ИНФОРМАЦИОННЫЙ ПОИСК — 1 КОНТРОЛЬНАЯ
ОГЛАВЛЕНИЕ

Теория

Вопрос 1. Основные свойства естественного языка

Естественный язык — в лингвистике и философии языка язык, используемый для общения людей и не созданный целенаправленно (в отличие от искусственных языков).

К основным свойствам естественного языка относят следующие:

- неограниченная семантическая мощность — принципиальная безграничность ноэтического поля языка, способность к передаче информации относительно любой области наблюдаемых или воображаемых фактов;
- эволютивность — неограниченная способность к бесконечному развитию и модификациям;
- манифестируемость в речи — проявление языка в виде речи, понимаемой как конкретное говорение, протекающее во времени и облечённое в звуковую или письменную форму;
- этничность — неотъемлемая и двусторонняя связь языка с этносом.

Существенным свойством языка является его двойственность, находящая своё выражение в существовании следующих языковых антиномий:

- антиномия объективного и субъективного в языке;
- антиномия языка как деятельности и как продукта деятельности;

антиномия устойчивости и изменчивости в языке;

Вопрос 2. Что такое графематический анализ? Что такое лемматизация

Графематический анализ - это программа начального анализа естественного текста, представленного в виде цепочки ASCII символов, вырабатывающая информацию, необходимую для дальнейшней обработки Морфологическим и Синтаксическим процессорами. В задачу графематического анализа входят:

- Разделение входного текста на слова, разделители и т.д.
- Сборка слов, написанных в разрядку;
- Выделение устойчивых оборотов, не имеющих словоизменительных вариантов;
- Выделение электронных адресов и имен файлов;
- Выделение предложений из входного текста;
- Выделение абзацев, заголовков, примечаний.

Морфологический анализ текста осуществляет приведение словоформ, встречающихся в тексте, к нормальному (словарному) виду и определяет морфологические характеристики словоформы

Упрощенная процедура: лемматизация

- Лемматизация – приведение слова к нормальной форме (лесные -> лесной)
- Стемминг –выделение псевдоосновы (лесной -> лес)

Обратная процедура: морфологический синтез.

Вопрос 3. Как работает словарный морфологический анализ?

Методы морфологического анализа:

1. словарный
   - со словарем словоформ

Каждой словоформе поставлена в соответствие основа или лемма
- со словарем основ

2. бессловарный (фактически – со словарем псевдоокончаний)
+ анализ по аналогии («предсказание»)

Схема морфологического анализа со словарем:
- Для неслужебных слов:
- Выделить возможные окончания слова длиной от 0 до 3 символов
- Для каждого полученного окончания определить код окончания по таблице окончаний и номер флективного класса по словарю основ (лемм)
- Если номер флективного класса и номер окончания найдены, то проверить их согласованность по морфологической таблице
- Если согласованность подтверждается, то сохранить данный вариант

Проблемы:
- Дают максимально полный анализ словоформы
- На реальных текстах дают сбои (опечатки, уникальные слова)
- Не существует абсолютно полных словарей – лексика языка непрерывно пополняется
- Для примера – невозможно включить в словарь всю существующую терминологию, имена, фамилии и т.д.

Вопрос 4. Как морфологические анализаторы обрабатывают слова, отсутствующие в словаре

Методы морфологического анализа:
1. словарный
   - со словарем словоформ

Каждой словоформе поставлена в соответствие основа или лемма
- со словарем основ

2. бессловарный (фактически – со словарем псевдоокончаний)
+ анализ по аналогии («предсказание»)

Предсказание в морфологическом анализе:
- Функциональное назначение предсказания – морфологический анализ слов (словоформ), отсутствующих в словаре
- Метод предсказания – выявление аналогий со словоформами, распознаваемыми имеющимся словарем

Алгоритм предсказания для новых слов:
1) предсказание префиксального образования
2) предсказание по концовке, взятой из известных словоформ

Предсказание по префиксу:
- попытка найти существующую словоформу языка, которая максимально совпадала бы справа со входным словом.
- Если левая часть (потенциальный префикс) не длиннее M символов (пяти), а правая часть (совпавшая с известной словоформой) не короче N символов (четырех), то слово разбирается по образцу известной словоформы.

[евро]технологию, *[супер]коньками*

Предсказание по концовке известной словоформы:

Отделяются инвертированные концовки известных словоформа – длины K (пять букв),
Сопоставляются с морфологическими характеристиками:

- *«анием»* - как «ср. род, ед. ч., тв. пад.»

