2　产品服务与创意

2.1　产品项目介绍

2.1.1　产品名称

AI 云学习 —— 一款基于 Spark 构建知识图谱的人工智能学习工具。

2.1.2　产品研发团队

牛头冲八仙下海创业团队。

2.1.3　产品系统组成

（1）基于 PathFinder 算法的主从分布式 Python 爬虫元数据获取子系统。

（2）基于 Spark 平台的元数据预处理子系统。

（3）基于 Jiagu 模型的知识关系抽取子系统。

（4）基于 PHP 与 MySQL 的关键词检索子系统。

（5）基于 amChart 4 的图谱渲染与展示子系统。

（6）云端服务器。

（7）Web 应用。

（8）手机 APP。

2.1.4　产品功能说明

（1）对用户输入的不在数据库中的关键词进行预检索处理，即以当前关键词作为主从分布式 Python 爬虫元数据获取子系统的输入来获取相应的元数据。

（2）对分布式 Python 爬虫元数据获取子系统所得到的元数据进行文档去重、清洗、网页标签过滤、敏感词筛除与文本自组织标记。

（3）对 Spark 平台的元数据预处理子系统所得到的预处理数据进行自然语言模型训

练并提取相应的知识关系。

（4）对 Jiagu 模型的知识关系抽取子系统所生成的三元组数据进行格式重调、MySQL 存储并给出用户使用与自定义知识图谱所需的“增删查改”功能。

（5）对 PHP 与 MySQL 的关键词检索子系统所返回的 json 格式数据进行力导向图渲染与展示。

（6）云服务器是部署知识图谱后端的主要平台，负责对用户数据的检索、元数据获取、文本预处理、知识关系抽取与知识图谱展示等一系列功能。

（7）Web 应用是供 PC 端与手机端用户实时检索所需知识图谱的前端平台，免去了安装专门应用的烦琐操作。

（8）手机 APP 分为 Android 与 iOS 版本，分别供 Android 用户和 iOS 用户安装使用，手机 APP 增强用户使用知识图谱的稳定性与安全性。

2.1.5 产品的技术领先性

（1）产品核心技术：

a) 借助主从分布式 Python 爬虫实现 PathFinder 算法。
b) 基于大数据处理平台 Spark 的文本预处理系统。
c) 基于国产开源自然语言工具 Jiagu 实现高效、快捷的知识关系抽取。
d) 基于数据仓库平台 hive 实现微秒级的数据库管理操作。
e) 基于 amChart 4 完成艺术级的图谱渲染效果与知识节点展示。
f) 云服务器实现了对用户输入数据的全自动元数据流式获取、文本预处理、知识抽取、数据库存储与图谱节点反馈。
g) Android、iOS 与 Web 应用提供了多种知识图谱访问操作。

（2）产品技术、应用与运营模式创新：

1) 技术创新：

a) 借助主从分布式 Python 爬虫实现 PathFinder 算法

本项目拟构建人工智能知识的知识图谱，但目前并不存在有关内容的开源数据库或信息源，因此，利用分布式爬虫获取内容是唯一有效的方法。然而，传统的分布式爬虫虽然可以有选择地访问网页与相关链接并获取所需信息，但获取内容仍含有一定的无价值数据。在大数据环境下，分布式架构的分布式爬虫比单机多核的串行爬虫具有更高的效率与

更新速度。爬取相关度更高的内容也是一个值得考虑的问题，为了解决这个问题，我们借助主从分布式爬虫实现 PathFinder 算法，根据相关度阈值获取内容。

理论计算与实验数据证明，本项目采用的 Python 爬虫方法在显著地提高了数据获取效率的同时，还极大地保证了数据的相关度。

b) 基于大数据处理平台 Spark 的文本预处理系统

文本预处理是将文本表示成一组特征项。将每个词作为文本的特征项是目前常用的处理方法，针对本项目的文本特征项主要是专有名词与术语，本项目在 Spark 平台下利用 Word 分词，实现分布式工作。Word 分词是用 Java 实现的，实现了多种分词算法，并利用 ngram 模型消除歧义，能有效对数量词、专有名词与人名进行识别。分词所得到词语组，主要用于信息联想，也就是在构建完成的知识图谱中检索与給定词语有关联的三元组。

c) 基于数据仓库平台 hive 实现微秒级的数据库管理操作

hive 是一个基于 Hadoop 的数据仓库平台，通过 hive 我们可以快速地对存储在数据库中数据进行抽取、加载与转换（Extract，Transform，Load，ETL）等操作。

2) 应用创新：

a) 基于国产开源自然语言工具 Jiagu 实现高效、快捷的知识关系抽取

Jiagu 模型是一个国产的开源自然语言处理工具，以 BiLSTM 等模型为基础，使用大规模语料训练而成。Jiagu 模型提供中文分词、词性标注、命名实体识别、情感分析、知识图谱关系抽取、关键词抽取、文本摘要、新词发现、情感分析、文本聚类等常用自然语言处理功能，API 丰富，且操作便捷、稳定性高。本文选择 Jiagu 模型作为知识抽取的工具，取得了十分理想的效果。

b) 基于 amChart 4 完成艺术级的图谱渲染效果与知识节点展示

amCharts 4 是一个基于 TypeScript 开源的可视化框架，具有图表种类丰富、图形效果炫丽、动画或静态呈现、与平台无关等特点，适用于各个行业的可视化需求场景，因此本文将其作为知识图谱的可视化工具。本文使用 HTML/CSS/JavaScript 设计页面元素及基本布局，并采用力导向图作为图谱的呈现形式。当用户在搜索框键入查询关键词时，通过 GET 请求关键字，后台通过 PHP 查询数据库并返回请求的数据。前端得到请求的数据后，通过 JavaScript 进行预处理并借助 amCharts 进行可视化展示。

c) 云服务器实现了对用户输入数据的全自动元数据流式获取、文本预处理、知识抽

取、数据库存储与图谱节点反馈

为了提高产品的可用性，本项目所设计的知识图谱除了提供对本地存储的知识节点查询外，还能以用户输入的关键词进行图谱拓展，概而言之就是：当输入关键词不匹配数据库内的任何结果时，将其作为 Python 爬虫的输入关键字爬取相关文本，并将所得文本按照既定的技术流程操作，得到与新关键词有关的知识图谱。这一方式使知识图谱的进一步拓展成为了可能。

3) 模式创新：

a) Android、iOS 与 Web 应用提供了多种知识图谱访问操作

本产品提供了多种操作终端，最大化地覆盖了各个平台的用户，以期为产品盈利带来更为广阔的使用人群，这增大了产品的被动测试与 BUG 反馈案例，为后期产品优化提供了绝佳的参考。

b) 产品提供免费与付费双重个性化服务

本产品面向广大用户提供日均一定数量的免费知识图谱检索服务的同时，设置了付费检索服务：付费用户凭支付一定量的费用享受次数更多、自定义操作更完善的知识节点检索服务。付费服务是本产品盈利的重要来源。

