分 类 号：_____ 密 级：_____

学校代码：10414 学 号：202141600008

江西师范大学

硕士专业学位研究生学位论文

基于无监督关键词提取算法的聚合搜索系统的设计与实现

Design and Implementation of an Aggregated Search System Based on Unsupervised Keyword Extraction algorithms

王烨锴

院　　所：数字产业学院　　导师姓名：揭安全 李宏伟

专业学位类别：电子信息　　专业领域：计算机技术

二〇二四年五月

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示谢意。

学位论文作者签名：王烨锴　　签字日期：2024年5月25日

学位论文版权使用授权书

本学位论文作者完全了解江西师范大学研究生院有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的电子版和纸质版，允许论文被查阅和借阅。本人授权江西师范大学研究生院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：王烨锴　　导师签名：揭安全

签字日期：2024年5月25日　　签字日期：2024年5月26日

摘　要

随着互联网的迅速扩张，信息量呈现指数级增长，覆盖网页、社交媒体、新闻和商品目录等多个领域。面对信息碎片化、孤岛和过载等挑战，传统搜索引擎在检索效率和满足用户多样化需求方面显示出不足。因此，构建一个综合性数据检索平台对于提升信息检索的效率和质量至关重要。

在关键词提取技术上，本文提出了一种无监督的指数权重衰减关键词提取方法 EdecayRank。该方法在现有最先进的 PromptRank 算法的基础上，融合了 TextRank 算法的连贯性、ChatGLM 算法的语义一致性以及 PattenRank 算法的多样性。在 Inspec、DUC2001、NUS、Krapivin 四个数据集上的实验结果显示，EdecayRank 的 F1@5 分数分别为 33.05、30.93、20.48、18.42，平均 F1@5 得分为 25.72，较第二名的 PromptRank 方法高出 2.60。此外，EdecayRank 在返回 top5、top10 和 top15 个关键词时的 F1 分数分别提升了 6.49%、5.94%和 4.42%，显著增强了关键词提取的精确度。

针对信息碎片化、孤岛化和过载等问题，本文介绍了一种高效的聚合搜索系统。该系统能够整合多源信息，使用 EdecayRank 方法进行关键词提取，并利用 Elasticsearch 进行高效检索和相关性排序，从而向用户提供精确且全面的搜索结果。系统首先通过爬虫技术和数据清洗确保数据质量，随后将提取的关键词与源数据一并存储至数据库中。通过设计的同步机制，系统与 Elasticsearch 保持实时数据同步，有效减少了数据传输成本和同步时间。整合 Elasticsearch 搜索引擎后的系统实现了快速文档搜索和实时数据分析的功能。这一系列技术的结合不仅显著提高了信息检索的效率，而且极大改善了用户的搜索体验，对信息检索技术的进步贡献了重要的力量。

关键词：关键词提取；无监督算法；集成学习；组合策略；聚合搜索系统；Elasticsearch

Abstract

With the rapid expansion of the Internet, the amount of information has grown exponentially, covering many fields such as web pages, social media, news and commodity catalogs. Faced with challenges such as information fragmentation, islanding, and overload, traditional search engines have shown shortcomings in terms of retrieval efficiency and meeting diverse user needs. Therefore, building a comprehensive data retrieval platform is crucial for improving the efficiency and quality of information retrieval.

In terms of keyword extraction technology, this paper proposes an unsupervised exponential weight decay keyword extraction method, EdecayRank. This method combines the coherence of TextRank algorithm, semantic consistency of ChatGLM algorithm, and diversity of PattenRank algorithm on the basis of the most advanced existing PromptRank algorithm. The experimental results on four datasets, namely Inspect, DUC2001, NUS, and Kravivin, show that EdecayRank's F1@5 The scores are 33.05, 30.93, 20.48, and 18.42 respectively, with an average of F1@5 The score is 25.72, which is 2.60 higher than the second place PromptRank method. In addition, EdecayRank's F1 scores improved by 6.49%, 5.94%, and 4.42% when returning the top 5, top 10, and top 15 keywords, respectively, significantly enhancing the accuracy of keyword extraction.

This article introduces an efficient aggregation search system to address issues such as information fragmentation, islanding, and overload. The system is capable of integrating multiple sources of information, using the EdecayRank method for keyword extraction, and utilizing Elasticsearch for efficient retrieval and relevance ranking, thereby providing users with accurate and comprehensive search results. The system first ensures data quality through web scraping technology and data cleaning, and then stores the extracted keywords along with the source data in the database. Through the designed synchronization mechanism, the system maintains real-time data synchronization with Elasticsearch, effectively reducing data transmission costs and synchronization time. The system integrated with Elasticsearch engine has achieved the functions of fast document search and real-time data analysis. The combination of this series of technologies not only significantly improves the efficiency of information retrieval, but also greatly improves the user search experience, contributing an important force to the progress of information retrieval technology.

Key words: keyword extraction; unsupervised algorithms; ensemble learning; combination strategy; aggregation search system; Elasticsearch

目　　录

1 绪　论

1.1 课题背景及意义

1.1.1 课题背景

互联网已经悄然融入生活的每一个角落，成为我们日常生活不可分割的一部分。随着全球网络用户数量逐渐增加，用户生成的信息量也在急剧膨胀，社交媒体、博客、视频平台等成为了内容创作和分享的温床，不断推动信息量的增长。从医疗健康到交通运输，从教育课程到餐饮购物服务，乃至金融、环保等多个行业领域，互联网的融合程度逐渐加强，预示着互联网内容将会以指数形式急剧增加。

在海量互联网数据中，目前常见的方法是运用通用搜索引擎来挖掘用户感兴趣的各种信息。这些搜索引擎，例如国外的 Yahoo、Google、Bing，以及国内的百度、搜狗、360 综合搜索等，都是从互联网上收集数据，来为用户提供搜索结果。虽然这些工具能够提供广泛的搜索选项，但当用户试图获取特定领域的具体信息时，这些通用的搜索引擎却不能完全满足他们的搜索需求。为了解决上述问题，垂直搜索引擎应运而生，它们主要针对特定领域或主题，提供更为精确和专业的搜索结果。市场中已经涌现了各种针对特定领域的搜索产品，例如专注于旅游领域的“携程”、专注于汽车领域的“汽车之家”等。

但是随着信息来源的多样化和信息量的激增，单一的垂直搜索引擎也难以完全满足用户的多元化需求。因此，出现了聚合搜索技术。该技术通过整合多个来源的信息，为用户提供更精确且全面的搜索结果，帮助用户更迅速地找到所需信息。聚合搜索技术的发展反映了人们对更精确、更全面、更个性化信息获取方式的追求，同时也推动了搜索技术的持续创新和发展。

由此可见，为用户搭建一个聚合搜索系统是提高信息检索效率和满足多样化需求的关键一环。

1.1.2 课题意义

在当前数据量急速膨胀的时代背景下，公众对于数据获取的效率与准确性提出了更为严苛的要求。为了适应这些高标准要求，迫切需要开发性能更优越的关键词提取算法，从而增强机器对文本内容的解析能力。这对于文本摘要生成、信息检索、文本分类以及主题识别等任务都将带来深远的影响。

尽管传统搜索引擎极大地简化了信息获取的流程，但其在处理信息碎片化、信息孤岛化以及信息过载等问题上存在一定的局限性，加重了用户检索信息的负担。垂直搜索和聚合搜索技术的兴起，有效弥补了传统搜索引擎的不足。垂直搜索专注于特定领域或类型的信息，提供更为精确和专业的搜索结果；而聚合搜索则通过整合多元信息源，为用户带来更为全面和丰富的搜索体验。

本文详细介绍了一种无监督的指数权重衰减关键词提取方法 EdecayRank，并在此方法的基础上设计和实现了一个聚合搜索系统。EdecayRank 方法对现有的无监督关键词提取方法进行了改进，显著提升了搜索引擎搜索结果的相关性；而聚合搜索系统通过整合多种信息源，为用户提供更为全面和个性化的搜索体验。这种技术的整合对信息检索领域的进步具有重要的推动作用。

1.2 国内外研究现状

1.2.1 关键词提取技术

关键词提取技术主要分为有监督方法和无监督方法。有监督方法虽然在精确度方面表现卓越，但其应用受限于对大规模标注数据集的依赖。相较之下，无监督方法因其不依赖于标注数据，且具备较强的通用性，在学术研究和实际应用中占据主导地位。其中 PromptRank[1]使用 prompt 来解决文档与候选关键词长度不一致的问题。该方法通过编码器－解码器结构将原始文档和 prompt 扩展的候选关键词映射到共享潜在空间，并利用解码器对候选关键词进行概率排名。尽管 PromptRank 提高了长文档处理的性能，但其依赖于手动设计的 prompt，存在主观性和局限性。此外，缺乏自动化的 prompt 优化方法限制了其发展。

未来的研究将致力于集成多种无监督关键词提取算法，目的是通过整合各个算法的优势，来提升关键词提取方面的性能和准确性。

1.2.2 爬虫技术

随着网络数据的飞速增长,网络爬虫技术已成为从互联网上搜集数据的重要工具。在国内，科学研究者与开发团队已经在爬虫算法优化、分布式爬虫系统构建，以及数据抓取行为的合法性和道德问题方面进行了深度研究和探讨。Python 语言凭借其简明的语法结构和充足的库支持，成为了开发爬虫程序首选的语言。其中，Scrapy 框架作为一个开源的 python 爬虫工具包，为爬虫的研究和使用带来了诸多强大工具，极大地简化了爬虫的开发流程，推动了爬虫技术的探索和实际应用。

Scrapy 框架以其模块化的设计理念，使得开发者能够便捷地对爬虫软件进行个性化定制以及功能的扩充。同时，该框架支持快速开发流程，极大地提高了开发效率，满足了对大规模数据捕获和处理的需求。此外，Scrapy 还内置了对 robots.txt 协议的支持,确保爬虫活动严格遵循网站的抓取规则，维护了数据抓取的合法性和道德标准。

未来的研究将进一步深化对爬虫技术的理解，并推动其在多个领域的广泛应用。

1.2.3 搜索引擎

在这个数据量迅速增长和数据来源日益多样化的时代背景下，传统的关键词匹配和索引方法已经不能达到用户对信息高效查找的期望。因此，搜索引擎目前主要聚焦于推动搜索技术的智能化、个性化以及提高搜索效果的准确性。为了能够更好地理解用户的查询意图并提高搜索结果的相关性和质量,搜索引擎需要整合先进的机器学习和自然语言处理等前沿技术。

在此背景之下，Elasticsearch 凭借其独特的架构和强大的功能引起了业界的广泛关注。这款具有高度扩展性的开源搜索引擎，整合了分布式存储和实时数据处理等多项功能，从而能够高效地操作大规模的结构化数据和非结构化数据。Elasticsearch 最大的优势在于其全文搜索能力，它不仅能够为用户提供迅捷、精确的搜索结果，还能在处理庞大数据量时保持一致的高性能，确保用户在任何情况下都能获得良好的搜索体验。

凭借其在大数据处理、实时搜索、高可伸缩性和易用性等方面的突出表现，Elasticsearch 在搜索引擎领域占据了重要地位。

1.2.4 聚合搜索

聚合搜索技术通过整合多个源数据，例如网页、新闻、图片、视频以及社交媒体等，使用户能够在一个界面中获取与特定主题或关键词相关的全面信息。这种技术提供了一种综合且多元化的搜索体验，让用户能够便捷地探索和访问丰富多样的搜索结果。

自从 2000 年开始商业化运营以来，韩国的搜索引擎 Naver 凭借其出色的聚合搜索功能在韩国的搜索领域迅速崭露头角，到 2011 年 6 月份，它的市场份额已经达到了 77%。尽管像 Google、Bing 和 Baidu 这样的搜索行业巨头也开始采用相似的技术手段，但每家公司在实施中都有着各自的特定方式。例如，Naver 选择将垂直搜索结果分开展示，而其他搜索引擎则偏向于将这些信息混合展示。这两种展示方式都满足了用户对信息整合和个性化展示的需求。

聚合搜索技术因其在信息检索领域的卓越表现而受到广泛关注。随着搜索技术的持续进步，预计聚合搜索技术将进一步发展，以提供更精准、便捷和个性化的用户体验。

1.3 研究内容

本文的研究致力于优化现有的无监督关键词提取算法，并在此基础上，搭建一个能够为用户提供全面而精准搜索体验的聚合搜索系统。研究工作主要围绕以下几个方面展开：

第一，对无监督关键词提取算法进行改进。使用组合策略，我们为多个关键词提取算法分配算法权重，以反映其在预测结果中的贡献。接着，利用各算法的相关性排序能力，为预测结果分配相关性权重。通过综合算法权重和相关性权重，我们对预测结果进行评分，最终选取得分最高的 TopN 结果作为优化后的关键词输出。

第二，研究了数据抓取与存储策略。利用网络爬虫工具 Scrapy，实现了对多个主流门户网站信息的高效抓取。并且设计了关系型数据库，用于存储各种来源，多种类型的数据。

第三，研究了 Elasticsearch 数据建模和数据同步技术。根据关系型数据库表中的字段值，构建 Elasticsearch 的基本索引结构，并设计了一套数据同步策略，

以实现数据从关系型数据库到 Elasticsearch 的平滑迁移。同时，借助 Elasticsearch 提供的可视化工具，实现了数据可视化的功能。

第四，设计与实现了基于无监督关键词提取算法的聚合搜索系统。设计了系统的界面和业务逻辑，深入研究并构建了一个高效的聚合搜索系统。通过对前期工作的整合与验证，完成了聚合搜索系统的设计和实现，为后续研究提供了坚实的基础。

1.4 论文的组织结构

论文的整体结构拆分成以下五个部分，全文核心内容按照整个系统搭建和项目的推进顺序完成论述：

第一章：绪论。本章主要介绍了课题研究的背景及意义，对核心研究内容的国内外研究情况进行简要的概括，较为详细地概述了论文的研究内容以及整篇论文的结构和章节的安排。

第二章：相关理论技术简介。本章主要介绍了与本研究紧密相关的技术概念，包括关键词提取算法、组合策略、搜索引擎的基本原理，以及全文检索技术和前后端开发技术。

第三章：一种无监督的指数权重衰减关键词提取方法。本章着重介绍了一种利用组合策略来改进无监督关键词提取算法的方法，并通过实验数据的对比验证了该算法的优化效果。

第四章：基于无监督关键词提取算法的聚合搜索系统。本章首先介绍了数据抓取的实现方法，包括使用爬虫工具进行数据采集以及执行数据清洗流程。接着，阐述了 MySQL 数据库的设计和数据存储策略。随后，本章展示了如何利用 EdecayRank 方法进行关键词抽取，并将关键词与源数据整合后存储到数据库中。此外，本章还将讨论 Elasticsearch 数据建模和数据同步策略。最终使用 Java 编程语言实现聚合搜索系统的开发工作。

第五章：总结与展望。对本文工作和项目进行总结，并对未来聚合搜索系统的发展潜力和研究方向提出展望。

2 相关理论技术简介

2.1 关键词提取技术

关键词提取技术（Keyword Extraction）是自然语言处理（NLP）领域的关键任务之一，旨在自动识别并提取文本中最能反映其主题或核心内容的词汇或短语[2]。在实际应用中，无监督关键词提取技术因其无需依赖大量标注数据的优势而得到了广泛使用。Papagiannopoulou 和 Tsoumakas（2020）对无监督关键词提取方法进行了分类，将其划分为基于统计、基于图和基于嵌入的三种主要方法[3]。

基于统计的方法，通过综合考量候选关键词的一系列统计特征，如词频（Term Frequency）、文档频率（Document Frequency）、词偏移量（Term Displacement）以及 n-gram 的出现频率，对关键词进行排序。YAKE[4]、EQPM[5]和 CQMine[6]等模型，不仅探索了传统的位置和频率特征，还引入了能够捕捉上下文信息的新统计特征，以增强关键词提取的准确性。

基于图的方法构建了词语之间的关联网络，并通过图算法进行关键词提取。TextRank[7]通过创建词语共现的图结构的方式，并且使用 PageRank 算法为各个顶点分配权重，从而有效地识别文本中的关键词。虽然 TextRank 算法简单有效，但是它对输入文本的质量有一定要求，并且大多基于词汇之间的共现关联，而这可能使得文本内部复杂的语义关联难以捕获。为了解决这些问题，后续的研究在 TextRank 上增加新的特征来进行改进，例如 SingleRank[8]、PositionRank[9]，以及致力于增强关键短语多样性的 TopicRank[10]和 MultipartiteRank[11]。

基于嵌入的方法则利用预训练语言模型（例如 ELMo[12]、Bert[13]、XLNet[14]）来引入丰富的外部知识和特征，从而提升关键词提取的性能。PatternRank[15]结合了预训练的语言模型 Sbert 和词性标注（Part-of-Speech Tagging, POS）技术，以无监督方式提取关键词。Key2vec[16]采用 Fasttext 构建短语和文档的嵌入表示，然后应用 PageRank 算法从候选关键短语中筛选出关键短语。EmbedRank[17]通过计算短语和文档嵌入之间的相似性来进行排名。SIFRank[18]

