

## Практическая работа №1.

Тямгин Иван

### Латентно-семантический анализ.

Для обучения поисковой системы были использованы 15 тем по 50-75 документов. Всего 950 документов. Названия документов для каждой темы были получены с помощью поискового запроса в википедии вида:

<http://ru.wikipedia.org/w/index.php?title=Служебная:Поиск&limit=50&offset=0&profile=default&search=Таврия>

Затем скачаны по полученным ссылкам средствами языка python.

Были выбраны следующие темы:

- Тригонометрия
- Грипп
- Автомобиль
- Хлеб
- Москва
- Техника
- Таврия
- Биржа
- Депутат
- Газ
- Яндекс
- Водка
- Русский
- Видео
- Собака

Первым шагом были вырезаны из текстов все символы, кроме русских букв. Таким образом слово – это непрерывная последовательность русских букв. Каждое слово подверглось воздействию алгоритма Портера, чтобы одинаковые слова с разными окончаниями не считались как разные.

Была построена матрица 31238x950, которая означала частоту i-го слова в j-м документе. Матрица была нормализована с помощью TF-IDF, чтобы ключевые слова получили большой вес, а предлоги и союзы незначительный.

После чего матрица была разложена с помощью svd по 10-ми признакам. (разложение по 2-3 признакам дали существенно худший результат).

## Дендрограмма

Далее построена иерархическая кластеризация документов. Расстоянием между двумя документами являлся угол между 10-мерными векторами.

Так как дендрограмма очень большая, приведена её часть:



Также попадались довольно абсурдные объединения документов на первом уровне:



Дерево было построено с помощью стандартной функции **hclust** в *R*. И нарисовано с помощью библиотеки **d3js** в *javascript*.

## Поиск

Для каждого слова был построен чемпионский список – список документов и расстояние до этого документа. Чем меньше расстояние – тем он релевантнее для данного слова.

Поисковой запрос – неупорядоченный набор слов. Ответ на запрос – упорядоченный список документов по релевантности. Релевантность документа D - это

$$\sum_{\text{слово S из запроса}} \frac{\text{расстояние от S до D}}{1 + \text{количество слов из запроса, присутствующих в D}}$$

Примеры запросов:

{Москва, вокзал}

1. [https://ru.wikipedia.org/wiki/Казанский вокзал](https://ru.wikipedia.org/wiki/Казанский_вокзал) 91.13%
2. [https://ru.wikipedia.org/wiki/Ярославский вокзал](https://ru.wikipedia.org/wiki/Ярославский_вокзал) 90.95%
3. [https://ru.wikipedia.org/wiki/Ленинградский вокзал](https://ru.wikipedia.org/wiki/Ленинградский_вокзал) 90.84%
4. [https://ru.wikipedia.org/wiki/Белорусский вокзал](https://ru.wikipedia.org/wiki/Белорусский_вокзал) 90.17%
5. [https://ru.wikipedia.org/wiki/Киевский вокзал](https://ru.wikipedia.org/wiki/Киевский_вокзал) 90.02%
6. [https://ru.wikipedia.org/wiki/Курский вокзал](https://ru.wikipedia.org/wiki/Курский_вокзал) 89.61%
7. <https://ru.wikipedia.org/wiki/Москва-Сити> 89.37%
8. <https://ru.wikipedia.org/wiki/Москва-Каланчёвская> 89.19%
9. [https://ru.wikipedia.org/wiki/Москва \(гостиница в Москве\)](https://ru.wikipedia.org/wiki/Москва_(гостиница_в_Москве)) 88.91%
10. [https://ru.wikipedia.org/wiki/Памятник Кутузову \(Москва\)](https://ru.wikipedia.org/wiki/Памятник_Кутузову_(Москва)) 88.91%

{Яндекс}

1. [https://ru.wikipedia.org/wiki/Яндекс Нано](https://ru.wikipedia.org/wiki/Яндекс_Нано) 99.88%
2. <https://ru.wikipedia.org/wiki/Яндекс.Календарь> 99.88%
3. [https://ru.wikipedia.org/wiki/Элементы Яндекса](https://ru.wikipedia.org/wiki/Элементы_Яндекса) 99.87%
4. <https://ru.wikipedia.org/wiki/Яндекс.Услуги> 99.85%
5. <https://ru.wikipedia.org/wiki/Яндекс.Открытки> 99.85%
6. <https://ru.wikipedia.org/wiki/Яндекс.Shell> 99.84%
7. <https://ru.wikipedia.org/wiki/Yandex.SpeechKit> 99.84%
8. <https://ru.wikipedia.org/wiki/Яндекс.XML> 99.83%
9. <https://ru.wikipedia.org/wiki/Яндекс.Недвижимость> 99.83%
10. <https://ru.wikipedia.org/wiki/Яндекс.Навигатор> 99.83%

Более простая версия формулы:

$$\sum_{\text{слово S из запроса}} \text{расстояние от S до D}$$

давала на вид более худший результат.