开始：文本表示

Word Representation

One-hot representation

词典：[我们，去，爬山，今天，你们，昨天，跑步]

0 3 6 7

每个单词的表示：

- 我们：(1, 0, 0, 0, 0, 0, 0) → 7维 = |词典|
- 爬山：(0, 0, 1, 0, 0, 0, 0) → 7维 = |词典|
- 跑步：(0, 0, 0, 0, 0, 0, 1) → 7维 = |词典|
- 昨天：(0, 0, 0, 0, 0, 1, 0) → 7维 = ...

Sentence Representation (boolean)

Boolean Representation

词典：[我们，又，去，爬山，今天，你们，昨天，跑步]

8个单词

每个句子的表示

我们 今天 去 爬山：(1, 0, 1, 1, 1, 0, 0, 0) → 8维=1词典

你们 昨天 跑步：(0, 0, 0, 0, 0, 1, 1, 1) → 8维=1词典

你们 又 去 爬山 又 去 跑步：(0, 1, 1, 1, 0, 1, 0, 1) → 8维=1词典

Sentence Representation (count)

Count-based representation

词典：[我们，又，去，爬山，今天，你们，昨天，跑步]

每个句子的表示

我们 今天 去 爬山：(1, 0, 1, 1, 1, 0, 0, 0) ⇒ 8维=1词典

你们 昨天 跑步：(0, 0, 0, 0, 0, 1, 1, 1) ⇒ 8维=1词典

你们 又 去 爬山 又 去 跑步：(0, 2, 2, 1, 0, 1, 0, 1) ⇒ 8维=1词典

Sentence Similarity

Distance

计算距离（欧式距离）：$d = |s1 - s2|$

$s1 = (x_1, x_2, x_3)$

$s2 = (y_1, y_2, y_3)$

$d = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2}$

S1: “ 我们 今天 去 爬山 ”= (1,0,1,1,0,0,0,0)

S2: " 你们 昨天 跑步 " = (0,0,0,0,0,1,1,1)

S3: “ 你们 又 去 爬山 又 去 跑步 " = (0,2,2,1,0,1,0,1)

$d(S_1, S_2) = \sqrt{1^2+1^2+1^2+1^2+1^2+1^2} = \sqrt{6}$

$d(S_1, S_3) = \sqrt{1^2+2^2+1^2+1^2+1^2} = \sqrt{8}$

$d(S_2, S_3) = \sqrt{2^2+2^2+1^2+1^2} = \sqrt{10}$

$sim(S_1, S_2) > sim(S_2, S_3)$; $sim(S_1, S_3) > sim(S_2, S_3)$

Sentence Similarity

$S_1 = (x_1, x_2, x_3)$
$S_2 = (y_1, y_2, y_3)$

计算相似度（余弦相似度）： $d = (s1 \cdot s2)/(|s1| * |s2|)$

$d = \frac{x_1y_1 + x_2y_2 + x_3y_3}{\sqrt{x_1^2+x_2^2+x_3^2}\sqrt{y_1^2+y_2^2+y_3^2}}$

内积　Normalization

S1: “ 我们 今天 去 爬山 ”= (1,0,1,1,0,0,0,0)

S2: " 你们 昨天 跑步 " = (0,0,0,0,0,1,1,1)

S3: “ 你们 又 去 爬山 又 去 跑步 " = (0,2,2,1,0,1,0,1)

$d(S_1, S_2) = 0/A = 0$

$d(S_1, S_3) = (2+1)/(\sqrt{3}\cdot\sqrt{11}) = \frac{3}{\sqrt{33}}$

$d(S_2, S_3) = 2/(\sqrt{3}\cdot\sqrt{11}) = \frac{2}{\sqrt{33}}$

$sim(S_1, S_3) > sim(S_2, S_3) > sim(S_1, S_2)$

Cosine Similarity

Sentence Similarity

句子1：He is going from Beijing to Shanghai

句子2：He denied my request, but he actually lied.