Такая строка заносится в исходный лексикон, если она встречается:

- не менее L раз (трех) и
- чаще конкурентов в пределах одной части речи

ВСЕГДА предусматривается разбор именем существительным, хотя бы неизменяемым.

Вопрос 5. Что такое постморфологический анализ. Основные методы.

- =предсинтаксический анализ
- Предназначен для устранения морфологической омонимии (многозначности) слов
  - Выбор правильной леммы
  - Уточнение морфологических характеристик

Основные методы

- Написание правил,
- Статистические методы, прежде всего, на основе морфологически размеченного корпуса

Примеры правил постморфологического анализа:

- Удаление признаков служебных частей речи для однобуквенных слов, за которыми следуют точки
- Удаление омонимов слова «уже», соответствующих прилагательным, если за ним не стоит запятая или слово в родительном падеже
- Удаление омонимов слова «сорока», если после слова следует числительное (сорок пять)
- Обработка предлогов: удаление у слова, следующего за предлогом, всех омонимов, не соответствующих падежам, которыми обычно управляет данный предлог

Морфологическая разметка корпуса:

- Частеречная разметка, морфологическая разметка (грамматическая разметка):

a) информация о морфологических (грамматических) характеристиках словоформ текста, включаемая в электронное представление этого текста (в виде тегов)
b) процедура добавления такой информации в электронное представление текста (как правило, частично или – редко – полностью автоматизированная)

Вопрос 6. Основные понятия информационного поиска

Сфера науки, которая исследует методы поиска информации, называется **информационный поиск** (information retrieval (IR)).

Документ

Докуме́нт — материальный объект, содержащий информацию в зафиксированном виде и специально предназначенный для её передачи во времени и пространстве.

Свойства:

1. Значительное текстовое содержание
2. Некоторая структура:
   - заголовок,автор, дата - для статей;
   - тема, отправитель, адресат - для писем

Примеры: Интернет-страницы, электронные письма, книги, новости, посты форумов, патенты и многое другое.

Записи базы данных (структурированные таблицы) состоят из хорошо определенных полей

и атрибутов.

Примеры: банковские записи балансы, номера счетов, имена, адреса, даты рождения, номера

социального обеспечения.

(опционально: сравнение документов и записей)

1. (запись) Легко сопоставлять запросы и поля таких баз данных (хорошо определенная семантика)
2. (документ) Текст более сложный – неструктурированная информация

Сопоставление текста запроса с текстом документа и определение того, что такое хорошее сопоставление – **базовый вопрос информационного поиска.**

Информационный поиск – это больше чем поиск по текстам, и больше,чем просто интернет-поиск, хотя эти вопросы являются центральнымиl Поиск осуществляется на основе разных типов данных, разных типов приложений и разных задач.

Задачи

- Ad-hoc поиск: Найти релевантный документ в ответ на произвольный
- Запрос
- Фильтрация: Отобрать нужные пользователю документы
- Классификация: Проставить рубрики документам
- Ответы на вопрос: Дать ответ на заданный вопрос
- Визуализация выдаваемой информации
- Аннотации (рефераты) и др.

Важные понятия:

Relevance – релевантность
Простое (и упрощающее) определение: Релевантный документ содержит информацию, которую искал пользователь, когда задавал запрос поисковой машине.

На релевантность оказывают влияние много **различных факторов**: задача, контекст, опыт пользователя, новизна, стиль.

Тематическая релевантность (отражение заданной темы) vs. Пользовательская релевантность (все остальные факторы)

Модели поиска отражают «взгляд» на релевантность.

Ранжирующие алгоритмы, используемые в поисковых машинах базируются на моделях Поиска.

Большинство моделей описывают статистические свойства текстов (а не лингвистические).

- Простые признаки текстов такие, как слова в отличие от синтаксического разбора и учета предложений
- Лингвистические признаки могут быть частью статистической модели

Evaluation - оценка качества
- Экспериментальные процедуры и меры для сравнения результатов работы систем с ожиданиями пользователей. Метода оценки качества поиска сейчас используются во многих областях. Типично используются тестовые коллекции документов, запросов, и оценки релевантности**. Полнота и точность** – простые примеры оценки качества.

