智能问答系统实践

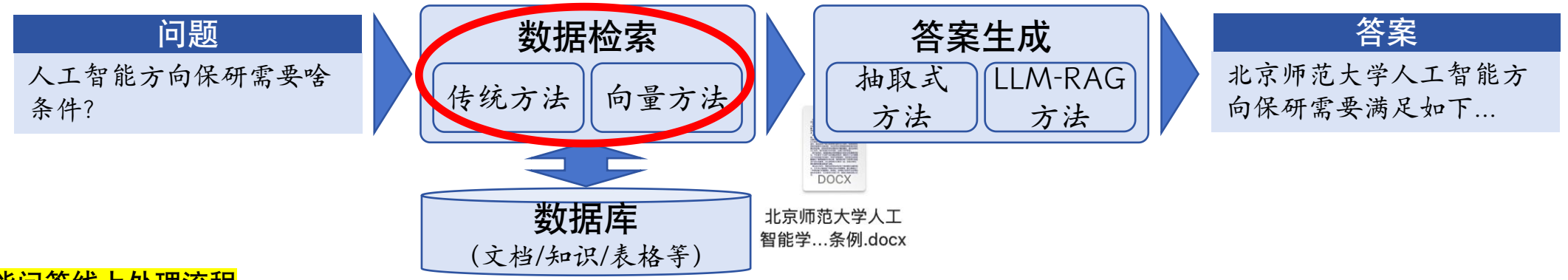
第四课：数据检索

姜文斌
北京师范大学人工智能学院
2025.03.20

我的位置

智能问答系统

针对用户提出的自然语言问题，从数据库中检索相关信息，并依据相关信息作出回答



智能问答线上处理流程

智能问答线下处理模块

问答数据库构建	数据检索模块构建	答案生成模块构建	效果评估模块构建
基于传统方法的数据建库	传统语义匹配模型构建	抽取式答案生成模型构建	文档检索效果评估
基于向量方法的数据建库	向量语义匹配模型构建	RAG式答案生成模型构建	问答整体效果评估

实验设置

基于传统方法的数据检索

- 问题和文档的TF-IDF向量表示
  - 将问题和文档表示为向量
- 基于TF-IDF向量的相似度计算
  - 基于余弦距离的相似度

基于向量方法的数据检索

- 基于BERT的问题向量表示
  - 将问题表示为深度语义向量
- 基于问题向量的文档检索
  - 在HNSW结构中递进式检索

目录

TF-IDF值计算

- TF（词语频率）
  - 公式：TF(t) = (词t在文档中出现的次数) / (文档中总词数)
  - 示例：文档“我/爱/自然/语言/处理”，词“自然”的TF值为1/5=0.2
- IDF（逆文档频率）
  - 公式：IDF(t) = log(总文档数 / 包含词t的文档数)
  - 示例：设总文档数为100，包含“自然”的文档数为10，则IDF值为log(100/10)=1
- TF-IDF计算
  - 公式：TF-IDF(t) = TF(t) × IDF(t)
  - 示例：词“自然”的TF-IDF值为0.2 × 1 = 0.2

余弦相似度计算

- 两个向量的余弦相似度是通过测量两个向量在方向上的相似性来计算的，它是向量空间中两个向量夹角的余弦值
  - 如果两个向量的方向相同，它们的余弦相似度接近1
  - 如果两个向量的方向完全相反，它们的余弦相似度接近-1
  - 如果两个向量正交，则它们的余弦相似度为0

$$\text{余弦相似度}(A,B) = \frac{A \cdot B}{\|A\|\|B\|}$$

$$= \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

A.B：　表示向量 A 和向量 B 的点积（内积）

‖A‖：　表示向量 A 的欧几里得范数（即长度）

‖B‖：　表示向量 B 的欧几里得范数

文本预处理

常见预处理操作

- **分词**：将文档分割成单词或词汇单元
- **去除停用词**：移除常见的无意义词汇

文档集合：

文档1：自然语言处理是人工智能的重要领域。

文档2：自然语言处理技术在搜索引擎中应用广泛。

文档3：人工智能改变了我们的生活。

分词并去除停用词：

文档1：自然、语言、处理、人工智能、重要、领域

文档2：自然、语言、处理、技术、搜索引擎、应用、广泛

文档3：人工智能、改变、生活

向量词汇表构建

- 词汇表是文档集合中所有唯一词汇的集合
- 步骤1：文本预处理
- 步骤2：统计词汇
  - 遍历所有文档，提取每个文档中的词汇
  - 将所有文档中的词汇合并，去重后形成词汇表

