ТЕХНОСФЕРА

Лекция
Классические модели текстовой релевантности

Владимир Гулин

октябрь 2020 г.

План лекции

Мотивация

Векторная модель ранжирования

Вероятностная модель ранжирования

Ранжированный поиск

- Поиск, который мы видели до этого был булевым
  - Документ либо подходит, либо нет
- Хорошо подходит для продвинутых экспертов, которые хорошо понимают что им нужно и что они могут найти в коллекции документов
  - Также хорошо подходит для приложений (анализ тысяч результатов)
- Плохо для большинства пользователей
  - Обычный пользователь никогда не будет писать булев запрос
  - Большинство пользователей не хотят просматривать тысячи результатов поиска

Проблемы булева поиска

- Булевы запросы часто возвращают либо слишком мало (=0) либо слишком много результатов (тысячи)
- Запрос 1: “скачать бесплатно” (4 млн. результатов)
- Запрос 2: “скачать бесплатно без регистрации без смс без ключа без хурмы без кидалова без торрента” (0 результатов)
- От пользователя это требудет соответсвующего навыка, чтобы составить запрос, который вернул бы приемлимое количество результатов
  - AND приводит к малому числу результатов
  - OR к слишком большому

Модели ранжированного поиска

- Ранжированный поиск возвращает упорядоченный список документов из коллекции по запросу
- Вместо языка запросов и операторов, используются просто слова из человеческого языка

Вопрос:

- Применяется ли в современных поисковых системах булев поиск?

Схема ранжирования в поиске

Этапы ранжирования

Релевантность

Прежде чем начать...

Вопрос:

- Что такое релеватный документ?

Релевантность

Сложное комплексное понятие, учитывающее множество факторов и зачастую крайне субьективное

В информационном поиске релеватность рассматривается с нескольких сторон

- тематическая релеватность
- пользовательская релеватность
- текстовая релеватность
- ...

Ранжированный поиск

Когда поисковая система возвращает ранжированный список результатов, большой объем не является проблемой

- Количество найденных результатов не является проблемой для пользователя
- Мы показываем только top $k(\approx 10)$ результатов
- Таким образом, мы не огорчаем пользователя

Предположение:

- Алгоритм ранжирования работает хорошо :)

Ранжированный поиск

Оценка релеватности документа

- Хотим вернуть документы, в порядке наиболее полезных для пользователя
- Каким образом мы можем составить такой порядок в соотвествии с запросом?
- Назначим каждому документу оценку (score) для каждого документа по запросу (например из $[0, 1]$)
- Эта оценка должна отражать на сколько хорошо документ подходит запросу

Вычисление веса

- Нужен способ назначения веса паре запрос-документ
- Начнем с запроса из одного термина
- Если термина нет в документе, то вес равен 0
- Чем чаще встречается термин в документе, тем выше вес

Модель мешка слов

- Не учитывается порядок слов в документе
- John is quicker than Mary и Mary is quicker than John
- Вася быстрее Маши и Маша быстрее Васи
- НО Мать любит дочь и Дочь любит мать (не ясно кто кого любит)
- Такая модель называется моделью мешка слов
- Это шаг назад, так как координатный индекс может различить такие докуенты
- Вернемся к использованию координатной информации позже

Частота термина

- Частота $tf_{t,d}$ термина t в документе d определяется как количество раз, сколько t встречается в d
- Хотим использовать tf при расчете весов. Но как?
- Просто частота не торт!
  - Документ с 10 вхождениями релевантнее документа с 1 вхождением (в 10 раз!!!).
- Релеватность не увеличивается пропорционально частоте

Логарифмическое взвешивание

- Логарифмическая частота термина t в d:

$$w_{t,d} = \begin{cases} 1 + log\ tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- Вес для пары запрос-документ: сумма по всем терминам t, входящим в q и d:
- Вес

$$= \sum_{t \in q \cap d} (1 + log\ tf_{t,d})$$

- Вес равен 0, если в документе нет ни одного термина из запроса.

