Классический информационный поиск: реализация и методы

Подготовила студентка 517 группы
Платонова Елена

Сегодня в теме:

- Задачи информационного поиска (ИП)
- Булев поиск
- Инвертированный список
- Схематизация документа
- Словари и нечеткий поиск
- Модель векторного пространства (ВП), tf-idf
- Эффективное ранжирование
- Оценка информационного поиска
- Сниппеты
- Обратная связь по релевантности
- Переформулирование запроса

Задачи ИП

- Задачи ИП
  - Постановка
  - Цели
  - Критерии
- Инвертированный список
- Булев поиск
- Схематизация документа
- Словари и нечеткий поиск
- Модель ВП, tf-idf
- Эффективное ранжирование
- Оценка ИП
- Сниппеты
- Обратная связь по релевантности
- Переформулирование запроса

Информационный поиск (англ. *Information retrieval*) — процесс поиска в большой коллекции (хранящейся, как правило, в памяти компьютеров) некоего неструктурированного материала (обычно - документа), удовлетворяющего информационные потребности.

Запрос — это формализованный способ выражения информационных потребностей. Для выражения информационной потребности используется язык поисковых запросов, синтаксис варьируется от системы к системе.

Объект запроса — это информационная сущность, которая хранится в базе автоматизированной системы поиска. Чаще всего это текстовый документ, но не существует никаких принципиальных ограничений: возможен поиск любой мультимедиаинформации.

Процесс занесения объектов поиска в ИПС называется **индексацией.**

Задачи ИП

Центральная задача ИП — удовлетворение информационной потребности пользователя, сформулированные в запросе.

Классическая задача ИП, с которой началось развитие этой области, — это поиск документов, удовлетворяющих запросу, в рамках некоторой *статической коллекции документов*.

Список задач ИП постоянно расширяется и теперь включает:

a) Классификацию документов;
b) Фильтрацию документов;
c) Кластеризацию документов;
d) Проектирование архитектур поисковых систем и пользовательских интерфейсов
e) Извлечение информации, в частности аннотирования и реферирования документов;
f) Языки запросов и др.

Также, перед движками ИП ставятся некоторые задачи по обработке естественных языков, что включает в себя *морфологический анализ, разрешение лексической многозначности* и так далее.

Задачи ИП

Пример: поиск документов, содержащий определенный набор слов. МГУ AND ВМК – смотрим коллекцию от начала до конца, отмечаем где содержатся слова запроса, а где нет (*прямой поиск*).

	Vmk-online	Vkontakte	Cmcmsu	Forum
МГУ	1	1	0	1
Студенты	0	1	1	0
ВМК	1	1	0	0
Лекции	0	0	1	0

Матрица «термин-документ»

1 Обработка большой коллекции документов – объемы данных растут не менее быстро, чем скорость работы компьютеров.
2. Гибкость при сравнении – Студенты NEAR ВМК, где NEAR может означать «не далее 5 слов»
3. Выполнение ранжированного поиска – нужен не просто ответ, а наилучший ответ на запрос относительно документов, содержащих определенное слово.

Булев поиск

- ✓ Задачи ИП
- ❑ Булев поиск
  - ❑ Недостатки
- ❑ Инвертированный список
- ❑ Схематизация документа
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Модель булева поиска – модель ИП, в ходе которого можно обрабатывать любой запрос, имеющий вид булева выражения (где есть AND, NOT, OR и тд)

	Vmk-online	Vkontakte	Cmcmsu	Forum
МГУ	1	1	0	1
Студенты	0	1	1	0
ВМК	1	1	0	0
Лекции	0	0	1	0

Пусть мы ищем сайты, где встречается **ВМК AND МГУ NOT Студенты.**
1100 & 1101 & 0111 =0100
Ответ на запрос Vkontakte.

Но:

