Natural Language Processing I

10. 词嵌入

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

2022/04/22

“喵说喵要喵”

> *เสด็จ ให้มาทูลถามเสด็จว่า จะเสด็จหรือไม่เสด็จ ถ้าเสด็จจะเสด็จ เสด็จจะเสด็จด้วย*
> *–《四朝代》*

- 泰语：针对皇室成员的称谓、动作情态和事物需要使用特殊的“皇语”

> ***皇后娘娘****遣（我）来请示****公主殿下****，****公主殿下****是否****去****（大殿听经），如果****公主殿下****打算****去****（听经），****皇后娘娘****也一同****去****（听经）。*

“喵说喵要喵”

> *เสด็จ ให้มาทูลถามเสด็จว่า จะเสด็จหรือไม่เสด็จ ถ้าเสด็จจะเสด็จ เสด็จจะเสด็จด้วย*
> *–《四朝代》*

- 泰语：针对皇室成员的称谓、动作情态和事物需要使用特殊的“皇语”

> ***皇后娘娘****遣（我）来请示****公主殿下****，****公主殿下****是否****去****（大殿听经），如果****公主殿下****打算****去****（听经），****皇后娘娘****也一同****去****（听经）。*

如果用“喵”字来替代这个词的话……

> ***喵****让我来问****喵****，****喵****要****喵****吗？如果****喵****要****喵****的话，****喵****也****喵****。*
> ***喵****说****喵****要****喵****。如果****喵****也****喵****，****喵****会很高兴。*

Word2vec

稀疏表示与信息压缩

TF-IDF和PPMI模型：**稀疏**向量、**矩阵**存储

- 元素：计数或计数的函数，大部分是0，**无有效信息**
  - 0值表示**不相关**，即同时出现的可能性为0

稀疏表示与信息压缩

TF-IDF和PPMI模型：**稀疏**向量、**矩阵**存储

- 元素：计数或计数的函数，大部分是0，**无有效信息**
  - 0值表示**不相关**，即同时出现的可能性为0
- 长向量：维数是语料集中的词汇数$|V|$，或文档数
  - 《现代汉语常用词表》2015年版：56008个词语

稀疏表示与信息压缩

TF-IDF和PPMI模型：**稀疏**向量、**矩阵**存储

- 元素：计数或计数的函数，大部分是0，**无有效信息**
  - 0值表示**不相关**，即同时出现的可能性为0
- 长向量：维数是语料集中的词汇数$|V|$，或文档数
  - 《现代汉语常用词表》2015年版：56008个词语

长向量导致超大矩阵：计算复杂度$O(|V|^2)$平方级增长

- **稀疏表示**：链表的变种，利用稀疏性

稀疏表示与信息压缩

TF-IDF和PPMI模型：**稀疏**向量、**矩阵**存储

- 元素：计数或计数的函数，大部分是0，**无有效信息**
  - 0值表示**不相关**，即同时出现的可能性为0
- 长向量：维数是语料集中的词汇数$|V|$，或文档数
  - 《现代汉语常用词表》2015年版：56008个词语

长向量导致超大矩阵：计算复杂度$O(|V|^2)$平方级增长

- **稀疏表示**：链表的变种，利用稀疏性
- 将信息压缩成稠密（的向量）表示
  - 称为**词嵌入 embedding**

词嵌入

词嵌入（向量）的元素：实数值

- 信息压缩通常导致**信息丢失**：$f: x \mapsto v$
  - 类比JPEG2000（小波变换）：切除高频信号，故多次保存后更模糊

词嵌入

词嵌入（向量）的元素：实数值

- 信息压缩通常导致**信息丢失**：$f: x \mapsto v$
  - 类比JPEG2000（小波变换）：切除高频信号，故多次保存后更模糊

- 缺点：很难对维度做出解释
  - 压缩相当于**编码**过程：词义随之消除
  - 例如 Morse 密码：4个“`H: ....`”可以压缩成“`16.`”
    - 但这无法从代码表找到解释

词嵌入

词嵌入（向量）的元素：实数值

- 信息压缩通常导致**信息丢失**：$f: x \mapsto v$
  - 类比JPEG2000（小波变换）：切除高频信号，故多次保存后更模糊

- 缺点：很难对维度做出解释
  - 压缩相当于**编码**过程：词义随之消除
  - 例如 Morse 密码：4个“`H: ....`”可以压缩成“`16.`”
    - 但这无法从代码表找到解释
- 对比词向量：特征维度有明确（以文档/词度量的）词义

词嵌入的优点

在所有NLP任务中都比稀疏表示的效能好（原因尚在研究）

- 短向量 = 特征少：**参数少**，易于训练
  - 参数空间小：过拟合问题相对轻，泛化性好

词嵌入的优点

在所有NLP任务中都比稀疏表示的效能好（原因尚在研究）

- 短向量 = 特征少：**参数少**，易于训练
  - 参数空间小：过拟合问题相对轻，泛化性好
- **低维度量**更有效：避免“维数灾难”，更容易发现词义相似的词
  - 例如：稀疏表示中`car`和`automobile`的维数差异大
    - ⇒ 词义完全无关，与经验不符

