

# 开始：文本表示

# Word Representation

词典：[我们，去，爬山，今天，你们，昨天，跑步]

每个单词的表示：

我们：

爬山：

跑步：

昨天：

# Sentence Representation (boolean)

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：

你们 昨天 跑步：

你们 又 去 爬山 又 去 跑步：

# Sentence Representation (count)

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：

你们 昨天 跑步：

你们 又 去 爬山 又 去 跑步：

结束：文本表示

# 开始：文本相似度

# Sentence Similarity

计算距离（欧式距离）： $d = |s1 - s2|$

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)

# Sentence Similarity

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

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)



# 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, 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)

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

# 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!

结束：文本相似度

# 开始：tf-idf 文本表示

# 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, 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)

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

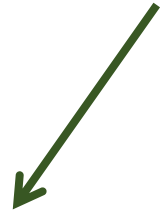
denied

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

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



文档 $d$ 中 $w$ 的词频


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

$N$ : 语料库中的文档总数

$N(w)$ : 词语 $w$ 出现在多少个文档?

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

今天 上 NLP 课程

今天 的 课程 有 意思

数据 课程 也 有 意思

# 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



结束： tf-idf 文本表示

# 开始：词向量介绍

# 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

# 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]

# Another Issue: Sparsity

我们 今天 打算 去 爬山

你们 昨天 做 什么了

明天 打算 去 上课

# From One-hot Representation to Distributed Representation

## 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]



## 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]

# 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]

# 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

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?

