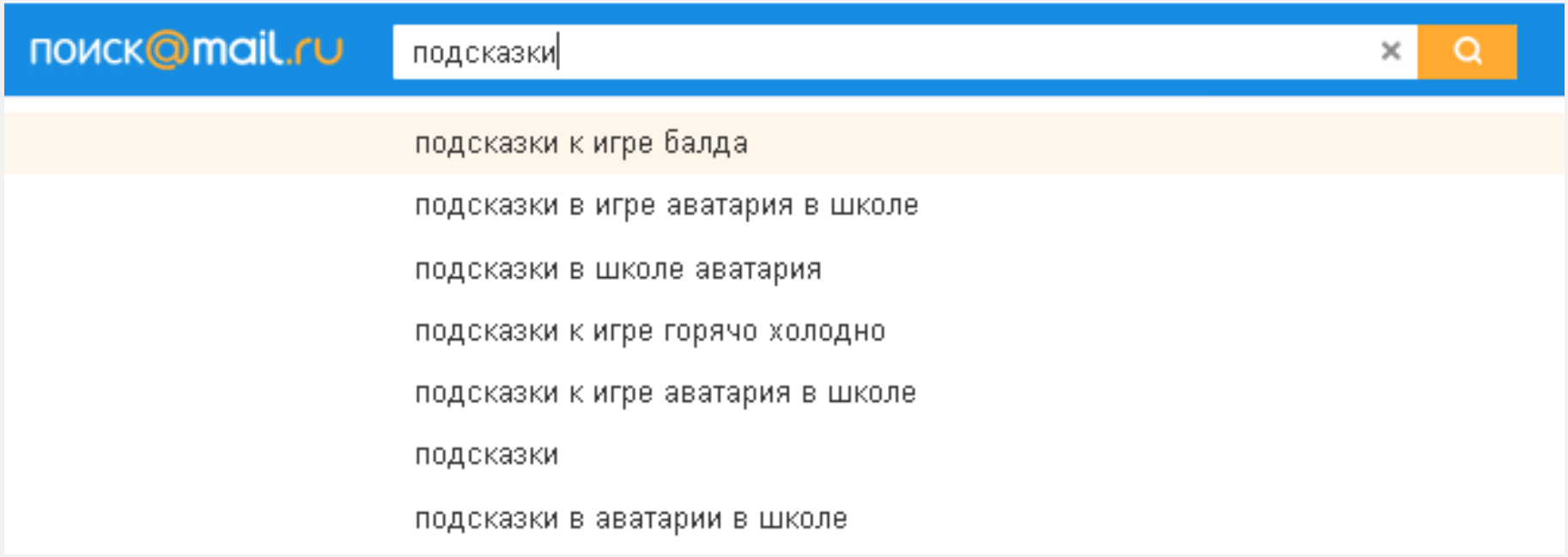
ТЕХНОСФЕРА

Лекция №14

Саджесты
Переформулировки
Классификаторы

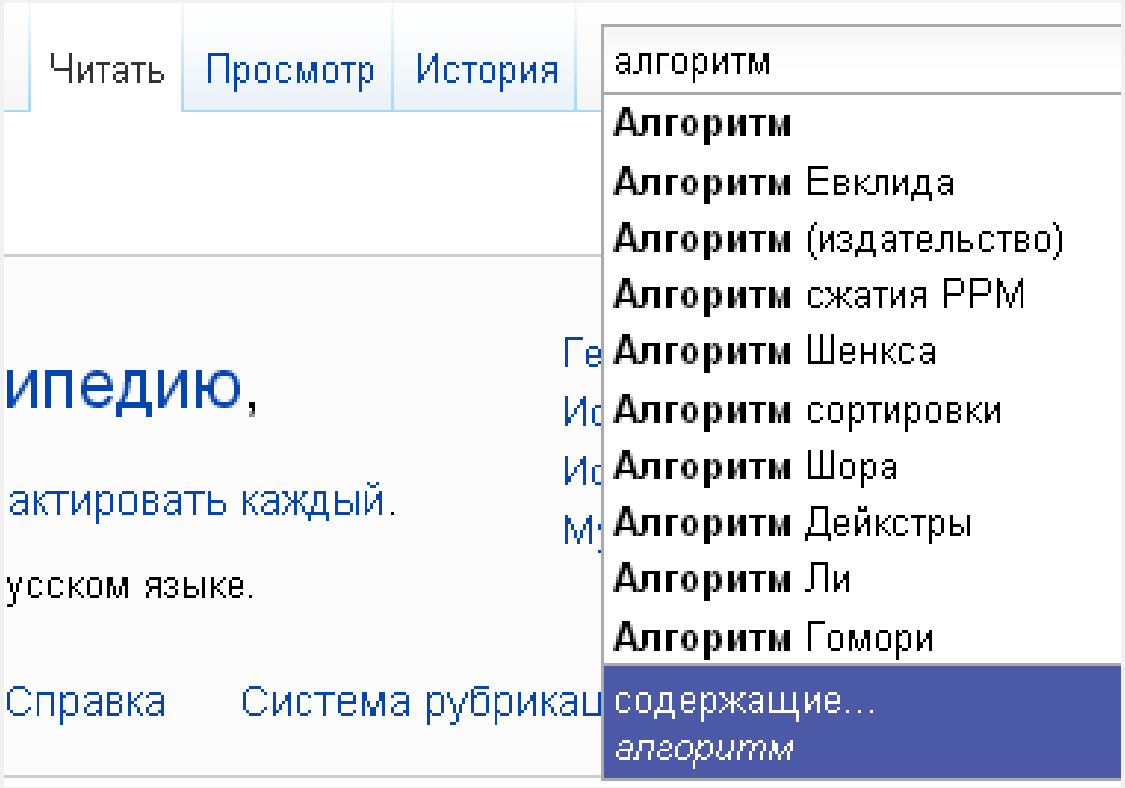
Евгений Чернов

Поисковые подсказки



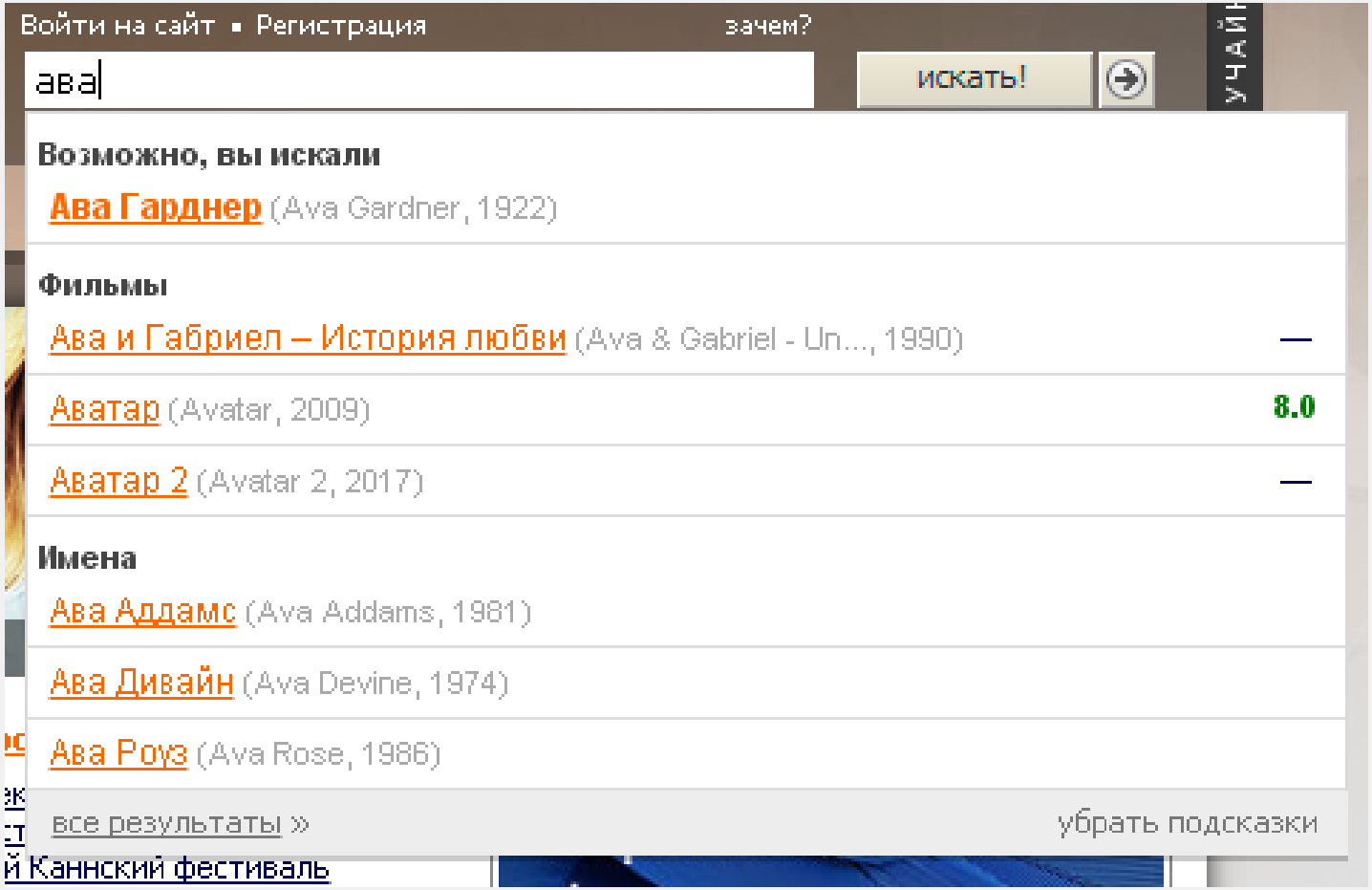
Примеры подсказок

wikipedia.org



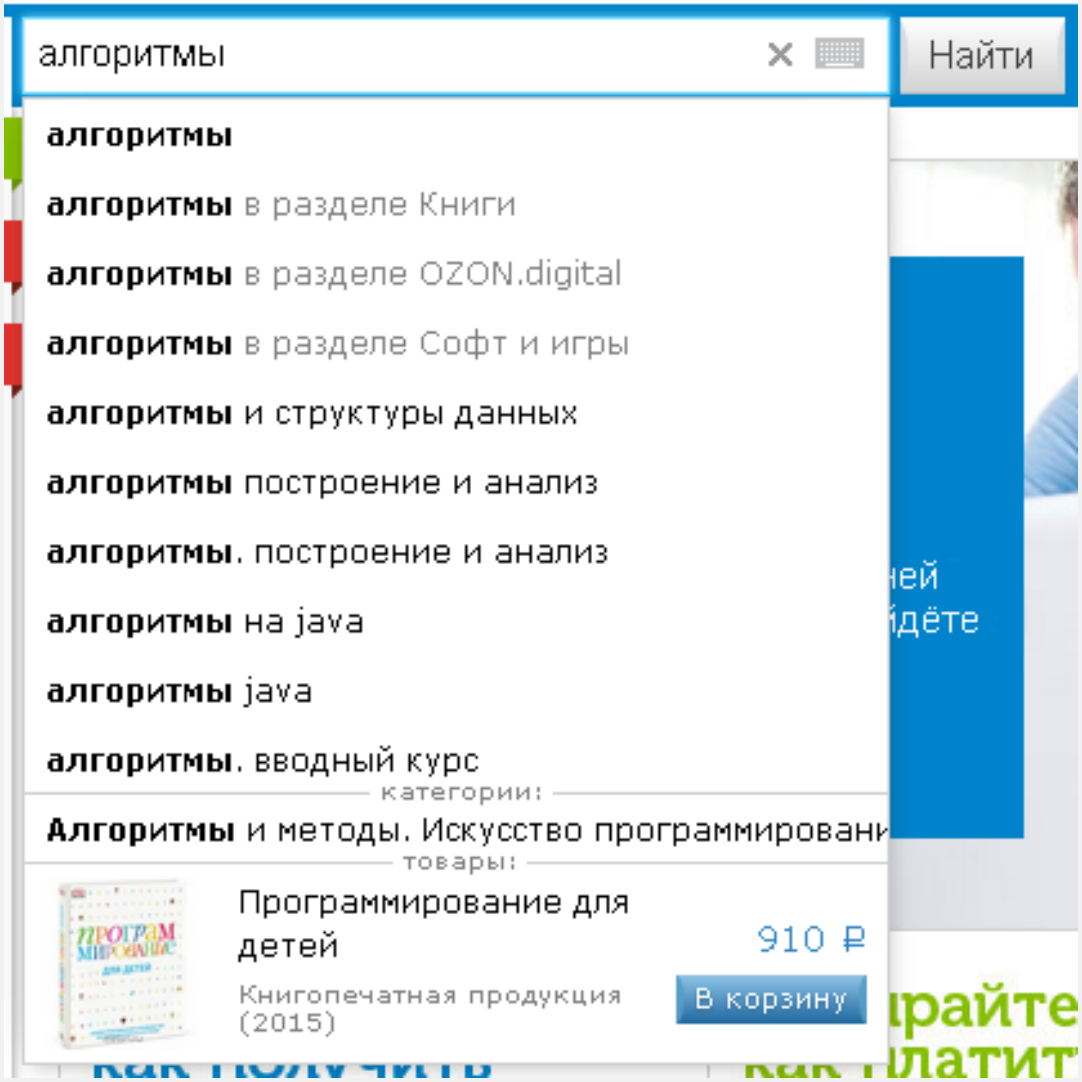
Примеры подсказок

kinopoisk.ru



Примеры подсказок

ozon.ru

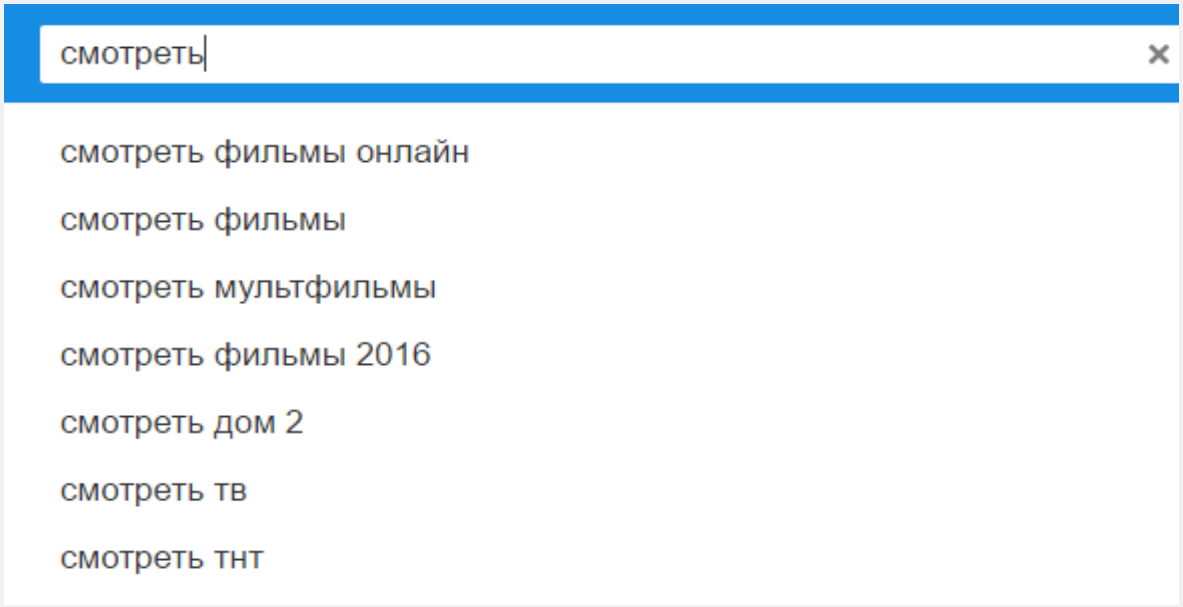


Примеры подсказок

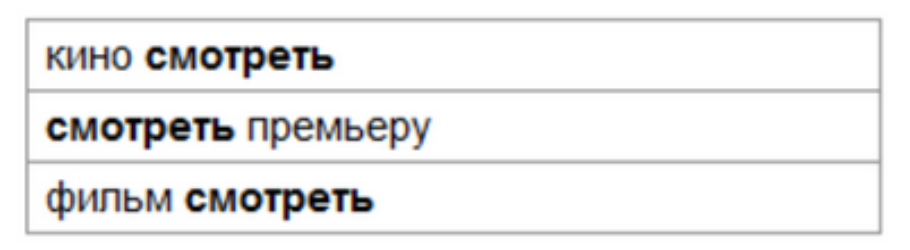
market.yandex.ru

Типы подсказок

- Префиксное дополнение



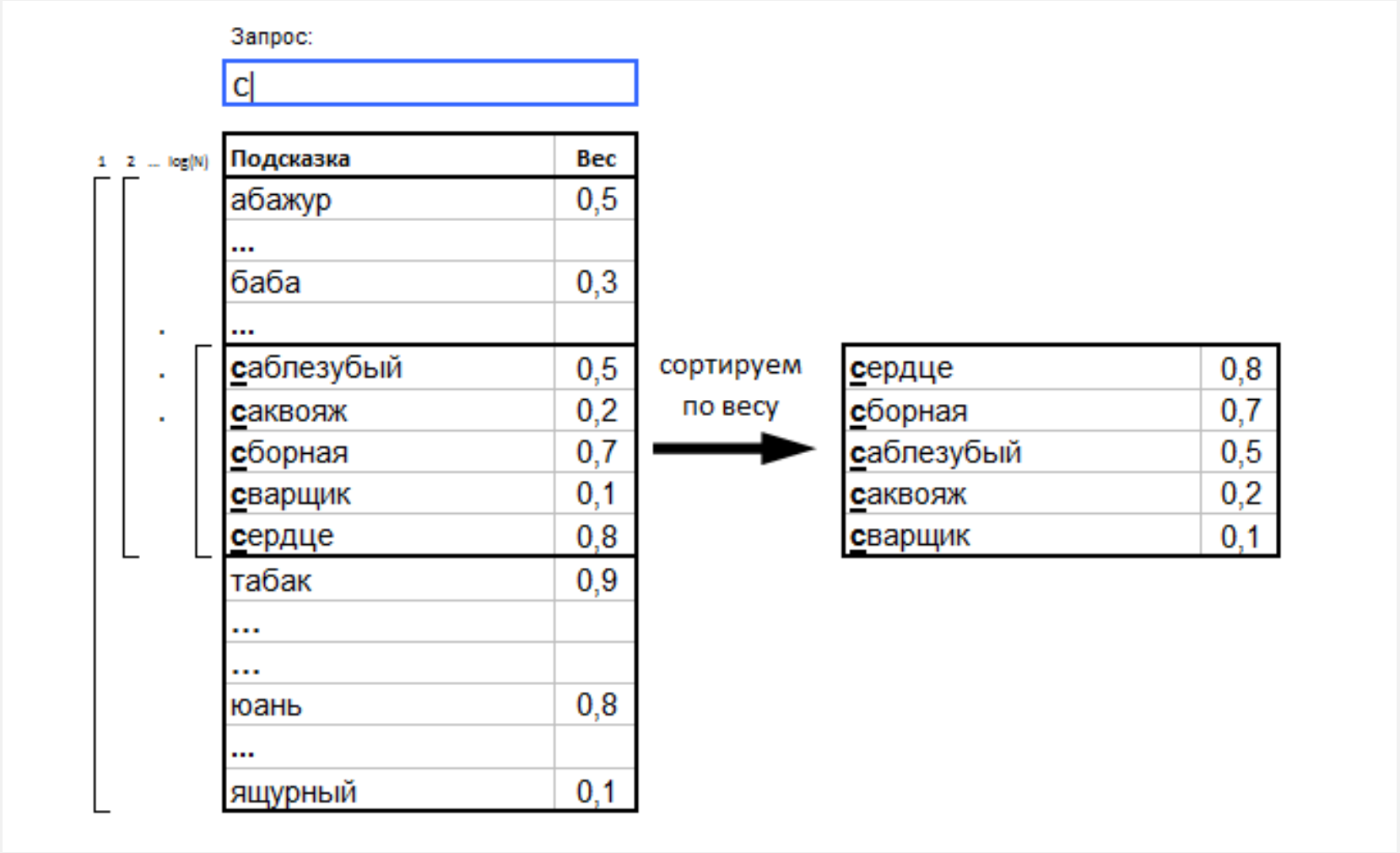
- Полнотекстовое дополнение



Префиксное дополнение

- Подсказки с весами (вес == популярность) сортируются в лексикографическом порядке по текстам подсказок