在 EmbedRank 的基础上，结合了预训练语言模型 ELMo 和句子嵌入模型 SIF[19]，以改进静态嵌入的局限性。AttentionRank[20]采用预训练语言模型来计算自我注意和交叉注意，以此确定文档中候选短语的重要性和语义相关性。PromptRank 使用 prompt 技术处理文档与候选关键词长度差异,通过将文档和候选关键词映射到共享潜在空间并进行概率排名来识别关键词。

在本文中，我们使用了四种关键词提取算法：PromptRank、ChatGLM、TextRank 和 PatternRank。接下来将详细介绍每种算法的工作原理。

2.1.1 PromptRank 算法

PromptRank[1]是一种无监督的关键词提取算法，它利用了预训练语言模型（PLM）和编码器—解码器架构。该算法的核心机制在于通过编码器－解码器结构将原始文档和 prompt 扩展的候选关键词映射到共享潜在空间，并利用解码器对候选关键词进行概率排名。根据 PromptRank 算法做关键词提取的步骤如下所示：

（1）通过正则表达式从文档中提取名词短语，作为候选关键词。正则表达式的构造遵循特定的语法规则，以确保提取的短语符合特定词性序列的要求。正则表达式的形式如 2-1 所示。

$$(\{NOUN*|ADJ\}*\{NOUN*\}) \quad (2\text{-}1)$$

（2）将文档输入编码器。对于每个候选关键词，算法设计了一个特定的 Prompt 来将其进行扩展，然后通过解码器计算生成该关键词的条件概率。这一概率反映了候选关键词与文档内容的匹配程度，概率越高，表明关键词的重要性越大。

（3）算法根据候选关键词在文档中的位置信息，计算位置惩罚因子。位置越靠前的关键词，其位置惩罚越小，从而在最终得分中获得更高的权重。

（4）结合候选关键词的生成概率和位置惩罚得分，对所有候选关键词进行综合排序。得分最高的前 K 个关键词被选为最终的关键词提取结果。

PromptRank 算法的这一设计充分利用了预训练语言模型的强大能力，不仅提高了关键词提取的准确性，而且通过考虑关键词在文档中的相对位置，增强了对关键词重要性的捕捉。

2.1.2 ChatGLM 算法

ChatGLM-6B[21]是一个开源的双语对话语言模型，支持中英文对话。该模型以 General Language Model (GLM)为核心，遵循传统的 GPT 风格，采用解码器自回归语言建模架构。GLM 基于 Transformer 架构，其训练目标是通过自回归填充空白（autoregressive blank infilling）来实现文本序列的生成。

具体而言，该模型针对给定的文本序列 $x=[x_1, \cdots, x_n]$，会从中选取一个子序列 $span\{s_1, \cdots, s_m\}$，这个子序列由连续的标记 s_i 组成，并将这些选定的子序列统一替换为一个特殊的掩码符号。模型的任务是自回归地恢复这些被掩码的部分，与 GPT 系列模型不同的是，GLM 在未被掩码的位置上采用双向注意力机制，并结合了两种不同的掩码类型：[MASK] 用于表示短空白，其长度被添加到输入序列的一部分；[gMASK] 则用来表示具有随机长度的长空白，这些空白被附加在提供前缀上下文的句子末尾。

ChatGLM-6B 集成了自回归模型、自编码模型以及编码器—解码器模型等多种预训练技术，针对中文对话场景进行了深度优化，以生成更自然的人类风格回答，提升了模型在特定任务的性能，并增强了其多语言对话的适应性。

2.1.3 TextRank 算法

TextRank 算法[7]是一种基于词图模型的关键词提取方法，其核心原理在于构建一个词图，其中每个词语或句子被视为图中的一个节点。通过迭代计算节点间的权重，该算法逐步收敛至稳定的权重分布。最终，根据权重值的降序排列，TextRank 选取排名最高的前 N 个节点作为关键词。

TextRank 算法扩展了 PageRank 的理论框架和应用范围。PageRank 算法通过评估网页间的链接关系来确定网页的重要性，构建了一个以网页为节点、链接为加权有向边的网络模型。该算法通过模拟投票过程为网页赋予权重，其核心计算公式如 2-2 所示。

$$PR(i) = \frac{1-d}{N} + d\sum_{j \in ln(i)} \frac{PR(j)}{|Out(j)|} \quad (2\text{-}2)$$

其中，*PR(i)* 代表网页 i 的访问概率，*ln(i)*是指向网页 i 的网页集合，*Out(j)* 是网页 j 发出的链接数量。PageRank 可能在特定情况下（如遇到环形或孤立节点）

无法稳定收敛，为此算法引入了一个随机概率因子 d（通常取 0.85）来解决这一问题。

TextRank 算法借鉴了 PageRank 的原理，通过以下两个原则来评估单词的重要性：

（1）频繁出现在其他单词附近的单词重要性更高。

（2）一个具有高 TextRank 值的单词也能提升其后继单词的 TextRank 值。

该算法将句子视作节点，并通过计算节点间的权重（基于句子间的相似度）来构建加权无向图。在具体实施时，TextRank 算法使用一个长度为 N 的滑动窗口来确定相邻的单词节点。

TextRank 算法的计算如式 2-3 所示。其中 $G(V,E)$ 表示点集为 V，边集为 E 的有向图，$ln(v_i)$ 表示指向节点 v_i 的点集，$Out(v_i)$ 表示从节点 v_i 指出的点集。

$$S(v_i) = (1-d) + \sum_{j \in ln(v_j)} \frac{W_{ji}}{\sum_{v_k \in Out(v_j)} W_{jk}} \quad (2\text{-}3)$$

根据 TextRank 算法做关键词提取的步骤如下所示：

（1）将文本分割成句子，获得句子集合；

（2）经停用词过滤和词性标注后，筛选出候选关键词；

（3）以候选关键词为节点，根据共现关系构建节点间的带权有向边，并计算权重；

（4）迭代更新节点权重，直至收敛，最终选取权重最高的前 N 个词作为关键词。

2.1.4 PatternRank 算法

PatternRank 算法[15]是一种无监督的关键词提取方法，它利用预训练语言模型和词性来提取关键词。根据该算法进行关键词提取的步骤如下：

（1）将输入的文本进行词汇切分，并为每个词汇标注词性。然后，选择一个复杂的词性模式，筛选出符合该模式的词汇作为候选关键词。词性模型公式如 2-4 所示。

$$\left(\left(\{.*\}\{HYPH\}\{.*\}\right)\{NOUN\}*\right)\left|\left(\left(\{VBG\}|\{VBN\}\right)?\{ADJ\}*\{NOUN\}+\right)\right. \quad (2\text{-}4)$$

（2）利用预训练语言模型对候选关键词进行排序。通过将整个文本以及所有候选关键词转化为语义向量表示，计算文本表示与候选关键词表示之间的余弦相似度，根据相似度得分对候选关键词进行降序排序。

（3）从排序后的候选关键词中提取排名靠前的关键词作为最终的关键词，这些关键词被认为是最能代表输入文本的关键词。

由于预训练语言模型可以捕捉文本的语义信息，而词性信息可以帮助筛选出更具有代表性的候选关键词。PatternRank 模型通过结合预训练语言模型和词性信息，在关键词提取任务中取得了较好的性能。

2.2 组合策略

组合策略是集成学习（Ensemble Learning）中的一个关键组成部分，它旨在整合多个基本学习算法的预测结果，进而提高模型的预测精度和泛化能力[22]。

集成学习主要分为依赖和独立两大类方法。依赖性方法，例如 Boosting 和 Dagging，通过将前一个学习器的输出反馈给后续学习器，从而实现知识的传递。而独立性方法，例如 Bagging 和 Random Subspace，则并行构建基学习器，并采用多数投票等策略合成最终输出。这些方法各有特点，共同构成了集成学习中的一个完整的技术体系[23]。

在本文中，我们选择了投票方法作为组合策略。这种方法基于集体智慧的原则，即通过多个独立分类器的协同决策，得出更精确且更稳定的预测结果。投票方法的核心在于运用多数投票或加权投票机制，综合多个分类器的决策，以此来确定最终的预测类别。常见的投票方法包括：

1. 多数投票（Majority Voting）：在多数投票中，每个基分类器对样本进行独立分类，其预测结果被视为一票。集成模型的最终预测是获得最多票数的类别。

2. 加权多数投票（Weighted Majority Voting）：在加权多数投票中，每个基本分类器的预测结果都被赋予一个权重，这些权重可以根据基本分类器的性能或其他因素进行分配。在投票过程中，根据权重对不同基本分类器的预测结果进行加权，然后选择加权得分最高的类别作为集成预测结果。

3. 朴素贝叶斯组合规则（Naïve Bayes Combination Rule）：朴素贝叶斯组合规则基于贝叶斯定理，将基本分类器的预测结果与先验概率结合起来，计算后验

概率，并选择具有最高后验概率的类别作为集成预测结果。

4. 行为知识空间方法（Behavioral Knowledge Space Method）：行为知识空间方法通过将基本分类器的预测结果映射到一个行为知识空间中，利用空间中的几何关系来进行决策。该方法可以处理基本分类器之间的冲突和不确定性。

5. 概率近似（Probabilistic Approximation）：概率近似是一种将基础分类器的预测转化为概率估算的方式，接着把各个概率值相互组合，以得到一个最终的集成概率分布结果。

尽管传统集成学习主要集中在监督学习任务上，但其通过模型组合提升性能的核心思想也适合应用在无监督学习领域中。在本文中，我们引入集成学习的概念，并通过加权多数投票方法的组合策略，构建了一个集成模型。该模型综合考虑多个关键词提取算法的预测结果，以获得更好的关键词提取效果。

2.3 全文检索技术

Elasticsearch 是一个开源的实时分布式搜索和分析引擎，其构建基于 Apache Lucene 库。该引擎专为提供高效、可扩展且用户友好的全文检索功能而设计，能够处理大规模数据集。Elasticsearch 的核心概念包括文档、索引和分片。文档作为数据的基本单位，通常以 JSON 格式编码，能够包含结构化和非结构化数据的多种类型。索引是文档的集合，根据其特征进行组织，旨在加速数据检索。分片则是索引的逻辑分割，它允许数据在集群的不同节点上分布式存储和处理。

Elasticsearch 的核心技术之一是倒排索引，它将文档中的词汇转换为指向文档列表的数据结构，从而实现快速定位包含特定关键词的文档[24]。与传统数据库的正向索引不同，倒排索引通过对所有文档进行分词，然后根据查询关键词反向检索文档位置。Elasticsearch 的倒排索引由关键词和倒排列表组成，其中关键词存储在词典中，每个关键词都关联一个倒排索引，用于快速检索。由于词典规模庞大，倒排索引高效记录了关键词的文档映射、频次和在文档中的定位信息。

Elasticsearch 不仅支持全文检索，还具备强大的分析和聚合功能。它支持复杂的查询语法和过滤条件，能够执行基于文本内容的高级查询和聚合操作。此外，Elasticsearch 支持实时数据分析和可视化，通过配套的 Kibana 工具，用户可以实现数据的可视化展示和交互式分析。

Elasticsearch 通过先进的算法来评估查询词与文档之间的相关性。在 5.0 版之前，Elasticsearch 主要使用 TF-IDF 算法，该算法基于词频（TF）和逆文档频率（IDF）的加权和来评估相关度。用户发起搜索后，Elasticsearch 计算查询词与文档集的相关性得分，并按分数排序展示结果。从 5.0 版开始，Elasticsearch 默认采用 BM25 算法，这是一个考虑查询词频和文档长度的概率评分模型，用于更精确地确定文档的相关性。

2.4 前后端框架

在 Web 开发领域，SSM 框架（Spring + SpringMVC + MyBatis）的应用显著提高了开发效率。Spring 框架通过其工厂模式，实现了 Java 对象生命周期的自动化管理，包括对象的创建和依赖注入，从而简化了开发流程并降低了耦合度。Spring 还支持面向切面编程（AOP），使得权限控制和运行监控更加便捷。此外，Spring 的容器管理机制简化了对其他优秀框架和 API 的集成，同时其提供的单元测试框架增强了系统测试的可扩展性。

SpringBoot 的出现极大地简化了配置过程。与传统 Spring 框架相比，SpringBoot 遵循“约定优于配置”的原则，提供了默认配置，从而减少了开发人员在细节配置上的负担。并且，SpringBoot 还内嵌了 Tomcat 等常用 Web 服务器，无需额外安装和配置外部服务器，极大的降低了集成的复杂性。

在前端技术方面，Vue.js 框架因其简洁性和灵活性而受到广泛欢迎。该框架提供了简单灵活的 API 以及响应式数据绑定，简化了交互式界面的开发工作。并且 Vue.js 采用组件化开发模式，允许开发者将应用拆分为多个独立且可复用的组件，提高了代码的可维护性和可复用性。

2.5 搜索引擎技术

搜索引擎通过爬虫技术自动收集互联网上的网页数据，之后利用特定算法对这些数据进行处理，建立起索引，并根据用户的查询需求，通过检索系统返回并展示相关结果。

2.5.1 架构

搜索引擎的架构主要由四个关键组件构成：爬虫（搜索器）、索引器、检索器和用户接口等四个部分。

1. 爬虫：作为搜索引擎的数据采集模块，爬虫程序自动遍历互联网，抓取网页内容并记录其 URL。在规定的 IP 范围内，爬虫持续执行数据抓取任务，为搜索引擎的数据库提供原始数据支撑。

2. 索引器：索引器主要用于对爬虫获取的网页数据的解析与结构化处理，从中提取出关键信息，并归纳网页的实质内容。处理后的数据被存储于搜索引擎数据库中，为快速响应用户查询提供可能。

3. 检索器：检索器负责解析并处理用户提交的查询请求，并根据数据库中的索引数据进行匹配操作。检索器采用特定的排序算法对匹配结果进行排序，并将有序的结果集展示给用户。

4. 用户接口：作为用户与搜索引擎交互的平台，用户接口通常以 Web 界面的方式进行展示。用户需要在这里输入查询关键词，随后搜索引擎对这些输入进行相应的处理并返回已经排序好的结果，用户便能从中选取并查阅相关信息。

2.5.2 搜索引擎

传统的搜索引擎可以根据其功能和应用领域划分为多种类型，其中包括目录式搜索引擎、垂直搜索引擎、全文搜索引擎以及通用搜索引擎等。本文采用聚合搜索引擎技术，通过整合多个垂直搜索引擎的输出结果，目的是为了解决信息碎片化、信息孤岛和信息过载等问题，进而提升用户的搜索效率。

垂直搜索引擎专注于从特定领域中筛选出数据，并提供针对性的信息检索服务[34]。垂直搜索引擎虽然覆盖的信息量较小，但与通用搜索引擎相比，其所提供的信息与特定领域紧密相关，并且更新速度快，能够确保搜索结果的准确性、具体性和深度。垂直搜索引擎是对传统搜索引擎的进一步细化和延伸，当涉及在特定领域提供的信息查找和匹配服务时，它通常展现出比传统搜索引擎更为出色的表现。

聚合搜索引擎的工作原理涉及同时向多个独立的搜索引擎或数据库发起查询，并整合和排序各个来源的数据，最终以统一的界面展示给用户。为了确保搜

索结果的完整性与精确性，这种整合过程可以采用多种算法和策略。聚合搜索引擎的主要优势在于能够提供更为全面的搜索结果，降低了用户在多种搜索引擎之间更换的频率，进一步节约了用户的思考空间和精力。此外，聚合搜索引擎还能够根据用户的偏好和搜索历史，对搜索结果进行个性化排序和展示，以便给予用户更为个性化的搜索体验。在实际应用中，这种搜索引擎的设计和实现对于提高用户的满意度以及搜索的效率都是极其关键的。

2.6 本章小结

本章对系统的相关技术进行了概括和综述。首先介绍了相关的关键词提取算法以及搜索引擎的基本原理。然后探讨了前后端框架技术，这些技术为构建高效、可扩展的 Web 应用提供了基础。此外，作为搜索引擎的核心技术之一，全文检索技术在本章中也得到了细致地分析，以确保其在实际应用中的有效性和效率。

3 一种无监督的指数权重衰减关键词提取方法

3.1 EdecayRank 方法

在信息过载的背景下，文本数据的高效处理和关键信息的精准提取对于知识发现和决策制定具有重要意义。为此，本文提出了一种名为 EdecayRank 的无监督指数权重衰减关键词提取方法，旨在提升自然语言处理领域关键词提取的效率和准确性。

该方法通过整合四种先进的关键词提取算法 PromptRank[1]、TextRank[7]、ChatGLM2-6B[21]和 PatternRank[15]的优势，克服了单一模型的局限性，增强了对文本核心概念和关键特征的识别精度，从而提升了关键词提取的整体性能和泛化能力。此外，EdecayRank 方法还利用关键词提取算法的排序能力，为每个关键词分配相关性权重，以体现关键词在文本中的相关性。

在研究过程中，我们遵循了一种策略性的研究思路，即通过算法集成和权重调整来提升关键词提取的准确性。在算法集成方面，我们根据每个算法在关键词提取任务中的表现来分配相应的算法权重，以充分利用各算法的优势。在权重调整方面，我们引入了指数衰减（Exponential Decay）模型来量化和标识关键词的相关性。该数学模型通过给予关键词列表中首要关键词更高的权重，并随着列表中关键词位置的后移逐渐降低其权重，实现了对关键词相关性的区分。

