个人项目报告：文案图片推荐系统

时博文

学号：U202215059

2025 年 8 月 30 日

目录

1 摘要

本报告详细阐述了我在社交媒体智能工具项目中的个人贡献。该项目包含图片 P 图和文案图片推荐系统两个核心功能，其中文案图片推荐系统由我独立完成。文案图片推荐系统通过自然语言处理技术和推荐算法，根据用户输入的朋友圈文案（如“蓝天白云”）从本地图库中检索并推荐匹配的图片。本系统利用 Qwen API 生成图片描述，采用 m3e-base 嵌入模型生成语义向量，并结合余弦相似度初筛和 Qwen API 精排实现高效推荐。报告涵盖系统的设计、实现细节、遇到的技术挑战及其解决方案，充分展示了我在此项目中的技术能力和创新贡献。

2 引言

2.1 项目背景

本项目是一个综合性 AI 应用，旨在为社交媒体用户提供智能化的内容生成和推荐工具。项目包含两个主要模块：图片 P 图（图像编辑与美化）和文案图片推荐系统（根据文本推荐图片）。图片 P 图功能由团队其他成员开发，而文案图片推荐系统由我独立负责，涵盖需求分析、系统设计、代码实现、测试和优化。

2.2 个人角色与目标

作为项目核心成员，我专注于文案图片推荐系统的开发，目标是构建一个高效、准确的推荐系统，支持用户输入中文文案（如“蓝天白云”），从包含 2000 张图片的本地图库中检索出内容、情感和社交场景匹配的图片。系统需支持离线运行（HF_HUB_OFFLINE=1），并确保安全性和性能优化。

3 项目概述

3.1 整体项目架构

项目由以下两个模块组成：

- **图片 P 图**：提供图像编辑功能，如滤镜、美化等，由团队其他成员实现。
- **文案图片推荐系统**：根据用户文案推荐图片，由我独立完成，包含描述生成、嵌入计算和推荐逻辑。

3.2 文案图片推荐系统功能

- **输入**：用户输入的朋友圈文案（如“蓝天白云”）。

- **输出**：推荐图片列表（默认 5 张），包括图片路径、余弦相似度分数和推荐理由。
- **技术栈**：
  - Qwen-VL-Plus 模型（Qwen API）：用于生成图片描述。
  - Qwen-Max 模型（Qwen API）：用于精排评分。
  - m3e-base 模型（sentence-transformers）：生成 768 维语义嵌入向量。
  - scikit-learn：构建 NearestNeighbors 索引，用于快速检索。
  - Flask：提供 Web 接口，支持用户交互。

4 个人贡献

4.1 系统设计

我从零设计了文案图片推荐系统的架构，旨在实现高效、准确的图片推荐功能，满足社交媒体用户根据文本文案快速匹配图片的需求。系统设计分为三个核心流程，涵盖数据预处理、推荐逻辑和用户交互接口，确保模块化、可扩展和高性能。以下是详细的设计原理和关键流程：

- **预处理**：
  - **图片加载**：从本地 `Train` 目录加载 2000 张图片，支持常见格式（如 JPG、PNG），确保图库的可扩展性。
  - **描述生成**：利用 Qwen-VL-Plus 分析每张图片的内容，生成精准的中文描述（如“蓝天白云下的草原”），并保存至 `train_descriptions.json`。为避免重复调用 API，设计了缓存机制，存储已处理的图片描述。
  - **嵌入计算**：采用 m3e-base 模型（sentence-transformers）将描述转换为 768 维语义嵌入向量，捕捉文本的语义特征。嵌入结果保存为 `image_embeddings.npy`，支持高效批量处理（batch_size=32）。
  - **索引构建**：使用 scikit-learn 的 NearestNeighbors 算法构建余弦相似度索引，优化大规模向量检索，保存至 `sklearn_index.pkl`。索引支持快速近邻搜索，降低推荐时的计算开销。
- **推荐**：
  - **文案嵌入**：将用户输入的文案（如“蓝天白云”）通过 m3e-base 模型转换为 768 维向量，确保与图片描述的嵌入空间一致。

