



ТЕХНОСФЕРА

Приоритезация краулера

Дмитрий Соловьев, ведущий
разработчик группы ранжирования
проекта Поиск@Mail.Ru

Москва 2016


- <http://go.mail.ru>
- Аудитории поиска:
 - Русская
 - Украинская
 - Казахская
- 9% рынка
- Примерно 100 разработчиков




хью лори фото

Интернет Картинки Видео Новости Обсуждения Ответы

Картинки

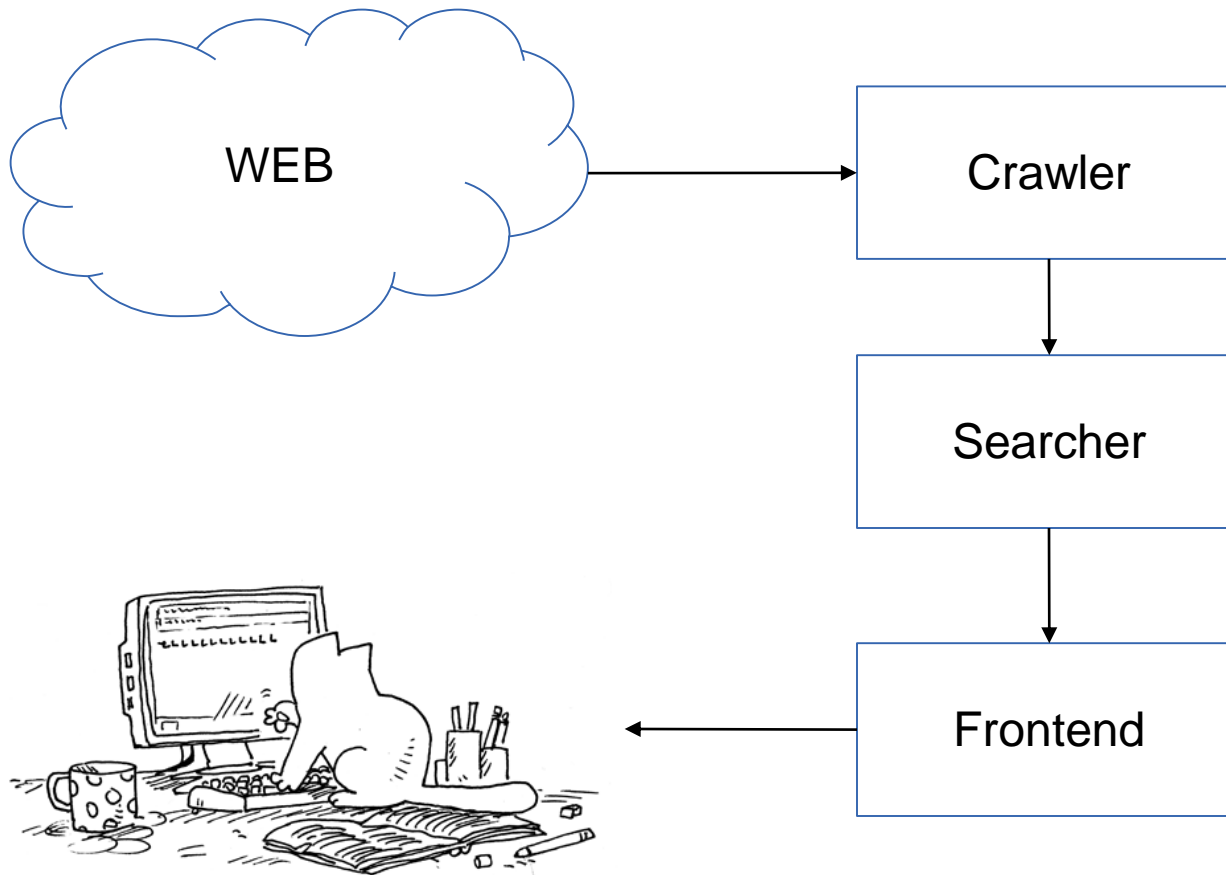


Лори, Хью — Википедия
ru.wikipedia.org/wiki/Лори,_Хью
Родители **Хью Лори** — шотландцы. ... **Хью Лори** и актриса Имельда Стонтон дважды появлялись на экране в роли мужа и жены: в фильме «Разум и чувства» (1995) и «Друзья Питера» (1992). ... **Фотографии** разных лет.



Хью Лори
afisha.mail.ru
Фотографии

Актер **Хью Лори** родился в семье врача и домохозяйки. Учился в частных школах, окончил Кембриджский университет. В университете был участником любительского театра Footlights Dramatic Club. Уже в...



«Running a web crawler is a challenging task»

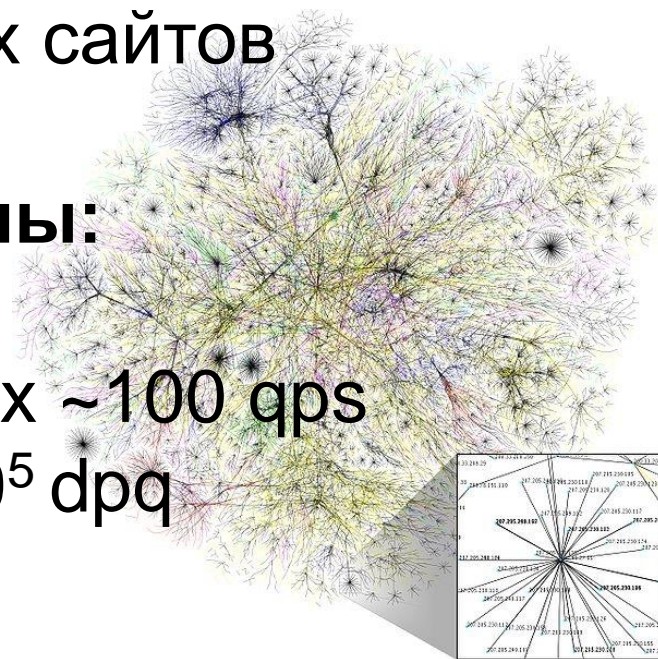
Sergey Brin and Lawrence Page, 1998

Состав:

- Веб сайты: $\sim 10^7$
- Веб страницы: $\sim 10^{10}$
- Урлы:
- 60% покрывают 10000 топовых сайтов

Ограничения поисковой машины:

- Индекс: $\sim 10^9$ страниц
- Пропускной способностью: max ~ 100 qps
- Временем ранжирования: $\sim 10^5$ drq



Краулер: обзор

Scheduler (aka long-term scheduler)

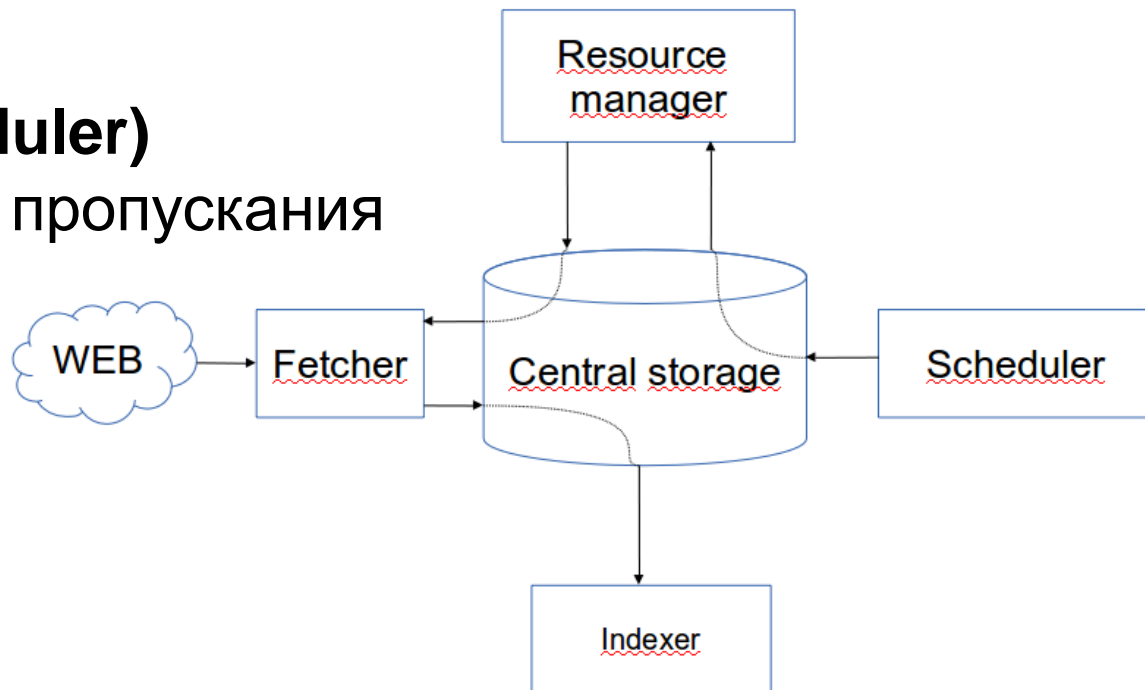
- Политика отбора
- Политика обновления

Resource manager (aka short-term scheduler)