常用词嵌入方法

静态嵌入：只计算一次词汇的固定嵌入表示

- NLP 启发的模型
  - word2vec (skipgram, CBOW), GloVe

常用词嵌入方法

静态嵌入：只计算一次词汇的固定嵌入表示

- NLP 启发的模型
  - word2vec (skipgram, CBOW), GloVe
- 矩阵分析、计算：如SVD
  - Latent Semantic Analysis (LSA)

常用词嵌入方法

静态嵌入：只计算一次词汇的固定嵌入表示

- NLP 启发的模型
  - word2vec (skipgram, CBOW), GloVe
- 矩阵分析、计算：如SVD
  - Latent Semantic Analysis (LSA)

动态嵌入：上下文不同，嵌入也不同

- ELMo、BERT

可在线下载的静态嵌入

[Mikolov 2013] word2vec

- https://code.google.com/archive/p/word2vec/
- 非常流行，训练速度快

[Pennington 2014] GloVe

- http://nlp.stanford.edu/projects/glove/

word2vec：原理

预测，而不是计数

- 训练一个分类器：“食堂”在“下课”的上下文出现的**可能性**是多少？
- 对比之前：出现的次数是多少？简单假定统计词频为可能性

word2vec：原理

预测，而不是计数

- 训练一个分类器：“食堂”在“下课”的上下文出现的**可能性**是多少？
- 对比之前：出现的次数是多少？简单假定统计词频为可能性

但是，我们并不关心分类的结果

- 将分类器**学到的权重**作为词嵌入，即当作**编码器**

自监督

词嵌入对机器学习的变革性贡献

- 文本的序列关系：天然可以作为训练数据的监督标注
  - 上下文可以看成**隐式标签**

他下课后去了食堂
他下课后去了□□

自监督

词嵌入对机器学习的变革性贡献

- 文本的序列关系：天然可以作为训练数据的监督标注
  - 上下文可以看成**隐式标签**

他下课后去了食堂
他下课后去了□□

这在机器学习分类中称为**自监督 self-supervision 学习**

- 避免了人工标定监督关系的步骤
- [Bengio 2003, Collobert 2011] 使用文本序列本身学习词嵌入

word2vec 是简化方案

word2vec 比基于神经网络的语言模型简单得多

- 任务简化：转化成**二分类问题**
  - 只需预测词出现的可能性，而非预测选择哪个词

word2vec 是简化方案

word2vec 比基于神经网络的语言模型简单得多

- 任务简化：转化成**二分类问题**
  - 只需预测词出现的可能性，而非预测选择哪个词
- 架构简化：只需逻辑回归分类器，而非复杂神经网络模型

word2vec 是简化方案

word2vec 比基于神经网络的语言模型简单得多

- 任务简化：转化成**二分类问题**
  - 只需预测词出现的可能性，而非预测选择哪个词
- 架构简化：只需逻辑回归分类器，而非复杂神经网络模型

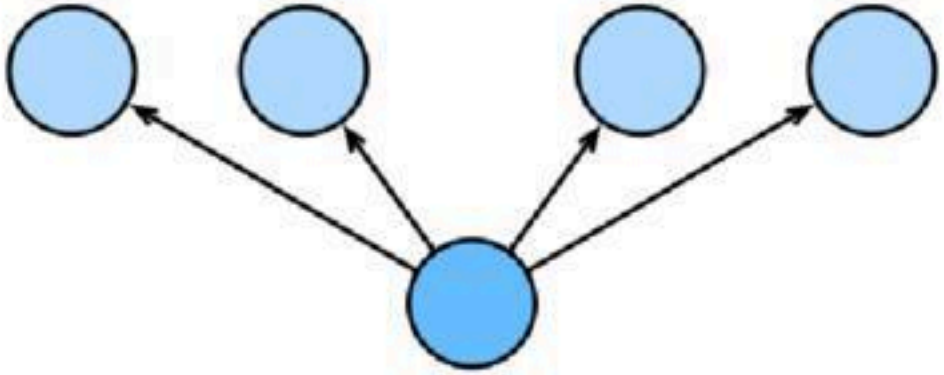
我们主要讨论word2vec其中的一个版本

- [Mikolov 2013] **skip-gram with negative sampling (SGNS)**
  - skip-gram也称**跳元模型**

skip-gram：工作流程

为了预测词c是目标词w的上下文

1. 将词对(w, c)看成一个**正例**
2. 随机采样词w'，并将词对(w', c)看成一个**负例**
3. 使用逻辑回归训练一个**分类器**
4. 将分类器学到的**权重**用作词嵌入



skip-gram：示例

考虑如下文本：假定使用大小为±2的上下文窗口

如	[今	人	方	为	刀]	俎
	c_1	c_2	**w**	c_3	c_4	

skip-gram：示例

考虑如下文本：假定使用大小为±2的上下文窗口

如	[今	人	方	为	刀]	俎
	c_1	c_2	**w**	c_3	c_4	

分类任务的构造：$P(+|w,c)$

- 词对：(方, 今)，(方, 俎)，(方, 鱼)
- $P(-|w,c) = 1 - P(+|w,c)$

skip-gram：原理

基于**相似度**的概率值

- 嵌入向量的相似度高：词对共同出现的可能性高

skip-gram：原理

基于**相似度**的概率值

- 嵌入向量的相似度高：词对共同出现的可能性高
- 向量相似：内积数值大
  - 余弦值是**规范化**内积：取值$[-1, 1]$

skip-gram：原理

基于**相似度**的概率值

- 嵌入向量的相似度高：词对共同出现的可能性高
- 向量相似：内积数值大
  - 余弦值是**规范化**内积：取值$[-1, 1]$

因此：$P(+|w, c) \propto \mathbf{c} \cdot \mathbf{w}$

skip-gram：原理

基于**相似度**的概率值

- 嵌入向量的相似度高：词对共同出现的可能性高
- 向量相似：内积数值大
  - 余弦值是**规范化**内积：取值$[-1, 1]$

因此：$P(+|w, c) \propto \mathbf{c} \cdot \mathbf{w}$

- 但是，转化成概率值需要**归一化**
  - 余弦值也不是概率：如何将数值挤压到$[0, 1]$?

