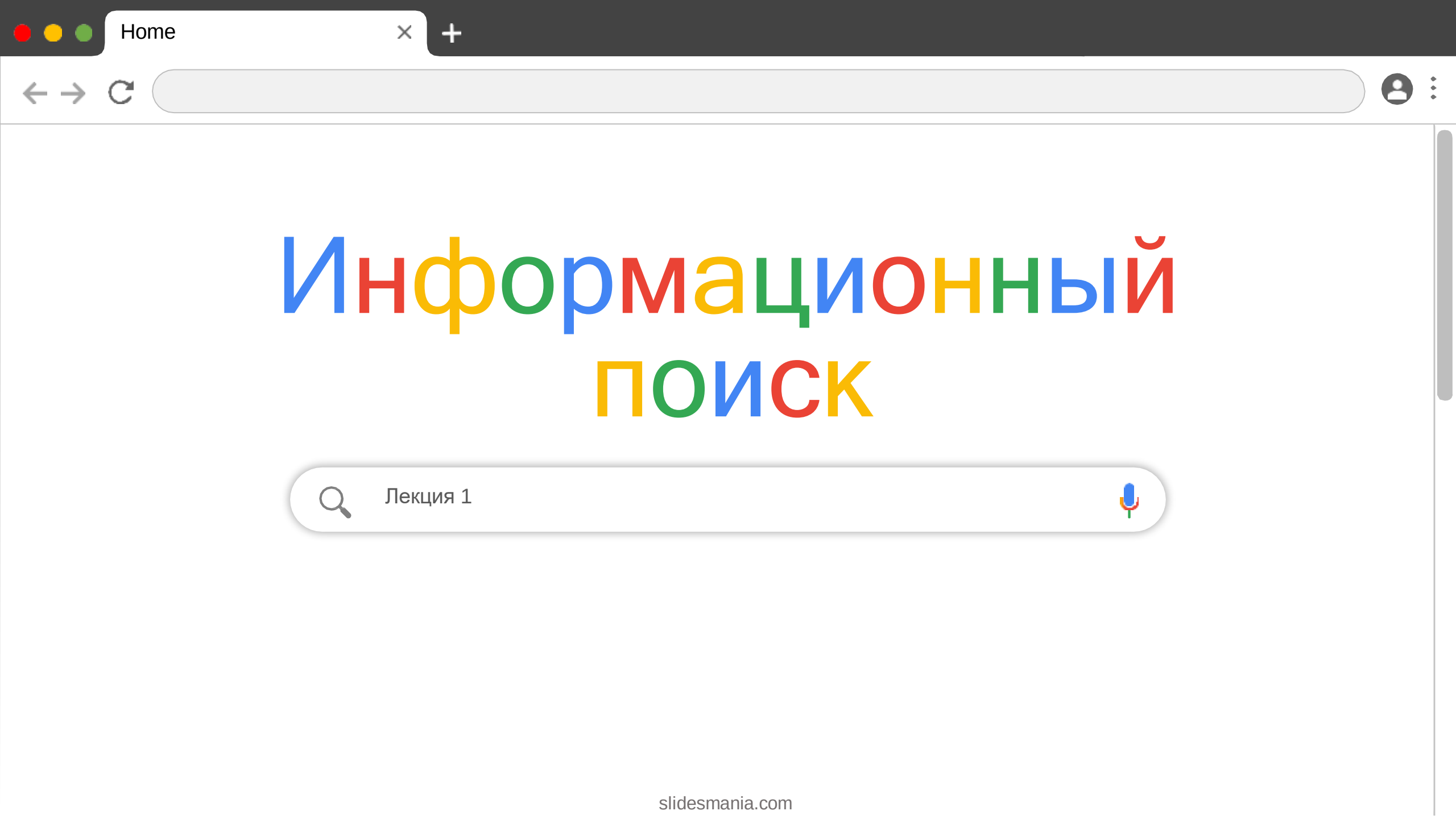


Формула оценки

О = 0.6 * дз + 0.4 * проект

Темы ближайших занятий

1. Разные виды задач: IR, KR, рексистемы
2. IR: что это и как работает, индекс и его виды
3. Разные способы векторизации (возможно, здесь будет две пары)
4. Докер. Теория
5. Докер. Практика

Home | Recent Tabs | Current Page

You can type something here…

Чат в ТГ

Home | Recent Tabs | Current Page

You can type something here…

Репозиторий (там пока старая версия)