Документная частота

- Редкие термины информативнее частотных (стоп-слова)
- Рассмотрим термин запроса, который редко встречается в корпусе (например *“серобуромалиновый”*)
- Любой документ, содержащий в себе этот термин, скорее всего будет релевантен запросу *“серобуромалиновый”*
- То есть, имеет смысл давать больший вес редким терминам

Обратная документная частота

IDF

- df_t - документная частота термина t (количество документов, содержащих t)
- df_t - обратная мера инфортивности t
- $df_t \leq N$
- Опредилим idf (inverse document frequency) термина как

$$idf_t = log \frac{N}{df_t}$$

Пример idf

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

$$idf_t = log \frac{N}{df_t}$$

- $N = 10^6$
- Для каждого термина в корпусе только одно значение idf

IDF в ранжировании

Значим ли IDF для запросов для одного термина

- IPhone

IDF не влияет на однословные запросы

- Idf влияет на ранжирование запросов из двух и более слов
- Для запросов типа “серобуромалиновые штаны”, взвешивание по IDF приводит к большому вкладу термина “серобуромалиновый”, чем термин “штаны’

Взвешивание TF-IDF

- Вес термина $tf - idf$ это произведение его весов tf и idf:

$$w_{t,d} = (1 + log\ tf_{t,d}) \cdot log \frac{N}{df_t}$$

- Самая известная модель взвешивания в информационом поиске
- Растет с ростом числа вхождений слова в документ
- Растет со степенью редкости термина

Ранжирование по TF-IDF

$$Score(q, d) = \sum_{t \in q \cap d} tf.idf_{t,d}$$

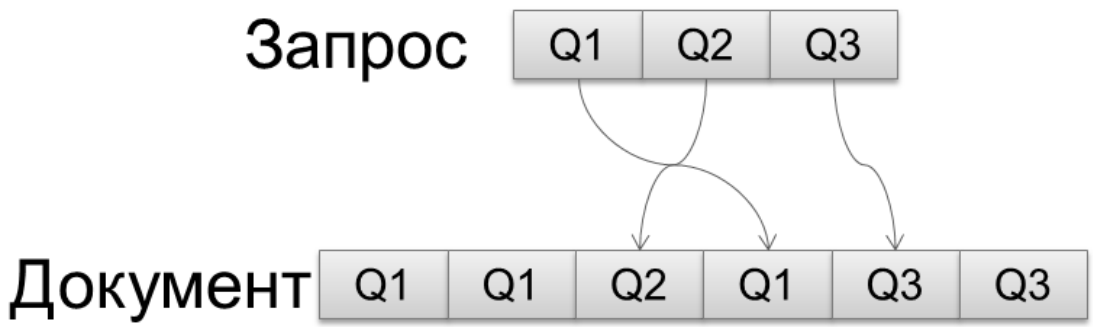
N-gramm TF-IDF

- Ничего не мешает нам аналогично с вхождениями слов рассматривать вхождения n-грамм
- А еще мы их можем взять с разным весом

$$Score(q, d) = \sum_{n} \alpha_n \sum_{t_n \in q \cap d} tf.idf_{t_n, d}$$

Пассажный алгоритм

Пассаж - фрагмент документа, размера, не превышающего заданный, в котором встречаются все термы запроса, либо значительная часть термов запроса, суммарный IDF которых превышает заданное ограничение.



Пассажный алгоритм

Оценка пассажа

- TF-IDF
- Полнота
- Порядок слов
- Правильность словоформ
- Кучность
- Близость к началу
- Особенность зоны документы

Параметрические зоны и индексы

- До сих пор документ представлялся последовательностью терминов
- Обычно документы состоят из нескольких частей, с определённой семантикой
  - Автор
  - Заголовок
  - Дата публикации
  - Язык
  - Формат
  - И т.п.
- Это все метаданные

Компактность вхождений

- Текстовые запросы: набор терминов, введённых в поисковую строчку
- Пользователи предпочитают документы, где термины запроса находятся на небольшом расстоянии друг относительно друга
- Пусть w будет наименьшим окном в документе, содержащим все термины запроса
- Например для запроса [strained mercy] такое окно в документе The quality of mercy is not strained равно 4 (в словах)
- Как учесть это в итоговом scrore?

