Информационный поиск в Интернете

Павел Морозов
pmorozov@bellintegrator.ru

План лекции

- Модели информационного поиска
  - Булевская модель
  - Векторная модель
  - Вероятностная модель

План лекции

- Модели информационного поиска
  - Булевская модель
  - Векторная модель
  - Вероятностная модель
- Архитектура поисковой системы

План лекции

- Модели информационного поиска
  - Булевская модель
  - Векторная модель
  - Вероятностная модель
- Архитектура поисковой системы
- PageRank

Модели информационного поиска

Что такое документ?

Что такое запрос?

При каком условии документ соответствует запросу?

Булевская модель

Словарь: $T = \{t_1, \ldots t_n\}$

Документ: $D \subset T$, иначе говоря $D \in \{0, 1\}^n$

Запрос: t_5 OR t_7 NOT t_{12}

Булевская модель

Словарь: $T = \{t_1, \ldots t_n\}$

Документ: $D \subset T$, иначе говоря $D \in \{0, 1\}^n$

Запрос: t_5 OR t_7 NOT t_{12}

Соответствие:

Формула запроса должна быть выполнена на документе.

Булевская модель

Словарь: $T = \{t_1, \ldots t_n\}$

Документ: $D \subset T$, иначе говоря $D \in \{0, 1\}^n$

Запрос: t_5 OR t_7 NOT t_{12}

Соответствие:

Формула запроса должна быть выполнена на документе.

Недостатки модели?

Векторная модель

Снова коллекция документов, каждый из которых теперь является мультимножеством слов.

Определим матрицу M по формуле $M_{ij} = TF_{ij} \cdot IDF_i$, где:

- Частота терма TF_{ij} — относительная доля слова i в тексте j
- Обратная встречаемость в документах IDF_i — величина, обратная количеству документов, содержащих слово i

Векторная модель

Снова коллекция документов, каждый из которых теперь является мультимножеством слов.

Определим матрицу M по формуле $M_{ij} = TF_{ij} \cdot IDF_i$, где:

- Частота терма TF_{ij} — относительная доля слова i в тексте j
- Обратная встречаемость в документах IDF_i — величина, обратная количеству документов, содержащих слово i

Физический смысл M_{ij} — степень соответствия слова i тексту j

Векторная модель

Снова коллекция документов, каждый из которых теперь является мультимножеством слов.

Определим матрицу M по формуле $M_{ij} = TF_{ij} \cdot IDF_i$, где:

- Частота терма TF_{ij} — относительная доля слова i в тексте j
- Обратная встречаемость в документах IDF_i — величина, обратная количеству документов, содержащих слово i

Физический смысл M_{ij} — степень соответствия слова i тексту j

Запрос: t_3 AND t_5 (разрешаем только AND)

Релевантность в векторной модели

Запишем запрос в виде вектора:

$Q = t_3 \text{ AND } t_5 \sim \{0, 0, 1, 0, 1, 0, \ldots, 0\}$

Мерой релевантности будет **косинус** между запросом и документом:

$$R(Q, D) = \frac{Q \cdot D}{|D||Q|}$$

Вероятностная модель
для чайников

Документ: множество слов (булевский вектор) $D = \{d_1, \ldots, d_n\}$

Запрос: Q_k — тоже, но храним как множество

Вероятностная модель
для чайников

Документ: множество слов (булевский вектор) $D = \{d_1, \ldots, d_n\}$

Запрос: Q_k — тоже, но храним как множество

Соответствие:

- Зафиксируем запрос Q_k
- Пусть есть распределение вероятностей на все $P(R|Q_k, D)$ "быть релевантным запросу Q_k": обозначаем
- Пусть есть распределение вероятностей на всех $P(\bar{R}|Q_k, D)$ "быть НЕрелевантным запросу Q_k": обозначаем
- Функцией со $\frac{P(R|Q_k,D)}{P(\bar{R}|Q_k,D)}$ я будет их отношение (или логарифм этой дроби):

Вычисляем функцию соответствия

Воспользуемся теоремой Байеса ($P(a|b) = P(b|a)\frac{P(a)}{P(b)}$

Вычисляем функцию соответствия

Воспользуемся теоремой Байеса ($P(a|b) = P(b|a)\frac{P(a)}{P(b)}$

$$\frac{P(R|Q_k, D)}{P(\bar{R}|Q_k, D)} = \frac{P(R|Q_k)}{P(\bar{R}|Q_k)} \frac{P(D|R, Q_k)}{P(D|\bar{R}, Q_k)}$$

Первый сомножитель одинаков для всех документов.