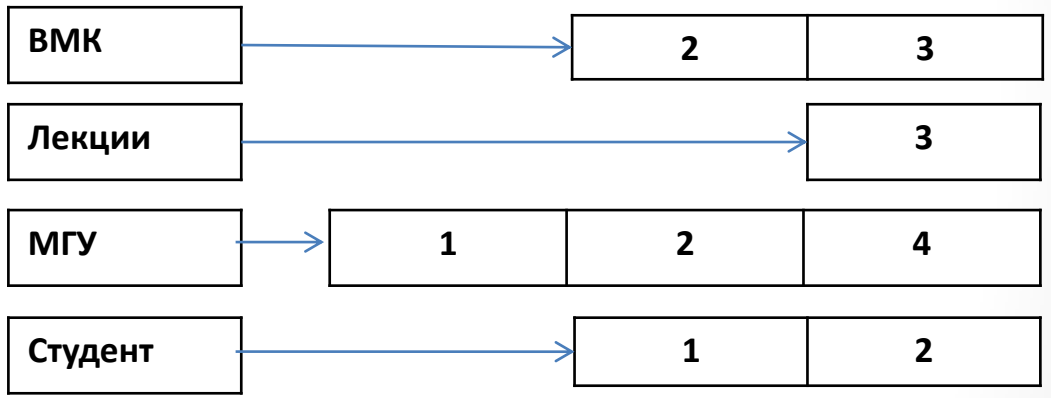
- Если информационная потребность более сложная?
- Если много разных терминов в документах, много документов, то размер матрицы огромен, но она сильно разрежена при этом?
- Как учесть повторяемость терминов?

Инвертированный список

- ✓ Задачи ИП
- ✓ Булев поиск
- ❑ **Инвертированный список**
  - ❑ Понятие
  - ❑ Построение
  - ❑ Пример
- ❑ Схематизация документа
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Инвертированный список – первая важная концепция ИП.

1) Записываем словарь (dictionary) терминов
2) Для каждого термина указываем документы, его содержащие
3) Соответствующий список – список словопозиций или ИС.

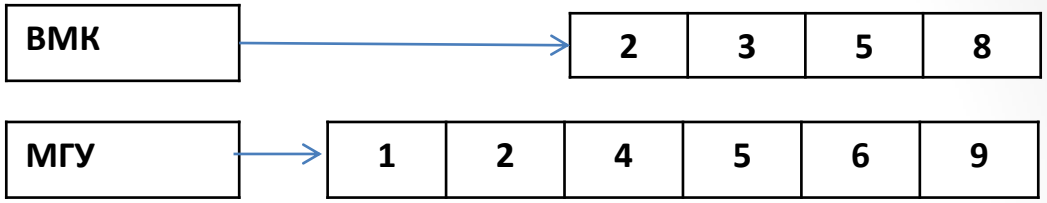


Важно:

- Термины в списке расположены в алфавитном порядке
- Не учитывается количество повторов слова
- Идентификаторы документов в итоговом списке отсортированы

Инвертированный список

- ✓ Задачи ИП
- ✓ Булев поиск
- ❑ **Инвертированный список**
  - ✓ Понятие
  - ✓ Построение
  - ❑ Пример
- ❑ Схематизация документа
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса



Рассмотрим обработку конъюнктивного запроса ***МГУ AND ВМК.***

1) Обнаруживаем термин МГУ в словаре
2) Находим список его словопозиций
3) Обнаруживаем термин ВМК в словаре
4) Находим список его словопозиций
5) Находим пересечение этих списков

```
INTERSECT (p1,p2)
        answer ←()
        while p1≠ NIL and p2 ≠ NIL
        do If docID(p1)=docID(p2)
                then Add(answer, docID(p1))
                        p1 ← next(p1)
                        p2 ← next(p2)
        else if docID(p1)<docID(p2)
                then p1 ← next(p1)
                then p2 ← next(p2)
        answer
```

Инвертированный список

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
  - ✓ Понятие
  - ✓ Построение
  - ✓ Пример
- ❑ Схематизация документа
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Плюсы булевых запросов:

- ✓ Точность (документ либо удовлетворяет запросу, либо нет)
- ✓ Прозрачность результатов (в отличие от вероятностной модели)
- ✓ Возможно эффективное ранжирование (законы по дате выхода)
- ✓ Выше точность – ниже полнота для AND, наоборот для OR

Что не учитывается:

- Лучше определить набор терминов в словаре и повысить устойчивость к опечаткам и нечеткому выбору слов для формулировках.
- Запросы, учитывающие близость терминов (NEAR)
- Взвешивать документы на основе частоты терминов в документе
- Вместо «мешка» релевантных документов отранжированные результаты поиска – определение степени соответствия запросу

Коммерческая служба булева поиска **Westlaw** – крупнейшая юридическая поисковая служба (ок. 500 т. подписчиков, млн поисковых запросов). В 2005г. булев поиск оставался основным механизмом поиска и использовался большим количеством пользователей, хотя с 1992г. Уже были реализованы свободные текстовые запросы.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ❑ Выделение последовательности
  - ❑ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Этапы построения инвертированного индекса:

1) Собираем документы, подлежащие индексированию
2) Разбиваем текст на лексемы
3) Выполняем предварительную обработку лексем
4) Индексируем документы по каждому термину

Выделение лексем – процесс разделения потока символов на лексемы. В процессе предварительной обработки лексем возникают классы эквивалентных лексем образующие **множество терминов**, по которым идет индексирование.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ❑ Выделение последовательности
  - ❑ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Входные документы – последовательность байтов.