2.2 产品系统总体技术方案

随着 Web 技术飞跃式发展，互联网先后经历了三个时代，它们分别具有不同的特征：文档互联的“Web 1.0”时代，数据互联为特征的“Web 2.0”时代以及当下正在发展的知识互联的崭新“Web 3.0”时代。知识互联为人们的学习与交流提供了极大便利，人类的知识交互达到了历史的新高峰。然而，互联网上的知识来源复杂、良莠不一，零散混乱、体系松散，尤其是在大数据的时代背景下，这給内容的筛选、组织与评价带来了极大挑战。知识图谱（Knowledge Graph）是人工智能（Artificial Intelligence，简称 AI）领域一项重要的技术分支，具有强大的语义处理能力与开放互联能力。值得注意的是，目前国内尚无针对人工智能这一领域的知识图谱工具。人工智能正处于快速发展阶段，了解、学习、掌握有关知识与技术是学生、工程师、科研人员所面临的一大挑战，优秀的知识架构可以帮助学习者达到事半功倍的效果。

目前，已经有许多大型知识图谱被构建出来，如 DBpedia、Freebase 等，然而，当前的知识图谱工具普遍存在以下问题：1）通用知识图谱工具涉面较广，但知识冗余混乱、

组织零散、系统性差，不利于用户的专业学习；2）垂直知识图谱工具种类少，成熟的应用仅限于某些领域，在一些具有较大应用需求的领域未获重视，前景广阔。

综上所述，本项目的目的是构建一个面向学习者尤其是本科生的人工智能领域的垂直知识图谱，意义在于通过 Spark 完成人工智能知识的重整，实现了一个学习者尤其是本科生适用的知识图谱工具。人工智能领域繁多，为消减技术流程的复杂度，我们选取机器学习（Machine Learning，ML）、自然语言处理（Natural Language Processing，NLP）与机器视觉（Machine Vision，MV）等三个领域作为代表。构建知识图谱的一般技术流程如图 2.2.1 所示。

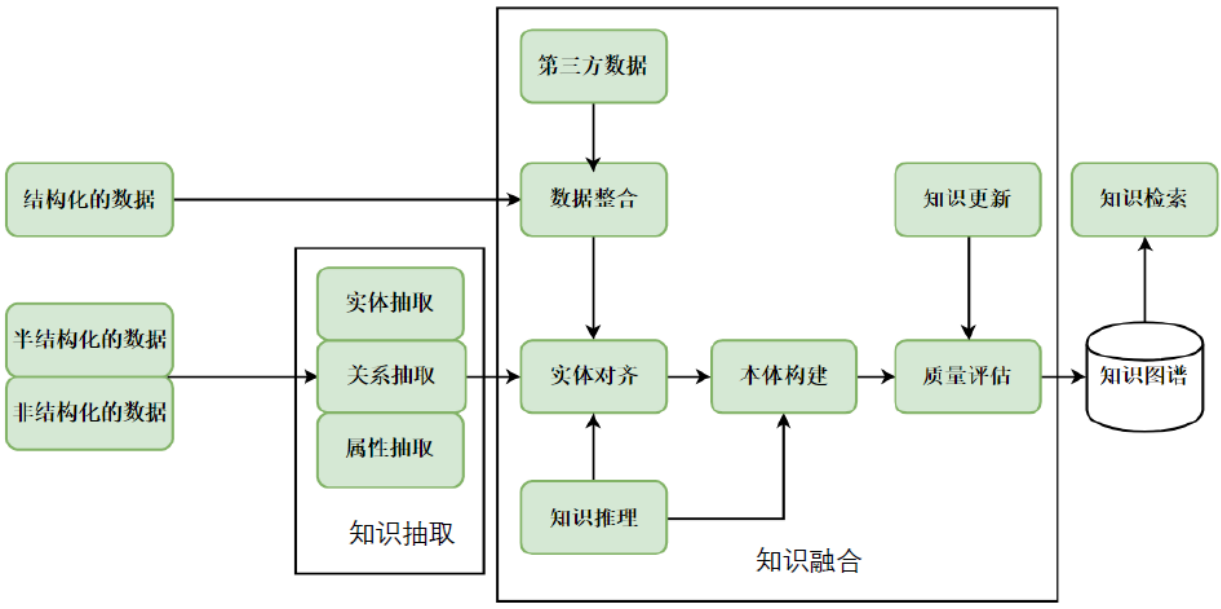

图 2.1.5.1 知识图谱构建流程

（一）数据类型

构建知识图谱的元数据有三种常见的类型：结构化数据、半结构化数据与非结构化数据。

结构化的数据是指可以使用关系型数据库表示和存储，表现为二维形式的数据。一般特点是：数据以行为单位，一行数据表示一个实体的信息，每一行数据的属性是相同的。常见的结构化数据为数据库。

半结构化数据是结构化数据的一种形式，它并不符合关系型数据库或其他数据表的

形式关联起来的数据模型结构，但包含相关标记，用来分隔语义元素以及对记录和字段进行分层。因此，它也被称为自描述的结构。对于半结构化数据，属于同一类实体可以有不同的属性，即使他们被组合在一起，这些属性的顺序并不重要。常见的半结构数据有XML 和 JSON 格式。

非结构化数据是没有固定结构的数据。各种文档、图片、视频/音频等都属于非结构化数据。对于这类数据，我们一般直接整体进行存储，而且一般存储为二进制的数据格式。

（二）知识抽取

知识抽取是指把蕴含于信息源中的知识经过识别、理解、筛选、归纳等过程抽取出来，存储形成知识元库。知识抽取是构建知识图谱的首个关键步骤与基础，直接影响了后续工作的成效与最终构建所得图谱的质量。知识图谱构建中知识抽取分为：实体抽取、关系抽取与属性抽取。

实体抽取又称命名实体识别，包括实体的检测（find）：识别命名实体的文本范围，实体的分类（classify）：分类为预定义的类别，学术上所涉及一般包含三大类，实体类、时间类、数字类和 7 个小类，如人、地名、时间、组织、日期、货币、百分比等。

关系抽取主要负责从文本中识别出实体，抽取实体间的语义关系，在知识图谱构建中一般以三元组的形式来表征。

属性抽取的任务为识别实体的属性名与识别实体的属性值，而属性值结构一般是不确定的。

（三）知识融合

知识融合，即合并两个知识图谱(实体及其对应关系)，其基本问题是研究怎样将来自多个来源的关于同一个实体或概念的描述信息融合起来。由于知识图谱中的知识来源广泛，存在知识质量良莠不齐、来自不同数据源的知识重复、知识间的关联不够明确等问题，所以需要进行知识的融合。知识融合是高层次的知识组织，使来自不同的知识源的知识在同一框架规范下进行异构数据整合、消歧、加工、推理验证、更新等步骤，达到数据、信息、方法、经验以及人的思想的融合，形成高质量的知识库。

经过上述步骤后，方能得到可供进行知识检索的有效知识图谱。接下来详细介绍本产品构建知识图谱的技术流程。

2.2.1 数据来源

1) 爬取工具的选择

本文选择 CSDN 与博客园作为主要的元数据（Metadata）获取平台，因其主要数据采用网页来展现，所以本文选择 Python 作为爬取工具。Python 不但用于抓取网页文档的接口简洁，同时其访问网页文档的 API 也相当完整。