分词并去除停用词：

文档1：自然、语言、处理、人工智能、重要、领域

文档2：自然、语言、处理、技术、搜索引擎、应用、广泛

文档3：人工智能、改变、生活

构建词汇表：

{自然、语言、处理、人工智能、重要、领域、技术、搜索引擎、应用、广泛、改变、生活}

问题/文档向量化

- 针对所有文档执行文本预处理
- 构建词汇表
- 计算文档中词语的TF-IDF值
  - 计算TF
  - 计算IDF
- 生成文档对应的TF-IDF向量

文档集合：

文档1：自然语言处理是人工智能的重要领域。

文档2：自然语言处理技术在搜索引擎中应用广泛。

文档3：人工智能改变了我们的生活。

文档4：自然语言处理与机器学习紧密结合。

文档5：机器学习是人工智能的核心技术。

词汇表：

{自然、语言、处理、人工智能、重要、领域...}

文档1的TF-IDF向量：

[0.085,0.085,0.085,0.037,0.269,0.269...]

文档检索过程

- 在基于TF-IDF的信息检索中，检索过程通常分为粗筛和精排两个阶段

- 粗筛
  - **关键词提取**：从查询语句中提取关键词
  - **匹配文档**：根据查询中的关键词，通过倒排索引找到包含这些关键词的文档集合
- 精排
  - **查询向量化**：依据文档集合的词汇表，将查询转换为TF-IDF向量
  - **相似度计算**：使用余弦相似度计算查询向量与候选文档向量之间的相似度
  - **候选文档排序**：根据相似度对候选文档进行排序，返回相似度最高的文档

目录

向量检索 vs 传统检索

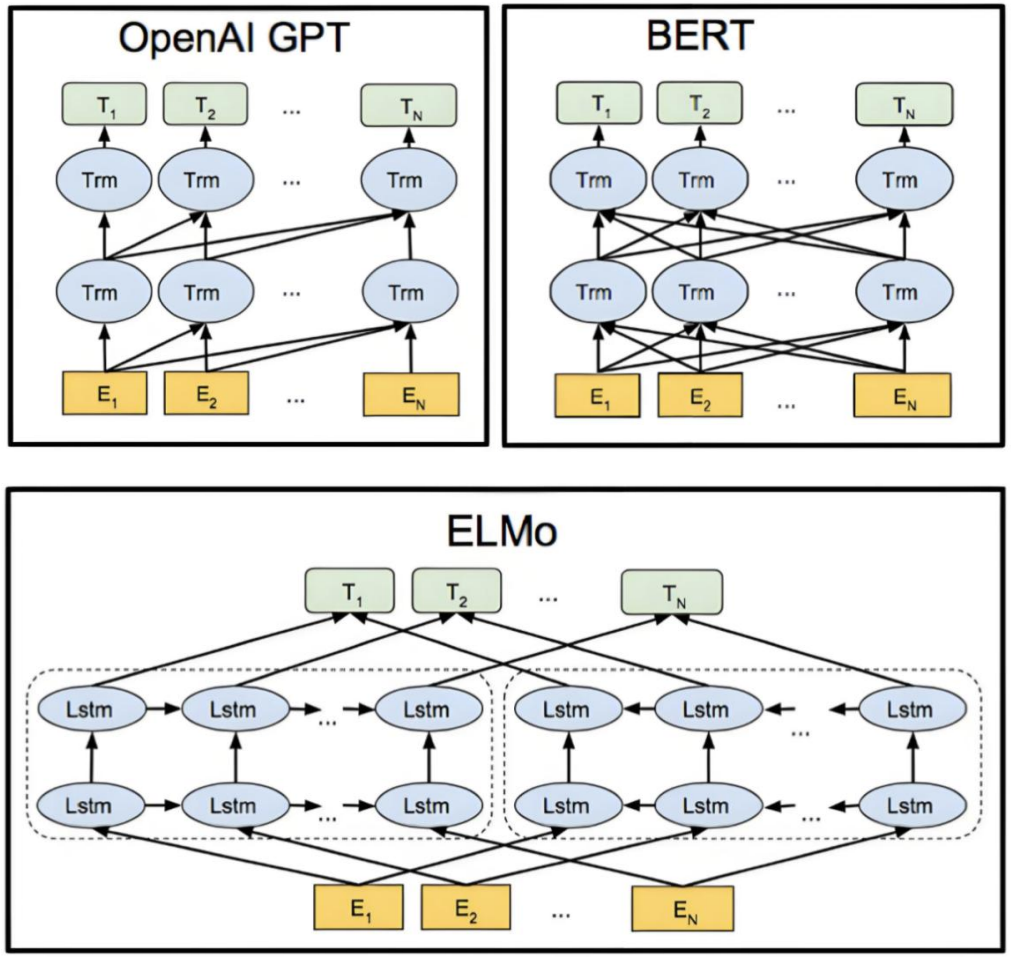
- 传统检索：提取查询关键词，基于直接匹配检索
  - 优点：高效，可解释
  - 缺点：无法处理语义相似性（如“苹果”和“苹果公司”）
- 向量检索：将文本映射为向量，计算语义相似度
  - 优点：刻画深层语义，支持模糊匹配
  - 缺点：计算复杂度高，需要大规模算力
- 思考：两种方法的优缺点？结合的可能性？