skip-gram：概率构造

回忆：**S形曲线**恰好有将数值挤压到$[0, 1]$的性质

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

skip-gram：概率构造

回忆：**S形曲线**恰好有将数值挤压到$[0,1]$的性质

$$\sigma(x)=\frac{1}{1+\exp(-x)}$$

挤压相似度数值：

$$P(+|w,c)=\sigma(\mathbf{c}\cdot\mathbf{w})=\frac{1}{1+\exp(-\mathbf{c}\cdot\mathbf{w})}$$

$$P(-|w,c)=1-P(+|w,c)=\frac{1}{1+\exp(\mathbf{c}\cdot\mathbf{w})}$$

skip-gram：多个上下文

skip-gram 假设所有上下文词是**相互独立**的

$$P(+|w, c_{1:L}) = \prod_i \sigma(\mathbf{c}_i \cdot \mathbf{w})$$

- 这些词的解耦合可以由**分词预处理**来保证

他 下课后 去了 食堂

skip-gram：多个上下文

skip-gram 假设所有上下文词是**相互独立**的

$$P(+|w, c_{1:L}) = \prod_i \sigma(\mathbf{c}_i \cdot \mathbf{w})$$

- 这些词的解耦合可以由**分词预处理**来保证

他 下课后 去了 食堂

- 取对数：避免多个小数相乘

$$\log P(+|w, c_{1:L}) = \sum_i \log \sigma(\mathbf{c}_i \cdot \mathbf{w})$$

skip-gram：小结

skip-gram 构造了一个概率二分类器

- 输入：目标词$\mathbf{w}$，上下文$c_{1:L}$

skip-gram：小结

skip-gram 构造了一个概率二分类器

- 输入：目标词$\mathbf{w}$，上下文$c_{1:L}$

$\mathbf{w}$在上下文窗口的概率：由向量相似度估计

- 基于假设：相关联的词出现在相同上下文
- 计算前提：需要所有词的嵌入向量

skip-gram：小结

skip-gram 构造了一个概率二分类器

- 输入：目标词$\mathbf{w}$，上下文$c_{1:L}$

$\mathbf{w}$在上下文窗口的概率：由向量相似度估计

- 基于假设：相关联的词出现在相同上下文
- 计算前提：需要所有词的嵌入向量

那么，如何学到这些词嵌入？

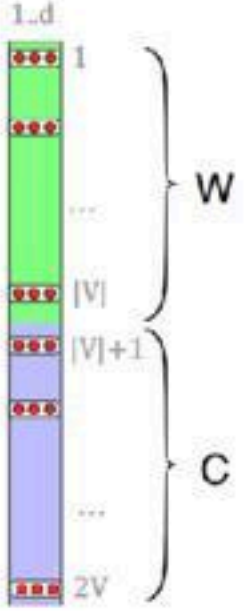
词嵌入集合

词嵌入的列表：即分类器的**参数集合**

- $f : x \mapsto v$

skip-gram 对每个词存储**两个词嵌入**

- 同一个词、两种表示
  - 分别用作目标、上下文



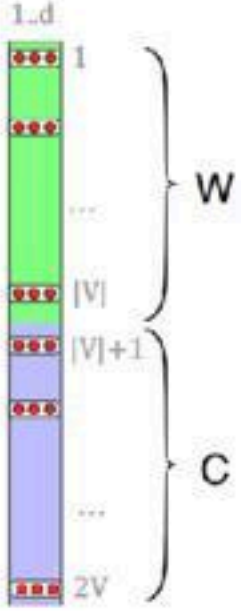
词嵌入集合

词嵌入的列表：即分类器的**参数集合**

- $f : x \mapsto v$

skip-gram 对每个词存储**两个词嵌入**

- 同一个词、两种表示
  - 分别用作目标、上下文



下面讨论如何学到这些词嵌入（训练分类器的真正目的）

Word2vec：学习词嵌入

skip-gram：负例

如	[今	人	方	为	刀]	俎
	c_1	c_2	**w**	c_3	c_4	

- 对每个正例：根据词频采样k个负例

skip-gram：负例

如	[今	人	方	为	刀]	俎
	c_1	c_2	**w**	c_3	c_4	

- 对每个正例：根据词频采样k个负例

正例 +

w	c_{pos}
方	今
方	人
方	为
方	刀

负例 -：$k = 2$

w	c_{neg}	w	c_{neg}
方	鱼	方	俎
方	攻	方	仁
方	饥	方	商
方	美	方	谋

- 这也是名称的来历：**Skip-Gram with Negative Sampling (SGNS)**

skip-gram：学习目标

初始状态：随机嵌入向量

学习过程逐渐达成**两个目标**，即目标损失函数的依据

