

Математические методы анализа текстов

Семинар 6

Информационный поиск.

Мурат Апишев (great-mel@yandex.ru) ¹

МГУ им. М. В. Ломоносова

20 апреля, 2018

¹Подготовлена на основе лекций **Алексея Шаграева** и **К.В. Воронцова** 

Содержание занятия

- ▶ Понятие информационного поиска, инвертированный индекс
- ▶ Обучение ранжированию в информационном поиске
- ▶ Оценивание качества поиска и ранжирования
- ▶ Разнообразие поисковой выдачи
- ▶ Разведочный поиск

Информационный поиск

Информационный поиск (Information retrieval) — наука о методах поиска неструктурированной информации, удовлетворяющей заданным параметрам

Типичный пример — поиск документов по коллекции, удовлетворяющих заданному запросу

Дано множество документов и множество поисковых запросов. Требуется для каждого запроса предоставить множество наиболее релевантных ему документов

Релевантность — степень смыслового соответствия документа поисковому запросу

Ранжирование — процесс назначения документам степени релевантности с последующей сортировкой

Информационный поиск

Прежде всего, необходимо решить задачу поиска документов

Простой подход — искать документы по словам запроса, используя различные алгоритмы поиска подстроки в строке

— **на практике неприменимо**

Современный подход (последние 20 лет) — *инвертированный индекс*

- ▶ Выделяем признаки из документов (униграммы, N-граммы и т.п.)
- ▶ Для каждого признака строим указатели на документы, в которых они имеют ненулевой вес

Готовый поисковый движок для своих проектов — **Apache Solr**

Задача ранжирования

- ▶ Пусть есть множество документов D и множество запросов Q
- ▶ Для некоторого запроса $q \in Q$ найдено множество документов \Rightarrow имеем выборку пар (d, q)
- ▶ Y — упорядоченное множество рейтинг, более высокое значение соответствует более высокой релевантности
- ▶ $y : X \rightarrow Y$ — оценки, проставленные ассессорами
- ▶ Правильный порядок определён только на множестве документов, найденных по одному запросу

Виды признаков

- ▶ функции только документа d (скорость загрузки страницы)
- ▶ функции только запроса q (тематика запроса)
- ▶ функции d и q (TF-IDF соответствие документа запросу)
 - ▶ текстовые
 - ▶ слова запроса q встречаются в d чаще обычного
 - ▶ слова запроса q каким-то образом выделены в d (заголовок)
- ▶ ссылочные
 - ▶ на документ d много ссылаются
 - ▶ документ d содержит много полезных ссылок
- ▶ кликовые
 - ▶ на документ d часто кликают
 - ▶ на документ d часто кликают по запросу q

Методология оценивания качества ранжирования

- ▶ Собирается выборка пользовательских запросов Q
- ▶ Набираются документы из поисковой выдачи на эти запросы, отправляются на оценку ассессорам
- ▶ Каждая пара (q, d) получает оценку релевантности y
- ▶ Исходя из этих оценок вычисляется какая-нибудь метрика

Яндекс Найти

ПОИСК КАРТИНКИ ВИДЕО КАРТЫ МАРКЕТ НОВОСТИ ПЕРЕВОДЧИК ЕЩЕ

- "Конституция Российской Федерации" (принята...)**
Статья 23 / КонсультантПлюс Статья 24 / КонсультантПлюс
Consultant.ru > document/cons_doc_LAW_28399/ +
• создала себя частью мирового сообщества
• принимаем **КОНСТИТУЦИЮ** РОССИЙСКОЙ ФЕДЕРАЦИИ.
- Конституция Российской Федерации — Википедия**
ru.wikipedia.org > Конституция Российской Федерации >
Конституция Российской Федерации — высший нормативный правовой акт Российской Федерации. Принята народом России 12 декабря 1993 года.
- Конституция Российской Федерации**
Основы конституционного строя Федеративное устройство Глава 6
constitution.ru >
Конституция Российской Федерации. Оптическая копия официального издания. Государственная власть **РФ**.
- Конституция Российской Федерации | Верховный Суд РФ**
constitution.kremlin.ru +
Разделение отдельных положений Конституции РФ 1993 года ... акты Президента РФ, ст. 90, акты, применяемые при разрешении споров: ст. 15.4, ст. 76.5.
- Конституция Российской Федерации**
constitution.garant.ru +
Конституция РФ (есть англ. вариант). Акты конституционного права. История принятия, конституции СССР и РСФСР (1918-1992). Научные работы. Конституции и Уставы субъектов **РФ**.
- Конституция Российской Федерации: Все главы и статьи**
kodeks.systems.ru > Конституция >
Кодексы и законы **РФ**. ... Конституция Российской Федерации Актуальная редакция Конституция от 21.07.2014 с изменениями, вступившими в силу с 21.07.2014.

