第3章：文本表示

课件制作：曲彦儒、钱利华、屠可伟
讲解人：曲彦儒、钱利华

3.1 词的表示

如何表示文本

- 离散符号表示
  - 需要人工构建字典、辞典
- 稀疏向量表示
  - 词共现矩阵
- 稠密向量表示
  - 词嵌入（word embedding）
  - 上下文相关词嵌入（contextualized word embedding）

3.2 稀疏向量表示

输入文档

- 上下文（context）
  - 出现在一个词周围的文本内容。如果用滑动窗口表示上下文，一篇文档中的每一个词，它的上下文定义为它前后出现的固定个数的词
- 词-词共现矩阵
  - 例：小王子

	prince	planet	stars	flower	sheep	fox
prince	-	26	2	29	6	12
planet	26	-	2	8	4	0
stars	2	2	-	4	1	0
flower	29	8	4	-	10	0
sheep	6	4	1	10	-	0
fox	12	0	0	0	0	-

3.3 稠密向量表示

- 传统方法
  - 基于奇异值分解（SVD）的潜在语义分析（LSA）
- 近期方法
  - word2vec [1]：包含skip-gram、CBOW两种算法
  - GloVe [2]
- 最新方法
  - 上下文相关词嵌入

[1] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space.
[2] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation.

word2vec

- skip-gram
  - 对于训练文本中的每个滑动窗口，希望使用中心词尽可能正确地预测上下文词。即给定中心词，上下文中词的条件概率尽可能大
  - 给定中心词c，预测上下文中词o的条件概率计算如下

$$P(o|c) = \frac{\exp(u_o^{\mathrm{T}} v_c)}{\sum_{w \in V} \exp(u_w^{\mathrm{T}} v_c)}$$

  - 损失函数为整个训练语料的条件概率乘积的负对数

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le \mathrm{j} \le \mathrm{m}, j \ne 0} \log P(w_{t+j} | w_t; \theta)$$

  - 一般使用随机梯度下降来最小化该损失函数，并使用负采样法来快速训练

上下文相关词嵌入

- 在word2vec模型中，每个词的词嵌入在学习结束之后便固定住了，使用时不会随上下文（语境）的改变而改变，称为静态词嵌入
- 事实上在不同的上下文中，词的含义会发生变化
  - 如“bank”上下文中如果出现了“river”，那么表示“堤坝”的可能性较大，如果上下文出现了“money”，那么表示“银行”的可能性较大
- 上下文相关词嵌入
  - 给定文本，每个词的词嵌入会根据上下文内容，通过神经网络变换为上下文相关词嵌入，具备更强的表达能力
  - 后续章节会详细介绍

总结

文本表示总结

- 离散符号表示
- 稀疏向量表示
  - 独热编码
  - 词共现矩阵
- 稠密向量表示
  - word2vec：skip-gram算法
  - 上下文相关词嵌入