

《电子商务平台 SEO 推荐系统》

设计与实现项目实践报告

徐鸣飞、黄梓霖、皮佳宇、陈其阳

2023 年 10 月 20 日

目录

第一章 系统概述	1
1.1 系统主要目标	1
1.2 设计约束	1
1.3 开发、测试与运行环境	1
第二章 软件系统结构设计	2
2.1 系统设计图	2
2.2 各子系统功能介绍	3
2.2.1 数据收集子系统	3
2.2.2 输入数据处理子系统	3
2.2.3 用户行为分析子系统	4
2.2.4 关键词优化子系统	4
2.2.5 搜索引擎优化子系统	4
第三章 功能模块设计	6
3.1 模块说明	6
3.1.1 数据收集子系统概要设计	6
3.1.2 输入数据处理子系统概要设计	7
3.1.3 用户行为分析子系统概要设计	7
3.1.4 关键词优化子系统概要设计	8
3.1.5 搜索引擎优化子系统概要设计	9
3.2 模块之间的关系	10

目录	II
3.2.1 数据收集模块与输入数据处理模块	10
3.2.2 数据收集模块与用户行为分析模块	11
3.2.3 输入数据处理模块与用户行为分析模块	11
3.2.4 关键词优化模块与输入数据处理模块、用户行为分析模块	11
3.2.5 搜索引擎优化模块与输入数据处理模块	11
第四章 数据库设计	12
4.1 数据库环境	12
4.2 数据库表设计	12
4.3 数据库结构设计	15
4.4 安全性设计	15
4.4.1 数据保护	15
4.4.2 用户认证和授权	15
4.4.3 日志记录	15
4.4.4 防止 SQL 注入	16
4.4.5 网络安全	16
第五章 用户界面设计	17
5.1 搜索页面	17
5.2 搜索结果排序	18
第六章 总结	19

第一章 系统概述

1.1 系统主要目标

该电子商务平台 SEO 推荐系统的主要目标是通过关键词优化和搜索引擎优化，提升商品在搜索结果中的曝光度，增加用户点击率和购买转化率。系统需满足电子商务平台的需求，为商家提供关键词推荐和 SEO 优化建议，以优化其商品在搜索引擎中的排名。

1.2 设计约束

- 软硬件环境约束：系统应在常见的 Web 服务器环境下运行，同时需要与电子商务平台的数据库进行交互。
- 接口/协议约束：与电子商务平台的数据接口应遵循标准的 HTTP 或 HTTPS 协议。

1.3 开发、测试与运行环境

- 开发环境：使用常见的 Web 开发工具，如 Visual Studio Code；使用常见的 Java 后端开发工具，如 IntelliJ IDEA；以及相关的开发框架和库。
- 测试环境：模拟真实电子商务平台的测试环境，包括数据库和服务器。
- 运行环境：支持常见的 Web 浏览器，如 Chrome、Firefox 等，以及与电子商务平台兼容的服务器环境。