Вычисляем функцию соответствия

Воспользуемся теоремой Байеса ($P(a|b) = P(b|a)\frac{P(a)}{P(b)}$

$$\frac{P(R|Q_k, D)}{P(\bar{R}|Q_k, D)} = \frac{P(R|Q_k)}{P(\bar{R}|Q_k)} \frac{P(D|R, Q_k)}{P(D|\bar{R}, Q_k)}$$

Первый сомножитель одинаков для всех документов.

Предполагая независимость всех слов, второй сомножитель можно представить как произведение:

$$\prod_{i=1}^{n} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\bar{R}, Q_k)}$$

Вычисляем функцию соответствия II

Введем обозначения: $\prod_{i=1}^{n} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\bar{R}, Q_k)}$

$$p_{ik} = P(x_i = 1|R, Q_k)$$

$$q_{ik} = P(x_i = 1|\bar{R}, Q_k)$$

Предположим, что для каждого слова i , не входящего в запрос,

$$p_{ik} = q_{ik}$$

Вычисляем функцию соответствия II

Введем обозначения: $\prod_{i=1}^{n} \frac{P(x_i = d_i|R, Q_k)}{P(x_i = d_i|\bar{R}, Q_k)}$

$$p_{ik} = P(x_i = 1|R, Q_k)$$
$$q_{ik} = P(x_i = 1|\bar{R}, Q_k)$$

Предположим, что для каждого слова i , не входящего в запрос,

$$p_{ik} = q_{ik}$$

Теперь мы можем переписать нашу дробь:

$$\prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}}$$

Вычисляем функцию соответствия III

$$\prod_{i \in Q_k \cap D} \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \prod_{i \in Q_k} \frac{1 - p_{ik}}{1 - q_{ik}}$$

Вычисляем функцию соответствия III

$$\prod_{i \in Q_k \cap D} \frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})} \prod_{i \in Q_k} \frac{1-p_{ik}}{1-q_{ik}}$$

Второй сомножитель одинаков для всех документов.

Забудем про него и возьмем логарифм от первого:

$$\sum_{i \in Q_k \cap D} c_{ik}, \qquad \text{где } c_{ik} = \log \frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}$$

Подбор параметров

$$\sum_{i \in Q_k \cap D} c_{ik}, \qquad \text{где } c_{ik} = \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}$$

Для использования полученной формулы нужно знать p_{ik} и q_{ik} .

Подбор параметров

$$\sum_{i \in Q_k \cap D} c_{ik}, \qquad \text{где } c_{ik} = \log \frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}$$

Для использования полученной формулы нужно знать p_{ik} и q_{ik} .

Рецепт: пусть у нас уже есть некий набор текстов, про которые мы знаем, релевантны они запросу Q_k или нет. Тогда мы можем использовать формулы:

$$p_{ik} = \frac{r_i}{r} \quad \text{и} \quad q_{ik} = \frac{f_i - r_i}{f - r},$$

Подбор параметров II

Тут $$p_{ik} = \frac{r_i}{r} \quad \text{и} \quad q_{ik} = \frac{f_i - r_i}{f - r},$$

f — общее число документов,

r — число релевантных документов,

r_i число релевантных документов, содержащих слово i ,

f_i — общее число документов со словом i .

Архитектура поисковой системы

В каком формате запоминать интернет-страницы?

В какой структуре данных их хранить?

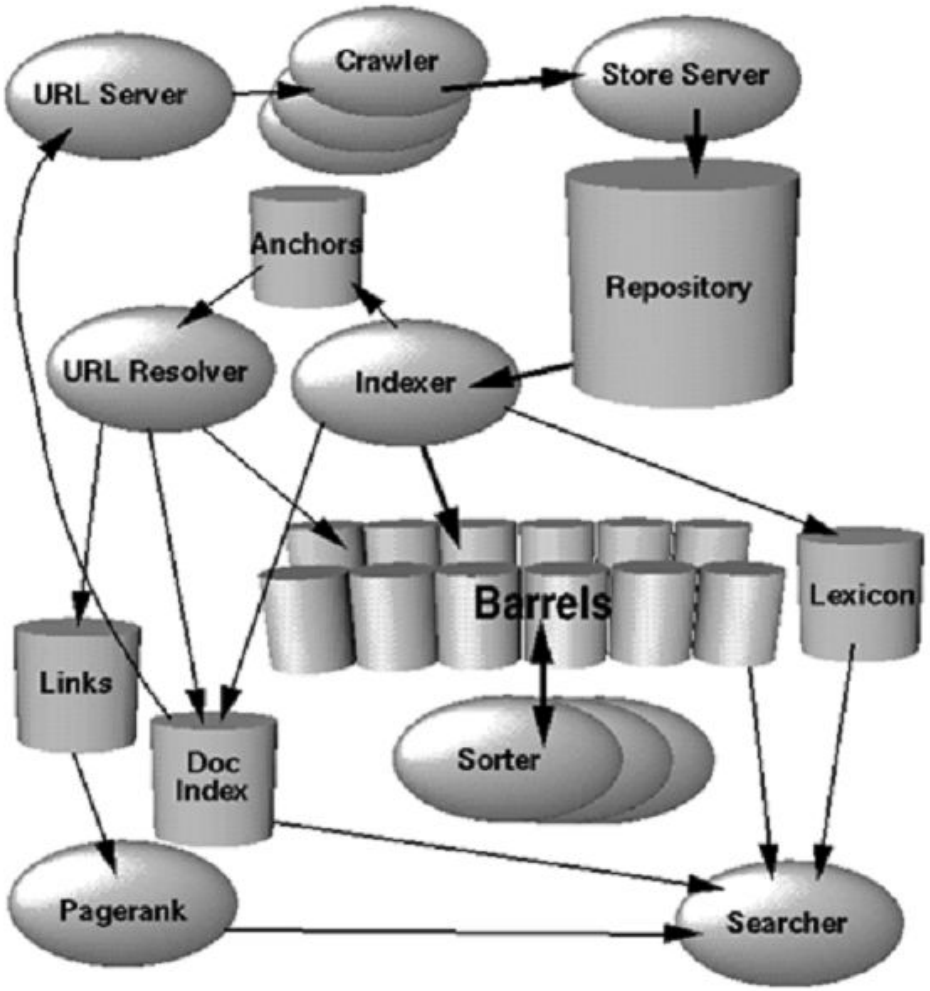
Как обрабатывать запрос пользователя?