- Когда пользователь вводит запрос (префикс), бинарным поиском находится подмножество подсказок
- Найденное подмножество сортируется по убыванию веса, ТОП самых «тяжёлых» подсказок отдаётся пользователю

Префиксное дополнение



Префиксное дополнение

+ Скорость $\log(N)$
+ Простая реализация
– Невозможность искать перестановки слов

Полнотекстовое дополнение

- Подсказки и запросы рассматриваются как последовательность слов.
- Для запроса в базе ищется множество подсказок, которые содержат все слова пользователя вне зависимости от их позиции
- Найденное подмножество сортируется по убыванию соответствия слов подсказки к словам запроса, затем по убыванию веса и пользователю возвращается ТОП самых «тяжёлых»

Полнотекстовое дополнение

Запрос:

смотр

Подсказка	Вес
кино **смотр**еть	0,3
смотреть премьеру	0,1
новинки кино **смотр**еть	0,4
фильмы **смотр**еть 2014	0,5

сортируем по порядку слов

смотреть премьеру	0,1
кино **смотр**еть	0,3
фильмы **смотр**еть 2014	0,5
новинки кино **смотр**еть	0,4

сортируем по убыванию веса

смотреть премьеру	0,1
фильмы **смотр**еть 2014	0,5
кино **смотр**еть	0,3
новинки кино **смотр**еть	0,4

Постановка задачи

- Имеется некоторое множество $Q = \{q_1, q_1, \ldots, q_n\}$ заранее известных запросов и их весов (частот)
- От пользователя приходит запрос, состоящий из m префиксов слов $P = p_1 p_1 \cdots p_m$
- Требуется для данного запроса P выбрать k самых частотных запросов из Q, в каждом из которых присутствуют слова, начинающиеся с префиксов $p_1, p_1, \cdots, p_m$

