Поиск

Обзор

- Какой бывает
- Из каких компонентов состоит
- Какие задачи возникают

Дисклеймер: примеры хаотичны

Виды поиска

Онлайн

- grep, ack
- SELECT ... LIKE **‘%text%’**
- Ctrl+F

✗ Без построения индекса
✓ Тривиальное добавление документа в коллекцию
✗ Поиск только полным сканированием

КМП, Алгоритм Бойера-Мура, regex

Виды поиска

Офлайн

- Elasticsearch (Lucene), Sphinx
- Google, Яндекс
- Ваш проект

✓ С построением индекса
✗ Добавление документа требует построения индекса (Δ?)
✓ Множество возможностей по оптимизации

Алгоритмы, Тервер, Теория автоматов, Машинное обучение…

Алгоритм поиска

Индексация

Поиск

Расходимся

Алгоритм поиска

Индексация

- Чтение источника документов (БД, csv, API, …)
- Парсинг документов
- Построение полнотекстового индекса

Поиск

- Парсинг запроса
- Матчинг
- Постобработка (ранжирование, бизнес-логика, …)

Модель поиска

Документы:

```
70100  The Matrix

70110  The Matrix Reloaded

70112  The Matrix Revolutions
```

Документы содержат слова

Модель поиска

Обратный индекс:

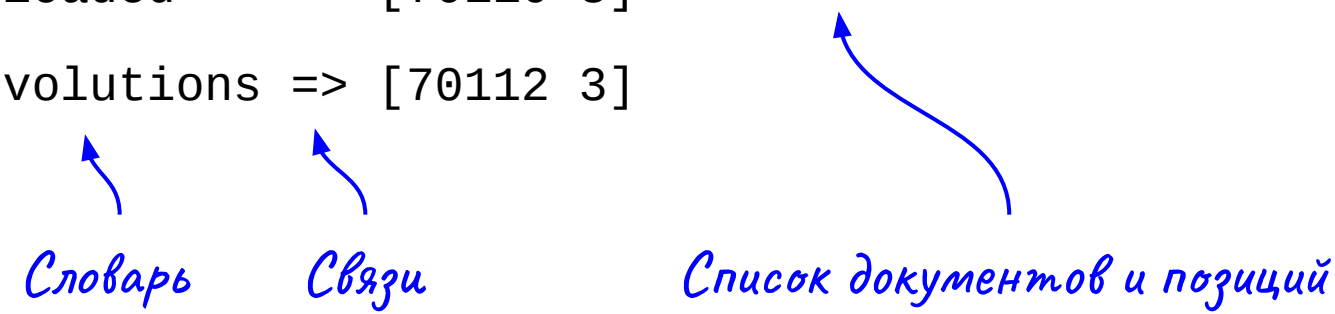
```
matrix      => [70100 2, 70110 2, 70112 2]
the         => [70100 1, 70110 1, 70112 1]
reloaded    => [70110 3]
revolutions => [70112 3]
```

Слова входят в документы

Модель поиска

Обратный индекс:

```
matrix      => [70100 2, 70110 2, 70112 2]
the         => [70100 1, 70110 1, 70112 1]
reloaded    => [70110 3]
revolutions => [70112 3]
```



Модель поиска

Обратный индекс:

```
matrix      => [70100 2, 70110 2, 70112 2]
               Δ [70100 2,    10 2,     2 2]
               + varint
```

– *Сортированный список*
– *Хранение в бинарном формате*
– *Сжатие*

Чтение источника документов

- У вас — csv-файл фиксированного формата
- В универсальной системе — произвольные выходные данные
  - Реляционная БД
  - csv, xml, json
  - html?
  - pdf?

Это архитектурное решение.

abstract base class DocumentSource?

Что есть документ?

- Денормализация: документ — это объект со множеством полей
  - title
  - author
  - ISBN
  - …
- Нормализация: документ — это поле + связи
  - Авторы *—* Книги

Парсинг

- Удаление пунктуации
- Разделение на слова
- Множественное восприятие?
- Специальная разметка?

rock’n’roll = rock n roll, rocknroll, rock’n’roll?
don’t = dont, don t, don’t, do not
re:Store = ???