Users and Information Needs –потребность пользователя, информационная потребность
Оценка качества поиска – является “пользователецентричной”. Ключевые слова – это слишком бедное описание действительных информационных потребностей. Взаимодействие и контекст – важны для понимания потребности пользователя. Методы уточнения запроса: расширение запроса, предложение запроса, relevance feedback.

Вопрос 7. Виды поисковых систем по охвату и направленности. Особенности разных типов поисковых систем

По охвату

1. Интернет-поиск
2. Корпоративный поиск

Сравнение

Интернет-поиск

- – Собирает результаты по общедоступному Интернету
- – Проблема ранжирования ранжирования результатов
- – Большие объемы
- – Громадная индустрия – Интернет-реклама
- – Активные исследования: хорошее качество

Корпоративный поиск

- – Собирает информацию разных форматов из совокупности хранилищ
- – Ранжирование документов разного типа
- – Относительно малый объем исследований
- – Хуже качество поиска – сложнее проводить сравнительные исследования
- – Активная сфера исследований

По типу

- Медицинский поиск
- Патентный поиск
- Поиск научных публикаций
- Поиск по законодательству
- Поиск по химическим документам
- …

Вопрос 8. Особенности научного поиска

Проблемы:

- Поиск научной литературы
- Рост публикаций
- Устаревание
- Проблема литературы до интернетовской эпохи
- Цитаты, которые можно использовать как ссылки в Интернет-поиске
    - Большое количество ссылок на работу – фактор значимости работы
    - Наукометрия

Причины цитированияв научной работе:

- Affirmational – базируется на цитируемой работе
- Assumptive – цитаты основоположников
- Conceptual – используются понятия, определения
- Contrastive – сравнение с близкими, но не совпадающими подходами
- Methodological – используется метод, оборудование, инструменты
- Negational – подвергает сомнению
- и др.

Анализ ссылок в литературе

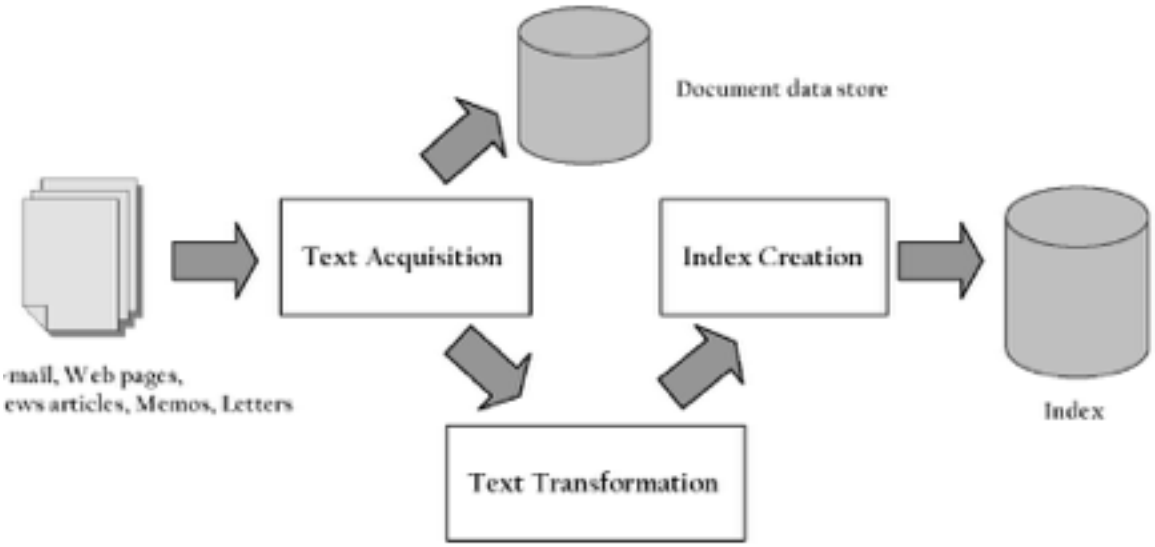
- Количество ссылок на работу
    - Импакт-фактор работы
- Сходство работ на основе ссылок
    - Если работа A цитирует работы B и C, то, возможно, имеется сходство между работами B и C
    - Если работы B и C цитируют одну и ту же работу A, то возможно есть сходство между этими работами
    - Если таких совпадений много, то связь между работами

Системы поиска научной литературы

- CiteCeer
- ScienceDirect
- Google Scholar
- IEEE Xplore

- Scopus
- Microsoft Academic Search

Вопрос 9. Основные этапы обработки текстов в поисковой машине



Извлечение текстов
Идентифицирует и сохраняет тексты для Индексирования.