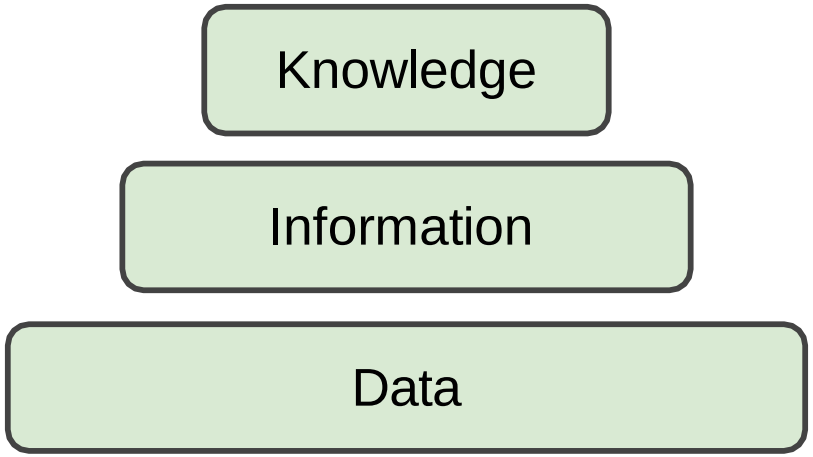
Данные, информация, знания

Есть три сущности: данные, информация и знание.

Данные — сырой набор объектов (картинок, текстов, аудио), содержащих информацию.

Информация — структурированный набор фактов (таблица, схема, осмысленная часть картинки или аудио).

Знание — абстрактное представление этой информации в голове человека.



Data

Пример:

- Логи работы сервиса
- Новости с сайта
- Все документы компании
- Показатели со счетчиков

Признаки:

- Объективность
- Нет конкретной цели
- Необработанность
- Можно оценить количественно

Правда ли совсем нет структуры?

Где их можно хранить?

Какие еще примеры вы можете привести?

Information

Пример:

- График частоты отказов системы
- Частоты тем новостей
- Все части документов по ИБ
- Сумма к оплате за месяц

Признаки:

- Объективность (желательно)
- Есть конкретная цель
- Результат обработки
- Можно оценить количественно

Почему мы хотим работать именно с информацией?

Важна ли связь информации с источником в данных?

Всегда ли эта связь один к одному?

Knowledge

Пример:

- Устранение уязвимости сервиса
- Реклама на основе тематики сайта
- Конфликты с новым правилом про ИБ
- Решение о покупке новой техники

Признаки:

- Субъективность
- Есть конкретная цель
- Результат обработки
- Нельзя оценить количественно

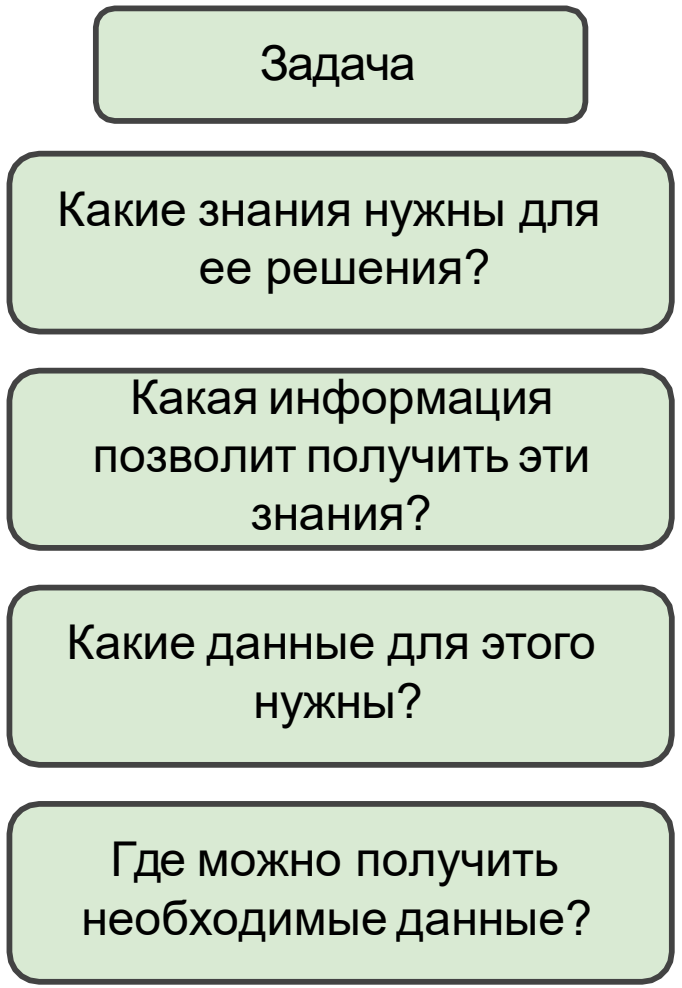
Можно ли получать знания автоматически? Например, как?

Могут ли знания быть объективными?

Почему субъективность усложняет автоматизацию?

