

**Московский авиационный институт
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной
математики**

Кафедра вычислительной математики и программирования

Лабораторная работа по курсу «Информационный поиск»

Студент: Н.Х.А. Нгуен
Преподаватель: А. А. Кухтичев
Группа: М8О-409Б-22
Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №1 «Добыча корпуса документов»

Описание

Источник и характеристика корпуса

В качестве источника данных был использован портал научных публикаций CyberLeninka (cyberleninka.ru). Сбор документов производился из двух тематических разделов: «Клиническая медицина» и «Науки о здоровье».

При анализе структуры HTML-документов было установлено, что страницы имеют семантическую разметку с использованием атрибутов микроразметки Schema.org. Заголовок статьи располагается в теге с атрибутом `itemprop="headline"`, а основной текст - в блоке `div` с `itemprop="articleBody"`. Данный подход позволил реализовать парсер, устойчивый к изменениям в верстке страниц.

В мета-тегах страниц также присутствует дополнительная информация: автор, год публикации, название журнала и др. Эти данные могут быть использованы в дальнейшем для реализации расширенного поиска. Тексты статей имеют научный стиль и стандартную структуру.

Анализ существующих поисковых систем

Был проведен сравнительный анализ двух поисковых систем: встроенного поиска на сайте CyberLeninka и поиска Google с оператором `site:cyberleninka.ru`.

Использование фотодинамической диагностики для выявления и лечения неинвазивного рака мочевого пузыря

Шахсуварян В. А.

использование фотодинамической диагностики для выявления и лечения неинвазивного рака мочевого пузыря В.А. ШАХСУВАРЯН Национальный центр онкологии им. В.А. Фанарджяна МЗ РА, отделение онкоурологии, г. Ереван Цель исследования - улучшение результатов лечения поверхностного рака мочевого пузыря. Для достижения этой цели в урологическом отделении НЦО РА с 2007 г. используется единственная в Армении эндоскопическая фотодинамическая система. Материал и методы. Проанализированы результаты обследования и лечения 157 пациентов с подозрением на рак мочевого пузыря. Все больные разделены на 2 группы. 1 группу составили 100 больных, эндоскопическое обследование и лечение (ТУР) которых проводилось без

2009 / Сибирский онкологический журнал

Использование фотодинамических методов в диагностике и лечении поверхностного рака мочевого пузыря

Аль-Шукри С. Х., Кузьмин И. В., Слесаревская М. Н., Соколов А. В.

о целесообразности проведения комбинированного лечения ПВИ с назначением иммунотропной терапии и выполнением лазерной аблации генитальных кандилом. использование фотодинамических методов в диагностике и лечении поверхностного рака мочевого пузыря © С.Х. Аль-Шукри, И.В. Кузьмин, М.Н. Слесаревская, А.В. Соколов ГБОУ ВПО «Первый Санкт-клеток во время ТУР. Использование флуоресцентного контроля во время ТУР мочевого пузыря способствует повышению радикальности операции за счет снижения частоты рецидивов опухоли на 30-40 %. В настоящее время перспективным направлением использования фотодинамических методов в лечении РМП считается комбинация фотодинамической диагностики (ФДД) и фотоди-

2016 / Урологические ведомости

Планирование и проведение сеанса фотодинамической терапии при CIN III с использованием флуоресцентной диагностики

Афанасьев М. С., Гришачёва Т. Г.

* Планирование и проведение сеанса фотодинамической терапии при CIN III с использованием флуоресцентной диагностики Ключевые слова: ФДТ, флуоресцентная диагностика, шейка матки
Keywords: PDT, fluorescence diagnosis, the cervix uteri Афанасьев М.С.1,2, Гришачёва Т.Г.3 1Центр инновационных медицинских технологий (Европейская клиника) (Москва,

CIN III с использованием флуоресцентной диагностики. Флуоресцентная диагностика повышает эффективность не инвазивной дифференциальной диагностики патологических процессов шейки матки в реальном времени, оптимизирует процедуру прицельной биопсии, позволяет проводить мониторинг результатов лечения после фотодинамической терапии. Фотодинамическая терапия

2016 / Research'n Practical Medicine Journal

Рис. 1: Запрос "использование фотодинамической диагностики" во встроенном поисковике CyberLeninka

Сниппеты во встроенном поиске включают два фрагмента текста, содержащие запрос. Ранжирование отдает приоритет статьям, где запрос встречается в заголовке. Присутствует указание автора и года публикации.