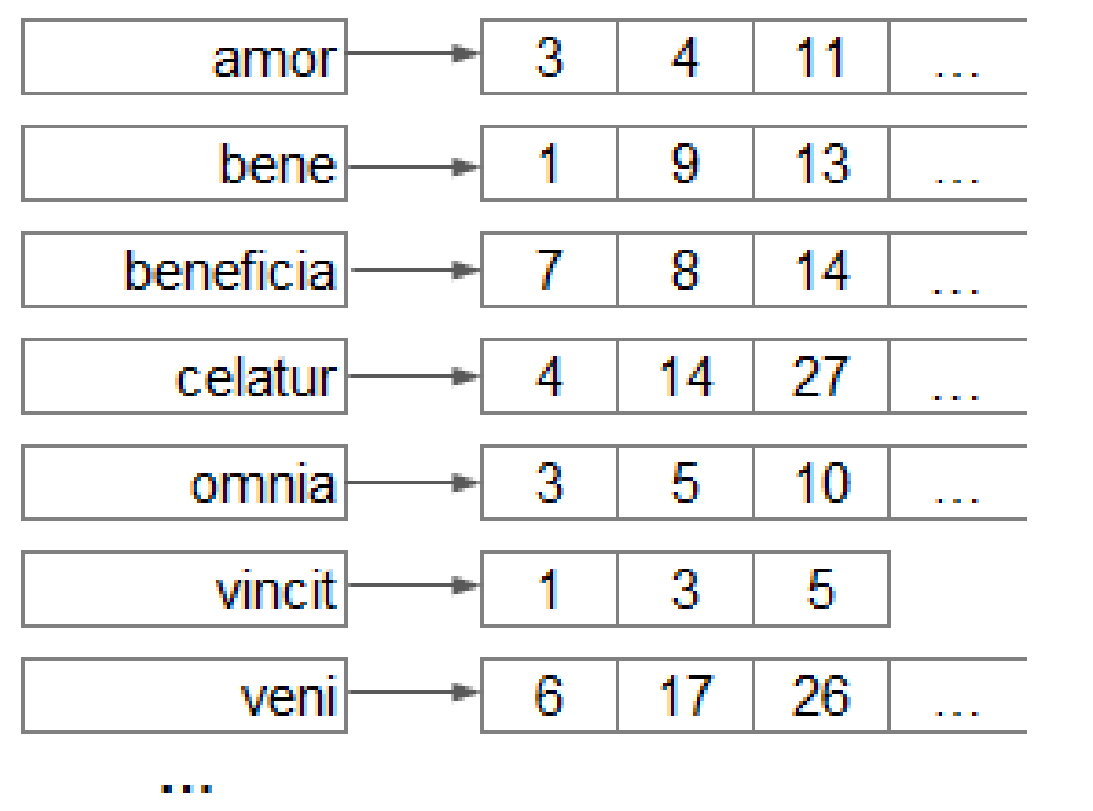
Прямой индекс

Список документов (запросов) для поиска по id

DocID	Text	*f*
1.	Bene vincit, qui se vincit in victoria.	0,5
2.	Nullum periculum sine periculo vincitur.	0,2
3.	Amor vincit omnia.	0,3
4.	Amor et tussis non celatur.	0,1
5.	Labor omnia vincit improbus.	0,7
6.	Veni sancte spiritus.	0,8
…	…	