值得一提的是，抓取网页有时需将爬虫（Crawler）程序伪装成普通的浏览器。因为许多网站都采取了防爬措施，单纯的爬取操作极容易被网站检测出来并封杀。Python 提供了许多鲁棒的第三方包如 requests、mechanize、selenium，可以帮助爬虫轻松地越过网站的防爬策略。

在抓取了网页之后，仍需进一步的处理，如过滤 html 标签，提取文本等，而 python 的 beautifulsoap 库等使编写非常简洁的代码即可完成大部分文档的处理成为可能。

2) 提高爬取效率的方法

传统的网络爬虫是运行在本地，稍优化的策略是采取“单机多核”的方式。为了更有效地解决爬取效率过低的问题，同时结合实际的实验条件，本文采用主从分布式爬虫（Master-Slave Distributed Crawler）[1]，并在其上实现 PathFinder 算法，据所列关键词的相关度按阈值排序获取特定内容。

本文将一台阿里云服务器作为 master 服务器，用于分发所需爬取内容的 URL，同时维护存储在 redis 中待爬取 URL 的列表。由三台本地的笔记本电脑组成 slave 服务器组，用于对各自从 master 服务器所获得的 URL 执行网页爬取任务；若 slave 在爬取过程中遇到新的 URL，一律将其返回 master 服务器由 master 解析处理，slave 服务器间不进行通信。本文所用 master 服务器与 slave 服务器组的性能配置如表 2.2.1.1 所示，主从分布式爬虫的逻辑结构如图 2.2.1.1 所示，爬虫的类图结构如图 2.2.1.2 所示。

表 2.2.1.1 master 服务器与 slave 服务器组性能配置

Server	Processor	RAM/GB	Storage/GB	CPU core(s)
master	Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz	2	40	1
slave 1	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz	16	128(SSD) + 1024	4
slave 2	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz	16	128(SSD) + 1024	4
slave 3	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz	16	128(SSD) + 1024	6