第二章 软件系统结构设计

2.1 系统设计图

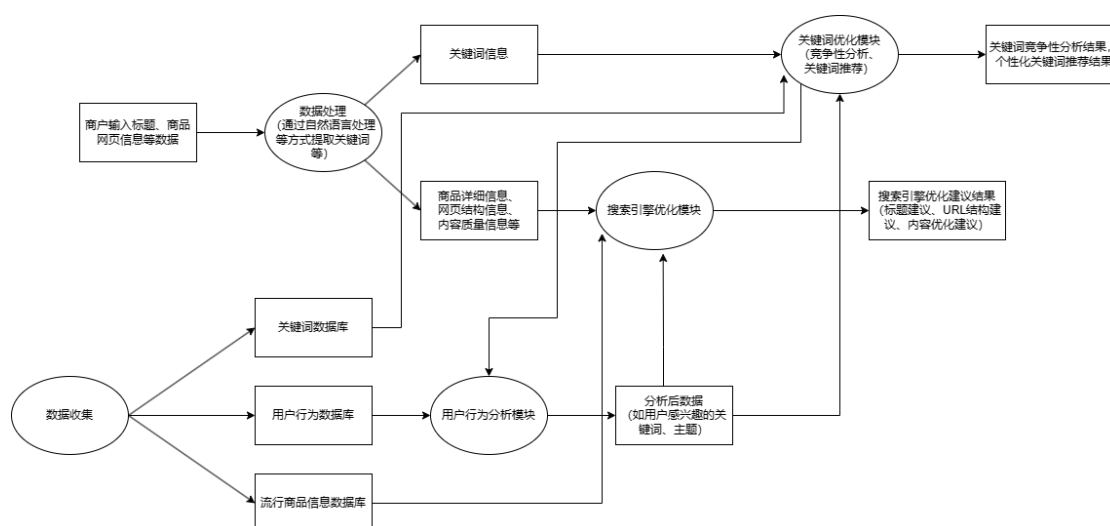


图 2.1: 系统结构设计示意图

全系统分为 5 个子系统：

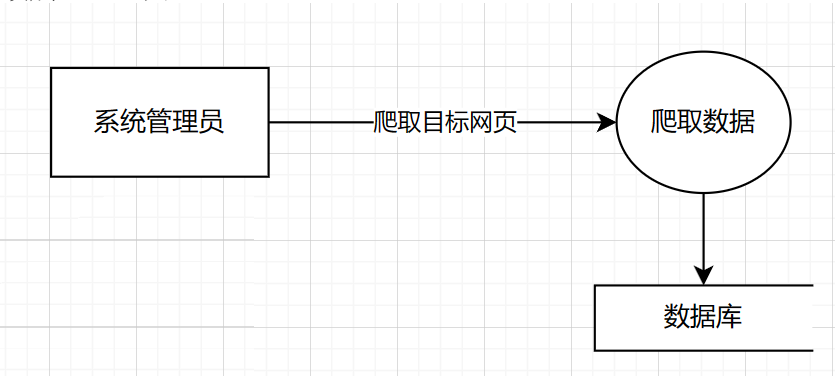
1. 数据收集子系统
2. 输入数据处理子系统
3. 用户行为分析子系统
4. 关键词优化子系统
5. 搜索引擎优化子系统

2.2 各子系统功能介绍

接下来粗略介绍各子系统功能。

2.2.1 数据收集子系统

顶层 DFD 图：

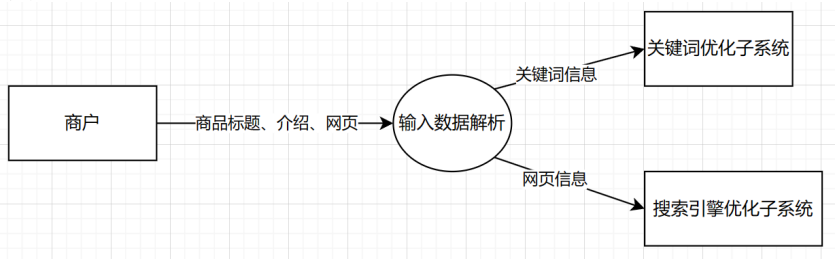


功能：

- 从互联网上爬取流行商品的网页数据，包括商品标题、描述、关键词等。
- 收集用户在电子商务平台上的搜索和购买行为，包括关键词、点击次数、购买次数等。
- 提供直接编辑 3 大数据库的 API 接口。

