# Информационный поиск и обработка естественно-языковых текстов, 2021

Преподаватель: Кухтичев Антон Алексеевич

Электронная почта: a.kukhtichev@mail.ru

Версия от 15.03.2021

## Оглавление

# График сдачи лабораторных работ

Студент _____, группа _____

| Курс | Работа | Дата сдачи | Оценка | Подпись преподавателя |
|---|---|---|---|---|
| **ИП** | **ЛР1: Добыча корпуса документов** | | | |
| **ОЕЯТ** | **ЛР1: Токенизация** | | | |
| **ИП** | **ЛР2: Оценка качества поиска** | | | |
| **ИП** | **ЛР3: Булев индекс** | | | |
| **ИП** | **ЛР4: Булев поиск** | | | |
| **ОЕЯТ** | **ЛР2: Закон Ципфа** | | | |
| **ИП** | ЛР5: Цитатный поиск, координаты | | | |
| **ИП** | ЛР6: Сжатие | | | |
| **ИП** | ЛР7: Ускорение, прыжки по индексу | | | |
| **ИП** | **ЛР8: TF-IDF** | | | |
| **ОЕЯТ** | **ЛР3: Лемматизация** | | | |
| **ОЕЯТ** | ЛР4: Построение сниппетов | | | |
| **ИП** | ЛР9: Зонный поиск | | | |
| **ОЕЯТ** | ЛР5: Поиск коллокаций | | | |

# Информационный поиск

## Общие требования к выполнению лабораторных работ

Лабораторные работы должны выполняться самостоятельно, использование чужого кода (плагиат) недопустимо. Кроме того, все структуры данных, используемые для построения поисковых индексов и поиска по нему, должны быть тоже сделаны самостоятельно, без использования похожих по функциональности библиотек и компонент выбранного языка программирования. В качестве языка программирования для всех основных компонент поисковой системы может быть выбран C или C++ без STL (STL можно применять только для токенизации). Для обвязки, выкачки, может быть выбран любой интерпретируемый язык программирования (Python, Perl, Shell, … ) и дополнительные утилиты (curl, wget, … )

Задания выдаются в начале семестра. Оценка за выполненное задание зависит от применимости его к корпусу большого размера. Оценка 3 ставится если задача выполнена для корпуса размером в 30-50 тысяч документов, оценка 5 ставится при количестве документов больше 1 миллиона (при условии, что они не помещаются все в оперативную память используемого компьютера). Так же оценивается знание материала, если студент не смог продемонстрировать базовое знание материала, необходимое для выполнения лабораторной работы, работа не принимается.

В качестве входного корпуса должен быть использован уникальный в рамках учебных групп (т.е., у каждого студента свой, повторение не допускается) набор документов, состоящий как минимум из 30 000 статей размером в несколько тысяч слов единой тематики. Хорошей подборкой таких статей является тематическая категория каталога Википедии.

Если в работе подразумевается интерфейсная часть (ввод поискового запроса, выдача), то она должна быть выполнена в двух видах:

1. Как веб-сервис с формой ввода и получением поисковых результатов в формате HTML.
2. Как утилита командной строки, получающая запросы со стандартного файла ввода и выдающая результат в стандартный файл вывода.

Продемонстрировать работу можно либо опубликовав (сделав доступным) сервис в сети Интернет и прислав ссылку на него по электронной почте преподавателю, либо продемонстрировав работу на ПК, доступном в присутственное время в МАИ (например, на личном ПК или на компьютере кафедры, предоставленном для выполнения лабораторных работ). Утилита командной строки должна использоваться для создания дампов выполнения лабораторной работы в процессе тестирования и для посылки решения преподавателю по электронной почте.

Для каждой лабораторной работы, в которой нужно создать какую-нибудь программу, нужно подготовить и обосновать план тестирования, максимально покрывающий работоспособность программы. По этому плану должны быть созданы автотесты, использующие утилиту командной строки, результат автотестов должен быть приложен к отчёту о выполнении лабораторной работе.

Кодировка файлов ввода-вывода должна быть единой для всех лабораторных работ, UTF-8.

Результатом выполнения лабораторной работы является отчёт о её выполнении. Состав отчёта:

1. Титульный лист: название работы, ФИО студента, место под отметку о сдаче и оценке работы.
2. Задание.
3. Краткое описание метода решения задачи.
4. Журнал выполнения задания. Описание возникших проблем и выбранных методов их решения.
5. План тестирования.