Возможные кодировки:

a) *ASCII*
b) *UNICODE UTF-8*
c) *Национальный стандарт*
d) *Стандарт поставщика документа*
e) *Двоичное представление (Word, zip)*
f) *XML-документы*
g) …

Например, «&amp» означает «&»

Даже линейный порядок символов не всегда верен:

Для решения задачи применяются:

- ✓ Классификация на основе машинного обучения
- ✓ Эвристические методы
- ✓ Выбор пользователя
- ✓ На основе метаданных о документе

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ❑ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Далее мы выбираем **структурную единицу документа** (document unit) для индексирования:

В системе Unix последовательность электронных сообщений хранится в одном файле, но можно работать как с отдельным документом, можно считать присоединенные файлы и сами сообщения отдельными документами. Приложение – архив можно распаковать и каждый файл считать документом.

В длинных документах проблема **детализации индексирования (indexing granularity).**

Пусть есть коллекция книг и запрос “Chinese toys”. При использовании БП слова могут находиться в разных главах книги и результат поиска был бы неудовлетворительным.

Можно выбрать в качестве структурной единицы *абзац, главу, фрагменты.* Или использовать поиск с учетом **близости слов запроса.**

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Далее необходимо разделить текст на **лексемы**.

Ввод: Friends, Romans, Countrymen, lend me your ears

Вывод:

Friends	Romans	Countrymen	lend	me	your	ears

Лексема (token) – это экземпляр последовательности символов в определенном документе, объединенных в семантическую единицу для обработки.

Тип (type) – это класс всех лексем, состоящих из одной и той же последовательности символов.

Термин (term) – это (возможно нормализованный) тип, включенный в словарь системы информационного поиска.

Пример:

«to sleep perchance to dream» – 5 лексем, 4 типа

Если «to» исключить как стоп-слово, то фраза состоит из 3 терминов: sleep, perchance, dream.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Нетипичные случаи разбиения:

Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.

neill	
oneill	
o'neill	
o'	neill
o	neill

aren't	
arent	
are	n't
aren	t

Самая простая стратегия -разбиение по небуквенным символам:

o	**neill**

выглядит правильно, но вторая лексема аналогично неправильна:

aren	**t**

Запросу neill AND capital соответствует три варианта из пяти.
Запросу o'neill AND capital только один из пяти.

Вывод: запрос должен обрабатываться как и коллекция.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ❑ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Специфические лексемы:

Неоднозначные написания слов в запросе и в словаре некоторых могут привести к снижению полноты поиска.

- **Распознавания лексем как терминов:**
  1. Языки программирования (C+, C#)
  2. Названия самолетов (B-52)
  3. Телешоу (M*A*S*H)
  4. Адреса электронной почты (abc@gmail.com)
  5. IP-адреса (142.12.23.456)
  6. URL (http://stuff.big.com/new/serd.html)
  7. …

- **Дефисы:**
  1. Разделение гласных (co-education)
  2. Объединение существительных (Hewlett-Parker)
  3. Группирование слов (the-hold-him-back-and-him-away maneuver)

- **Пробелы:**
  1. White space | whitespace
  2. San Francisco-Los Angeles
  3. 11.12.49 | 11 December 1949

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ❑ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Игнорирование распространенных терминов:

Иногда очень распространенные слова, не представляющие ценности для удовлетворения информационных потребностей, вообще исключаются из лексикона. Они называются **стоп-словами (stop-words).**

Термины упорядочиваются по частоте в коллекции и самые частые из них, отфильтрованные специальным образом, включаются в список стоп-слов, при индексировании его элементы отбрасываются.

Например:

a	at	has	its	to
an	be	he	of	was
and	by	in	on	were
are	for	is	that	will
as	from	it	the	with

Список из 25 стоп-слов, часто встречающихся в Reuters-RCV1

На практике список стоп слов от 7-12, во многих системах от них отказались, так как часто теряется смысл фразы и точность поиска существенно снижается.

Например лексема: *to be or not to be*

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ✓ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Классификация терминов по классам эквивалентности.