EdecayRank 方法通过这种创新的算法集成和权重调整策略，为无监督关键词提取领域带来新的视角和改进。

3.1.1 总体框架

本文提出 EdecayRank 模型架构，如图 3-1 所示，该架构由三个关键组成部分构成。

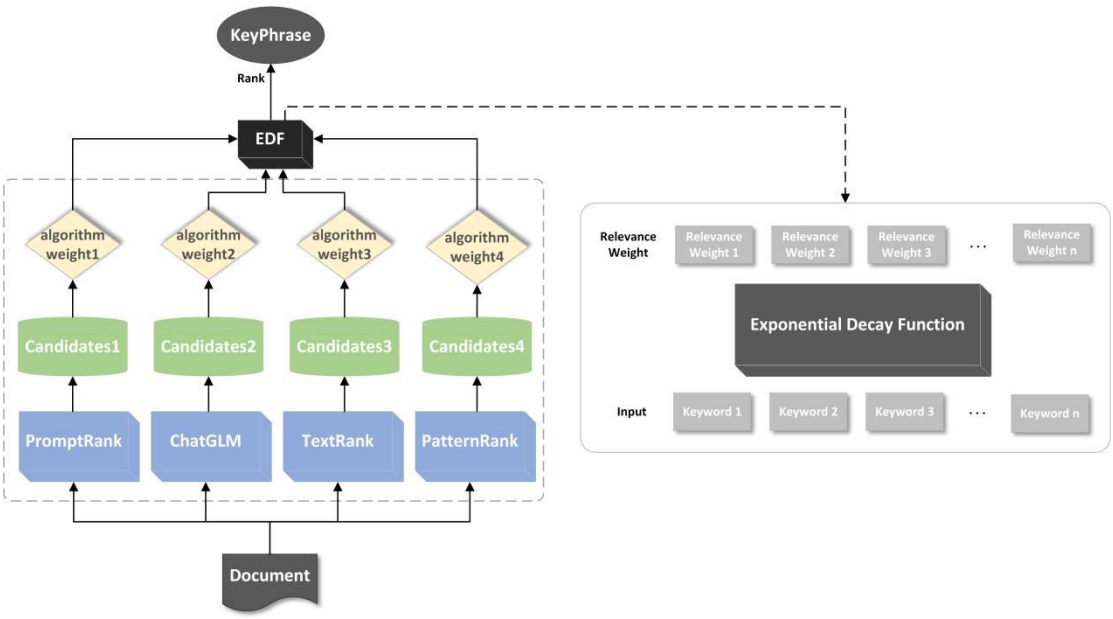


图 3-1 EdecayRank 方法模型架构图

在模型组件部分，我们精心挑选了四种基于不同算法的关键词提取模型：PromptRank、ChatGLM、TextRank 和 PatternRank。这些模型的整合遵循了多数投票机制的原则，即通过融合各个模型的预测结果，优化整体的预测精度和鲁棒性。鉴于每个模型在捕捉文本关键信息方面的独到之处，EdecayRank 能够从多个角度进行分析，从而提高关键词提取的精确性，降低模型偏差，增强系统的鲁棒性。此外，它还扩展了关键词选取的多样性，进一步提升了关键词提取的整体效果和质量。

在算法权重部分，不同模型在关键词提取任务中展现出不同效果。为了平衡每个模型在最终结果中的贡献，我们给每个模型分配了相应的算法权重。这些权重的分配考虑了模型的性能差异、可靠性、领域适应性以及可能的人工干预等因素，以确保获得更为精确、可靠的关键词提取结果。

在相关性权重部分，我们引入了指数衰减函数来量化关键词在文本中的相关性。这一策略的目的在于为那些位于列表前端、相关性较高的关键词提供更高的权重，同时逐步降低列表后部关键词所带来的影响。从而有效过滤噪声，强调核心概念，并提供个性化的权重调整，进一步提升关键词提取的准确性。

综合这三个部分，EdecayRank 方法可以充分利用多个模型的优势，显著提升关键词提取的准确性和鲁棒性，实现更为精确和全面的关键词提取结果。

3.1.2 模型输入

在研究模型输入时，本文选择将文档作为数据基础。并且为了保证文本数据的高质量，在提取关键字之前，我们进行了一系列的预处理，这些预处理包括文本清洗、分词处理以及停用词移除等多个步骤。

文本清洗的目标是去除文本中的噪声和非关键信息，剔除诸如特殊字符、数字等非关键信息，以便减少这些部分可能对模型性能的影响。通过分词操作，我们能够对连续的文字进行分解，将其划分为富有意义的词汇或短句。停用词移除涉及去除那些在语义上相对较少的常用词条，这不仅有助于减轻模型在计算上的复杂性，也有助于有效降低数据中的噪声水平。

这些预处理措施的目标是优化文本数据的处理流程，提升后续关键词提取任务的效率与准确性，为模型提供更加精炼和高质量的输入数据。

3.1.3 关键词提取

EdecayRank 方法在关键词提取中主要遵循以下四个步骤：

（1）首先，对于文本样本 d ，将其输入到四个不同的关键词提取算法中。这些算法独立运行，分别生成各自的关键词候选集，记为 $C_{PromptRank} = \{c_1, c_2, ..., c_n\}$，$C_{ChatGLM} = \{c_1, c_2, ..., c_n\}$，$C_{TextRank} = \{c_1, c_2, ..., c_n\}$，$C_{PatternRank} = \{c_1, c_2, ..., c_n\}$.

（2）随后，根据各个算法在关键词提取任务中的表现，为每个算法分配一个算法权重 aw。这些算法权重 aw 代表了各个算法对最终关键词提取结果的贡献度，确保了算法性能与权重之间的正相关性。

（3）然后，在算法权重的基础上，引入关键词的相关性权重 rw，并通过指数衰减函数量化关键词的相关性。其目的在于衡量关键词在文本中的重要性，其中指数衰减函数能够根据关键词在列表中的相对位置，为其分配一个相关性权重。

（4）最后，在综合考虑关键词的频率、算法权重 aw 以及相关性权重 rw 的基础上，计算出每个关键词的综合得分 *Score*。根据这些得分对关键词进行排序，确定最终的关键词提取结果。

为了便于读者理解本文中使用的符号和术语，本文提供了一个符号表 3-1。该符号表详细列出了本文中出现的所有符号及其定义，确保了论文的专业性和易读性。

表 3-1 符号表

符号	定义
$C_{PromptRank}$	PromptRank 模型提取的关键词列表
$C_{ChatGLM}$	ChatGLM 模型提取的关键词列表
$C_{TextRank}$	TextRank 模型提取的关键词列表
$C_{PatternRank}$	PatternRank 模型提取的关键词列表
c	单个关键词
aw	根据模型的性能分配的算法权重
rw	根据关键词的相关性特征分配的相关性权重
$Score$	根据关键词频率、aw、rw 计算所得分数
d	文本

在文中介绍了我们采用的四种不同关键词提取算法，每种算法都基于独特的方法论。

PromptRank 模型在处理文本时，首先执行分词和词性标注，然后利用正则表达式 3-1 来识别名词短语。

$$(\{NOUN*|ADJ\}*\{NOUN*\}) \quad (3\text{-}1)$$

为了解决文档和候选关键词长度的差异，PromptRank 使用设计好的 prompt 扩展候选关键词的长度。随后，该模型将文档和候选关键词短语映射到潜在空间，并利用解码器计算候选关键词的概率值，通过排序得到最终的关键词列表。

PatternRank 模型同样从分词和词性标注开始，但它采用一个复杂的词性模式 3-2 来提取候选关键词短语。为了获得语义上的表示，PatternRank 利用预训练的 Sbert 模型将文档和候选关键词短语转换为语义向量。最终，通过计算候选关键词与文档的余弦相似度，对这些关键词进行排序，从而得到关键词列表。

$$\big((\{.*\}\{HYPH\}\{.*\})\{NOUN\}*\big)|\big((\{VBG\}|\{VBN\})?\{ADJ\}*\{NOUN\}+\big) \quad (3\text{-}2)$$

TextRank 算法采用了一种基于图的方法，通过将文本中的句子或词语视为图中的节点，构建出一个无向图。该算法通过计算节点间的权重来评估每个节点的重要性，并通过迭代过程不断更新这些权重。最终，TextRank 选取权重较高的节点作为关键词输出。

ChatGLM 模型作为一种对话生成模型，能够通过对话交互的方式接收文档文本，并根据相关性对关键词进行排序。然而，在实际应用中，ChatGLM 偶尔会产生不符合预期的输出。为了确保关键词提取的准确性，我们在后续研究中对这些异常情况进行识别分析，并制定相应的处理策略。

3.1.4 权重分配

1 算法权重

算法权重的分配基于各关键词提取算法在关键词提取任务中的表现。权重设定的目的是确保集成模型能够充分利用每个算法的独特优势，以提高整体关键词提取的准确性和鲁棒性。

PromptRank 模型因其在关键词提取任务中达到了最先进（State-of-the-Art, SOTA）的性能水平，能够以较高的准确度识别候选关键词，因此被赋予了较高的权重。这一策略确保了模型在提取过程中能够优先考虑 PromptRank 识别出的关键词。

ChatGLM 模型能够捕捉到其他模型可能忽略的关键词，为了增强关键词列表的多样性，我们为 ChatGLM 分配了较高的权重。这种权重分配策略有助于在集成模型中平衡不同算法的输出，即使某个模型在特定情况下表现欠佳，通过其他模型的补充也能保证关键词提取的完整性。

TextRank 和 PatternRank 模型在关键词提取任务上也表现出了良好的性能，因此我们也为它们分配了适当的权重，以确保它们在集成过程发挥作用。

通过这种算法集成方法，我们得以将各个模型的优势整合起来，构建一个精确度、鲁棒性强的集成模型。

在算法权重分配上，我们根据各个算法模型在关键词提取任务中的表现，分别为它们赋予了不同的算法权重 aw。这些权重反映了每个模型在集成模型中的重要程度，是决定模型输出结果的关键因素。通过这种方法，我们确保了模型在处理不同文本时，能够综合利用各算法的优势，以达到更好的关键词提取效果。

$$aw = \left[aw_{PromptRank}, aw_{ChatGLM}, aw_{TextRank}, aw_{PatternRank}\right]$$

我们定义了各个算法模型提取的关键词集合，统一表示为 C。

$$C = \left\{C_{PromptRank}, C_{ChatGLM}, C_{TextRank}, C_{PatternRank}\right\}$$

为了量化和评估这些关键词的相关性和重要性，我们设计了一套评分机制，即关键词得分（*Score*）公式 3-3 ：

$$Score(c_i) = \sum_{M\in\{M_1,M_2,M_3,M_4\}} (aw(M) \times rw(c_i, M)) \qquad (3\text{-}3)$$

其中，M 表示参与集成的模型，c_i 代表集合 C 中的第 i 个关键词，$aw(M)$ 是分配给模型 M 的权重，反映该模型在集成中的贡献程度，M_1, M_2, M_3, M_4 是所有模型的集合即 $\{PromptRank, ChatGLM, TextRank, PatternRank\}$。

2 基于指数衰减函数的相关性权重

为了量化关键词的相关性信息，我们引入了指数衰减函数 3-4 来对关键词提取算法输出的关键词列表进行加权调整。该函数旨在度量关键词在文本中的重要性，使得关键词列表中排序靠前的关键词被赋予较高的权重，而排序靠后的关键词则获得较低的权重。这种权重调整策略基于一个普遍观察到的现象：在关键词提取过程中，往往排名靠前的关键词与文本的主题和内容更为紧密相关。

Zhang M 等人的研究揭示了衰减因子在 TextRank 算法中的关键作用，特别是在权重传递控制和算法收敛性方面[35]。实验结果表明，将衰减因子设定为 0.9 时，能够实现最优的关键词提取效果。这一发现不仅证实了衰减函数在提升关键词相关性和准确性方面的有效性，而且为本研究在设计和调整关键词提取模型时提供了重要的参考依据。因此，我们采用衰减函数来调整关键词权重，以期实现更为精确的文本分析和信息检索。

$$rw_{(c_i, C_M)} = a \times e^{(-b \times rank(c_i, C_M))} \tag{3-4}$$

其中，c_i 代表集合 C 中的第 i 个关键词，C 表示各个算法模型提取的关键词集合，C_M 表示当前模型输出的关键词列表。常数 a 决定了函数的初始值或幅度，常数 b 决定了衰减的速率。$rank(c_i, C_M)$ 表示关键词 c_i 在模型 M 输出的关键词列表的排名位置。通过调整参数 a 和 b，我们可以精细地控制权重的分布，以确保关键词的得分（$Score$）能够准确地反映其在文本中的重要性。

在加入相关性权重后，更新得到关键词得分的计算公式 3-5 ：

$$Score(c_i) = \sum_{M \in \{M_1, M_2, \ldots, M_4\}} (aw(M) \times rw(c_i, C_M)) \tag{3-5}$$

通过这种综合考虑算法权重、相关性权重的得分计算方法，我们能够更准确地捕捉和评估关键词的相关性，从而提高关键词提取的整体性能。

3.1.5 模型输出

我们首先将目标文档输入至 PromptRank、ChatGLM、TextRank 和 PatternRank 四种模型中。这些模型独立运行，各自生成并输出包含 20 个基于其算法预测的候选关键词的列表。

随后，我们为每个模型分配了相应的算法权重（aw）以及为列表中的每个关键词分配了相关性权重（rw）。算法权重体现了各模型在关键词提取任务中的相对贡献，而关键词的相关性权重则反映了关键词在文本中的重要性。

在完成权重分配后，我们计算了每个候选关键词的综合得分（*Score*），该得分综合考虑了算法权重和关键词的相关性权重。基于这些得分，我们对所有候选关键词进行了重新排序，并且选择了得分最高的前 15 个关键词作为最终的关键词提取结果。

3.1.6 异常结果处理

在实际的应用场景下，ChatGLM 模型因模型的复杂性、训练数据的限制性和特定知识的缺乏等原因，可能导致偶尔的输出结果与预先的期望相悖。为了减少不确定性因素对关键词提取整体效果的负面效果，我们制定了一套策略以衡量和优化 ChatGLM 的输出效果。

具体策略如下：首先，在输出完全满足我们预期的情况下，我们不进行任何附加操作，而是直接接受其产出的结果。其次，当 ChatGLM 的输出中有超过一半的关键词满足要求时，系统会执行过滤过程，仅选择那些已经满足要求的关键词保留。最后，如果 ChatGLM 的输出中有大量与要求不符的关键词，我们将判定这一模型在当前任务下的表现并不理想，并在接下来的计算中将其排除。

这种分层次的处理策略的目的是确保关键词提取结果的准确性和可靠性，同时最大限度地利用 ChatGLM 模型的有效输出。

3.2 实验

在本节中，我们通过一系列实验来验证我们提出方法的有效性。这些实验在六个广泛使用的数据集上对基准模型和 EdecayRank 进行了比较评估。此外，为了深入理解相关性权重在提升关键词提取性能中的作用，我们还进行了消融实验。

3.2.1 数据集和评价指标

为了确保评估的全面性和准确性，我们在 6 个广泛使用的数据集上对我们的模型进行了性能评估。这些数据集包括 Inspec[30]、SemEval-2010[36]、SemEval-2017[31]、DUC2001[37]、NUS[32]和 Krapivin[38]。表 3-2 总结了数据集的统计数据。在评估过程中，我们采用了 top5、top10、top15 候选关键词的 $F1$ 分数来衡量关键词提取的性能。在计算 $F1$ 分数时，我们去除了重复的关键词候选项，并采用了词干提取技术以确保评估的一致性和准确性。

表 3-2 6 个数据集的统计数据

DataSet	Type	#Doc.	Avg.#Words	Gold Keyphrase Distribution				
				1	2	3	4	≥5
Inspec	Scientific Paper Abstract	500	130.57	13.5	52.7	24.9	6.7	2.2
SemEval2017	Scientific Paper Abstract	493	176.13	25.7	34.4	17.5	8.8	13.6
SemEval2010	Full Scientific Paper	243	7434.52	20.5	53.6	18.9	4.9	2.1
DUC2001	News Document	308	724.63	17.3	61.3	17.8	2.5	1.1
NUS	Full Scientific Paper	211	7644.43	26.9	50.6	15.7	4.6	2.2
Krapivin	Full Scientific Paper	460	8420.76	17.8	62.2	16.4	2.9	0.7

表 3-2 给出了 6 个数据集的统计数据中，# Doc.是数据集中的文档数量。Avg.# Words 是所示数据集中文档的平均单词数。Gold Keyphrase Distribution 表示在每个数据集中具有不同长度的关键词的百分比。

为了评估关键词提取模型的性能，本文采用 $F1$ 作为评价指标。$F1$ 分数是一种广泛认可的综合性能度量，它融合了精确率（$Precision$）和召回率（$Recall$）两个关键指标，以提供一个平衡的模型性能评估。其中准确率（$Precision$）是指模型在所有样本中正确分类的比例。召回率（$Recall$）衡量了模型正确预测为正例的样本数占所有实际正例样本数的比例。$Precision$、$Recall$、$F1$ 的具体计算方法如式 3-6 至 3-8 所示。

$$Precision = \frac{TP}{TP + FP} \tag{3-6}$$

$$Recall = \frac{TP}{TP + FN} \tag{3-7}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3-8}$$