Качество поиска и ранжирования

Пусть $Y = \{0, 1\}$, $y(q, d)$ — релевантность,
 $a(q, d)$ — искомая функция ранжирования,
 $d_q^{(i)}$ — i -й документ по убыванию $a(q, d)$

Примеры метрик качества:

- ▶ Precision — доля релевантных документов среди первых n :

$$P_n(q) = \frac{1}{n} \sum_{i=1}^n y(q, d_q^{(i)})$$

- ▶ Average Precision — средняя P_n по позициям релевантных док-в:

$$AP(q) = \sum_n y(q, d_q^{(n)}) P_n(q) / \sum_n y(q, d_q^{(n)})$$

- ▶ Mean Average Precision (MAP) — средняя AP по всем запросам

Качество поиска и ранжирования

Пусть $Y \subseteq \mathbb{R}$, $y(q, d)$ — релевантность,
 $a(q, d)$ — искомая функция ранжирования,
 $d_q^{(i)}$ — i -й документ по убыванию $a(q, d)$

Примеры метрик качества:

- ▶ Доля «дефектных пар» — число инверсий порядка среди первых n документов:

$$DP_n(q) = \frac{2}{n(n-1)} \sum_{i < j}^n [y(q, d_q^{(i)}) < y(q, d_q^{(j)})]$$

- ▶ Cumulative Gain — выигрыш релевантности по первым n документам:

$$CG_n(q) = \sum_{i=1}^n (2^{y(q, d_q^{(i)})} - 1)$$

Качество поиска и ранжирования

Примеры метрик качества:

- ▶ Discount Cumulative Gain — важно, чтобы релевантные документы находились как можно ближе к началу выдачи:

$$DCG_n(q) = \sum_{i=1}^n (2^{y(q,d_q^{(i)})} - 1) \frac{1}{\log_2(i+1)}$$

- ▶ $NDCG_n(q) = \frac{DCG_n(q)}{\max DCG_n(q)}$

- ▶ *pFound* — метрика ранжирования Яндекса

pFound

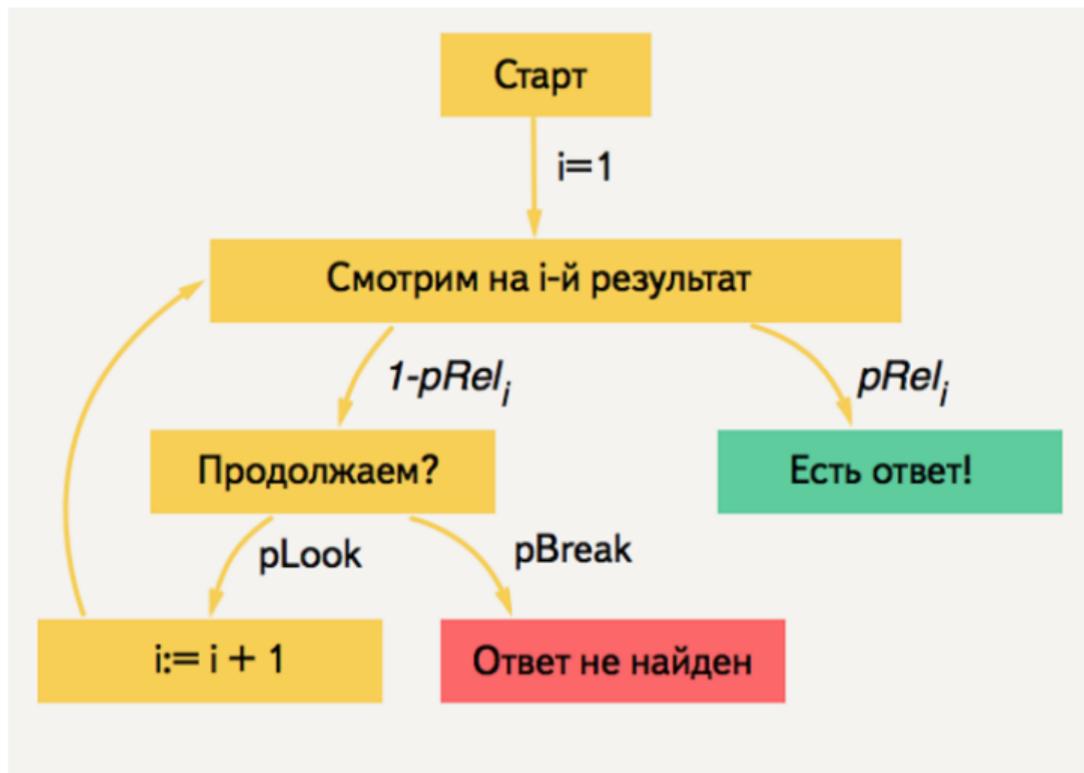
- ▶ Представляет бой вероятностную модель поведения пользователя
- ▶ Пользователь просматривает выдачу сверху вниз
- ▶ Величина $y(q, d)$ характеризует вероятность того, что пользователь с запросом q удовлетворится документом d
- ▶ При неуспехе с очередным документом выдачи пользователь разочаруется и уйдёт с вероятностью P_{out}