2.2.2 输入数据处理子系统

顶层 DFD 图：



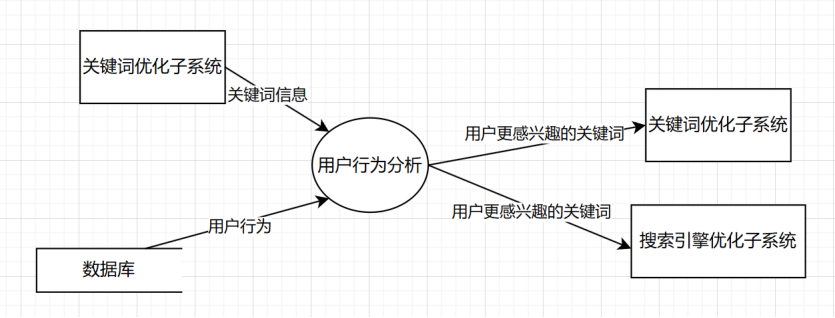
功能：

- 处理商家输入的商品信息，提取关键词组。

- 获取指定商品网页的结构数据和内容数据。
- 分析网页内容，生成内容质量数据。

2.2.3 用户行为分析子系统

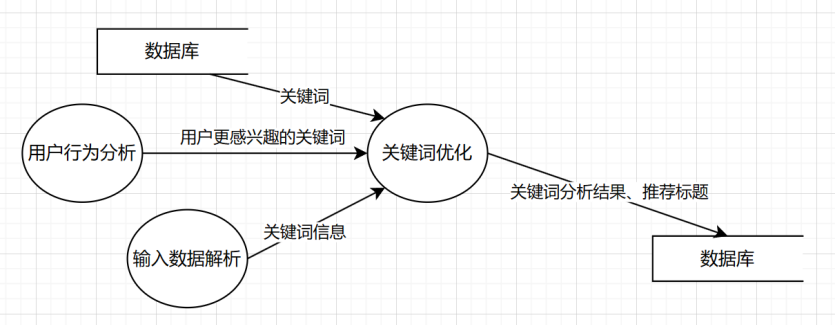
顶层 DFD 图：



功能：分析用户行为数据库和关键词数据库，识别用户感兴趣的商品关键词。

2.2.4 关键词优化子系统

顶层 DFD 图：

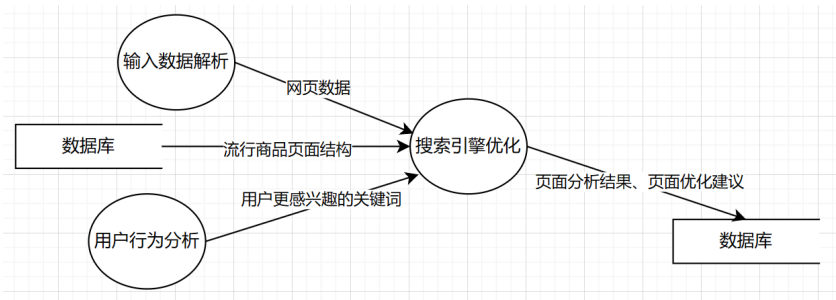


功能：

- 分析商家输入的关键词组和关键词数据库，提供关键词竞争性分析结果。
- 提供个性化的关键词推荐结果。

2.2.5 搜索引擎优化子系统

顶层 DFD 图：



功能：根据商家的网页内容、网页结构和内容质量数据，提供搜索引擎优化建议结果，包括标题建议、URL 结构建议、内容优化建议。

第三章 功能模块设计

3.1 模块说明

3.1.1 数据收集子系统概要设计

设计思路：该模块涉及 3 大数据库生成，将从数据库角度分析实现思路。

- 关键词数据库数据生产：关键词数据库将存储

< 关键词 : 中介关键词 : 竞争关键词 : 相关程度 >

每次运行 compkey 算法将生成一列。鉴于提前生成难度较大，将在每次商家提供商品标题、商品信息，并从中提取关键词组后，再运行 compkey 算法，并将运行结果存入关键词数据库以备使用。(训练数据集可上传指定，或使用默认训练数据集)