Как объединить все вместе?

$$Score(q, d) = \sum_n \alpha_n \sum_{t_n \in q \cap d} tf.idf_{t_n,d} + \sum_p score_p(q, d)$$

- Линейная модель
- Первое слагаемое n-граммный TF-IDF
- Второе слагаемое взвешенная сумма пассажных ранков

$$score_p(q, d) = score(pos, proximity, tf.idf, zone \ldots)$$

Векторная модель ранжирования

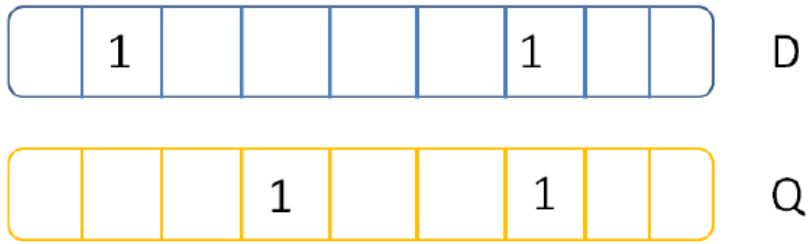
Документы и запросы - это вектора в T мерном пространтсве, где T - общее количетсво термов (словоформ, основ, фраз и т.д.)

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{iT}) \quad Q = (q_1, q_2, \ldots, q_T)$$

Коллекция документов предствляется матрицей.

	$Term_1$	$Term_2$	$\ldots$	$Term_t$
Doc_1	d_{11}	d_{12}	$\ldots$	d_{1t}
Doc_2	d_{21}	d_{22}	$\ldots$	d_{2t}
$\vdots$	$\vdots$			
Doc_n	d_{n1}	d_{n2}	$\ldots$	d_{nt}

Векторная модель ранжирования



Документы ранжируются в соответсвии с схожестью вектора соответствующего запросу и вектора соотвествующего документу

$$cosine(Q, D) = \frac{Q^T D}{\|Q\|\|D\|}$$

Вопрос:

- А почему именно косинус?

Векторная модель

- ✔ простая модель ранжирования
- ✔ можно использовать любую меру схожести векторов
- ✔ можно использовать любую схему взвешивания термов
- ✖ Модель работает в предположении о независимости термов
- ✖ Невозможно определить способ оптимального ранжирования