- Контроль полосы пропускания

Fetcher

- Просто WGET



Краулер: обзор

Indexer

- Анализирует скачанный контент

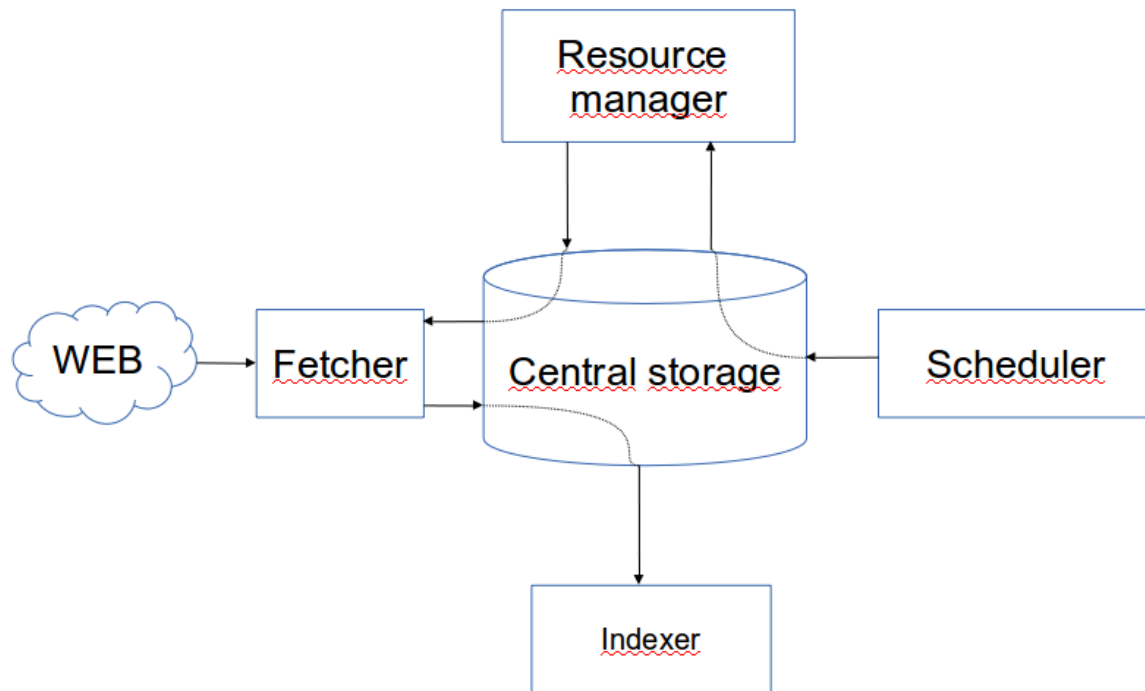
Дополнительные функции

- Логирование
- Статистика

Central storage



APACHE
HBASE



Планировщик, обзор

Примерный набор планировщиков

- Веб поиск:
 - Download scheduler
 - Indexing scheduler
 - Discovery scheduler
 - Sandbox scheduler
 - Analyzer scheduler
- Поиск по картинкам:
 - HTML scheduler
 - Image scheduler
- Эксперименты:

Планировщик: обзор

Цель краулера – загрузка страниц с тематическим контекстом.

Задачи фреймворка краулера:

1. Определить какие страницы *интересны для обхода*
2. Поиск *начального ядра* страниц
3. Используя тематические окрестности (*topical locality*) найти другие страницы

Что интересно качать для веб поиска?

- Интересны страницы с запросными ссылками (*Qlinks*)
- Мы ничего не можем сказать о страницах без запросных ссылок

Тематические окрестности

– Основанные на ссылках

Если $A \rightarrow B$ тогда B подобна A

– Основанные на URL страницы

```
^http://aldebaran.ru/lov/[a-z]+/[a-z]+[0-9]+/$
```

*

```
^http://aldebaran.ru/kid/krapiv/krapiv[0-9]*$
```

*

```
^http://materinstvo.ru/$  
+id+module  
~module=articles
```

– Основанные на содержании страницы

```
title="Анкета заблокирована"
```

```
query="Что вы думаете об этом товаре?"
```

Выделение сегментов сайта

- RFC 1738, RFC 3986
- Нас интересует схема `http` – address:
- `http://<host>:<port>/<path>?<query>#<fragment>`
- `<host>:<port>` - одинаковы для всего сайта
- Нас интересует `<path>` и `<query>`
- `path = segment * ["/" segment]`
 - `segment = * [uchar | ";" | ":" | "@" | "&" | "="]`
- `<query>` состоит из пар `name=value` разделенные `&`
- Порядок следования пар в `<query>` не важен

Анализ малых выборок

Дано:

α - встречаемость урла из группы на сайте

N - размер сэмплирования

Вероятность найти менее чем k урлов:

$$p_{N,k}(\alpha) = \sum_{i=1}^k \binom{i}{N} \alpha^i (1 - \alpha)^{(N-i)}$$

$$P_{1000,10}(0.01) \approx 0.58$$

$$P_{1000,10}(0.02) \approx 0.01$$

$$P_{1000,10}(0.03) \approx 2 \times 10^{-5}$$

石庭(sekitei). Алгоритм

1. Отбираем случайно N урлов
2. Создаем признаки для каждого адреса:
 - количество сегментов
 - список параметров
 - `<parameters=value>`
 -
3. Отбираем признаки по частотности αN
4. Кластеризуем:
 - Jaccard distance measure
 - Stack clustering

$$K(a, b) = \frac{|A \cap B|}{|A \cup B|}$$

石庭. Результаты

1. `^/wiki/File:[^/]+\.`jpg\$

/wiki/File:Spongilla_lacustris.jpg

2. `^/wiki/[^/]+\.`jpg\$

/wiki/Image:Deve.jpg

3. `^/wiki/Category:[^/]+$`

/wiki/Category:Roman-era_historians

4. `^/wiki/Talk:[^/]+$`

/wiki/Talk:North_Light

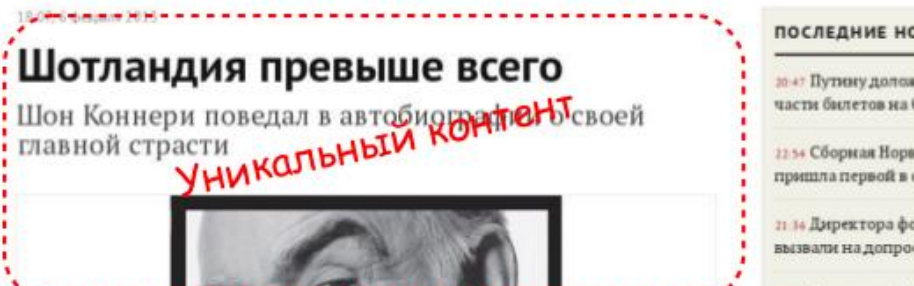
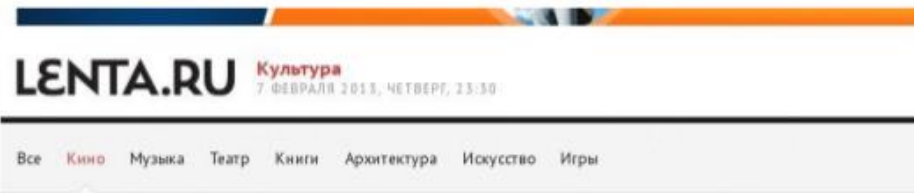
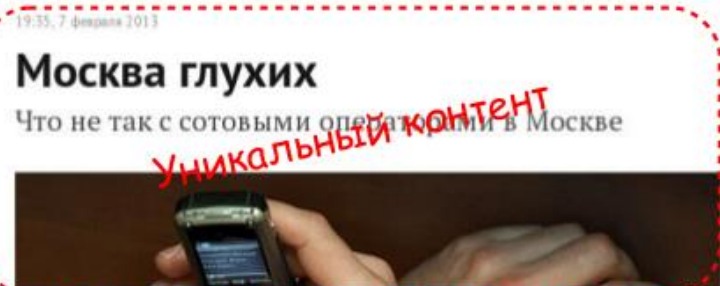
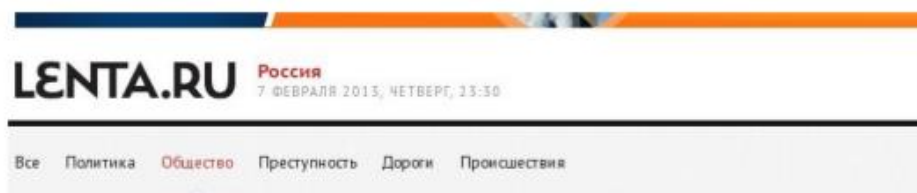
...