- 用户行为数据库数据生成：用户行为数据库将存储

< 用户行为类型 (点击/收藏/购买) : 商品名称 >

使用方式：提供关键词，查看包含该关键词的所有商品名称的点击/收藏/购买数。

获得方式：寻找开源数据或爬取购物网站相关数据 (商品名 + 访问次数/销售量/收藏人数)。

- 流行商品信息数据库数据生成：该数据库将存储

< 商品名 : 网页 URL : 商品热度指标 >

使用方式：提供关键词正则，找到商品名符合正则的记录，按热度指标排序，根据 URL 爬取网页内容和网页结构，以备搜索引擎优化模块的需求。

获得方式：爬取购物网站热销商品列表，并提取 URL，存入数据库。

3.1.2 输入数据处理子系统概要设计

该子系统涉及两方面内容，将依次介绍设计思路：

- 关键词优化模块：
 1. 使用自然语言处理（NLP）技术，可以考虑使用词袋模型（Bag of Words）或者更先进的词嵌入模型（如 Word2Vec、BERT）来提取商品名称和介绍中的关键词。
 2. 或利用分词工具对商品名称和介绍进行分词，然后通过关键词提取算法获取关键词。
- 搜索引擎优化模块：
 1. URL 处理：使用爬虫技术，例如 Python 中的 BeautifulSoup 或 Scrapy 库，来访问商品 URL 地址并获取网页内容。
 2. 网页结构：使用 BeautifulSoup 或类似的 HTML 解析库，遍历 HTML 文档的标签，获取标签的层次结构和关系。
 3. 内容提取：分析网页结构，可以使用正则表达式或者 XPath 等方法来提取网页中的关键信息。
 4. 内容质量评价：用自然语言处理（NLP）技术对网页内容进行情感分析，以了解评论或描述的情感倾向；使用模型检测文本的一致性和逻辑性，以评估网页描述的准确性和清晰度。（调用文言一心、GPT 等大模型，生成文本、图片的评价，后续可继续调用模型进行文本优化）

3.1.3 用户行为分析子系统概要设计

该子系统可以根据提供的关键词组，返回较受用户青睐的相关关键词组。

设计思路：该模块会被关键词优化子系统所调用，关键词优化子系统会提供一系列与商品关键词相关的关键词组，用户行为分析模块负责使用正则从用户行为数据库中统计与各关键词有关的用户行为数量，并以某类权重公式计算各类型用户行为后，返回 < 关键词: 推荐程度 > 集合。

3.1.4 关键词优化子系统概要设计

该子系统负责多个功能，将依次介绍设计思路：

- 相关关键词组生成：使用 Compkey 算法，种子关键词为之前输入数据处理模块提取的关键词组，从默认数据集（或指定数据集）中提取出一定数量的中介关键词和竞争关键词，计算关联程度，并存储到关键词数据库；如果数据库已存在该关键词的相关数据库，则直接读取。
- 竞争度分析：将竞争关键词和竞争程度返回给用户。如：
“手机”：“高”，“平板电脑”：“中”，“智能手表”：“中”
- 关键词推荐：将用户行为分析模块中推荐程度较高的关键词返回给用户。输入输出模拟：

输入：

```
{ "推荐关键词": "电脑", "推荐程度": 0.8 },  
{ "推荐关键词": "游戏", "推荐程度": 0.7 },  
{ "推荐关键词": "笔记本", "推荐程度": 0.9 },  
{ "推荐关键词": "性能", "推荐程度": 0.85 }
```

输出：

“笔记本”，“性能”，“电脑”

- 商品名称生成：使用 GPT、文言一心等大模型，使用固定语法提问，提供原始商品名称、原始关键词、推荐关键词等信息，生成合适的商品名称。输入输出模拟：

输入：

```
"原始商品名称": "智能手机",  
"原始关键词": ["手机", "智能"],  
"推荐关键词": ["高性能", "全面屏"]
```

输出：

“高性能智能手机全面屏版”

3.1.5 搜索引擎优化子系统概要设计

该子系统的作用是提供各类搜索引擎优化建议，将依次介绍不同类型建议生成的设计思路：

1. 关键词优化建议：

- (a) 统计网页中的出现的原始关键词、推荐关键词频率、分布，以及提供替代关键词建议。
- (b) 密度建议：根据关键词频率和网页总词数，提供关键词密度建议。关键词密度即关键词在文本中的出现次数与总词数的比例。
- (c) 分布建议：统计关键词的分布状况，提供关键词在网页中的合理分布建议。避免过度/过少使用关键词，并确保它们自然地融入文本中。
- (d) 替代关键词建议：使用了词袋模型和余弦相似度来度量用户提供的关键词与网页内容中其他词的相似度。然后，根据相似度排序，选择了最相似的词作为替代关键词，以丰富网页内容。