上述公式中 TP、FP、FN、TN 分别表示真正例、假正例、假反例和真反例。其中，TP 表示模型正确预测为正类的样本数，FP 表示模型错误预测为正类的

样本数，FN 表示模型错误预测为负类的样本数，而 TN 则表示模型正确预测为负类的样本数。$Precision$、$Recall$、$F1$ 值的取值范围为 0 到 1，它们的值越高，说明模型或分类器的性能越好。

3.2.2 基准模型

本文将所提出的模型与 9 个基准无监督关键词提取模型进行了比较，这些基准模型根据其核心方法论被划分为三个类别：

（1） 统计模型：YAKE！[4]，统计模型具有简单高效、自动学习和适用于多领域的特点，但存在对统计特征的依赖和对文本结构敏感的缺点；

（2） 基于图的模型：TextRank[7]、SingleRank[8]、TopicRank[10]、MultipartiteRank[11]，基于图的模型能够通过建立关键词间的相互关系图来捕获语义数据，但它高度重视文本的结构特性和意义，且对于大规模数据处理较为困难。

（3） 基于深度学习或混合模型：EmbedRank[17]、SIFRank[18]、MDERank[39]、PromptRank [40]，基于深度学习的模型能够通过学习文本的分布式表示来捕捉语义信息，但需要大量的训练数据和计算资源，并且对于模型的解释性较差。

3.2.3 实验设置

对于基准模型 PromptRank，我们采用了其默认配置模型 T5-base，同时沿用了超参数 α 和 γ 在各个数据集上已验证为最优的设置。对于 ChatGLM 模型、TextRank 模型以及 PatternRank 模型，我们遵循了它们的默认参数配置。

在综合考量了各模型在关键词提取任务中的性能表现后，我们为 PromptRank、ChatGLM2-6B、TextRank 以及 PatternRank 分别设置了算法权重 0.3，0.3，0.2 和 0.2。在特定情况下，即 ChatGLM 模型表现不佳时，我们会将算法权重调整为 0.4、0、0.3 和 0.3，以确保模型的鲁棒性。

在应用指数衰减函数来量化关键词的相关性时，我们将衰减函数的常数 a 设置为 1，衰减率 b 设置为 0.1，以确保权重的合理分配并反映关键词在文本中的相对重要性。为了简化结果并保持其可读性，我们限制了所有模型输出的关键词数量，仅选取各模型排名前 20 的关键词（n = 20）进行后续操作。

3.2.4 实验结果

为了验证方法的有效性，我们在 6 个标准数据集上对 EdecayRank 与基线模

型进行了实验对比，选取 top5、top10 和 top15 的 F1 分数作为关键词提取性能的评估指标。

表 3-3 各模型在 6 个数据集上的关键字提取性能 F1@K，K∈{5、10、15}

F1@K	Method	Dataset						AVG
		Inspec	SemEval2017	SemEval2010	DUC2001	NUS	Krapivin	
5	TextRank	21.58	16.43	7.42	11.02	1.80	6.04	10.72
	SingleRank	14.88	18.23	8.69	19.14	2.98	8.12	12.01
	TopicRank	12.20	17.10	9.93	19.97	4.54	8.94	12.11
	MultipartiteRank	13.41	17.39	10.13	21.70	6.17	9.29	13.02
	YAKE	8.02	11.84	6.82	11.99	7.85	8.09	9.10
	EmbedRank(Bert)	28.92	20.03	10.46	8.12	3.75	4.05	12.56
	SIFRank(ELMo)	29.38	22.38	11.16	24.30	3.01	1.62	15.31
	MDERank(Bert)	26.17	22.81	12.95	13.05	15.24	11.78	17.00
	PromptRank(T5)	31.73	**27.14**	**17.24**	27.39	17.24	16.11	22.81
	EdecayRank	**33.05**	26.02	16.84	**30.93**	**20.48**	**18.42**	**24.29**
10	TextRank	27.53	25.83	11.27	17.45	3.02	9.43	15.76
	SingleRank	21.50	27.73	12.94	23.86	4.51	10.53	16.85
	TopicRank	17.24	22.62	12.52	21.73	7.93	9.01	15.18
	MultipartiteRank	18.18	23.73	12.91	24.10	8.57	9.35	16.14
	YAKE	11.47	18.14	11.01	14.18	11.05	9.35	12.53
	EmbedRank(Bert)	38.55	31.01	16.35	11.62	6.34	6.60	18.41
	SIFRank(ELMo)	39.12	32.60	16.03	27.60	5.34	2.52	20.54
	MDERank(Bert)	33.81	32.51	17.07	17.31	18.33	12.93	21.99
	PromptRank(T5)	37.88	**37.76**	20.66	31.59	20.13	16.71	27.46
	EdecayRank	**40.19**	36.10	**21.06**	**34.47**	**23.32**	**19.42**	**29.09**
15	TextRank	27.62	30.50	13.47	18.84	3.53	9.95	17.32
	SingleRank	24.13	31.73	14.4	23.43	4.92	10.42	18.17
	TopicRank	19.33	24.87	12.26	20.97	9.37	8.30	15.85
	MultipartiteRank	20.52	26.87	13.24	23.62	10.82	9.16	17.37
	YAKE	13.65	20.55	12.55	14.28	13.09	9.12	13.87
	EmbedRank(Bert)	39.77	36.72	19.35	13.58	8.11	7.84	20.90
	SIFRank(ELMo)	39.82	37.25	18.42	27.96	5.86	3.00	22.05
	MDERank(Bert)	36.17	37.18	20.09	19.13	17.95	12.58	23.85
	PromptRank(T5)	38.17	**41.57**	21.35	31.01	20.12	16.02	28.04
	EdecayRank	**40.85**	39.54	**22.08**	**32.73**	**22.54**	**17.94**	**29.28**

表 3-3 展示了在6个数据集上，EdecayRank方法与基线模型在F1@5、F1@10和 F1@15 三个不同关键词数量阈值下的评估结果。结果表明，EdecayRank 方法在 F1@5 指标上的平均分数比当前最先进（State-of-the-Art, SOTA）的 PromptRank 方法高出了 1.48；在 F1@10 和 F1@15 指标上，相较于 PromptRank 方法，分别实现了 1.63 和 1.24 的提升。基于这些提升，EdecayRank 在实际应用中，即返回 top5、top10 和 top15 关键词的任务上，表现出了更为出色的性能，其 F1 分数分别提高了 6.49%、5.94%和 4.42%。

值得注意的是尽管在 SemEval-2017 数据集上，PromptRank 模型的表现略胜一筹，但 Edecay 方法在其余数据集上均展现出卓越的性能。这一现象不仅突显了 Edecay 方法在不同数据集上的泛化能力，同时也反映了该方法在处理多样化文本任务时的稳定性和可靠性。

3.2.5 消融实验

为了深入探究相关性权重在提升关键词提取性能中的作用，我们设计了一系列消融实验。实验结果如表 3-4 所示。

表 3-4 相关性权重的消融实验

F1@K	Method	Dataset						AVG
		Inspec	SemEval201	SemEval2010	DUC2001	NUS	Krapivin	
5	PromptRank(T5)	31.73	27.14	17.24	27.39	17.24	16.11	22.81
	$EdecayRank_{aw}$	30.68	23.94	14.65	28.04	18.77	17.24	22.22
	$EdecayRank_{aw+rw}$	33.05	26.02	16.84	30.93	20.48	18.42	24.29
10	PromptRank(T5)	37.88	37.76	20.66	31.59	20.13	16.71	27.46
	$EdecayRank_{aw}$	37.68	35.01	19.55	33.03	21.61	18.04	27.49
	$EdecayRank_{aw+rw}$	40.19	36.10	21.06	34.47	23.32	19.42	29.09
15	PromptRank(T5)	38.17	41.57	21.35	31.01	20.12	16.02	28.04
	$EdecayRank_{aw}$	38.31	39.46	20.95	33.04	21.37	16.90	28.34
	$EdecayRank_{aw+rw}$	40.85	39.54	22.08	32.73	22.54	17.94	29.28

实验数据显示，在 DUC2001、NUS、Krapivin 这三个数据集上，仅使用了算法权重的模型 $EdecayRank_{aw}$ 表现优于 PromptRank，但在 Inspec、SemEval2017、SemEval2010 数据集上则略逊一筹。然而，当引入相关性权重后，模型整体性能得到了显著提升。$EdecayRank_{aw+rw}$ 除了在 SemEval2017 数据集上的 F1 得分略低于 PromptRank 外，在其他数据集上的 F1 得分均显著超越 PromptRank，达到了最佳水平。

虽然基于算法权重的 $EdecayRank_{aw}$ 已在某些数据集上能够取得优异成果，但当引入关键词的相关性权重后，$EdecayRank_{aw+rw}$ 的表现得到显著加强。这一结果强调了在关键词提取中，关键词相关性权重的重要性，以及其在提升模型整体性能方面的作用。通过结合算法权重与相关性权重，EdecayRank 方法在多个数据集上展现出了出色的泛化能力和性能优势。

3.3 有效性威胁讨论

我们对所提出的关键词提取方法的有效性进行了全面评估，并同时探讨可能对结果产生影响的构造有效性、内部有效性和外部有效性威胁。

针对构造有效性威胁，我们采用了 F1 分数作为主要性能指标，以综合考虑精确率和召回率。为了确保评估的全面性，我们在实验中进行了消融实验，以验证相关性权重在提升性能中的作用。我们的实验数据在多个数据集上展示了模型的高性能，这支持了我们选择的评估指标的有效性。

在内部有效性方面，我们采取了严格的实验设计，包括在多个公认的标准数据集上进行评估。

外部有效性威胁关注模型在不同领域和文本类型上的泛化能力。我们的模型在多个数据集上的表现稳定，这为其泛化能力提供了初步的证据支持。然而我们意识到，为了进一步验证模型的泛化性，需要在更广泛的领域和语言中进行测试。尽管如此，我们的实验结果为模型在不同数据集上的稳定性和可靠性提供了有力的支持。

3.4 本章小结

本章介绍了一种无监督的指数权重衰减关键词提取方法 EdecayRank。该方法通过融合四种关键词提取算法 PromptRank、ChatGLM、TextRank 和 PatternRank 的预测结果，构建了一个集成模型。通过为各算法模型分配算法权重以及为每个关键词分配相关性权重，EdecayRank 旨在整合不同算法的优势，并利用关键词的相关性特征来提高关键词提取的准确性和全面性。经过对六个基准数据集的大量实验，与基线模型相比，EdecayRank 在关键词提取任务上取得了显著的性能提升。

4 基于无监督关键词提取算法的聚合搜索系统

4.1 系统需求分析

本章节对聚合搜索系统进行全面的需求分析。需求分析工作主要分为两个部分：一是功能性需求分析，二是非功能性需求分析。在功能性需求分析部分，我们将深入探讨系统内的各项具体功能，并对其功能需求提供详细的阐述。此外，非功能性需求分析的部分将集中探讨系统性能提升的关键指标和确保系统稳定运行的基本条件。

4.1.1 系统功能需求分析

功能性需求是系统的主要需求，从功能性需求的角度来分析，该基于无监督关键词提取算法的聚合搜索系统主要包括的功能有：相关内容的查询能力，对多源和多类型数据的检索功能以及对数据的可视化展示。

系统采用半自动化方法获取搜索系统索引数据，但所获数据需经过进一步处理以满足系统要求。数据处理流程涵盖数据清洗、关键词提取等关键步骤，处理后的数据将被存储于关系型数据库中。为了实现数据的有效管理和检索，系统对实体类型字段进行建模，并将数据同步至 Elasticsearch。此外，系统利用 Kibana 这一数据可视化工具，对索引数据进行直观展示。在用户交互层面，聚合搜索系统提供搜索查询功能，允许用户根据需求切换不同的数据源和数据类型，以实现快速、全面的信息检索，从而显著提升用户检索效率和搜索体验。

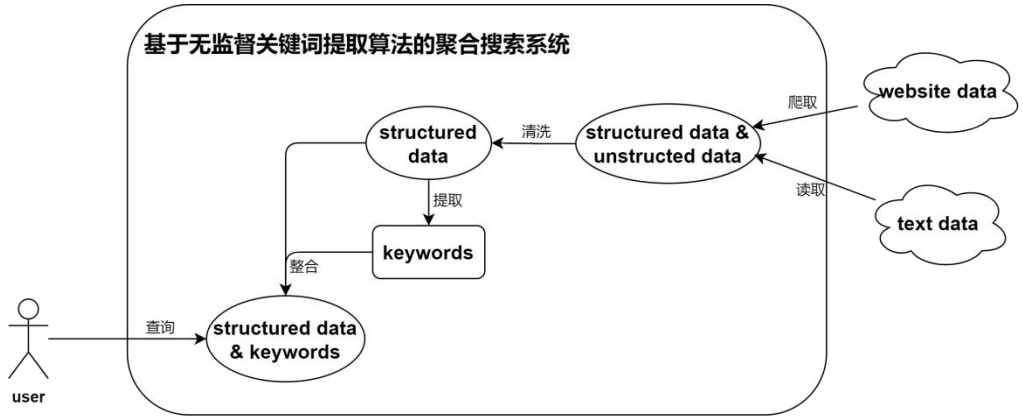


图 4-1 系统用例图

在本聚合搜索系统的技术实现层面，功能可划分为数据抓取与清洗、关键词提取、数据存储与同步，以及应用设计等方面。系统的数据抓取功能专注于从各类网页中采集信息，该过程需要依据数据源的特性定制爬虫程序。数据清洗环节对于后续操作至关重要，为后续使用 EdecayRank 方法提取关键词奠定了基础。只有精确地从数据中提取关键词，并将其与原始数据一并存储至数据库，才能在关键词查询过程中实现高效的检索性能。此外，为了支持数据的可视化展示及提供更为灵活、高效的全文搜索与复杂查询能力，系统需将数据库的数据同步至 Elasticsearch。最终，通过运用搜索查询技术、软件界面设计原理及交互设计算法，系统的功能模块得以实现，确保了用户操作的便捷性和交互体验的优化。

4.1.2 系统非功能需求分析

在完成功能需求分析之后，非功能性需求也是基于无监督关键词提取算法的聚合搜索系统的需求分析的一部分，并在很大程度上也影响系统的架构设计。若非功能性需求分析存在不足或遗漏，可能导致系统设计面临不可预见的风险和挑战。

（1）响应时间：在合理配置的环境下，系统应确保大部分功能请求的响应时间不超过 0.5 秒。即便在处理大量数据或复杂业务逻辑的场景下，系统的响应时间也应控制在 1 秒内，以完成数据的整合、传输和展示。

（2）业务量：系统设计应能够在不依赖于临时服务器扩容的情况下，支持至少一万用户同时进行查询操作。在访问量激增的情况下，系统应通过优化交互设计来应对潜在的加载问题。

（3）系统安全性：系统应定期更新信息，确保信息的可靠性和真实性，以更好地满足用户需求。

（4）兼容性：鉴于系统将服务于大量用户，且用户使用的终端设备多样，系统设计需考虑架构选择，并提前规划测试策略，确保系统兼容 Windows、Android、iOS 等不同操作系统及浏览器。

4.2 系统架构设计

在完成基于无监督关键词提取算法的聚合搜索系统完成需求分析之后，本研究选择了 B / S（浏览器/服务器）架构作为系统的应用开发体系结构。在这个系

统中，系统的技术架构可划分为数据抓取、关键词提取、知识持久化以及前端展示这四个部分。

数据抓取部分，对应系统架构图中的数据检索（retrieve）过程。该聚合搜索系统采用 Scrapy 框架，以并发请求和异步处理的方式，从多个数据源中高效检索数据。

关键词提取部分，对应系统架构图中的知识抽取（extract）部分。将数据抓取到本地后，利用本文提出的 EdecayRank 方法进行关键词提取。该算法在 PromptRank 算法的基础上，融合了 TextRank 的连贯性、ChatGLM 的语义一致性以及 PattenRank 的多样性优势，显著提升了关键词提取的精确度和广度，从而将原始数据转化为有价值的知识。

而知识持久化部分，对应架构图中的持久化（persist）阶段。系统需将提取的知识或信息存储于持久化存储介质中，以实现长期保存和即时访问。本文选择 MySQL 数据库作为数据存储方案，并设计了同步策略，将数据同步至 Elasticsearch 以提高搜索性能，同时建立了监控机制以确保数据的实时更新。

前端展示的部分，对应于架构图中的运行（running）阶段。后端根据业务需求，从 Elasticsearch 中检索相关文档。同时对于 Elasticsearch 中的数据，系统利用 Kibana 工具对其进行可视化展示以提供直观的数据视图。该系统的架构设计图如图 4-2 所示。