$$pFound_n(q) = \sum_{i=1}^n P_i y(q, d_q^{(i)}), \quad Y \subseteq [0, 1],$$

где P_i — вероятность дойти до i -го документа:

$$P_1 = 1, \quad P_{i+1} = P_i(1 - y(q, d_q^{(i)}))(1 - P_{put})$$

pFound



[Ссылка на источник картинки](#)

pFound

Параметры критерия pFound:

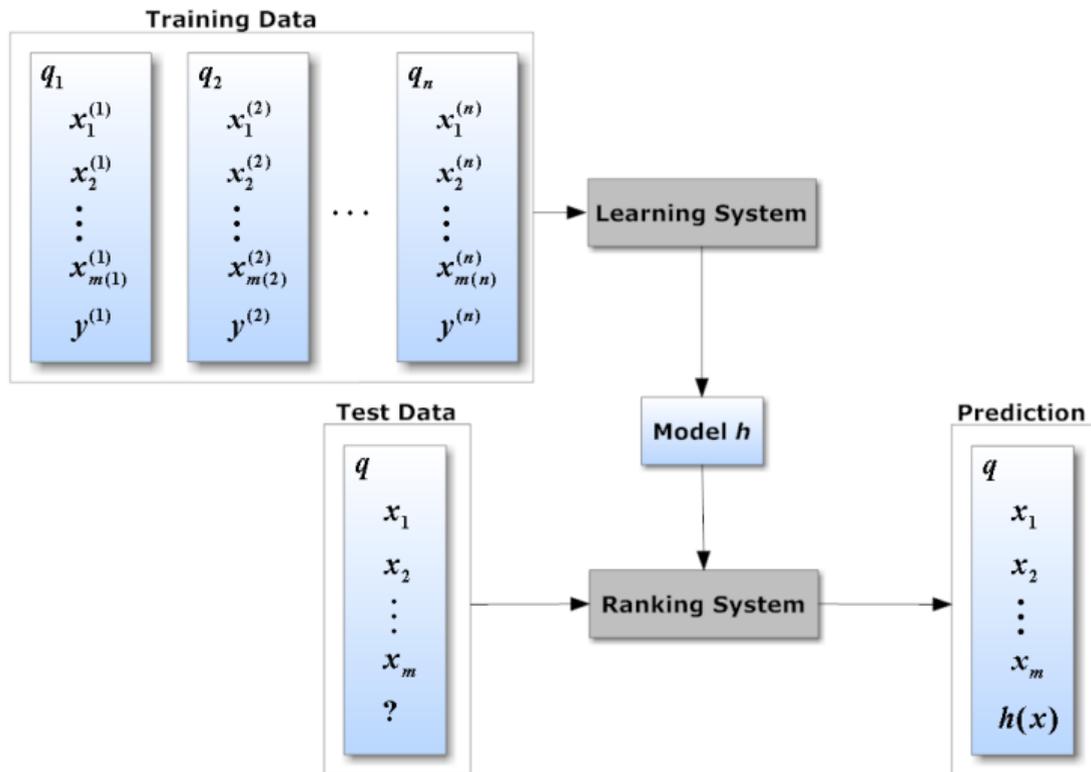
- ▶ $P_{out} = 0.15$ — вероятность прекратить поиск без ответа
- ▶ $y(q, d)$ — оценка вероятности найти ответ в документе