2. 网页结构优化建议：

- (a) 标题标签建议：检查每个标题标签的使用情况，并提供建议。可以检查层次结构深度（例如，h2 标签应该在 h1 标签之后使用，h3 标签应该在 h2 标签之后使用）、分布情况（每类标题标签的使用次数，以及它们在文档中的分布），提供建议如：

h1: 没有建议，使用适当。

h2: 建议：调整 *h2* 标签的层次结构或增加使用频率，以提高合适性。

h3: 建议：调整 *h3* 标签的层次结构或增加使用频率，以提高合适性。

h4: 建议：添加至少一个 *h4* 标签以提升页面结构。

h5: 建议：添加至少一个 *h5* 标签以提升页面结构。

- (b) 段落和文本结构建议：检查段落标签（如 p 标签）的使用情况，并生成建议。可以检查段落标签（如 p 标签）和其他文本标签的分布情况、平均段落长度、段落之间的平均间距等，提供建议如：

段落标签 *p* 使用情况: 2

其他文本标签使用情况: 3

建议:

长度建议: 某些段落过长, 请考虑分割为更短的段落。

标签使用建议: 考虑使用更多其他文本标签 (如 *span*、*div* 等) 来组织文本。

- (c) 图像和多媒体元素建议: 检查图像和多媒体元素的使用情况, 可以检查是否提供 alt 属性、图像大小格式等是否合适, 并提供建议, 如:

建议: 图像 *img_1* 缺少描述文本 (*alt* 属性), 请添加以提高可访问性。

建议: 图像 *img_2* 可能过大, 请考虑优化图像大小以提高加载性能。

- (d) 链接结构建议: 分析页面中的链接结构, 检查链接是否具有描述、是否为空链接等。

- (e) 页面加载速度建议: 分析页面加载速度, 提供建议以改善加载性能。如检查是否使用 CDN 网络、是否使用异步延迟加载、是否使用适当的缓存头 (如 Cache-Control)、图像格式是否为 WebP 格式而不是 PNG 或 JPEG 等。

3. 文本内容优化建议:

- (a) 根据输入数据处理模块提供的内容质量评价, 提示用户修改建议内容; 也可继续使用 GPT 大模型或其他文件优化类模型, 生成合适的替换内容以供用户选择。提问方式如: 根据以下网页文本信息, 根据 SEO 最佳实践建议, 提供语言和内容的优化建议:

3.2 模块之间的关系

3.2.1 数据收集模块与输入数据处理模块

- 数据收集子系统负责从互联网上爬取流行商品的网页数据, 并将其存储到数据库中。
- 输入数据处理子系统负责处理商家输入的商品信息, 提取关键词组, 并获取指定商品网页的结构数据和内容数据。
- 数据收集子系统提供的网页数据成为输入数据处理的一部分, 用于分析商家输入的商品信息, 从而提取关键词组和生成内容质量数据。

3.2.2 数据收集模块与用户行为分析模块

- 数据收集子系统收集用户在电子商务平台上的搜索和购买行为数据。
- 用户行为分析子系统利用数据收集子系统提供的用户行为数据库，分析用户的搜索和购买行为，以识别用户感兴趣的关键词。

3.2.3 输入数据处理模块与用户行为分析模块

- 输入数据处理子系统提供了商家输入的商品信息和相应网页数据。
- 用户行为分析子系统结合商家输入的商品信息和用户行为数据，分析用户对商品的兴趣，进一步提供优选关键词。

3.2.4 关键词优化模块与输入数据处理模块、用户行为分析模块

- 关键词优化子系统分析商家输入的关键词组和关键词数据库。
- 输入数据处理子系统提供了商家输入的商品信息和关键词组，而用户行为分析子系统提供了用户对商品的兴趣数据。
- 关键词优化子系统综合这些信息，提供关键词竞争性分析和个性化的关键词推荐结果。