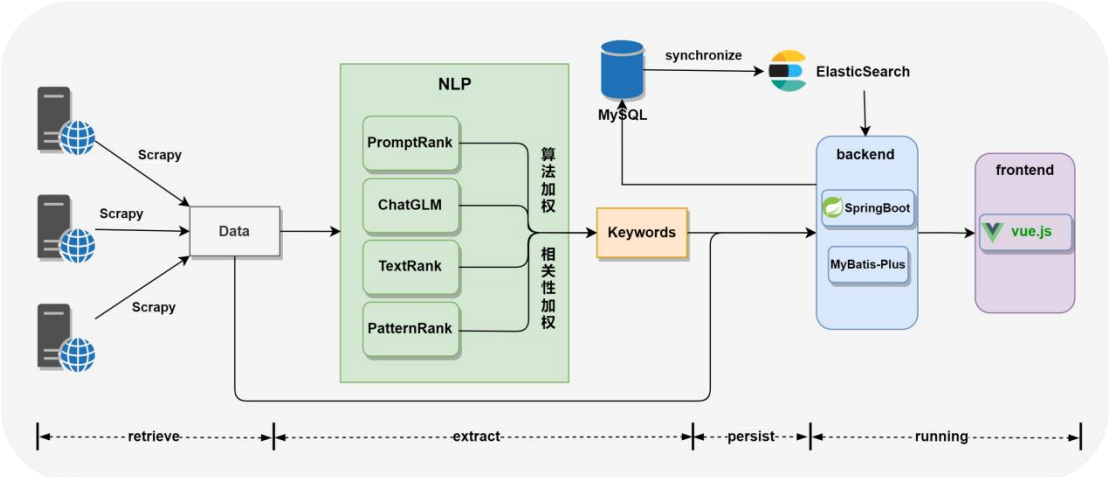


图 4-2 系统架构图

4.3 系统概要设计

在完成系统需求分析后，我们结合系统的架构设计，从技术的角度出发，对系统的功能进行了概要设计，为系统的详细设计与实现奠定基础。

4.3.1 数据抓取与清洗

基于无监督关键词提取算法的聚合搜索系统的设计目标是从多样化的信息源中高效地抓取数据。理论上，我们的系统设计允许接入无限多的数据源，但考虑到实际操作中的个人能力和可用资源的限制，我们决定将重点放在国内几个知名的网站上。国内技术社区与学术搜索平台，如 AMiner 学术搜索平台和 CSDN，因其丰富的技术与学术资源，成为本研究的首选数据源。AMiner 平台提供了包括论文标题、摘要及研究人员在内的详尽数据；而 CSDN 则以其广泛的技术博客、问答、文章和开发工具等资源，覆盖了软件开发、互联网、大数据、人工智能、区块链等多个关键领域。此外，为了进一步丰富我们的搜索结果，我们还整合了 Bing 图片搜索引擎，以便为用户提供更加全面和多样化的多媒体内容。

在数据抓取方面，我们主要采用了爬虫技术，同时也结合了人工收集的方式，以确保数据的完整性与真实性。对于不同的数据类型，我们制订了相应的数据清洗策略。例如，在处理结构化的数据时，我们使用 Python 进行 ETL（提取、转换、加载）处理，这一步骤有效地去除了数据中的“脏数据”，从而为接下来的关键词提取提供了高质量的数据基础。

4.3.2 关键词提取

关键词提取的核心是从文本自动识别并筛选出最能反映其主题和核心思想的词语或者短语。在本文提出的基于无监督关键词提取算法的聚合搜索系统中，关键词的提取环节无疑是该系统的关键组成部分。

关键词提取技术主要分为有监督和无监督两大类。有监督方法通过大量标注数据提升了模型的精确度和泛化能力,然而这种方法依赖数据标注，在实际应用成本较高。无监督方法不依赖于标注数据，因此得到了更广泛的应用。

考虑到本系统中数据源的多元化以及手动标注的成本高昂，我们选择了无监督关键词提取算法作为关键词提取的主要手段。为了进一步提升关键词

提取的效果，本文对现有的无监督算法进行了改进，提出了 EdecayRank 方法。

4.3.3 知识持久化

为了确保我们的聚合搜索系统能够提供有效且可靠的查询服务，并且数据得到长期稳定的保存，我们采取了一系列持久化存储策略来保障信息的完整性。在实际操作中，我们首先将搜集到的信息存储到 MySQL 数据库中。随后，我们建立了一套监控机制，利用 MySQL 的二进制日志（binlog）、数据库触发器或者定时轮询等机制，捕获需要更新的数据集，最终借助 Alibaba 开源中间件 Canal，来确保数据能够及时同步到 Elasticsearch 中。

在本系统中，我们将 Elasticsearch 作为核心搜索引擎。Elasticsearch 出色的全文搜索能力以及强大的实时数据处理功能，完美符合我们对系统性能的要求。

4.3.4 应用设计

在完成数据抓取、数据清洗、关键词提取以及知识持久化等基础工作之后，系统需进一步将处理后的信息呈现给用户，为此系统的应用设计成为关键环节。根据对系统需求的深入分析，我们将系统功能划分为信息展示与信息查询两大模块：信息展示模块负责对系统内的信息数据进行综合分析，并以直观的方式展示给用户；而信息查询模块则允许用户根据特定需求，检索并获取关键信息。系统的功能结构图如图 4-3 所示。

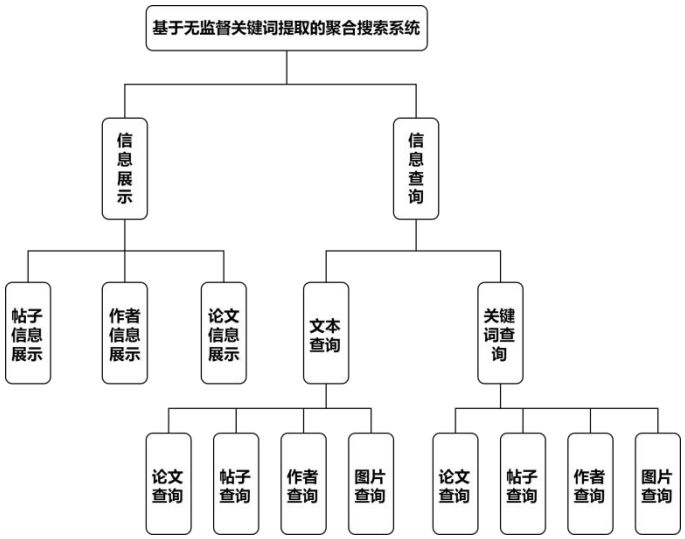


图 4-3 系统功能结构图

4.4 数据库设计

系统数据库的设计主要包括对系统中的概念进行定义以及对系统的物理结构进行描述。数据库的设计对系统品质和效能具有决定性作用，并直接影响系统的整体表现。这一设计工作涵盖了传统关系型数据库 MySQL 的结构定义，以及对 Elasticsearch 的数据建模。通过将数据存储到数据库中，并确保与 Elasticsearch 的同步，系统得以充分利用 Elasticsearch 的强大搜索性能、可扩展性和丰富的搜索功能，从而为用户提供更出色的搜索体验。

4.4.1 概念模型设计

在深入分析系统的需求并细致规划功能模块之后，我们针对系统的特性，明确了核心数据设计。该系统致力于提供一个用户友好的搜索平台，使得管理员在完成系统部署和数据录入后，用户能够通过简单的文本输入，执行各类信息的查询与检索任务。系统数据库的设计遵循了经典的实体－关系（Entity-Relationship，简称 ER）模型，其结构如图 4-4 所示。

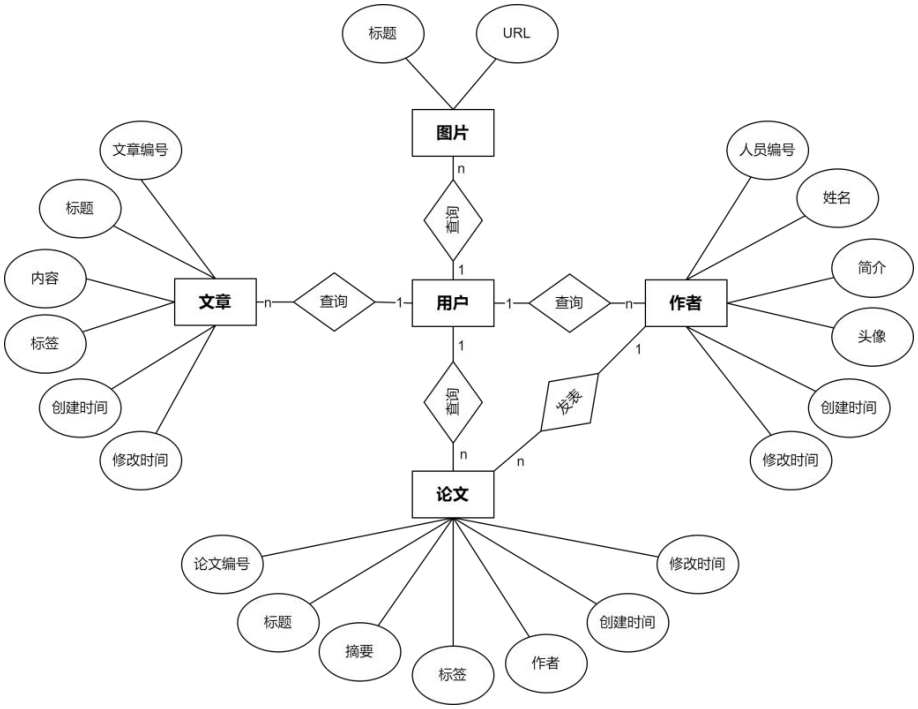


图 4-4 系统 E-R 图

该系统共定义了四种核心数据实体，分别为：文章、论文、作者以及图片。这些实体各自拥有一组特定的属性集，这些属性集不仅反映了实体的本质特征，也支持了系统的搜索和检索功能。在数据管理层面，管理员负责对这些实体及其

属性进行维护和更新，以确保数据的准确性和时效性。经过管理员的管理，这些数据实体将以结构化的形式展现给最终用户，从而提供丰富且易于导航的信息检索体验。

4.4.2 系统数据库设计

在本系统的数据库中，我们设计了一系列数据表。

如表 4-1 所示，作者信息表是用于记录系统中作者详细信息的数据库表。作者的 id 作为唯一标识，不允许为空。同样，作者的姓名、简介、创建时间、修改时间也被设置为必填项。创建作者记录时，系统会自动分配一个唯一的 id，并记录下作者信息的创建时间点，作为该作者的创建时间和修改时间的初始值。

表 4-1 作者信息表

字段名称	类型	是否允许为空	字段描述
author_id	bigint	否	作者的 id
author_name	varchar(128)	否	作者的姓名
author_profile	text	否	作者的简介
author_avatar	varchar(255)	是	作者的头像
create_time	datetime	否	创建时间
update_time	datetime	否	修改时间

如表 4-2 所示，文章信息表是用于记录系统中文章详细信息的数据库表。文章的 id 作为唯一标识，不允许为空。文章的标题、内容、标签、创建日期时间以及修改时间同样不允许为空。文章记录在被创建时，系统会自动分配一个唯一的 id 进行标识，并记录下文章信息的创建时间点，作为该文章的创建时间和修改时间的初始值。

表 4-2 文章信息表

字段名称	类型	是否允许为空	字段描述
post_id	bigint	否	文章的 id
post_title	varchar(255)	否	文章的标题
post_content	text	否	文章的内容
post_label	varchar(255)	否	文章的标签
create_time	datatime	否	创建时间
update_time	datetime	否	修改时间

如表 4-3 所示，论文信息表是用于记录系统中论文详细信息的数据库表。在论文的属性信息中，除论文的 id 不允许为空之外，论文的标题、摘要、作者、创建时间以及修改时间也不允许为空。论文记录在被创建时，系统会自动分配一个唯一的 id 进行标识，并记录下论文信息的创建时间点，作为该论文的创建时间和修改时间的初始值。

表 4-3 论文信息表

字段名称	类型	是否允许为空	字段描述
paper_id	bigint	否	论文的 id
paper_title	varchar(255)	否	论文的标题
paper_abstract	text	否	论文的摘要
paper_label	varchar(255)	否	论文的标签
paper_author	varchar(128)	否	论文的作者
create_time	datatime	否	创建时间
update_time	datetime	否	修改时间

如表 4-4 所示，为图片信息表，记录系统中图片的信息。在图片的属性信息中，图片的标题以及 url 不允许为空。

表 4-4 图片信息表

字段名称	类型	是否允许为空	字段描述
picture_title	varchar(255)	否	图片的标题
picture_url	varchar(255)	否	图片的链接

4.5 系统详细设计与实现

在系统的详细设计阶段，我们对系统架构中的各个模块进行了深入地阐述和分析，针对每个模块，详细描述了其核心功能，并对其设计方法进行了详尽的说明。

4.5.1 数据抓取与清洗

数据检索环节主要包括了数据抓取和数据清洗两个步骤。

1 数据抓取

考虑到数据来源的多样性以及各网站结构的差异性，为确保向用户提供具有全面性和准确性的检索结果，多数据源的数据获取成为项目实施的关键基础环节，所有后续的系统实现均依赖于这一环节的高效执行。

尽管网站间在样式和结构上存在差异，通过明确定义的建模字段，我们利用 Scrapy 框架进行爬虫开发，使得初始开发成本虽高，但后续针对不同网站的代码调整工作量得以降低。鉴于数据抓取任务的规模性，执行过程中需对响应数据进行人工验证，以保证数据的准确性。

数据爬取的时序图如图 4-5 所示。数据抓取流程通常由调度器、URL 管理器、下载器、解析器和数据清洗器五个关键组件组成。为了应对传统爬虫中各环节的紧密依赖性所带来的挑战，我们引入调度器作为中介，实现了组件间的有效解耦，这一架构改进增强了系统的稳定性和组织性。URL 管理器负责搜集和筛选 URL，下载器高效地获取网页内容，解析器将非结构化数据转换为结构化形式，而数据清洗器则负责剔除无效或错误的数据，确保数据质量满足系统需求。

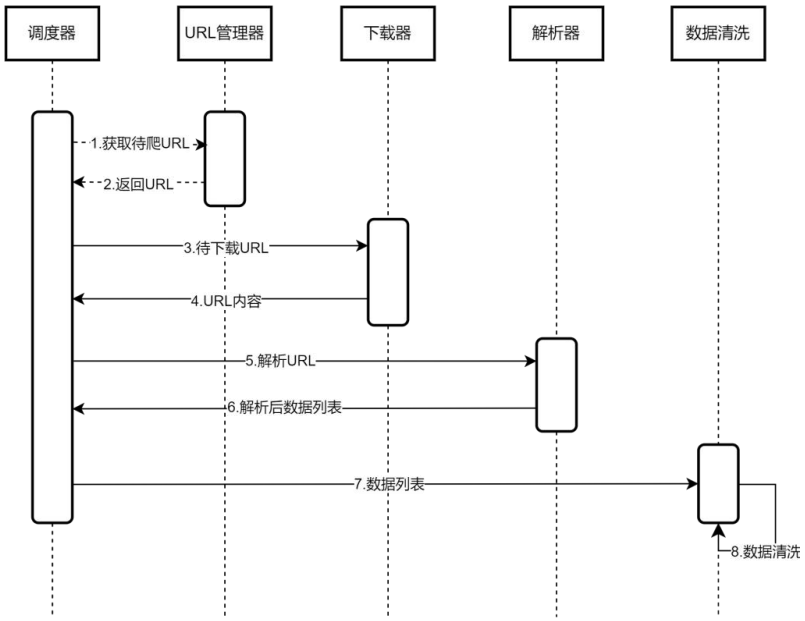


图 4-5 数据爬取时序图

以 AMiner 数据源为例，该平台会在一个界面上展示同一作者的所有相关论文列表，若想获取每篇论文的详细信息，则需跳转到一个专门设计的细节视图页面。在这一页面上，用户能够获得包括论文摘要、发表的期刊等关键数据，如图 4-6 所示。