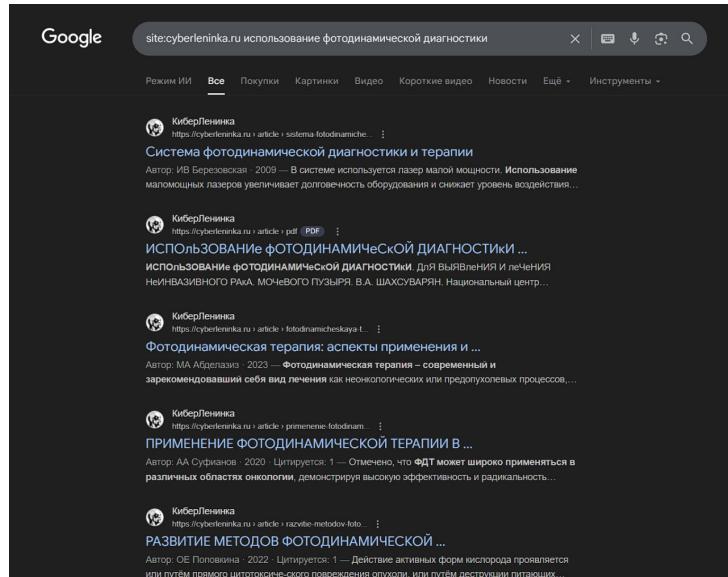


Рис. 2: Запрос "использование фотодинамической диагностики" в поисковике Google

Ранжирование, вероятно, отдает приоритет наиболее популярным статьям. Также присутствует указание автора и года публикации.

Итоговая статистика корпуса

По результатам работы был собран и обработан корпус из 30176 документов.

Параметр	Значение
Источник	cyberleninka.ru
Количество документов	30176
Общий размер «сырых» HTML	2.26 ГБ
Общий размер очищенного текста (TXT)	759.89 МБ
Средний размер «сырого» документа	76.67 KB
Средний объём текста в документе	25.79 KB

Исходный код

<https://github.com/tng00/information-retrieval/tree/main/lab1>

Выводы

В ходе выполнения данной лабораторной работы я освоил практические методы сбора и анализа текстового корпуса для задач информационного поиска. Я научился программно извлекать и очищать данные из HTML-документов, используя их семантическую структуру для отделения контента от служебной информации. Анализ характеристик собранного корпуса и существующих поисковых систем дал понимание масштаба задачи и сформировал основу для последующих этапов разработки. Полученные навыки являются фундаментальными для любых проектов, связанных с обработкой текстовых данных.

Лабораторная работа №2 «Поисковый робот»

Описание

Для сбора корпуса документов был разработан поисковый робот на языке Python. Архитектура робота включает следующие компоненты:

- **Управляющий конфигурационный файл config.yaml**, который задает стартовые URL, разрешенные домены и параметры работы.
- **База данных MongoDB** для хранения скачанных HTML-документов и очереди URL для обхода. Использование двух коллекций (для документов и для очереди) обеспечивает отказоустойчивость. Робот может быть остановлен и перезапущен, продолжая работу с места остановки.
- **Основной цикл**, который атомарно извлекает URL из очереди, скачивает страницу, сохраняет ее и извлекает новые ссылки.
- **Механизм «вежливости»** - реализована задержка между запросами для снижения нагрузки на сервер.
- **Детектор CAPTCHA**, который останавливает работу при обнаружении признаков защиты от автоматических запросов, чтобы избежать блокировки.

Исходный код

<https://github.com/tng00/information-retrieval/tree/main/lab2>

Выводы

В рамках второй лабораторной работы я реализовал поисковый робот. Главным практическим навыком, полученным в ходе работы, стало проектирование многокомпонентной системы с внешним хранилищем состояния в MongoDB, что обеспечило возможность возобновления работы после сбоев. Я научился управлять очередью задач на скачивание, извлекать и фильтровать гиперссылки. Эта работа дала мне понимание сложностей, связанных с автоматизированным сбором веб-данных в больших объемах.