Нормализация лексем (token normalization) – это процесс приведения лексем к канонической форме, чтобы устранить несущественные различия между последовательностями символов.

В процессе нормализации, как правило, создаются **классы эквивалентности**, которые обычно называются по имени одного из их членов.

Пример: anti-discriminatory и antidiscriminatory

Альтернативный метод основан на поддержании связей между ненормализованными лексемами. Например, расширение запроса с помощью **словаря синонимов** (объединение ИС для слов-синонимов)

Пример: car и automobile

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ✓ Нормализация
  - ❑ Стемминг
  - ❑ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Распространенные виды нормализации:

❑ **Ударения и диакритические символы**

Так как редко встречаются, то можно считать naïve и naive одинаковыми словами. В некоторых языках это критично, например, peña означает утёс, а pena – горе. Но важно и как это слово будут писать в запросе пользователи.

❑ **Использование заглавных букв – игнорирование регистра**

В большинстве случаев – это удачное решение:
Automobile и automobile
Проблематичнее с именами собственными:
Black, the Fed, General Motors

Специфика английского языка:

Американизмы: colour / color
Даты: 3.12.90 – в США означает март, в Европе – дкабрь

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ✓ Нормализация
  - ✓ Стемминг
  - ✓ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Стемминг и лемматизация

Стемминг (stemming) – приближенный эвристический процесс, в ходе которого от слов отбрасываются окончания в расчете на то, что в большинстве случаев это себя оправдает. Часто происходит удаление производных аффиксов.

Лемматизация (lemmatization) – точный процесс с использованием лексикона и морфологического анализа слов, в результате которого удаляются только флективные окончания и возвращается основная или словарная, форма слова, называемая леммой.

Стемминг	**Лемматизация**
«склеивает» производные однокоренные слова	«склеивает» флективные формы одной леммы

Пример:
Лексема saw после стемминга может стать буквой s, а в ходе лемматизации либо словом see, либо saw, в зависимости от того, глагол это или существительное.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ✓ Нормализация
  - ✓ Стемминг
  - ✓ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Алгоритмы стемминга

Оригинал текста:

Such an analysis can reveal features that are not easily visible from the variations in the individual genes.

Алгоритм Портера – 5 этапов сокращения, выполняемых последовательно:

SSES → SS
IES → I
SS → SS
S →

Such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene.

Алгоритм Ловинса – относительно старый однопроходный алгоритм Ловинса

Such an analysis can reve featur that ar not eas vis from th vari in th individu gen.

Алгоритм Пейса-Хаска

Such an analysis can rev feat that are not easy vis from the vary in the individ gen.

Схематизация документа

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ❑ Схематизация документа
  - ✓ Выделение последовательности
  - ✓ Разделение на лексемы
  - ✓ Стоп-слова
  - ✓ Нормализация
  - ✓ Стемминг
  - ✓ Лемматизация
- ❑ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Сравнение стемминга и лемматизации

Полный морфологический анализ при лемматизации дает довольно скромный выигрыш при информационном поиске. Ни одна из форм нормализации не повышает суммарную эффективность поиска информации на английском языке, по крайней мере, значительно. Для некоторых запросов результат лучше, для других производительность снижается. Стемминг повышает полноту, но снижает точность.

Оригинал текста:
operate operating operates operation operatives operational

После стемминга: oper
После лемматизации:operate

Запрос: operating AND system
Ни oper AND system, ни operate AND system не полностью соответствуют начальному запросу. Даже если повысится полнота, то скорее всего снизится точность результата.

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ❑ Запросы с джокером
  - ❑ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Хэширование и деревья поиска.

Имея инвертированный индекс и запрос, необходимо определить, существует ли в лексиконе каждый термин из запроса и далее идентифицировать указатель на соответствующие термину словопозиции. Для поиска в лексиконе используется классическая структура данных – **словарь (dictionary)**, термины называются **ключами**.