6. Результаты, включающие в себя количественные измерения, оценку качества поиска по подходящим метрикам (P, DCG, NDCG, ERR), должен быть продемонстрирован рост метрик или, как минимум, незначительное падение для лабораторных работ, связанных со сжатием данных, иллюстративный материал.
7. Выводы. Обязателен критический анализ работы, указание недостатков и способов их решения.

Отчёт должен быть представлен в двух видах: исходном документе, доступном для редактирования, и в формате PDF.

К отчёту должны быть приложены:

1. Все исходные тексты, необходимые для запуска программы.
2. Текстовое описание требований к компиляции и запуску программы.
3. Скрипт, собирающий программы, запускающий автотесты. Должна быть обеспечена простая повторяемость выполнения тестового плана «с нуля».
4. Результат прогона тестового плана, в котором должны быть легко отделимы тесткейсы.

Корпус документов присылается один раз, ссылкой на архив в интернете или передаётся лично на флешке.

Отчёт присылается преподавателю по электронной почте одним архивом, оценка ставится после очного опроса студента по теме лабораторной работы. Отметка о сдаче должна быть поставлена студенту в его график сдачи лабораторных работ (для этого нужно распечатать график со второй страницы), хранимый у студента. Во время сдачи экзамена студент сдаёт полный отчёт о выполнении лабораторных работ, состоящий из:

1. Титульного листа.
2. Графика выполнения лабораторных работ.
3. Компакт диска или флешки с исходными данными и всеми архивами с отчётами и исходными текстами, принятыми преподавателем в течение семестра.

## ЛР1: 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.

## ЛР2: Оценка качества поиска

Необходимо оценить качество своего поиска и сравнить их с двумя альтернативами (для Википедии можно собственный поиск по Википедии, поиск Google или Яндекса с ограничением по сайту Википедии). Как минимум, нужно измерить P, DCG, NDCG и ERR уровней @1, @3 и @5, приветствуется использование дополнительных метрик качества.

Для оценки качества необходимо придумать 30 запросов, отражающих интересы пользователей или, если есть доступ к настоящим запросам пользователей, то выбрать репрезентативную подборку.

В качестве примера посмотрите на 10 запросов к поиску по всей Википедии, подумайте о том, почему именно они были выбраны и какую сложность для поисковой системы они представляют:

1. [ из каких книг состоит библия ]
2. [ что где когда ]
3. [ игра ]
4. [ российские авиазаводы ]
5. [ без меня народ не полный ]
6. [ как называют жителей набережных челнов ]
7. [ где короновали николая 2 ]
8. [ товарищ прокурора ]
9. [ цари газы ]
10. [ администрааивный кодекс ]

Проведите анализ результатов оценки качества. Какие у какой поисковой системы сильные и слабые стороны? Как можно бороться с недостатками, что можно сделать, чтобы улучшить качество?

## ЛР3: Булев индекс

Требуется построить поисковый индекс, пригодный для булева поиска, по подготовленному в ЛР1 корпусу документов.

Требования к индексу:

- Самостоятельно разработанный, бинарный формат представления данных. Формат необходимо описать в отчёте, в побайтовом (или побитовом) представлении.
- Формат должен предполагать расширение, т.к. в следующих работах он будет меняться под требования новых лабораторных работ.
- Использование текстового представления или готовых баз данных не допускается.
- Кроме обратного индекса, должен быть создан «прямой» индекс, содержащий в себе как минимум заголовки документов и ссылки на них (понадобятся для выполнения ЛР4, при генерации страницы поисковой выдачи).
- Для термов должна быть как минимум понижена капитализация.

В отчёте должно быть отмечено как минимум:

- Выбранное внутренне представление документов после токенизации.
- Выбранный метод сортировки, его достоинства и недостатки для задачи индексации.

Среди результатов и выводов работы нужно указать:

- Количество термов.
- Средняя длина терма. Сравнить со средней длинной токена, вычисленной в ЛР1 по курсу ОТЕЯ. Объяснить причину отличий.

- Скорость индексации: общую, в расчёте на один документ, на килобайт текста.
- Оптимальна ли работа индексации? Что можно ускорить? Каким образом? Чем она ограниченна? Что произойдёт, если объём входных данных увеличится в 10 раз, в 100 раз, в 1000 раз?

## ЛР4: Булев поиск

Нужно реализовать ввод поисковых запросов и их выполнение над индексом, получение поисковой выдачи.

