

Индексация и модель мешка слов

Полнотекстовый поиск

- Найти документ, в котором есть слово, совпадающее с запросом
- Булева модель: совпадает/не совпадает
- Ранжированный поиск: качество совпадения
- Возникновение не связано с компьютером, но развитие стимулировалось дигитализацией текстов

Поисковые системы в исторической перспективе

- 1247 – Hugo de St. Caro – было задействовано 500 монахов для составления указателя ключевых слов к Библии
- И.Сегалович *«поменялась парадигма пользования системами»*

Алгоритмы поиска

- Прямой поиск

прямой поиск

Простейшая его версия знакома многим, и нет программиста, который бы не написал хотя бы раз в своей жизни подобный код:

```
char* strstr(char *big, char *little)
{
    char *x, *y, *z;
    for (x = big; *x; x++)
    {
        for (y = little, z = x; *y;
             ++y, ++z)
        {
            if (*y != *z)
                break;
        }
        if (!*y)
            return x;
    }
    return 0;
}
```

В этой функции языка C текст строки **big** просматривают слева направо и для каждой позиции **x** запускают последовательное сравнение с искомой подстрокой **little**. Для этого, двигая одновременно два указателя **y** и **z**, попарно сравнивают все символы. Если мы успешно дошли до конца искомой подстроки, значит она найдена.

В чем недостатки прямого поиска?

Алгоритмы поиска: прямой поиск

- Недостатки:

- низкая скорость

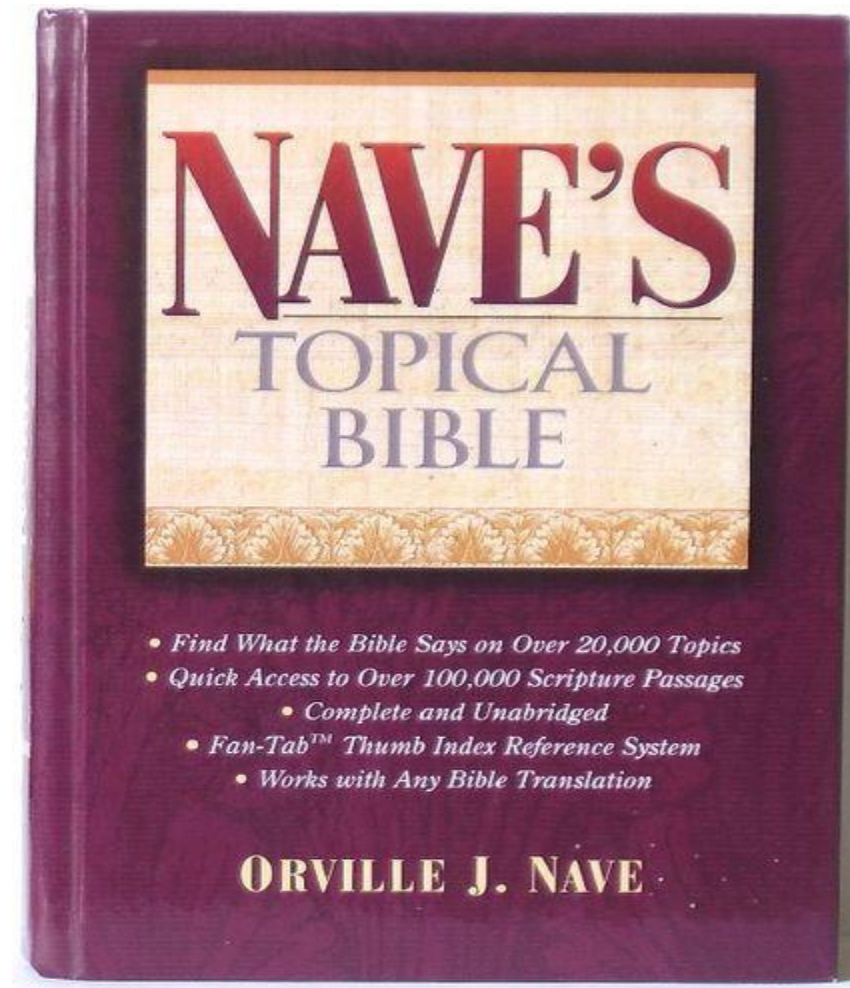
- Слишком сложно выполнять гибкие запросы, типа **отцы Near дети**, или **отцы NOT дети**

- Как найти наилучший ответ?

- Достоинства:

- неограниченные возможности по приближенному и нечеткому поиску, поиск происходит без упрощения терминов

Orville James Nave (1841-1917)



Download this
book as PDF

Information

Table of Contents

Page

Nave's Topical Bible

Search within book:

Go

Earn an
MBA

rooted
in the
Christian
faith
online

Nave's Topical Bible.