- 最大化正例词对的相似度
- 最小化负例词对的相似度

skip-gram：训练模型

损失函数

$$
\begin{aligned}
L_{CE} &= -\log\left[P(+|w,c_{pos})\prod_{i=1}^{k}P(-|w,c_{neg}^{i})\right] \\
&= -\left[\log P(+|w,c_{pos}) + \sum_{i=1}^{k}\log P(-|w,c_{neg}^{i})\right] \\
&= -\left[\log P(+|w,c_{pos}) + \sum_{i=1}^{k}\log(1-P(+|w,c_{neg}^{i}))\right] \\
&= -\left[\log \sigma(c_{pos}\cdot w) + \sum_{i=1}^{k}\log \sigma(-c_{neg}\cdot w)\right]
\end{aligned}
$$

优化器：随机梯度下降

skip-gram：损失函数求导

$$L_{CE} = -\left[\log \sigma(c_{pos} \cdot w) + \sum_{i=1}^{k} \log \sigma(-c_{neg} \cdot w)\right]$$

$$\frac{\partial L_{CE}}{\partial c_{pos}} = [\sigma(c_{pos} \cdot w) - 1]w$$

$$\frac{\partial L_{CE}}{\partial c_{neg}} = [\sigma(c_{neg} \cdot w)]w$$

$$\frac{\partial L_{CE}}{\partial w} = [\sigma(c_{pos} \cdot w) - 1]c_{pos} + \sum_{i=1}^{k} [\sigma(c_{neg}^{i} \cdot w)]c_{neg}^{i}$$

最终词嵌入

回顾：skip-gram 对每个词存储**两个词嵌入**

- 分别是目标词嵌入$\mathbf{w}_i$、上下文词嵌入$\mathbf{c}_i$

最终词嵌入

回顾：skip-gram 对每个词存储**两个词嵌入**

- 分别是目标词嵌入$\mathbf{w}_i$、上下文词嵌入$\mathbf{c}_i$

通常使用两者之和：$\mathbf{w}_i + \mathbf{c}_i$

- 也有算法只使用目标词嵌入$\mathbf{w}_i$

实验：Penn Tree Bank（PTB）

实验：skip-gram

word2vec (skip-gram) 训练：小结

1. 随机初始化词嵌入

2. 根据嵌入向量间相似度训练二分类器

- 将上下文窗口中出现的词对作为正例
- 对每个正例：根据词频采样k个负例
- 训练过程：最大化正例相似度，最小化负例相似度
- 抛弃掉分类器代码，**只保留词嵌入**：类比“微调网络”

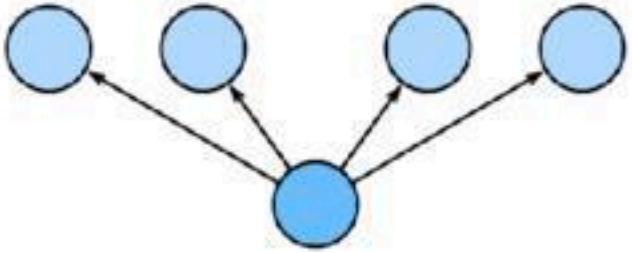
其他静态嵌入

连续词袋

[Mikolov 2013] **连续词袋 CBOW**模型类似于skip-gram

skip-gram

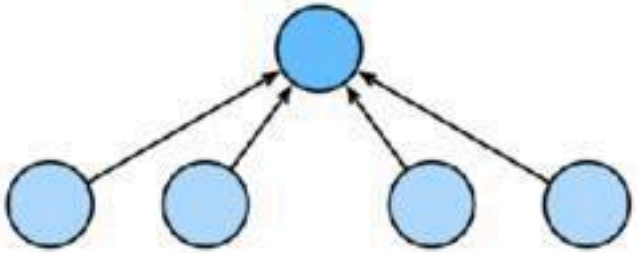
- 假设中心词生成上下文



- $P(今,人,为,刀|方) = P(今|方)P(人|方)P(为|方)P(刀|方$

CBOW

- 假设上下文生成中心词



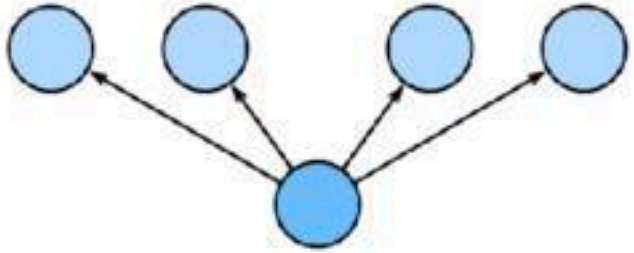
- $P(方|今,人,为,刀)$

连续词袋

[Mikolov 2013] **连续词袋 CBOW**模型类似于skip-gram

skip-gram

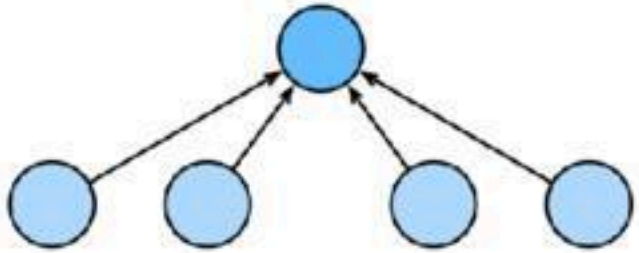
- 假设中心词生成上下文



- $P(\text{今},\text{人},\text{为},\text{刀}|\text{方}) = P(\text{今}|\text{方})P(\text{人}|\text{方})P(\text{为}|\text{方})P(\text{刀}|\text{方}$

