МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»

Лабораторная работа №*3*
по курсу «Информационный поиск»

«Токенизация»

Выполнил: *Ермаков Ярослав Валерьевич*
Группа: *М8О-407Б-22*
Преподаватели: *А.А. Кухтичев*

Москва, *2025*

Условия

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?

Описание

Цель работы - реализовать токенизацию текстов документов корпуса, чтобы затем использовать полученные токены при построении индексов и поиске. Токенизация выполняется как последовательный проход по тексту документа с формированием токенов по заданным правилам, с приведением к единой форме и дополнительной нормализацией для русского языка. Отдельно предусмотрен режим со стеммингом, чтобы на следующих этапах уменьшать число различных словоформ.

Правила токенизации

В качестве токена рассматривается непрерывная последовательность символов, относящихся к «словным» или «числовым» (буквы латиницы/кириллицы и цифры). Разделителями считаются пробельные символы и знаки пунктуации. Перед добавлением токена выполняются нормализации: понижение регистра, приведение некоторых вариантов написания к одному виду (например, ё -> е). Для технических терминов допускаются составные формы: дефис внутри токена сохраняется, что позволяет не разрушать записи вида state-of-the-art, 0-shot, x-ray, номера версий и похожие обозначения. Токены короче минимального порога (например, 1 символ) отбрасываются, чтобы уменьшить шум от одиночных букв и случайных символов.

Достоинства подхода: простая и быстрая реализация, линейная сложность по длине текста, устойчивая работа на больших объёмах данных, единая нормализация регистра и русских вариантов написания.

Недостатки: часть “шумных” токенов всё равно остаётся, а также возможны спорные случаи с апострофами, сокращениями и HTML/служебными фрагментами.

Примеры неудачных токенов и возможные улучшения

При токенизации научных HTML-страниц встречаются токены, которые не несут смысловой нагрузки для поиска:

- Чистые числа и «нулевые» группы: 00, 000 и т.п. Они часто возникают из-за разметки, ссылок, идентификаторов и версий. (Улучшение: отбрасывать токены, состоящие только из цифр (или понижать их вес на этапе ранжирования)).
- Слова с апострофом и сокращения: 0's и подобные формы. (Улучшение: определить правило для апострофов: либо оставлять внутри токена только для буквенных сокращений, либо заменять апостроф на разделитель.)

- Составные технические идентификаторы: 000-document, 0-8b-instruct. Для поиска по смыслу они полезны не всегда, но могут быть полезны для технических запросов. (Улучшение: оставить как есть (для совместимости с техтекстами) или вводить отдельную нормализацию для “модельных/версионных” токенов.)
- Артефакты HTML/кодировок в тексте: иногда в поле title/тексте встречаются искажённые последовательности символов из-за особенностей HTML или кодировки. (Улучшение: усилить очистку HTML до токенизации (удалять навигационные элементы, мусорные строки, проверять корректность декодирования UTF-8).)

Результаты и статистические данные

Токенизация была выполнена на полном корпусе из 58 866 документов. Получены следующие измерения:

- **docs = 58866** - сколько документов из списка было обработано токенизатором.
- **total_bytes = 529704878** - суммарный объём входного текста всех документов (в байтах), который был прочитан и токенизирован.
- **token_count = 70584169** - общее число выделенных токенов по всему корпусу.
- **avg_token_len_chars = 6.00442** - средняя длина одного токена в символах (среднее по всем токенам).
- **time_sec = 6.38815** - время работы программы токенизации на всём корпусе (в секундах).
- **tokens_per_kb = 136.45** - плотность токенов: сколько токенов в среднем приходится на 1 килобайт входного текста (удобно для сравнения разных корпусов/настроек).
- **stemming = 1** - стемминг был включён при токенизации (если 0, то токены сохранялись без стемминга).

Время выполнения, зависимость от объёма и скорость

Алгоритм токенизации выполняет один линейный проход по тексту каждого документа, поэтому время работы **пропорционально общему объёму входных данных**(т.е линейна). На полном корпусе объём входного текста составил около **517 тыс. KiB**, а время выполнения - **6.39 сек**, что даёт скорость порядка **~81 000 KiB/сек**. В пересчёте на токены это примерно **~11 млн токенов/сек**.

Такая скорость для однопроходной токенизации близка к практическому пределу, и дальнейшее ускорение обычно упирается в ввод-вывод и обработку Unicode. **Потенциальные способы ускорения:** чтение крупными буферами/mmap, уменьшение числа выделений памяти, упрощение проверок символов, распараллеливание по документам, а также оптимизация/отключение стемминга при измерении “чистой” токенизации.

Выводы

В лабораторной работе реализована токенизация корпуса документов с едиными правилами выделения токенов и нормализацией. Получены статистические показатели (количество токенов, средняя длина токена, время работы и скорость), а также выявлены типичные примеры “шумных” токенов и направления для улучшения качества токенизации на смешанных HTML-данных.