« FAMILIAR SPIRITS

FAMILY

FAMINE »

FAMILY

-OF SAINTS

.Blessed [Ps 128:3,6](#)

.Should be taught God's word [De 4:9,10](#)

.Worship God together [1Co 16:19](#)

.Be duly regulated [Pr 31:27](#); [1Ti 3:4,5,12](#)

.Live in unity [Ge 45:24](#); [Ps 133:1](#)

.Live in mutual forbearance [Ge 50:17-21](#); [Mt 18:21,22](#)

.Rejoice together before God [De 14:26](#)

.Deceivers and liars should be removed from [Ps 101:7](#)

.Warned against departing from God [De 29:18](#)

.Punishment of irreligious [Jer 10:25](#)

-GOOD, EXEMPLIFIED

.Abraham [Ge 18:19](#)

.Jacob [Ge 35:2](#)

Other Dictionaries

- **Nave's Topical Bible.**
- [The Catholic Encyclopedia, Volume 5: Diocese-Fathers of Mercy](#)

USER ACCOUNT

- [Login](#)
- [Register](#)

Ads by Google

Free Bible Courses

Email/postal World Bible School. Self-paced with study helpers.
www.wbschool.org

Free Bible Course


Bible study course and literature in English - by post or online
www.thisisyoubible.com




Symbols Of The Revelation

Learn the meaning of the symbols
in the

СЛОВАРЬ
ЯЗЫКА
ДОСТОЕВСКОГО

СЛОВАРЬ
ЯЗЫКА
ПУШКИНА

сегодняшняя "столь нечаянная и столь роковая встреча их **навек**и веков". (Бс 495)  Он выводил перед нами приобретателей, кулаков, обирателей и всяких заседателей. Ему стоило указать на них пальцем, и уже на лбу их зажигалось клеймо **навек**и веков, и мы уже наизусть знали: кто они и, главное, как называются. (Пб 18: 59)

Словоуказатель  **навек** ЗМ 111, 155 ПН 351, 398, 398 Ид 59 Бс 500 Пд 99, 336, 373, 414 БрК 99, 99, 279, 279, 279, 460, 504 БКа 188[19] **навек**и БЛ 93 Дв 207 Хз 302, 318, 319 БН 140 НН 179 ДС 396 СС 40, 40, 103, 105, 136, 147, 158 УО 196, 216, 223, 223, 223, 223, 297, 308, 322, 322, 396, 427, 429, 442 ЗМ 10, 125 ЗЗ 65 ЗП 105, 118, 120, 120, 161, 177 Кр 182 Иг 290 ПН 37, 61, 149, 239, 322, 326, 327, 397, 402, 406, 413 Ид 207, 285, 322, 328, 336, 349, 418, 461, 471 ВМ 106, 106 Бс 11, 27, 34, 71, 73, 198, 209, 273, 282, 365, 367, 367, 369, 377, 378, 388, 411, 480, 495, 506, 507 Пд 100, 114, 116, 130, 172, 278, 280, 378, 382, 384, 387 БрК 98, 108, 135, 147, 172, 175, 186, 187, 200, 212, 224, 232, 241, 255, 257, 270, 279, 328, 330, 346, 358, 387, 387, 394, 394, 464, 480, 506 БКа 68, 94, 173, 173, 175 Кт 22, 24, 34 СЧ 112, 118 [132] **навек**и-с БрК 187[1]  **навек** ДП 25: 30[1] **навек**и Пб 18: 59, 96 Пб 19: 67, 70, 78 Пб 21: 179, 187, 192, 193, 2, 203, 203 ДП 22: 8, 19, 43, 51, 51, 71, 108 ДП 23: 21, 120 ДП 24: 37, 43, 63 ДП 25: 32, 147, 147, 157, 158, 165 ДП 26: 7, 8, 15, 17, 20, 22, 74, 76, 83, 88, 142, 144, 151, 168 ДП 27: 36, 36[171]  **навек**и-с

Архитектура поисковой системы (очень грубо)

- Робот (краулер, спайдер, индексатор) обходит тексты и создает индекс
- Базы данных – хранят индекс
- Клиент (обработка запроса) – находит индекс

Откуда берется индекс?

Алгоритмы поиска: Булев поиск

- В каких баснях И.А. Крылова встречается *соловей, кукушка*, но не встречается *петух*?