# Questions

Q: 怎么学习每一个单词的分布式表示（词向量）？

# 结束：词向量介绍

# 开始：学习词向量

# Learn Word Embeddings

我们今天去爬山

你么昨天运动

你们去爬山

## 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]

# Essence of Word Embedding

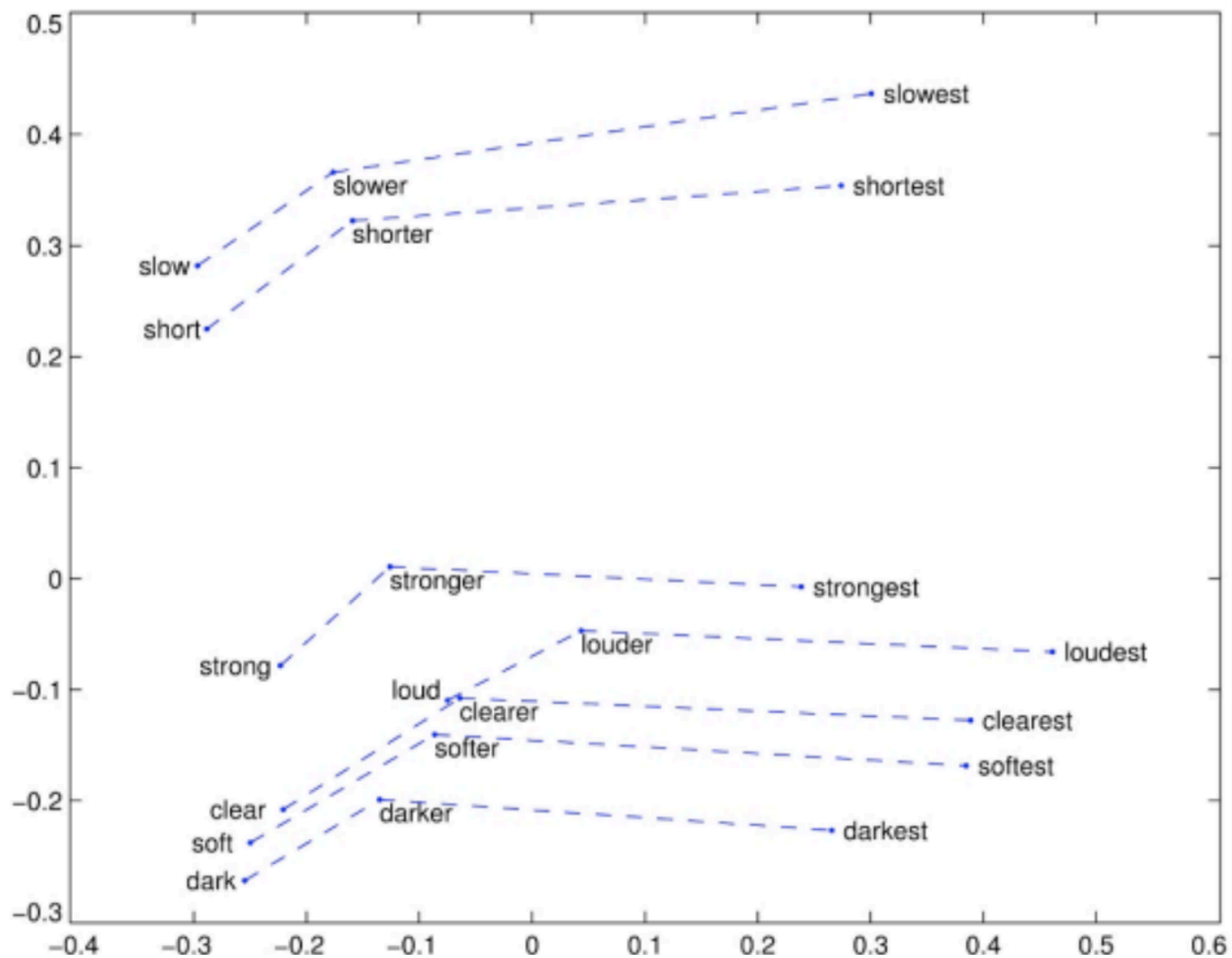
## Distributed Representation

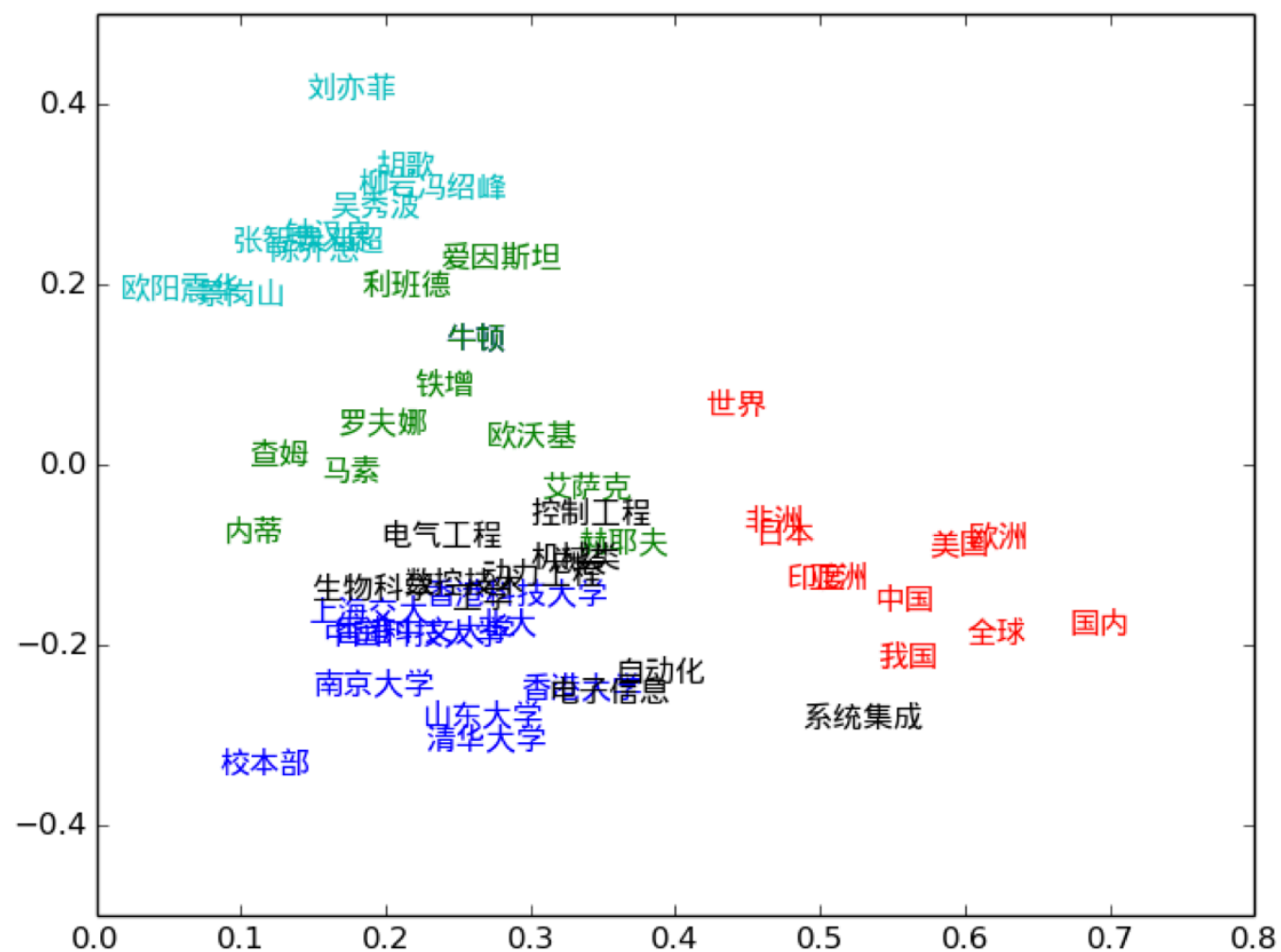
我们: [0.1, 0.2, 0.4, 0.2]

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]