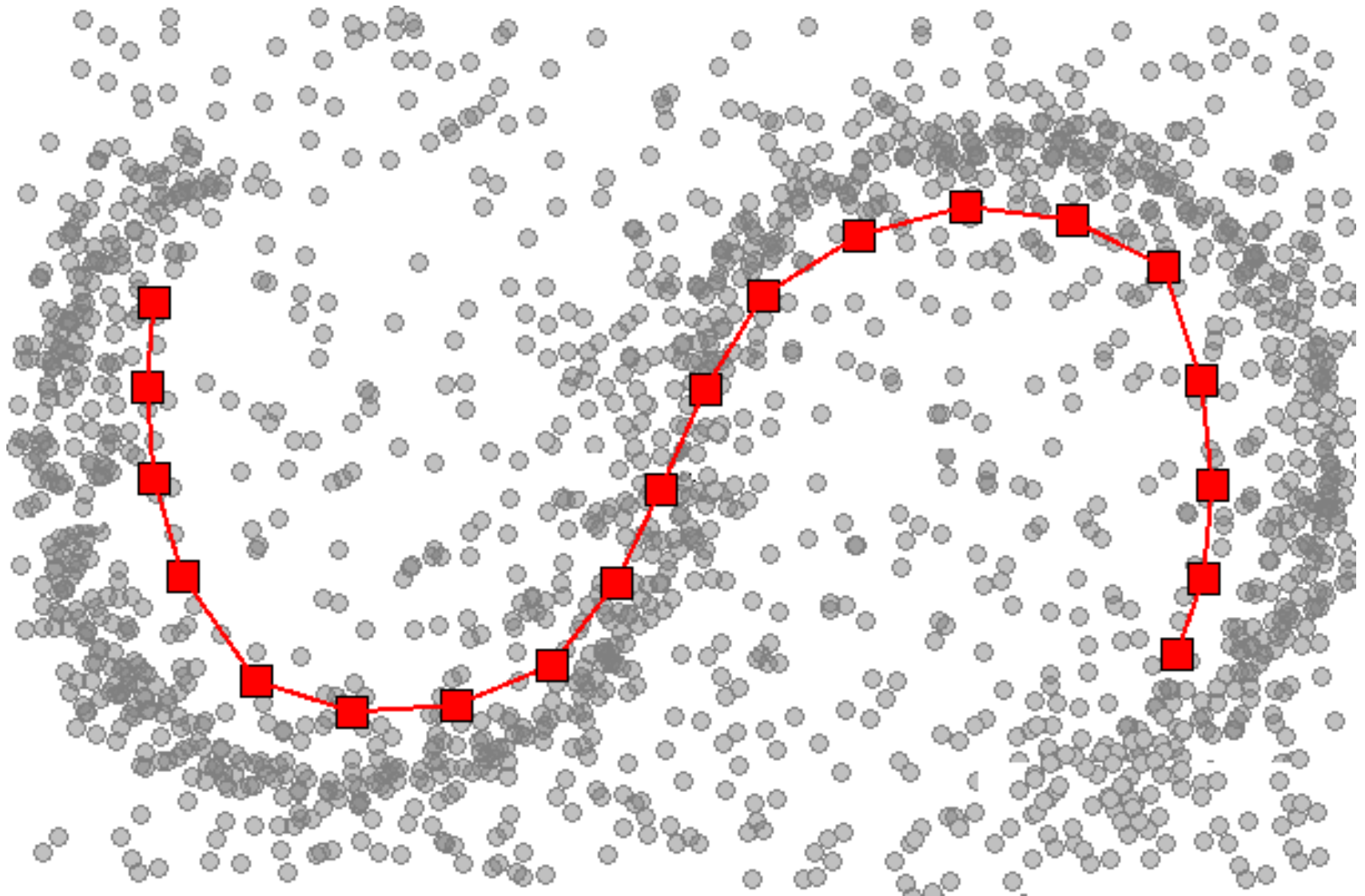
WIKIPEDIA
The Free Encyclopedia

石庭. Применение

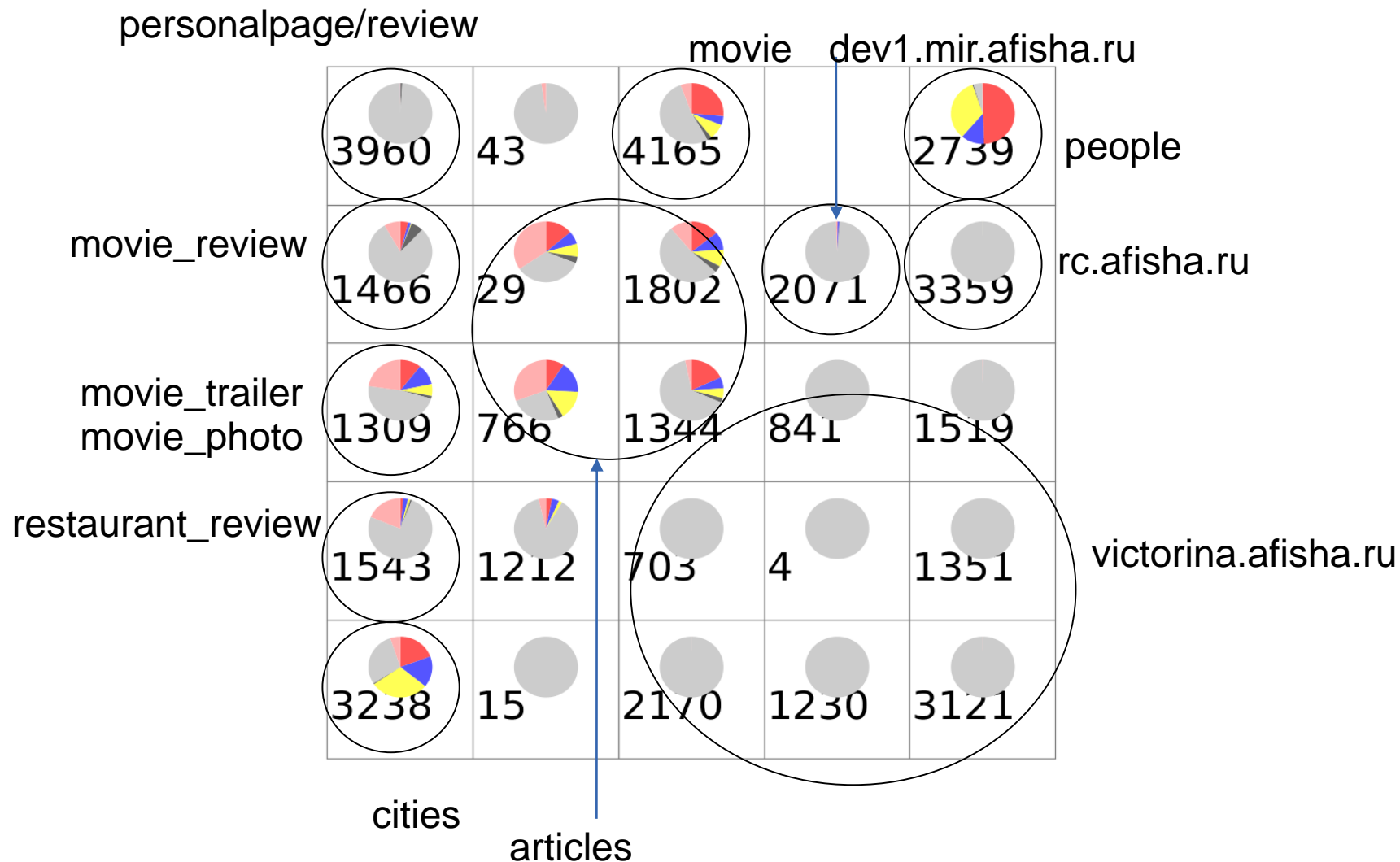
- 404, 500, timeout ...
- спам, порно ...
- выявление дубликатов
- НОВОСТИ
- удаление навигационной обвязки
- ...