Оценка ассессора	$y(q, d)$
Vital	0.61
Useful	0.41
Relevant+	0.14
Relevant-	0.07
Not Relevant	0.00

Гулин А., Карпович П., Расковалов Д., Сегалович И.

Оптимизация алгоритмов ранжирования методами машинного обучения // РОМИП-2009.

Обучение ранжированию



Ссылка на источник картинки

Подходы к построению алгоритмов ранжирования

1. Point-wise — поточечный

Каждой паре (q, d) проставлена некоторая численная оценка, и задача сводится к построению регрессии (или классификации, если оценок всего несколько)

2. Pair-wise — попарный

Для двух документов, соответствующих одному запросу, решается задача бинарной классификации (какой из них релевантнее)

3. List-wise — списочный

На вход поступает список всех документов, на выходе — их перестановка. Модель напрямую оптимизирует одну из описанных выше метрик (точнее, её гладкую аппроксимацию)

Подходы к построению алгоритмов ранжирования

1. Point-wise — поточечный

Даёт менее высокое качество, чем другие подходы

2. Pair-wise — попарный

Обычно используется на практике, примеры:

RankNet, FRank, RankBoost, RankSVM, IR-SVM

3. List-wise — списочный

Существенной сложнее прочих типов, примеры:

SoftRank, SVM^{map}, AdaRank, RankGP, ListNet, ListMLE

Попарный подход к обучению ранжированию

Описанные выше метрики не являются гладкими

Перейдём к гладкому функционалу ранжирования:

$$Q(a) = \sum_{i \prec j} \underbrace{[a(x_j) - a(x_i) < 0]}_{\text{Margin}(i,j)} = \sum_{i \prec j} \mathcal{L}(a(x_j) - a(x_i)) \rightarrow \min,$$

где $a(x)$ — алгоритм ранжирования,

$i \prec j$ означает, что i менее релевантен, чем j ,

$\mathcal{L}(M)$ — убывающая непрерывная функция отступа $M(i, j)$:

- ▶ $\mathcal{L}(M) = (1 - M)_+$ — RankSVM
- ▶ $\mathcal{L}(M) = \exp(-M)$ — RankBoost
- ▶ $\mathcal{L}(M) = \log(1 + \exp(-M))$ — RankNet

Ranking SVM

Постановка задачи SVM для попарного подхода:

$$Q(a) = \frac{1}{2} \|w\|^2 + C \sum_{i < j} \underbrace{\mathcal{L}(a(x_j) - a(x_i))}_{\text{Margin}(i,j)} \rightarrow \min_a,$$

где $a(x) = \langle w, x \rangle$ — обучаемая функция ранжирования,

$\mathcal{L}(M) = (1 - M)_+$ — функция потерь,

$M = \text{Margin}(i, j) = \langle w, x_j - x_i \rangle$ — отступ,

Постановка задачи квадратичного программирования:

$$\alpha(x) = \begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i < j} \xi_{ij} \rightarrow \min_{w, \xi} \\ \langle w, x_j - x_i \rangle \geq (1 - \xi_{ij}), & i < j \\ \xi_{ij} \geq 0, & i < j \end{cases}$$

От RankNet до LambdaRank

RankNet: гладкий функционал качества ранжирования:

$$Q(a) = \sum_{i \prec j} \mathcal{L}(a(x_j) - a(x_i)) \rightarrow \min$$

при $\mathcal{L}(M) = \log(1 + \exp(-\sigma M))$ и линейной модели $a(x) = \langle w, x \rangle$

Метод стохастического градиента:

На каждой итерации выбираем $q, i \prec j$ случайно:

$$w^{new} = w^{old} + \eta \frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)} (x_j - x_i)$$

Christopher J.C. Burges From RankNet to LambdaRank to LambdaMART:
An Overview // Microsoft Research Technical Report MSR-TR-2010-82.
2010.