句子3：Mike lost the phone, and phone was in the car

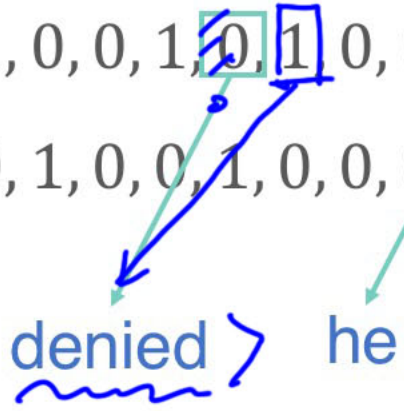
=> 词频

句子1：$(0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0)$

句子2：$(1, 0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0)$

句子3：$(0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 2, 0, 0, 2, 0, 1)$

count-based representation



Sentence Similarity

句子1：He is going from Beijing to Shanghai

句子2：He denied my request, but he actually lied.

句子3：Mike lost the phone, and phone was in the car

句子1：$(0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0)$

句子2：$(1, 0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0)$

句子3：$(0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 2, 0, 0, 2, 0, 1)$

denied

he

① 并不是出现的越多就越重要！

② 并不是出现的越少就越不重要！

Sentence Similarity

句子1：He is going from Beijing to Shanghai

句子2：He denied my request, but he actually lied.

句子3：Mike lost the phone, and phone was in the car

句子1：$(0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0)$

句子2：$(1, 0, 0, 1, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0)$

句子3：$(0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 2, 0, 0, 2, 0, 1)$

denied

he

并不是出现的越多就越重要！

并不是出现的越少就越不重要！

Tf-idf Representation

$$tfidf(w) = tf(d, w) * idf(w)$$

考虑单词的重要性

文档d中w的词频

$$\log \frac{N}{N(w)}$$

i.e. $N = 100$

$N(w) = 5$

$\frac{N}{N(w)} = 20$

N: 语料库中的文档总数

N(w): 词语w出现在多少个文档?

$$tfidf(w) = tf(d,w) * idf(w)$$

① 词典：[今天, 上, NLP, 课程, 的, 有, 意思, 数据, 也]

|词典| = 9

句子1 今天 上 NLP 课程

句子2 今天 的 课程 有 意思

句子3 数据 课程 也 有 意思

⇒ 句子1

$= (1\cdot\log\frac{3}{2}, 1\cdot\log\frac{3}{1}, 1\cdot\log\frac{3}{1}, 1\cdot\log\frac{3}{3}, 0, 0, 0, 0, 0)$

$= (\log\frac{3}{2}, \log 3, \log 3, \log 1, 0, 0, 0, 0, 0)$ ⇒ 9维 = |词典|

tf-idf 向量

⇒ 句子2

$= (1\cdot\log\frac{3}{2}, 0, 0, 1\cdot\log\frac{3}{3}, 1\cdot\log\frac{3}{1}, 1\cdot\log\frac{3}{2}, 1\cdot\log\frac{3}{2}, 0, 0)$

$= (\log\frac{3}{2}, 0, 0, \log 1, \log 3, \log\frac{3}{2}, \log\frac{3}{2}, 0, 0)$ ⇒ 9维 = |词典|

tf-idf 向量

⇒ 句子3

$= (0, 0, 0, 1\cdot\log\frac{3}{3}, 0, 1\cdot\log\frac{3}{2}, 1\cdot\log\frac{3}{2}, 1\cdot\log\frac{3}{1}, 1\cdot\log\frac{3}{1})$

$= (0, 0, 0, \log 1, 0, \log\frac{3}{2}, \log\frac{3}{2}, \log 3, \log 3)$ ⇒ 9维 = |词典|

Measure Similarity Between Words

下面哪些单词之间语义相似度更高?

我们，爬山，运动，昨天

我们=(0,1 0, 0 - - -0)

爬山=(0、0、1,0 - - -0)

Sim(我们, 爬山) < Sim(运动, 爬山)

o One-hot representation

- boolean-based repr
- count-based "
- tf-idf-based repres

Measure Similarity Between Words

下面哪些单词之间语义相似度更高?

我们，爬山，运动，昨天

我们：(0, 1, 0, 0, 0, 0)

爬山：(0, 0, 1, 0, 0, 0)

运动：(1, 0, 0, 0, 0, 0)

昨天：(0, 0, 0, 1, 0, 0)

① Euclidean Distance

$d(\text{我们}, \text{爬山}) = \sqrt{2}$

$d(\text{我们}, \text{运动}) = \sqrt{2}$

$d(\text{运动}, \text{爬山}) = \sqrt{2}$

$d(\text{运动}, \text{昨天}) = \sqrt{2}$

✗

② Cosine Similarity

$\text{sim}(\text{我们}, \text{爬山}) = \frac{0}{1} = 0$

$\text{sim}(\text{我们}, \text{运动}) = 0$

$\text{sim}(\text{运动}, \text{爬山}) = 0$

$\text{sim}(\text{运动}, \text{昨天}) = 0$

✗

- One-hot representation

✗ 有没有可能用单词表示语义相似度?