Краулер: Идентифицирует и извлекает документы для поисковой машины. Много типов – интернет, предприятие, компьютер. Интернет-краулеры используют ссылки, чтобы найти документы. Должны найти огромное количество веб-страниц (покрытие) и сохранять их в актуальном состоянии.

Виды краулеров:

• Краулеры сайтов

• Тематические краулеры для вертикального поиска

Краулеры документов для поиска по документам предприятия или компьютера используют ссылки и сканируют директории.

Фиды

- Потоки документов в реальном потоке времени: Новости, блоги, видео, радио, tv

- RSS – стандарт: RSS читалка обеспечивает новые XML документы поисковой машине **Конвертация**

- Конвертирует форматы в текст плюс мета-данные: HTML, XML, Word, PDF, и др. → XML

- Конвертирует кодировки для различных языков. Например, в кодировку UTF-8

Хранилище документов

Хранит тексты, метаданные и другое содержание документов. Метаданные: тип, дата создания ссылки, текст ссылки - обеспечивает быстрый доступ к созданию документов. Порождение списка результатов. Эффективное хранение - не реляционная база данных.

Трансформация текстов
Трансформирует документы в индексные термы.

Анализатор (Parser)

Обрабатывает последовательность токенов в документе, распознает структурные элементы - Заголовки, ссылки, подзаголовки и др**. Токенизатор** распознает «слова» в тексте. Производится обработка капитализации, кавычек, дефисов и т.д. Обработка структуры, задаваемой HTML, XML. (теги)

Парсер использует синтаксис языка разметки чтобы идентифицировать структуру документа.

Нужно учитывать:

1. **Стоп-слова** - удаление наиболее частотных слов : Предлоги, союзы, артикли.. Но может быть проблемы с некоторыми запросами (быть или не быть)
2. **Стемминг** (морф. анализ) “computer”, “computers”, “computing”, “compute”

   Обычно эффективен, но не для всех запросов. Разное действие для разных языков

Анализ ссылок

Анализ ссылок важен для определения популярности сайта и сообщества, связанного с сайтом, например, PageRank. Текст ссылки может значительно уточнить содержание связанных страниц.

Извлечение информации

Идентифицирует семантические классы индексных термов, которые важны для конкретных приложений индекс, например, распознавание имен людей, географических мест, компаний, дат…

Классификатор

Отнесение текста к категориям, тематика, тональность, жанры и др.Зависит от приложения

Создание индексов
Берет индексные термы и создает индексы для быстрого поиска.

Статистика по документам

Собирает частоты и позиции слов и других признаков. Используется в ранжирующем алгоритме

Определение весов

Вычисление веса для индексных термов. Используется в алгоритме ранжирования, например, вес tf.idf. Комбинирование частоты слова в документе и инверсной подокументной частоты слова в коллекции.