一般流程

- **查询向量化**
  - 通过预训练模型（语言模型或词向量模型）生成查询向量
- **查询-文档匹配**
  - 计算查询向量与文档向量的相似度，如余弦相似度、欧氏距离
- **与传统方法结合**
  - 通过传统方法（如TF-IDF）检索出一个候选文档列表
  - 通过向量匹配方法，从新评估每个候选文档的相似度

查询向量化

- **方法1：基于预训练语言模型**
  - 基于BERT预训练模型进行查询语句编码
  - 取[CLS]的向量或所有Token向量平均
- **方法2：基于词向量加权聚合**
  - 基于Word2Vec或GloVe获取各个词向量
  - 基于TF-IDF评估重要度，据此加权求和
- **思考：这两种方法有何优缺点？**



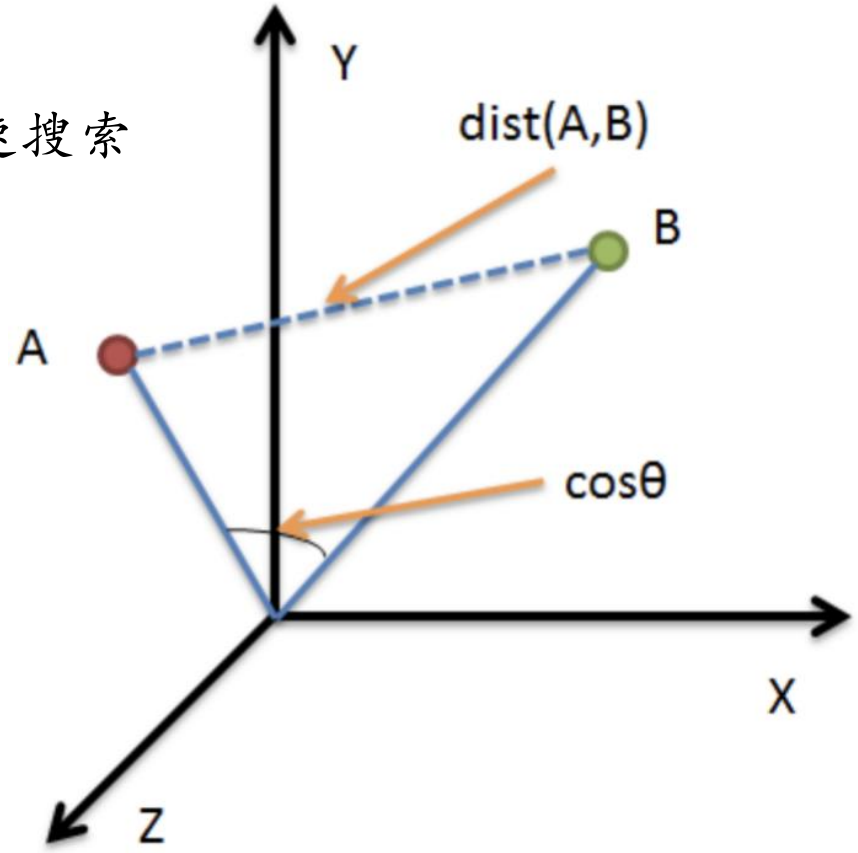
查询-文档匹配

检索模式

- 暴力检索：计算所有向量相似度
- 近似最近邻（ANN）：如HNSW，构建图结构加速搜索

匹配方法

- 余弦相似度：A · B / (|A||B|)
  - 优点：方向敏感，适合文本
- 欧氏距离：$\sqrt{(\Sigma(a\_i - b\_i)^2)}$
  - 优点：绝对距离，适合数值型数据



