МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Отчет по лабораторной работе №2

по предмету «Информационный поиск»

«Поисковый робот»

Группа: М8О-412Б-22

Студент: Муратов А.А.

Оценка:

Дата сдачи:

Москва, 2025

Задание

Необходимо написать парсер на любом языке программирования.

Написать поисковый робот — компоненты обкачки документов, используя любой язык программирования;

Единственным аргументом поисковому роботу подаётся путь до yaml-конфига, содержащий:

Данные для базы данные в секции db;

Данные для робота в секции logic: задержка между обкачкой страницы;

Любые другие данные, необходимые для реализовации логики поискового робота.

Сохранять в базе данных (например, MongoDB) документы со следующими полями:

- url, нормализованный;
- «сырой» html-текст документа;
- название источника;
- Дата обкачки документа в формате Unix time stamp.

Поисковый робот можно остановить в любой момент и при повторном запуске робот должен начать с того документа, с которого он остановился;

Периодически он должен уметь переобкачивать документы, которые уже есть в базе, но только в том случае, если они изменились.

Описание метода решения

Использованы новости Interfax и TASS с sitemap-индексами (SEO_SiteMapIndex.xml и sitemap.xml). Для попадания в корпус URL должен соответствовать шаблону вида *https://www.interfax.ru/категория/число* или *https://tass.ru/категория/число*.

Ссылки из sitemap проходят фильтр по регулярному выражению и кладутся в коллекцию crawl_queue базы searchdoc со статусом pending. На один источник задан лимит target.

URL нормализуется. Перед постановкой проверяется наличие в documents; тем самым очередь не получает уже сохранённые материалы.

Рекурсивный проход sitemapindex и urlset с учётом глобального лимита, пропуская ошибки загрузки sitemap.

Выкачка файлов происходит так: воркер берёт pending через find_one_and_update, ставит status=running, скачивает HTML, сохраняет в documents поля url, source, raw_html, fetched_at, content_hash. Между запросами выдерживается delay_seconds. При совпадении контент-хэша документ не перезаписывается, только обновляется статус в очереди.

Обработка ошибок: коды 301/302/303/307/308/403/404/410/451 считаются терминальными — запись помечается done с текстом ошибки и больше не повторяется. Иные ошибки возвращают запись в pending с логированием. При старте fetch все running переводятся обратно в pending.

Переобкачка: отдельный режим recrawl возвращает в очередь документы старше заданного порога (recrawl_hours).

Журнал выполнения задания

Настроена фильтрация URL по паттернам, устранены ошибки экранирования в YAML.

Добавлен обход sitemap с ограничением на общее число ссылок и обработкой 403.

Реализован вывод прогресса в enqueue и fetch, а также добавлен сброс зависших running задач при старте fetch.

Исправлены бесконечные повторы на 404 и 307: терминальные коды переводились в done с ошибкой.

В результате работы были получены такие результаты:

Итоговый корпус в **33 182** документа, общий объем базы данных - 5.8 ГБ.

Interfax — 19 942 документов; средний размер ~54 941 байт; минимум 36 636 байт; максимум 87 627 байт; последний fetch: Unix 1 765 808 866.

TASS — 13 240 документов; средний размер ~471 199 байт; минимум 78 187 байт; максимум 721 486 байт; последний fetch: Unix 1 765 978 449

Выводы

Мной реализован выполняющий свои функции эффективно поисковый робот на языке Python для выкачки и переобкачки документов. Он использует карты сайта источника для эффективного обхода и соблюдения параметров закачки документов. Он может переобкачивать документы из источников для обновления.

В результате работы данного поискового робота был сформирован корпус документов, имеющий достаточный объём и качество для использования его в следующих лабораторных работах на курсе.