Решение задачи

- Какой вид транспорта выбрать для путешествия?
- Как с точки зрения типологии выглядит порядок слов?
- Автоматически определить, нужно ли заблокировать пользователя форума



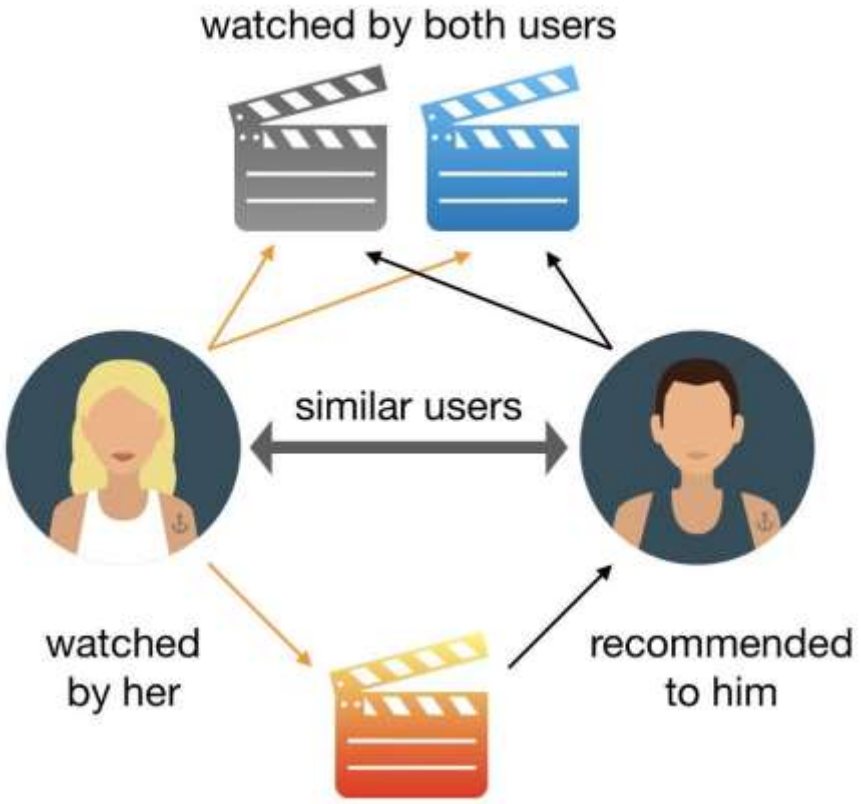
Зачем это надо?

Рекомендательные системы

Задача: рекомендовать пользователю сервиса фильмы/книги/музыку, которая ему понравится

Формально:

- есть две группы объектов, какие-то из них уже связаны, нужно предположить, где еще должны быть связи
- есть объект и набор возможных рекомендаций, нужно выбрать лучшую

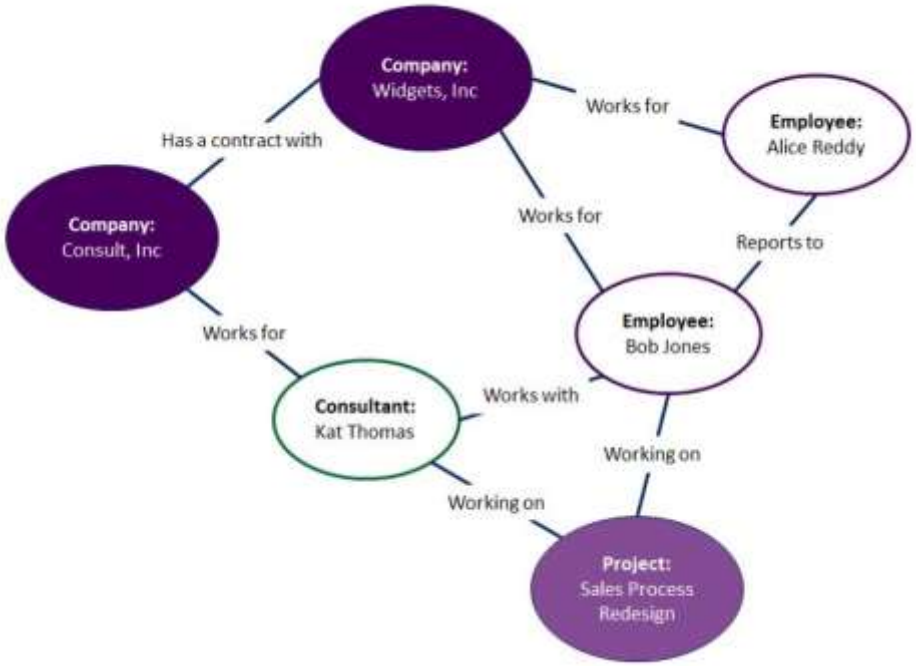


Источник графики: https://vitalflux.com/recommender-systems-in-machine-learning-examples/

Knowledge Retrieval

Задача: автоматически сформировать на основе информации/данных какую-то систему знаний (часто граф)

Формально: выделить объекты и отношения между ними



Какие здесь могут быть сложности?