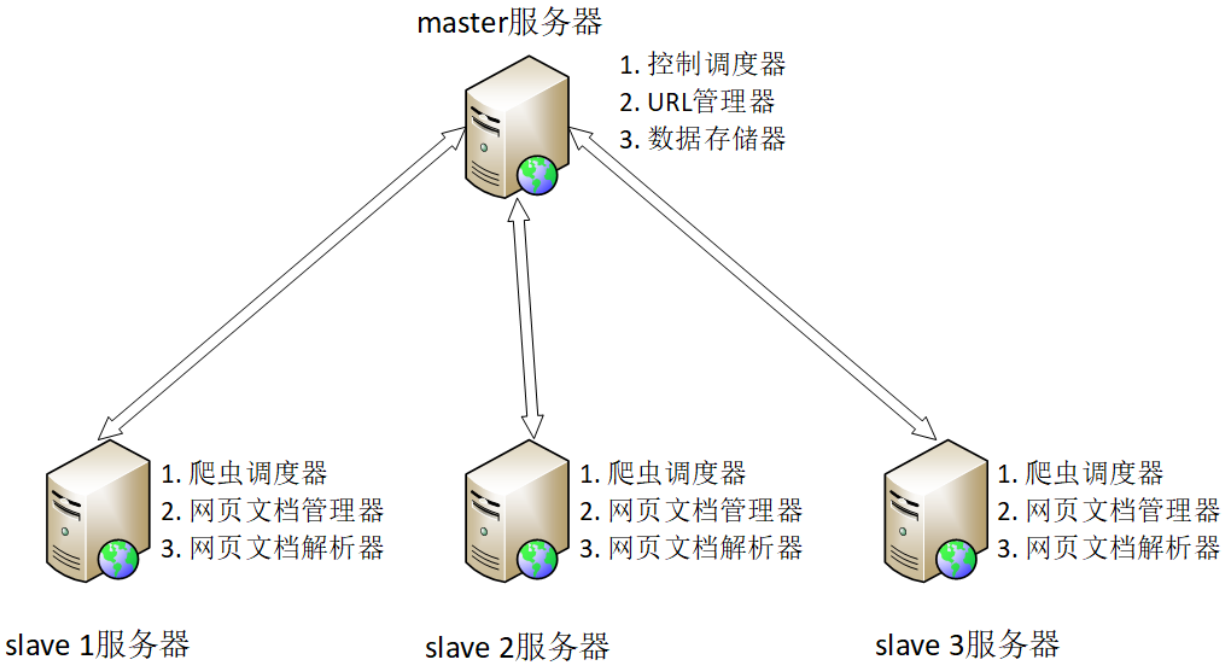

图 2.2.1.1 主从式分布爬虫逻辑结构

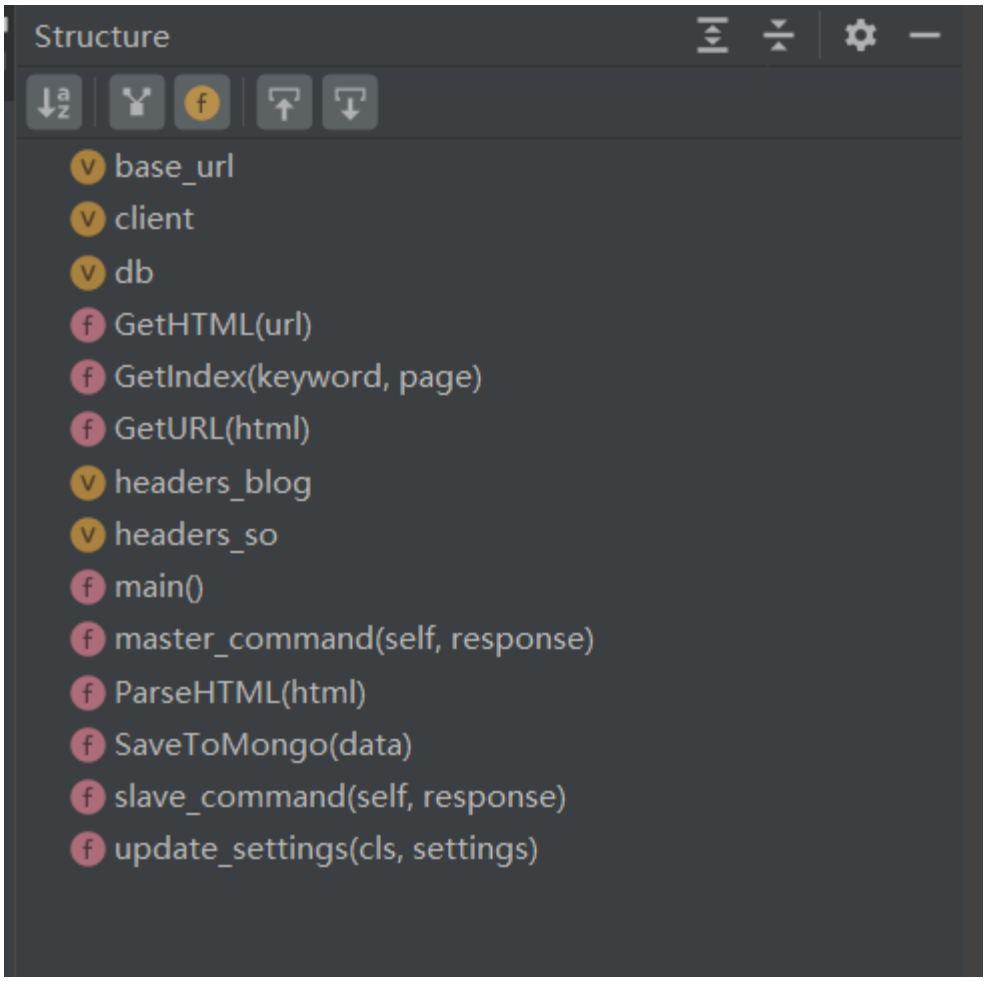

图 2.2.1.2 爬虫程序的类图结构

此外，为了防止网站服务器锁定爬虫的 IP，本文所使用的爬虫程序对爬取频率进行了限制，以及使用代理 IP 池。

2.2.2 Spark 与 Jiagu 模型

1) Spark 与 hive 平台

Spark[2]是基于内存计算的大数据并行计算框架，因为它基于内存计算，所以提高了在大数据环境下数据处理的实时性，同时保证了高容错性和高可伸缩性，允许用户将 Spark 部署在大量廉价硬件之上，形成集群。hive[3]是一个基于 Hadoop 的数据仓库平台，通过 hive 我们可以快速地对存储在数据库中数据进行抽取、加载与转换（Extract，

Transform，Load，ETL）等操作。hive 定义了一个类似于 SQL 的查询语言：HQL，能够将用户编写的查询语句转化为相应的 MapReduce 程序并基于 Hadoop 执行。需要注意的是，hive 本身并不存储数据，因而用户需要选择一个传统的数据库进行数据存储，基于可操作性与成本等角度考虑，本项目采用 MySQL。

本项目将使用 Spark 平台的相关工具进行数据预处理。

2) 数据预处理

元数据杂源异质，散乱冗余，并且由于网页文本本身的结构导致数据中存在大量标签，无法直接用于下一步操作。因此本文借助 Spark 平台快速的数据处理能力及 hive 对数据库高效的 ETL 操作，对文本进行预处理。

首先，在 spark-shell 上将数据成功加载到 hive 中，为后续存取提供了数据来源。其次，在 hive 上创建了数据库，在 spark-shell 上依次将爬虫爬取的 json 文件导入成表。而后，在 IDEA 上编程对数据去重，这里主要使用了 Spark 的几个 API，如：duplicate、filter、regexp_replace、regexp_extract 等。完成数据的存储、去重和标签过滤后，借助于 github 上开源的敏感词汇库[4]，对表数据进行敏感词（Sensitive Word）过滤，以此得到更干净的数据。本文所用部分 spark-shell 处理命令如图 2.2.2.1，数据预处理的程序类图如图 2.2.2.2 所示，预处理后的部分数据如图 2.2.2.3 所示。

```
scala> val dataDF1 =
spark.read.format("json").load("file:///home/hadoop001/hadoop/data/Spider-
Data/cnblog_computer_version.json")

scala> dataDF1.select(dataDF1.col("author"),
dataDF1.col("content"),dataDF1.col("date"),
dataDF1.col("title")).write.saveAsTable("dachuangppreprocessingdata.cnblog_computer
_version")
```

图 2.2.2.1 spark-shell 处理命令

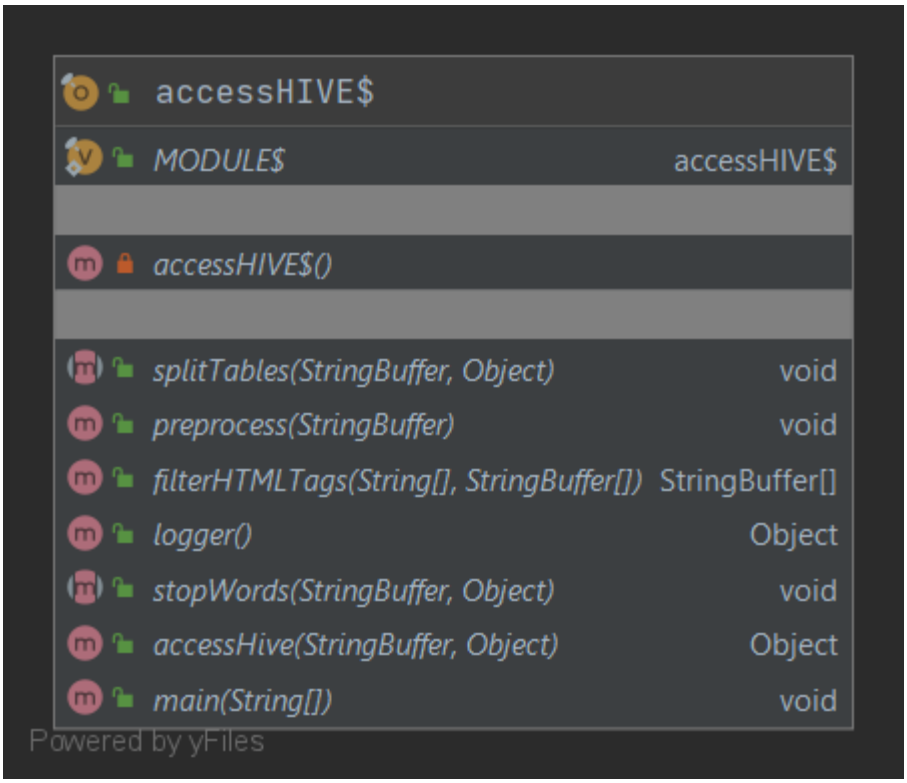

图 2.2.2.2 数据预处理程序的类图

```
提到 人工智能 ， 大多数 人 的 第一 反应 就是 距离 我们 太远 了 。 智能 机器人 、 无人驾驶 ， 这些 好像 都 是 未来式 。 我们
比如 ， 应用 最广 的 美颜 自拍 ， 更 准确 的 说 ， 是 人像 处理 。
现在 的 人像 软件 之所以 能 帮助 人们 从 繁琐 的 PS 中 解放出来 ， 就是 因为 利用 了 大量 计算机视觉 技术 ， 像是 人脸 定
今天 就 以 天天 P 图为例 ， 看看 是 什么 让 他们 成 了 一款 AI 软件 。
AI 赋能 ， 让 手机 如何 读懂 你 的 脸
人像 处理软件 之所以 能 成为 AI 产品 ， 是因为 有 了 大量 图片 数据 ， 尤其 是 人脸 数据 的 累积 。 而 通过 大量 图片 数据
以 天天 P 图 的 自动 美颜 功能 为例 ， 软件 之所以 能 放大 眼睛 、 添加 贴图 动效 ， 是因为 准确 的 找到 了 人脸 和 五官 的
在 每 帧 图像 中 准确 的 找到 人脸 和 五官 后 ， 就 可以 “ 加 特效 ” 了 — — 增加 美妆 、 萌宠 贴图 ， 自然美 妆 。
除了 对 人脸 的 识别 和 处理 ， 为了 给 用户 提供 更 多 丰富 智能 的 玩法 ， P 图 团队 还 联合 优图 团队 对 视频流 进行 了
背景 分割 也 是 另 一项 基于 AI 的 创造性 玩法 ， 通过 深度 优化 加速 后 的 神经网络 ， 使得 P 图 可以 在 移动 端 实现 对
打造 图像处理 云 ， 美颜 AI
能 做到 的 不仅仅 是 变脸
美颜 AI 能 做到 的 不仅仅 是 变脸 。
在 大多数 人 的 印象 中 ， 天天 P 图 这 类人 像 处理软件 即使 有 AI 技术 ， 基本 也 是 应用 于 自己 的 产品 之中 ， 缺乏
细心 的 人会 发现 ， 军装 H5 并非 在 终端 上 进行 运算 ， 而是 通过 H5 上 传到 云端 处理 。 基于 云端 的 人脸识别 ， 五官
除了 家喻户晓 的 军装 照 ， 天天 P 图 最近 推出 的 萌偶 功能 也 利用 了 AI 图像处理 云 。
通过 在 云端 的 神经网络 ， 找到 与 用户 五官 相似 度 最高 的 卡通 素材 。 建立 标准 人脸 和 标准 卡通 人脸 间 的 映射 关系
这些 能够 提供 丰富 玩法 的 AI 图像处理 云 ， 也 解决 了 深度 神经网络 模型 可能 过大 ， 无法 在 终端 运行 的 问题 。 天天
强大 的 分布式 部署 能力 降低 了 客户端 的 门槛 ， 使得 算法 可以 配适 各种 环境 ： 手机 、 电脑 、 电视 、 App 、 H5 … …
作为 用户 可能 很难 明确 感受 到 图像处理 云 的 存在 ， 但 这项 能力 却 为 天天 P 图 打开 了 更 多 依靠 AI 创造 营收 的 路
从 隐性 到 显性 ， 人像 处理 AI
```