3.2.5 搜索引擎优化模块与输入数据处理模块

- 搜索引擎优化子系统根据商家的网页内容、网页结构和内容质量数据提供优化建议。
- 输入数据处理子系统提供了商家输入的商品信息和相应网页数据，为搜索引擎优化提供基础数据。

第四章 数据库设计

4.1 数据库环境

数据库系统：MySQL 8。

设计工具：Navicat。

编程工具：Navicat 提供的 SQL 编辑器。

4.2 数据库表设计

数据收集模块需要 3 个表格：

```
1  —————
2  — Table structure for behavior
3  —————
4  DROP TABLE IF EXISTS behavior;
5  CREATE TABLE behavior (
6      id int NOT NULL AUTO_INCREMENT,
7      behavior int NOT NULL,
8      commodity varchar(255) CHARACTER SET utf8mb4 COLLATE
9          utf8mb4_0900_ai_ci NOT NULL,
10     PRIMARY KEY (id) USING BTREE
11 ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE =
    utf8mb4_0900_ai_ci ROW_FORMAT = Dynamic;
```

```
12  —————
13  — Table structure for fashion
14  —————
15  DROP TABLE IF EXISTS fashion;
16  CREATE TABLE fashion (
17      id int NOT NULL AUTO_INCREMENT,
18      commodity varchar(255) CHARACTER SET utf8mb4 COLLATE
19          utf8mb4_0900_ai_ci NOT NULL,
20      URL varchar(255) CHARACTER SET utf8mb4 COLLATE
21          utf8mb4_0900_ai_ci NOT NULL,
22      degree float NOT NULL,
23      PRIMARY KEY (id) USING BTREE
24  ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE =
25      utf8mb4_0900_ai_ci ROW_FORMAT = Dynamic;
26
27  —————
28  — Table structure for keyword
29  —————
30  DROP TABLE IF EXISTS keyword;
31  CREATE TABLE keyword (
32      id int UNSIGNED NOT NULL AUTO_INCREMENT,
33      keyword varchar(255) CHARACTER SET utf8mb4 COLLATE
34          utf8mb4_0900_ai_ci NOT NULL,
35      between varchar(255) CHARACTER SET utf8mb4 COLLATE
36          utf8mb4_0900_ai_ci NOT NULL,
37      compete varchar(255) CHARACTER SET utf8mb4 COLLATE
38          utf8mb4_0900_ai_ci NOT NULL,
39      correlation float NOT NULL,
40      PRIMARY KEY (id) USING BTREE
```



```
35 ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE =  
    utf8mb4_0900_ai_ci ROW_FORMAT = Dynamic;
```

其他表格：此程序主体只需要上述三个表格，其他表格用于使应用更加完善。表格结构设计已于3.1.1中介绍。用户表：

```
1  —————  
2  — Table structure for user  
3  —————  
4  DROP TABLE IF EXISTS user;  
5  CREATE TABLE user (  
6      id int NOT NULL AUTO_INCREMENT,  
7      name varchar(255) CHARACTER SET utf8mb4 COLLATE  
        utf8mb4_0900_ai_ci NOT NULL,  
8      password varchar(255) CHARACTER SET utf8mb4 COLLATE  
        utf8mb4_0900_ai_ci NOT NULL,  
9      usertype int NOT NULL,  
10     PRIMARY KEY (id) USING BTREE  
11 ) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE =  
    utf8mb4_0900_ai_ci ROW_FORMAT = Dynamic;
```

结果记录表：

```
1  —————  
2  — Table structure for result  
3  —————  
4  DROP TABLE IF EXISTS result;  
5  CREATE TABLE result (  
6      id int NOT NULL AUTO_INCREMENT,  
7      userid int NOT NULL,  
8      keyword_result varchar(255) CHARACTER SET utf8mb4
```

```
9          COLLATE utf8mb4_0900_ai_ci NULL DEFAULT NULL,  
seo_result varchar(255) CHARACTER SET utf8mb4 COLLATE  
10          utf8mb4_0900_ai_ci NULL DEFAULT NULL,  
11 PRIMARY KEY (id) USING BTREE  
) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE =  
utf8mb4_0900_ai_ci ROW_FORMAT = Dynamic;
```