Лабораторная работа №3 «Токенизация»

Описание

Правила токенизации

Процесс разбиения текста на токены реализован на C++ в виде утилиты командной строки, работающей как фильтр. Были выработаны следующие правила:

- **Приведение к нижнему регистру:** Все символы кириллицы и латиницы переводятся в нижний регистр.
- **Определение границ токена:** Разделителями считаются пробелы, знаки препинания и символы перевода строки. Они не включаются в состав токена.
- **Состав токена:** Токеном считается непрерывная последовательность букв, цифр и дефиса (если он не в начале или конце слова).
- **Фильтрация:** Отбрасываются токены, состоящие только из цифр или дефисов, а также токены длиной менее двух символов после всех преобразований.

Достоинства: Высокая скорость обработки за счет реализации на C++ и использования конечного автомата, простота правил. **Недостатки:** Возможна неверная обработка сложносоставных слов или аббревиатур (например, «С.-Петербург»).

Результаты и анализ производительности

- Количество токенов: **81 326 141**
- Средняя длина токена: **11.90**
- Скорость токенизации: **19.45 мБ/с**

Анализ распределения: Закон Ципфа

Для анализа частотного распределения терминов в корпусе был построен график в логарифмических координатах (Рис. 3). На график наложено эмпирическое распределение и теоретическая кривая Закона Ципфа.

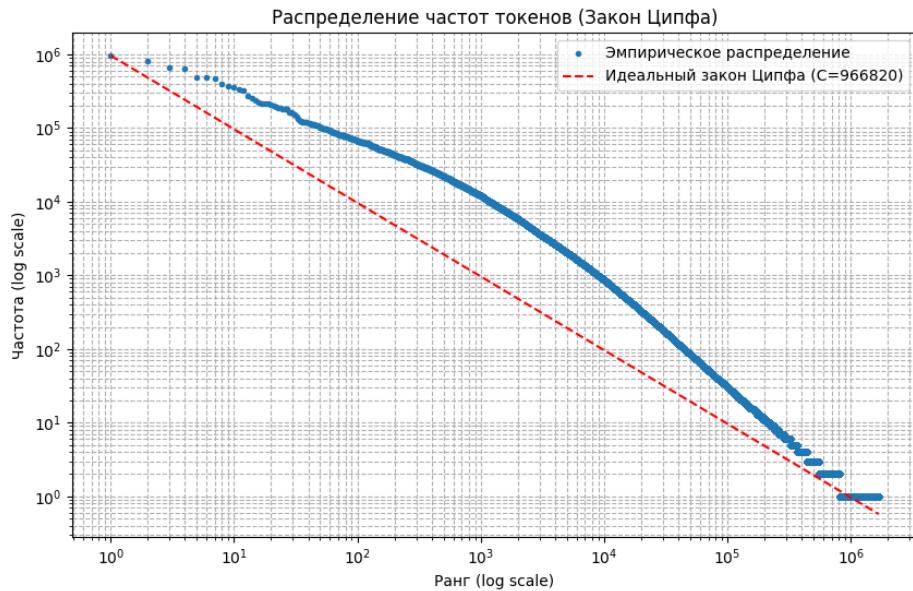


Рис. 3: Распределение частот токенов в логарифмических координатах

График демонстрирует, что распределение в целом следует закону Ципфа, однако наблюдаются характерные отклонения от идеальной модели:

- **«Голова» (начало графика):** Небольшое количество самых частотных терминов встречаются чаще, чем предсказывает теория. Это объясняется тематической однородностью корпуса (медицина), где ключевые термины («пациент», «лечение») доминируют.
- **«Тело» (середина графика):** Эмпирическая кривая идет почти параллельно теоретической, что подтверждает общую закономерность.
- **«Хвост» (конец графика):** Наблюдается большое количество слов с очень низкой частотой, включая множество слов, встретившихся всего один раз (гапаксы). Это приводит к появлению характерных «ступеней» на графике и является фундаментальным свойством естественного языка.

Нормализация: Стемминг

Для нормализации токенов и сокращения словаря был реализован и интегрирован в токенизатор стеммер Портера для русского языка. Стемминг применяется к каждому токену после приведения к нижнему регистру и до проверки на валидность.

Поскольку полноценная поисковая система еще не реализована, оценка качества проводилась качественно, на основе анализа гипотетических запросов.

