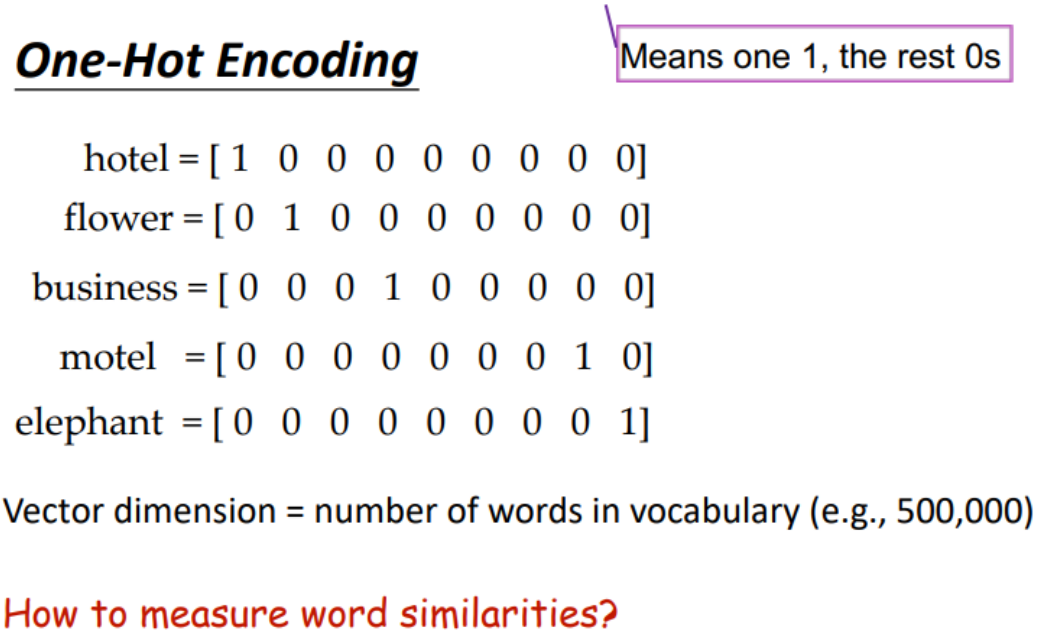
11.word2vec

one-hot encoding

- Words can be represented by **one-hot** vectors:



problem

用户搜索Minhang motel时，想进行匹配，但是

motel = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

hotel = [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

两个向量是正交的，并没有任何关系

word emmbeding

词嵌入最粗浅的理解

o 词映射到低维连续向量(如图)

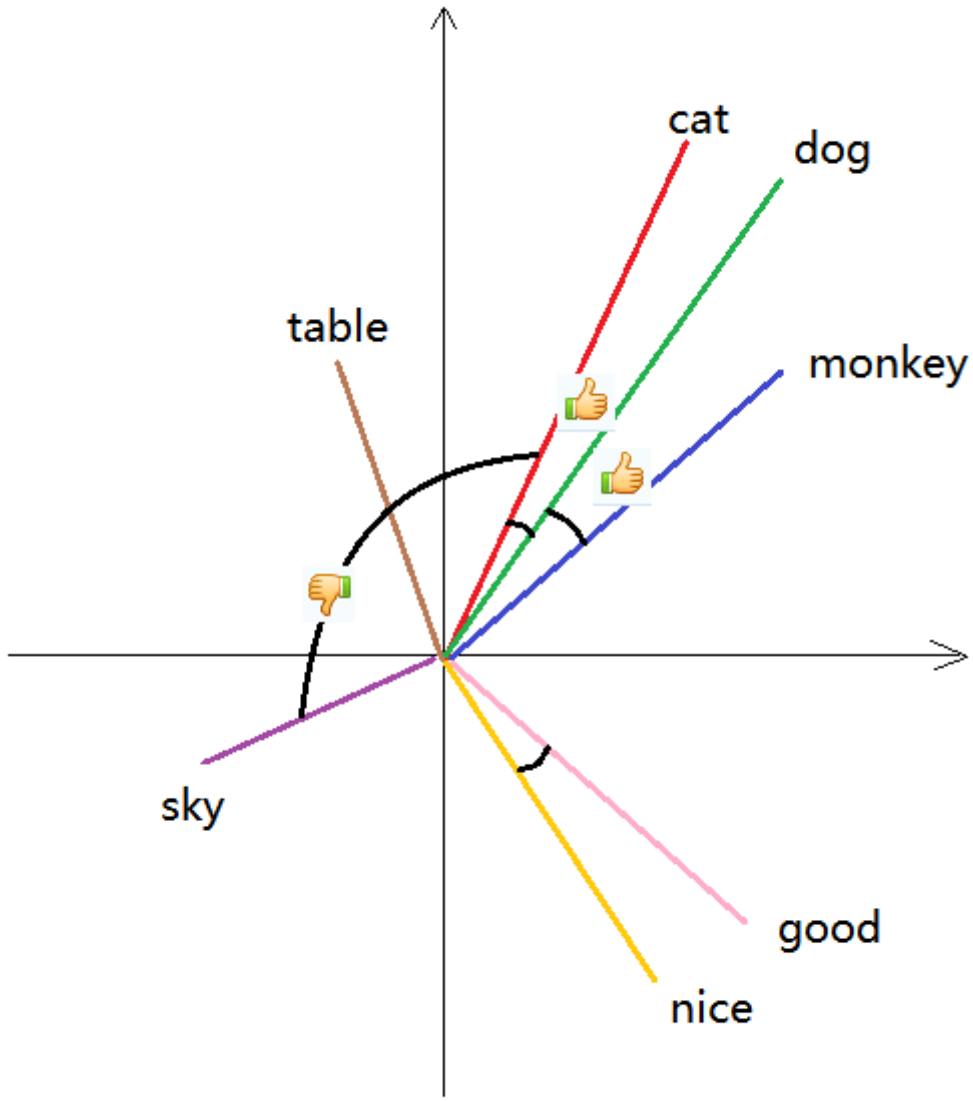
cat: (-0.065, -0.035, 0.019, -0.026, 0.085,...)