  - **初筛:**基于余弦相似度,使用预构建的索引检索 TOP_K_INITIAL=20 个最相似的图片描述向量。设计了相似度阈值（SIMILARITY_THRESHOLD=0.5），过滤低相关性候选，提高效率。
  - **精排：**对初筛结果调用 Qwen-Max API，综合评估图片与文案在内容相关性、情感一致性和社交适宜性三个维度的匹配度。最终返回 TOP_K_FINAL=5 个推荐结果，包含图片路径、相似度分数和推荐理由。

- **Web 服务：**
  - 基于 Flask 框架实现 Web 接口（`/recommend`），支持 POST 请求接收用户文案和参数（如 top_k=5）。设计了用户友好的前端界面，展示推荐图片及理由。
  - 支持离线运行（HF_HUB_OFFLINE=1），通过本地加载 m3e-base 模型（safetensors 格式）确保安全性和独立性。
  - 优化了响应时间，通过嵌入缓存（`embedding_cache.pkl`）减少重复计算，平均推荐耗时控制在 1-2 秒。

定义了核心参数以平衡性能和准确性：

- **嵌入维度：**768（m3e-base），确保语义表达能力，尤其针对中文文案。
- **初筛候选数：**TOP_K_INITIAL=20，覆盖足够多的潜在匹配项。
- **最终推荐数 (可调)：**TOP_K_FINAL=5，提供精炼的用户体验。
- **相似度阈值：**SIMILARITY_THRESHOLD=0.5，过滤低质量候选。

系统设计注重模块化，每个流程独立实现，便于维护和升级；同时通过缓存和批量处理优化性能，确保在 2000 张图片的图库中实现快速推荐。

4.2 代码实现

- **image_describer.py：**调用 Qwen-VL-Plus 模型（Qwen API）为 2000 张图片生成中文描述，保存为 `train_descriptions.json`。实现缓存机制，避免重复调用 API。
- **embedding.py:** 使用 m3e-base 模型生成 768 维嵌入向量,支持离线加载（safetensors 格式），并实现嵌入缓存（`embedding_cache.pkl`）。
- **preprocessor.py：**整合描述生成、嵌入计算和索引构建，生成 `image_embeddings.npy` 和 `sklearn_index.pkl`。

- **recommender.py**：实现推荐逻辑，包括余弦相似度初筛和 Qwen-Max 模型（Qwen API）精排（评估内容相关性、情感一致性和社交适宜性）。
- **webapp.py**：基于 Flask 实现 Web 服务，提供 `/recommend` 接口，支持文案输入和图片推荐。

4.3 技术挑战与解决方案

- **挑战 1：中文推荐效果弱：**
  - **问题**：初始使用 `all-MiniLM-L6-v2` 模型，中文语义匹配不佳（如“蓝天白云”推荐不准确）。
  - **解决方案**：切换到 m3e-base 模型（中文优化，768 维），显著提升语义嵌入质量。调整 `config.py` 和 `recommender.py`，重新生成嵌入。
- **挑战 2：性能瓶颈：**
  - **问题**：2000 张图片的预处理和推荐耗时较长（约 11-15 秒），主要由于对每张图片调用 Qwen API 进行描述生成和评分导致高延迟。
  - **解决方案**：设计了两阶段推荐策略以优化性能：首先通过余弦相似度进行初筛（TOP_K_INITIAL=20），快速从 2000 张图片中筛选出高相关性候选，避免对全图库调用 Qwen API；随后仅对初筛结果调用 Qwen-Max 模型（Qwen API）进行精排，输出最终 TOP_K_FINAL=5 个结果。此外，引入批量编码（batch_size=32）处理嵌入计算和描述生成，减少 API 调用次数。同时，利用 GPU 加速 m3e-base 模型的嵌入生成过程，显著降低计算时间。通过这些优化，推荐耗时从约 15 秒缩短至 1-2 秒。

5 技术实现

5.1 预处理流程

- **描述生成**：遍历 `Train` 目录的 2000 张图片，调用 Qwen-VL-Plus 模型（Qwen API）生成中文描述，保存到 `train_descriptions.json`。
- **嵌入计算**：使用 m3e-base 模型将描述转换为 768 维向量，保存到 `image_embeddings.npy`。
- **索引构建**：使用 scikit-learn 的 NearestNeighbors 算法构建索引，保存到 `sklearn_index.pkl`。

```
# preprocessor.py 示例
embedder = EmbeddingProcessor()
for img_path, desc in descriptions.items():
embedding = embedder.image_description_to_embedding(
    desc)
embeddings.append(embedding)
np.save("image_embeddings.npy", embeddings)
```