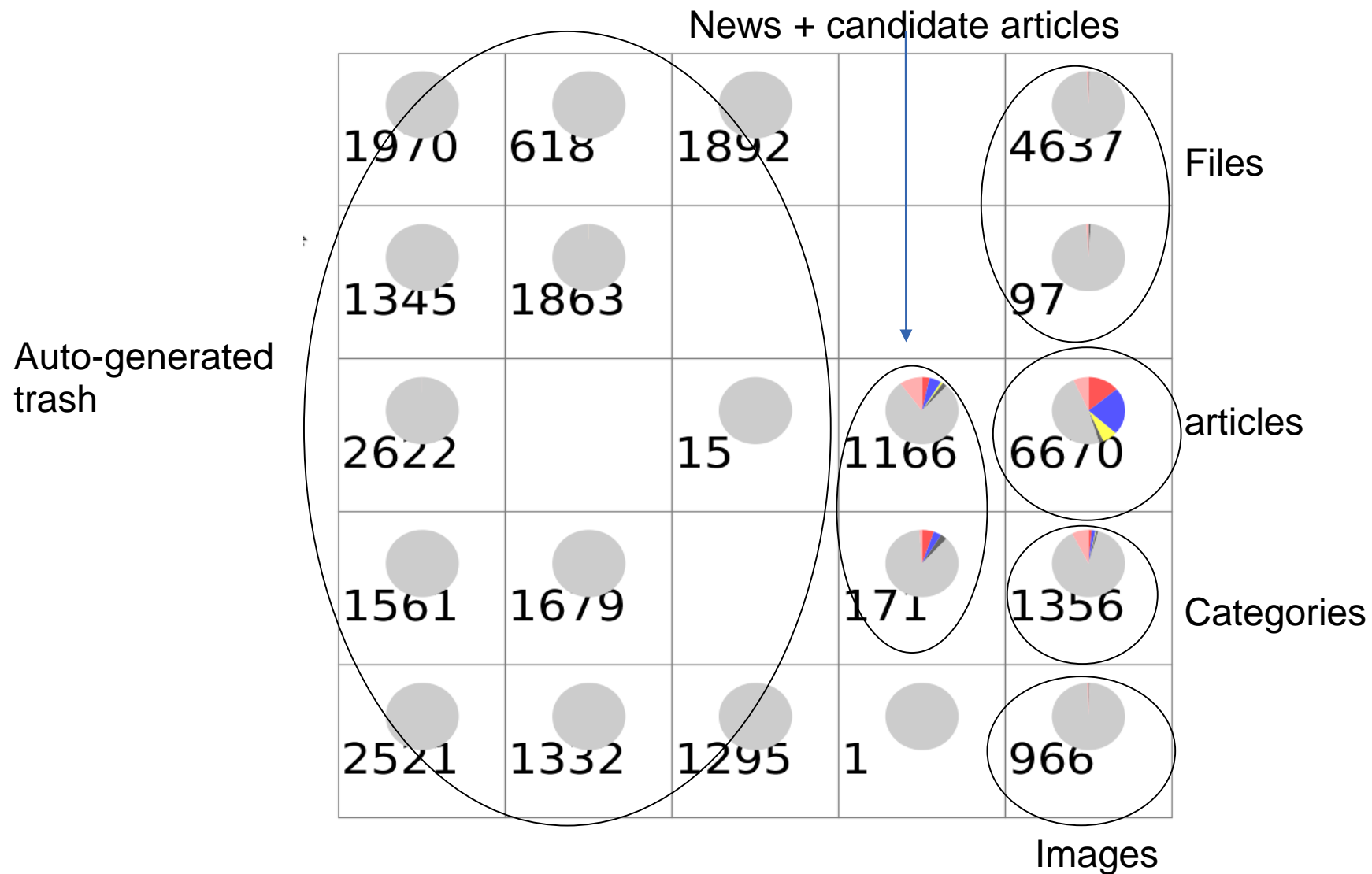
SOM / Kohonen MAP



Анализ кластеров сайта: afisha.ru



Анализ кластеров сайта: absurdopedia.net



Эксперименты с квотированием

Цель: собрать индекс фиксированного размера

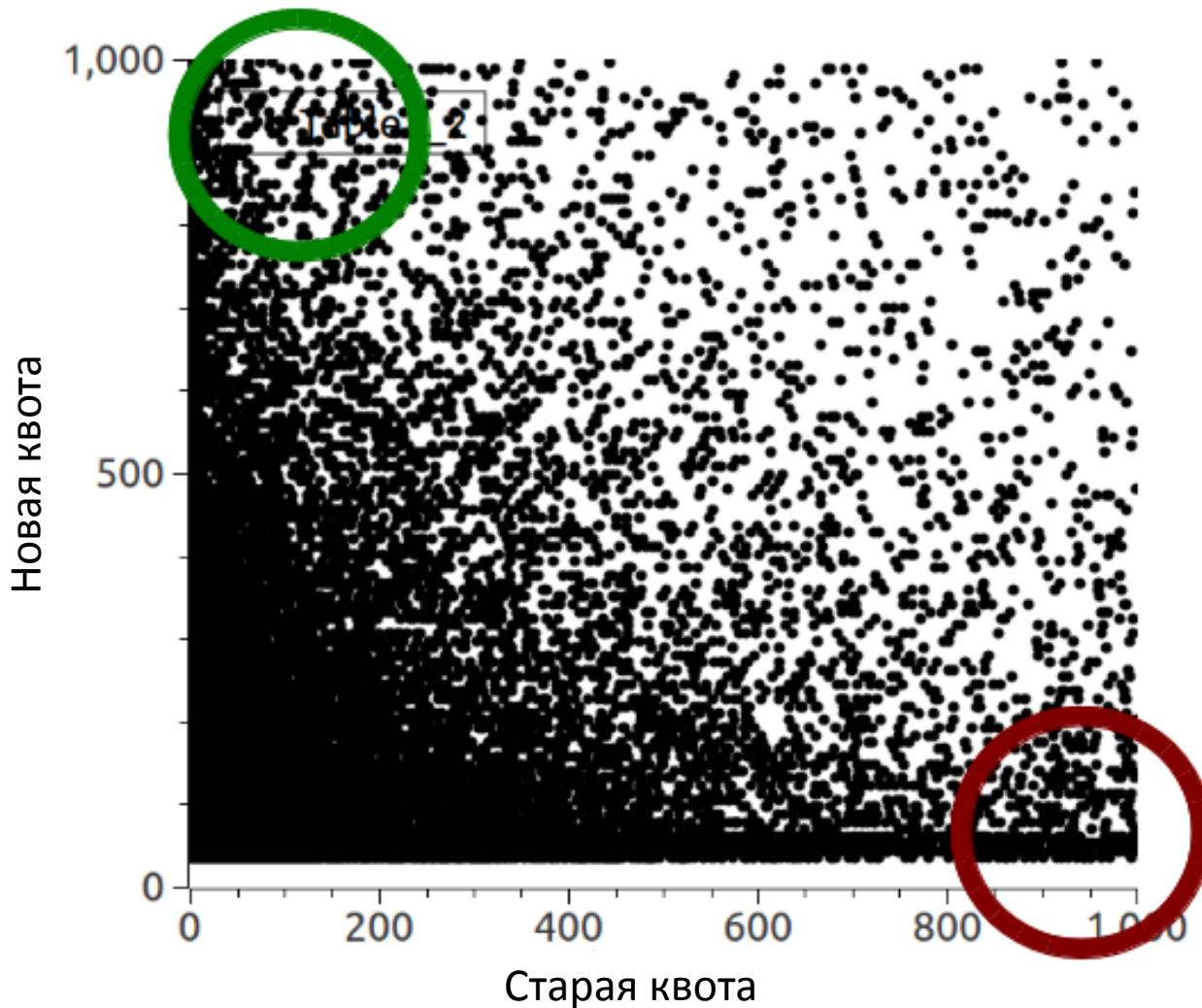
Данные:

- Сайты с хостингов (~ 1000 сайтов в зоне ru) берем домены уровня 3 (zadornov.livejournal.com)
- Остальные сайты берем уровень домена 2 (mail.ru)

Квота:

| Алгоритм Секитей | Жадный алгоритм |
|--------------------------------------|--------------------------------|
| MIN_QUOTA ~ 100 | MIN_QUOTA ~ 100 |
| QUOTA = #PagesWithQlinks * MIN_QUOTA | QUOTA = F(#Visits) * MIN_QUOTA |
| Квота по камням | Квота для сайта |

Квотирование: Секитей vs. жадный алгоритм



Квотирование: Секитей vs. жадный алгоритм

Хорошо

Блоги

blogspot/livejornal

Хорошие сайты

use4blog.com

gagadget.com

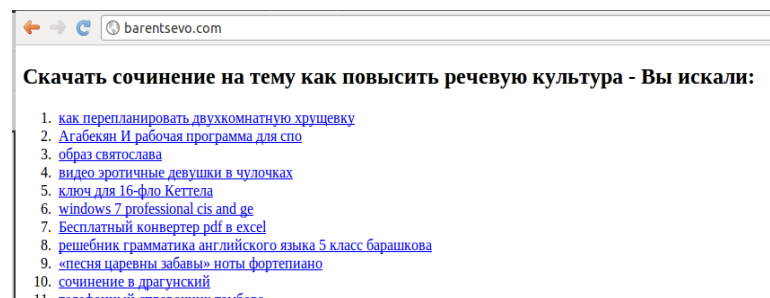
Популярные иностранные сайты

Robots, ban, ...