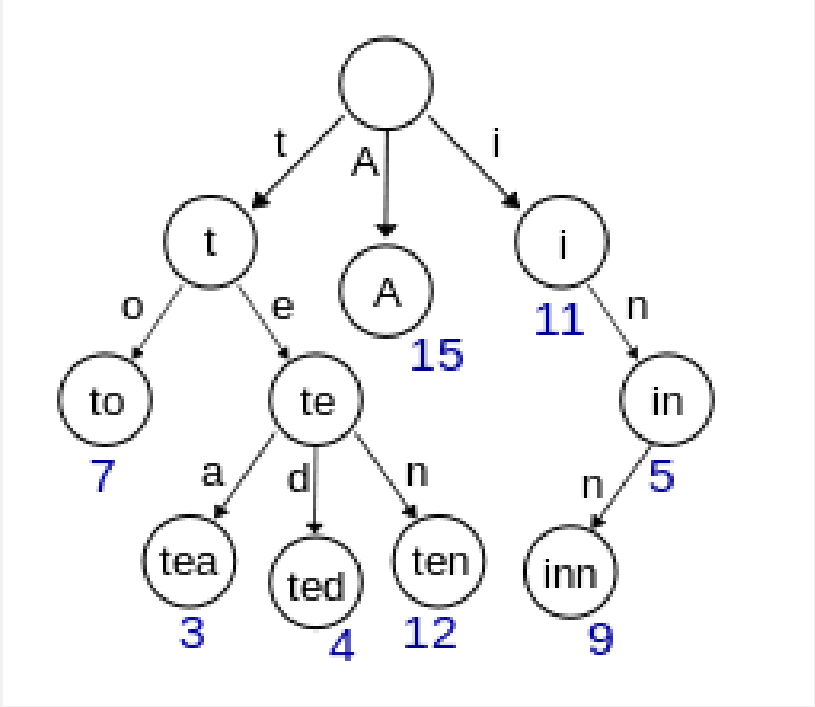
Обратный индекс

По слову находим список запросов с этим словом



Но у нас должен быть поиск по префиксам

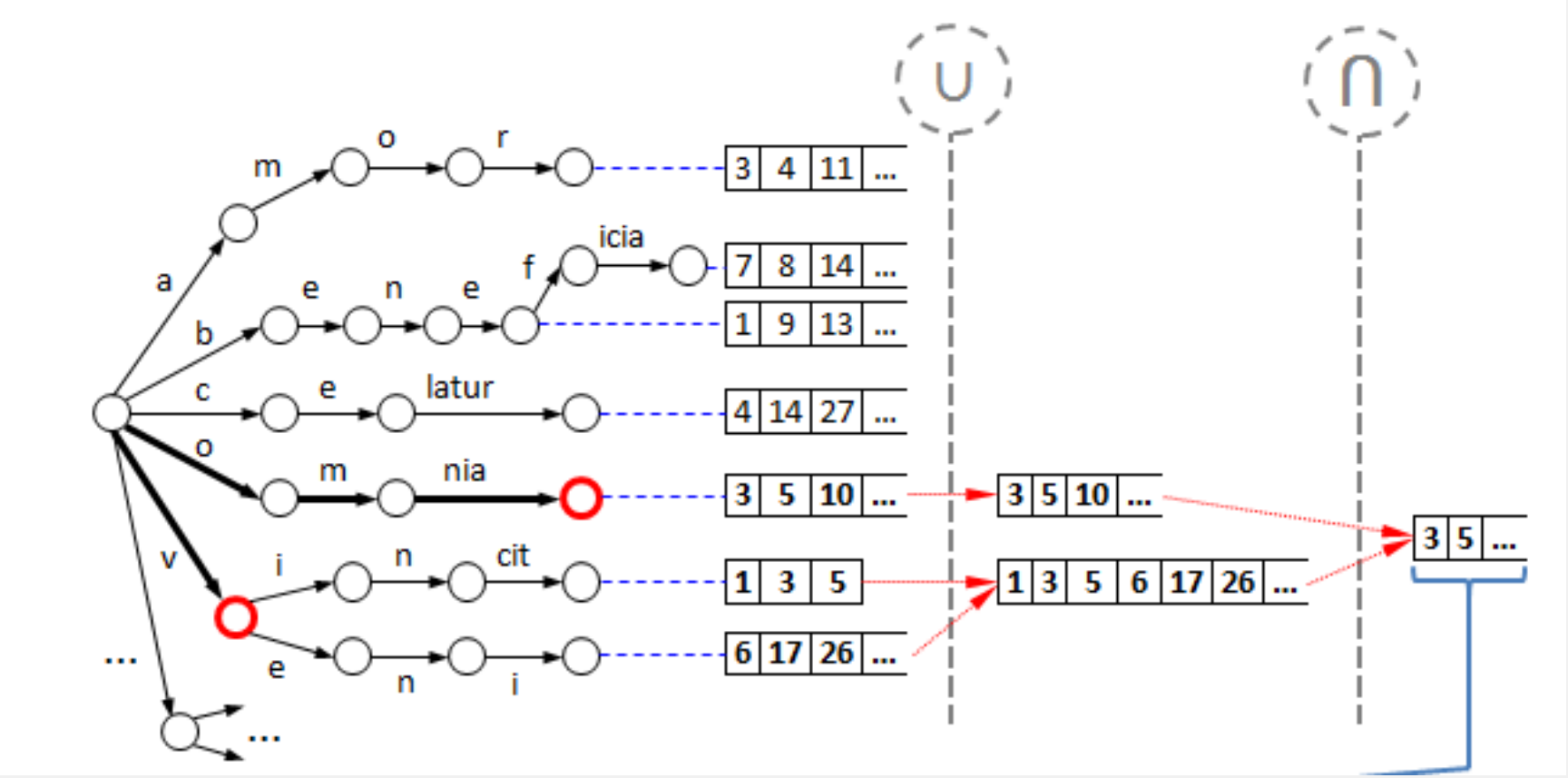
Trie



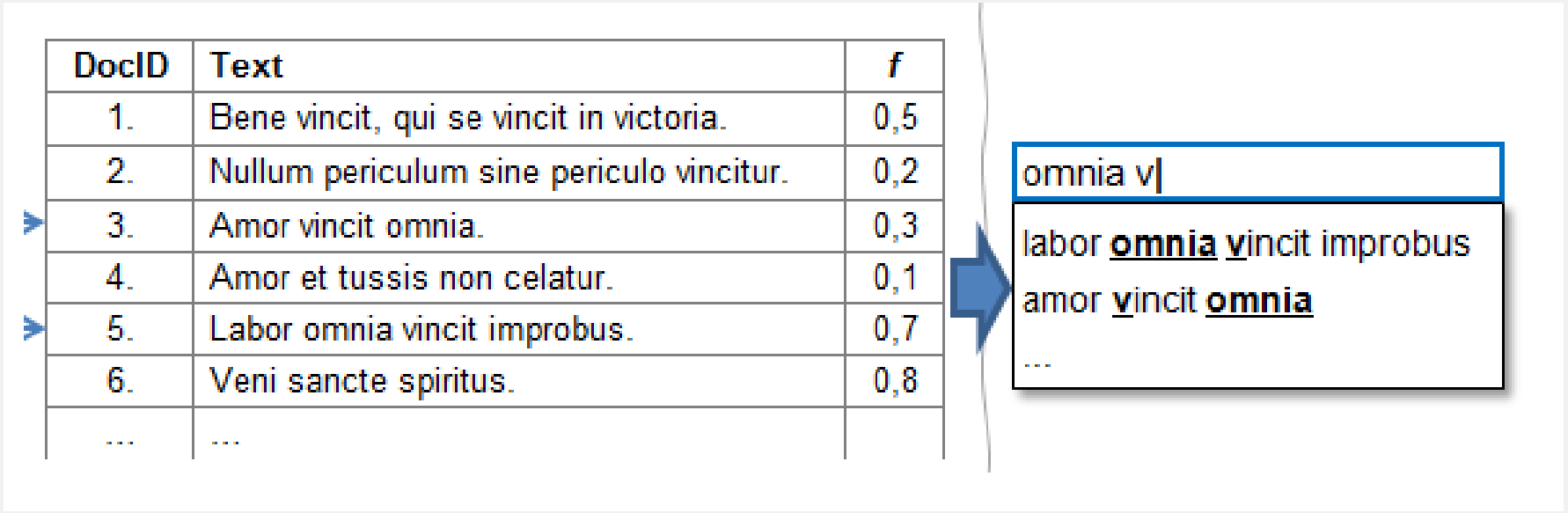
- Поиск префикса за время O(log(n))
- Для префикса легко найти все варианты слов

Trie и обратный индекс

Запрос: *omnia v*



Поиск в прямом индексе



DocID	Text	*f*
1.	Bene vincit, qui se vincit in victoria.	0,5
2.	Nullum periculum sine periculo vincitur.	0,2
3.	Amor vincit omnia.	0,3
4.	Amor et tussis non celatur.	0,1
5.	Labor omnia vincit improbus.	0,7
6.	Veni sancte spiritus.	0,8
…	…	

Проблема скорости №1

Медленное объединение списков:

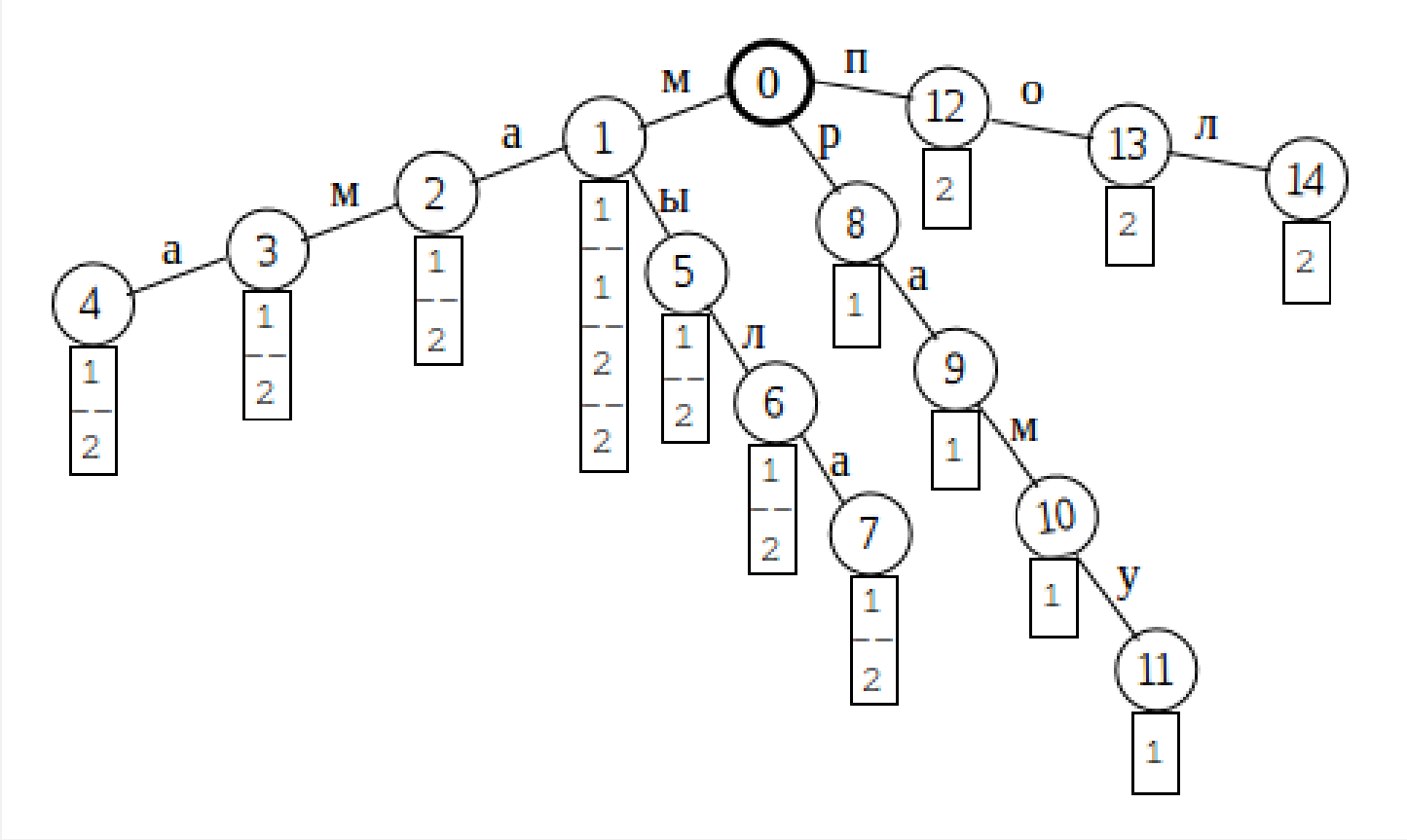
- Пусть есть 1000 слов на букву “а“ (в реальности больше)
- В дереве минимум 1000 списков id (каждый список не маленький)
- При вводе пользователем буквы “а“ происходит объединение 1000 списков