Где может требоваться такая система?

Источник графики: https://enterprise-knowledge.com/what-is-an-ontology/

Information Retrieval

В очень широком смысле — поиск объектов в массиве по условию.
Теоретически, многие задачи NLP можно сформулировать в подобном виде.

В чуть менее широком — выделение информации из неструктурированных данных.

В контексте NLP — извлечение информации из текста.

Типы поиска

По типу данных

- Текстовый
- По картинкам
- По аудио
- По метаинформации

По условию поиска

- Булев поиск
- Поиск по сходству
- Поиск по релевантности

Home | Recent Tabs | Current Page

You can type something here…

Поиск по тексту

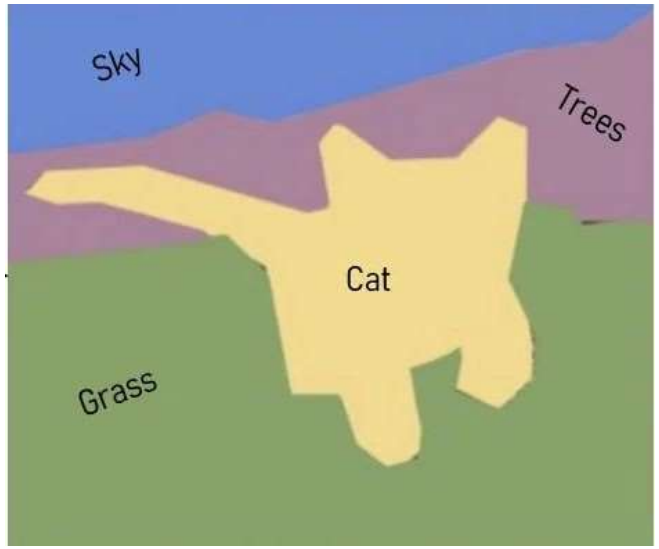
Чаще всего полнотекстовый поиск — поиск по содержанию документа.

