Бази от данни

Пълнотекстово търсене

доц. д-р Димитър Димитров

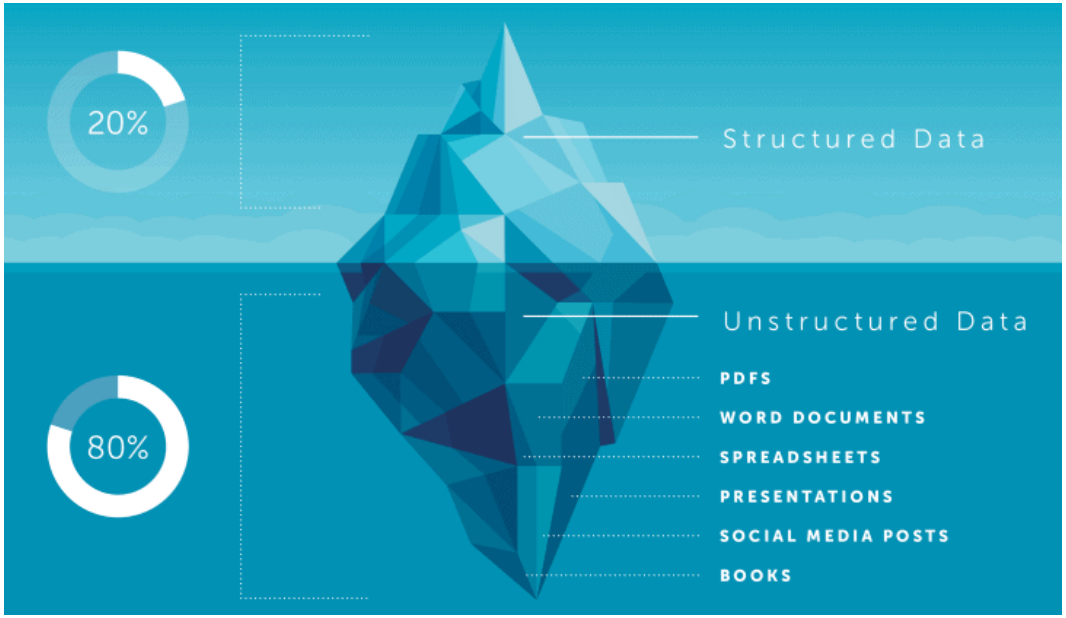
- Настоящата лекция е допълнителен материал, т.е. не е необходимо да се изучава за изпитите