Анатомия поисковой системы

Любая поисковая система содержит три базовые части:

- Робот (он же краулер, спайдер или индексатор)
- Базы данных
- Клиент (обработка запросов)

Схема из [Brin,Page, 1998]



Прямой и обратный индекс

Прямой индекс — записи отсортированы по документам

- Номер документа
- Отсортированный список слов
- Для каждого слова: первые несколько вхождений, частота вхождений, формат вхождений

Обратный индекс — записи отсортированы по словам

- Номер слова
- Отсортированный список документов
- Для каждого документа: информация о вхождении

Релевантность

- Наличие слов на сайте
- Частота слов
- Форматирование
- Близость слов друг к другу
- Количество ссылок с других страниц на данную
- Качество ссылок
- Соответствие тематик сайта и запроса
- Регистрация в каталоге, связанном с поисковой системой

Как работает клиент?

- Разбирает запрос на слова
- Переводит слова в их идентификаторы
- Для каждого слова находит в обратном индексе список документов, его содержащих
- Одновременно бежит по этим спискам, ища общий документ
- Для каждого найденного документа вычисляет степень релевантности
- Сортирует образовавшийся список по релевантности

Качество поиска

- **Полнота:** отношение количества найденных релевантных документов к общему количеству релевантных документов
- **Точность:** доля релевантных документов в общем количестве найденных документов
- **Benchmarks:** показатели системы на контрольных запросах и специальных коллекциях документов
- **Оценка экспертов**

PageRank

- Как определить ссылочную популярность страницы (PageRank)?
- Как быстро вычислить приближение PageRank?

PageRank: постановка задачи

Хотим для каждой страницы сосчитать показатель ее “качества”.

Идея [Брин, 1998]: Определить рейтинг страницы через количество ведущих на нее ссылок и рейтинг ссылающихся страниц

Другие методы:

- Учет частоты обновляемости страницы
- Учет посещаемости
- Учет регистрации в каталоге-спутнике поисковой системы

Модель случайного блуждания

Сеть:

- Вершины
- Ориентированные ребра (ссылки)

Передвижение пользователей по сети

- Стартуем в случайной вершине
- С вероятностью ε переходим в случайную вершину
- С вероятностью $1 - \varepsilon$ переходим по случайному исходящему ребру

Предельные вероятности

Для каждого k можно определить $PR_k(i)$ как вероятность оказаться в вершине i через k шагов

$$\lim_{k \to \infty} PR_k(i) = PR(i)$$

то есть для каждой вершины есть предельная вероятность находится именно в ней

Основное уравнение PageRank

Пусть $T1, \ldots, Tn$ — вершины, из которых идут ребра в i , C(X) — обозначение для исходящей степени вершины X.

$$PR(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}$$

По определению $PR_k\ (i\ )$ верно следующее:

$$PR_0(i) = 1/N$$
$$PR_k(i) = \varepsilon/N + (1-\varepsilon)\sum_{i=1}^{n} \frac{PR_{k-1}(T_i)}{C(T_i)}$$

Нужно перейти к пределу!

Практическое решение: вместо $PR(i\ )$ используют $PR_{50}(i\ )$, вычисленное по итеративной формуле.

Вопросы?

pmorozov@bellintegrator.ru