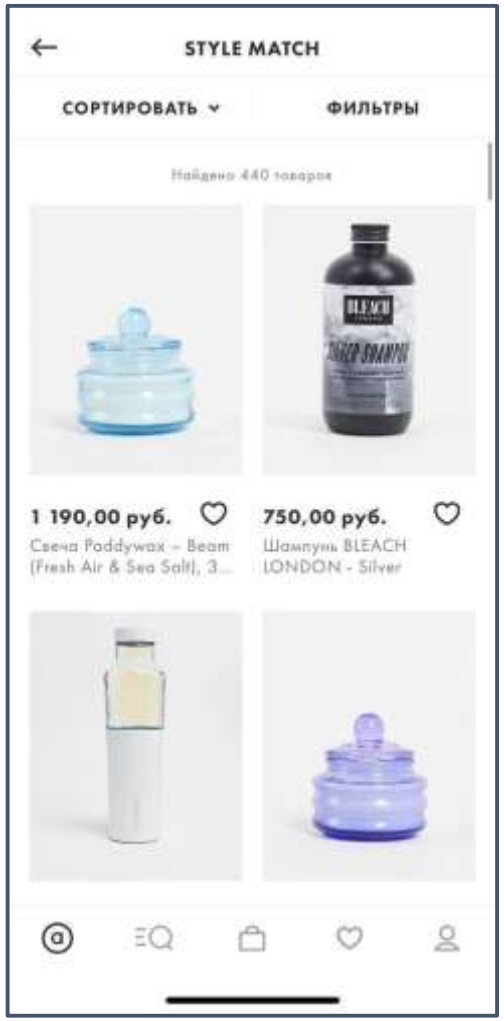
Home | Recent Tabs | Current Page

You can type something here…

Поиск по картинке

В том числе задачи классификации и сегментации изображений



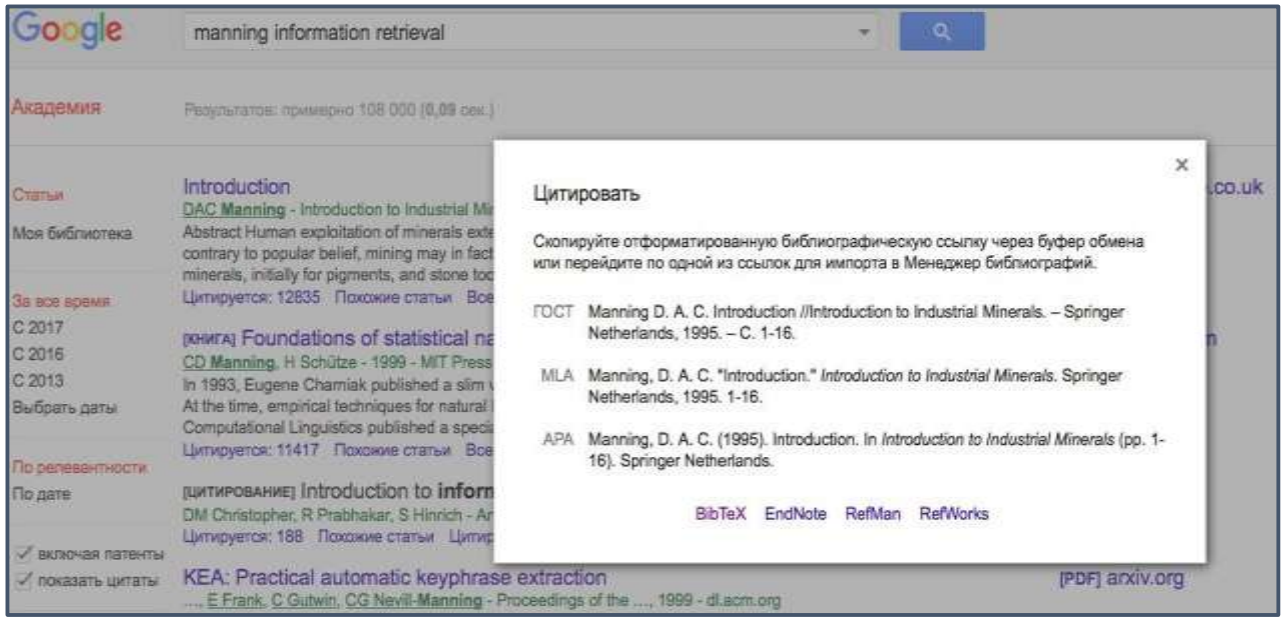


Home | Recent Tabs | Current Page

You can type something here…

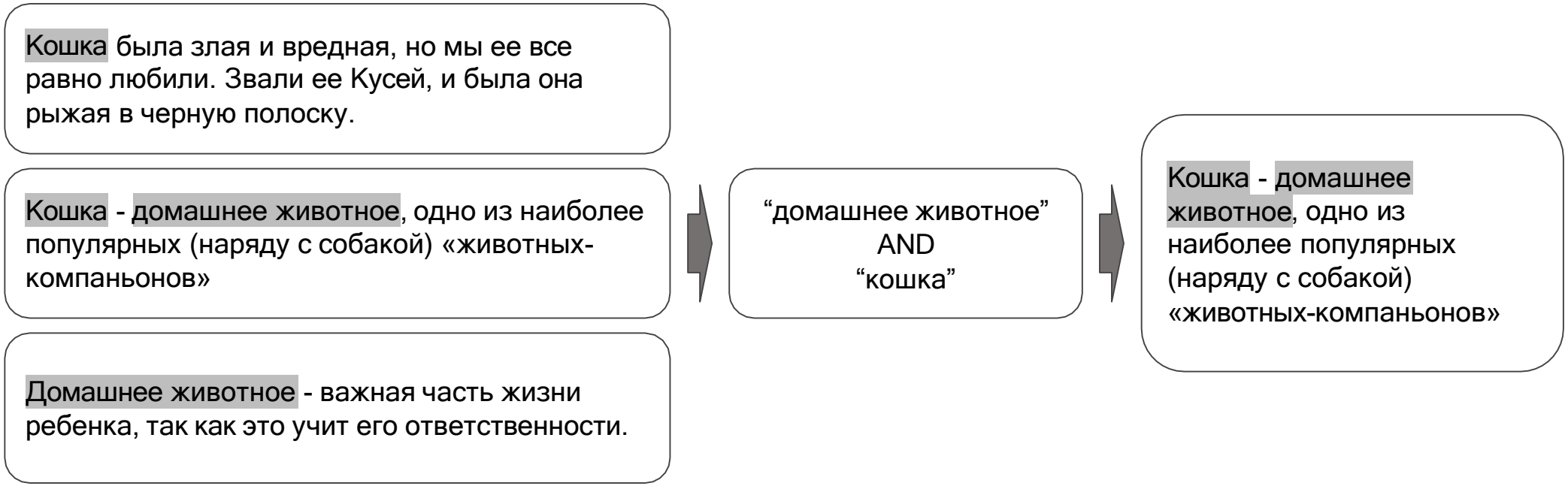
Поиск по метаинформации

Фактически — поиск информации по информации, представленной в другом формате.