CBOW

- 假设上下文生成中心词



- $P(\text{方}|\text{今},\text{人},\text{为},\text{刀})$

两者训练流程相同，仅公式计算有区别

全局向量的词嵌入

[Pennington 2014] **全局向量的词嵌入 GloVe**：预先提取语料集的全局统计信息

- 以 w_i 为中心词的上下文可能有多个
- **重数** x_{ij} ： 上下文中的词 w_j 与 w_i 共现的全局计数

全局向量的词嵌入

[Pennington 2014] **全局向量的词嵌入 GloVe**：预先提取语料集的全局统计信息

- 以 w_i 为中心词的上下文可能有多个
- **重数** x_{ij} ： 上下文中的词 w_j 与 w_i 共现的全局计数

（带全局语料统计的）skip-gram损失函数：$-\sum_{ij} x_{ij} \log P(w_j|w_i)$

全局向量的词嵌入

[Pennington 2014] **全局向量的词嵌入 GloVe**：预先提取语料集的全局统计信息

- 以 w_i 为中心词的上下文可能有多个
- **重数** x_{ij} ： 上下文中的词 w_j 与 w_i 共现的全局计数

（带全局语料统计的）skip-gram损失函数：$-\sum_{ij} x_{ij} \log P(w_j|w_i)$

GloVe损失函数：

$$\sum_{ij} h(x_{ij}) \left(w_j \cdot w_i + b_i + c_j - \log x_{ij}\right)^2$$

- 平方损失
- 中心词偏置 b_i 和上下文词偏置 c_i
- 权重函数 $h(x_{ij})$： $h(x)$ 在$[0, 1]$递增

fastText模型

[Bojanowski 2017] fastText模型是一种**子词嵌入**方法

- 子词是基于单字符的N元语法
  - 可以被认为是子词级skip-gram

fastText模型

[Bojanowski 2017] fastText模型是一种**子词嵌入**方法

- 子词是基于单字符的N元语法
  - 可以被认为是子词级skip-gram

例如单词“where”

```
“<wh”、“whe”、“her”、“ere”、“re>”、“<where>”
```

fastText模型

[Bojanowski 2017] fastText模型是一种**子词嵌入**方法

- 子词是基于单字符的N元语法
  - 可以被认为是子词级skip-gram

例如单词“where”

```
“<wh”、“whe”、“her”、“ere”、“re>”、“<where>”
```

开源：https://fasttext.cc

- 解决测试集中未知词<unk>的问题
- 解决词变形很多、在文本中罕见的问题

词嵌入的语义属性

上下文窗口

窗口大小决定上下文词汇，进而决定语言模型的**词汇分布**

- 小窗口$C = \pm 2$：在同一类别中，**语义相近、词类相同**的词
  - “刘备”：“关羽”、“诸葛亮”
  - [Levy 2014] Hogwarts: Sunnydale, Evernight

上下文窗口

窗口大小决定上下文词汇，进而决定语言模型的**词汇分布**

- 小窗口$C = \pm 2$：在同一类别中，**语义相近、词类相同**的词
  - “刘备”：“关羽”、“诸葛亮”
  - [Levy 2014] Hogwarts: Sunnydale, Evernight

- 大窗口$C = \pm 5$：**主题相关，但语义并不相似**的词
  - “刘备”：“皇叔”、“蜀汉”
  - Hogwarts: Dumbledore, Malfoy, half-blood

两类关联关系

[Schütze 1993] 区别两类共生关系：关联度

- **一阶共生**：也称**横组合 syntagmatic 关联**，经常成对出现的词
  - 写：文章，作业

两类关联关系

[Schütze 1993] 区别两类共生关系：关联度

- **一阶共生**：也称**横组合 syntagmatic 关联**，经常成对出现的词
  - 写：文章，作业
- **二阶共生**：也称**纵聚合 paradigmatic 关联**，有相似的共同邻居
  - 写：发表，批改