4.3 数据库结构设计

4.4 安全性设计

4.4.1 数据保护

为了确保敏感数据的机密性和完整性，采取以下措施：

- 数据库连接采用加密传输，使用 SSL/TLS 协议。
- 敏感数据字段采用加密算法 RSA 进行存储。
- 实施备份策略，确保数据定期备份，并存储在安全的地方。

4.4.2 用户认证和授权

确保只有授权用户能够访问系统，并按照其角色分配相应权限：

- 用户密码存储采用哈希算法，并添加盐以增强安全性。
- 实施多因素身份验证，增加用户登录的安全性。
- 对用户进行细粒度的授权，限制其访问敏感数据和操作。

4.4.3 日志记录

建立全面的日志记录系统，以便监控系统的行为并进行安全审计：

- 记录所有用户登录和操作历史。
- 使用合适的日志级别，确保能够追踪潜在的安全威胁。
- 对日志进行定期审查，及时发现异常或可疑活动。

4.4.4 防止 SQL 注入

通过使用参数化查询或预编译语句等方式，防止 SQL 注入攻击：

- 不直接拼接用户输入到 SQL 查询中。
- 使用 ORM（对象关系映射）框架，自动处理参数化查询。
- 对输入数据进行严格的验证和过滤，防止恶意输入。

4.4.5 网络安全

加强网络安全措施，防范网络攻击：

- 使用防火墙限制对数据库的访问。
- 定期更新系统和数据库软件，修复潜在的安全漏洞。
- 监测和阻止恶意网络流量，例如 DDoS 攻击。

第五章 用户界面设计

5.1 搜索页面

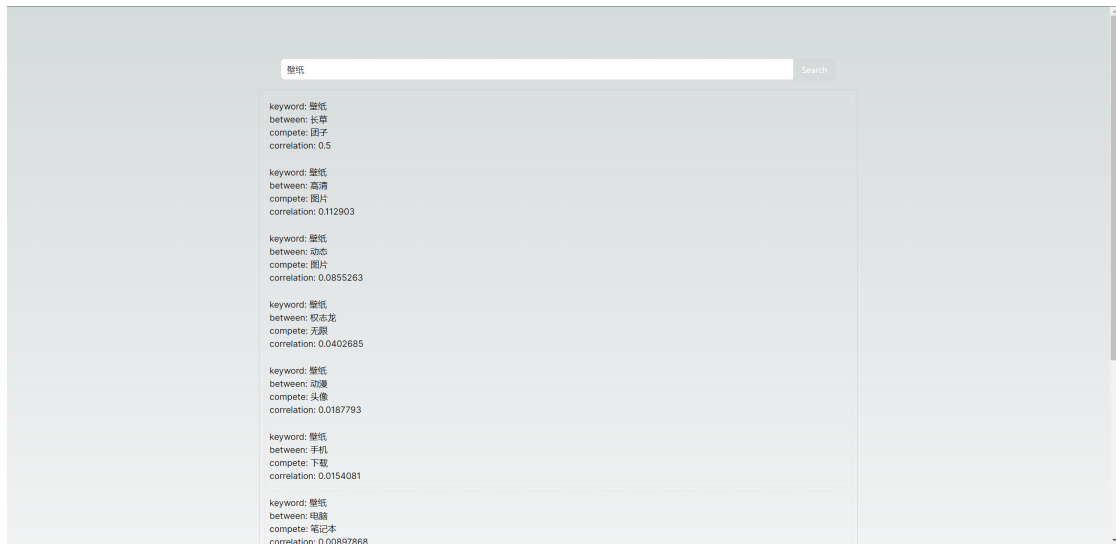


图 5.1: 搜索页面

- ** 搜索栏: ** [文本框] [搜索按钮]