Алгоритмы поиска: Булев поиск

соловей111010000
AND
кукушка:001010000
NOT
петух:010111111
=000010000

	<i>осел и соловей</i>	<i>квартет</i>	<i>кукушка и петух</i>	<i>лев и барс</i>	<i>кукушка и орел</i>	<i>ворона и лисица</i>	<i>лисица и осел</i>	<i>лев и лисица</i>	<i>слон в случае</i>
осел	1	1	0	1	0	0	1	0	1
петух	1	0	1	0	0	0	0	0	0
мартышка	0	1	0	1	0	0	0	0	0
кукушка	0	0	1	0	1	0	0	0	0
соловей	1	1	1	0	1	0	0	0	0
лисица	0	0	0	1	0	1	1	1	1

Алгоритмы поиска: Булев поиск

соловей	111010000
AND	
кукушка:	001010000
NOT	
петух:	010111111
=	000010000

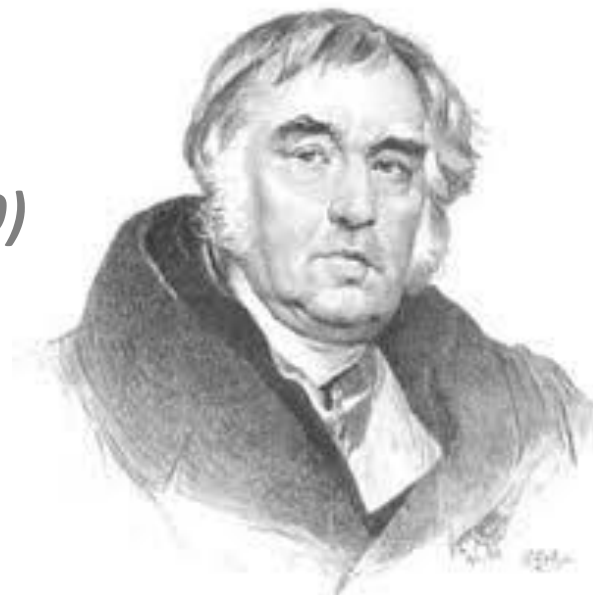
	осел и соловей	квартет	кукушка и петух	лев и барс	кукушка и орел	ворона и лисица	лисица и осел	лев и лисица	слон в случае
осел	1	1	0	1	0	0	1	0	1
петух	1	0	1	0	0	0	0	0	0
мартышка	0	1	0	1	0	0	0	0	0
кукушка	0	0	1	0	1	0	0	0	0
соловей	1	1	1	0	1	0	0	0	0
лисица	0	0	0	1	0	1	1	1	1

Алгоритмы поиска: Булев поиск

- В каких баснях И.А. Крылова встречается *соловей, кукушка*, но не встречается *петух*?

И. А. Крылов. Кукушка и орел (1829)

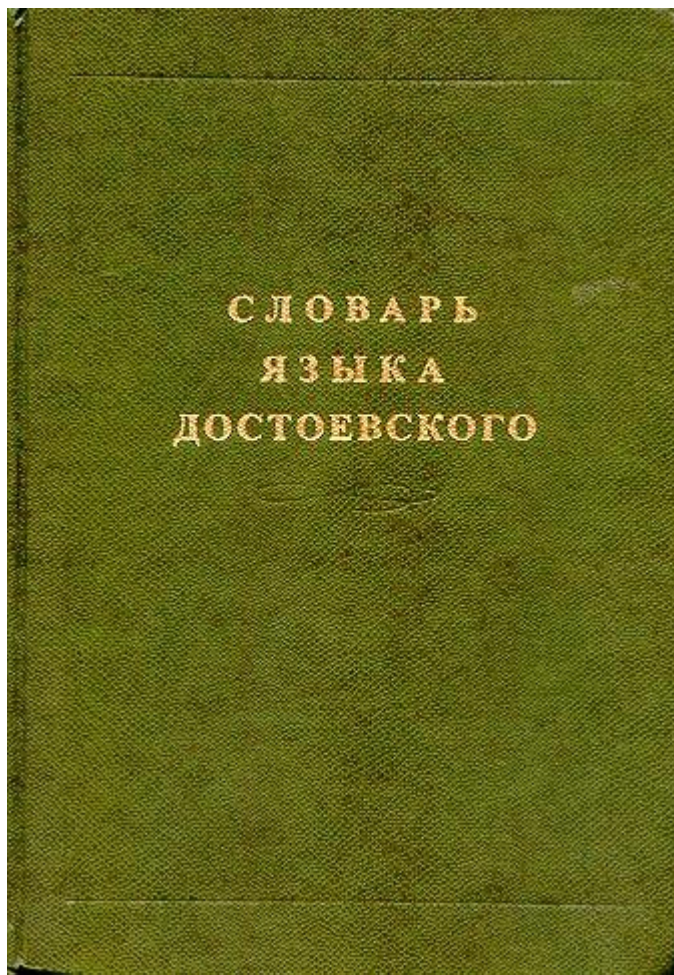
Орел пожаловал **Кукушку** в **Соловьи**
Кукушка, в новом чине,
Усевшись важно на осине,
Таланты в музыке свои
Выказывать пустилась;
Глядит — все прочь летят,
Одни смеются ей, а те ее бранят.