Обработка стоп-слов

Стоп-слова — слова, не влияющие на релевантность

- the, a, in, at, …
- Частотные слова в конкретной коллекции
- to be or not to be = GNU
- Конфликтующие слова
  - «Продукты» — гипермаркеты или киоск у дома?
  - Имя в названии совпадает с именем автора?

Поиск

- Парсинг запроса
- Поиск в словаре
- Матчинг
- Постобработка
    - Ранжирование
    - Фильтрация
    - Обогащение выдачи
    - Любая бизнес-логика

Парсинг запроса

- Почти как в индексации, но…
  - Другой или отсутствующий язык разметки
    - «аквариум -группа»
  - Модификаторы запроса
    - «рядом»
  - Запрос — не обязательно только текст!
    - местоположение

Поиск в словаре

Словарь:

- Хеш-таблица
- Сжатый сортированный список слов
- Префиксное дерево
- Минимизированное префиксное дерево

https://habr.com/ru/post/190694/

Что мы ищем?

БАНКА

Что мы ищем?

ОТДЕЛЕНИЕ БАНКА **ОГУРЦОВ**

О морфологии

Слово «банка» соответствует формам разных лексем:

- Банк
- Банка

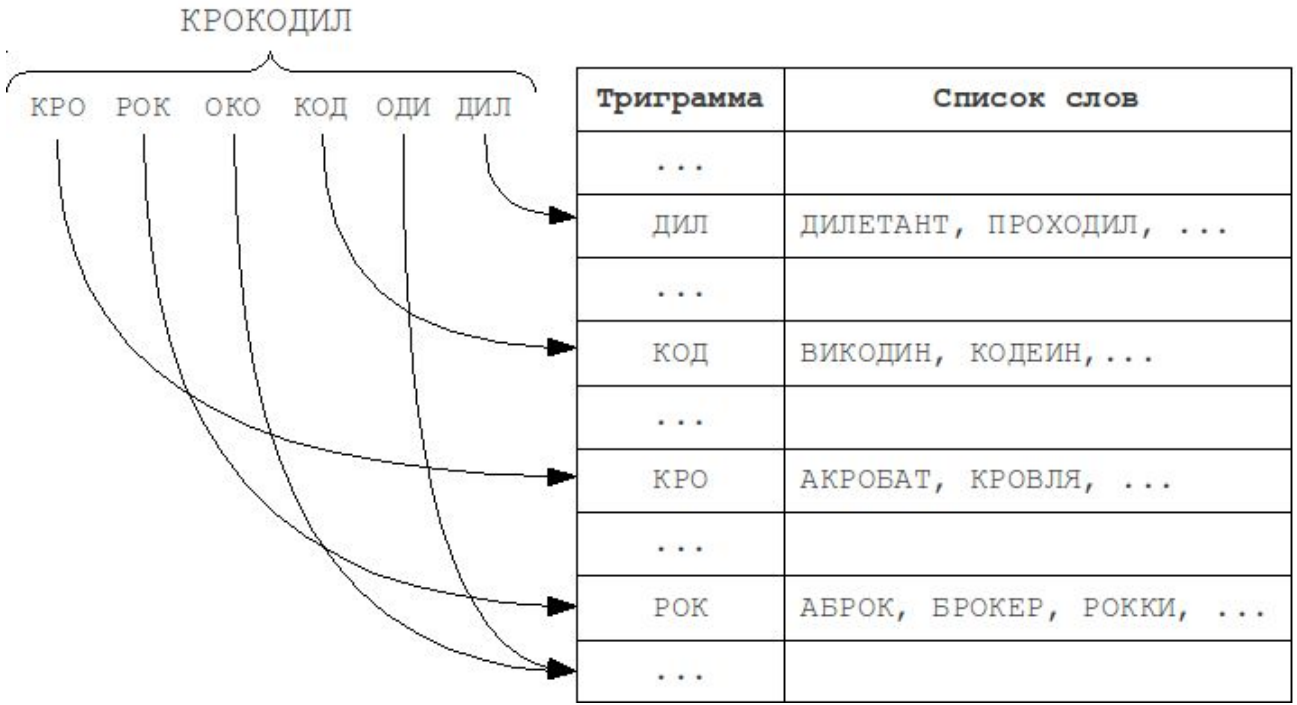
Стемминг — нахождение основы слова.
https://github.com/zvelo/libstemmer

Лемматизация — определение леммы слова.