类比相似度

[Rumelhart 1973] **平行四边形 parallelogram 模型**：词义的类比

- 认知、发展心理学：**语言是词义理解的外在表现**，例如“费曼学习法”

```
apple:tree::grape:?
```

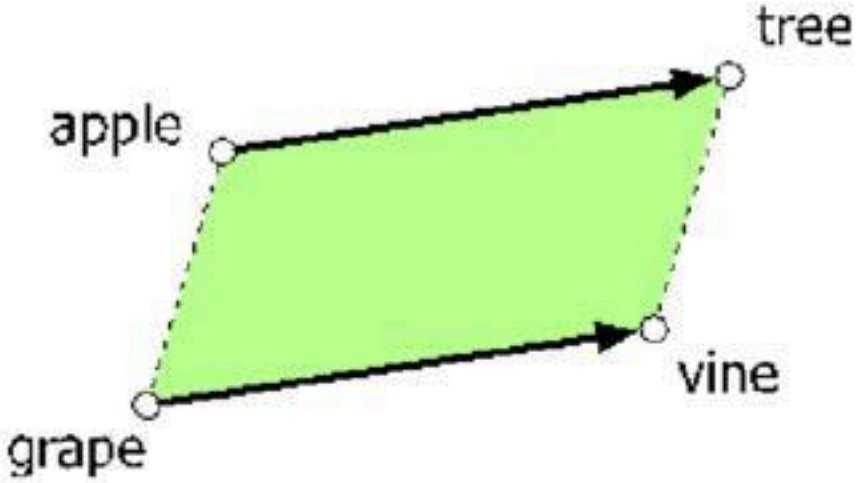
类比相似度

[Rumelhart 1973] **平行四边形 parallelogram 模型**：词义的类比

- 认知、发展心理学：**语言是词义理解的外在表现**，例如“费曼学习法”

```
apple:tree::grape:?
```

- 可用（向量）平行四边形法则找出目标词汇（的大概位置）



平行四边形法则

平行四边形法则可以用于**类比问题**

- [Turney 2005, Mikolov 2013] 对稀疏、稠密词嵌入都有效

```
王:男::后:?
中国:北京::法国:?
```

平行四边形法则

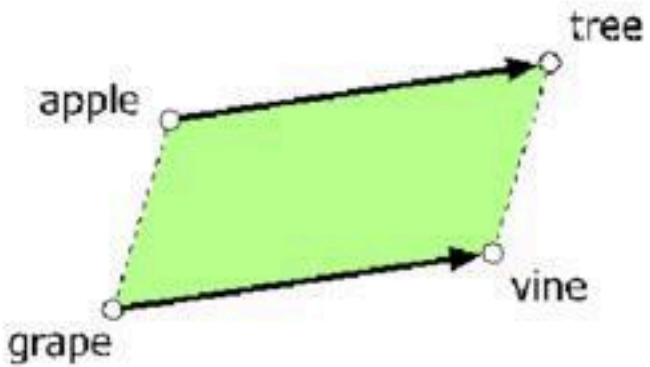
平行四边形法则可以用于**类比问题**

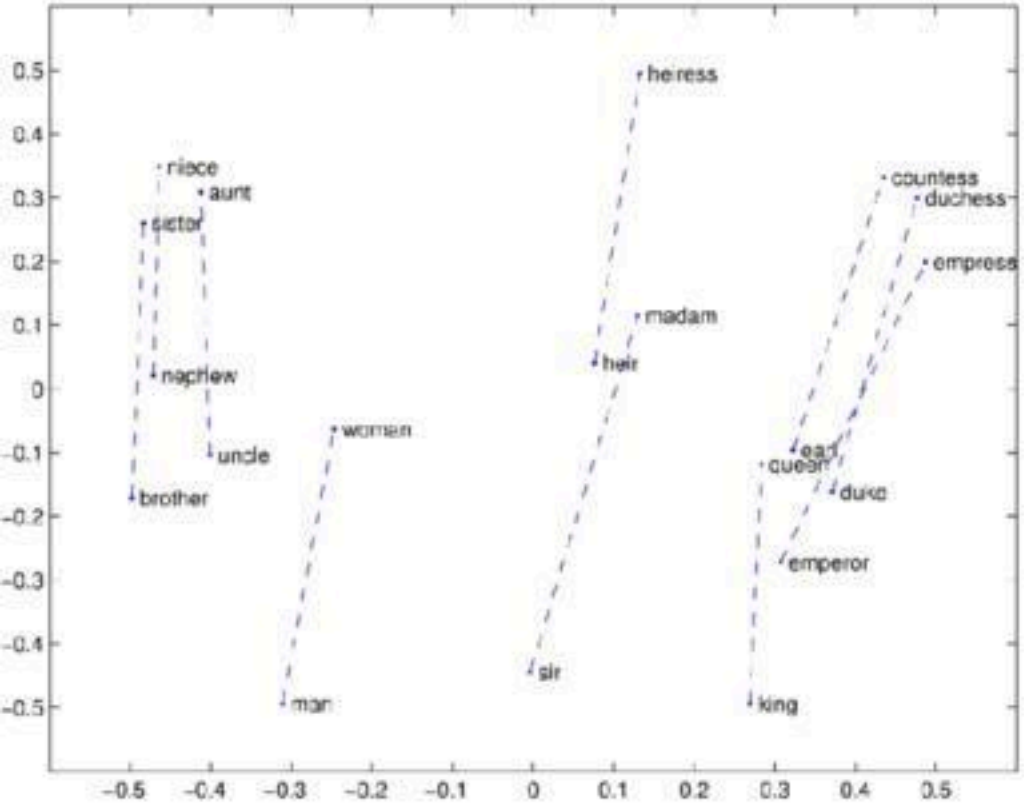
- [Turney 2005, Mikolov 2013] 对稀疏、稠密词嵌入都有效