Но что если у нас большие коллекции?

- Как построить матрицу на 1 миллион документов, в каждом из которых примерно 1000 слов?
- Инвертированный индекс!

Инвертированный индекс



На самом деле – хорошо забытое старое

1. **НАВЕК, НАВЕКИ** <214:152,48,13,1>

ДОСТ;253518678

Навсегда, на всю жизнь.

☞ [Девушкин] Этим они [его превосходительство] меня самого себе возвратили. Этим поступком они

мой дух воскресили, ж
сделали, и я твердо ув
перед всевышним, но
благополучии его пре
престола его!.. (БЛ 93)
[Ордынгов] почти чувс
встретить ее как светл
впечатления, таким м
пробуждении вновь е
обдало душу его, что ж
напряженною деятель
перерваться, разруши
угаснуть **навек**. (Хз 2
через родителя моего

Словосказатель ☞ **навек** ЗМ 111, 155 ПН 351, 398, 398 Ид 59 Бс 500 Пд 99, 336, 373, 414 БрК 99, 99, 279, 279, 279, 460, 504 БКа 188[19] **навеки** БЛ 93 Дв 207 Хз 302, 318, 319 БН 140 НН 179 ДС 396 СС 40, 40, 103, 105, 136, 147, 158 УО 196, 216, 223, 223, 223, 223, 297, 308, 322, 322, 396, 427, 429, 442 ЗМ 10, 125 ЗЗ 65 ЗП 105, 118, 120, 120, 161, 177 Кр 182 Иг 290 ПН 37, 61, 149, 239, 322, 326, 327, 397, 402, 406, 413 Ид 207, 285, 322, 328, 336, 349, 418, 461, 471 ВМ 106, 106 Бс 11, 27, 34, 71, 73, 198, 209, 273, 282, 365, 367, 367, 369, 377, 378, 388, 411, 480, 495, 506, 507 Пд 100, 114, 116, 130, 172, 278, 280, 378, 382, 384, 387 БрК 98, 108, 135, 147, 172, 175, 186, 187, 200, 212, 224, 232, 241, 255, 257, 270, 279, 328, 330, 346, 358, 387, 387, 394, 394, 464, 480, 506 БКа 68, 94, 173, 173, 175 Кт 22, 24, 34 СЧ 112, 118 [132] **навек**-с БрК 187[1] ☞ **навек** ДП 25: 30[1] **навек** Пб 18: 59, 96 Пб 19: 67, 70, 78 Пб 21: 179, 187, 192, 193, 2, 203, 203 ДП 22: 8, 19, 43, 51, 51, 71,

Индексирование

- Номеруем все документы, и каждому слову приписываем id документов, в которых оно встречается
- соловей 1 2 3 5 18 33 47 83
- кукушка 3 5 14 25 103
- петух 1 3 57

Индексирование

Документ 1

Орел пожаловал кукушку в соловьи,
Кукушка, в новом чине,
Усевшись важно на осине,

Документ 2

За что же не боясь греха кукушка
хвалит петуха

Орел	1
пожаловал	1
кукушку	1
в	1
соловьи	1
Кукушка	1
в	1
новом	1
чине	1
усевшись	1
важно	1
на	1
осине	1

За	2
что	2
же	2
не	2
боясь	2
греха	2
кукушка	2
хвалит	2
петуха	2

Индексирование: объединяем таблицы

Документ 1

Орел пожаловал кукушку в соловьи

Документ 2

За что же не боясь греха кукушка
хвалит петуха

Орел	1
пожаловал	1
кукушку	1
в	1
соловьи	1
За	2
что	2
же	2
не	2
боясь	2
греха	2
кукушка	2
хвалит	2
петуха	2

Индексирование: нормализуем и сортируем

Документ 1

Орел пожаловал кукушку в соловьи

Документ 2

За что же не боясь греха кукушка хвалит петуха

term	term freq	doc id
бояться	1	2
в	1	1
грех	1	2
же	1	2
за	1	2
кукушка	2	1 -> 2
не	1	2
орел	1	1
петух	1	2
пожаловать	1	1
соловей	1	1
хвалит	1	2
что	1	2

Базовые ступени текстового процессинга (препроцессинга)