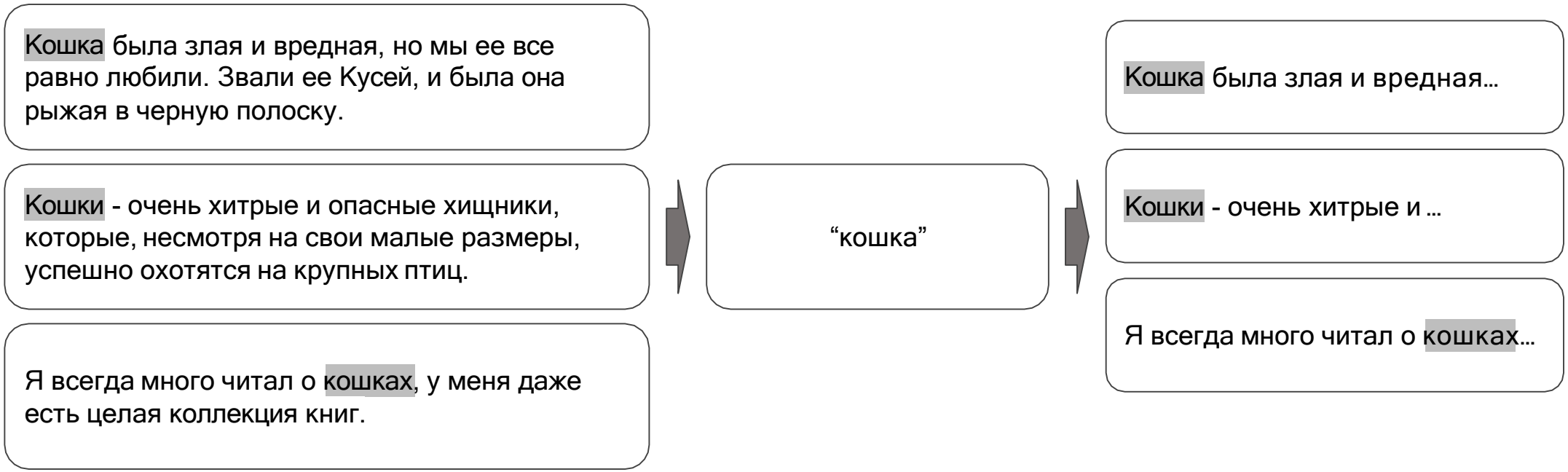
Булев поиск

Используются операторы алгебры логики (or, and, xor). Есть только два варианта: объект либо подходит под запрос, либо нет.



Поиск по сходству

Надстройка над булевым поиском: допускаются отклонения от запроса. Чаще всего оно ограничивается небольшим расстоянием Левенштейна.