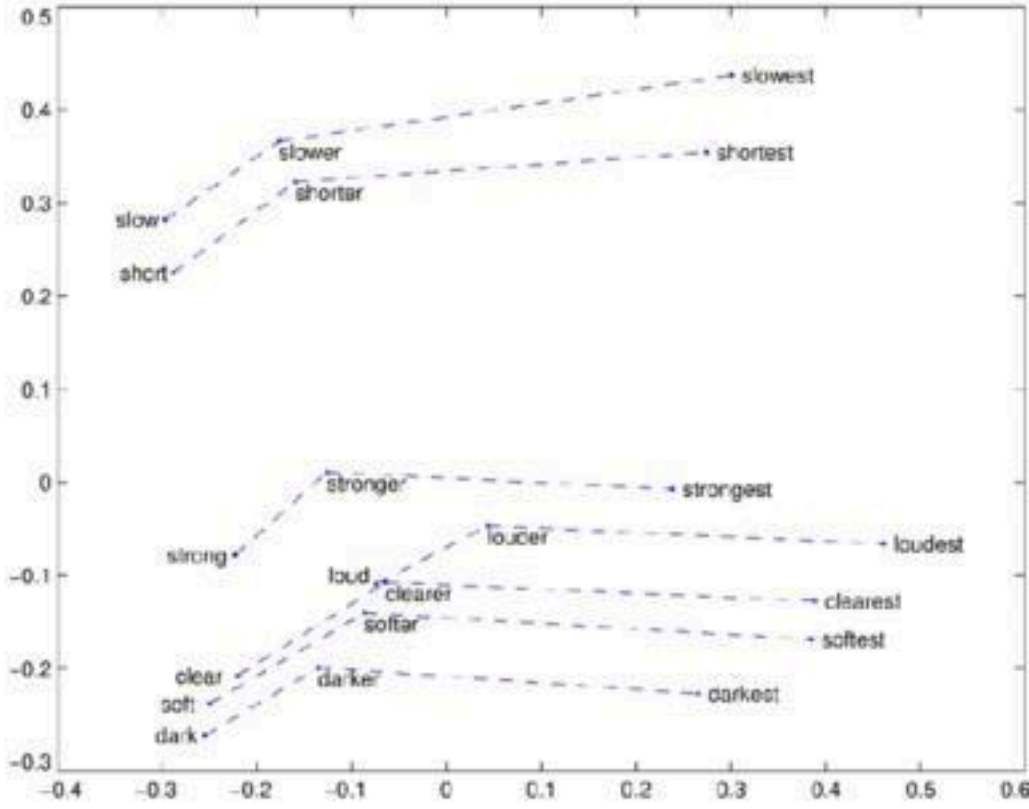
王:男::后:?
中国:北京::法国:?

$a:b::a':b^*$问题的一般公式：求解优化问题

$$\hat{b}^* = \arg\min_x \Delta(x, b - a + a')$$




niece
aunt
sister
nephew
uncle
brother
women
heiress
madam
heir
sir
man
countess
duchess
empress
queen
duke
emperor
king


slowest
slower
shortest
shorter
slow
short
stronger
strongest
louder
loudest
strong
loud
clearer
clearest
softer
softest
clear
soft
dark
darker
darkest

补充说明

$a : b :: a' :?:$ 经常会返回输入的三个词，或它们的变形

- 解决方法：**词形还原**，并去除

补充说明

$a : b :: a' :?$：经常会返回输入的三个词，或它们的变形

- 解决方法：**词形还原**，并去除

通常对高频词比较有效，此外两种有效的特例：

- 向量间距离短：维数灾难导致**距离度量失效**
  - 维数还不能太低（否则不足以区分语义）
- 特殊类比：国家首都、词类明确

补充说明

$a : b :: a' :?$：经常会返回输入的三个词，或它们的变形

- 解决方法：**词形还原**，并去除

通常对高频词比较有效，此外两种有效的特例：

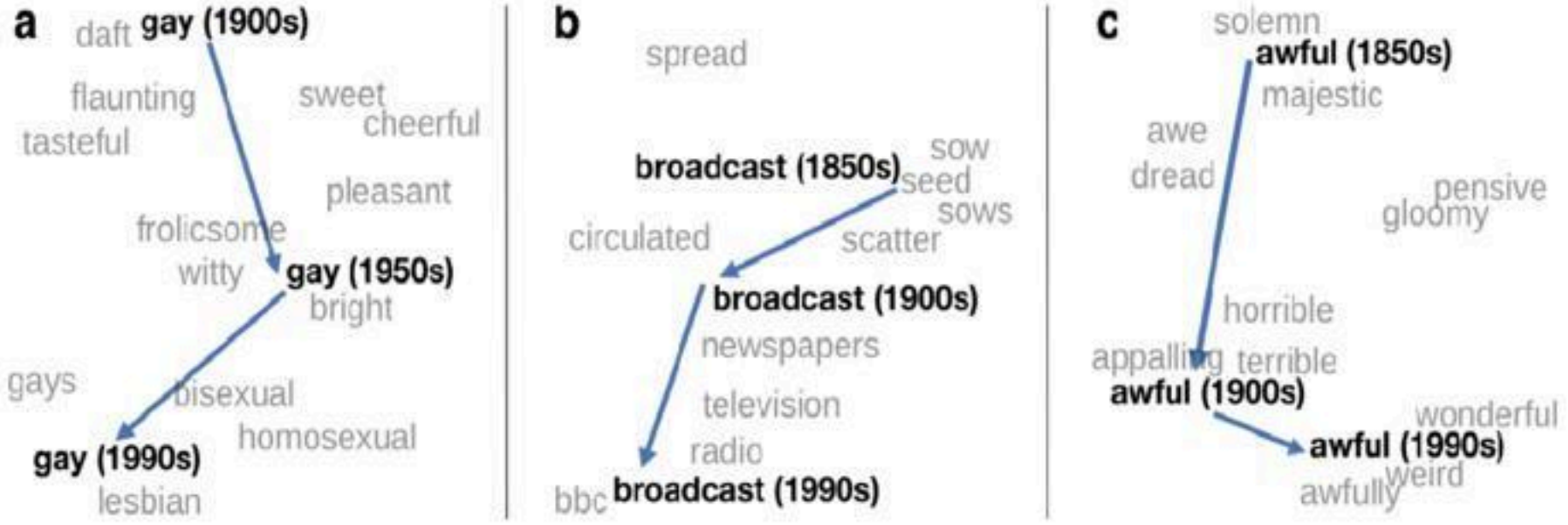
- 向量间距离短：维数灾难导致**距离度量失效**
  - 维数还不能太低（否则不足以区分语义）
- 特殊类比：国家首都、词类明确