Вероятностная модель ранжирования

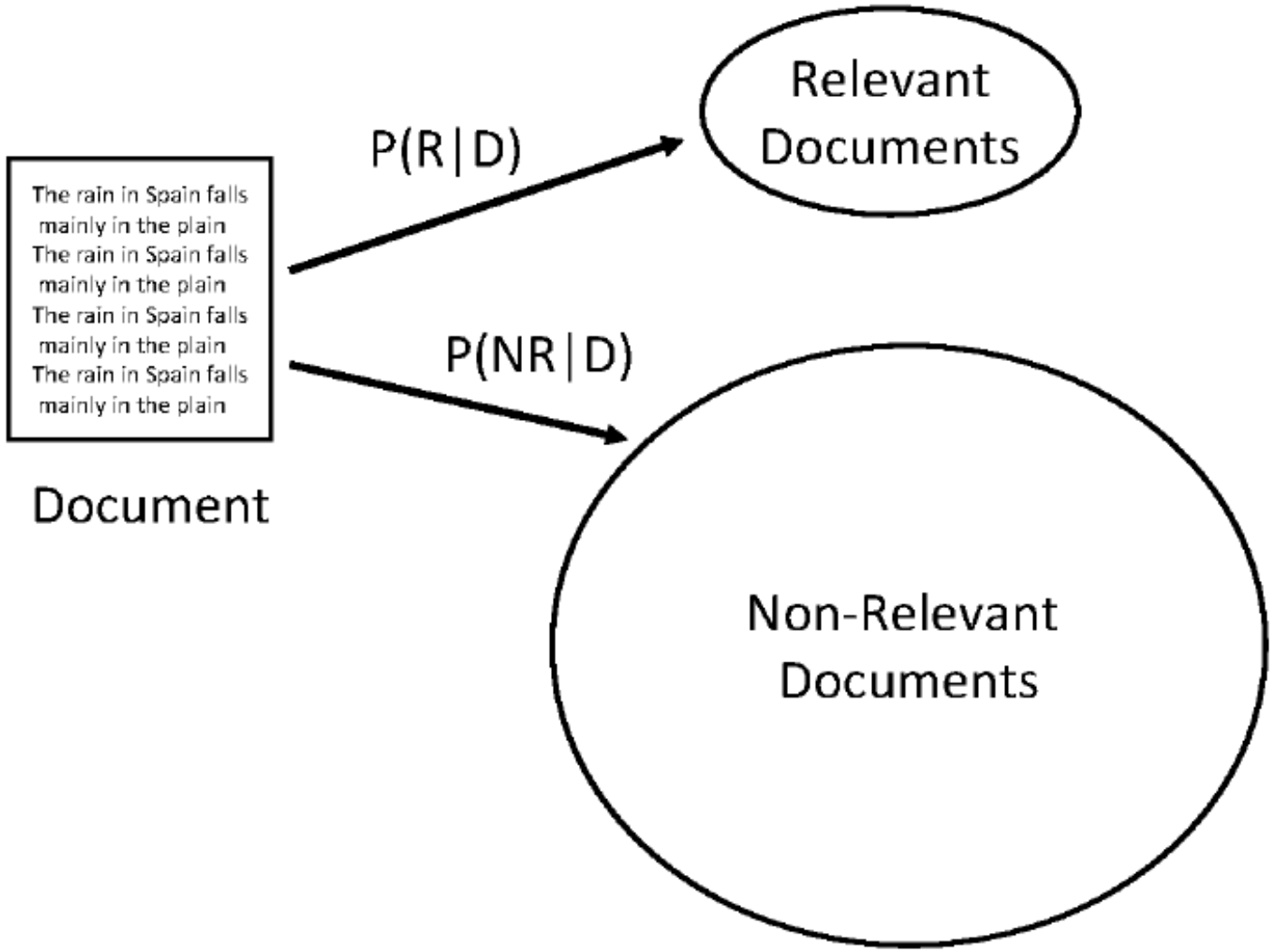
Принцип вероятностного ранжирования

- Если поисковая система в ответ на запрос пользователя отдает документы в порядке уменьшения вероятности их релеватности запросу, где веротяности оценены как можно более точно на основе доступных данных, то качество такой поисковой системы будет максимальным на этих данных (Robertson, 1977)

Ключевой вопрос:

- Какова вероятность того, что пользователь оценит данный документ как релеватный для этого запроса?

Решаем задачу бинарной классификации



Байесовский классификатор

Оптимальное решающее правило

- Документ D релевантен запросу Q, если $P(R = 1|D, Q) > P(R = 0|D, Q)$

Оценка вероятностей

- Воспользуемся формулой Байеса

$$p(R|D) = \frac{P(D|R)P(R)}{P(D)}$$

Теорема

Принцип вероятностоного ранжирования ясвляется оптимальным в том смысле, что он минимизирует ожидаемые потери (байсесовский риск) в рамках модели бинарных потерь (доля найденных нерелевтных документов)

Модель бинарной независимости

- Документ представляет собой бинарный вектор термов
- Запрос представляет собой бинарный вектор термов
- Предположение о независимости появления термов в документе

$$P(D|R) = \prod_{i=1}^{t} P(d_i|R)$$

Модель бинарной независимости

p_i - вероятность встретить i-ый термин в релевантных документах

s_i - вероятность встретить i-ый термин в нерелевантных документах

$$\frac{P(D|R)}{P(D|NR)} = \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i} =$$

$$= \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \left(\prod_{i:d_i=1} \frac{1-p_i}{1-s_i} \cdot \prod_{i:d_i=1} \frac{1-s_i}{1-p_i} \right) \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i} =$$

$$= \prod_{i:d_i=1} \frac{p_i(1-s_i)}{s_i(1-p_i)} \cdot \prod_{i} \frac{1-p_i}{1-s_i}$$