图 2.2.2.3 预处理后的部分数据

3) Jiagu 模型

Jiagu 模型[5]是一个国产的开源自然语言处理工具，以 BiLSTM 等模型为基础，使用大规模语料训练而成。Jiagu 模型提供中文分词、词性标注、命名实体识别、情感分析、知识图谱关系抽取、关键词抽取、文本摘要、新词发现、情感分析、文本聚类等常用自然语言处理功能，API 丰富，且操作便捷、稳定性高。本文选择 Jiagu 模型作为知识抽取的工具，取得了十分理想的效果。

4) 知识抽取

在知识图谱中，知识一般以三元组(p, r, q)的形式来表示，其中 p 与 q 分别代表前后两个实体，r 代表前后实体之间的关系[6]。显然三元组是构建知识图谱的重要基础，三元组中实体间的关系是否准确、完整等也是知识图谱的构建成功与否的重要判据。

本文采用 BIO 方式[7]对待训练文本进行实体命名标记，每行一个字符，并按 19:5 的比例分别设置训练数据与验证数据，且为测试训练所得模型的准确程度设置了较训练数据 75%的测试数据，详细信息如表 2.2.2.1 所示。在分别调节学习率（Learning Rate）、迭代次数（Iterations）、阻尼系数（Damping Coefficient）等参数后对标记文本进行训练，参数详情如表 2.2.2.2 所示。实验结果用 held-out 方法[8]进行评估，即统计知识图谱中已有的实体被 Jiagu 模型检测出的数量，正确的实体被排序靠前的数量愈多，则在准确率/召回率曲线上，随着召回率（Recall Rate）的增长准确率（Accuracy Rating）就下降得越慢，也即知识抽取的质量愈高。实验结果的准确率/召回率曲线如图 2.2.2.4 所示，所得部分三元组如图 2.2.2.5 所示。

表 2.2.2.1 数据集的统计信息

数据集	关系数量	语料行数
训练集	10	2435796
验证集		634547
测试集		1849620

表 2.2.2.2 所用训练参数

Learning rate	Iterations	Damping coefficient
0.001	50000	0.85

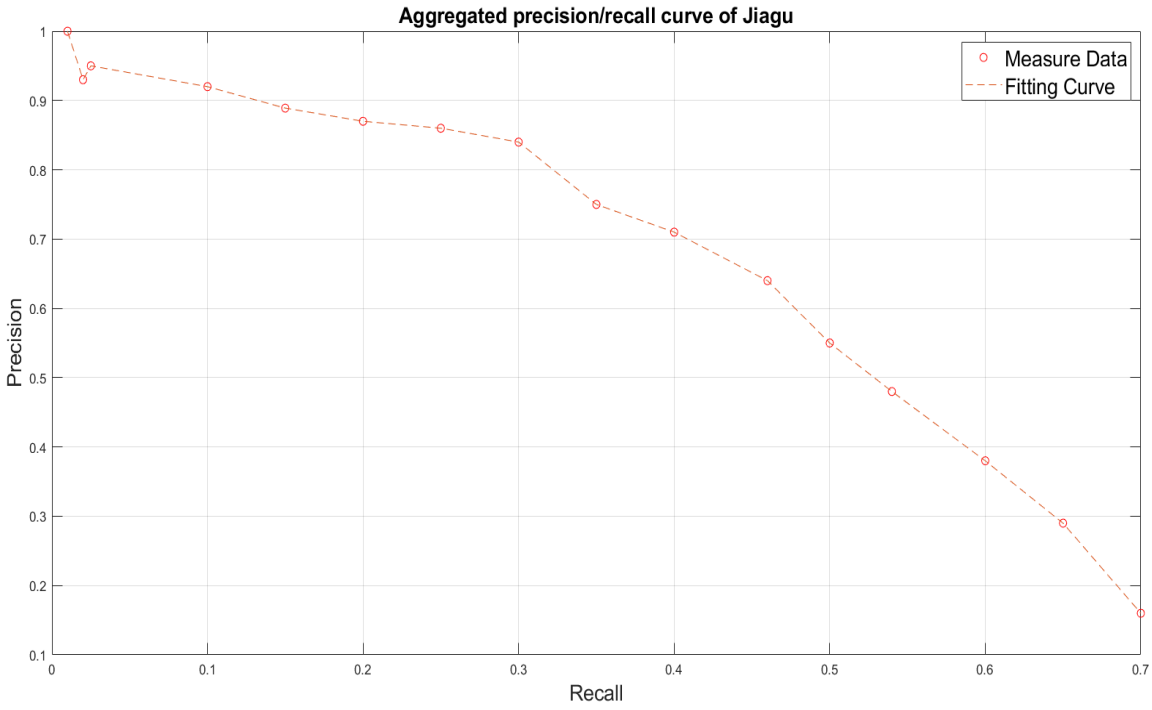

图 2.2.2.4 准确率/召回率

```
逻辑回归,优势,处理非线性效应
逻辑回归,缺点,仅用于二进制分类
随机森林,优势,防止过拟合
随机森林,用于1,回归
随机森林,用于2,分类
随机森林,缺点1,容易生长
随机森林,缺点2,随机子集高
评价矩阵,术语1,真阳性(TP)
评价矩阵,术语2,真阴性(TN)
评价矩阵,术语3,假阳性(FP)(I型错误)
评价矩阵,术语4,假阴性(FN)(II型错误)
特征选择,也称为1,变量选择
特征选择,也称为2,属性选择
特征选择,也称为3,变量子集
特征选择,选择,最佳相关特征
特征选择,帮助1,简化ML模型
特征选择,帮助2,提高ML模型的准确性
特征选择,有助于,更快地训练
特征选择,防止,过拟合
```