Решение “в лоб”:

- Храним списки id для каждого префикса

Trie + обратный индекс

1. *мама мыла раму*
2. *мама мыла пол*



Поиск в обратном индексе

Запрос “мама мы”:

- разбиваем запрос на префиксы: "*мама*" и "*мы*"
- для каждого префикса ищем список id запросов (узлы 4 и 5)
- находим пересечение этих списков: [1, 2]

Размер структуры:

- при $|Q| = 20$ млн обратный и прямой индекс ~5.5Гб

Проблемы скорости №2,3

Медленное пересечение списков:

- В большой базе списки большие, особенно для коротких префиксов

Ранжирование:

- Нужно отсортировать по весу большие списки

Решение:

- Для префикса храним уже отсортированные списки
- Пересекаем только первые k элементов списков

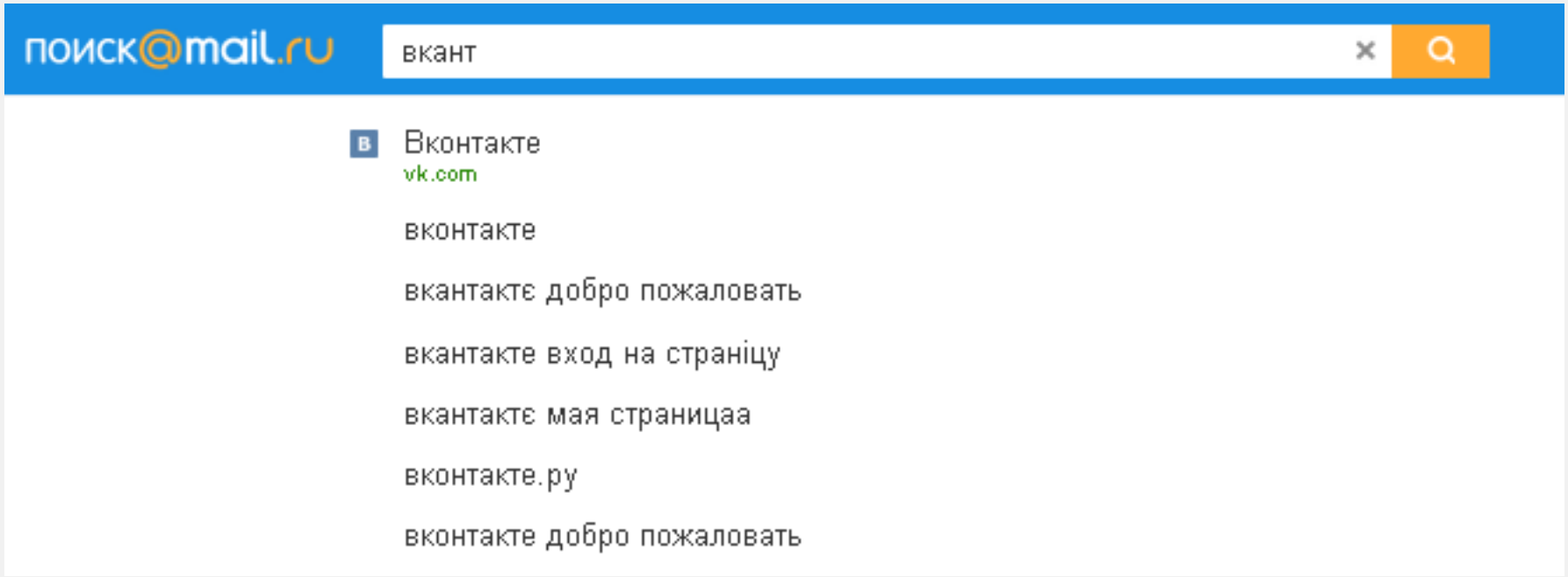
Ранжирование

Факторы:

- порядок слов
- пропуски между словами
- недонабранный хвост слова
- частота запроса
- частота слова