Синтаксис поисковых запросов:

- Пробел или два амперсанда, «&&», соответствуют логической операции «И».
- Две вертикальных «палочки», «||» – логическая операция «ИЛИ»
- Восклицательный знак, «!» – логическая операция «НЕТ»
- Могут использоваться скобки.

Парсер поисковых запросов должен быть устойчив к переменному числу пробелов, максимально толерантен к введённому поисковому запросу.

Примеры запросов:

- [ московский авиационный институт ]
- [ (красный || желтый) автомобиль ]
- [ руки !ноги ]

Для демонстрации работы поисковой системы должен быть реализован веб-сервис, реализующий базовую функциональность поиска из двух страниц:

- Начальная страница с формой ввода поискового запроса.
- Страница поисковой выдачи, содержащая в себе форму ввода поискового запроса, 50 результатов поиска в виде текстов заголовков документов и ссылок на эти документы, а так же ссылку на получение следующих 50 результатов.

Так же должна быть реализована утилита командной строки, загружающая индекс и выполняющая поиск по нему для каждого запроса на отдельной строчке входного файла.

В отчёте должно быть отмечено:

- Скорость выполнения поисковых запросов.
- Примеры сложных поисковых запросов, вызывающих длительную работу.
- Каким образом тестировалась корректность поисковой выдачи.

## ЛР5: Поиск цитат, координатный индекс

В этом задании необходимо расширить язык запросов булева поиска новым элементом – поиском цитат. Синтаксис этого элемента следующий:

- [ «что где когда» ] – кавычки, включают режим цитатного поиска для терминов внутри кавычек. Этому запросу удовлетворяют документы, содержащие в себе все термины *что*, *где* и *когда*, причём они должны встретиться внутри документа ровно в этой последовательности, без каких либо вкраплений других терминов.
- [ «что где когда» / 5 ] – аналогично предыдущему пункту, но допускаются вкрапления других терминов так, чтобы расстояние от первого термина цитаты до последнего не превышало бы 5.

Новый элемент может комбинироваться с другими стандартными средствами булева поиска, например:

- [ «что где когда» && друзь ]
- [ «что где когда» || квн ]
- [ «что где когда» && !«хрустальная сова» ]

Для реализации цитатного поиска нужно использовать координатный индекс, т.е. для каждого вхождения термина в документ построить и сохранить список позиций внутри документа, где этот термин встречался.

В отчёте нужно описать формат координатного индекса. Привести статистические данные:

- Размер получившегося индекса.
- Время построения индекса.
- Общее количество позиций. Среднее количество позиций на термин и на пару термин-документ.
- Скорость индексации (кб входных данных в секунду)
- Время выполнения поисковых запросов.
- Примеры долго выполняющихся запросов.

Кроме того, нужно привести примеры запросов и результаты их выполнения. В выводах должны быть указаны недостатки работы, приведены примеры их решения. Что можно сделать, чтобы ускорить «долгие» запросы?

## ЛР6: Сжатие

В этом задании необходимо применить алгоритмы сжатия к координатным блокам. Исследовать изменения в размерах частей индекса, влияние на скорость индексации и поиска.

В отчёте нужно указать:

- Выбранный метод сжатия. Привести побитовую схему хранения данных в индексе. Описать причины, по которым был выбран именно этот метод сжатия.
- Влияние сжатия на размер и скорость прохождения по координатным блокам всех терминов, редких терминов, терминов средней частотности и высокочастотных терминов.
- Обосновать, почему поиск после внедрения сжатия работает корректно. Как производилось тестирование?

## ЛР7: Ускорение, прыжки по индексу

В полученный в предыдущих лабораторных работах индекс нужно добавить специальную информацию, позволяющую выполнить «прыжки по индексу», чтобы ускорить пересечение высокочастотных терминов.

Нужно выбрать расстояние, на которое можно выполнить прыжок, таким образом, чтобы получить максимальное ускорение на исследуемом пуле поисковых запросов. Продемонстрировать, что на другом, тестовом пуле поисковых запросов, ускорение тоже есть. Если ускорение будет разным, объяснить причины.

В отчёте должна быть представлена побитовая схема хранения индекса, с дополненными полями для эффективного выполнения прыжков по индексу. Должна быть показана зависимость между размером прыжка и средней скоростью выполнения запросов.