Хэширование:

1) Отображение ключа на довольно большой интервал
2) Для разрешения коллизий используются вспомогательные структуры
3) Близкие варианты термина найти непросто
4) Большая вероятность, что по прошествии времени система станет неработоспособной

Двоичное дерево:

1) Эффективный поиск при сбалансированном дереве
2) Необходимо восстанавливать баланс после add/del
3) Можно использовать B-дерево с кол-м поддеревьев [a,b]
4) Использование с упорядоченными алфавитами

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ❑ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Запросы с джокером используются когда:

- ❑ Неизвестно, как правильно пишется термин
- ❑ Есть несколько вариантов написания
- ❑ Нужны документы, содержащие вариант термина, унифицированный в результате стемминга, но не известно, выполняет ли поисковик стемминг

abc* - запросы с замыкающим джокером
Оптимальный словарь – двоичное дерево

***abc – запросы с ведущим джокером**
Словарь – обратное двоичное дерево

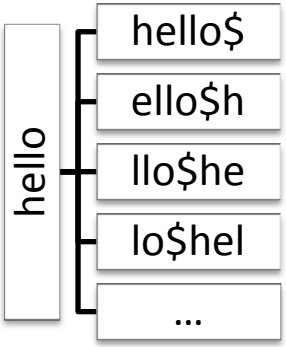
abc*def
Словарь – B-дерево и обратное B-дерево: получаем от каждого множество соответствующих началу и концу терминов, далее ищем пересечение.
Необходимо только исключать случаи типа abc*abc.

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ **Словари и нечеткий поиск**
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ❑ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Запросы с джокером общего вида.

Перестановочный индекс (ПИ)



$ - специальный символ, конец термина
Чтобы построить ПИ, надо перебрать все варианты исходного термина с $ в конце, полученные циклической перестановкой символов.
Набор терминов с перестановкой символов называется **лексиконом перестановок (permuterm vocabulary)**.

Пусть есть запрос *m*n.*

1) Применяем циклическую перестановку: n$m*.
2) Ищем в перестановочном индексе такую строку
3) Восстанавливаем по перестановкам исходные термины

А как обработать запрос вида *fi*mo*er*?

1) Найдем все термины словаря, соотв. индексу er$fi*.
2) Полным перебором отфильтруем термины и оставим те, где есть символы mo.

Недостаток: словарь становится довольно большим.

Словари и нечеткий поиск

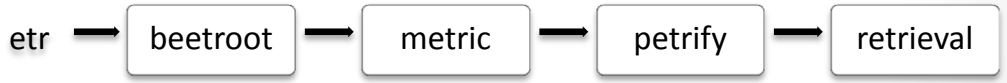
- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ❑ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Запросы с джокером общего вида.

К-граммный индекс для шаблонных запросов

К-грамма – это последовательность, состоящая из К символов. Пример 3-грамм для термина castle: $ca, cas, ast, stl, tle, le$, e$c.

КГ индекс содержит все К-граммы, образованные из всех терминов лексикона. Каждый инвертированный список ставит в соответствие К-грамме все термины лексикона, её содержащие.



Пример: запрос re*ve

1)Переформируем запрос $re AND ve$

2)Поиск по 3-граммному индексу: relieve, remove…

3)Поиск соответствующих терминов в стандартном инвертированном индексе

4)Постфильтрация – сравнение с исходным шаблоном (важно если запрос red*; поиск по $re AND RED; *retired*)

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Реализация исправления ошибок.

Два этапа решения задачи:

1. Основанный на расстоянии редактирования
2. Основанный на пересечении К-грамм

В основе большинства алгоритмов лежит 2 принципа:

1. Выбирается «ближайший» правильный способ написания искаженных запросов
2. Если два правильных варианта одинаково близки, то выбирается более распространенный (по коллекции/по запросам)

2 метода исправления ошибок:

1. Исправление изолированного термина
   carot / carrot
2. Исправление с учетом контекста
   flew form Heathrow / flew from Heathrow

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Реализация исправления ошибок.

Расстояние редактирования между двумя строками s1 и s2 – это минимальное количество операций редактирования, с помощью которых строку s1 можно трансформировать в s2.

Операции редактирования – расстояние Левенштейна:

1) Вставка символа в стр.
2) Удаление символа в стр.
3) Замена символа другим символом

Можно обобщить, присвоив разным операциям разные веса (например, по близости кнопок на клавиатуре или фонетическому принципу)

Для заданных множества строк V и строки запроса q необходимо найти строку или строки из V с минимальным расстоянием редактирования до запроса q.
Вычисляем расстояние от q до всех строк множества V, затем выбираем строку с наименьшим расстоянием.
Очень затратный поиск, поэтому применяются эвристики для сокращения перебора, например, что в заглавной букве не ошибаются.