- Документы



- Токенизация

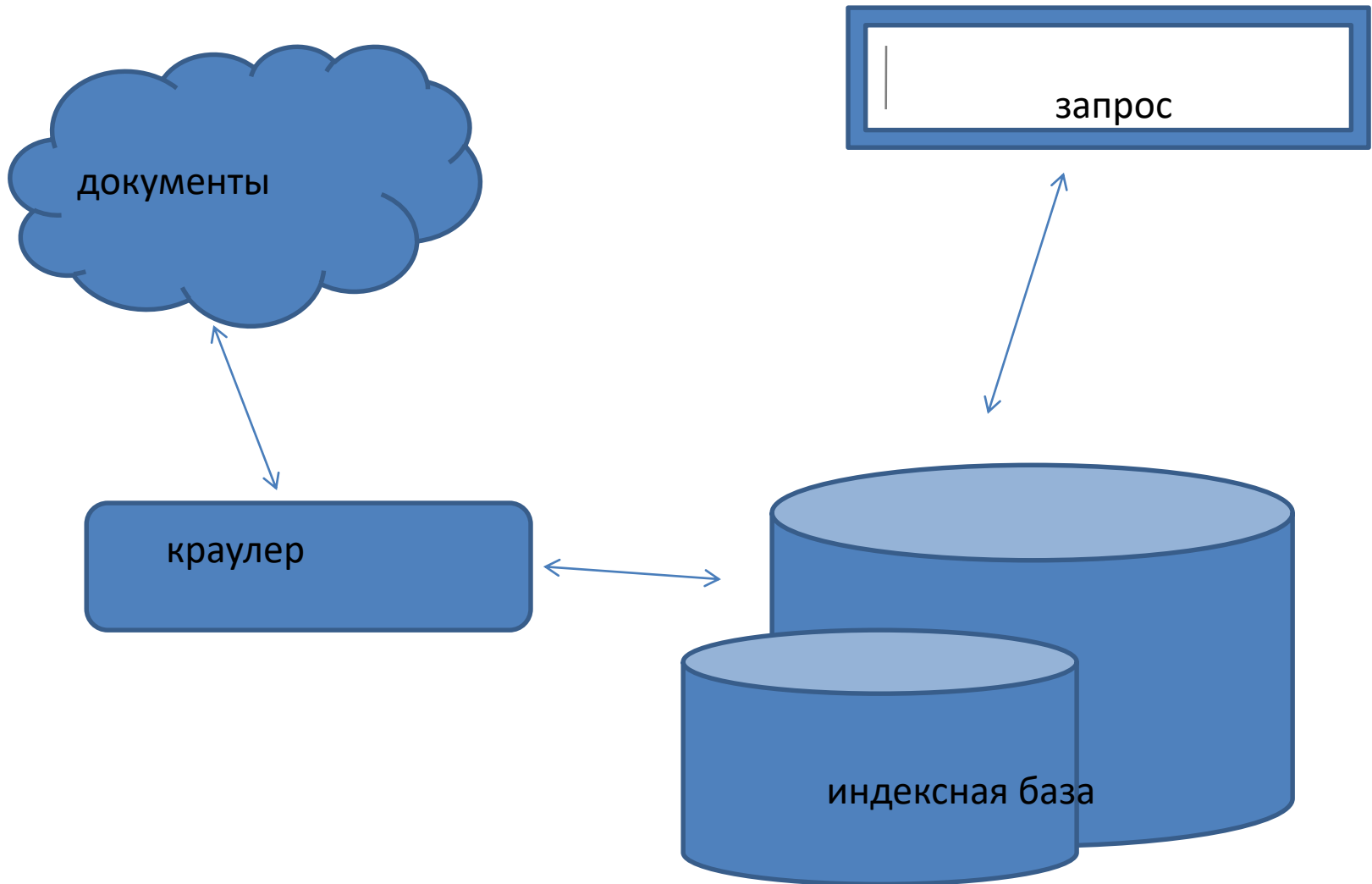


- Лемматизация



- Индексатор

Архитектура поисковой системы



Сложные запросы: кукушка and соловей

- 1) найди в словаре *соловей*, выпиши номера его вхождений
- 2) найди в словаре *кукушка*, выпиши номера его вхождений
- 3) пересеки два набора номеров документов

- ***соловей***

1 2 3 5 18 33 47 83

- ***кукушка***

3 5 14 25 103

Сложные запросы: пересечение



1 2 3 5 18 33 47 83

соловей

3 5 14 25 103

кукушка

- нужно идти одновременно по двум рядам, сравнивая их друг с другом
- Важно! номера документов должны быть отсортированы

Обработка запроса

1. Двигаемся одновременно по двум рядам пойнтеров.
2. На каждом шаге сравниваем оба пойнтера.
3. Если они равны – то это искомое пересечение.
4. Если они не равны, то двигаем меньший.