### ЛР8: Ранжирование TF-IDF

Необходимо сделать ранжированный поиск на основании схемы ранжирования TF-IDF. Теперь, если запрос содержит в себе только термины через пробелы, то его надо трактовать как нечёткий запрос, т.е. допускать неполное соответствие документа терминам запроса и т.п. Примеры запросов:

- [ роза цветок ]
- [ московский авиационный институт ]

Если запрос содержит в себе операторы булева поиска, то запрос надо трактовать как булев, т.е. соответствие должно быть строгим, но порядок выдачи должен быть определён ранжированием TF-IDF. Например:

- [ роза && цветок ]
- [ московский && авиационный && институт ]

В отчёте нужно привести несколько примеров выполнения запросов, как удачных, так и не удачных.

### ЛР9: Зонный поиск

Необходимо добавить в поисковый индекс информацию о зонах, в которых встретились термины. Как минимум, нужно сделать отдельные зоны для заголовков документов. Так же, необходимо учесть эти зоны в ранжировании, причём таким образом, чтобы поиск стал искать лучше.

В отчёте нужно привести:

- Побитовое описание индекса с зонами.
- Формулу ранжирования, подобранные веса.
- Оценку качества поиска после внедрения зон.

Есть ли запросы, по которым качество ухудшилось? Почему? Что можно сделать, чтобы качество поиска по ним улучшилось, а по остальным запросам – не ухудшилось бы?

### Курсовой проект

Варианты тем:

- Поиск по полному дампу русскоязычной Википедии.
- Спеллчекер для поиска.
- Поисковые подсказки.
- Поиск музыки по услышанному фрагменту.
- Диалоговый интерфейс (бот), скилл, навык для голосового помощника.
- Поиск новостей, твитов, музыки, картинок, видео, субтитров, …
- Тему можно придумать самостоятельно и согласовать её с преподавателем.

## Обработка текстов на естественном языке

Требования к лабораторным работам аналогичны требованиям к лабораторным работам по курсу информационного поиска.

### ЛР1: Токенизация

Нужно реализовать  процесс разбиения текстов документов на токены, который потом будет использоваться при индексации. Для этого потребуется выработать правила, по которым текст делится на токены. Необходимо описать их в отчёте, указать достоинства и недостатки

выбранного метода. Привести примеры токенов, которые были выделены неудачно, объяснить, как можно было бы поправить правила, чтобы исправить найденные проблемы.

В результатах выполнения работы нужно указать следующие статистические данные:

- Количество токенов.
- Среднюю длину токена.

Кроме того, нужно привести время выполнения программы, указать зависимость времени от объёма входных данных. Указать скорость токенизации в расчёте на килобайт входного текста. Является ли эта скорость оптимальной? Как её можно ускорить?

## ЛР2: Закон Ципфа

Для своего корпуса необходимо построить график распределения терминов по частотностям в логарифмической шкале, наложить на этот график закон Ципфа. Объяснить причины расхождения.

В качестве дополнительного задания, можно (но необязательно) подобрать константы для закона Мандельброта, наложить полученный график на график распределения терминов по частотностям. Привести выбранные константы.

## ЛР3: Лемматизация

Добавить в созданную поисковую систему (ЛР 1-8 по курсу «Информационный поиск») лемматизацию. В простейшем случае, это просто поиск без учёта словоформ. В более сложном случае, можно давать бонус большего размера за точное совпадение слов. Лемматизацию можно добавлять на этапе индексации, можно на этапе выполнения поискового запроса.

В отчёте должна быть включена оценка качества поиска, после внедрения лемматизации. Стало ли лучше? Изучите запросы, где качество ухудшилось. Объясните причину ухудшения и как можно было бы улучшить качество поиска по этим запросам, не ухудшая остальные запросы?

## ЛР4: Построение сниппетов

Необходимо добавить в поисковую систему построение цитат (сниппетов), реферирование документов, найденных по запросу.

Сниппеты должны содержать слова запроса и давать пользователю представление о том, насколько документ отвечает поисковому запросу. Длинна сниппета должна быть ограничена двумя-тремя строчками.

В отчёте нужно привести описание алгоритма построения сниппетов, примеры.

## ЛР5: Поиск коллокаций

Необходимо найти коллокации в имеющемся корпусе, использованного для построения поисковой системы (или его случайному подмножеству достаточного размера). Для поиска коллокаций необходимо использовать как минимум два статистических алгоритма из рассмотренных на лекциях. Сравнить выделенные коллокации между собой, пояснить различия с точки зрения использованых алгоритмов.

В отчёте должны быть приведены количество найденных коллокаций, оценка точности метода, примеры найденных коллокаций и ошибочно найденных словосочетаний.