Модель бинарной независимости

- Оценка отношения правдоподобия
- RSV_d - Retrieval Status Value

$$RSV_d = \sum_{i:d_i=1} log \frac{p_i(1-s_i)}{s_i(1-p_i)}$$

- Запрос несет информацию о релеватных документах
- Если нет никакой дополнительной информации, то пологаем, что p_i мала и постоянна, а s_i можно оценить по коллекции

$$log \frac{p_i}{1-p_i} + \frac{1-s_i}{s_i} = log \frac{(1-\frac{n_i}{N})}{\frac{n_i}{N}} = log \frac{N-n_i}{n_i}$$

Модель бинарной независимости

Честная оценка

	Relevant	Non-relevant	Total
$d_i = 1$	r_i	$n_i - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - r_i$
Total	R	$N - R$	N

$$p_i = (r_i + 0.5)/(R + 1)$$

$$s_i = (n_i - r_i + 0.5)/(N - R + 1)$$

- Функция оценки:

$$\sum_{i:d_i=q_i=1} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}$$

BM25

Классический алгоритм ранжирования, основанный на модели бинарной независимости

$$\sum_{i \in Q} log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1 + 1)f_i}{K + f_i} \cdot \frac{(k_2 + 1)qf_i}{k_2 + qf_i}$$

- значения k_1, k_2, K подбираются эмпирически

$$K = k_1((1 - b) + b \cdot \frac{dl}{avgdl})$$

- На TREC $k_1 = 1.2$, $k_2 \in [0, 1000]$, $b = 0.75$

BM25

Пример:

- Запрос: “president lincoln” (qf $= 1$)
- $r = R = 0$ (нет информации о релевантных документах)
- Размер коллекции $N = 500000$ документов
- Термин “president” содержится в 40000 документов ($n_1 = 40000$)
- Термин “lincoln” содержится в 300 документов ($n_2 = 300$)
- Термин “president” встречается 15 раз в документе ($f_1 = 15$)
- Термин “lincoln” встречается 25 раз в документе ($f_2 = 25$)
- $\frac{dl}{avdl} = 0.9$
- $k_1 = 1.2$, $b = 0.75$, $k_2 = 100$

BM25

Пример:

$$BM25(Q, D) =$$

$$= log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(40000 - 0 + 0.5)/(500000 - 40000 - 0 + 0 + 0.5)} \times \frac{15(1.2 + 1)}{1.11 + 15} \times \frac{1(100 + 1)}{100 + 1} +$$

$$+ log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(300 - 0 + 0.5)/(500000 - 300 - 0 + 0 + 0.5)} \times \frac{25(1.2 + 1)}{1.11 + 25} \times \frac{1(100 + 1)}{100 + 1} =$$

$$= log \left[\frac{460000.5}{40000.5} \cdot \frac{33}{16.11} \cdot \frac{101}{101}\right] + log \left[\frac{499700.5}{300.5} \cdot \frac{55}{26.11} \cdot \frac{101}{101}\right] = 20.66$$

BM25F

BM25

$$score(Q, D) = \sum_{i=1}^{n} Idf(q_i) \frac{f(q_i, D)(k_1 + 1)}{f(q_i, D) + k_1(1 - b + b\frac{dl}{avgdl})}$$

BM25F

$$score(Q, D) = \sum_{i=1}^{n} Idf(q_i) \frac{\sum_E rank(E)(k_1 + 1)}{\sum_E rank(E) + k_1(1 - b + b\frac{dl}{avgdl})}$$

$rank(E)$ - функция взвешивания вхождения слова в документ (зависит от зоны, позиции и т.д.)

Вопросы