dog: (-0.019, -0.076, 0.044, 0.021,0.095,...)

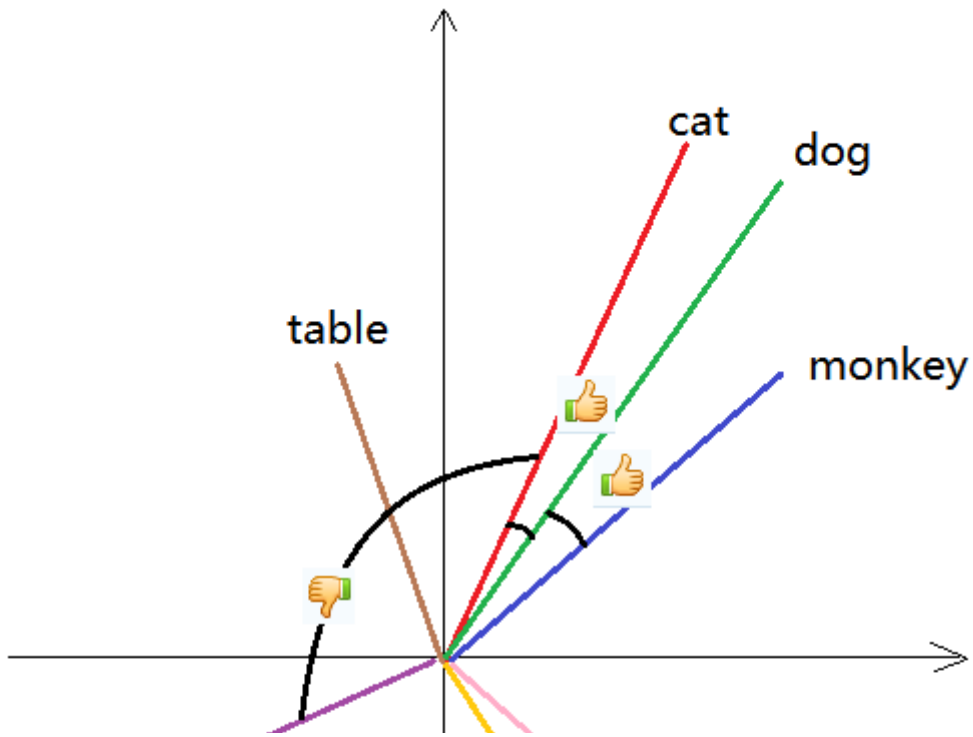
table: (0.027, 0.013, 0.006, -0.023, 0.014, ...)