Обработка запроса

```
INTERSECT(p1, p2)  
1      answer ← [ ]  
2      while p1 ≠ NIL and p2 ≠ NIL  
3      do if docID(p1) = docID(p2)  
4      then ADD(answer, docID(p1))  
5          p1 ← next(p1)  
6          p2 ← next(p2)  
7      else if docID(p1) < docID(p2)  
8          then p1 ← next(p1)  
9          else p2 ← next(p2)  
10     return answer
```

Time: $O(x+y)$;

x: number of entries of the first posting list

y: number of entries of the second posting list

$O(\text{number_of_documents})$:

Булева модель

- Основной инструмент поиска трех десятилетий
- Очень точный: документ либо попадает, либо нет
- До сих пор многие системы используют Булев поиск (поиск файлов, библиотечный каталог, поиск в почте)
- You know exactly what you are getting

Векторная модель

- Понятие *релевантности* (документ интересный пользователю)
- Понятие *ранжирования* (упорядочивание документов от наиболее релевантных, к наименее релевантным)
- Модель мешка слов: вероятность встретить слово в тексте никак не зависит от встречаемости других слов

Признаки:

координаты в пространстве

Близость (подобие):

близость в пространстве

Поисковый образ:

вектор в пространстве признаков

Законы Ципфа универсальны. В принципе, они применимы не только к текстам. В аналогичную форму выливается, например, зависимость количества городов от числа проживающих в них жителей. Характеристики популярности узлов в сети Интернет -- тоже отвечают законам Ципфа. Не исключено, что в законах отражается "человеческое" происхождение объекта. Так, например, ученые давно бьются над расшифровкой манускриптов Войнича. Никто не знает, на каком языке написаны тексты и тексты ли это вообще. Однако исследование манускриптов на соответствие законам Ципфа доказало: это созданные человеком тексты. Графики для манускриптов Войнича точно повторили графики для текстов на известных языках.

5	В
3	Ципфа
3	НЕ
3	ТЕКСТЫ
3	НА
3	МАНУСКРИПТОВ
2	ЗАКОНАМ
2	НАПРИМЕР
2	ЭТО
2	ВОЙНИЧА
2	ДЛЯ
2	ГРАФИКИ
1	БЬЮТСЯ
1	ЧЕЛОВЕЧЕСКОЕ
1	К
1	ОТ
1	ПРОЖИВАЮЩИХ
1	ТЕКСТОВ

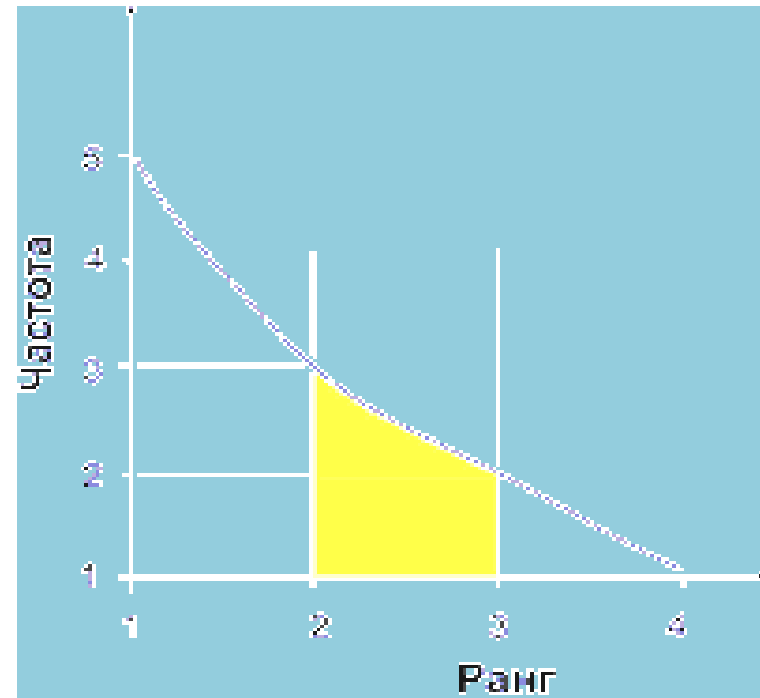
1	ТОЖЕ
1	ФОРМУ
1	ИНТЕРНЕТ
1	ЧЕЛОВЕКОМ
1	ЯЗЫКЕ
1	ЗАКОНАХ
1	КОЛИЧЕСТВА
1	НИКТО
1	НАПИСАНЫ
1	ОТВЕЧАЮТ
1	ПРОИСХОЖДЕНИЕ
1	ТАК
1	ХАРАКТЕРИСТИКИ
1	ИСКЛЮЧЕНО
1	АНАЛОГИЧНУЮ
1	ЧТО
1	ЯЗЫКАХ
1	КАКОМ
1	НИХ