Мотивация

- Поле за търсене на филми по ключови думи в UI
- В таблица имаме колона от текстов тип
- ```
  SELECT title, year
  FROM Movie
  WHERE title LIKE '%star%';
  ```
  - (Ако сравнението не е case-sensitive, трябва лека модификация)
- Стандартен индекс върху title ще ускори ли търсенето?
- Можем ли да модифицираме заявката, така че да изключим заглавия като StartUp и Mostar? – Можем
- А ако искаме да изключим StartUp, но да включим Starred Up?
  ```

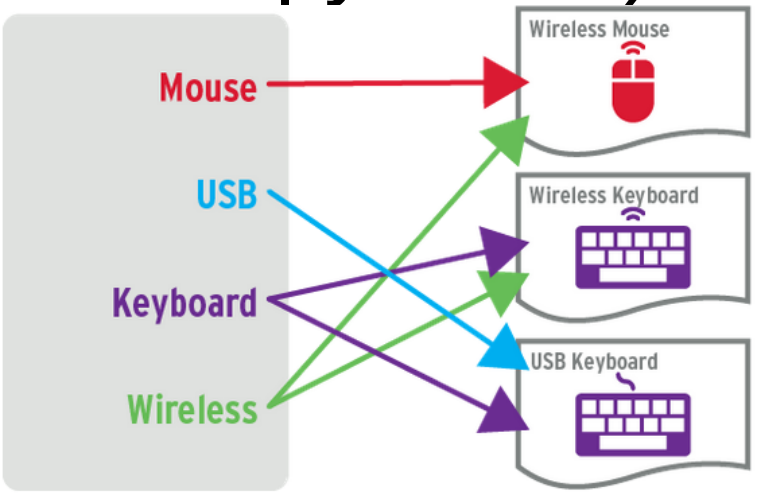
Проблемът е по-общ

- Търсене не само в БД, но и в произволна колекция от неструктурирани данни
- Уеб търсачки
- Етап от работата на някои алгоритми в ИИ



Решение

- Пълнотекстово търсене (Full-text search)
- Специални оптимизирани за целта индекси
- Специални оператори и функции в SQL (или външни инструменти)



Видове

- Просто – търсене по ключови думи
- Булево – Star AND NOT Wars
- Размито (fuzzy) – намира срещания с печатни грешки, алтернативно изписване и т.н.
- Wildcard – Star* ще намери StartUp, Stardust, …
- По фраза
- И други

Индекси за пълнотекстово търсене

- Грубо казано, текстът се разбива на цели думи
- По-точно – на tokens (официален превод: жетони)
  - Най-малка единица за търсене
  - За разлика от LIKE
- За всеки token се пази списък от местоположенията му в индексираната колона/и

Токенизация

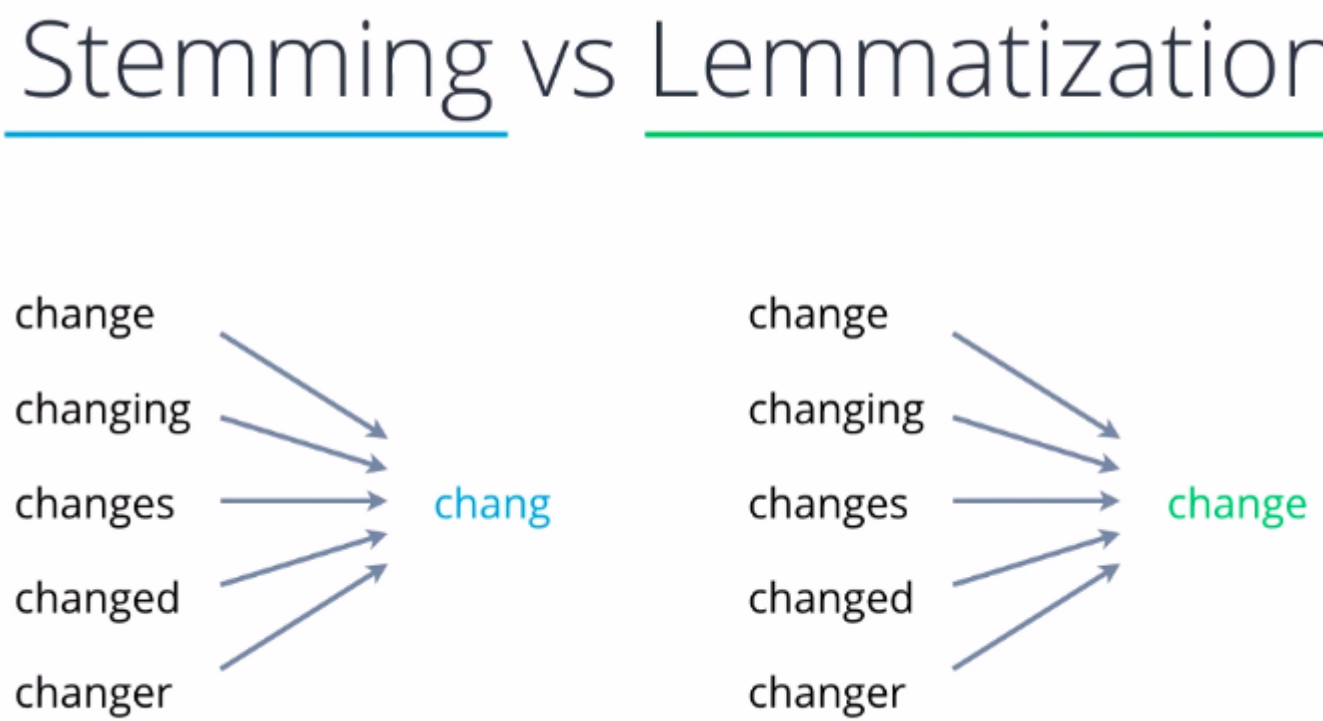
- Множество предизвикателства
  - Не е просто **`string.split(" ")`**
- Премахване на пунктуация
- Тирета и апострофи
  - Twenty-five – един token
  - Project-based – два token-а
  - It’s (it is), Maria’s
- Числа – също са token-и
- Дати, телефонни номера – различни формати
- Езици, различни от английския
  - Katzenpfotenballenhimbeermarmeladenglastunker – много token-и
- Има различни имплементации, няма перфектно решение

Филтриране при токенизация

- Филтриране
  - Напр. на HTML тагове
  - Стоп думи (“a”, “the”) – често срещани, не носят съществен смисъл
  - Заместване на диакритични знаци (ä → a, ae)

Нормализация

- Заместване с корен (може да не е истинска дума) или с лема



Структури от данни

- Инвертиран индекс (Inverted index)
  - Най-често
  - Съответсвие от token-и към местоположения
  - Бавно добавяне на нов запис/документ
  - (обратното на Forward index – от документ/местоположения към ключови думи)
  - https://cs.stackexchange.com/questions/130814/why-is-the-inverted-index-called-so-and-not-simply-index

Изпълнение на заявка

- Подобно на стандартните индекси
  - в СД на индекса, а не в самите данни
- Заявката би могла да има булеви оператори и др.
- Може да се прилага размито търсене и др.

Scoring and ranking

- Вероятно заявката намира много записи/документи
- Оценка на релевантността на всеки намерен резултат
  - Колко често се среща търсената дума в даден резултат?
  - Нормализация на дължината на резултатите, понеже при по-дълъг е по-вероятно да има по-висока честота на срещане
  - Търсената дума рядко срещана ли е по принцип в индекса?
  - Приближеност на търсените думи в резултата
  - Къде се среща търсената дума в резултата – в заглавие или в съдържанието?
  - Машинно обучение
- Сортиране на резултите
- Таблицата с резулти може да включва и изчислените оценки за всеки резултат

Поддръжка

- Можем да използваме пълнотекстово търсене:
  - Със средствата, предоставени ни от СУБД (ако има такива), или
  - С външен инструмент, който достъпва БД
    - Sphinx
    - Solr
    - Elasticsearch

Въпроси?

В презентацията са използвани материали, копирани от презентации на проф. д-р Калинка Калоянова