Инвертирование

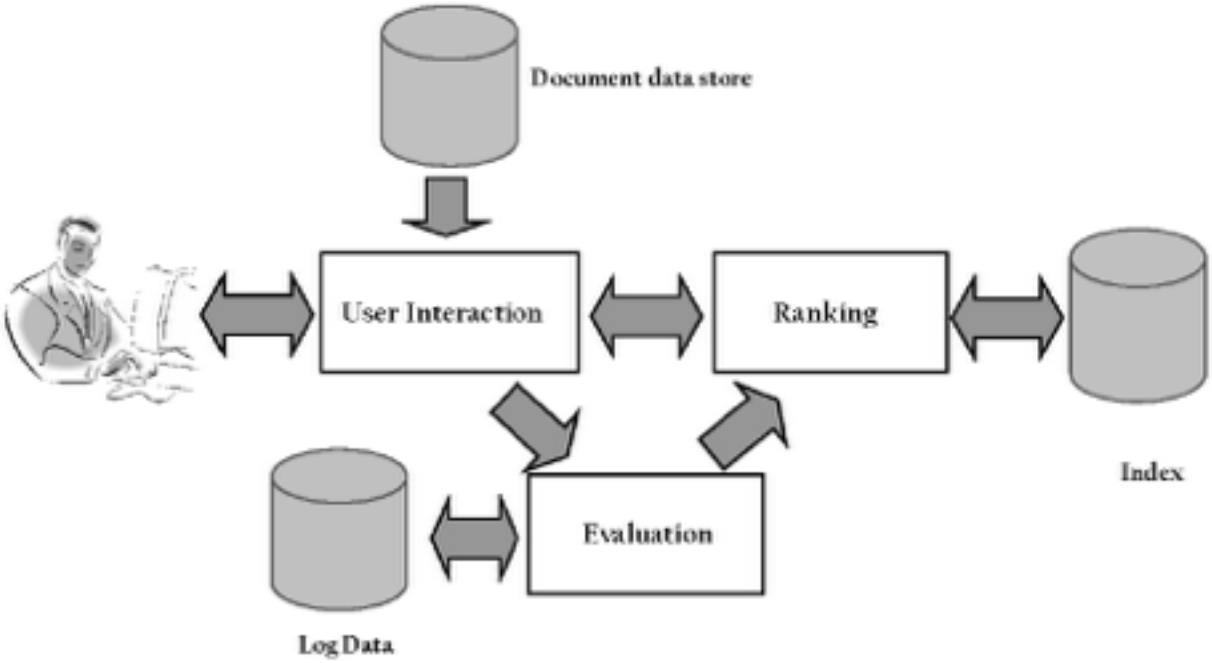
Преобразует матрицу документ-терм в данные терм-документ, необходимые для индексирования. Сложно для большого числа документов. Формат инвертированного файла – для быстрой обработки запросов - Должен обрабатывать изменения индекса и сжимать данные.

Распределенное хранение индекса

Распределяет индексы по многим компьютерам и/или многим дата-центрам. Необходимо для быстрой обработки запросов. Много вариантов: Подокументное распределенное хранение, распределенное хранение термов, повтор данных

Особое направление исследований: Distributed IR – распределенный информационный поиск.

Вопрос 10. Основные этапы обработки запроса в поисковой машине



Взаимодействие с пользователем
Поддерживает создание и уточнение запроса, показ результатов.

Ввод запроса

Интерфейс и парсер для языка запросов. Большинство интернет запросов – простые Язык запросов нужен для описания сложных запросов и результатов трансформации запросов(работа т.н. колдунщиков запросов)

- - Булевские запросы
- - Специализированные языки запросов для информационно-поисковых систем (Indri, Galago)

Сходны с SQL языками, используемыми в базах данных.

Трансформация запросов - улучшает исходный запрос

- Спеллчекинг
- Подсказка запроса
- Автоматическое расширение запроса – пополнение его дополнительными словами
- Relevance feedback – автоматизированная технология с участием пользователя - пользователь размечает релевантные документы.

Выдача результатов

- Строит поисковую выдачу (SERP).
- Порождает сниппеты, чтобы отразить соответствие документа запросу Подсвечивает важные слова
- Показывает релевантную рекламу – основной источник прибыли Интернет-поисковых систем
- Может обеспечивать кластеризацию результатов и другие виды визуализации

Ранжирование
Использует запрос и индексы породить ранжированный лист документов.

Присваивание веса соответствия документа запросам. Веса использует алгоритм ранжирования

- Базовый компонент поисковой машины
- Базовое вычисление веса $\sum$ qi di (qi и di – веса слова запроса и документа)
- Много вариантов алгоритмов вычисления весов и ранжирования

Оптимизация выполнения запроса

Ранжирующие алгоритмы должны позволять эффективное исполнение

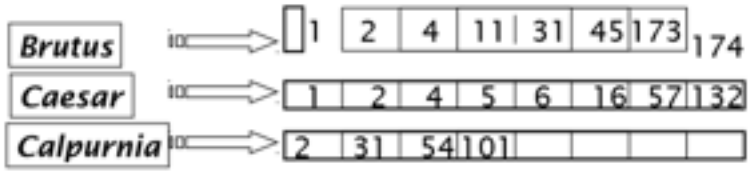
Распределенное выполнение

- Обработка запросов в распределенной среде
- Брокер запросов рассылает запросы и собирает результаты
- Кэширование

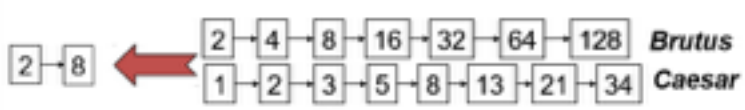
Оценка качества
Мониторит и измеряет качество поиска