1	ИССЛЕДОВАНИЕ
1	СЕТИ
1	ВООБЩЕ
1	ЧИСЛА
1	ЗАВИСИМОСТЬ
1	ДОКАЗАЛО
1	ЛИ
1	ОДНАКО
1	ПОВТОРИЛИ
1	ПРИНЦИПЕ
1	ТОЧНО
1	УЗЛОВ
1	И
1	ГОРОДОВ
1	СОЗДАННЫЕ
1	ВЫЛИВАЕТСЯ
1	ЗНАЕТ
1	ЗАКОНЫ
1	ДАВНО

Векторная модель

Смысл абзаца очень точно выражают слова: *цифра*, *манускриптов*, *войнича*, *законам*. Запрос типа: + "закон* *цифра*" + "манускрипт* *войнича*" непременно найдет нам этот документ.

Однако в область попали и слова: *на*, *не*, *для*, например, это. Эти слова являются "шумом", помехой, которая затрудняет правильный выбор.



Векторная модель

$$Tf = 3$$

3	ЦИПФА
3	НЕ
3	ТЕКСТЫ

Как различить *не*, *тексты* и *ципфа*?

Векторная модель

— idf:

*Инверсная документная частота термина $i = \log$
(количество документов в базе данных / количество
документов с термином i).*

- Каждому термину можно присвоить весовой коэффициент, отражающий его значимость:

Вес термина i в документе $j = \text{частота термина } i \text{ в документе } j \times \text{инверсная документная частота термина } i$.

Векторная модель

- tf_{ik} – частота термина T_k в документе D_i
- idf_k – обратная документальная частота для термина T_k в коллекции S
- N – общее число документов в коллекции
- N_k – количество документов в коллекции S , содержащих термин T_k


$$w_{ik} = tf_{ik} \cdot idf_k$$
$$idf_k = \log \frac{N}{N_k}$$

Векторная модель

- Бинарные веса:

$W_{ij}=1$ если документ d_i содержит термин t_j , иначе 0.

- Частота термина tf_{ij} , т.е. сколько раз встретился термин t_j в документе d_i (или относительная частота: сколько раз термин встретился в документе / количество токенов в документе)
- *tf x idf*:
 - чем выше частота термина в документе — тем выше его вес, но
 - термин должен не часто встречаться во всей коллекции документов

Векторная модель

- вместо tf используют wf

—

$$wf_{t,d} = \begin{cases} 1 + \log tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}.$$

$$wf-idf_{t,d} = wf_{t,d} \times idf_t.$$

Векторная модель

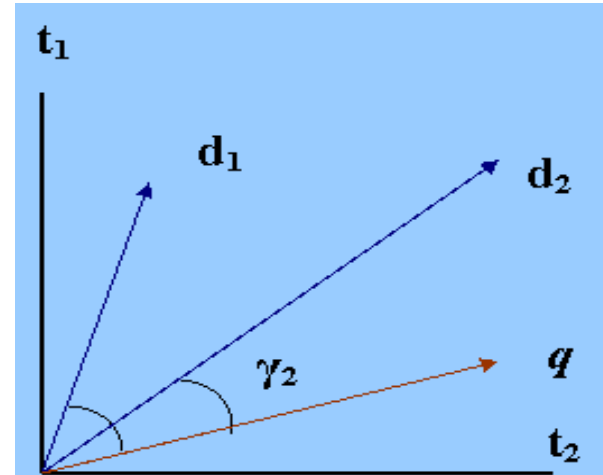
- учитывает частотные характеристики слов
- операция сравнения документов уподобляется отношению расстояния между векторами
- Ранжирование:
 - чем больше локальная частота термина в документе (**TF**) и больше «редкость» (т.е. обратная встречаемость в документах) термина в коллекции (**IDF**), тем выше вес данного документа по отношению к термину

Подробнее можно прочитать:

Ch6. Scoring, term weighting & the vector space model. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze *An Introduction to Information Retrieval*. Cambridge University Press. — 2009. — 544 pp. ([Draft. Online edition](#)) (или русский перевод)

Векторная модель

- Релевантность выражается через подобие векторов
- Для вычисления подобия векторов используется косинусная метрика



$$S(q, d) = \frac{q \cdot d}{|q| \cdot |d|} = \frac{|q| \cdot |d| \cdot \cos \gamma}{|q| \cdot |d|}$$