От RankNet до LambdaRank

Метод стохастического градиента:

$$w^{new} = w^{old} + \eta \underbrace{\frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)}}_{\lambda_{ij}} (x_j - x_i)$$

Оказывается, для оптимизации негладких функционалов MAP, NDCG, rFound достаточно домножить λ_{ij} на изменение данного функционала при перестановке местами $x_i \leftrightarrow x_j$

LambdaRank: домножение на изменение NDCG при перестановке $x_i \leftrightarrow x_j$ приводит к оптимизации NDCG:

$$w^{new} = w^{old} + \eta \underbrace{\frac{\sigma}{1 + \exp(\sigma \langle x_j - x_i, w \rangle)}}_{\lambda_{ij}} |\Delta NDCG_{ij}| (x_j - x_i)$$

Дополнительные факты о ранжировании

- ▶ LambdaMart — алгоритм, комбинирующий LambdaRank и градиентный бустинг над решающими деревьями, работает лучше RankNet и LambdaRank
- ▶ В Яндексе работает технология MatrixNet — градиентный бустинг над небрежными решающими деревьями (ODT)
- ▶ За несколько лет придумано и проверено порядка тысячи признаков для ранжирования
- ▶ Ежемесячно в выборку добавляется более 50 000 оценок ассессоров

Оптимизация кликовых метрик

Релевантность — не единственный критерий качества выдачи поисковой системы

Много информации для обучения можно получить, анализируя поведение пользователей

Оптимизация кликовых метрик:

- ▶ Скачиваются результаты нескольких поисковых систем
- ▶ Из них формируется поисковая выдача, анализируются клики пользователей по её документам
- ▶ Обучается модель, предсказывающая документы, которые привлекут наиболее число кликов

Разнообразиие поисковой выдачи

Проблема №1: неоднозначные запросы

Пример: «ягуар»

- ▶ животное?
- ▶ марка автомобиля?
- ▶ танк? (немецкий или китайский?)
- ▶ напиток?

С точки зрения обычных метрик, весь топ выдачи нужно замостить одинаковыми релевантными документами

Разнообразие позволяет собрать разнородную выдачу, чтобы удовлетворить в среднем всех

Wide pFound

- ▶ Предполагается, что пользователь, делая запрос, мог иметь ввиду один из *интентов* $I = \{I_1, \dots, I_m\}$
- ▶ Примеры интентов: автомобили, картинки, новости, животные, ...
- ▶ Каждый интент имеет некоторую вероятность $p(I_i)$ и порождает собственное распределение релевантностей на документах
- ▶ Тогда для каждого интента можно вычислить его pFound и взять взвешенную сумму:

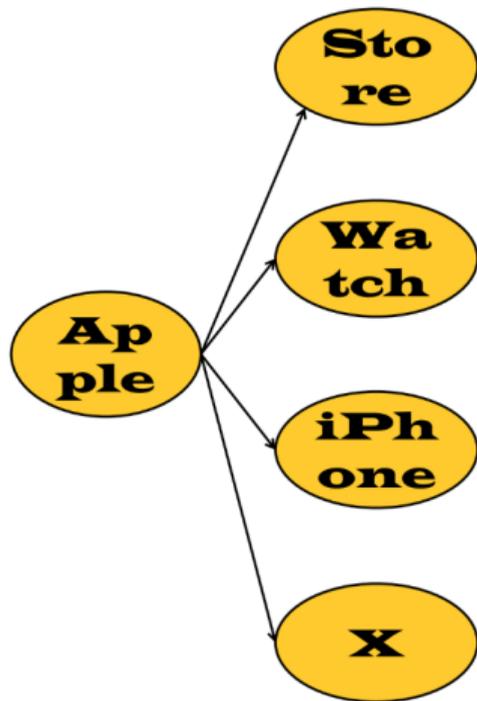
$$\text{wide pFound} = \sum_{i=1}^m p(I_i) \text{pFound}(I_i)$$

- ▶ Как вычислять вероятности интентов?