Поиск по релевантности

Каждому документу присваивается число — мера его релевантности запросу

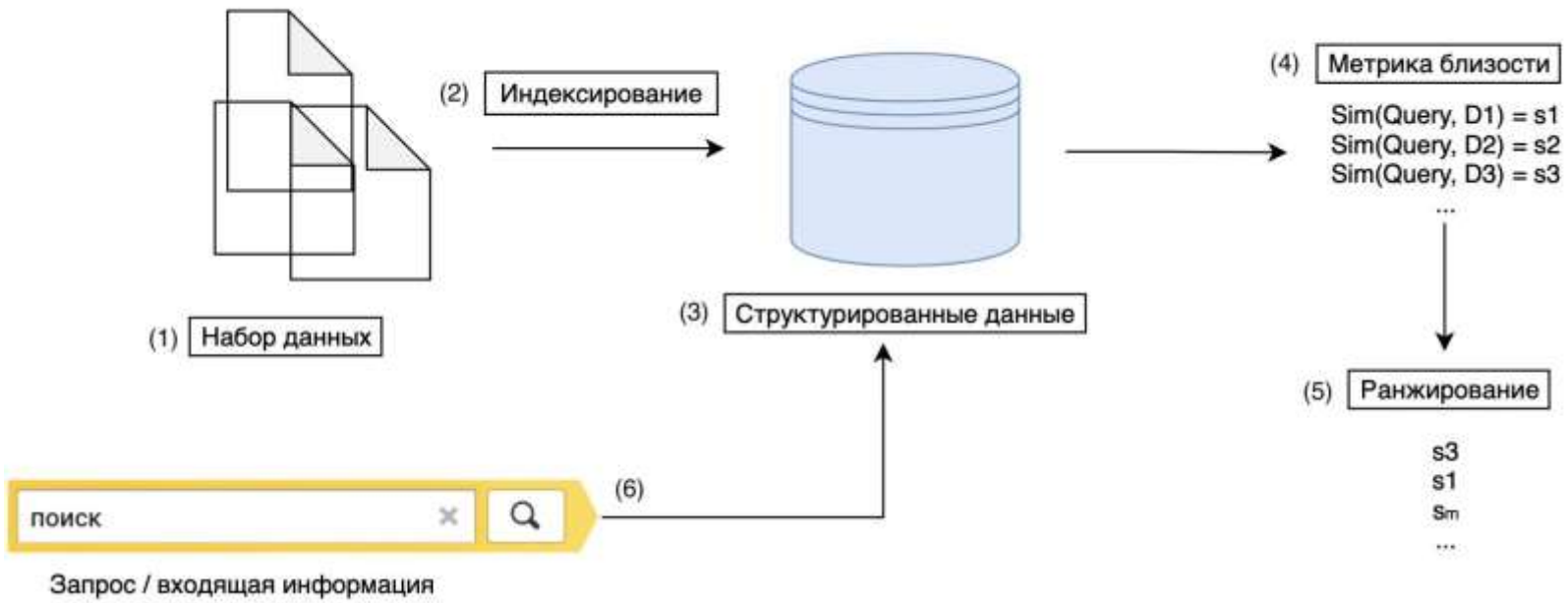
Примеры задач

А точнее проблемы, где необходимо использование методов инфопоиска

Ты шеф в большой компании. У тебя много разных отделов. Твоему стажеру надо узнать, как работает нечто, разработанное в другом отделе. Он не знает кому писать или боится спрашивать, но в итоге как-то находит источник информации.

Ты РЖД. У тебя есть колл-центр. Его задача отвечать на вопросы клиентов РЖД. Ты знаешь, что 60% вопросов повторяются из раза в раз. Использовать для этого человеческие ресурсы - дорого и малоэффективно.

Последовательность действий



Общая постановка задачи

Дано

Набор объектов = база данных (1)

- Набор корпоративных документов
- Набор типичных вопросов и ответов на них
- Набор продаваемых товаров

Задача

Пришел новый объект - запрос (6)

- Описание сервиса, к которому ищем документацию
- Новый вопрос от юзера
- Фото штанов, которые нужно найти среди товаров

Надо найти самый подходящий к нему объект из базы

Шаг 1. Индексируем данные (2)

Что значит “индексируем”?

Ищем, обрабатываем и сохраняем данные таким образом, чтобы потом по ним было удобно искать.

Индекси́рование, совершаемое поисковой машиной, — процесс сбора, сортировки и хранения данных с целью обеспечить быстрый и точный поиск информации (то же самое на языке Википедии)

Шаг 2. Сохраняем индекс (3)

Что такое индекс?

В результате индексирования получаются структурированные данные или индекс.

Именно к этим данным — индексу, мы обращаемся во время поиска. Исходные данные, из которых он был получен, можно не использовать.

Почему? Смотри определение индекса.

Шаг 3. Выбираем метрику близости (4)

Что такое метрика?

Любая метрика (функция) близости, подходящая для измерения схожести тех объектов, с которыми мы работаем.

Это может быть:

- Сумма и среднее отклонений
- Или квадратов отклонений
- Косинусная близость
- И т.д.

Шаг 4. Ранжируем результаты (5)

Что значит “ранжируем”?

Сортируем в соответствии со значением метрики. На первом месте должен оказаться самый релевантный объект к запросу.

Ранжи́рование — сортировка сайтов в поисковой выдаче, применяемая в поисковых системах (и снова мнение википедии).

Прямой индекс

Есть корпус, состоящий из нескольких текстов:

doc_1 = Буря мглою небо кроет

doc_2 = Вихри снежные крутя

doc_3 = То, как зверь, она

завоет doc_4 = То заплачет, как

дитя

Прямой индекс ставит каждому документу в соответствие слова, содержащиеся в нем. Например, в виде списка

Документ	Список слов
doc_1	буря, кроет, мглою, небо
doc_2	вихри, крутя, снежные
doc_3	завоет, зверь, как, она, то
doc_4	дитя, заплачет, как, то

Обратный индекс

В обратном индексе каждому слову ставится в соответствие набор документов, где оно встречается. Может быть представлен (как и прямой индекс):

В виде словаря:

```
{
        "буря": [
                "doc_1"
        ],
        "то": [
                "doc_3",
                "doc_4"
        ],
        "как": [
                "doc_3",
                "doc_4"
        ],
  ...
}
```

В виде Document-Term матрицы:

	буря	мглою	небо	кроет	вихри	снежные	крутя	...
doc_1	1	1	1	1	0	0	0	
doc_2	0	0	0	0	1	1	1	
doc_3	0	0	0	0	0	0	0	
doc_4	0	0	0	0	0	0	0	

И другие индексы...