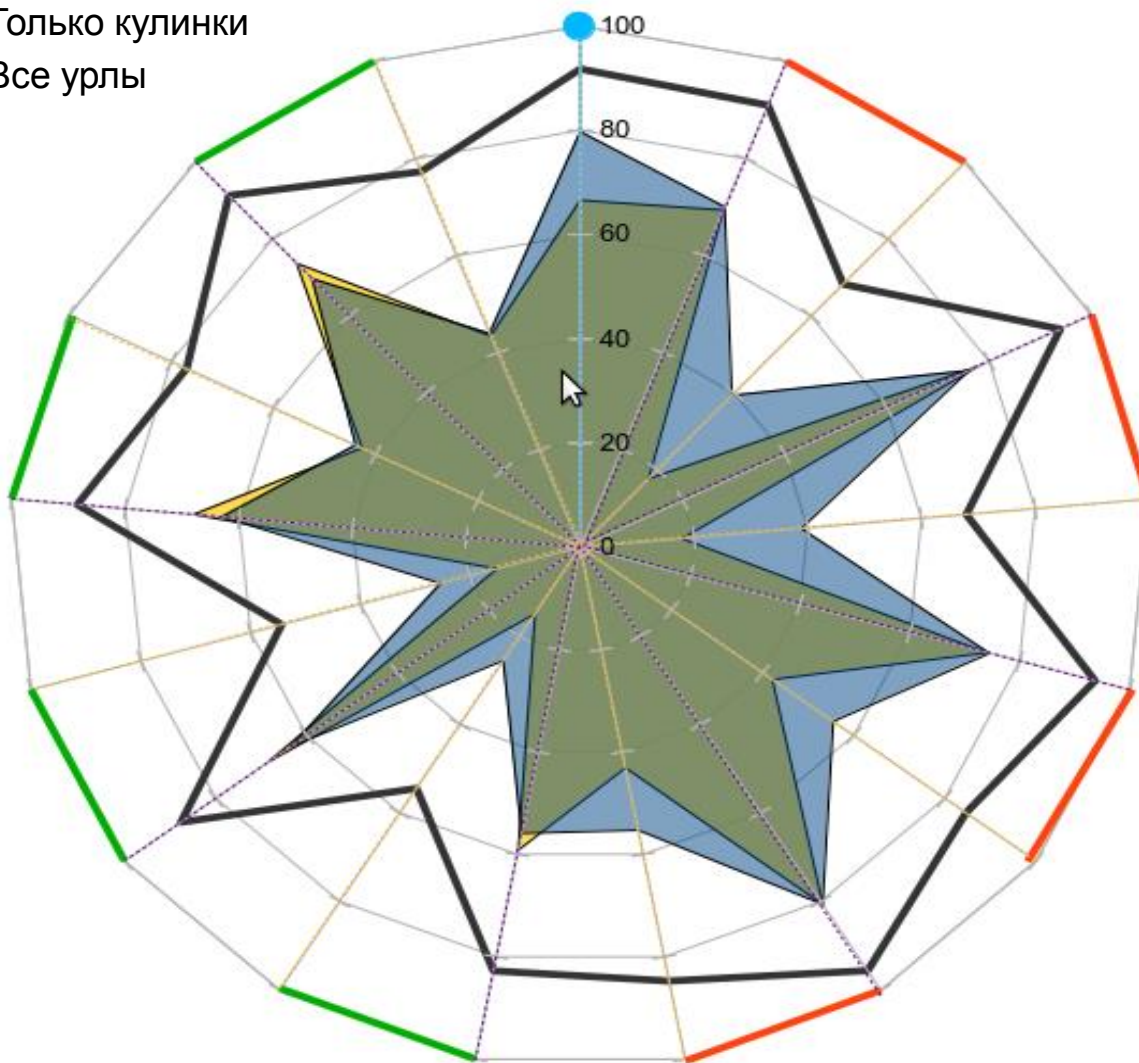
Плохо

Мусорные сайты



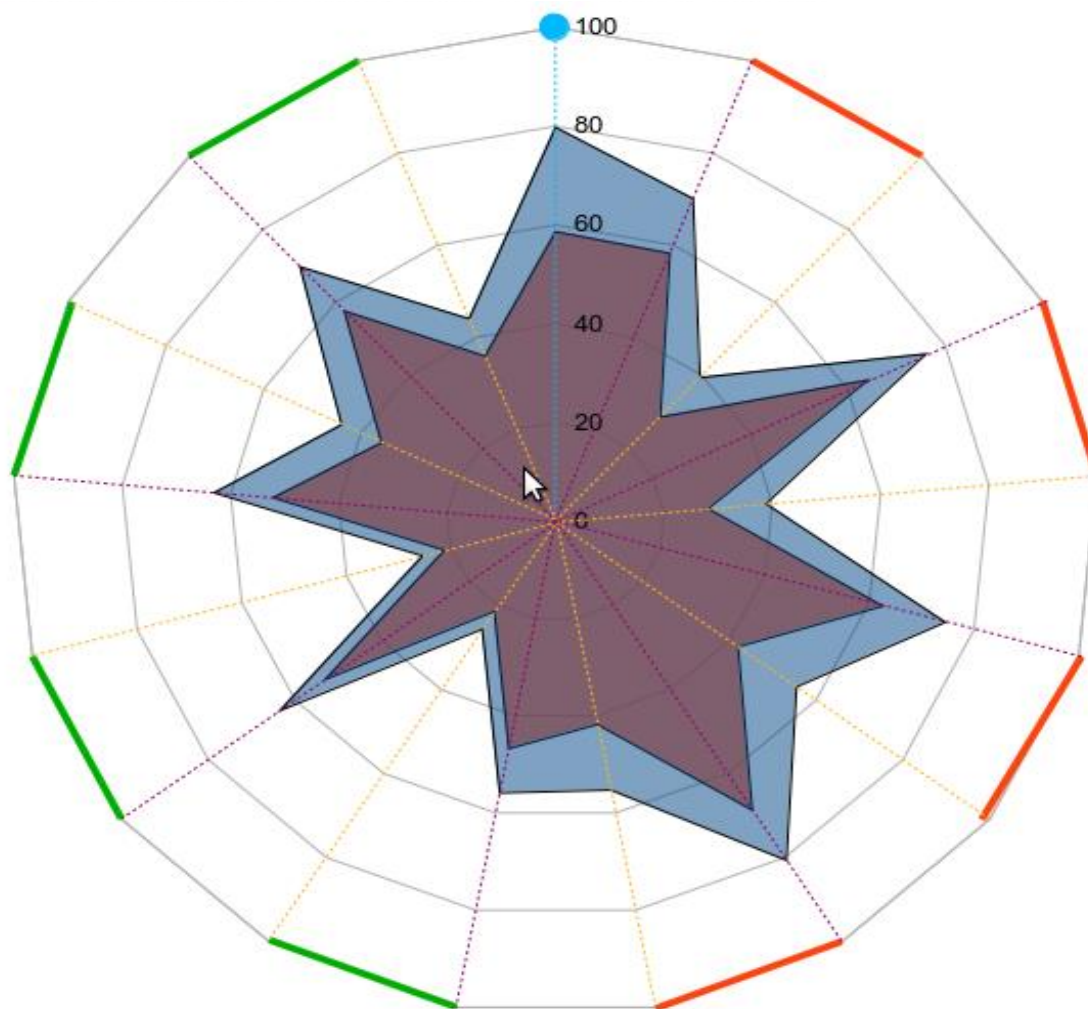
Качество индекса: baseline

- Жадный индекс
- Только кулинки
- Все урлы



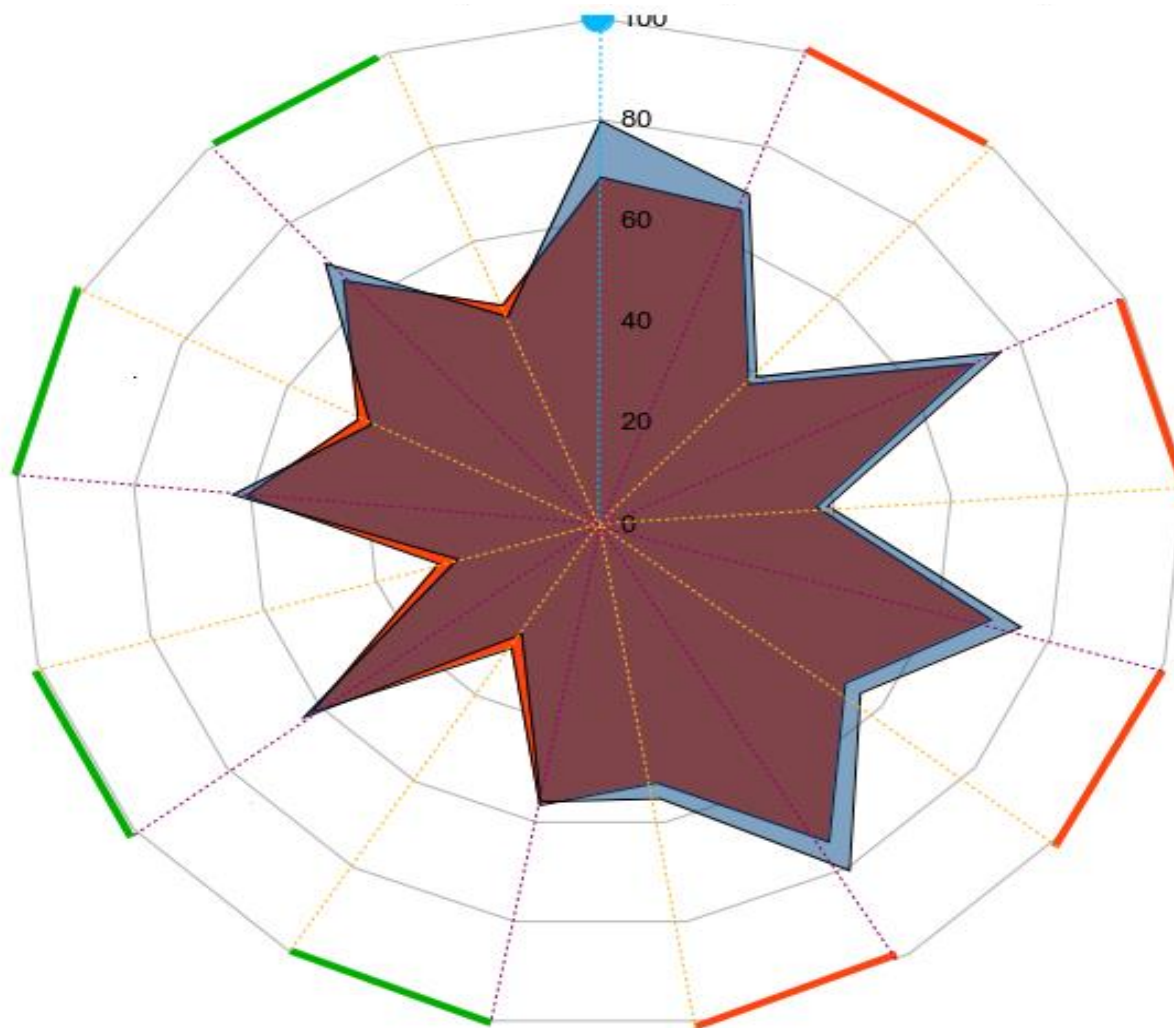
Качество индекса: старая квота

- Жадный индекс
- Жадный индекс, одинаковый размер, меньшая квота



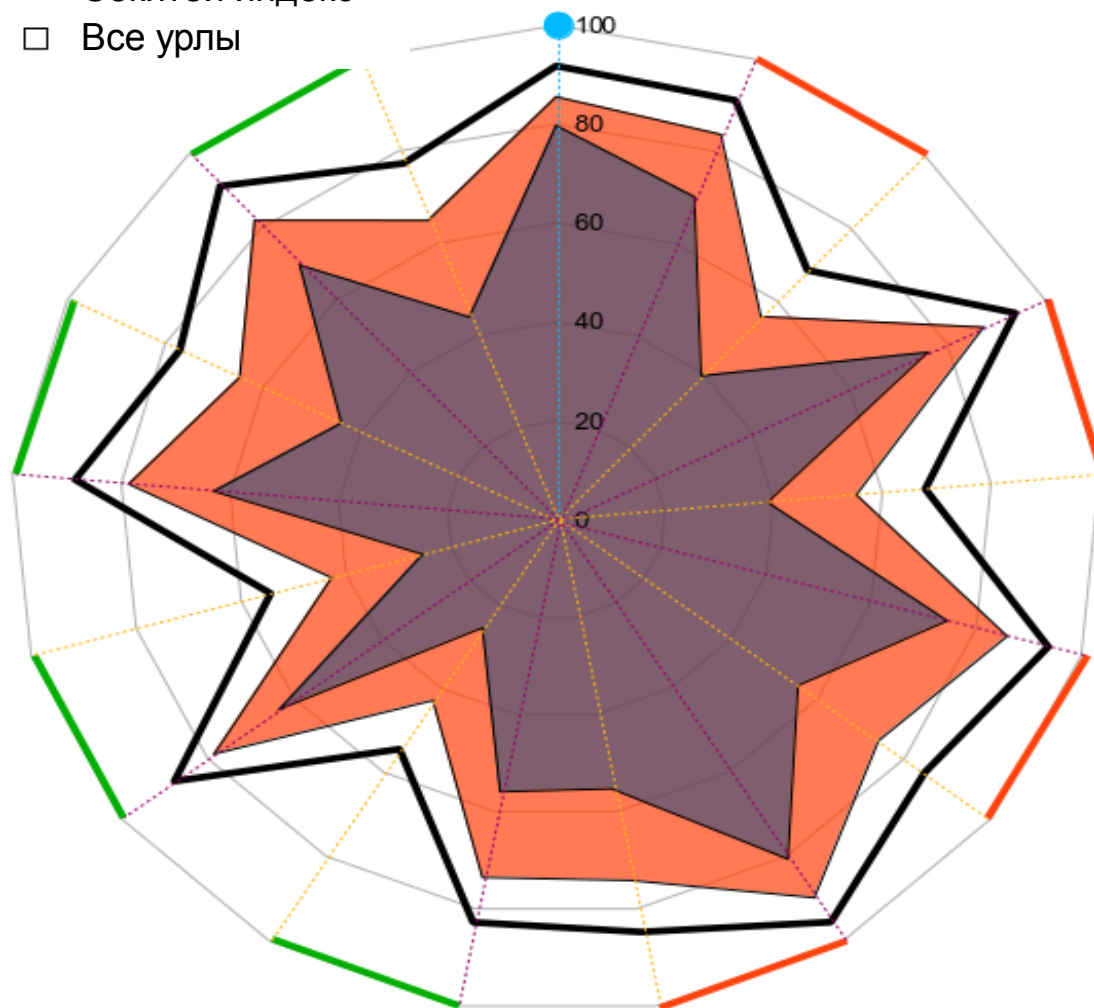
Качество индекса: новая квота

- Жадный индекс
- Жадный индекс с новой квотой (в два раза меньший размер)



Качество индекса: общее

- Жадный индекс
- Сектей индекс
- Все урлы



Оптимизация построения индекса

Цель

1. Оптимизировать скорость поиска (10^5 dpq)
2. Оптимизировать качество поиска

Нужно найти баланс между скоростью и качеством.

Вариант - сортировка индекса.