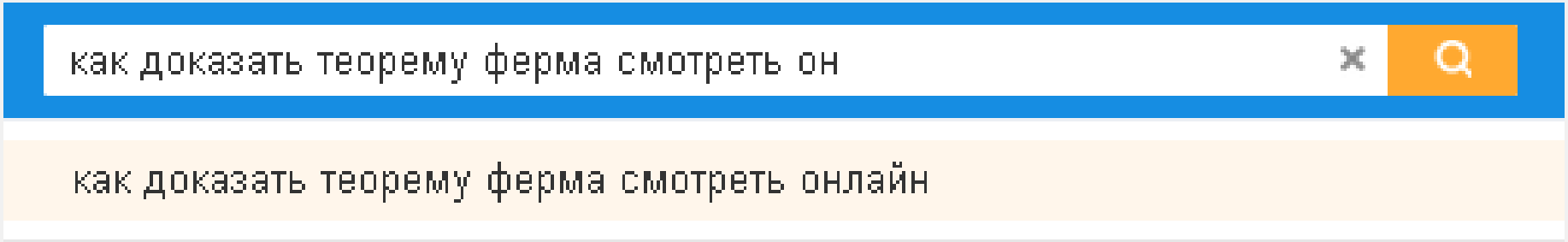
Исправление опечаток

Используем результат опечаточника

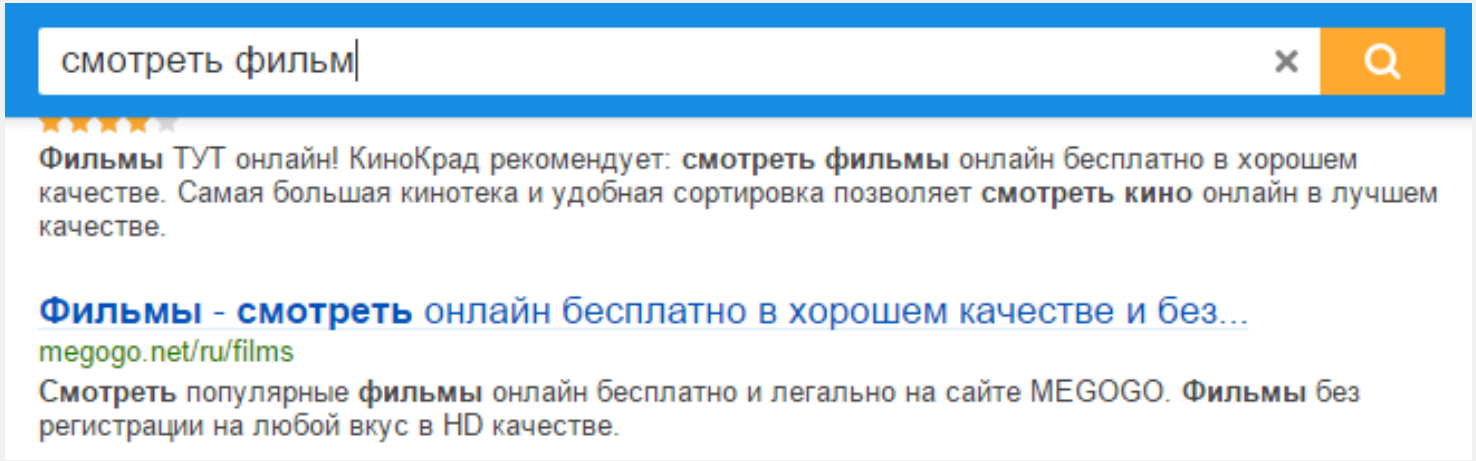


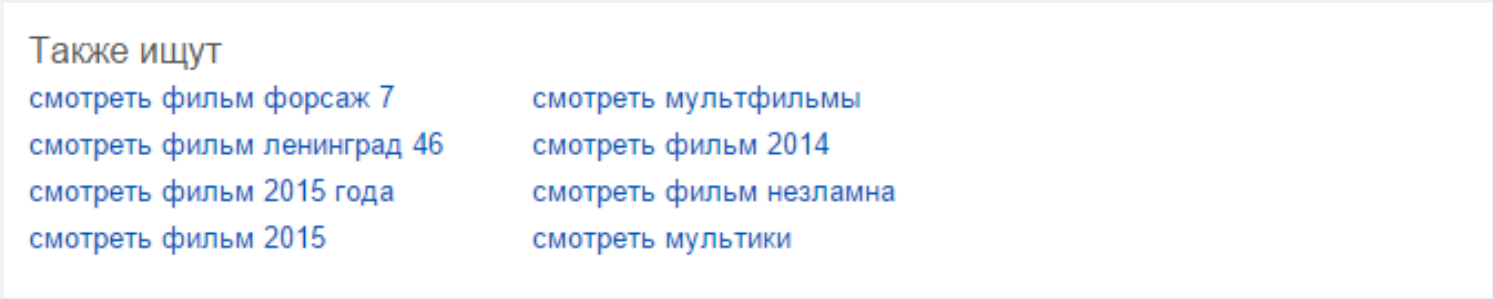
Автодополнение

- Если не можем подсказать весь запрос, подсказываем по нескольким последним словам
- Контекст состоит из 1,2,3 слов



Переформулировки





Что такое переформулировки?

- Набор запросов, которые имеют что-то общее с заданным:
  - *грозный -> {грозный чечня, иван грозный, терек}*
- Данные – логи запросов:
  - Если после запроса A пользователи часто вводят запрос B, то B – переформулировка A
- Для редких запросов статистики мало -> меняем не весь запрос, а его часть:
  - *изготовление кислородного коктейля:*
  - *изготовление –> {рецепт, оборудование для, печать}*
  - *оборудование для кислородного коктейля*
  - *печать для кислородного коктейля*

Поиск устойчивых пар запросов

Рассмотрим пару запросов (q_1, q_2)

Если появление q_2 зависит от q_1, то пара устойчивая и q_2 является переформулировкой q_1

Для определения зависимости используем логарифмическое отношение максимального правдоподобия (**Log Likelihood Ratio – LLR**)

Условные обозначения

N – общее количество запросов

c_1 – количество появлений q_1

c_2 – количество появлений q_2

$c_{1,2}$ – количество появлений $q_1 \rightarrow q_2$

Количество выборок:

- $q_1 \rightarrow q_2$: $c_{1,2}$ штук
- $\neg q_1 \rightarrow q_2$: $(c_2 - c_{1,2})$ штук
- $q_1 \rightarrow \neg q_2$: $(c_1 - c_{1,2})$ штук
- $\neg q_1 \rightarrow \neg q_2$: $(N - c_1 - c_2 + c_{1,2})$ штук

Функция правдоподобия

Произведение вероятностей появления всех элементов выборки:

$$\wedge = p(q_2|q_1)^{c_{1,2}} p(q_2|\neg q_1)^{c_2 - c_{1,2}} p(\neg q_2|q_1)^{c_1 - c_{1,2}} p(\neg q_2|\neg q_1)^{N - c_1 - c_2 + c_{1,2}}$$

Гипотеза 1

Запросы независимы:

$$P(q_2|q_1) = p = P(q_2|\neg q_1)$$

Тогда:

- $p = \frac{c_2}{N}$
- $P(\neg q_2|q_1) = P(\neg q_2|\neg q_1) = 1 - p$

Гипотеза 2

Запросы зависимы:

$$P(q_2|q_1) = p_1 \neq p_2 = P(q_2|\neg q_1)$$

- $p_1 = \frac{c_{1,2}}{c_1}$
- $p_2 = \frac{c_2 - c_{1,2}}{N - c_1}$
- $P(\neg q_2|q_1) = 1 - p_1$
- $P(\neg q_2|\neg q_1) = 1 - p_2$

Правдоподобие гипотезы 2

$$LLR = -2\log\frac{\Lambda_1}{\Lambda_2} = -2\log\Lambda_1 + 2\log\Lambda_2$$

$$LLR = -2(\log L(c_{1,2}, c_1, p) + \log L(c_2 - c_{1,2}, N - c_1, p) - \log L(c_{1,2}, c_1, p_1) - \log L(c_2 - c_{1,2}, N - c_1, p_2))$$

$$L(k, n, x) = x^k * (1 - x)^{(n-k)}$$

Чем больше LLR, тем больше вероятность, что q_1 и q_2 зависимы

Частотные переформулировки

Для запроса “*документы на загранпаспорт*” может быть переформулировка “*одноклассники*”

Вводим специальное условие:

- если $p_1 < p_2$, то $LLR = 0$

Подстановки для частей

Делим запрос на части:

- Перебираем все возможные разбиения
- Рассчитываем для них связность
- Выбираем вариант с максимальной суммой связностей

Пример:

карта ростовской области –> (карта) (ростовской области)

Связность части:

$$c(s) = freq(s)MI(w_1 \dots w_{n-1}, w_2 \dots w_n), \text{где } s - w_1 \dots w_n$$

$MI(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$ – точечная взаимная информация

(Pointwise Mutual Information)

Получение частей

Из двух запросов, отличающихся одной частью, получаем пару частей:

(ежики с рисом) (рецепты)

(ежики с рисом) (приготовление)

Получили части: *рецепты -> приготовление*

Ранжирование

Машинное обучение на факторах:

- LLR
- word-dist – нормализованное расстояние Левенштейна по словам:
  - lev(*день рождения Ленина, день рождения Адольфа Гитлера*) = 2/4 =0.5
- длина наибольшего общего префикса
- частоты запросов и их отношения

Проблемы

Основная проблема – шумы

Решение – фильтрация данных на разных этапах:

- запросы от роботов
- запросы из офиса
- запросы не с первой страницы
- запросы меньше 3 символов
- совпадение запросов по набору лемм
- запросы с ссылками

Классификаторы

Типы запросов

- Гео-запросы:
  - *аптеки*
  - *авто мойка*
- Навигационные запросы
  - *вконтакте*
  - *лента ру*
- Фильмо-запросы
  - *смотреть онлайн форсаж*
  - *аватар*
- Порно-запросы
  - *секс онлайн*

Запросы с подмесами

картинок:

- *цветы*
- *обои на рабочий стол*

видео

- *смешные ролики*
- *видео падения самолета*

карт

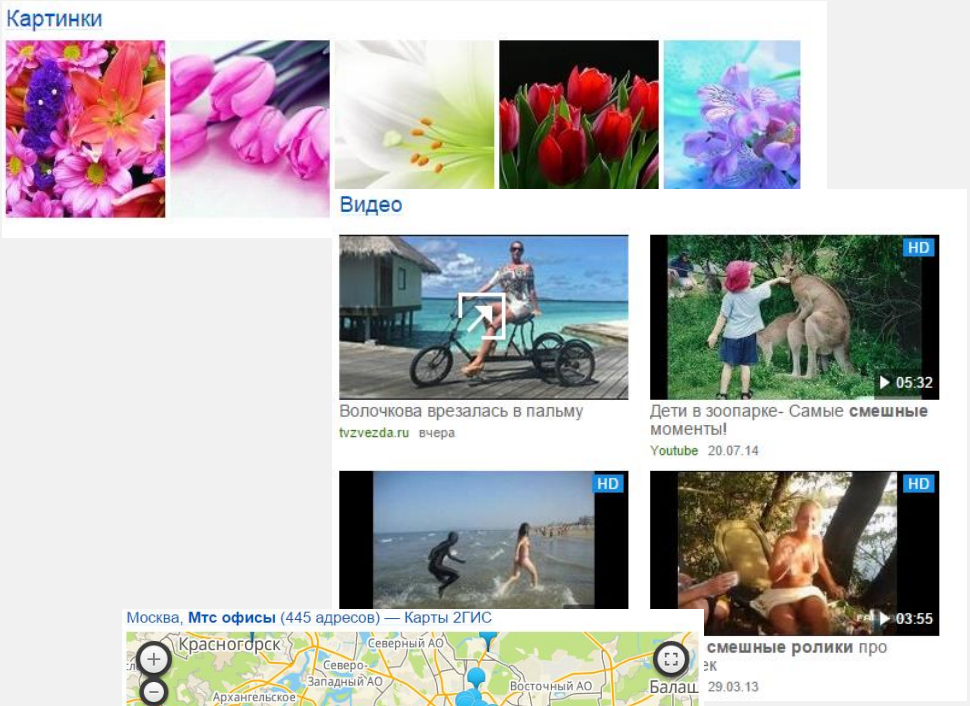
- *фитнес worldclass*
- *адреса офисов МТС*

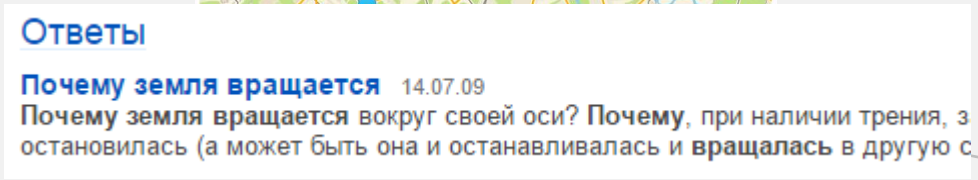
ответов

- *почему земля вращается*

товаров

- *купить iphone 6*





Способы классификации

1. Маркеры / регэкспы
   - **картинки** с днем рождения
   - **смотреть** фильмы онлайн
   - **рецепт** стейка из семги
2. Языковая статистика
   - вычисляем вероятности принадлежности слов нужному классу
3. Анализ выдачи
   - смотрим распределение типичных для класса сайтов в выдаче

Маркеры / регэкспы

+ Простота создания маркеров и реализации
- Низкая полнота
  - разные формы маркера:
    - *картинки с днем рождения*
    - *картиночки с днем рождения*
    - *хочу картинок с днем рождения*
  - отсутствие маркеров:
    - *открытка с днем рождения*
    - *на рабочий стол*
    - *борщ*
- Низкая точность:
  - *рецепт здорового образа жизни*
  - *как сделать цветное фото*
  - *гадание по фото*

Языковая статистика

+ Более общий подход, чем маркеры
+ Можно использовать n-граммную статистику
- Требуется качественное множество запросов из нужного класса:
  - нужно обеспечить полноту
  - нужно обеспечить точность
- Множество должно быть актуальным
  - статистика языка со временем сильно меняется

Анализ выдачи

+ Не требует разметки сайтов:
  - запоминаем распределение сайтов, которые отвечают на запросы из обучающего множества
+ Не нужно пересобирать обучающий корпус:
  - результат классификации меняется при изменении индекса:
  - *цезарь* – рецептный запрос или нет?
– Работает только на множествах запросы / сайты, замкнутых относительно друг друга:
  - по запросам класса поднимается определенное множество сайтов
  - это множество сайтов поднимается только по запросам данного класса

Фильмо-запросы. Маркеры

- фильм
- кинофильм
- сериал
- ситком
- аниме
- мультик
- трейлер
- серия
- сезон
- в хорошем качестве

- Запросы из названий фильмов:
  - *аватар*
  - *терминатор*
  - *одноклассники*
- Ложные срабатывания:
  - *номер и серия паспорта*
  - *летний сезон*
  - *боб фильм*
  - *поиск по сериалам*

Фильмо-запросы. Языковая модель

- Часто появляются новые фильмы:
  - Боги Египта
  - Руфус
  - Энгри Бердс
  - Хани мани
  - Запрос в друзья
- Популярные фильмы становятся менее популярными
- Требуется достаточно частые обновления моделей

Фильмо-запросы. Анализ выдачи

Около 10000 сайтов появляются по фильмо-запросам:

- kinogo.net
- onlinemultfilmy.ru
- bobfilm.net
- afisha.mail.ru

Запросы не фильмы фильмовыми не становятся:

- *одноклассники*
- *победа*

Множество самобалансируется:

- *вор* – сейчас фильмо-запрос, может им перестать быть

Не совсем замкнутые множества:

- сайты поднимаются по запросам актеров: *кира найтли*
- сайты поднимаются по навигационным запросам: *боб фильм*

Классификатор подмеса картинок. Маркеры

- фото
- фотка
- фотография
- картинка
- картина
- photo
- image
- picture

- Запросы без маркеров:
  - *схема метро*
  - *пингвины*
  - *расскраски*
- Ложные срабатывания:
  - *как обрезать фото в vk*
  - *поиск по фото*
  - *фото.ру*
  - *изготовление фото*

Классификатор подмеса картинок. Языковая статистика.

Изменение статистики происходит нечасто

По размеченному множеству запросов высчитывается статистика для каждого слова:

- фото 91,2%
- картинка 86,5%

Отдельно считаем статистику по словосочетаниям и сохраняем сильно связные:

- четыре **всадника апокалипсиса**
- **образец портфолио** группы
- фигурист **руслан жиганшин**
- озеро **северной америки**
- **нива шевроле** новая

Классификатор подмеса картинок. Анализ выдачи.

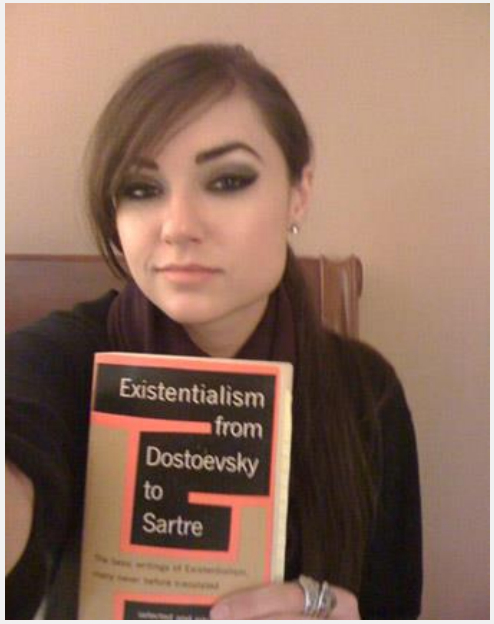
Не работает

Почему?

Множество запросов не замкнуто относительно множества сайтов

Порно-запросы. Маркеры

- Слишком много форм (*порно, порнушечка, порево, порноонлайн*)
- Наличие слова *порно* не всегда говорит о «взрослом» запросе (*Зак и Мири снимают порно*, *незаконное распространение порнографии*)
- Все меняется – тяжело поддерживать
- Морфология часто мешает (*вафли → вафлить*)
- Потеряли из саджестов *чЕБурашку, аЭРОфлот, оПОРНый прыжок*



Порно-запросы. Анализ выдачи

Множество запросов и сайтов хорошо замкнуты друг относительно друга

Но!

По запросам *мулатки, малолетки, бесплатное видео* около 80% страниц из выдачи определялись как порнография

Эти запросы не должны определяться как порно

Помогает языковая статистика:

- общеупотребительные слова имеют низкую вероятность в классе порно

ТЕХНОСФЕРА

Отмечайтесь и оставляйте отзыв

Спасибо за внимание!

Евгений Чернов

e.chernov@corp.mail.ru