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Вычисление расстояния Левенштейна

		F		A		s		t	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
C	**1**	1	2	2	3	3	4	4	5
	1	2	1	2	2	3	3	4	4
a	**2**	2	2	1	3	3	4	4	5
	2	3	2	3	1	2	2	3	3
t	**3**	3	3	3	2	2	3	2	4
	3	4	3	4	2	3	2	3	3
s	**4**	4	4	4	3	2	3	3	3
	4	5	4	5	3	4	2	3	3

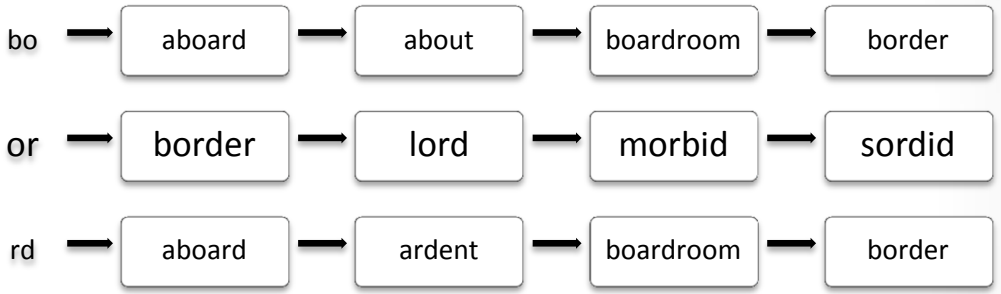
```
EditDistance (s1,s2)
        int m[|s1|,|s2|]=0
        for i ← 1 to |s1|
        do m[i,0]=i
        for j ← 1 to |s2|
        do m[0,j]=j
        for i ← 1 to |s1|
        do for j ← 1 to |s2|
                do m[i,j]=min{m[i-1;j-1]+if (s1[i]=s2[j])
                                then 0 else 1 fi,
                              m[i-1,j]+1,
                              m[I,j-1]+1}
        return m[|s1|,|s2|]
```

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

К-граммные индексы

Чтобы сократить количество вычисляемых расстояний, найдем самые похожие запросы, используя К-значные фрагменты запроса. Пусть сделан запрос *bord.*



Если считать по фиксированному количеству биграмм, то выпадают aboard, border и boardroom. Последний термин почти наверно не является исправлением запроса bord. Чтобы усовершенствовать способ применяется коэффициент Жаккара: $J = |A \cap B| / |A \cup B|$, где A и B – множества. Пусть мы рассматриваем строки q=bord и t=boardroom, для каждой есть множество к-грамм.
J=2/(8+3-2) (числитель – количество совпадений в записях, знаменатель – разность между суммой биграмм и кол-м совпадений в записях).

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ❑ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Исправление опечаток с учетом контекста

Если все термины в запросе по отдельности написаны правильно, но поиск выдал лишь небольшое количество документов, то система может принять решение исправить запрос.

Пример: *flew form Heathrow*

Самое очевидное – перечислить все исправления для терминов, попробовать заменить каждый из них.

Flew from Heathrow OR flew fore Heathrow.

Для каждой фразы система выполняет поиск и ищет соответствия.

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ✓ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Фонетические исправления

Метод фонетических исправлений используется в основном для пользователей, которые записывают запрос так, как они его слышат. Особо полезно при поиске имен людей.

Основная идея: терминам, звучащим одинаково, поставить в соответствие одно и то же число.

Алгоритмы фонетического хэширования обычно называются алгоритмами Soundex. Оригинальный алгоритм выглядит так:

1. Преобразуем каждый индексируемый термин в *четырех символьную сокращенную форму*. По этим сокращенным формам строим инв. индекс для поиска исходных терминов, sound-index.
2. Делаем тоже самое для терминов запроса.
3. Если поступает запрос на сравнение строк по звучанию, выполняем *поиск по индексу Soundex*.

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ✓ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Фонетические исправления

Варианты Soundex зависят от метода преобразования терминов в 4х-символьное представление. В результате применения самого распространенного метода получается 4х-символьный код, в котором первый символ – это буква алфавита, а остальные три – цифры от 0 до 9.

Оригинальный Soundex

B P F V	1
C S K G J Q X Z	2
D T	3
L	4
M N	5
R	6

Улучшенный Soundex

B P	1
F V	2
C K S	3
G J	4
Q X Z	5
D T	6
L	7
M N	8
R	9

Словари и нечеткий поиск

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ❑ Словари и нечеткий поиск
  - ✓ Поисковые структуры
  - ✓ Запросы с джокером
  - ✓ Исправление опечаток
  - ✓ Фонетические исправления
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Фонетические исправления

В улучшенной версии, буквы разбиты на большее количество групп. Буквы H и W просто игнорируются. Кроме того, код не имеет фиксированной длины и не обрезается.