Статический ранк (Static rank)

Выделяем признаки:

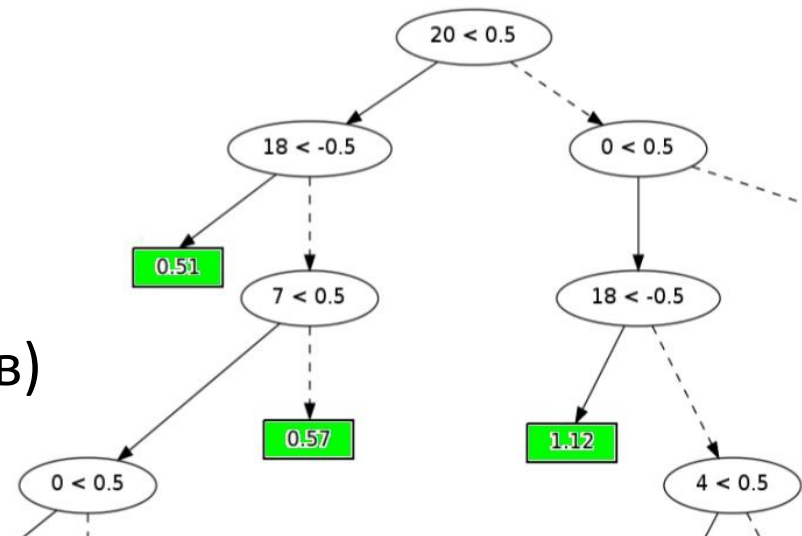
- 石庭 (sekitei)
- Ссылочные (Indegree, PR, etc.)
- Антиспам (например, #links с плохих сайтов)

Строим модель: gradient boosting decision trees

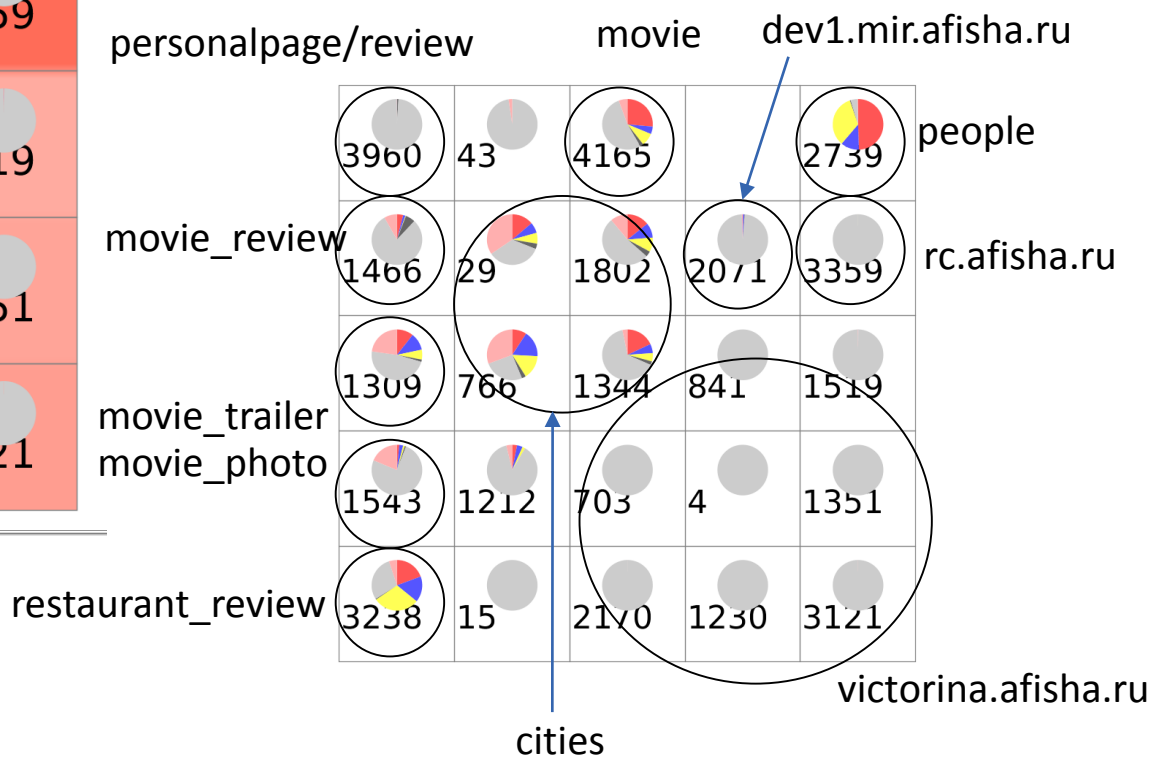
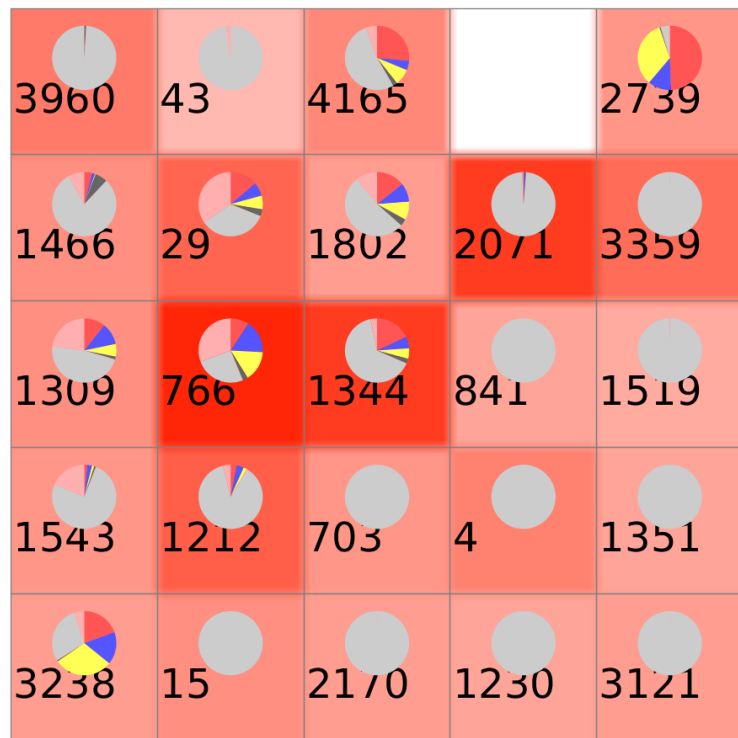
Цель: предсказать количество #qlink на странице

Два вида моделей:

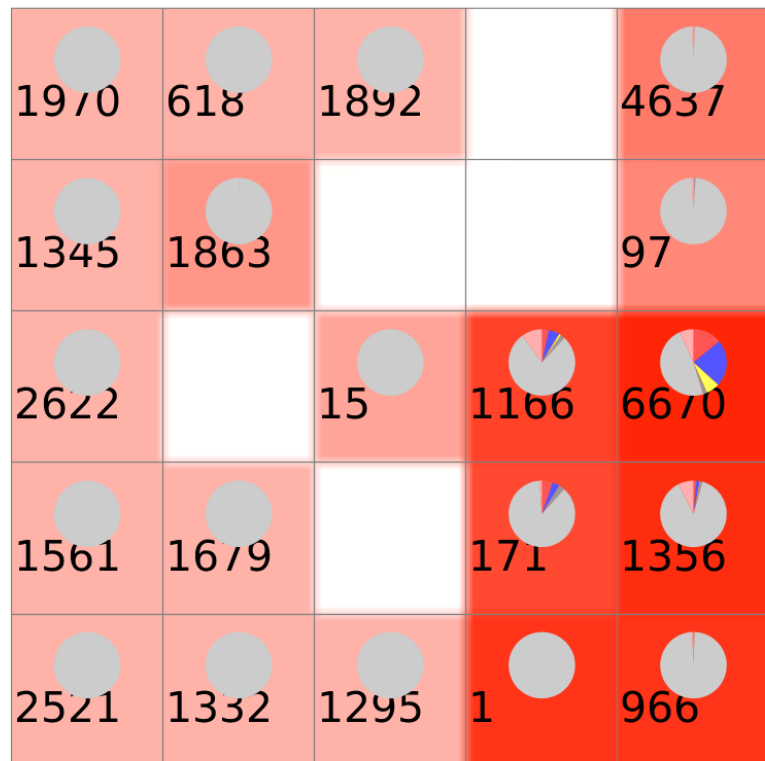
- персональная (для больших сайтов)
- общая для остальных