与传统方法结合

- 向量方法可以与传统方法配合，以粗筛和精排两个阶段实现高效检索

- 粗筛
  - 基于传统方法（TF-IDF）初步检索得到候选的文档列表
  - 这里可以进一步采用“粗筛-精排”策略：关键词匹配粗筛+TF-IDF精排

- 精排
  - 基于向量方法对每个候选文档，重新评估其与查询的相似度

由粗到精，是一项很经典的工程优化策略，用以实现效果和效率的平衡

目录

向量匹配概览

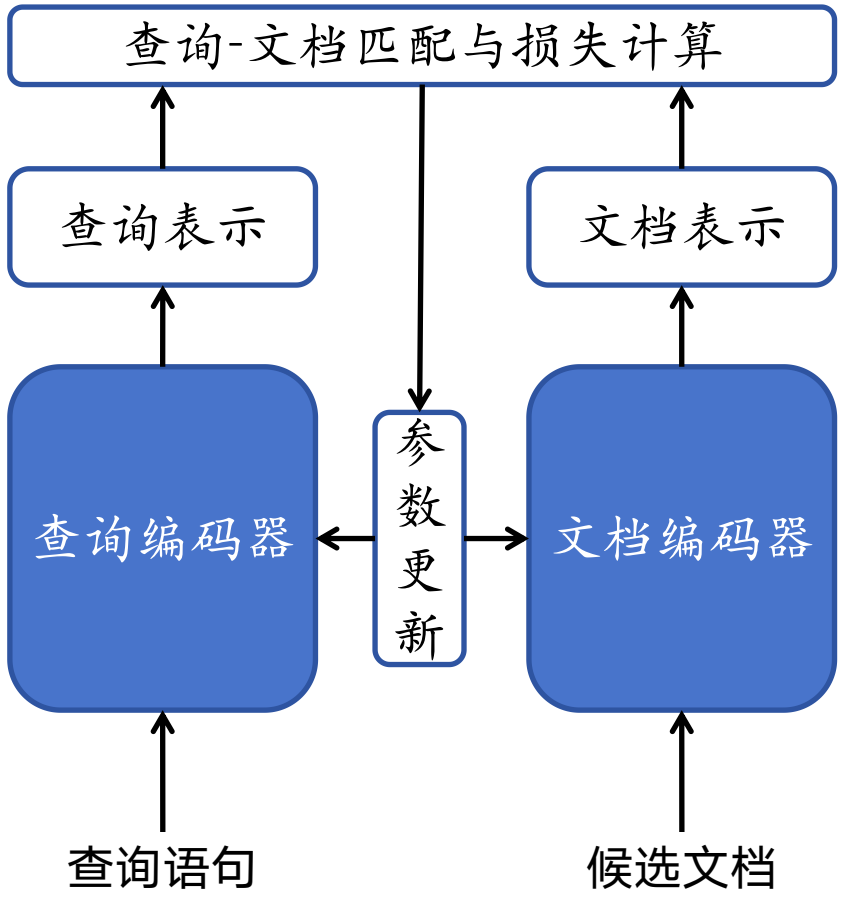
- 基于TF-IDF向量匹配
  - 看似是向量，但本质是衡量词语集合的重合度
- 基于预训练词向量匹配
  - 预训练词向量具备一定语义，但缺乏上下文关联
- 基于预训练语言模型匹配
  - 上下文关联的语义，能更好的衡量语义相似度
- 思考：这些方法共性是什么？有什么缺点？

相似度 vs 支持度

- 基于简单匹配的信息检索，基于这样一个简单假设
  - 只要文档和查询相似，该文档就有望支持查询问题求解
- 相似度可以通过匹配来度量，支持度通过什么来度量？
  - 依然可以通过匹配来度量，根据支持程度来定义匹配程度
- 思考：需要怎么办，才能让匹配考察支持程度？

面向最终效果的表示优化

- **基于查询-文档标注数据的直接优化**
  - 构建<查询，文档>样例集合，文档用于回答查询
  - 构造双塔模型，将查询与文档表示进行对比学习
  - <查询，文档>作正例，自动构造负例，训练模型
  - 用训练好的双塔模型，作为查询和文档的编码器
- **基于最终问答效果驱动的间接优化**
  - 根据智能问答全流程，优化查询匹配模型
  - 根据最终任务效果，可优化全流程的可学习模块



谢谢大家！