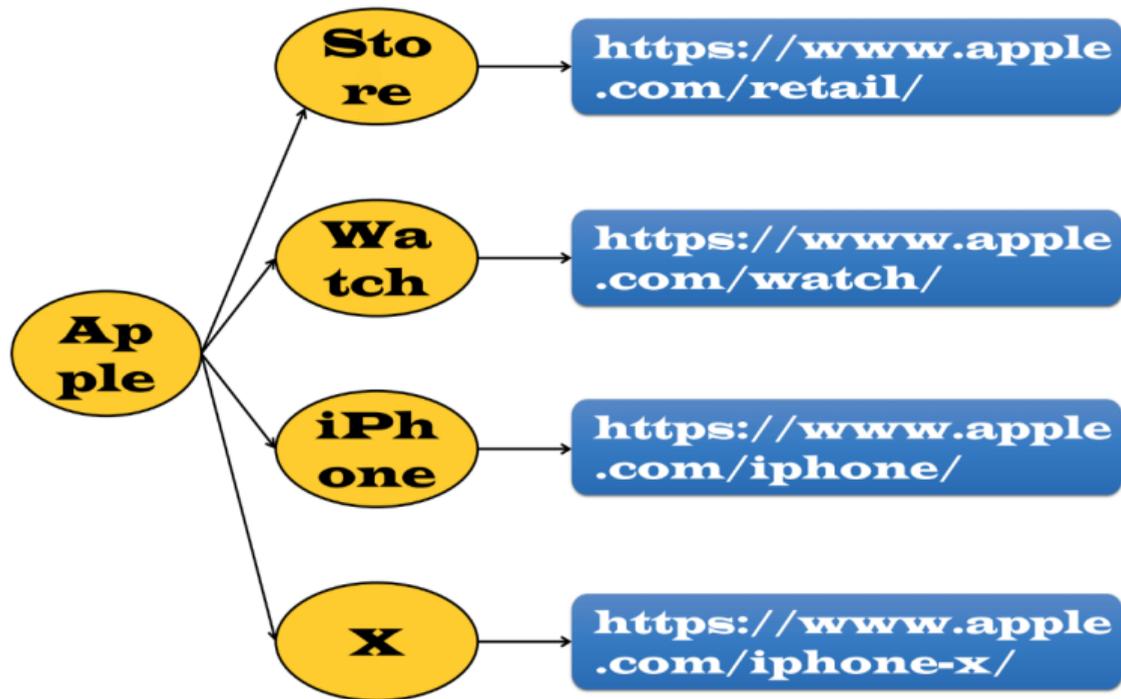
Технология «Спектр»

- ▶ Интент пользователя определяется по продолжениям ведённого запроса
- ▶ Продолжения классифицируются по различным тематикам
- ▶ Тематики являются интенентами
- ▶ Вероятности определяются по частоте соответствующих продолжений запросов

