Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Информационный поиск»

Студент: Н. С. Федоров
Преподаватель: А. А. Кухтичев
Группа: М8О-410Б

Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1 «Добыча корпуса документов»

Необходимо подготовить корпус документов, который будет использован при выполнении остальных лабораторных работ:

- Скачать его к себе на компьютер. В отчёте нужно указать источник данных.
- Ознакомиться с ним, изучить его характеристики. Из чего состоит текст? Есть ли дополнительная мета-информация? Если разметка текста, какая она?
- Разбить на документы.
- Выделить текст.
- Найти существующие поисковики, которые уже можно использовать для поиска по выбранному набору документов (встроенный поиск Википедии, поиск Google с использованием ограничений на URL или на сайт). Если такого поиска найти невозможно, то использовать корпус для выполнения лабораторных работ нельзя!
- Привести несколько примеров запросов к существующим поисковикам, указать недостатки в полученной поисковой выдаче.

В результатах работы должна быть указаны статистическая информация о корпусе:

- Размер «сырых» данных.
- Количество документов.
- Размер текста, выделенного из «сырых» данных.
- Средний размер документа, средний объём текста в документе.

1 Описание

В рамках второго домашнего задания был реализован поисковый робот, предназначенный для автоматической обкачки веб-документов и сохранения их в базе данных. Поисковый робот используется для сбора HTML-страниц с заданных источников и может быть применён в дальнейшем для формирования корпуса документов и проведения экспериментов в области информационного поиска.

Робот принимает на вход путь к YAML-конфигурационному файлу, содержащему параметры подключения к базе данных, а также настройки логики обхода. Такой подход позволяет гибко настраивать работу робота без изменения исходного кода и соответствует требованиям задания.

В конфигурационном файле задаются параметры подключения к базе данных MongoDB, включая адрес сервера, имя базы данных и коллекции. Также в секции логики указываются начальный URL для обхода, допустимые страницы для перехода, глубина обхода, задержка между запросами и вспомогательные параметры, необходимые для корректной работы робота.

Поисковый робот осуществляет последовательный обход страниц, начиная с заданного стартового URL. Для каждой страницы выполняется HTTP-запрос с указанием пользовательского агента, после чего полученный HTML-код анализируется. Если страница удовлетворяет условиям обработки, её содержимое сохраняется в базе данных.

Каждый документ в базе данных содержит следующие поля:

- нормализованный URL страницы;
- «сырой» HTML-текст документа;
- название источника, определяемое по доменному имени;
- дата обкачки документа.

Для обеспечения возможности повторного запуска робота реализован механизм продолжения обхода. При старте робот считывает уже сохранённые URL из базы данных и не обрабатывает их повторно. Текущая позиция обхода и идентификаторы документов сохраняются в конфигурационном файле, что позволяет продолжить работу с места остановки в случае прерывания выполнения.

Для поддержки переобкачки документов реализована проверка изменений содержимого страниц. Для каждого HTML-документа вычисляется хеш-сумма, которая сохраняется в базе данных. При повторной обкачке страницы новая хеш-сумма сравнивается с ранее сохранённой. Если содержимое страницы изменилось, документ в базе данных обновляется, а дата обкачки перезаписывается. Если изменений не обнаружено, обновление документа не производится.

Между запросами к серверу используется настраиваемая задержка, что позволяет снизить нагрузку на обрабатываемые сайты и соответствует принципам корректного веб-сканирования.

Таким образом, реализованный поисковый робот удовлетворяет всем требованиям задания: использует конфигурационный файл, сохраняет необходимые данные в базе, поддерживает возобновление работы и умеет переобкачивать изменённые документы.

2 Исходный код

Поисковый робот реализован на языке Python с использованием стандартных и сторонних библиотек для работы с HTTP-запросами, HTML-документами, конфигурационными файлами и базой данных MongoDB.

Программа начинается с загрузки конфигурационного файла в формате YAML, из которого извлекаются параметры подключения к базе данных и настройки логики обхода. Подключение к MongoDB осуществляется при помощи соответствующей клиентской библиотеки.

Основная логика робота реализована в виде цикла обхода страниц. Для каждой страницы выполняется HTTP-запрос, после чего HTML-документ разбирается и анализируется. Из документа извлекаются ссылки, которые нормализуются и добавляются в очередь обхода при условии, что они соответствуют заданным ограничениям.

Для хранения состояния обхода используется база данных и конфигурационный файл. Уже посещённые страницы учитываются при помощи выборки URL из базы данных, что предотвращает повторную обработку документов. В случае прерывания работы робота текущее состояние сохраняется, что позволяет продолжить обход при следующем запуске.

Для контроля актуальности документов используется вычисление хеш-суммы HTML-кода страницы. Это позволяет эффективно определять, изменилось ли содержимое документа, и обновлять данные в базе только при необходимости.

Результатом работы программы является база данных, содержащая актуальные HTML-документы с информацией об источнике и времени обкачки, готовая для дальнейшей обработки и использования в задачах информационного поиска.

3 Выводы

В ходе выполнения лабораторной работы по курсу я получил практический опыт разработки поискового робота для автоматической обкачки веб-документов. Была изучена архитектура простого веб-краулера и основные принципы его работы.

В процессе реализации стало очевидно, что важной задачей является корректное управление состоянием обхода, особенно в условиях возможного прерывания работы программы. Использование базы данных и конфигурационного файла позволило обеспечить возобновление работы без потери уже собранных данных.

Также был получен опыт реализации механизма переобкачки документов и проверки изменений содержимого страниц. Это позволило лучше понять проблемы актуальности данных и способы их решения при работе с веб-источниками.

Полученные знания и навыки будут полезны при дальнейшем изучении информа-

ционного поиска, разработке собственных поисковых систем, а также при работе с большими объёмами веб-данных и распределёнными источниками информации.

Список литературы

[1] Маннинг, Рагхаван, Шютце *Введение в информационный поиск* — Издательский дом «Вильямс», 2011. Перевод с английского: доктор физ.-мат. наук Д. А. Клюшина — 528 с. (ISBN 978-5-8459-1623-4 (рус.))