[Peterson 2020] 人类对类比的认知过程（因果推断）仍然是开放研究

应用：语义演化

[Hamilton 2016] 对不同时代的历史文档训练词嵌入

- ~30 million books, 1850-1990, Google Books data



应用：偏见、歧视问题

[Bolukbasi 2016] 学到的词嵌入带有训练文本中的偏见

```
Q: 父亲:医生::母亲:?
A: 护士
```

应用：偏见、歧视问题

[Bolukbasi 2016] 学到的词嵌入带有训练文本中的偏见

```
Q: 父亲:医生::母亲:?
A: 护士
```

[Crawford 2017, Blodgett 2020] **错配问题 allocational harm**

- 例如：招聘系统对性别的歧视性过滤，保险公司对人群的歧视性保费

应用：偏见、歧视问题

[Bolukbasi 2016] 学到的词嵌入带有训练文本中的偏见

```
Q: 父亲:医生::母亲:?
A: 护士
```

[Crawford 2017, Blodgett 2020] **错配问题 allocational harm**

- 例如：招聘系统对性别的歧视性过滤，保险公司对人群的歧视性保费

[Zhao 2017, Jia 2020] 词嵌入会**强化偏见**

- 嵌入空间中相似的向量会聚合

历史、文化偏见

古代中国对外国人的观点：

俗无礼义，人性犷暴。形貌鄙陋，衣服毡褐。眼多碧绿，异于诸国。--《大唐西域记》
人皆高鼻深目，如回回状，身穿锁袱披裘，以皮为裤，又以皮囊其阴物，露出于外。头目常看书，取而视之，乃佛经也。--《静虚斋惜阴录》，明朝

历史、文化偏见

古代中国对外国人的观点：

俗无礼义，人性犷暴。形貌鄙陋，衣服毡褐。眼多碧绿，异于诸国。--《大唐西域记》
人皆高鼻深目，如回回状，身穿锁袱披裘，以皮为裤，又以皮囊其阴物，露出于外。头目常看书，取而视之，乃佛经也。--《静虚斋惜阴录》，明朝

近、现中国对外国人的观点：

为了把我自己打扮得像个西洋人……把自己装点成《老爷杂志》上的外国贵族模样……穿着最讲究的英国料子西服……手提“文明棍”，戴着德国蔡司厂出品的眼镜……身边还跟着两条或三条德国猎犬和奇装异服的一妻一妾。--《我的前半生》
“德先生”和“赛先生”，“民主”和“科学”--新文化运动

历史、文化偏见

古代中国对外国人的观点：

俗无礼义，人性犷暴。形貌鄙陋，衣服毡褐。眼多碧绿，异于诸国。--《大唐西域记》
人皆高鼻深目，如回回状，身穿锁袱披裘，以皮为裤，又以皮囊其阴物，露出于外。头目常看书，取而视之，乃佛经也。--《静虚斋惜阴录》，明朝

近、现中国对外国人的观点：

为了把我自己打扮得像个西洋人……把自己装点成《老爷杂志》上的外国贵族模样……穿着最讲究的英国料子西服……手提“文明棍”，戴着德国蔡司厂出品的眼镜……身边还跟着两条或三条德国猎犬和奇装异服的一妻一妾。--《我的前半生》
“德先生”和“赛先生”，“民主”和“科学”--新文化运动

当代审美本质上是政治宣传：从“公知带路”到走出国门看真相

实验：词的相似性、类比任务

Review

本章内容

词嵌入 word2vec。跳元模型 skip-gram。其他静态嵌入。BPE 子词嵌入。词嵌入的语义属性。

重点：word2vec；skip-gram；连续词袋 CBOW；全局向量的词嵌入 GloVe；BPE 子词嵌入；词嵌入的语义属性。

难点：词嵌入表示的特点、优缺点。

学习目标

- 理解词嵌入表示的特点（稠密），及其优缺点（效能高，丢失信息、词义）
- 理解 word2vec 的原理（训练二分类模型当作编码器）、skip-gram 的工作流程
- 理解 CBOW、GloVe 静态嵌入算法
- 理解 BPE 子词嵌入算法，及其优点（有效解决未知词问题）
- 理解使用词嵌入做类比推断的原理（平行四边形法则），并能举例说明几个语义分析应用

问题

简述词嵌入表示的特点，及其优缺点。

简述 word2vec 的原理、skip-gram 的工作流程。

简述 CBOW、GloVe 静态嵌入算法

简述 BPE 子词嵌入算法

简述使用词嵌入做类比推断的原理，并举例说明几个语义分析应用。