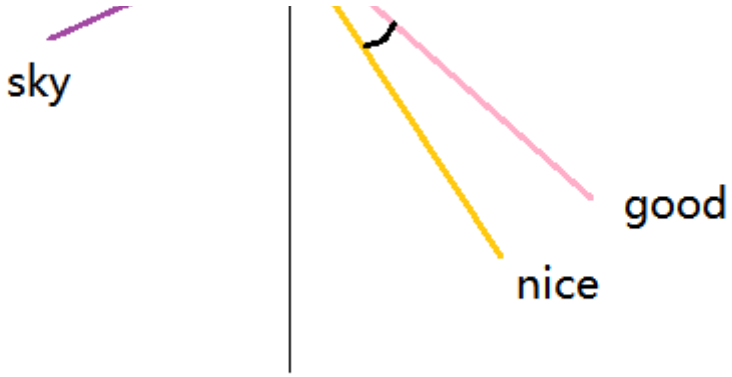
o 相似词映射到相似方向 -- 语义相似性被编码了

o Cosine相似度衡量方向



Projection of the embedding vectors to 2-D





Projection of the embedding vectors to 2-D

词嵌入可以做类比题

- v(“国王”) – v(“王后”) ≈ v(“男”) – v(“女”) o v(“英国”) + v(“首都”) ≈ v(“伦敦”)
- 反映出语义空间中的线性关系 o词嵌入编码了语义空间中的线性关系, 向量的不同部分对应不同的语义 o 质疑：然而并没有什么x用？ o
- 两个句子: A含“英国”,“首都”，不含“伦敦”；B含“伦敦” o 所有词的词向量的和表示句子
- 两个句子仍会比较相似

a:b :: c:?

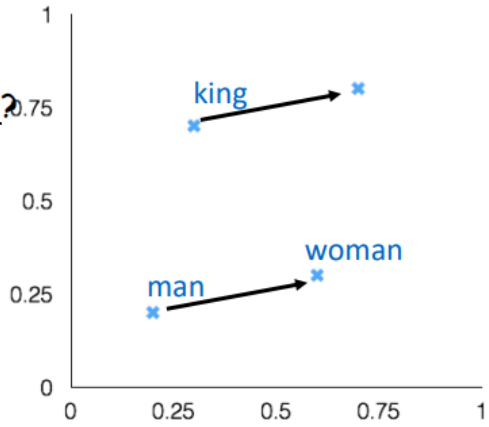
man:woman :: king:?

$$d = \arg\max_i \frac{(x_b - x_a + x_c)^T x_i}{||x_b - x_a + x_c||}$$

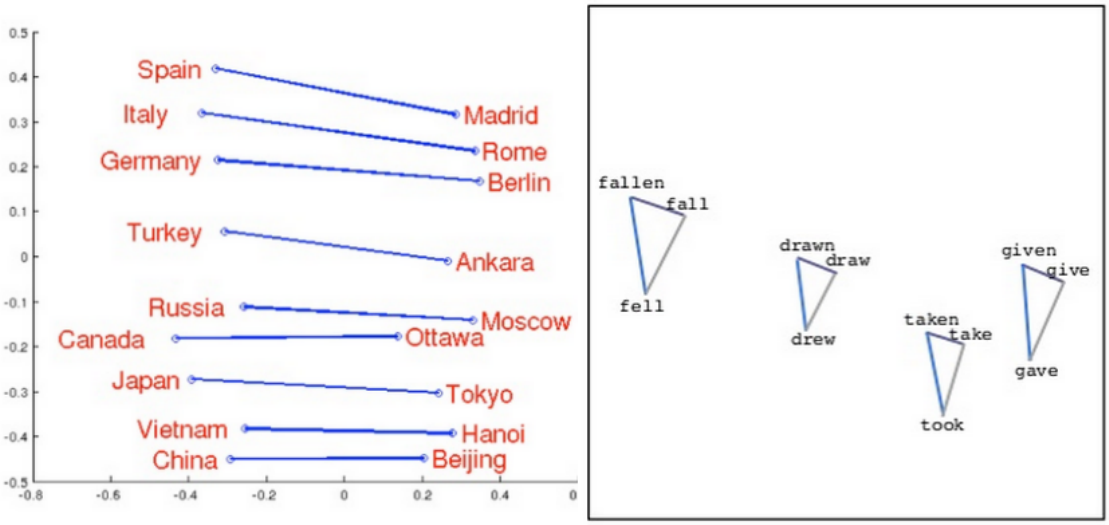
- Man is to Woman as King is to___ ?
- Good is to Best as Smart is to___?
- China is to Beijing as America is to___?