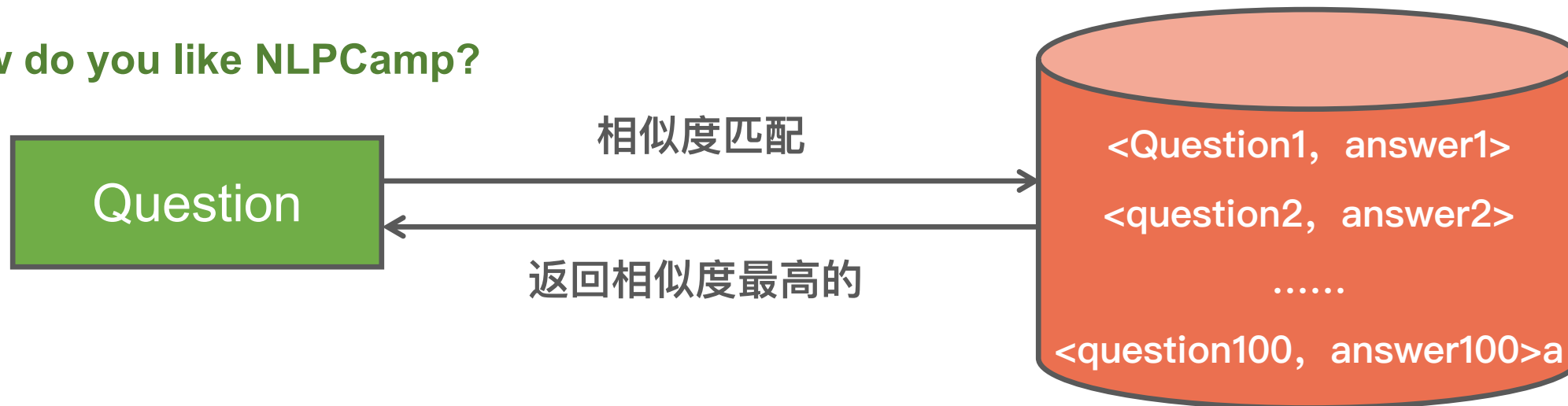
# From Word Embedding to Sentence Embedding

结束：学习词向量

# 开始：基于检索的问答系统缺点

# Recap: Retrieval-based QA System

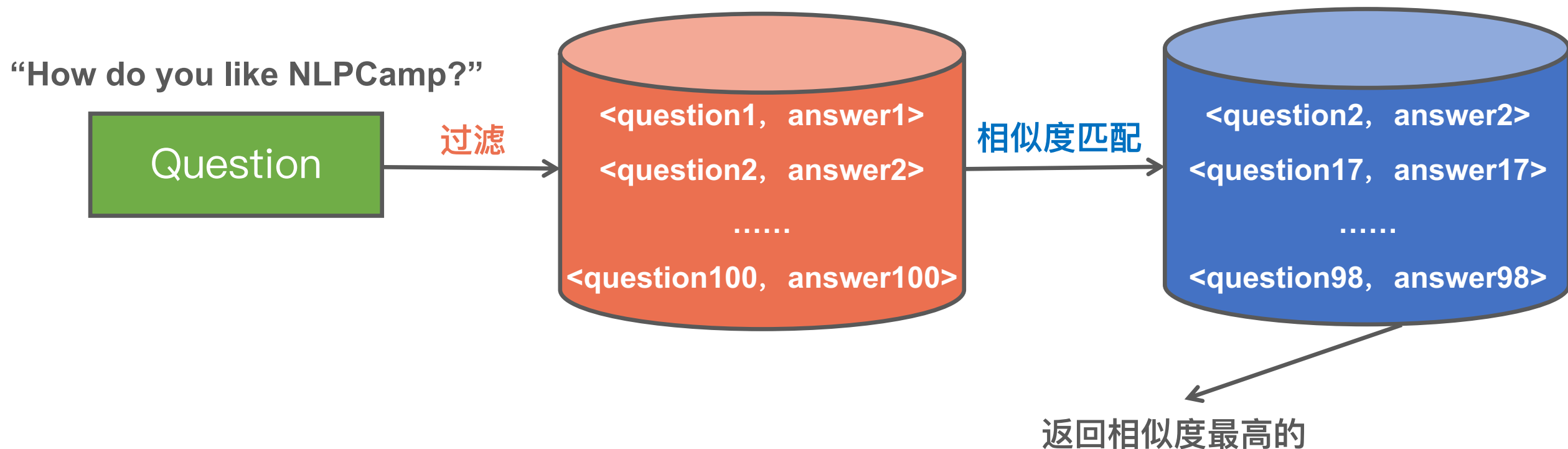
How do you like NLPCamp?



# How to Reduce Time Complexity?

核心思路：“层次过滤思想”

# Recap: Retrieval-based QA System



## 结束：基于检索的问答系统缺点

开始：倒排表



# Introducing Inverted Index

# Constructing Inverted Index

今天 上 NLP 课程

今天 的 课程 有 意思

数据 分析 也 有 意思

昨天 天气 很好

# Using Inverted Index

今天 上 NLP 课程

今天 的 课程 有 意思

数据 分析 也 有 意思

昨天 天气 很好

Is there any issue?

结束：倒排表