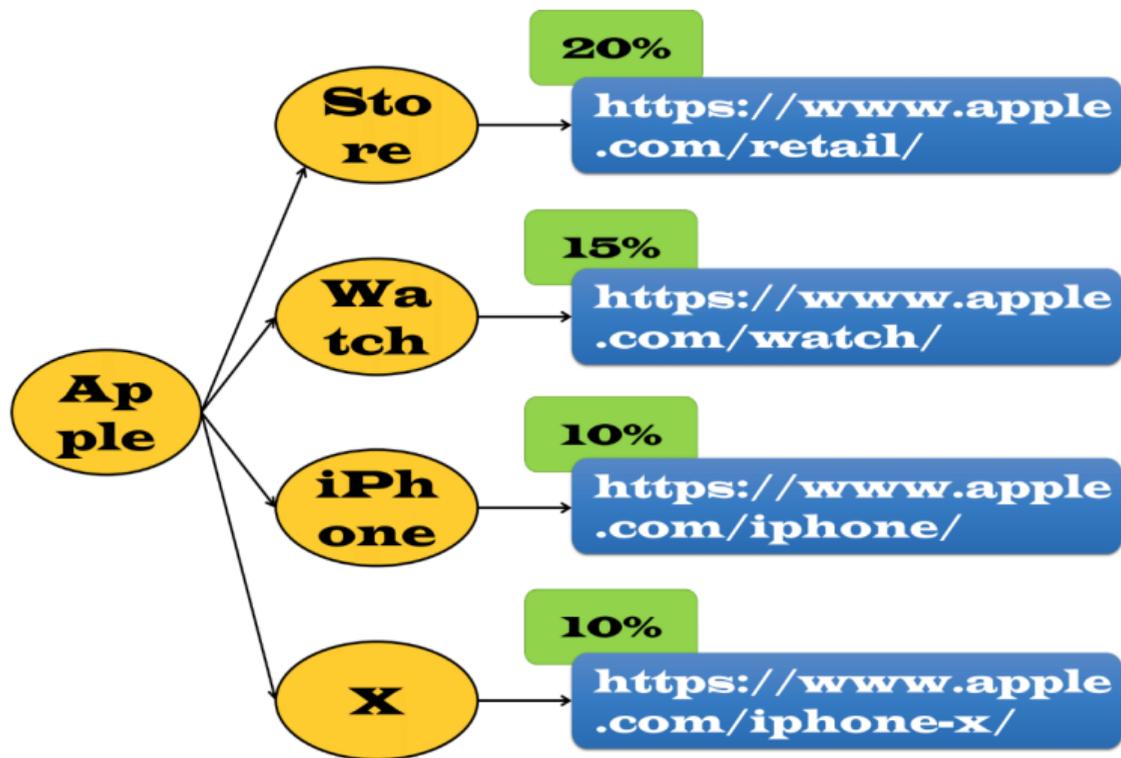
Оценка вероятностей интенгов



Оценка вероятностей интенгов



Оценка вероятностей интентов



Разнообразие поисковой выдачи

Проблема №2: различные типы информации

— как выбирать и комбинировать?

Кошка — Википедия

<https://ru.wikipedia.org/wiki/Кошка> ▼

Ко́шка, или домашняя ко́шка (лат. *Félis silvestris catus*), — домашнее животное, одно из наиболее популярных (наряду с собакой) «животных-компаньонов». С зоологической точки зрения домашняя кошка — млекопитающее семейства кошачьих отряда хищных. Ранее домашнюю кошку нередко ...

Тайская кошка · Персидская кошка · Трёхцветная кошка · Русская голубая кошка

Кошка (значения) — Википедия

[https://ru.wikipedia.org/wiki/Кошка_\(значения\)](https://ru.wikipedia.org/wiki/Кошка_(значения)) ▼

Ко́шка: Кошка, или домашняя кошка (лат. *Felis silvestris catus*) — хищное млекопитающее рода кошек, популярное домашнее животное. Кошка — общее название некоторых видов хищных млекопитающих рода Кошки (лат. *Felis*) и некоторых других родов семейства кошачьих. Кошка — упрямднённое ...

Кошка — Lurkmore

lurkmore.to/Кошка ▼

5 февр. 2018 г. - Кошка (женоненавистн. кот, интернет. кота) — млекопитающее семейства кошачьих с четырёх лапах, с одним хвостом и множеством усов. Имеет шерсть и хвост, окрас которых может быть разным. Извращённые селекционерские идеи людийшек породили и разновидности без шерсти ...

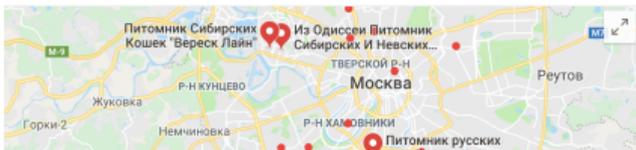
Кошка, перекрашенная хозяйкой в розовый цвет, умерла - YouTube



https://www.youtube.com/watch?v=WmG2e_LmBoA ▼

3 мар. 2015 г. - Добавлено пользователем TomoNews Russia

Вот эта женщина с невозможной причёской и розовым котёнком – русская писательница Лена Ленина. Это она решила покрасить свою кошку в розовый цвет и, как го...