- It turns out that word2vec is good for such analogy task.

$V_{king} - V_{man} + V_{woman} = V_{queen}$



Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. NIPS, 1-9.



相似词映射到相似方向：为什么

- 基本假设：“相似”词的邻居词分布类似 o 倒推：两个词邻居词分布类似 → 两个词语义相近
- **猫** 宠物 主人 喂食 蹭 喵
- **狗** 宠物 主人 喂食 咬 汪
- v(“猫”)≈v(“狗”)
- Apple: tree red growth design music company engineering executive
- **v**(“apple”)≈**v**(“orange”), **v**(“apple”)≈**v**(“microsoft”)

词嵌入的优点 传统one-hot编码： “天气”: (1,0,0…,0)，“气候”: (0,1,0,…0) 权力/的/游戏： (1,0,0,1,1,0,0,…) 冰/与/火/之/歌: (0,1,1,0,0,1,1,…) o 维度高（几千–几万维稀疏向量），

- 数据稀疏
- 没有编码不同词之间的语义相似性
- 难以做模糊匹配

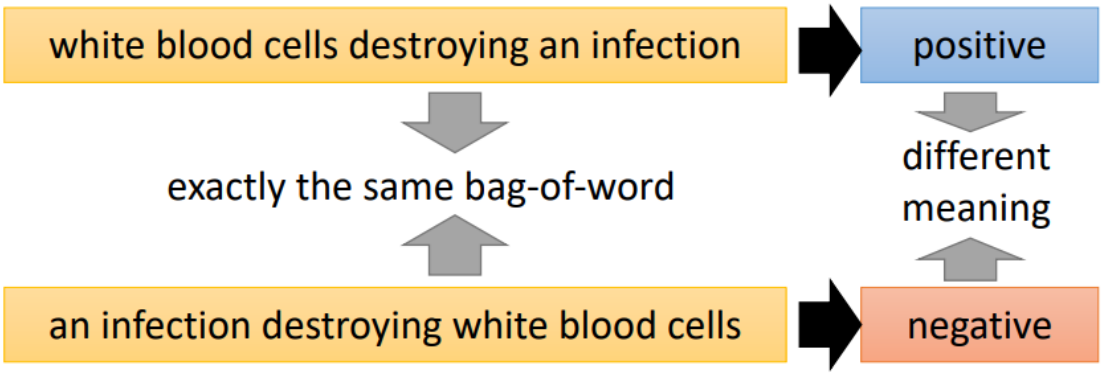
词嵌入：

- 维度低（100 – 500维），连续向量，方便机器学习模型处理
- 无监督学习，容易获得大语料
- 天然有聚类后的效果
- 一个向量可以编码一词多义 (歧义的问题需另外处理)
- 罕见词也可以学到不错的表示：“风姿绰约” ≈ “漂亮”

局限

不能表示语句中词语顺序的关系

- To understand the meaning of a sentence, the order of the words can not be ignored.



the vector space model

- The cornerstone technology in information retrieval

Term-Document Matrix

Each cell is the count of word *t* in document *d*

	D1	D2	D3	D4	D5
ekonomi	0	1	40	38	1
pusing	4	5	1	3	30
keuangan	1	2	30	25	2
sakit	4	6	0	4	25
inflasi	8	1	15	14	1

Two documents are similar if they have similar vector!
D3 = [40, 1, 30, 0, 15]
D4 = [38, 3, 25, 4, 14]

Term-Document Matrix

Each cell is the count of word *t* in document *d*

	D1	D2	D3	D4	D5
ekonomi	0	1	40	38	1
pusing	4	5	1	3	30
keuangan	1	2	30	25	2
sakit	4	6	0	4	25
inflasi	8	1	15	14	1

Vector of word "sakit" = [4, 6, 0, 4, 25]

Term-Document Matrix

Each cell is the count of word *t* in document *d*

	D1	D2	D3	D4	D5
ekonomi	0	1	40	38	1
pusing	4	5	1	3	30
keuangan	1	2	30	25	2
sakit	4	6	0	4	25
inflasi	8	1	15	14	1

Two words are similar if they have similar vector!
pusing = [4, 5, 1, 3, 30]
sakit = [4, 6, 0, 4, 25]

缺点

- long (length |V|= 20,000 to 50,000)
- sparse (most elements are zero)
- 难以用作机器学习中的特征（需要调整的权重更多）
- n存储显式计数可能很难推广