При выборе индекса важно учитывать следующие характеристики:

- Быстродействие: как долго выполняется поиск (почему не важно время построения индекса?)
- Отказоустойчивость: насколько легко случайно/специально сломать систему, как она себя ведет в случае поломки
- Объем памяти: чаще всего у нас есть ограничения по количеству памяти на сервере
- Простота поддержки: насколько легко удалять/добавлять элементы
- Универсальность: можно ли использовать индекс повторно в других задачах

ML постановка задачи: Document Ranking

У нас есть данные:

- Список документов
- Список запросов
- Тройки вида (q, d1, d2), где q — это запрос, d1 — более релевантный запросу документ, d2 - менее релевантный.

Задача:

Для каждого запроса q упорядочить документы d1..dn так, чтобы это как можно точнее соответствовало соотношениям в тройках. То есть нужна модель, которая для менее релевантного выдаст меньшую оценку.

Другая постановка задачи

С парами документов неудобно работать: непонятно, как формализовать такой формат данных для функции потерь модели.

Будем работать по-другому:

- Объекты: пары из запроса и документа
- Ответы: такие числа, что большему числу соответствует большая релевантность (и должна соответствовать большая, но не обязательно такая же оценка от модели)

Фактически, вместо того, чтобы руками определить меру релевантности, мы учим для этого модель.

Варианты задачи

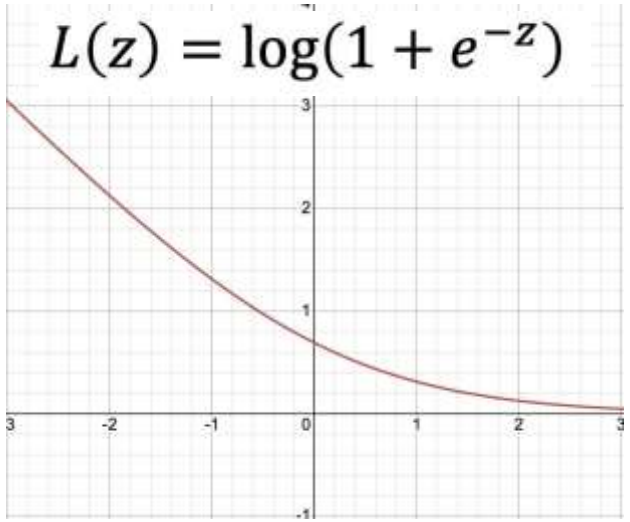
1. Pointwise (поточечная):
   - ➢ Предсказываем конкретное число.
   - ➢ Можно использовать любую модель для задачи регрессии
   - ➢ Проблема: никак не учитывается порядок и относительность оценок
2. Pairwise (попарная):
   - ➢ Предсказываем конкретное число, но учимся на парах: штрафуем за неправильную разницу в паре предсказаний
   - ➢ Сложнее постановка задачи, сложнее оптимизировать
   - ➢ Обычно дает более высокое качество

Pairwise Loss

Источник формул: https://github.com/hse-ds/iad-intro-ds/blob/master/2023/lectures/lecture18-ranking.pdf

Сначала запишем в том виде, в котором сформулировали:

$$\sum_{(q,d_i,d_j)\in R} [a(q,d_i) - a(q,d_j) < 0]$$



Неприятно, что полученная штука дискретна и производную посчитать не получится. Сделаем так, чтобы было можно:

$$\sum_{(q,d_i,d_j)\in R} [a(q,x_i) - a(q,x_j) < 0] \leq \sum_{(q,d_i,d_j)\in R} L\left(a(q,x_i) - a(q,x_j)\right)$$

$$L(z) = \log(1 + e^{-z})$$

Метрики: Precision

k — для скольки первых документов мы считаем оценку

Precision-top-k: доля документов, где хотя бы одно из k предсказаний оказалось релевантным

Precision@k: средняя доля документов, которые оказались релевантны запросу

Average Precision@k: среднее значение Precision@i для i = 1..k (считается для одного запроса)

Mean Average Precision@k: среднее Average Precision@k по всем запросам

Вот тут можно почитать с картинками:
https://www.evidentlyai.com/ranking-metrics/mean-average-precision-map

DCG (Discounted cumulative gain)

Формула взята вот отсюда и здесь же можно почитать подробнее:
https://www.evidentlyai.com/ranking-metrics/ndcg-metric

$$\text{DCG@K} = \sum_{k=1}^{K} \frac{rel_i}{\log_2(i+1)}$$

k - для скольки первых документов мы считаем оценку

i - позиция объекта в предсказанном рейтинге

rel_i - оценка релевантности от модели

Усредняются для получения общей оценки