Measure Similarity Between Words

利用 One-hot 表示法表达单词之间相似度?

每个单词的表示:

我们: [1, 0, 0, 0, 0, 0, 0]

爬山: [0, 0, 1, 0, 0, 0, 0]

运动: [0, 0, 0, 0, 0, 0, 1]

昨天: [0, 0, 0, 0, 0, 1, 0]

NO!

Another Issue: Sparsity

1 我们 今天 打算 去 爬山 = ( ) ⇒ 向量的大小 = 词典

2 你们 昨天 做 什么了 ⇒

3 明天 打算 去 上课 ⇒ (0.0..1..0..0..1..1)

i.e, 中文词典 $10^5 \sim 10^6$

$10^5 \sim 10^6$

$10^5 - 4$ 位置 全是 0

One-hot representation

① 不能表示词汇的相似性

② Sparsity

From One-hot Representation to Distributed Representation

分布式表示法？

One-Hot Representation

我们： [1, 0, 0, 0, 0, 0, 0]
爬山： [0, 0, 1, 0, 0, 0, 0]
运动： [0, 0, 0, 0, 0, 0, 1]
昨天： [0, 0, 0, 0, 0, 1, 0]

长度=词典
$10^5 \sim 10^6$

≫

Distributed Representation

我们： [0.1, 0.2, 0.4, 0.2]
爬山： [0.2, 0.3, 0.7, 0.1]
运动： [0.2, 0.3, 0.6, 0.2]
昨天： [0.5, 0.9, 0.1, 0.3]

长度 = 100, 200, 300
(Sparsity)

Measure Similarity Between Words

Distributed Representation

我们: [0.1, 0.2, 0.4, 0.2]
爬山: [0.2, 0.3, 0.7, 0.1]
运动: [0.2, 0.3, 0.6, 0.2]
昨天: [0.5, 0.9, 0.1, 0.3]

四维向量

① Euclidean Distance

$d(\text{我们}, \text{爬山}) = \sqrt{0.1^2 + 0.1^2 + 0.3^2 + 0.1^2}$

$= \sqrt{0.01 + 0.01 + 0.09 + 0.01} = \sqrt{0.12}$

$d(\text{运动}, \text{爬山}) = \sqrt{0.1^2 + 0.1^2} = \sqrt{0.02}$

$d(\text{运动}, \text{爬山}) < d(\text{我们}, \text{爬山})$

$\Leftrightarrow sim(\text{运动}, \text{爬山}) > sim(\text{我们}, \text{爬山})$

② 类似 $d(\text{爬山}, \text{运动}) < d(\text{昨天}, \text{运动})$

$\Leftrightarrow sim(\text{爬山}, \text{运动}) > sim(\text{昨天}, \text{运动})$

分布式表示方法(单词) ⇓ 词向量(Word Vectors)

Comparing the Capacities

Q: 100 维的 One-Hot 表示法最多可以表达多少个不同的单词?

Q: 100 维的 分布式 表示法最多可以表达多少个不同的单词?

Comparing the Capacities

容量空间

Q: 100 维的 One-Hot 表示法最多可以表达多少个不同的单词?

向量大小 = 100

我们 = (1, 0, 0 ··· 0)
99个

运动 = (0, 0, -- 1, 0 -- 0)

100个单词!

Q: 100 维的 分布式 表示法最多可以表达多少个不同的单词?

我们: (0.1, 0.2, 0.5 ··· 0.1)
100维

+∞

Binary

我们: (1, 0, 1, 0 ··· 0 1 ···)
100维
0/1 0/1

2^{100} 不同的单词!

