{[1:n]}|x_{[1:n]}) \propto \frac{1}{Z_{y_{[1:n]}}} \prod_{t=1}^{n} \exp\left(\begin{matrix}\sum_j \lambda_j f_j(y_t, y_{t-1}) \\ + \sum_k \mu_k g_k(y_t, x_t)\end{matrix}\right)$$

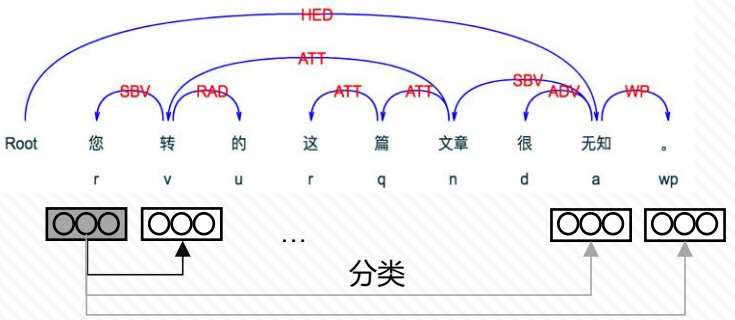



- 维特比（Viterbi）解码算法



# 结构预测问题：图结构生成

- 基于图的算法
  - 计算图中任意两个节点之间连边的分数（类别）
  - 根据解码算法生成图（树）结构




- 基于转移的算法
  - 将图结构的构建过程转化为状态转移序列
    - 如标准弧转移算法
    - 转移状态由一个栈和一个队列构成
    - 三种转移动作
      - 移进（SH）、左弧（RL）和右弧（RR）
  - 对旧状态执行一个移动作转换为新状态
  - 转移动作的选择本质上也是分类问题

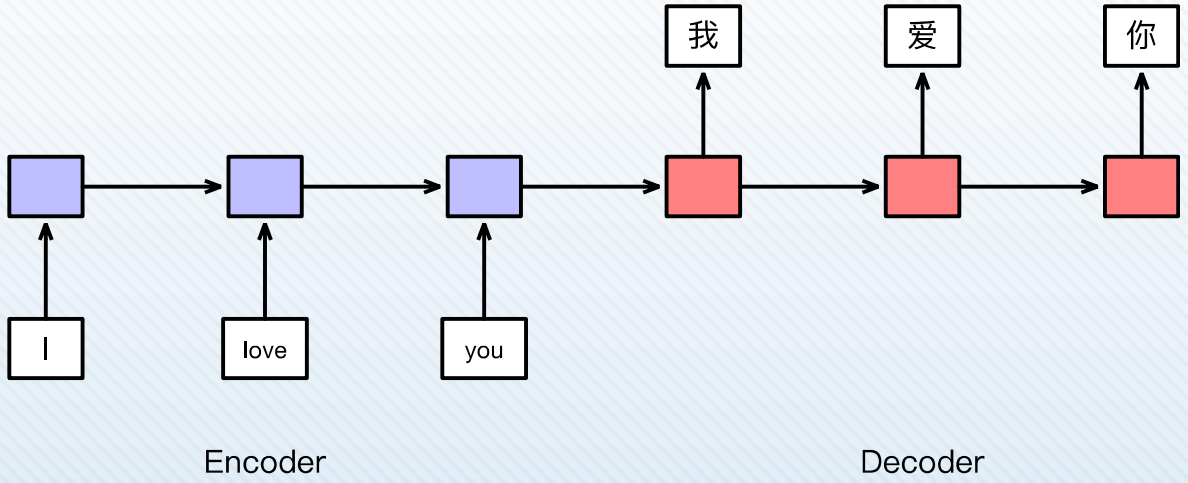
| 步骤 | 栈 | 队列 | 下一步动作 |
|---|---|---|---|
| 0 | | 他 喜欢 下 象棋 | SH |
| 1 | 他 | 喜欢 下 象棋 | SH |
| 2 | 他 喜欢 | 下 象棋 | RL |
| 3 | 喜欢<br>他 | 下 象棋 | SH |
| 4 | 喜欢 下<br>他 | 象棋 | SH |
| 5 | 喜欢 下 象棋<br>他 | | RR |
| 6 | 喜欢 下<br>他 象棋 | | RR |
| 7 | 喜欢<br>他 下<br>象棋 | | FIN |

面向依存句法分析的标准弧转移算法示例

# 序列到序列问题

- 将输入序列转换为输出序列
  - 输入和输出的序列不要求等长，也不要求词表一致
  - 泛化为“编码器—解码器”模型（Encoder-Decoder）
    - 本质上也是分类问题
- 典型任务

| 任务 | 输入 | 输出 |
| --- | --- | --- |
| 机器翻译 | 源语言 | 目标语言 |
| 文本摘要 | 原文 | 摘要 |
| 回复生成 | 用户语句 | 机器回复 |
| 图片描述生成 | 图片 | 文本描述 |
| 语音识别 | 语音 | 转写文本 |




“编码器—解码器”模型示例

# 目 录

CONTENTS

# 自然语言处理的评价

- 准确率（Accuracy）
  - 最简单直观的评价指标，常被用于文本分类、词性标注等问题

$$\mathrm{ACC}^{\mathrm{cls}} = \frac{\text{正确分类的文本数}}{\text{测试文本总数}}$$

$$\mathrm{ACC}^{\mathrm{pos}} = \frac{\text{正确标注的词数}}{\text{测试文本中词的总数}}$$

- F值
  - 针对某一类别的评价

$$\text{F 值} = \frac{(\beta^2+1)PR}{\beta^2(P+R)}$$

$$P = \frac{\text{正确识别的命名实体数目}}{\text{识别出的命名实体总数}}$$

$$R = \frac{\text{正确识别的命名实体数目}}{\text{测试文本中命名实体的总数}}$$

  - $\beta$是加权调和参数；$P$是精确率（Precision）；$R$是召回率（Recall）
  - 当权重为$\beta = 1$时，表示精确率和召回率同样重要，也称F1值

# 自然语言处理的评价

- 依存分析的评价
  - UAS（Unlabeled Attachment Score）
    - 词的父节点被正确识别的准确率
  - LAS（Labeled Attachment Score）
    - 词的父节点以及与父节点的句法关系都被正确识别的准确率
- 机器翻译的评价
  - BLEU（BiLingual Evaluation Understudy）值
    - 统计机器译文与参考译文（可以不只一个）中N-gram匹配的数目占机器译文中所有N-gram总数的比率
- 对话系统的评价
  - 由于回复的开放性，没有标准答案，很难自动评价
  - 由于对话的交互性，不能简单地通过一轮人机对话就对系统进行评价
  - 目前往往采用人工评价（流畅度、相关度、准确性等等）

# 谢 谢!

理解语言，认知社会

以中文技术，助民族复兴

长按二维码，关注哈工大SCIR

微信号：HIT_SCIR