客服工作台系统开发作业报告

本报告针对智能客服系统的“FAQ类目管理”与“FAQ录入管理”功能，从技术实现方案到算法路径进行了详细规划。

作业1：开发角色职责与技术选型总结

在本场景下，后端开发与算法开发需要紧密配合，构建一个高可用、高准确率的自动化问答系统。

角色分工与核心任务

角色	核心职责
后端开发	负责业务逻辑实现（CRUD）、数据库 Schema 设计、类目树层级维护、搜索接口封装、系统缓存优化以及与算法服务的 API 对接。
算法开发	负责文本预处理、BERT 模型微调/部署、向量库索引构建、相似度算法调优（召回与精排）以及语义理解（NLU）。

技术需求深度回答

1. 是否需要设计数据库？

必须设计。 数据库是整个系统的基石，主要包含以下核心表结构：

- **类目表**：存储一级、二级类目，采用父子 ID 结构（PID）实现树状管理。
- **FAQ 主表**：存储标准问题（Standard Question）、回答（Answer）、生效时间、创建人等。
- **相似问法表**：一个标准问题对应多个相似问法（1:N），用于扩大匹配范围。

2. 需要使用什么模型？

核心采用 **BERT（Bidirectional Encoder Representations from Transformers）** 及其变体（如 RoBERTa、MacBERT）。为了兼顾效率与精度，通常采用“向量召回 + 语义精排”的双阶段架构。

3. 如何使用 BERT？

- **特征提取**：将 FAQ 中的问题通过 BERT 转化为高维向量（Embedding）并存入向量数据库。
- **语义编码**：用户输入问题时，实时调用 BERT 提取特征向量。
- **相似度比对**：通过余弦相似度计算用户向量与库中向量的距离。

4. 是否需要使用大模型（LLM）?

建议结合使用。 虽然 BERT 擅长判别式匹配，但大模型（如 GPT-4，Qwen）可以用于：

- **语料扩充**：自动生成相似提问。
- **回答润色**：将 FAQ 的硬核回复转化为更拟人化的口吻。
- **长尾意图识别**：处理 BERT 难以理解的复杂指令。

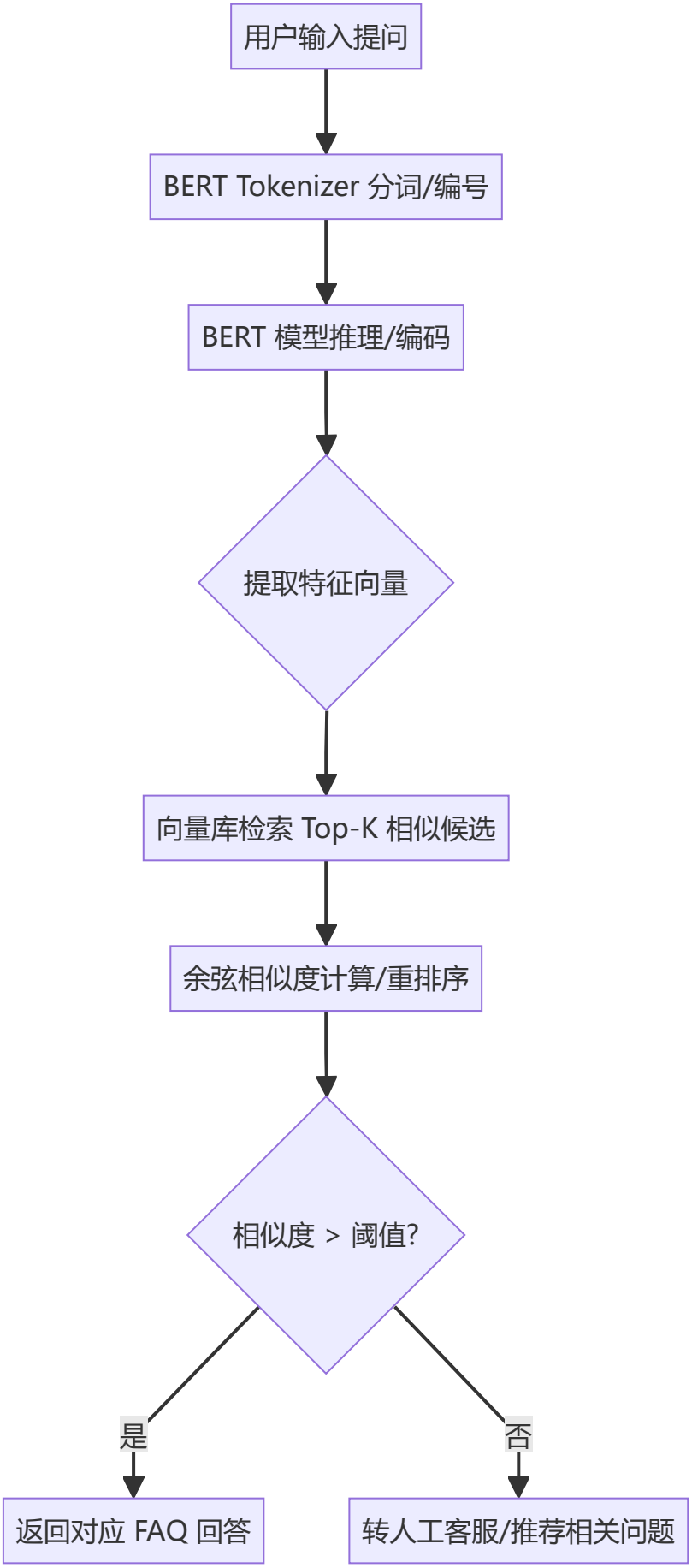
作业2：BERT 文本编码与相似度计算技术方案

本方案详细说明如何利用 BERT 模型实现从文本到语义向量的转化，并完成高效的相似度匹配。

技术方案说明

1. **文本预处理**：对原始文本进行清洗，包括去除特殊字符、分词（使用 BERT 自带的 WordPiece tokenizer）。
2. **编码阶段（Encoding）**：
    1. 将 Token 序列输入 BERT。
    2. 取 `[CLS]` 位的输出向量或对所有 Token 向量进行 `Mean Pooling`（平均池化），得到一个 768 维的稠密向量。
3. **索引与存储**：将历史问题的向量预先存入向量检索工具（如 FAISS，Milvus 或 OpenSearch）。
4. **计算阶段**：
    1. 在线计算用户输入的 Embedding。
    2. 使用**余弦相似度（Cosine Similarity）**公式计算夹角余弦值，分值越接近 1 则相似度越高。

相似度计算业务流程图


用户输入提问
BERT Tokenizer 分词/编号
BERT 模型推理/编码
提取特征向量
向量库检索 Top-K 相似候选
余弦相似度计算/重排序
相似度 > 阈值?
是
否
返回对应 FAQ 回答
转人工客服/推荐相关问题

方案核心指标

- **响应时间**：单次推理控制在 100ms 以内。
- **准确率（Top-1 Accuracy）**：通过对 `[CLS]` 向量进行三元组损失（Triplet Loss）微调，提升区分度。