Примеры:

Оригинальный Soundex:

D341 → Дедловский, Дедловских, Дидилев, Дителев, Дудалев, Дудолев, Дутлов, Дыдалев, Дятлов, Дятлович.

N251 → Нагимов, Нагмбетов, Назимов, Насимов, Нассонов, Нежнов, Незнаев, Несмеев, Нижневский, Никонов, Никонович, Нисенблат, Нисенбаум, Ниссенбаум, Ногинов, Ножнов.

Улучшенный Soundex:

N8030802 → Насимов, Нассонов, Никонов.

N80308108 → Нисенбаум, Ниссенбаум.

N8040802 → Нагимов, Нагонов, Неганов, Ногинов.

N804810602 → Нагмбетов.

N8050802 → Назимов, Нежнов, Ножнов.

В среднем, на одно значение кода Soundex приходится 21 фамилия. В случае же улучшенной версии Soundex, к одному и тому же коду преобразуются всего 2-3 фамилии.

Модель ВП, TF-IDF

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Альтернативой модели булева поиска являются **модели поиска с ранжированием**, например, модель векторного пространства, где используются свободные текстовые запросы, а не строгие конструкции с операторами.

Недостатки булевой модели:

- ❑ Не учитывается частота вхождения терминов в документ
- ❑ В результате поиска выдается неупорядоченное множество документов. Хотелось бы найти эффективный метод, позволяющий ранжировать результаты поиска.

Релевантность – степень соответствия документа заданному запросу.

Далее будем рассматривать каждый документ как вектор весов, заданных на основе частоты термина. Найдем соответствие между запросом и каждым из документов. Этот подход называется ранжированием на основе модели векторного пространства.

Модель ВП, TF-IDF

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Частота термина tf_{td} («Term Frequency» — частота слова) — количество вхождений термина t в документ F.

Для документа d набор весов tf можно интерпретировать как характеристику документа. В научной литературе эта модель называется мешком слов (bag of words model). В рамках данной модели точный порядок следования терминов в документе игнорируется. Тем не менее два документа с одинаковыми «мешками слов» по содержанию очень сходны. Но не все же слова вносят одинаковый вклад в поисковые результаты.

Частота в коллекции cf («collection frequency») – позволяет снизить веса у терминов, представляет собой общее количество вхождений термина в коллекцию

Документная частота df_i - количество документов в коллекции, содержащих термин t.

Слово	cf	df
try	10422	8760
insurance	10440	3997

Модель ВП, TF-IDF

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ❑ **Модель ВП, tf-idf**
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

На рис. видно, что документов содержащих термин insurance меньше, хотя появляемость в группе хорошая и примерно у них одинаковая.

Слово	cf	df
try	10422	8760
insurance	10440	3997

Как корректируется такая разница?

Обратная документная частота (invers document fraquency) термина t вычисляется так:

$$idf_t = log \frac{N}{df_t}.$$

Отсюда idf редко встречающихся термина является большой, в то время как для часто встречающегося термина она невелика.

Модель ВП, TF-IDF

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ❑ **Модель ВП, tf-idf**
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Если скомбинируем частоту термина в документе (term frequency) и обратную документную частоту, то получим вес каждого термина в каждом документе.
Схема *tf-idf* присваивает каждому термину t его вес в документе d на основе формулы:

$$tf - idftd = tf \times idf_t$$

Вес обладает следующими свойствами:

- ✓ Достигает максимального значения, если термин t встречается много раз в небольшом количестве документов (тем самым увеличивая их отличие от других документов)
- ✓ Уменьшается, если термин встречается в каком-то документе лишь несколько раз или встречается во многих документах.
- ✓ Достигает минимального значения, если термин встречается практически во всех документах.

Релевантность документа:

$$Score(q,d) = \sum_{t \in d} tf - idftd$$

Модель ВП, TF-IDF

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ❑ Модель ВП, tf-idf
- ❑ Эффективное ранжирование
- ❑ Оценка ИП
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Обозначим через $\vec{V}$(d) **вектор, построенный по документу** d, в котором каждому термину документа соответствует отдельный компонент.

Чтобы выразить сходство в векторном пространстве в количественной форме вычислим косинусную меру сходства

$$sim\left((d_1,(d_1))\right) = \frac{\left(\vec{V}(d_1),\vec{V}(d_2)\right)}{\left\|\vec{V}(d_1)\right\|\left\|\vec{V}(d_2)\right\|}$$

Здесь числитель – скалярное произведение, а знаменатель нормирует по длине – произведение евклидовых норм..