图 2.2.2.5 三元组数据

2.2.3 知识图谱的可视化

1) 三元组的转化

本项目所选可视化工具为基于 TypeScript 开源的可视化框架 amCharts 4，其与 TypeScript、Angular、React、Vue 和纯 JavaScript(ES6)进行了原生集成[9]。由于用户通过某个关键字请求实体的三元组信息时，其数据量可能是非常大的。此外，amCharts 4 要求数据以特定的 json 格式存储，显然 2.2.3 节所得的三元组无法直接用于可视化（Visualization）。出于存取效率、数据可拓展性等因素考虑，本文将三元组数据预先导入 MySQL 数据库，当前端发出数据请求时，通过 PHP 编程实现从服务器端查找相应的原始三元组数据并使用相应 API 转换为 json 格式返回给前端。前端在接收到 PHP 返回的原始三元组数据后，需要对原始三元组数据进行预处理，将原始的 json 数据转化为 amCharts 可识别的特定格式 json 数组，并最终作为 amCharts 的数据源加载，渲染（Render）到指定的 SVG 画布上，最终形成可操作的力导向图谱。具体交互的流程如图 2.2.3.1 所示。

2) 图谱可视化

amCharts 4 是一个基于 TypeScript 开源的可视化框架，具有图表种类丰富、图形效果炫丽、动画或静态呈现、与平台无关等特点，适用于各个行业的可视化需求场景，因此本文将其作为知识图谱的可视化工具。本文使用 HTML/CSS/JavaScript 设计页面元素及基本布局，并采用力导向图作为图谱的呈现形式。当用户在搜索框键入查询关键词时，通过 GET 请求关键字，后台通过 PHP 查询数据库并返回请求的数据。前端得到请求的数据后，通过 JavaScript 进行预处理并借助 amCharts 进行可视化展示。