Скалярное произведение векторов

$$\cos(q, d) = \frac{\sum_1^n (q_i \cdot d_i)}{\sqrt{\sum_1^n q_i^2} \cdot \sqrt{\sum_1^n d_i^2}}$$

Скалярное произведение векторов

где \sum — сумма по всем i , i — i -ая координата в n -мерном векторе,
координат столько — сколько несовпадающих слов в корпусе

w_{qi} — значение i -ой координаты вектора q

w_{di} — значение i -ой координаты вектора d

Как вычислить косинус угла

Скалярное произведение двух векторов – произведение их длин на косинус угла между ними. $a \cdot b = |a| \cdot |b| \cos(\angle a b)$

Например, для двумерных векторов:

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cos(\angle \vec{a} \vec{b})$$

Если векторы \vec{a} и \vec{b} заданы своими координатами: $\vec{a}=(a_1;a_2;a_3)$,
 $\vec{b}=(b_1;b_2;b_3)$,

то их скалярное произведение вычисляется по формуле:

$$(\vec{a}, \vec{b})=a_1b_1+a_2b_2+a_3b_3$$

Например,

$$\vec{a}=(2;5;3),$$

$$\vec{b}=(3;4;1)$$

$$(\vec{a}, \vec{b})=a_1b_1+a_2b_2+a_3b_3 = 2 \cdot 3 + 5 \cdot 4 + 3 \cdot 1 = 29$$

Длина вектора

Скалярное произведение двух векторов – произведение их длин на косинус угла между ними. $a \cdot b = |a| \cdot |b| \cos(\angle a, b)$

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cos(\angle \vec{a}, \vec{b})$$

Длина вектора

Пусть $\vec{a}=(a_1; a_2; a_3)$, длина вектора - $|\vec{a}|$

$$|\vec{a}| = \sqrt{\sum_{i=1}^n a_i^2}$$

Пусть $\vec{a}=(5; \sqrt{3}; 6)$

$$|\vec{a}| = \sqrt{5^2 + (\sqrt{3})^2 + 6^2} = \sqrt{25 + 3 + 36} = \sqrt{64} = 8$$

Векторная модель

- Релевантность выражается через подобие векторов
- Для вычисления подобия векторов используется косинусная метрика

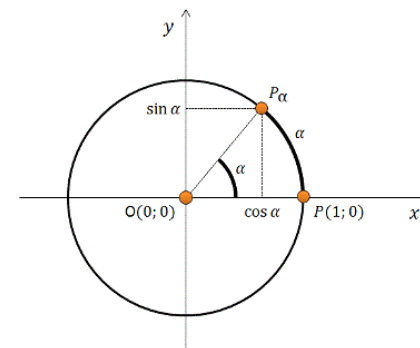
Пусть

Вектора:

текст 1 - “ворон, ворон, ворон, летит” $T = (3,1,0)$

запрос “ворон, летит” $Q = (1,1,0)$

текст 2 “воробей, летит” $F = (0,1,1)$



$$\cos(T, Q) = \frac{\sum_1^n (t_i \cdot q_i)}{\sqrt{\sum_1^n t_i^2} \cdot \sqrt{\sum_1^n q_i^2}} = \frac{3 \cdot 1 + 1 \cdot 1 + 0 \cdot 0}{\sqrt{9+1} \cdot \sqrt{1+1}} = \frac{4}{3,16 \cdot 1,4} = 0,9$$

!!!NB косинус нуля = 1; чем больше значение косинуса, тем меньше угол между векторами

$$\cos(F, Q) = \frac{1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1}{\sqrt{1+1} \cdot \sqrt{1+1}} = \frac{1}{2} = 0,5;$$

$$\cos(T, F) = \frac{3 \cdot 0 + 1 \cdot 1 + 0 \cdot 1}{\sqrt{3+1} \cdot \sqrt{1+1}} = 0,22$$

Оценка качества поиска

- **Релевантность**

- Полнота (recall) R
- Точность (precision) P

документы	выданные	невыданные
релевантные	a	c
нерелевантные	b	d

Точность $P = a/a+c$

Полнота $R = a/a+b$

F мера = $2pr/(p+r)$ - *гармоническое среднее*

Оценка качества системы

Категория i		Экспертная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	TP	FP
	Отрицательная	FN	TN

- TP — истинно-положительное решение;
- TN — истинно-отрицательное решение;
- FP — ложно-положительное решение;
- FN — ложно-отрицательное решение.

Тогда, точность и полнота определяются следующим образом:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Точность или полнота

- F-мера – нечто среднее
- Accuracy – доля правильных ответов

$$F_1 = \frac{2PR}{P + R}$$