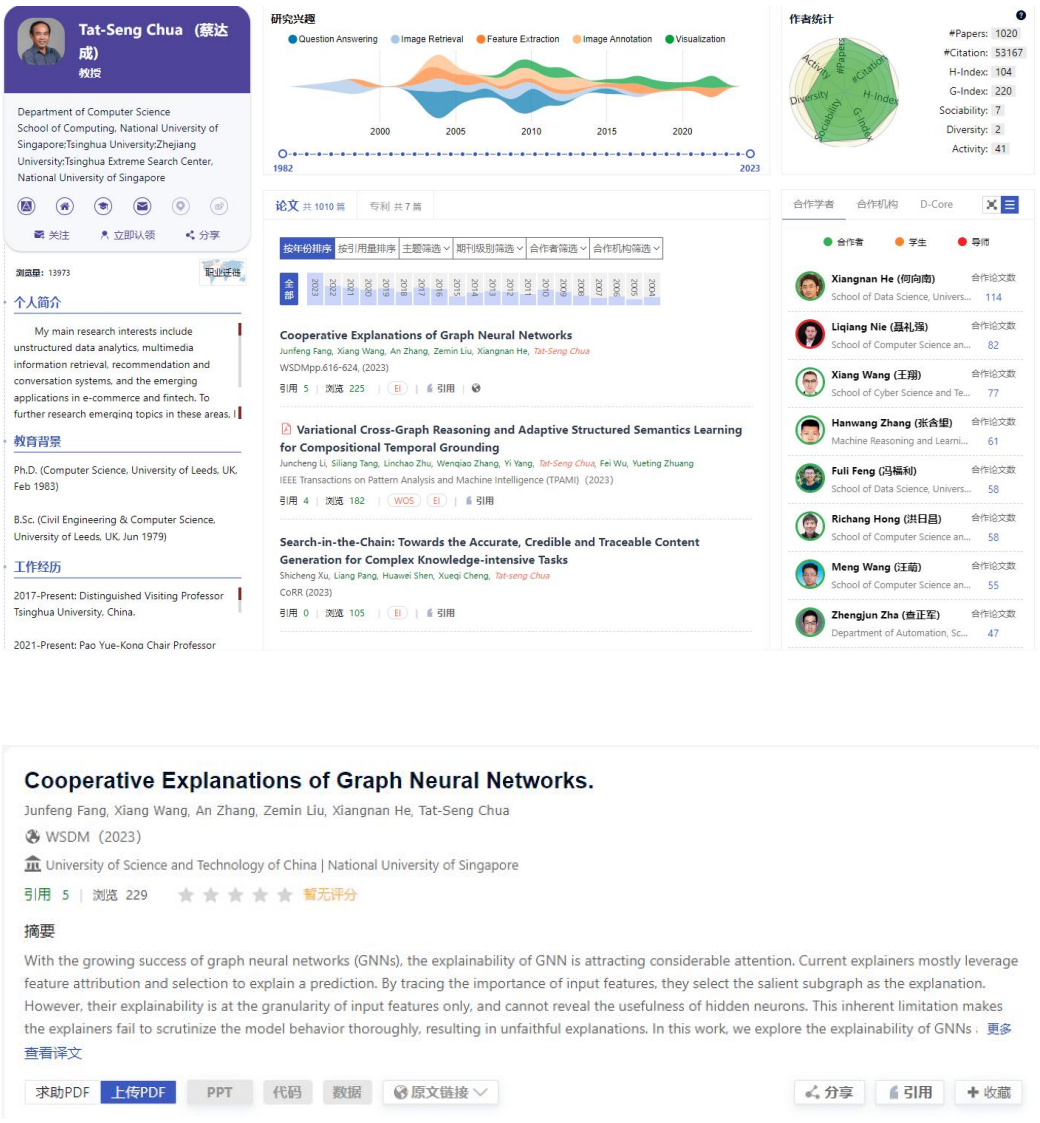


图 4-6 AMnier 论文索引界面

通过数据爬取流程，本文成功从 Aminer 中爬取了 13067 条论文数据。这些数据按照作者进行分类，并以.csv 格式文件存储。其中每条数据记录都包含了丰富的信息字段，包括论文的地址（Url），标题（Title），摘要（Abstract），发表信息（Journal And Time）。作者论文数量分布图如图 4-7 所示，其中横轴代表不同的作者，纵轴代表作者发表或参与的论文数量。

2 数据清洗

在数据抓取过程中，所获取的数据往往存在不完整性、噪声干扰以及一致性问题。为了确保数据质量，数据清洗是一个不可或缺的环节。数据清洗的方法多样，且每种方法都有其特定的适用场景和相应的成本。因此，选择合适的数据清洗策略需基于数据中存在的具体错误类型。

在数据清洗前，我们的首要任务是将数据进行预处理并存储到数据库中。接着，通过审核元数据，如字段名、来源和描述，以获得数据的清晰认识，为进一步处理做准备。数据清洗包括三个主要步骤：

首先，处理缺失值，采用人工追踪和筛选方法，根据原则进行填充；

其次，执行格式内容清洗，解决多个数据源中的时间、日期和数值格式不一致以及错误字符等问题，并通过半自动化校验统一格式，增强数据集的可读性和一致性；

最后，进行逻辑错误清洗，其中包括数据去重、剔除异常值和修正矛盾信息，确保通过逻辑检验的数据输入到模型中。

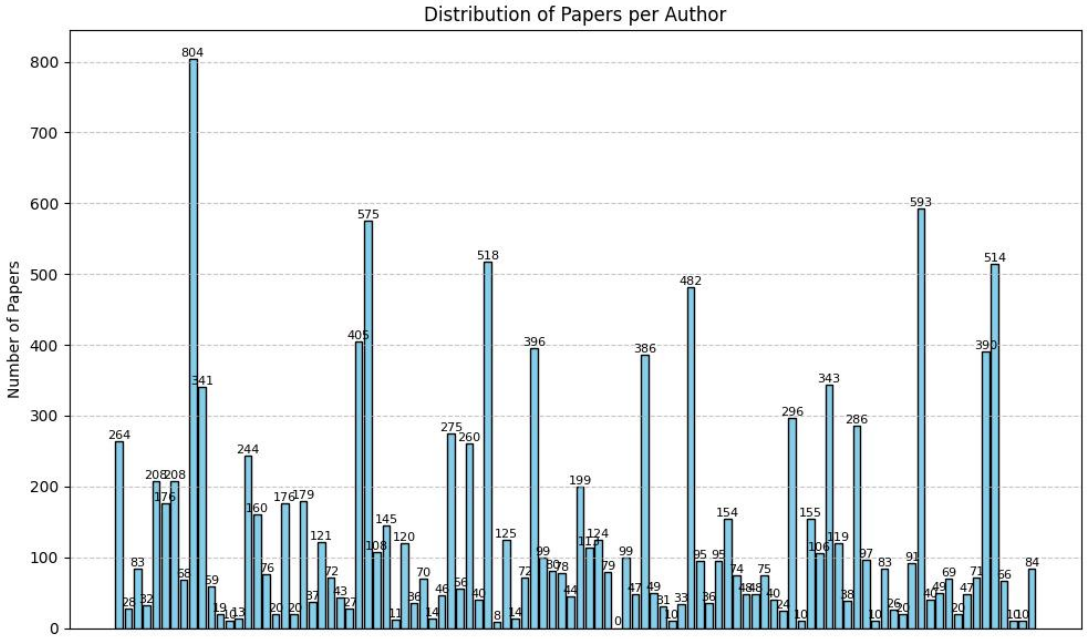


图 4-7 作者论文数量分布图

4.5.2 关键词提取

关键短语提取是自然语言处理（NLP）领域的关键技术，其目的是从文本中识别并提取能够概括文档核心内容的短语。这一过程不仅能够显著提升信息检索的效率，还能帮助用户迅速且准确地从大量文本中提取关键信息。

在本文的 3.1 节中，我们提出了一种无监督的指数权重衰减关键词提取算法 EdecayRank，通过集成四种关键词提取算法来增强关键词提取的精确度和全面性。该方法在现有的最先进（State-of-the-Art, SOTA）PromptRank 模型的基础上引入了 TextRank 模型、ChatGLM2-6B 模型以及 PattenRank 模型。每种关

键词提取算法都有其独特的优势与局限性，而采用多种无监督算法能够收集到更广泛的关键词候选集，从而在一定程度上弥补单一算法的不足。考虑到不同算法可能提取出相同的关键词，我们认为这些关键词代表了研究的核心概念和关键特征。通过合理组合调整各个算法的权重，可以增加这些关键词的权重，最终输出具有较高权重的核心关键词。

此外，关键词提取算法能够根据关键词的相关性进行排序，理论上输出的关键词列表中排名靠前的关键词与文本的相关性更强。本方法利用这一特性，使用指数衰减函数为关键词分配相关性权重，使得排名靠前的关键词获得更高的权重。

具体实施过程如下：首先，将经过清洗的数据作为输入。每个模型独立提取出 20 个候选关键词，分别表示为 $C_{PromptRank} = \{c_1, c_2, ..., c_{20}\}$，$C_{ChatGLM} = \{c_1, c_2, ..., c_{20}\}$，$C_{TextRank} = \{c_1, c_2, ..., c_{20}\}$，$C_{PatternRank} = \{c_1, c_2, ..., c_{20}\}$。其中 $c_1, c_2, ..., c_{20}$ 代表一个个关键词。接着，根据各算法的性能，为它们分配相应的权重，以体现其在关键词提取过程中的贡献。然后，考虑关键词的相关性信息，并采用指数衰减函数量化关键词的相关性权重。最后，结合关键词的频率、算法权重以及相关性权重计算出关键词的得分（$Score$），并根据得分对关键词排名。最终选择 top15 关键词作为输出结果。

以一篇论文摘要为例，在将一段文本（Abstract）输入 EdecayRank 后，模型的输出结果如图 4-8 所示。

Abstract : nanoparticle tracking analysis (nta) has been applied to characterising soot agglomerates of particles and compared with transmission electron microscoscopy (tem). soot nanoparticles were extracted from used oil drawn from the sump of a light duty automotive diesel engine. the samples were prepared for analysis by diluting with heptane. individual tracking of soot agglomerates allows for size distribution analysis. the size of soot was compared with length measurements of projected two-dimensional tem images of agglomerates. both the techniques show that soot-in-oil exists as agglomerates with average size of 120nm. nta is able to measure particles in polydisperse solutions and reports the size and volume distribution of soot-in-oil aggregates; it has the advantages of being fast and relatively low cost if compared with tem.nanoparticle tracking analysis (nta) has been applied to characterising soot agglomerates of particles and compared with transmission electron microscoscopy (tem). soot nanoparticles were extracted from used oil drawn from the sump of a light duty automotive diesel engine. the samples were prepared for analysis by diluting with heptane. individual tracking of soot agglomerates allows for size distribution analysis. the size of soot was compared with length measurements of projected two-dimensional tem images of agglomerates. both the techniques show that soot-in-oil exists as agglomerates with average size of 120nm. nta is able to measure particles in polydisperse solutions and reports the size and volume distribution of soot-in-oil aggregates; it has the advantages of being fast and relatively low cost if compared with tem.

TOP-K : ['soot agglomerates', 'soot', 'soot nanoparticles', 'agglomerates', 'polydisperse solutions', 'oil', 'size distribution analysis', 'particles', 'analysis', 'individual tracking', 'heptane', 'nta', 'low cost', 'volume distribution', 'used oil']

图 4-8 EdecayRank 输出关键词列表

由图 4-8 可知，EdecayRank 输出的 top15 个关键词分别是“soot agglomerates”，“soot”, “soot nanoparticles”, “agglomerates”, “polydisperse solutions”,“oil”, “size distribution analysis”, “particles”, “analysis”, “individual tracking”, “heptane”, “nta”, “low cost”, “volume distribution”, “used oil”。

4.5.3 知识持久化

在完成数据抓取、清洗及关键词提取之后，系统需将处理后的知识进行存储。知识存储作为系统架构的核心组成部分，承担着将数据持久化到 MySQL 数据库，并同步至 Elasticsearch 的至关重要的任务。

1 关系型数据库

首先，我们分析整个系统的数据要求，设计了一个符合要求的数据库模型。这个模型定义了数据的类型、表与表之间的关联以及索引。我们的目的是确保数据在存储上的高效率与准确性，同时降低数据的冗余性并保障其保持一致性。

其次，我们编写了程序来实现数据的写入和更新操作。这涉及对数据库的 CRUD 操作（增删改查），确保在数据写入和更新过程中，数据的一致性和完整性。

2 索引数据存储

为了更有效地在 MySQL 和 Elasticsearch 之间同步数据，我们选择了一种增量同步策略，这种策略依靠 MySQL 的二进制日志（binlog）来跟踪数据的变动情况。只需要传输自上一次同步后发生更改的数据，这样不仅能够确保数据的同步性和准确性，而且大幅降低了数据传输的负担，缩短了同步所需的时间。

在开始同步前，我们首先需要确认 MySQL 的 binlog 功能已启用，这是通过调整 MySQL 的配置设置并重启服务来完成的。然后，我们部署了 Alibaba 的 Canal 工具，它通过模拟 MySQL Slave 的行为，向 MySQL Master 发起数据 dump 的请求。MySQL Master 响应这些请求，并开始将更新的 binlog 发送给 Canal。Canal 对这些日志进行解析，并将格式化后的数据发送到 Kafka 队列中。我们的 Adapter 组件订阅 Kafka 队列，从中提取数据，并将其转换后更新到 Elasticsearch 中。Canal 架构如图 4-9 所示。

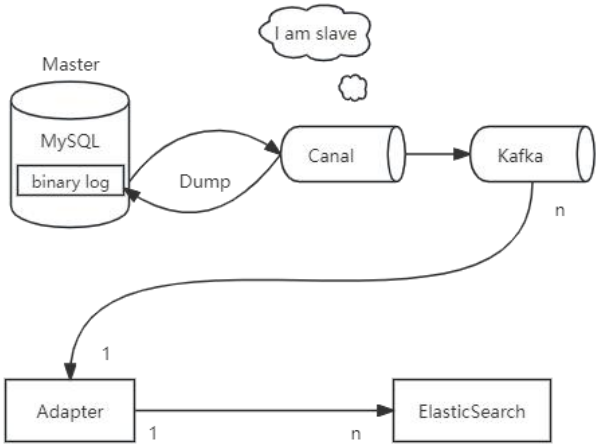


图 4-9 Canal 架构图

在 Elasticsearch 中，数据被组织成一系列的索引，其中每个索引都包含了若干个独立的文档，并以 JSON 文件格式进行存储。索引可以进一步划分为多个分片，而这些分片都可以在集群的不同节点上复制，这样的设计既支持了数据的水平扩展，同时也提供了容错机制。Elasticsearch 利用倒排索引机制，为用户提供了强大的全文搜索能力，数据实际以倒排索引的形式存储在磁盘上，而每个分片都对应着一个独立的 Lucene 索引。每当用户发起搜索请求时，Elasticsearch 会将查询任务分发到所有的分片上并行处理，然后将各个分片的搜索结果汇总起来，快速地返回给用户。

4.5.4 应用设计

在完成了数据抓取、清洗、关键词提取和知识整合后，聚合搜索系统需向用户展示处理结果。为此，我们专注于应用设计，实现查询功能并优化用户界面与交互，采用同级列表布局清晰展示跨数据源信息，增强用户体验及信息获取的完整性。

1 搜索查询

本系统的信息查询主要依托于搜索查询功能，允许用户在搜索框中输入关键词或特定短语，以检索相关数据。用户也可以切换同级列表，查看来自不同来源和类型的信息。

用户在前端界面的搜索框中输入搜索条件来启动查询程序。从时序图的角度分析，用户在输入搜索关键词后，前端界面会捕获这些输入，并将其转

换成一个 HTTP 请求。当后端接收到请求之后，它会根据请求的 URL 以及 HTTP 方法将请求路由至相应的 Controller。然后调用服务层来构建查询请求，并通过 Elasticsearch 提供的 RESTful API 对 Elasticsearch 发起查询操作。Elasticsearch 处理查询请求然后向后端返回查询结果。后端应用程序在接收这些数据后，会将其封装成一个 HTTP 相应，然后发送回前端。前端会对数据进行相应的渲染和展示，最终将搜索结果以直观的形式呈现给用户。搜索查询的时序图如图 4-10 所示。

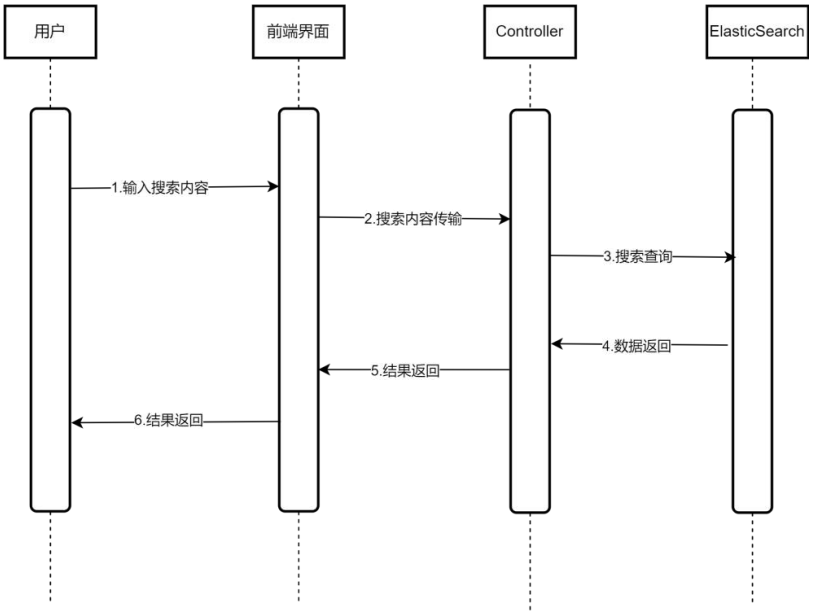


图 4-10 搜索查询时序图

Controller 类作为控制器的实现，负责根据前端请求调用相应服务，并将处理结果反馈给前端。SearchFacade 类则运用了设计模式中的门面模式，巧妙封装了子系统的核心功能和实现细节，通过定义简化的接口实现对子系统的调用，降低了系统耦合度，提高了灵活性。该类根据传入的参数类型，智能地将请求分发至相应的子系统。子系统完成查询操作后，将结果返回给 Controller，再由 Controller 将结果传递给前端界面，完成用户界面的展示。搜索查询的类图如图 4-11 所示。