- ** 搜索结果: **

结果 1

结果 2

结果 3

5.2 搜索结果排序

搜索结果根据竞争度进行排序展示，可以在页面上方或侧边显示排序选项。

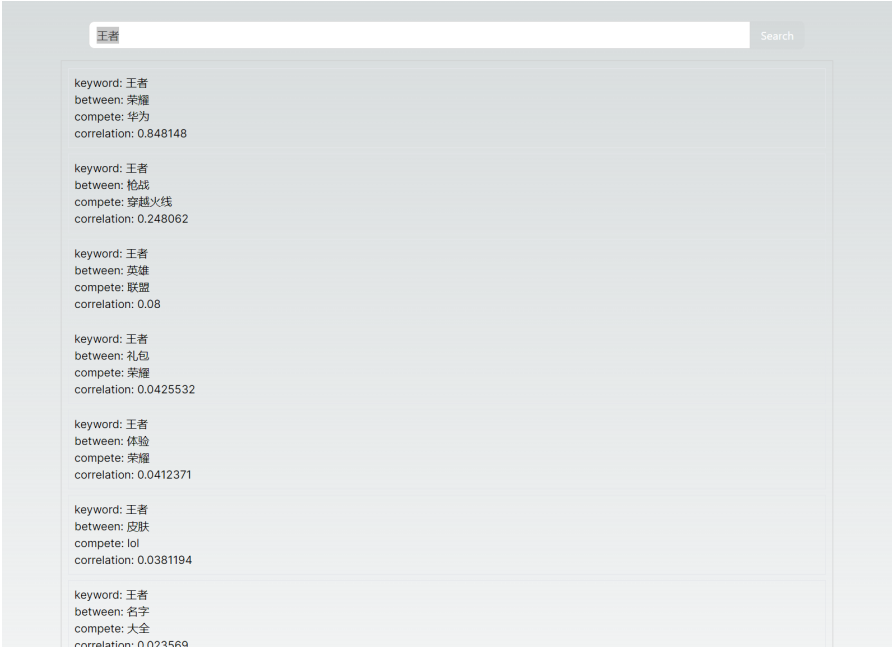


图 5.2: 搜索结果排序

- ** 排序选项: **

默认排序

竞争度高到低

竞争度低到高

第六章 总结

电子商务平台 SEO 推荐系统的核心目标是通过关键词优化和搜索引擎优化，提升商品在搜索结果中的曝光度，以增加用户点击率和购买转化率。系统设计以满足电子商务平台的需求为基础，为商家提供关键词推荐和 SEO 优化建议，从而最大程度地优化其商品在搜索引擎中的排名。

在本项目中，我们建立了一个综合性的解决方案，包括数据收集模块、用户管理模块和推荐系统。数据收集模块广泛涵盖了用户行为、商品信息和关键词等多个方面的数据，为系统提供了丰富的信息基础。用户管理模块通过用户表和结果记录表，实现了对用户身份的有效管理和对推荐结果的记录。推荐系统充分利用了关键词、商品信息和用户行为等多源数据，通过竞争度排序等策略为商家提供有针对性的 SEO 优化建议。

通过这个综合性的解决方案，电子商务平台能够更好地理解用户行为、商品特征以及市场竞争状况。商家可以根据系统提供的推荐，有针对性地进行关键词优化和 SEO 调整，从而提高商品在搜索引擎中的曝光度，吸引更多潜在客户。同时，系统的用户管理模块也为商家提供了用户行为的记录和分析，有助于更好地了解用户需求，提高用户体验。

在未来的发展中，该系统可以通过引入机器学习算法不断优化推荐策略，提高推荐的精准度。同时，随着电子商务行业的不断发展，系统的功能和性能也可以不断升级，以适应不断变化的市场需求。总体而言，电子商务平台 SEO 推荐系统的实施将为商家带来更为精细化、智能化的运营管理，提升竞争力，实现更好的商业价值。