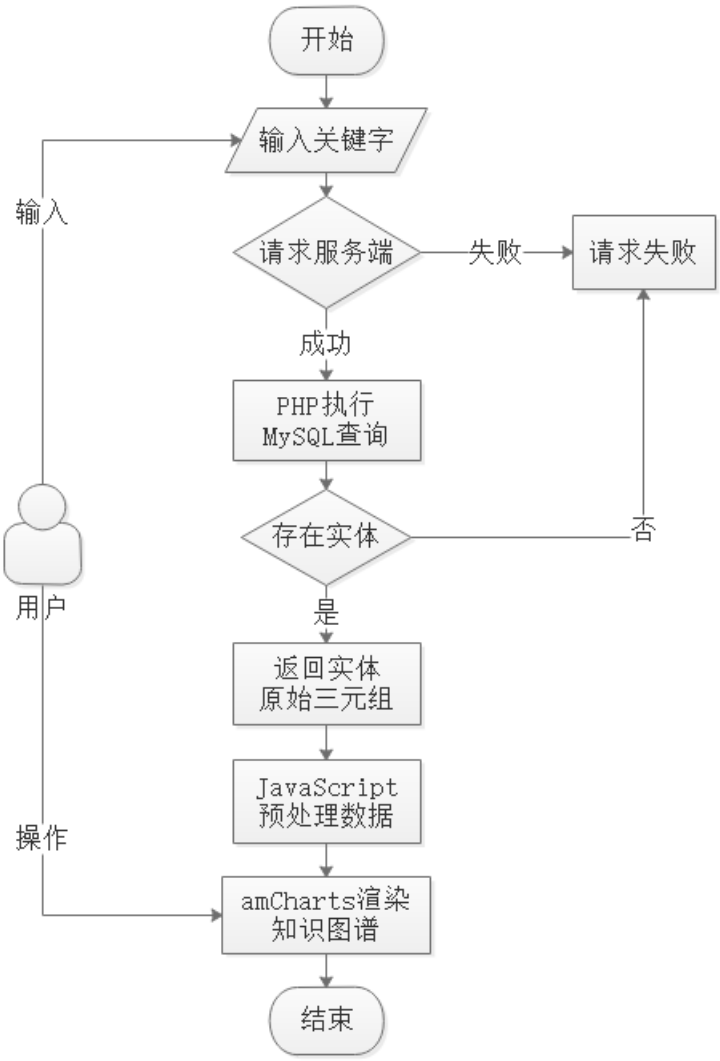

图 2.2.3.1 知识图谱可视化流程图

2.3 产品硬件配置、软件截图与说明

2.3.1 产品硬件配置

云端服务器配置信息，即产品硬件配置信息如图 2.3.1 所示。

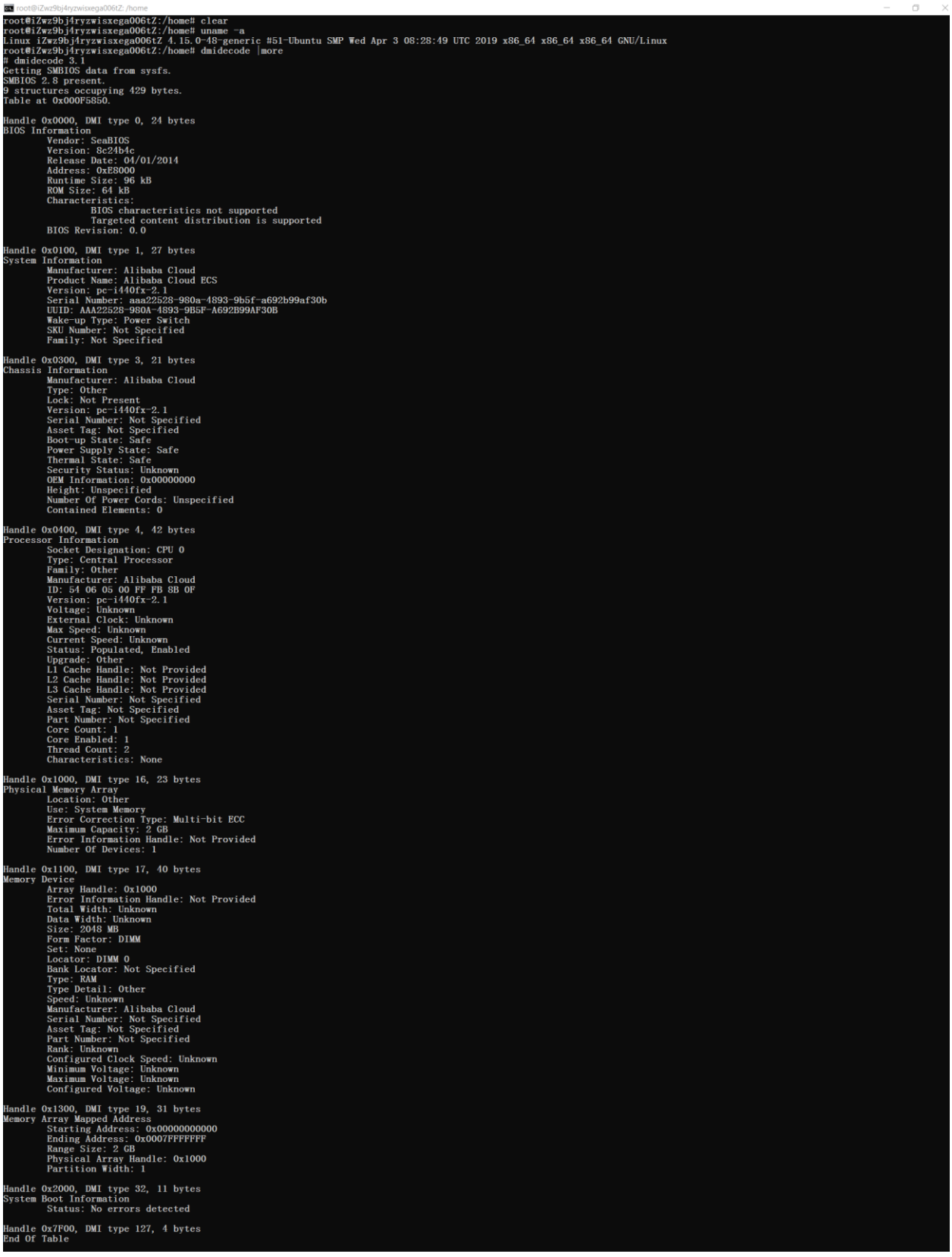

图 2.3.1.1 云端服务器硬件配置信息

2.3.2 软件截图

Web 应用运行状况，如图 2.3.2.1 至图 2.3.2.6 所示。

图 2.3.2.1 运行截图-1（检索关键词：机器人）

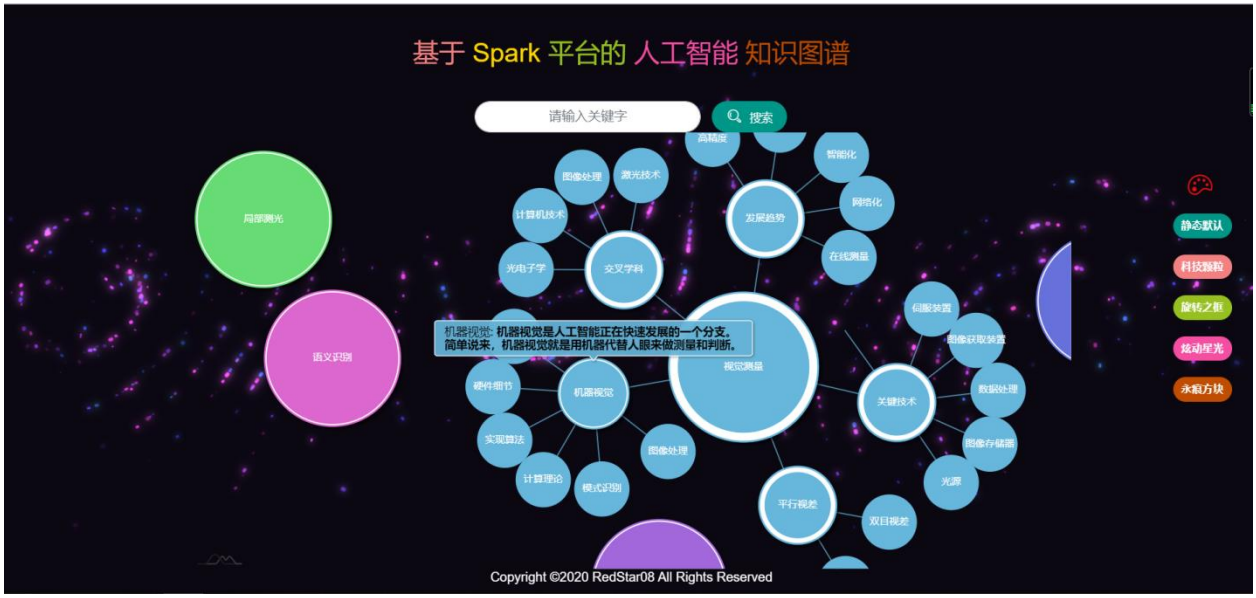

图 2.3.2.2 运行截图-2（检索关键词：视觉测量）

图 2.3.2.3 运行截图-3（检索关键词：人工智能）

图 2.3.2.4 运行截图-4（检索关键词：AI 开发）

图 2.3.2.5 运行截图-5（检索关键词：k 近邻算法）

图 2.3.2.6 运行截图-6（检索关键词：NLP 技术）

本项目所采用的图谱可视化工具支持多种主题背景的选择，如图 2.3.2.7 至图 2.3.2.10 所示。

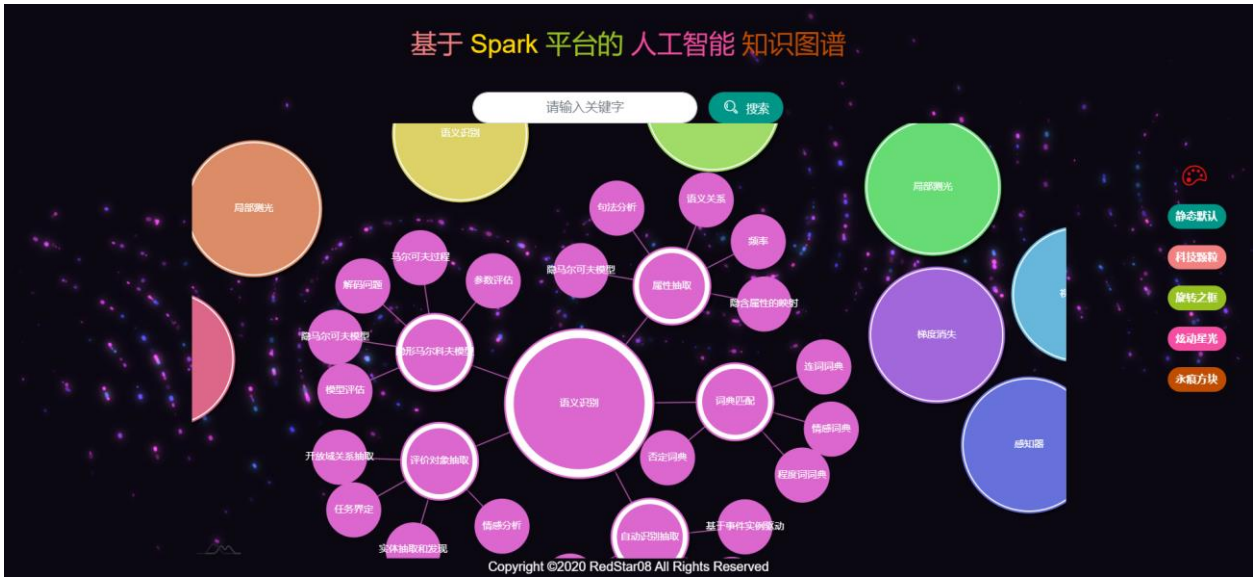

图 2.3.2.7 主题 2“科技颗粒”

图 2.3.2.8 主题 3“旋转之框”