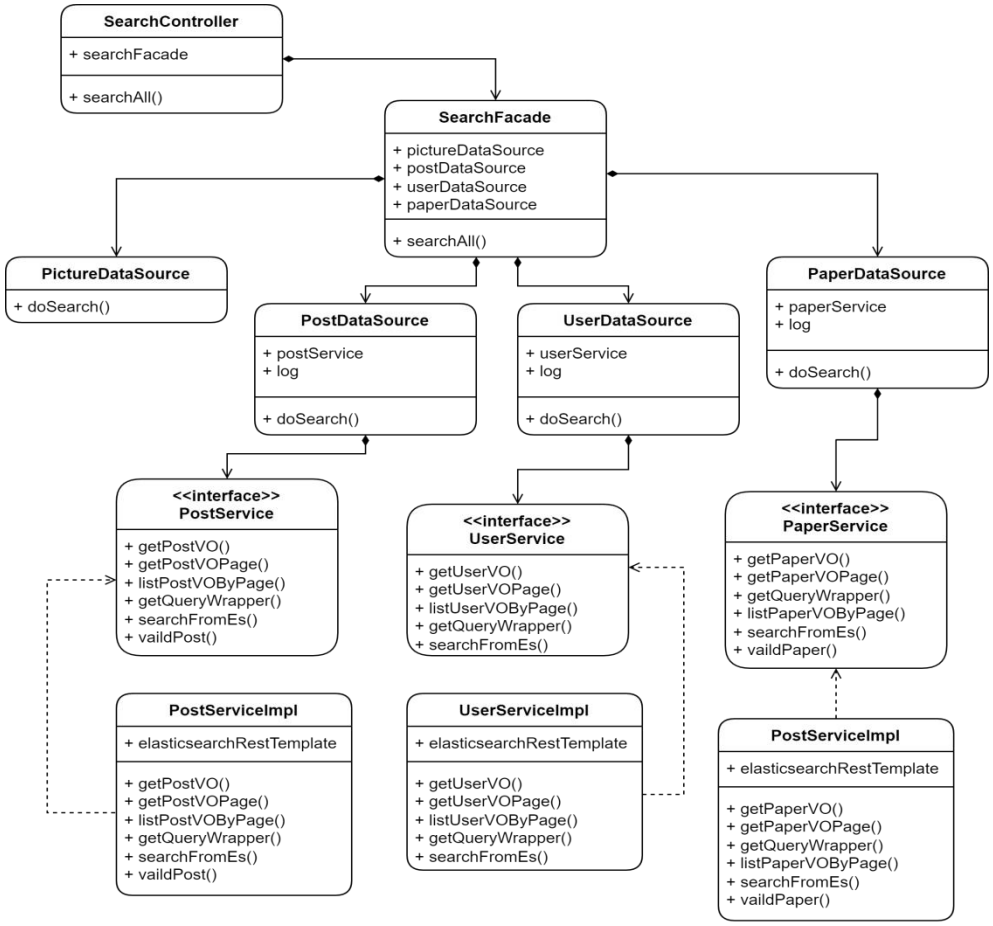


图 4-11 搜索查询类图

本项目的使用遵循简洁直观的步骤，便于用户高效地进行信息检索。首先，用户通过互联网在线访问本项目地址，启动查询操作。页面设计以用户友好为核心，顶部设有一个清晰的输入框，用户在此键入检索关键词或条件。随后，页面布局展示了四个主要的同级信息分类：文章、论文、图片和作者，每个分类均由独立的数据源提供支持，确保了数据的多元性。

当用户通过输入框输入特定关键词，例如“LSTM”，并选择相应的数据类别，如“论文”，系统便启动其综合搜索流程。该流程涉及对大量数据的检索与分析，目的在于筛选出与用户查询高度相关的信息。搜索结果将以一种用户友好的界面展示，涵盖了与查询关键词紧密相关的多种学术资源。论文搜索结果如图 4-12 所示。

为了进一步提升用户体验，系统设计了多功能的数据源切换机制。这使得用户能够根据需求，从不同的数据源中提取信息，从而满足对多样化信息来源的检索需求。例如，当用户希望深入了解 LSTM 相关技术时，他们

可以选择“文章”类别，系统将展示一系列与 LSTM 技术相关的博客文章，如图 4-13 所示。

若用户需要直观理解 LSTM 的架构，他们可以选择“图片”类别，系统将提供一系列与 LSTM 架构相关的图像资料，如图 4-14 所示。

此外，系统还提供了“作者”类别，记录了论文作者的个人信息。通过这一功能，用户可以通过作者姓名快速获取其详细信息，如图 4-15 所示。

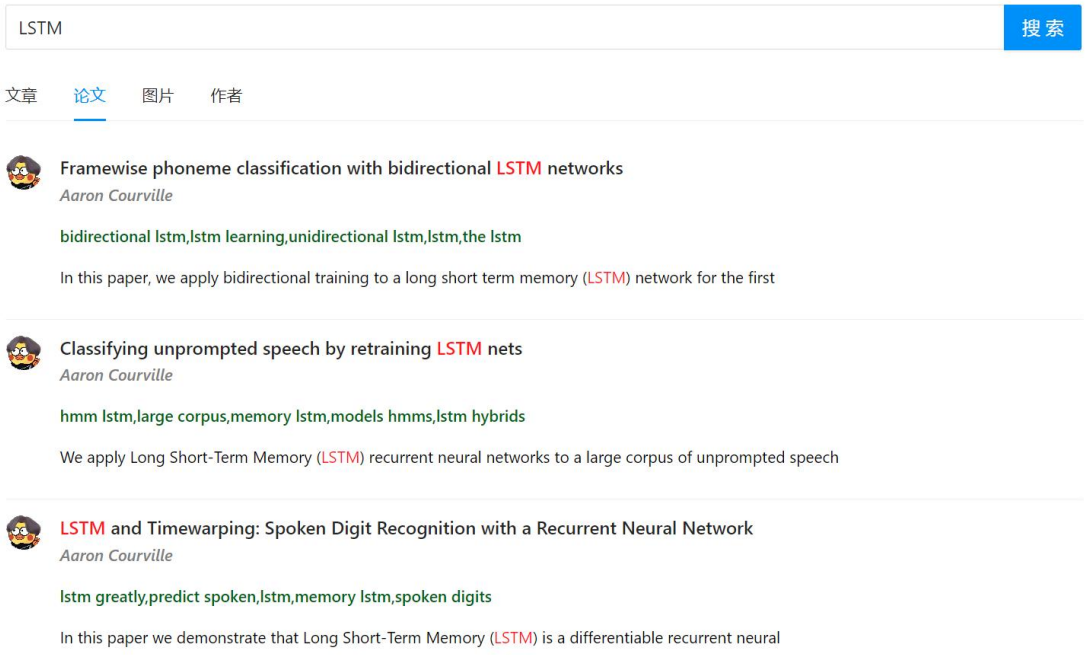


图 4-12 论文搜索查询界面图

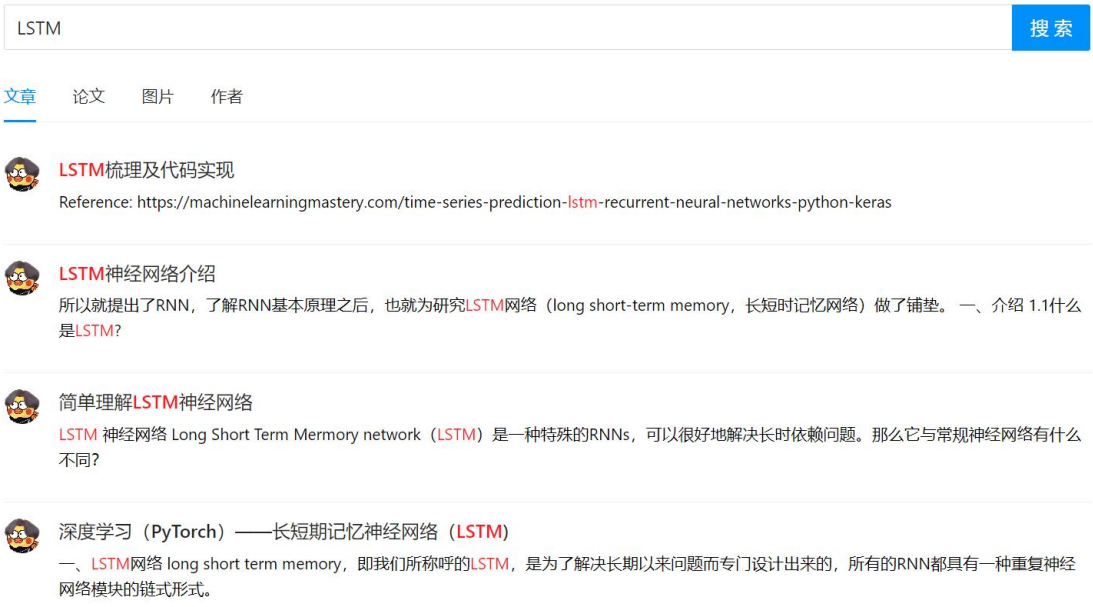


图 4-13 文章搜索查询界面图

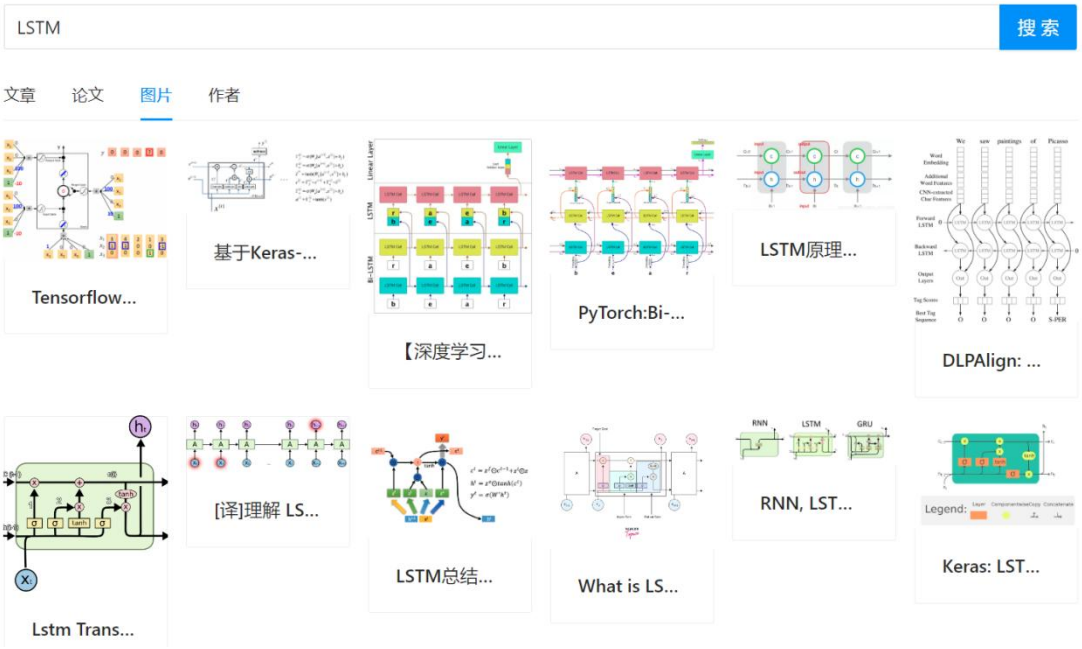


图 4-14 图片搜索查询界面图

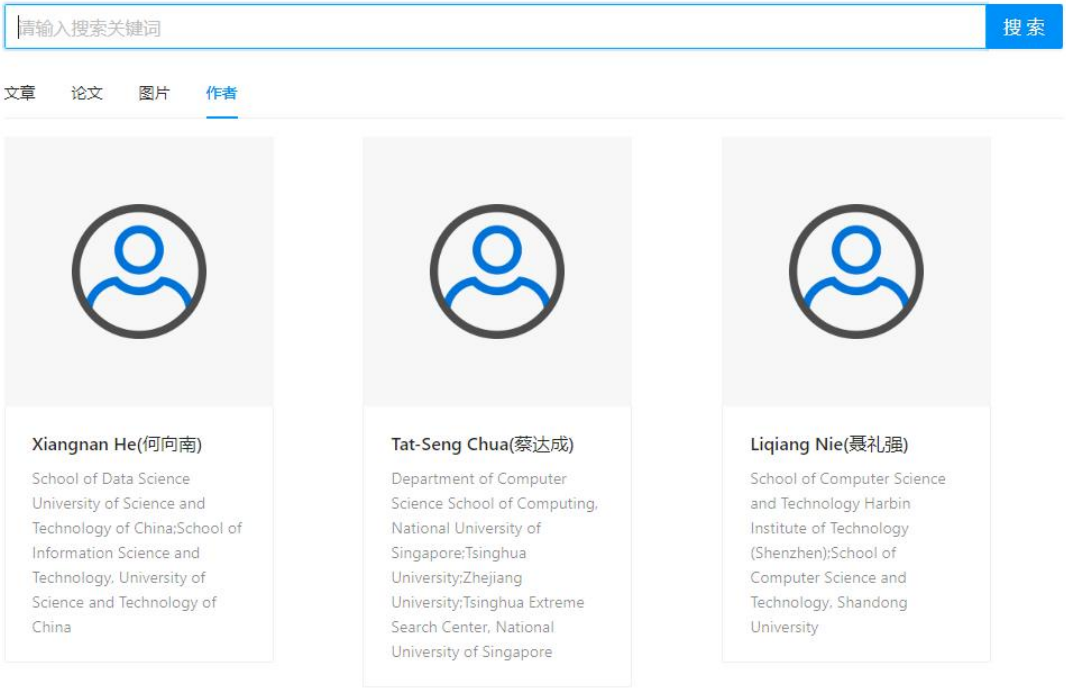


图 4-15 作者搜索查询界面图

2 可视化看板

本系统集成了 Kibana 这一开源的数据分析和可视化平台，实现了将存储在 Elasticsearch 中的数据以图表、表格等形式直观展示的功能，帮助用户快速识别数据中的深层模式和发展趋势。可视化图表如图 4-16、4-17 所示。

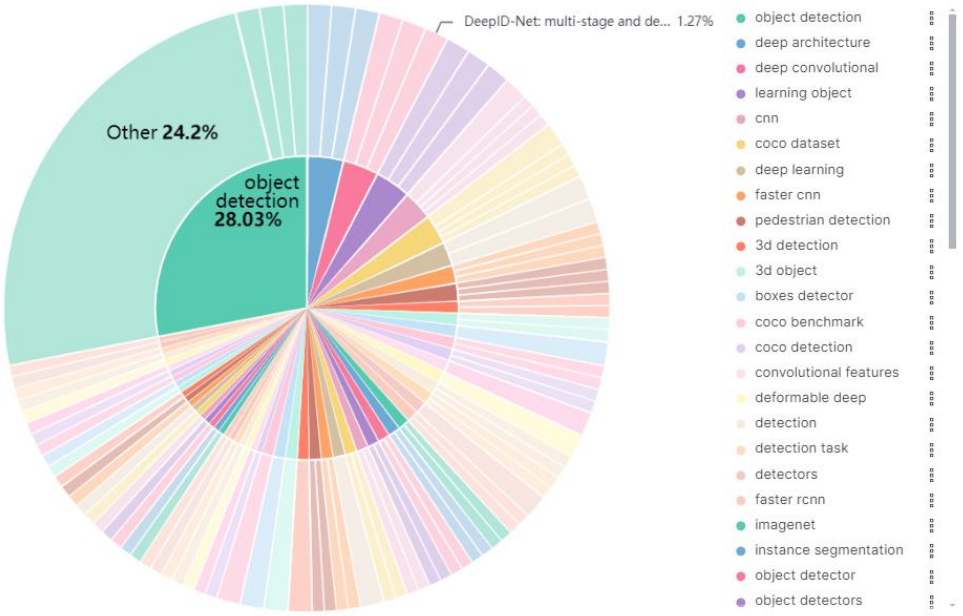


图 4-16 可视化图表 1

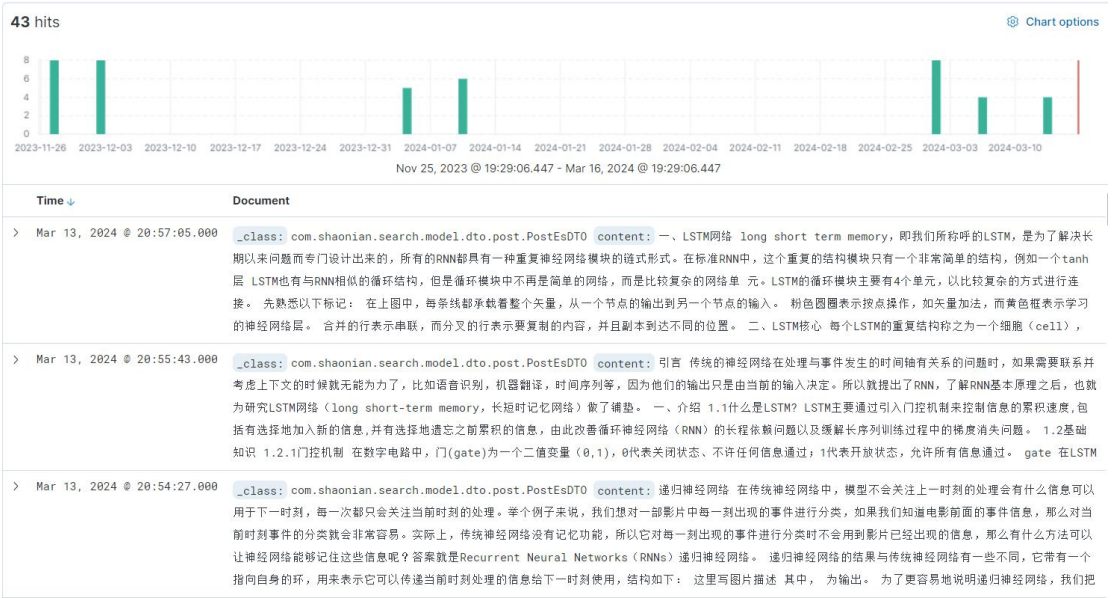


图 4-17 可视化图表 2

4.6 测试

系统测试是针对整个系统进行全面审查的过程，目的是揭示潜在错误，验证系统的可靠性，保证系统能更好地服务用户。

4.6.1 功能性测试

在本系统中采用动态黑盒测试方法，其核心在于忽略代码实现细节，仅评估系统功能是否达到预期目标。

测试流程分为三个阶段：首先，测试人员构造一系列输入用例，并通过系统执行以产生相应的输出结果；其次，将实际输出与预定的预期结果进行对照分析，以验证软件功能的准确性与完整性；最后，基于对照分析的结果，做出是否推进至生产环境部署或需进行进一步调试的决策。在此过程中，黑盒测试方法特别适用于识别软件界面的可用性问题、功能的实现缺陷以及输出结果的异常情况，从而为提升软件的可靠性与用户满意度提供重要的质量保证。

