Теория информационного поиска

Лекция 3. Расширения индекса

Дмитрий Грановский

СПбГУ

07.04.2025

В предыдущей серии

1. Просматривать всю коллекцию на каждый запрос слишком долго
2. Поэтому сделаем что-то похожее на алфавитный указатель и назовем (обратным) **индексом**
3. Левая часть индекса — словарь терминов, и он отсортирован
4. Правая часть индекса — списки словопозиций, которые тоже отсортированы
5. Теперь можно разбить запрос на термины и искать по пересечению (или объединению) терминов
6. Это называется **булев поиск** и с ним есть ряд проблем

Проблема расстояний

[московский кремль]

Проблема расстояний

[московский кремль]

Получаются одинаково релевантны документы:

- “**московский кремль** построен в конце XV века”
- “экспозиция «псковский **кремль**» по адресу: **московский** проспект…”

Проблема расстояний

[московский кремль]

Получаются одинаково релевантны документы:

- “**московский кремль** построен в конце XV века”
- “экспозиция «псковский **кремль**» по адресу: **московский** проспект…”

Это проблема модели мешка слов (*bag-of-words*).

N-граммный индекс

словарь	словопозиции
кремль	7, 62, 106, 115, 290, 304
московский	31, 62, 290, 300, 304
московский кремль	62, 290, 304
псковский	106, 109
псковский кремль	106

N-граммный индекс

словарь	словопозиции
кремль	7, 62, 106, 115, 290, 304
московский	31, 62, 290, 300, 304
московский кремль	62, 290, 304
псковский	106, 109
псковский кремль	106

- может быть отдельным индексом

N-граммный индекс

- обобщается на запросы длины >2:
  - [ремонт электронных часов] = [ремонт электронных] ∩ [электронных часов]
- проблемы:
  - распухает словарь
  - слишком жесткое ограничение: [суп из грибов] vs “суп из белых грибов”

Альтернативная идея

словарь	словопозиции
кремль	1: [2], 2: [2], 3: [1]
московский	1: [1], 2: [3]
псковский	2: [1]

1. “московский кремль построен xv”
2. “псковский кремль московский пр”
3. “кремль сообщает”

Альтернативная идея

- это **позиционный**, или **координатный индекс**
- теперь в запросах можно учитывать расстояние:
  - `[московский /+1 кремль]`
  - это называется *поиском в окне*
- также можно искать точные фразы
  - `["московский кремль построен"]`
  - это называется *фразовым поиском*
- как расставлять эти операторы в запросе?
  - ждать от пользователя (сомнительно)
  - пытаться автоматически выделять группы
    - например, синтаксические или named entities

Проблема структуры документа

- документ неоднороден, например:
  - заголовок и подзаголовки
  - ключевые слова
  - аннотация (если это статья)
  - автор
- какие-то части документа, возможно, важнее других
- назовем эти части **зонами**

Поиск с учетом зон

- присвоим каждой зоне вес w_i так, чтобы:
  - $\forall i : w_i \in (0, 1]$
  - $\sum w_i = 1$ (зачем?)
- добавим информацию о зоне в индекс
- при расчете релевантности будем считать вхождения в зону i с весом w_i

Поиск с учетом зон

кремль	1: [T:2, 17], 2: [20], 3: [T:1]
московский	1: [T:1, 16], 2: [3]
псковский	2: [1]

Поиск с учетом зон

кремль	1: [T:2, 17], 2: [20], 3: [T:1]
московский	1: [T:1, 16], 2: [3]
псковский	2: [1]

Допустим, вес заголовка $(T) = 0.8$

Поиск с учетом зон

кремль	1: [T:2, 17], 2: [20], 3: [T:1]
московский	1: [T:1, 16], 2: [3]
псковский	2: [1]

Допустим, вес заголовка $(T) = 0.8$

$$Rel(D_1, \text{кремль}) = 0.8 \times 1 + 0.2 \times 1 = 1.0$$

Поиск с учетом зон

кремль	1: [T:2, 17], 2: [20], 3: [T:1]
московский	1: [T:1, 16], 2: [3]
псковский	2: [1]

Допустим, вес заголовка $(T) = 0.8$

$$Rel(D_1, \text{кремль}) = 0.8 \times 1 + 0.2 \times 1 = 1.0$$

$$Rel(D_2, \text{кремль}) = 0.8 \times 0 + 0.2 \times 1 = 0.2$$

Поиск с учетом зон

кремль	1: [T:2, 17], 2: [20], 3: [T:1]
московский	1: [T:1, 16], 2: [3]
псковский	2: [1]

Допустим, вес заголовка $(T) = 0.8$

$$Rel(D_1, \text{кремль}) = 0.8 \times 1 + 0.2 \times 1 = 1.0$$

$$Rel(D_2, \text{кремль}) = 0.8 \times 0 + 0.2 \times 1 = 0.2$$

$$Rel(D_3, \text{кремль}) = 0.8 \times 1 + 0.2 \times 0 = 0.8$$

Атрибуты

Иногда понятно, что некоторые документы релевантнее других исходя из *метаданных*:

- совпадение языка с языком запроса
- более свежая дата
- более подходящий формат (например, .html vs .pdf)
- более короткий размер и т. д.

Атрибуты

Идея: добавим к документу некоторые пары ключ-значение и назовем их **атрибутами**:

- `date=2025-02-14`
- `lang=ru`
- `format=html`
- `size=100500*`
- `...`

Атрибуты

- можем использовать как для фильтрации, так и для ранжирования
- можем добавлять к запросу сами или предоставить пользователю дополнительные операторы:
  - `[Imagine /+1 Dragons` **`lang:ru`**`]`
  - `[расписание /+1 электричек` **`date:>2024-08-31`**`]`
- или добавить элементы интерфейса, модифицирующие запрос («расширенный поиск»)

Статический ранг

- идея: некоторые документы более ценны независимо от запроса, например:
  - находятся на «хороших» сайтах
  - грамотно написаны
  - их часто дочитывают до конца
- как использовать:
  - для дополнительного ранжирования
  - класть в начало постинга
  - чаще переиндексировать
  - лучше реплицировать и т. д.

Re: Проблемы

- обязательность всех терминов (AND) — не решили
- мешок слов — решили
- отсутствие ранжирования — начали решать
- не учитывается частота термина — не решили

Резюме

1. Проблему мешка слов можно решить с помощью N-граммного индекса.
2. Или с помощью координатного индекса.
3. Терминам в разных частях документа (зонах) могут быть присвоены различные веса.
4. Иногда полезно хранить для документа дополнительные атрибуты.
5. Некоторые документы могут быть полезнее других независимо от запроса.