Questions

Q: 怎么学习每一个单词的分布式表示（词向量）？

我们：(0.2, 0.3, 0.1, 0.2) ？

Learn Word Embeddings

输入(Input): String

我们今天去爬山

你么昨天运动

你们去爬山

1B: string contains 10^9 tokens

10B

100B

Skip-Gram

Glove

CBOW

RNN/LSTM

MF

Gaussian Embedding

Distributed Representation

我们: [0.1, 0.2, 0.4, 0.2]

爬山: [0.2, 0.3, 0.7, 0.1]

运动: [0.2, 0.3, 0.6, 0.2]

昨天: [0.5, 0.9, 0.1, 0.3]

dim/D: 100/200/300, 50 *

Essence of Word Embedding

Meaning

词向量代表单词的意思
(Meaning)

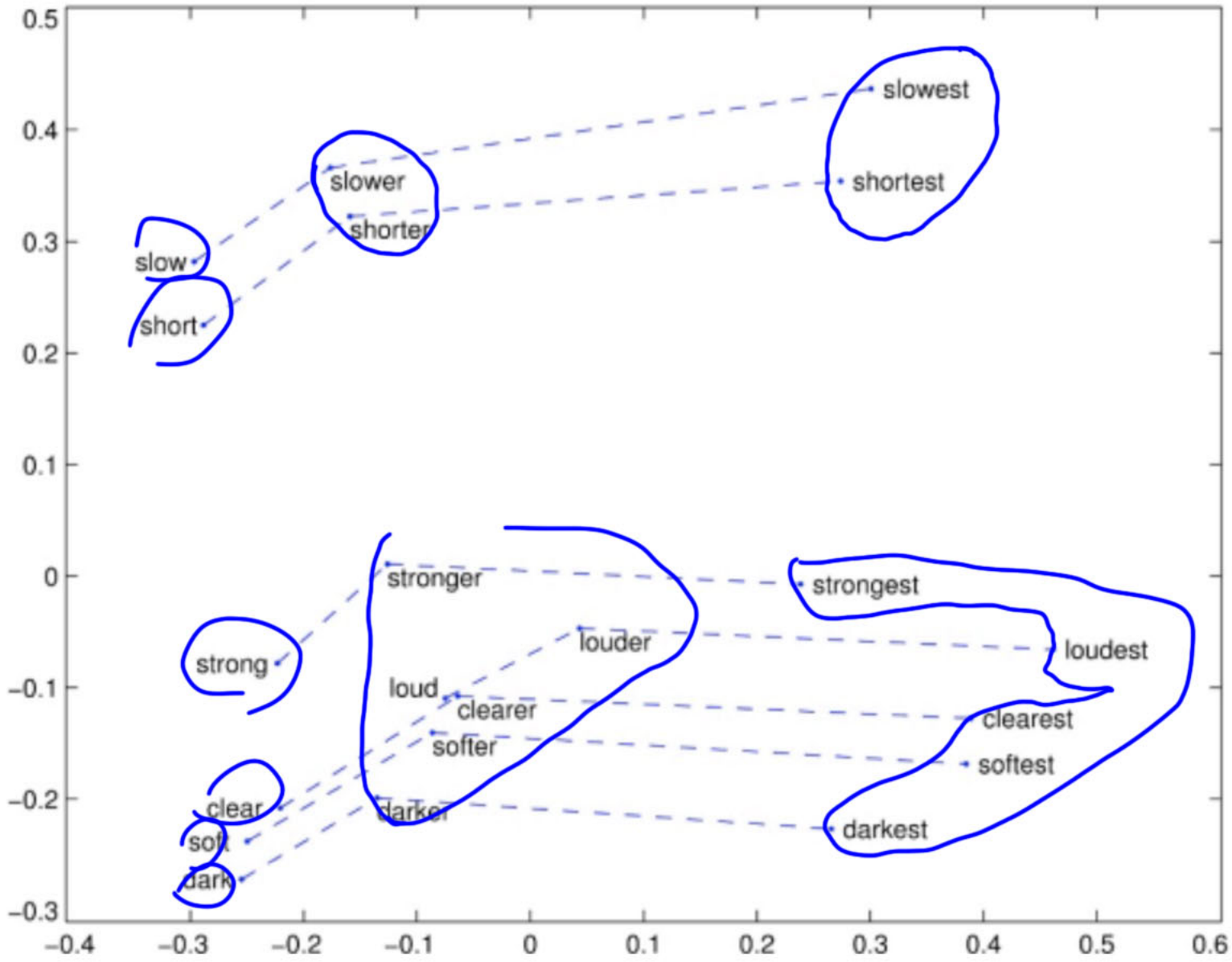
Distributed Representation

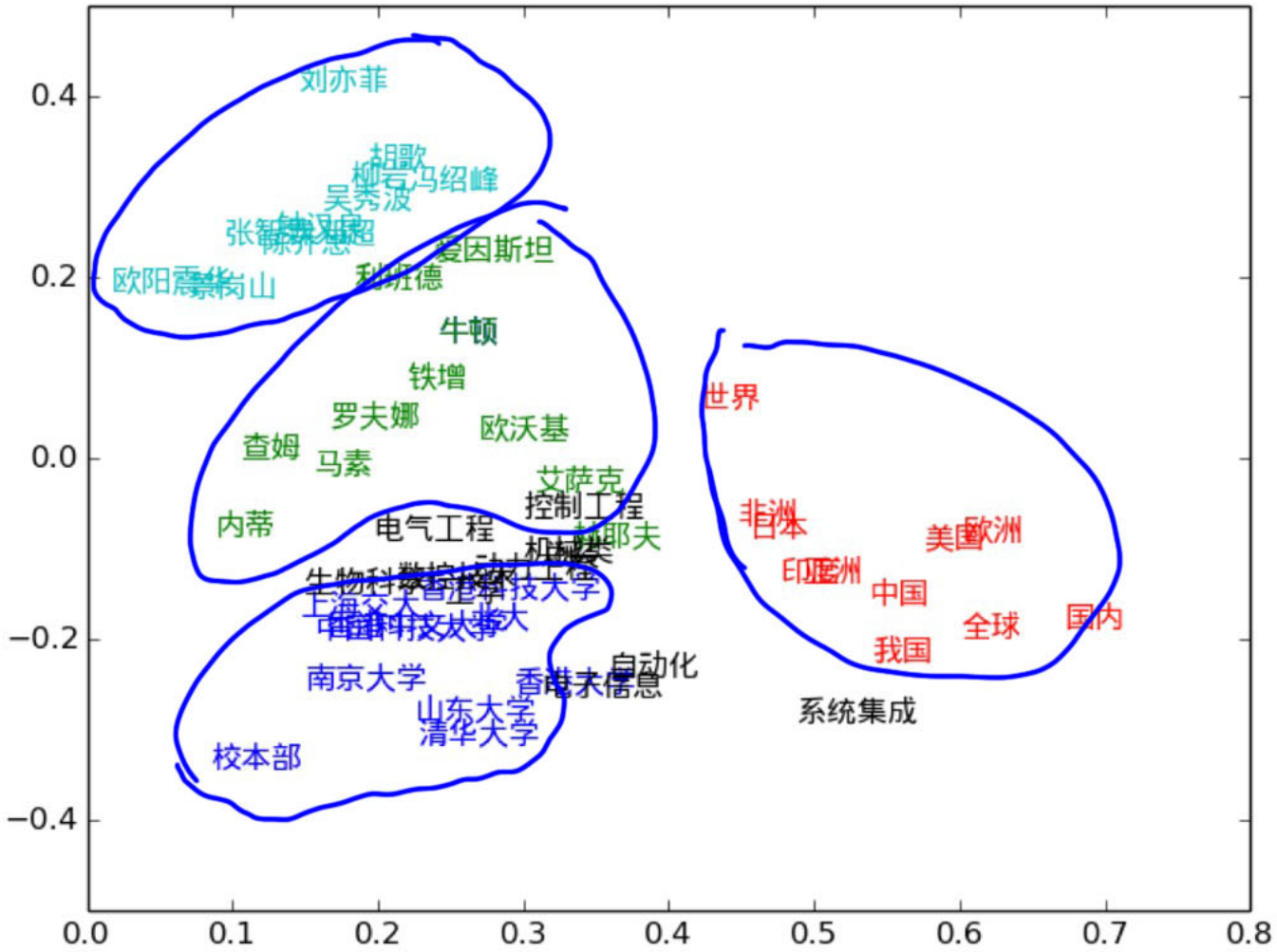
我们: [0.1, 0.2, 0.4, 0.2]
爬山: [0.2, 0.3, 0.7, 0.1]
运动: [0.2, 0.3, 0.6, 0.2]
昨天: [0.5, 0.9, 0.1, 0.3]

(Meaning)

Word2Vec 基于分布式词向量
(Meaning)

woman - man
≈ girl - boy


0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
slow
short
slower
shorter
slowest
shortest
strong
stronger
strongest
loud
louder
loudest
clear
clearer
clearest
soft
softer
softest
dark
darker
darkest