Запрос тоже можно рассматривать как **вектор,** тогда каждому документу мы присваиваем значение релевантности:

$$score(q,d)=\frac{\left(\tilde{V}(d_1),\tilde{V}(d_1)\right)}{\left\|\vec{V}(d_1)\right\|\left\|\vec{V}(d_2)\right\|}$$

На практике подобное ранжирование может оказаться довольно затратным, так как придется находить скалярное произведение для векторов, у которых может быть десятки тысяч компонентов.

Оценка ИП

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ✓ Модель ВП, tf-idf
- ✓ Эффективное ранжирование
- ❑ Оценка ИП
  - ❑ Полнота, точность
  - ❑ F-мера
- ❑ Сниппеты
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Результаты ИП можно оценить по:

1. **Полноте** - отношение числа **найденных** релевантных документов, к общему числу релевантных документов в базе.

2. **Точности** - отношение числа релевантных документов, найденных ИПС, к общему числу найденных документов.

3. **F-мере** – среднее гармоническое полноты R и точности P, которое позволяет совместно оценить результаты поиска:

$$F = \frac{1}{\alpha\frac{1}{P}+(1-\alpha)\frac{1}{R}}$$

Сниппеты

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ✓ Модель ВП, tf-idf
- ✓ Эффективное ранжирование
- ✓ Оценка ИП
- ❑ **Сниппеты**
- ❑ Обратная связь по релевантности
- ❑ Переформулирование запроса

Сниппеты – короткие аннотации документов, по которым пользователи могли бы оценить релевантность документов.

Как правило, сниппеты состоят из заголовка документа и краткой аннотации, извлекаемой автоматически.

Как представить аннотацию в максимально полезном для пользователя виде?

Аннотации бывают:

1. Статические – одинаковые для всех запросов

Фиксированный фрагмент документа, первые 50 слов или определенная зона (заголовок, автор), могут использоваться метаданные, ассоциируемые с документом. Низкая цена и много дисковой памяти – проще сохранять локальную копию всех документов.

2. Динамические/запросозависимые – сниппеты с ключевыми словами в контексте.

Показывают один или несколько фрагментов документа, содержащих несколько термино взапроса, но так же могут включать заголовок, не зависящий от запроса, как и в статическом случае. Удобно, но надо много ресурсов.

Обратная связь по релевантности

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ✓ Модель ВП, tf-idf
- ✓ Эффективное ранжирование
- ✓ Оценка ИП
- ✓ Сниппеты
- ❑ **Обратная связь по релевантности**
  - ✓ Глобальные методы
  - ✓ Локальные меотды

Во многих коллекциях одно и тоже понятие может выражаться разными словами. Это явление, известное как **синонимия,** влияет на полноту поиска в большинстве поисковых систем.

Система может уточнить запрос двумя способами:

- *С участием пользователя*
- *Автоматически*

Методы решения задачи делятся на две категории:

Глобальные

- ❑ Расширение с помощью спец. тезауруса
- ❑ Расширение с помощью автом. генерации тезауруса
- ❑ Методы, похожие на исправление ошибок

Локальные

- ❑ Обратная связь по релевантности
- ❑ Обратная связь по псевдорелевантности
- ❑ Неявная обратная связь по релевантности

Переформулирование запроса

- ✓ Задачи ИП
- ✓ Булев поиск
- ✓ Инвертированный список
- ✓ Схематизация документа
- ✓ Словари и нечеткий поиск
- ✓ Модель ВП, tf-idf
- ✓ Эффективное ранжирование
- ✓ Оценка ИП
- ✓ Сниппеты
- ❑ Обратная связь по релевантности
  - ✓ Глобальные методы
  - ✓ Локальные методы

Обратная связь по релевантности

Метод основан на *привлечении пользователя* к процессу поиска, чтобы улучшить итоговый список.

Схема действий:

1) Пользователь делает короткий/простой запрос
2) Система возвращает первоначальный список результатов
3) Пользователь отмечает релевантные и нет
4) Система учитывает уточняет представление информационной потребности после выбора пользователя

Проходит одна или несколько итераций

Проблемы, которые метод не решает:

- ❑ Неправильное правописание: неправильный запрос – неполный ответ
- ❑ Многоязычные информационные системы – документы на разных языках находятся слишком далеко друг от друга
- ❑ Несогласованность словаря пользователя и коллекции: несовпадение терминов – некорректный ответ