Улучшение качества: Стемминг значительно повышает полноту поиска. Например, запрос [современная терапия] после обработки превратится в [современ терап]. Такой запрос найдет документы, содержащие словосочетания «современной терапии», «современную терапию» и т.д., которые не были бы найдены без нормализации.

Ухудшение качества (потеря точности): Агрессивный характер стемминга может приводить к ложным срабатываниям. Например, стеммер может ошибочно свести разные по смыслу слова к одной основе. Гипотетический пример: слова «универсальный» и «университет» могут быть сведены к общей основе «универс», что приведет к появлению нерелевантных документов в выдаче.

Способы улучшения: Для решения проблемы потери точности можно использовать более сложный метод нормализации - лемматизацию, которая приводит слово к его нормальной словарной форме (лемме) с использованием словарей. Однако этот метод значительно медленнее стемминга.

Исходный код

<https://github.com/tng00/information-retrieval/tree/main/lab3>

Выводы

В этой работе я создал ключевые компоненты для лингвистической обработки текста. Реализация токенизатора и последующий частотный анализ по закону Ципфа позволили на практике увидеть статистические закономерности реального текстового корпуса. Интеграция стеммера Портера продемонстрировала компромисс между повышением полноты поиска и риском снижения его точности.

Лабораторная работа №4 «Булев индекс»

Описание

Внутреннее представление данных

Для хранения индекса был разработан собственный бинарный формат, состоящий из трех файлов.

- **Прямой индекс (forward.idx):** Хранит URL и заголовок для каждого документа. Файл состоит из таблицы смещений и области данных, что обеспечивает быстрый доступ к информации по DocID.
- **Словарь (dictionary.dat):** Содержит лексикографически отсортированный список всех уникальных термов. Для каждого терма хранится его длина, частота встречаемости в документах (`doc_frequency`) и смещение в файле `postings.dat`.
- **Списки вхождений (postings.dat):** Хранит последовательности DocID для каждого терма. Для экономии места и подготовки к сжатию хранятся не сами DocID, а их разницы (`d-gaps`).

Метод сортировки

Для лексикографической сортировки словаря в памяти использовалась стандартная функция C `qsort`.

- **Достоинства:** Реализация быстрой сортировки (в среднем $O(N \log N)$), доступная в стандартной библиотеке C, не требует написания с нуля.
- **Недостатки:** Алгоритм работает только с данными, полностью помещающимися в оперативную память. Для корпусов большего размера он неприменим.

Результаты

- Количество термов: **1 672 481**
- Средняя длина терма: **14.71 символов**
- Скорость индексации (общая): **202.82 мс**
- Скорость индексации (на документ): **6.76 мс**

Анализ производительности и масштабирования: Текущая реализация ограничена объемом оперативной памяти. Малый размер хеш-таблицы к длинным цепочкам коллизий, что замедляет поиск. Для ускорения можно увеличить этот размер. При увеличении объема данных в 10-1000 раз программа может завершиться с ошибкой нехватки памяти.

Исходный код

<https://github.com/tng00/information-retrieval/tree/main/lab4>

Выводы

В ходе этой лабораторной работы я реализовал ядро поисковой системы - индексатор, создающий прямой и обратный индексы. Я на практике освоил проектирование собственных бинарных форматов данных, что позволило эффективно хранить словарь и списки вхождений. Также я реализовал собственную хеш-таблицу для ускорения процесса построения индекса.

Лабораторная работа №5 «Булев поиск»

Описание

Скорость выполнения запросов

Скорость измерялась на наборе из 1000 тестовых запросов разной сложности.

- Средняя скорость выполнения запроса: **1339.13 мс**

Примеры сложных запросов

- **Запросы с большим количеством операторов OR:** Запрос вида [результат || исследование || данные] вызывает длительную работу, так как требует слияния нескольких очень длинных списков документов, что создает большие промежуточные результаты.
- **Запросы с оператором NOT от редкого слова:** Запрос [!гапакс] является дорогим, так как требует создания списка почти всех документов корпуса и последующего исключения из него одного элемента. В реальных системах такие запросы часто запрещают или выполняют в связке с AND.

Тестирование корректности

Корректность поисковой выдачи проверялась в два этапа:

1. **Модульное тестирование:** Были созданы небольшие, заранее определенные списки DocID, на которых проверялась корректность работы функций пересечения (AND), объединения (OR) и инверсии (NOT).
2. **Интеграционное тестирование:** Был создан тестовый мини-корпус из 5 документов, для которого индекс был построен «вручную». Затем было выполнено 10-15 булевых запросов разной сложности. Результаты, выданные программой, сравнивались с ожидаемыми, вычисленными вручную. Этот метод позволил проверить корректность всей цепочки: от парсинга запроса до выполнения операций.