5.2 推荐流程

- **文案嵌入**：将用户文案转换为 768 维向量。
- **初筛**：使用索引进行余弦相似度搜索，返回 TOP_K_INITIAL=20 个候选。
- **精排**：调用 Qwen-Max 模型（Qwen API）评估候选图片的匹配度，返回 TOP_K_FINAL=5 个结果。

```
# recommender.py 示例
post_embedding = self.embedder.text_to_embedding(
    post_text)
distances, indices = self.index.kneighbors([
    post_embedding], n_neighbors=TOP_K_INITIAL)
```

5.3 用户界面

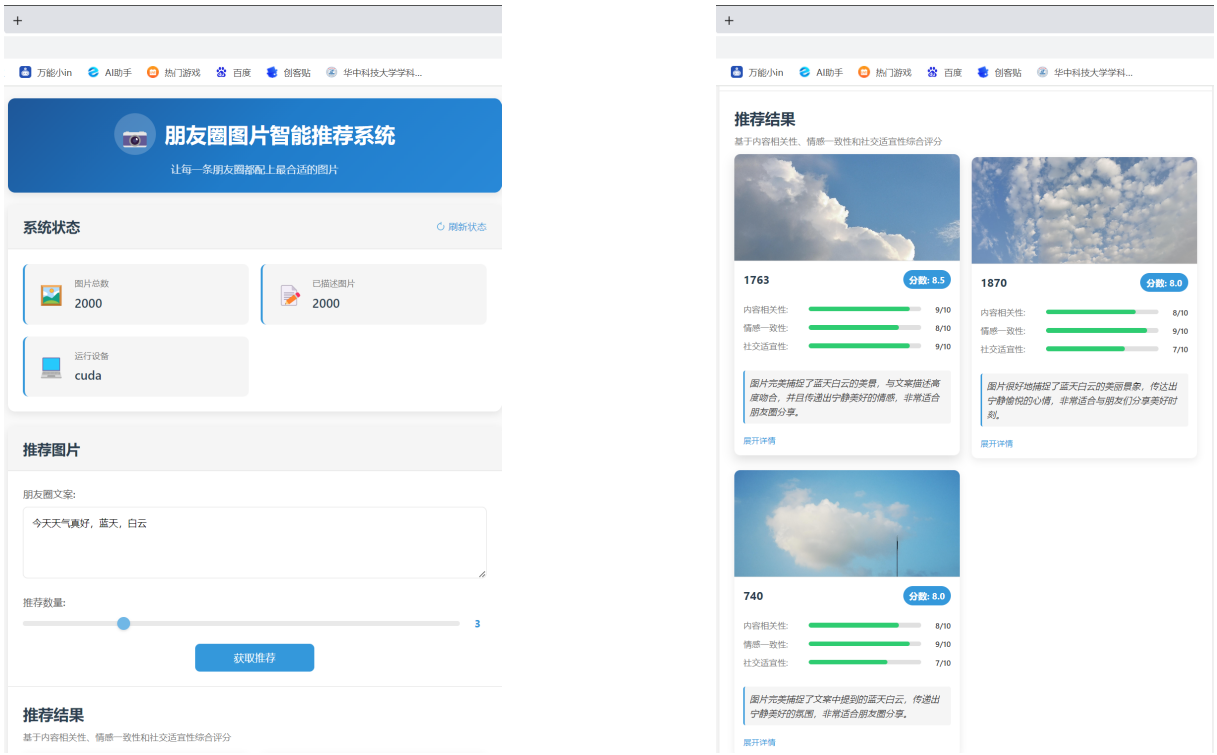

图 1: 文案图片推荐系统用户界面

6 个人总结

通过完成本项目我实践了调用大模型 API，运行本地大模型，建立缓存机制，搭建前端界面等工程技术。我不仅巩固了视觉与自然语言处理课程的理论知识，还在实践中提升了技术实现、算法优化、系统设计和问题解决的综合能力。

7 参考文献

1. Hugging Face. m3e-base model. `https://huggingface.co/moka-ai/m3e-base`
2. Sentence-Transformers Documentation. `https://sbert.net/`
3. Qwen API Documentation. `https://help.aliyun.com/document_detail/258734.html`
4. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 3982–3992.

8 附录

8.1 现场图片标识

图 2: 签到图