Исправление опечаток

- Точное совпадение не гарантирует единственный результат
- Полезно знать больше, чем есть (маты)
- Вредно знать слишком много (редкие слова, про валерьянку)

Метод N-грам



Триграмма	Список слов
...	
ДИЛ	ДИЛЕТАНТ, ПРОХОДИЛ, ...
...	
КОД	ВИКОДИН, КОДЕИН,...
...	
КРО	АКРОБАТ, КРОВЛЯ, ...
...	
РОК	АБРОК, БРОКЕР, РОККИ, ...
...	

Опечатки

Примеры:

- Фонетические ошибки: «Цирк»
- Морфология: «AVANGARDEN 2.0»
- Ошибки в данных: «Продажа молотового кофе»
- PR-агентствам и ГАИ досталось
- Бывает и такое: https://go.2gis.com/bj68c

Матчинг

- Для каждого терма извлечение списка документов
- Объединение этих списков

Ранжирование

- Априорные веса
- Оценка покрытия текста

Априорные веса

В books.csv — средняя оценка и количество оценок.

Пример:

```
id  avg  count
1   4.8  10000
2   4.9   8000
3   1.2   4000
4   5.0     16
```

Как отсортировать?

Априорные веса

В books.csv — средняя оценка и количество оценок.

Пример:

```
id  avg  count
1   4.8  10000   4.8 * 10000 / 10000
2   4.9   8000   4.9 *  8000 /  8000
3   1.2   4000   1.2 *  4000 /  4000
4   5.0     16   5.0 *    16 /    16
```

Априорные веса

В books.csv — средняя оценка и количество оценок.

Пример:

```
id  avg  count
1   4.8  10000   (4.8 * 10000 + 15) / (10000 + 5)
2   4.9   8000   (4.9 *  8000 + 15) / ( 8000 + 5)
3   1.2   4000   (1.2 *  4000 + 15) / ( 4000 + 5)
4   5.0     16   (5.0 *    16 + 15) / (   16 + 5)
```

15 = 1 + 2 + 3 + 4 + 5

Априорные веса

В books.csv — средняя оценка и количество оценок.

Пример:

```
id  avg  count
1   4.8  10000   4.7991  2
2   4.9   8000   4.8988  1
3   1.2   4000   1.2022  4
4   5.0     16   4.5238  3
```

Оценка покрытия заголовка

TF/IDF

https://csc-cpp.readthedocs.io/ru/2022/s1/4-searcher.html

Постобработка

- Отсечение мусора
  - best_score * margin
- Раскрытие результатов
  - Если результат — автор книги
  - …актер на кинопоиске
  - …рубрика в справочнике

Качество поиска

- Производительность
- Релевантность

SEO

Search Engine Optimization

«Сколько нужно сеошников, чтобы вкрутить лампочку лампы накаливания осветительные приборы монтаж»

- Синонимы (старые или «народные» названия)
- Разметка значимости слов
- Уточняющие слова («поселок», «улица»…)

Корпус запросов и оценка качества

Корпус запросов

- В идеале — из пользовательской статистики
- Что, если ее нет?

Метрика качества: что учитывать?

Первичная грубая оценка

Генерируем запросы по данным

- Точное совпадение = ожидаем единственный результат
  - N — Штраф за мусор
  - M — Штраф за не первое место
  - K — Штраф за отсутствие ожидаемого результата в выдаче
- Генерируем запросы на основе точного:
  - Модификации заголовка: удаление, вставка и перестановка
  - С искусственными опечатками

Use Python, Luke!

Итог

- Поиск — простор для экспериментов
- Системно — модель (документы, слова) и обратный индекс
- Базовые детали — классическое программирование