Интерфейсная часть

Поиск

ишемия || глазница

Найдено документов: 3299

[Последние достижения в применении высоких доз аторвастатина у больных ишемической болезнью сердца](#)
<https://cyberleninka.ru/article/n/poslednie-dostizheniya-v-primenenii-vysokih-doz-atorvastatina-u-bolnyh-ishemicheskoy-boleznyu-serdtsa>

[Метод исследования и диагностики состояния патологически измененных структур желудочков сердца при их коррекции](#)
<https://cyberleninka.ru/article/n/metod-issledovaniya-i-diagnostiki-sostoyaniya-patologicheski-izmenennyh-struktur-zheludochkov-serdtsa-pri-ih-korrektcii>

[Остеоартроз: факторы риска, патогенез и современная терапия](#)
<https://cyberleninka.ru/article/n/osteoartroz-faktory-riska-patogenez-i-sovremenennaya-terapiya>

[Интенсивная терапия аторвастатином. Повышение эффективности лечения](#)
<https://cyberleninka.ru/article/n/intensivnaya-terapiya-atorvastatinom-povyshenie-effektivnosti-lecheniya>

[Актуальность применения высоких доз аторвастатина у больных ишемической болезнью сердца](#)
<https://cyberleninka.ru/article/n/aktualnost-primeneniya-vysokih-doz-atorvastatina-u-bolnyh-ishemicheskoy-boleznyu-serdtsa>

Рис. 4: Пример запроса

[Лапароскопическая резекция почки с применением радиочастотной термоабляции](#)
<https://cyberleninka.ru/article/n/laparoskopicheskaya-rezektsiya-pochki-s-primeneniem-radiochastotnoy-termoablatsii>

[Новые возможности органосохраняющего лечения локализованного почечно-клеточного рака и его рецидивы](#)
<https://cyberleninka.ru/article/n/novye-vozmozhnosti-organosohranyayuscheego-lecheniya-lokalizovannogo-pochechno-kletchnogo-raka-i-ego-retsidivy>

Страницы: ← Предыдущая 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 Следующая →

Рис. 5: Пагинация

Исходный код

<https://github.com/tng00/information-retrieval/tree/main/lab5>

Выходы

В рамках данной работы была создана полноценная поисковая система с веб-интерфейсом. Я реализовал синтаксический анализатор булевых запросов с помощью алгоритма "Сортировочная станция" что позволило корректно обрабатывать скобки и приоритет операций. Эта лабораторная работа объединила все предыдущие наработки в единый продукт и дала практический опыт в проектировании клиент-серверной архитектуры для поисковых задач.

Требования к компиляции и запуску

Требования к окружению

- ОС: Linux-совместимая (например, Ubuntu в WSL)
- Компилятор C++: g++
- Интерпретатор Python 3.8+
- Система управления пакетами Python: pip
- База данных: MongoDB (запущенная через Docker)

Последовательность сборки и запуска

1. Установить зависимости Python из файла requirements.txt:

```
1 || pip install -r requirements.txt
```

2. Скомпилировать C++-утилиты (токенизатор, индексатор, поисковик):

```
1 || g++ -O2 -o tokenizer tokenizer.cpp  
2 || g++ -O2 -o indexer indexer.cpp  
3 || g++ -O2 -o searcher searcher.cpp
```

3. Запустить Docker-контейнер с MongoDB:

```
1 || docker start mongo_container
```

4. Запустить процесс построения индекса с помощью скрипта-оркестратора:

```
1 || ./run_indexing.sh
```

5. Запустить веб-сервис для демонстрации поиска:

```
1 || python3 app.py
```

6. Открыть в браузере адрес <http://127.0.0.1:5000>.

Список литературы

- [1] Маннинг К., Рагхаван П., Шютце Х. *Введение в информационный поиск.* — М.: Вильямс, 2011.
- [2] Ахо А., Ульман Дж. *Теория синтаксического анализа, перевода и компиляции. Том 1: Синтаксический анализ.* — М.: Мир, 1978.