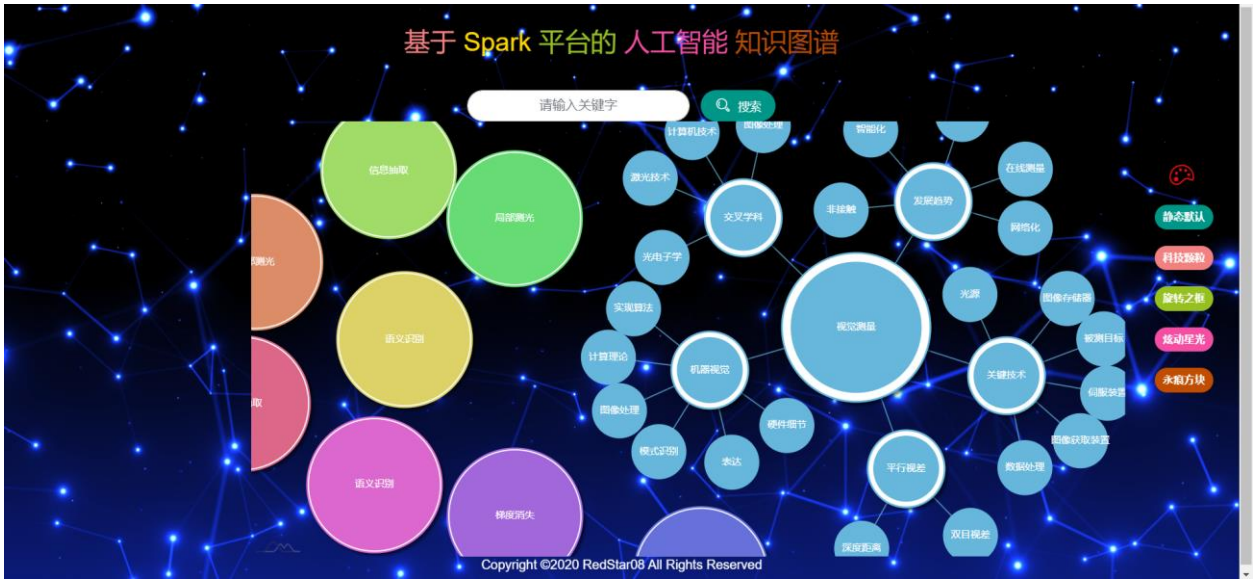

图 2.3.2.9 主题 4“炫动星光”

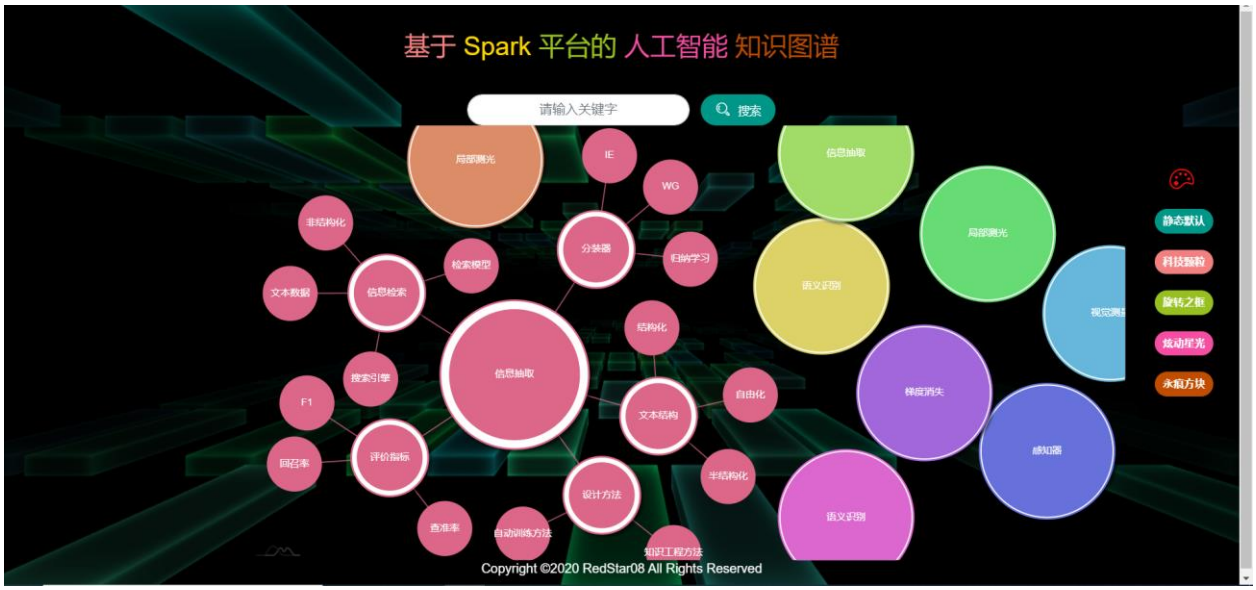

图 2.3.2.10 主题 5“永痕方块”

本产品的 Android 与 iOS 端应用正在紧急开发中，不日即可上线服务。

2.4 产品升级计划

1、知识图谱的升级

（1）目前知识图谱本地存储的知识节点数量较小，在未来的上线服务与用户反馈后，将逐步增大知识节点的数量，扩展图谱规模。

（2）目前采用 Jiagu 自然语言处理工具所提供的知识关系抽取功能需要提供大量的人工标记数据进行模型训练，人工标记数据耗费大量的人力与时间，在下一步的研究中将会尝试使用远程监督模型对原始数据进行标记，减少人力成本的同时，提高了工作效率。

2、云端服务器的升级

（1）当前云端服务器是租用阿里云的轻量服务器，性能一般，将来随着用户的增多与产品盈利，将会改换为性能更优越的服务器，按需增加服务器数量。

（2）用户日均访问量增长的同时会带来巨大的流量消耗，届时将采用分布式云服务器处理框架，并对每台服务器负载均衡技术，减轻单台服务器的计算压力。

3、客户端产品的升级

当前的知识图谱工具只能从 Web 端访问，随着 Android 与 iOS 端应用开发完成，本产品将如期向所有平台的用户提供全方位的知识图谱检索服务。

3　参考文献

[1]刘泽华, 赵文琦, 张楠. 基于 Scrapy 技术的分布式爬虫的设计与优化[J]. 信息技术与信息化, 2018 年 2 - 3 期: 121 – 126.

[2]赛金辰. 基于 Spark 的 SVM 算法优化及其应用[D]. 北京邮电大学, 2017 年 1 月.

[3]李爽. 基于 Spark 的数据处理分析系统的设计与实现[D]. 北京交通大学, 2015 年 6 月.

[4]https://github.com/fighting41love/funNLP

[5]https://github.com/ownthink/Jiagu

[6]徐增林, 盛泳潘, 贺丽荣, 王雅芳. 知识图谱技术综述[J]. 电子科技大学学报, 2016 年 7 月, 第 45 卷第 4 期: 589 – 606.

[7]刘哲宁, 朱聪慧, 郑德权, 赵铁军. 面向特定标注数据稀缺领域的命名实体识别[J]. 指挥信息系统与技术, 2019 年 10 月, 第 10 卷第 5 期: 14 – 18.

[8]MINTZ, BILLS S, SNOW R, et al. Distant supervision for relation extraction without labeled data[C]// Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th

International Joint Conference on Natural Language Proceeding of the AFNLP. Stroudsburg: ACCL, 2009: 1003 – 1011.

[9] 孙启民, 胡莉丽, 黄威. 基于 SNMP&Amcharts 的性能监测技术在动环监控系统的应用[J]. 技术创新, 2016 年 02 期: 35 – 38.