在本系统中，表 4-5 展示了关键的测试用例及其预期与实际测试结果。该表详细说明了选定测试用例和预期输出，以便通过测试反馈确认用例的通过情况。

表 4-5 测试用例表

编号	测试用例说明	预测结果	测试结果
001	用户输入关键词后选择文章列表	用户输入想要查询的关键词并选择文章列表后，所展示出来的词条信息，呈现与关键词相关的词条	测试通过
002	用户输入关键词后选择论文列表	用户输入想要查询的关键词并选择论文列表后，所展示出来的词条信息，呈现与关键词相关的词条	测试通过
003	用户输入关键词后选择作者列表	用户输入想要查询的关键词并选择作者列表后，所展示出来的词条信息，呈现与关键词相关的词条	测试通过
004	用户输入关键词后选择图片列表	用户输入想要查询的关键词并选择图片列表后，显示相关的图片	测试通过
005	用户搜索时输入特定词语或短语并选择相应列表	用户输入特定词语或短语选择相应的列表后，呈现包含用户输入内容的文档	测试通过
006	用户查看数据可视化面板	用户查看数据可视化面板后，显示数据的可视化面板	测试通过

4.6.2 性能测试

为了评估系统的性能，我们进行了详细的性能评估。利用对系统响应速度和吞吐率等关键指标的实时监控，来全面评价系统的执行效能和性能。

1 性能测试内容

（1）负载测试：负载测试用于评估系统在面临用户需求逐渐增多时的性能表现，尤其是识别系统所能支持的最高并发用户数量。

（2）压力测试：压力测试则进一步推动系统至极限，目的是识别系统崩溃的临界点和崩溃方式，确定系统在资源过载情况下的处理能力。

2 性能测试指标

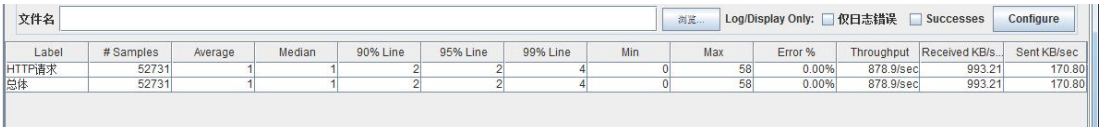
（1）响应时间：响应时间是系统接收并响应请求所需的时间，是衡量用户体验的重要指标。

（2）每秒查询率：每秒查询率（QPS）专注于评估查询服务器在规定时间内处理请求的能力。

（3）吞吐量：吞吐量直接反映了系统在单位时间内处理请求的总数，是评估系统处理能力的核心指标。

3 性能测试结果及分析

为了评价系统的各项性能，本文采用了 JMeter 这一多功能的测试工具。JMeter 不仅提供了一个用户友好的图形界面，还允许用户通过编程脚本进行高级测试配置。JMeter 支持包括 HTTP、FTP、JMS、SOAP、LDAP 在内的多种网络协议，从而确保测试工作能够覆盖到各种应用场景。测试结果如图 4-18 所示。


文件名 浏览... Log/Display Only: 仅日志错误 Successes Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/s	Sent KB/sec
HTTP请求	52731	1	1	2	2	4	0	58	0.00%	878.9/sec	993.21	170.80
总体	52731	1	1	2	2	4	0	58	0.00%	878.9/sec	993.21	170.80

图 4-18 测试结果图

4.7 本章小结

本章首先对基于无监督关键词提取算法的聚合搜索系统进行了需求分析，从功能性需求和非功能性需求两个方面深入分析了系统的需求。基于这些需求分析的结果，本章明确介绍了系统的设计目标，并对技术实现架构和总体功能结构进行了概要设计。在总体功能结构方面，系统被划分为若干功能模块，

并对每个模块的功能进行了详尽的阐述与分析。此外，本章还详细介绍了检索系统的数据库结构设计，包括数据库的概念模型和具体的表结构设计。最后，为了确保系统各功能模块的正确性和性能，对系统实施了系统测试，以验证各功能模块是否正常运行并评估系统性能。

5 总结与展望

本章将回顾基于无监督关键词提取算法的聚合搜索系统的主要工作，并概述论文的核心内容。同时，对系统未来可能的改进方向提出了建议和预期。

5.1 全文总结

本文构建了一个聚合搜索系统，该系统能够周期性地从多个门户网站自动收集数据。经过一系列数据处理流程，如爬取、筛选、清洗，构建了一个综合的数据仓库，确保用户能够通过该系统获取到高质量的数据结果。系统进一步使用EdecayRank 关键词提取方法，有效识别文本中的核心主题和关键信息，从而提升系统返回结果的相关性和准确性。该系统实现了多源数据的整合，并在一个统一的界面上提供了搜索与分析功能，满足了用户对于搜索效率的需求。

本文主要完成的工作有：

（1）完成了对关键词提取技术、爬虫技术、搜索引擎、聚合搜索相关背景及意义的研究，并总结了它们在国内外发展状况。

（2）利用爬虫技术结合人工采集的方式，完成了对主流门户网站数据的收集与预处理，确保了数据集的完整性与真实性。

（3）提出了一种无监督的指数权重衰减关键词提取算法 EdecayRank，该算法显著提高了关键词提取的准确性。与现有最先进的 PromptRank 方法相比，EdecayRank 在返回 top5、top10 和 top15 个结果时的 F1 分数分别提升了 6.49%、5.94%和 4.42%。

（4）设计并实施了一套数据同步方案，使用 Canal 中间件实现了从 MySQL 数据库到 Elasticsearch 的无缝数据迁移，极大地提升了搜索系统的数据索引效率和搜索性能，确保了用户能够享受到更加迅捷和精准的搜索体验。

（5）通过前后端代码的编写实现了基于无监督关键词提取算法的聚合搜索系统，设计了友好的用户界面，并完成了系统搜索查询和数据可视化的功能。

（6）对聚合搜索系统的核心功能进行了全面的测试与评估，确保了系统在线上生产环境中的稳定性和用户服务的可靠性。

5.2 未来扩展方向

本系统已开发完成并投入线上运行。鉴于项目时间限制，系统仍有进一步优化的空间。随着系统的持续迭代更新，未来将针对以下方面进行改进：

（1）本文提出的无监督关键词提取算法 EdecayRank 基于集成学习原则，结合多种无监督技术和关键词的相关性特征，以提升提取效率。随着技术进步，计划引入更先进的模型以进一步提高算法性能。此外，考虑整合知识图谱等外部知识资源，以增强关键词提取的准确性和深度。

（2）当前聚合搜索系统在处理用户查询时，仅提供文章、用户、文献和图片四种数据源的结构化信息。为增强用户体验，未来版本将探索集成更多类型的数据源，以提升查询效率和满足用户的多元化需求。

（3）利用用户的历史搜索记录，实现个性化内容推送的功能。通过对用户过去搜索行为的深入分析，系统能够准确地识别和预测用户的兴趣和偏好，进而主动地推荐与这些偏好相关的信息，以进一步提高用户获取信息的效率和满意度。

参考文献

[1] DUIN R P, TAX D M. Experiments with classifier combining rules[C]. International Workshop on Multiple Classifier Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000: 16-29.

[2] 常耀成, 张宇翔, 王红等. 特征驱动的关键词提取算法综述[J]. 软件学报, 2018, 29(07): 2046-2070.

[3] Dietterich T G. Ensemble methods in machine learning[C]. International workshop on multiple classifier systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000: 1-15.

[4] CAMPOS R, MANGARAVITE V, PASQUALI A, et al. YAKE! Keyword Extraction from Single Documents using Multiple Local Features[J]. Information Sciences, 2020, 509: 257-289.

[5] LI B, YANG X, WANG B, et al. Efficiently mining high quality phrases from texts[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2017 ,31(1).

[6] LI B, YANG X, ZHOU R, et al. An efficient method for high quality and cohesive topical phrase mining[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(1): 120-137.

[7] MIHALCEA R, TARAU P. TextRank: Bringing Order into Texts[C]. Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004: 404–411.

[8] WAN X, XIAO J. Single document keyphrase extraction using neighborhood knowledge[C], proceedings of the AAAI, 2008 ,8: 855-860.

[9] FLORESCU C, CARAGEA C. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents[C]. Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), 2017: 1105-1115.

[10] BOUGOUIN A, BOUDIN F, DAILLE B. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction[C]. International joint conference on natural language processing (IJCNLP), 2013: 543–551.

[11] BOUDIN F. Unsupervised keyphrase extraction with multipartite graphs[C]. Proceedings of the 2018 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, 2: 667－672.

[12] PETERS M, NEUMANN M, IYYER M, et al. Deep Contextualized Word Representations[C]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018: 2227－2237.

[13] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. North American Chapter of the Association for Computational Linguistics (NAACL), 2018: 4171－86.

[14] YANG Z, DAI Z, YANG Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding[J]. Advances in neural information processing systems, 2019, 32.

[15] ALZAIDY R, CARAGEA C, GILES C L. Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents[C]. proceedings of the The World Wide Web Conference, 2019: 2551-2557.

[16] MAHATA D, KURIAKOSE J, SHAH R, et al. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings[C]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018: 634-639.

[17] BENNANI-SMIRES K, MUSAT C, HOSSMANN A, et al. Simple unsupervised keyphrase extraction using sentence embeddings[C]. Proceedings of the 22nd Conference on Computational Natural Language Learning, 2018: 221－229.

[18] SUN Y, QIU H, ZHENG Y, et al. SIFRank: A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model[J]. IEEE Access, 2020, 8: 10896-10906.

[19] ARORA S, LIANG Y, MA T. A simple but tough-to-beat baseline for sentence embeddings[C]. proceedings of the International conference on learning representations, 2017.

[20] WANG S, YAO X. Diversity analysis on imbalanced data sets by using ensemble models[C]. 2009 IEEE symposium on computational intelligence and data mining. IEEE, 2009: 324-331.

[21] Du Z, Qian Y, Liu X, et al. Glm: General language model pretraining with autoregressive blank infilling[J]. arXiv preprint arXiv: 2103.10360, 2021.

[22] 卫梅特, 任洪敏. 基于特征优选的软件缺陷预测集成学习方法[J]. 计算机仿真, 2023, 40(07): 331-336.

[23] 徐继伟, 杨云. 集成学习方法: 研究综述[J]. 云南大学学报(自然科学版), 2018, 40(06): 1082-1092.

[24] 杨婷, 莫若玉, 张秀娟等. 轻量级缓存策略的关系型数据库全文搜索加强与扩展[J]. 计算机应用, 2023, 43(08): 2431-2438.

[25] SUN Z, TANG J, DU P, et al. DivGraphPointer: A Graph Pointer Network for Extracting Diverse Keyphrases[C]. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 755–64.

[26] KIM S N, MEDELYAN O, KAN M-Y, et al. Automatic keyphrase extraction from scientific articles[J]. Language resources and evaluation, 2013, 47: 723-42.

[27] CARAGEA C, BULGAROV F, GODEA A, et al. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1435–1446.

[28] MENG R, ZHAO S, HAN S, et al. Deep Keyphrase Generation[J]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017: 582–92.

[29] PAPAGIANNOPOULOU E, TSOUMAKAS G. A review of keyphrase extraction[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2020, 10(2): e1339.

[30] HULTH A. Improved Automatic Keyword Extraction Given More Linguistic Knowledge[C]. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003: 216–23.

[31] AUGENSTEIN I, DAS M, RIEDEL S, et al. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications [J]. Computing Research Repository, 2017, abs/1704.02853: 546-55.

[32] NGUYEN T D, KAN M Y. Keyphrase Extraction in Scientific Publications [C]. Proceedings of the 10th international conference on Asian

digital libraries: looking back 10 years and forging new frontiers, 2007, 4822: 317 - 326.

[33] 阎红灿, 李铂初, 谷建涛. 一种基于共现关键词的 TextRank 文摘自动生成算法[J]. 计算机工程与科学, 2023, 45(11): 2060-2069.

[34] 王超, 刘奕群, 马少平. 搜索引擎点击模型综述[J]. 智能系统学报, 2016, 11(06): 711-718.

[35] Zhang M, Li X, Yue S, et al. An empirical study of TextRank for keyword extraction[J]. IEEE access, 2020, 8:178849-178858

[36] KIM S N, MEDELYAN O, KAN M Y, et al. SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. Proc. 5th Int. Workshop Semantic Eval. 21-26.

[37] WAN X, XIAO J. Single Document Keyphrase Extraction Using Neighborhood Knowledge[C]. AAAI. 2008, 8: 855-860.

[38] KRAPIVIN M, AUTAEU A, MARCHESE M. Large Dataset for Keyphrases Extraction[J]. 2009: 271 - 278.

[39] ZHANG L, CHEN Q, WANG W, et al. MDERank: A Masked Document Embedding Rank Approach for Unsupervised Keyphrase Extraction[J]. Findings of the Association for Computational Linguistics: ACL 2022, 2021: 396 - 409.

[40] KONG A, ZHAO S, CHEN H, et al. PromptRank: Unsupervised Keyphrase Extraction Using Prompt[J]. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, 1: 9788 - 801.

致　　谢

研究生生涯即将结束，我也即将告别我在江西师范大学三年的生活。在这段宝贵的时间里，江西师范大学见证了我在为人、学术以及生活方面的成长。

首先，我最需要感谢的人就是我的导师揭安全老师和李宏伟老师。两位导师不论是在学术研究上对我的方向和研究方式进行引导，还是在平时的生活过程中，教会我很多人生的经验。李宏伟老师在学术上有着广泛的涉猎以及极高的见解，并且温文尔雅的举止时常让我如沐春风，虽然工作非常繁忙，但是他们从来没有忽视过我的学习，在学习上我们定期交流，并且耐心地为我解答疑惑。除此之外，不管是在学术论文的撰写还是在这次的论文写作中，两位导师通过自己丰富的科研经验在论文写作中不断地给我提出修改意见，给予了我很大的帮助，同时导师严谨的钻研精神和认真负责的态度，给我留下了深刻的印象，鼓励着我不断地进步和前行。

其次，我还要感谢我宿舍和实验室的小伙伴们以及我的同学们，这三年的研究生生活，感谢他们让我的研究生生活有了家的感觉。

另外，我要感谢我的家人，是他们在背后一直默默的支持着我，让我能没有顾虑地投入到学习中去。

最后我要感谢所有帮助过我的人，也很感谢在百忙之中参与到我的论文评阅和答辩工作中的每一位老师，非常感谢你们。

在读期间公开发表论文（著）及科研情况

1. 硕士期间获奖情况

[1] 硕士研究生省学业奖学金，江西师范大学，2023 年

[2] 硕士研究生三好学生，江西师范大学，2023 年

[3] 硕士研究生数学建模竞赛一等奖，江西省学位与研究生教育学会，2023 年

[4] 第十三届蓝桥杯大赛研究生组二等奖，蓝桥杯大赛组委会，2022 年

[5] 第五届中国虚拟现实大赛 CCVR2022 三等奖，中国仿真学会，2022 年

[6] 第三届全国大学生算法设计与编程挑战赛铜奖，全国大学生算法设计与编程挑战赛组织委员会，2022 年

2. 硕士期间科研活动参与的基金项目

[1] 国家自然科学基金项目（61966019）:动态邻域空间学习的群智能算法及其在节能覆盖中的应用研究，2020.01-2023.12.(参与)

[2] 江西省自然科学基金项目（20224BAB202018）:基于智能移动交互设备的三维数字模型和图像快速绘制与构建方法研究，2023.01-2025.12.（参与）

3. 硕士期间技术实践

[1] 项目部署上线与维护:基于无监督关键词提取算法的聚合搜索系统，2024.02-2024.03