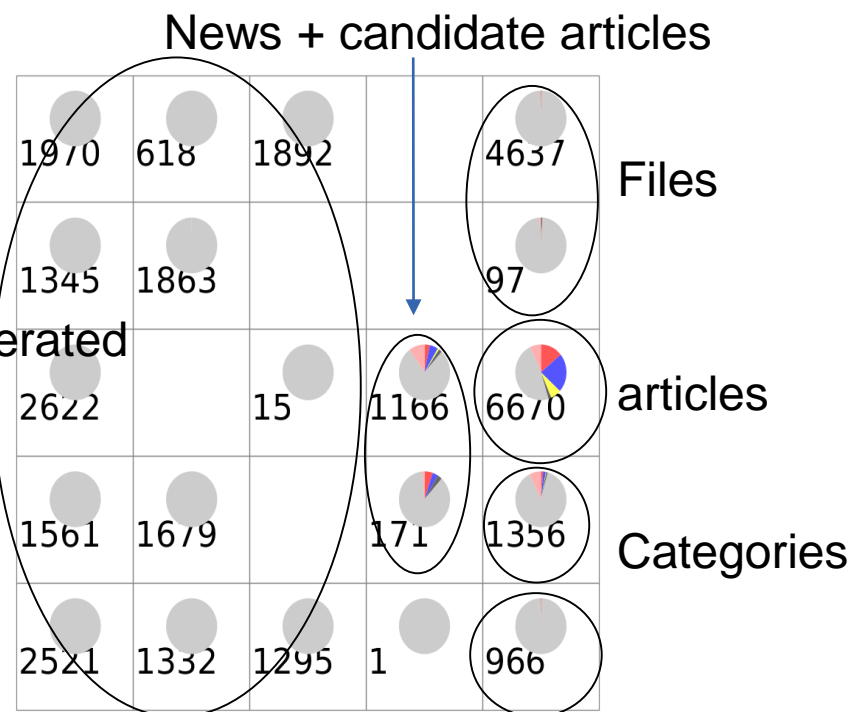
SOM карта для afisha.ru



SOM карта для absurdopedia.net

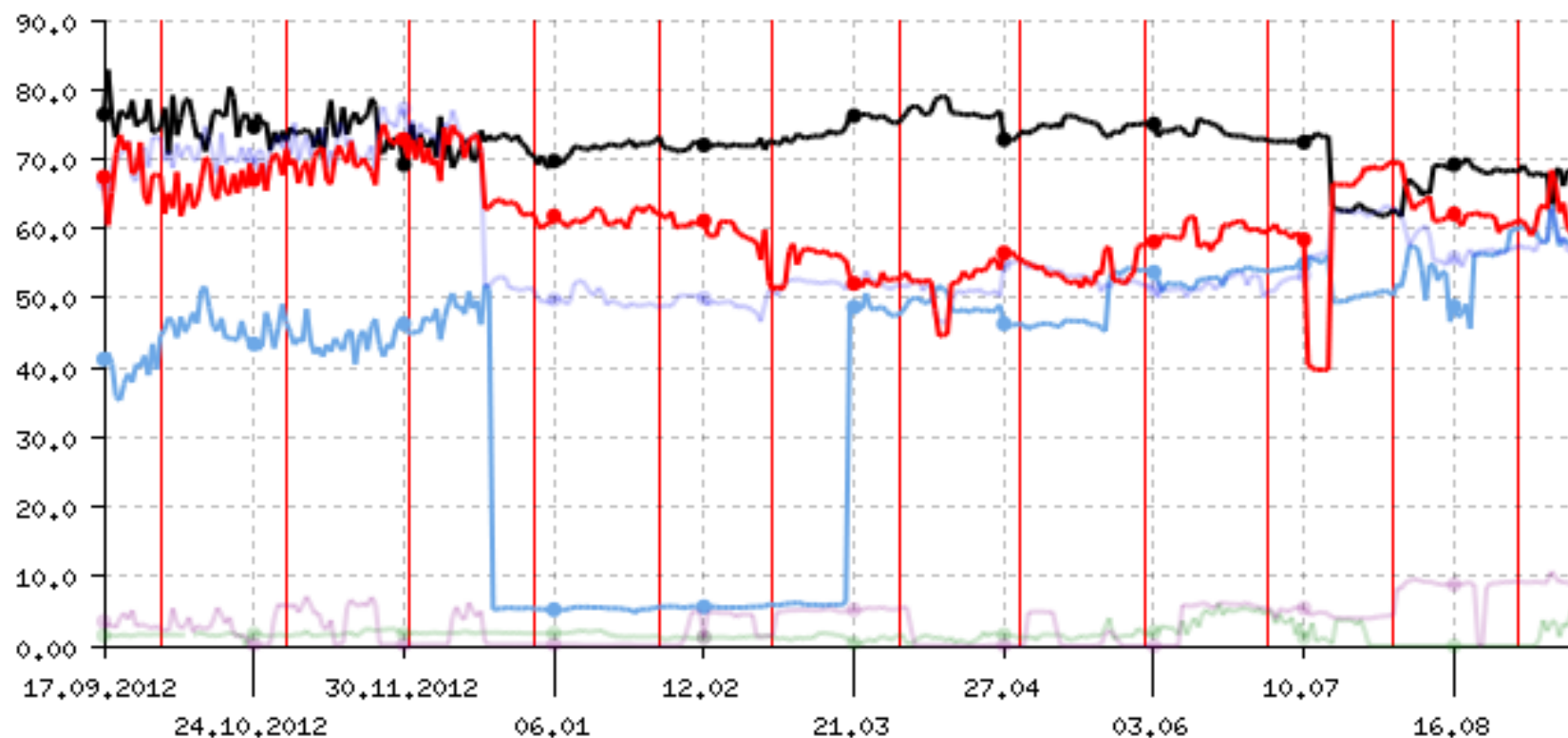


Auto-generated
trash



<http://analyzethis.ru/>

Полнота



Mail
Yandex
Google

石庭(sekitei). Алгоритм (Шаг 1)

1. Отбираем случайно N урлов.

- Сколько урлов отбирать?

$$p_{N,k}(\alpha) = \sum_i^k \binom{i}{N} \alpha^i (1 - \alpha)^{N-i}$$

$$P_{1000,10}(0.01) \approx 0.58$$

$$P_{1000,10}(0.02) \approx 0.01$$

$$P_{1000,10}(0.03) \approx 2 \times 10^{-5}$$

- Отбираем ~1000 урлов
- Состав урлов?
- Известные и неизвестные урлы в отношении 50/50.

石庭(sekitei). Алгоритм

2. Создаем признаки для каждого адреса:
 1. Количество сегментов в пути
 2. Список имен параметров запросной части (может быть пустым)
 3. Присутствие в запросной части пары `<parameters=value>`
 4. Сегмент пути на позиции :
 - a) Совпадает со значением `<строка>`
 - b) Состоит из цифр
 - c) Совпадает со значением `<строка с точностью до комбинации цифр>`: `<строка><цифры><строка>`
 - d) Имеет заданное расширение
 - e) Комбинация из двух последних вариантов

石庭(sekitei). Алгоритм (Шаг 2)

2. Создаем признаки для каждого адреса (пример):

<http://www.sports.ru/tags/1365242.html?p=57&type=photo>

| № | Признак | Тип признака |
|---|------------------------------------|--------------|
| 1 | 2 Сегмента | 1 |
| 2 | Запрос состоит из двух параметров | 2 |
| 3 | 0-й сегмент пути: tags | 4.a |
| 4 | 1-й сегмент пути: 1365242\.html | 4.a |
| 5 | 1-й сегмент пути; [0-9]+\. | 4.b |
| 6 | 1-й сегмент пути: [^\.]+\. | 4.c |
| 7 | В запросе есть параметр p=57 | 3 |
| 8 | В запросе есть параметр type=photo | 3 |

`/[^\.]+/[0-9]+\.`

`+p+type`

`~type=photo`

石庭(sekitei). Алгоритм (Шаг 3)

3. Отбираем признаки по частотности αN :
- Отбираем признаки для sport.ru

| № | N | Признак |
|---|-----|---------------------------------|
| 1 | 759 | Пустой запрос |
| 2 | 379 | В пути ровно два сегмента |
| 3 | 328 | 0-й сегмент: fantasy |
| 4 | 321 | 1-й сегмент пути: [^/]+\..html |
| 5 | 315 | 1-й сегмент пути: [0-9]+\..html |
| 6 | 266 | 1-й сегмент пути: football |
| 7 | 249 | В пути ровно 4 сегмента |

石庭(sekitei). Алгоритм (Шаг 4)

4. Кластеризация:

- Используем любой алгоритм, который позволяет нам найти кластера по выделенным признакам.
- Формируем регулярные выражения в формате PCRE для найденных кластеров.
- Из оставшихся урлов формируем остаток.

Домашнее задание

Домашнее задание

- Тут: <https://sphere.mail.ru/blog/topic/1356/>
- Всего 5 сайтов
- Каждый сайт это 20К обычных ссылок и 2К хороших ссылок
- Три открыты – обучающая выборка.
- Два скрыты - тестовая выборка.
- Для сайтов нужно сделать алгоритм “Сад камней” - выделение признаков.

Домашнее задание - требования

- Python 2.7
- Реализуется модуль `extract_features`, который экспортирует функцию `extract_features` с параметрами
 - Вход – файл с хорошими урлами
 - Вход – файл с обычными урлами
 - Файл, в который будут записаны результаты
- Шаблон прилагается в архиве
- Запуск проверки `python ./check-features.py`
- Смотрим результаты.



ТЕХНОСФЕРА

Вопросы

Ссылки:

- **Ricardo Baeza-Yates**. Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition), 2011