刘亦菲
胡歌
柳岩
冯绍峰
吴秀波
张智霖
陈乔恩
爱因斯坦
欧阳震华
利维德
牛顿
铁增
罗夫娜
欧沃基
查姆
马素
艾萨克
控制工程
内蒂
电气工程
机械类
生物科学
上海交大
北大
南京大学
香港大学
自动化
信息
山东大学
清华大学
校本部
系统集成
世界
非洲
日本
欧洲
美国
印度
中国
全球
国内
我国
0.4
0.2
0.0
−0.2
−0.4
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

From Word Embedding to Sentence Embedding

我们：(0.1, 0.2, 0.1, 0.3)

去：(0.3, 0.2, 0.15, 0.2)

+ 运动：(0.2, 0.15, 0.4, 0.7)

0.6 0.55 0.65 1.2

(0.2 0.18 0.22 0.4)

A: "我们去运动"

① Averaging 法则

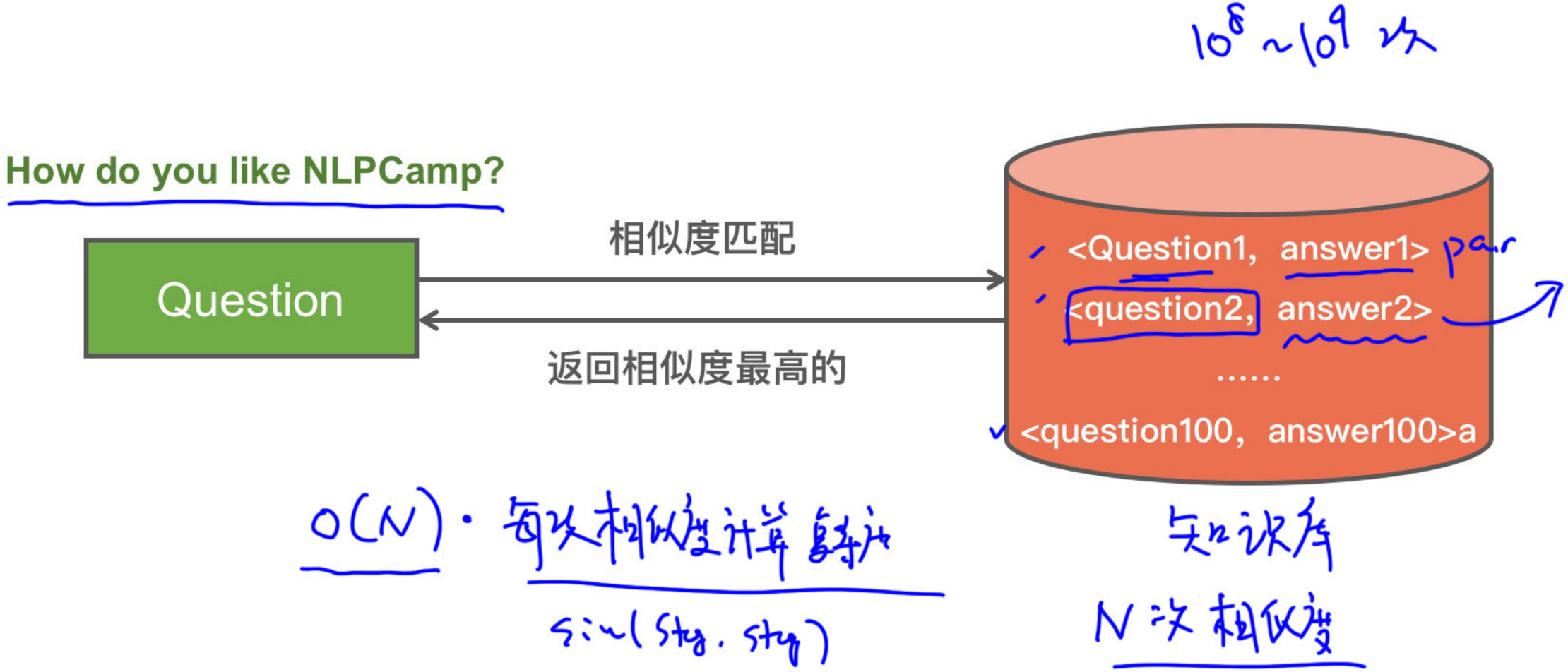
句子向量

=(0.2, 0.18, 0.22, 0.4) ✓

B: "我们是朋友" =(0.2, 0.2, 0.25, 0.4) ✓

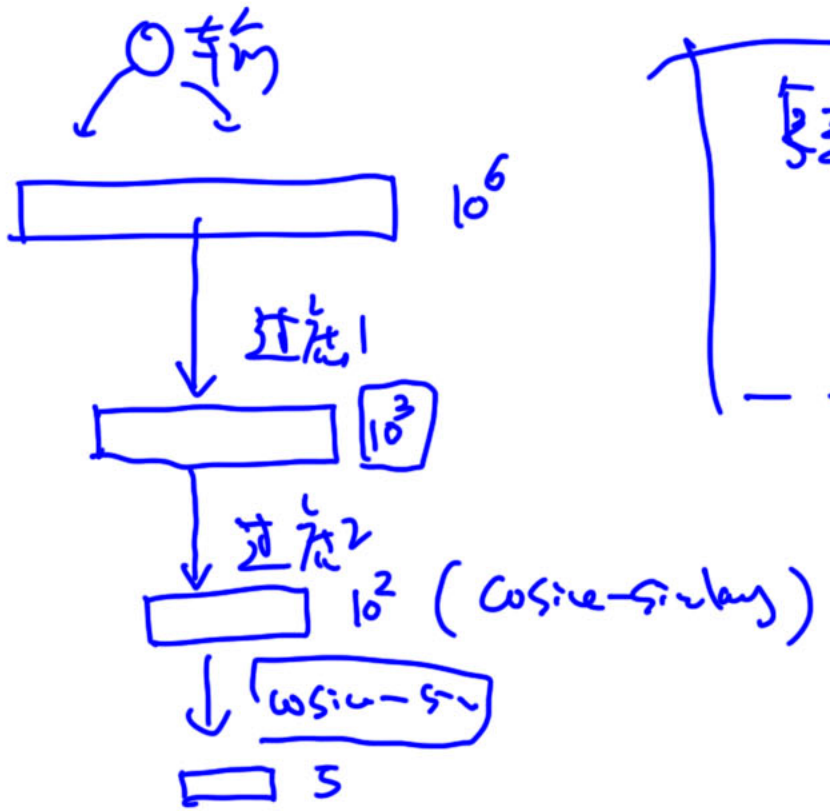
② LSTM/RNN

Recap: Retrieval-based QA System



How to Reduce Time Complexity?

核心思路：“层次过滤思想”



O(N)

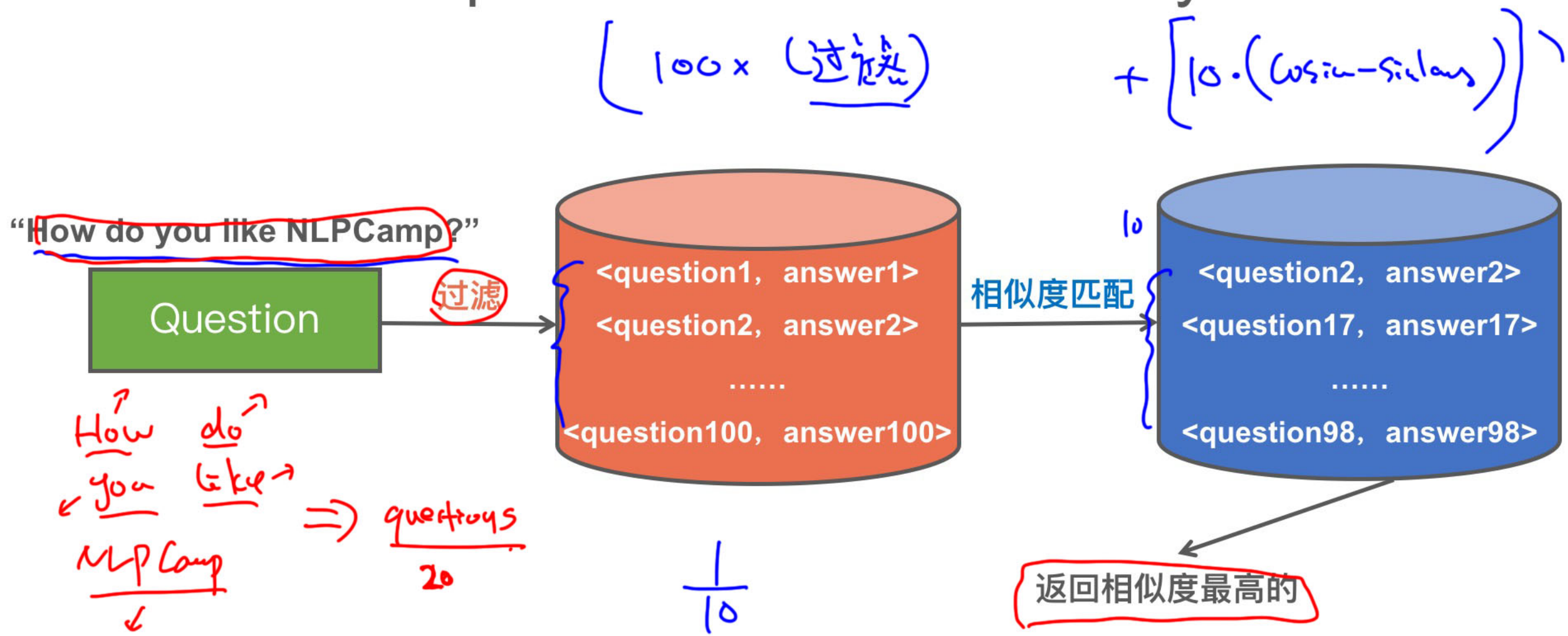
N非常大 ⇒ O(N)