Кошка

Животное

Ко́шка, или домашняя ко́шка, — домашнее животное, одно из наиболее популярных «животных-компаньонов». С зоологической точки зрения домашняя кошка — млекопитающее семейства кошачьих отряда хищных. [Википедия](#)

Латинское название: *Felis catus*

Продолжительность жизни: 4 – 5 лет (в дикой природе)

Период беременности: 64 – 67 дней

Масса: 3,6 – 4,5 кг (взрослая особь)

Ежедневный сон: 12 – 16 часов

Рост: 23 – 25 см

Породы

Ещё 20+



Британская короткош...
кошка



Сямская
кошка



Персидская
кошка



Рэгдолл



Мейн-кун

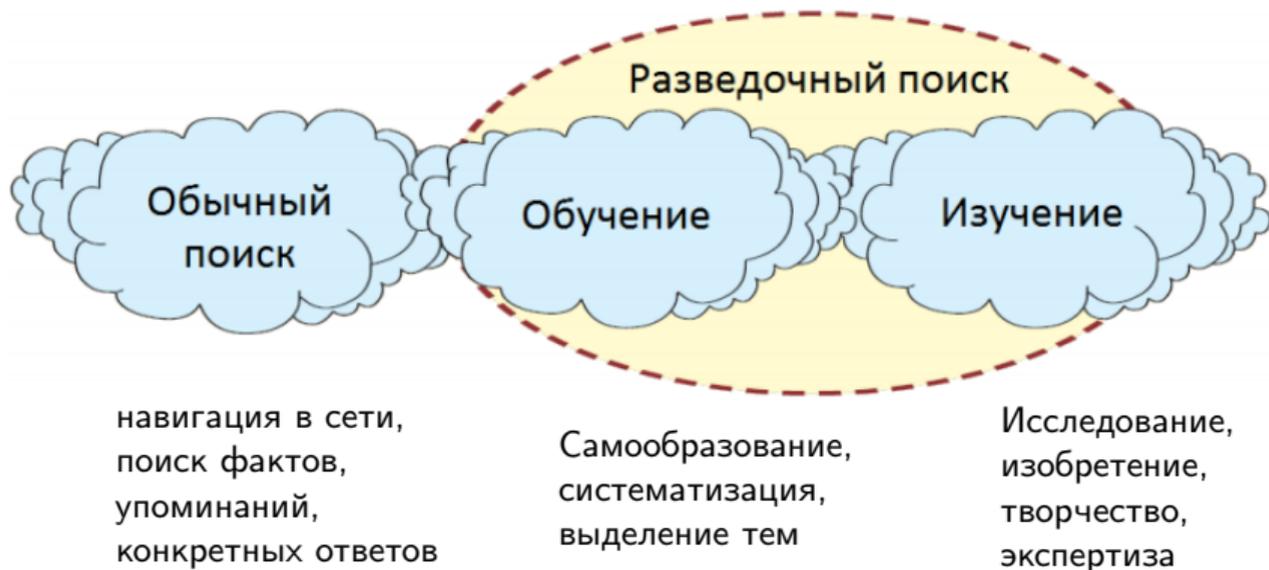
Оставить отзыв

Размещение контента

- ▶ Поисковая выдача — это не только сниппеты сайтов, это картинки, видео, геолокации и т.п.
- ▶ Для каждого объекта нужно определять оптимальную позицию на экране по данному запросу
- ▶ Обучение можно производить по кликам, выборка при этом собирается в продакшене путём варьирования положения объекта на странице
- ▶ Задача формулируется как задача «многоруких контекстных бандитов»
- ▶ [Ссылка](#) на статью про поисковую свежесть

Концепция разведочного поиска

- ▶ Пользователь может не знать ключевых терминов
- ▶ Пользователя может интересовать множество ответов



Gary Marhionini., Exploratory Search: from finding to understanding. 2006.

Возможный сценарий разведочного поиска

Поисковый запрос:

- ▶ документ любой длины или коллекция документов

Поисковая потребность:

- ▶ к каким темам относится мой запрос?
- ▶ что ещё известно по этим темам?
- ▶ какова тематическая структура этой предметной области?
- ▶ какие области являются смежными?
- ▶ что ещё есть понятного, обзорного, важного, свежего?

Сценарий поиска: имея любой текст под рукой, в любом приложении, получаем картин содержащихся в нём тем-подтем и «дорожную карту» предметной области в целом.

Технологические элементы разведочного поиска

По всем элементам имеются готовые решения:

1. Интернет-краулинг
2. Фильтрация контента
3. Движок (тематическое моделирование)
4. Инвертированный индекс
5. Ранжирование
6. Визуализация
7. Персонализация

Успехов!