实时性

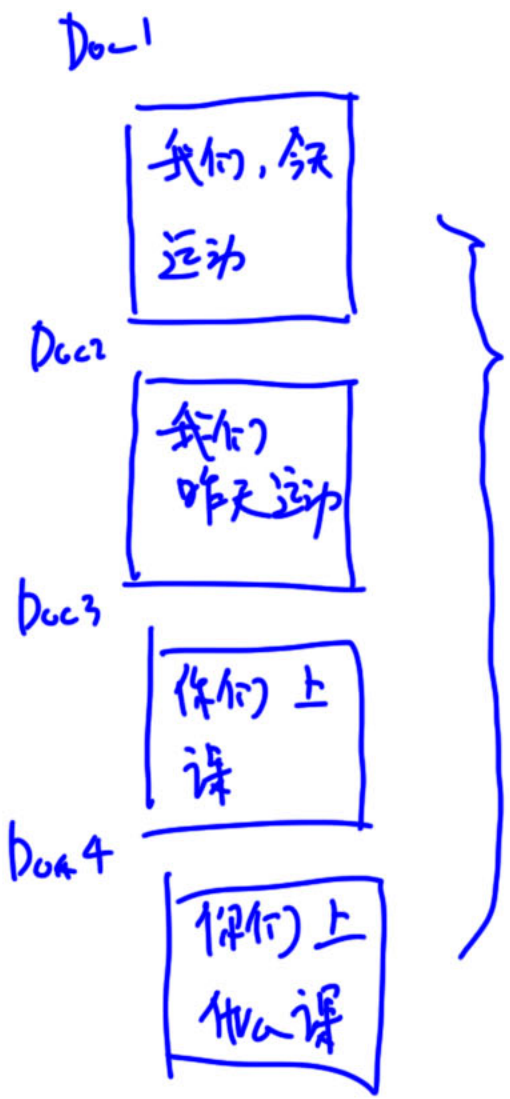
复杂度(过滤1) < 复杂度(过滤2)

< 复杂度(cosine-similarity)

Recap: Retrieval-based QA System



Introducing Inverted Index



词典：[我们，今天，运动，昨天，上，课，什么]

我们：[Doc1, Doc2]

今天：[Doc1]

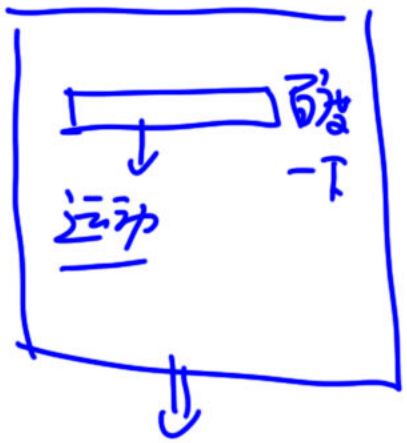
运动：[Doc1, Doc2]

昨天：[Doc2]

上：[Doc3, Doc4]

课：[Doc3. Doc4]

什么：[Doc4]



return Doc1 ∩ Doc2

输入：我们 上课

我们：[Doc1, Doc2]

上课：[Doc3. Doc4]

我们 ∩ 上课 = ∅

[Doc1, Doc2. Doc3. Doc4]

⇓

Ranking