Вопрос 11. Алгоритм сопоставления запроса с документами (Алгоритм Merge

Для каждого терма t нужно хранить список документов, которые содержат t. Каждый документ идентифицируется номером документа — docID:



Обработка запроса: AND
• Запрос: «Brutus AND Caesar»
- Найти Brutus в словаре, извлечь все его docid.
–Найти Caesar в словаре, извлечь все его docid.
–“Merge” (пересечь) docid :



Алгоритм:

```
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
```

Вопрос 12. Булевская модель информационного поиска. Преимущества и недостатки булевской модели поиска

Булевская модель рассматривает запрос как булевское выражение. Булевские запросы используют связки AND, OR и NOT между термами запроса, поиск происходит по точному сопоставлению.
- Каждый документ - набор термов
- Каждый запрос – по умолчанию операция И между термами

- Точная выдача: документ либо подходит к условию, либо не подходит (два возможных результата сопоставления запроса и документа: TRUE или FALSE).

Преимущества:
- Результаты предсказуемы, их легко объяснить
- Могут быть встроены многие различные признаки
- Эффективная обработка

Недостатки:
- Качество выдачи зависит исключительно от пользователя
- Простые запросы дают слишком много документов (нет упорядочения)
- Длинные запросы сложно составить

Вопрос 13. Как измеряется качество булевского поиска

Булевский поиск – не имеет ранжирования (упорядочения).
Поисковая система разделяет коллекцию на два множества: выдано ответ на запрос – не выдано.
Эксперты: релевантен – нерелевантен.

Меры качества:
– Точность (Precision (точность): доля релевантных документов в выданных: P(relevant|retrieved))
– Полнота (Recall (полнота): доля выданных документов среди релевантных докуменов = P(retrieved| relevant))

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

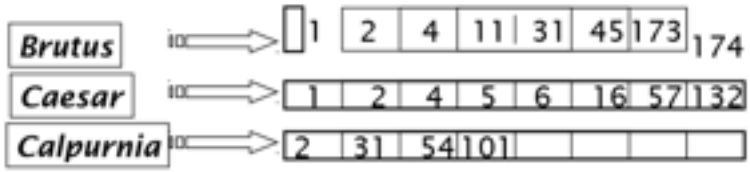
- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

–F-мера:

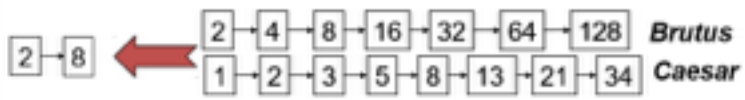
$$F1 = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R}$$

Вопрос 14. Алгоритм сопоставления запроса с документами (Алгоритм Merge)

Для каждого терма t нужно хранить список документов, которые содержат t. Каждый документ идентифицируется номером документа — docID:



Обработка запроса: AND
• Запрос: «Brutus AND Caesar»
- Найти Brutus в словаре, извлечь все его docid.
–Найти Caesar в словаре, извлечь все его docid.
–“Merge” (пересечь) docid :



Алгоритм:

```
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
```

Вопрос 15. Векторная модель информационного поиска. Показатели *idf* и *tf.idf*.

Векторная модель информационного поиска предполагает представление документов и запроса в виде числовых векторов. Слова – это отдельные позиции (оси) в векторах. Документы – точки или векторы в пространстве.
Векторы очень разреженные (много нулей) и очень большой размерности.
Векторное представление не учитывает порядок слов в тексте, не учитывает никакие связи между словами (синтаксические, семантические).

Df – подокументная частотность терма t: количество документов, содержащих t.
Мы определяем idf (обратная подокументная частота):
idf = log 10 (N/Df), N – количество документов.
Логарифм используется, чтобы «смягчить» разницу в употреблении более частотных и менее частотных слов.
Tf- количество вхождений терма в документ.
Вес tf-idf - это произведение tf на idf.
• Могут использоваться всякие модификации tf и idf:
• 1) w t,d=tf*idf, tf=count
• 2) w t,d = log(1+tf)*log10(N/df)
• tf.idf увеличивается при росте количества упоминаний в документе
• Увеличивается для редких терминов в коллекции

у меня есть из инета текст про tf.idf, он по-моему более адекватный, но не из презентаций.
TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency) — статистическая мера, используемая для оценки важности слова в тексте, являющегося частью коллекции текстов. Вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции, то есть если слово встречается в каком-либо документе часто, при этом встречаясь редко во всех остальных документах — это слово имеет большую значимость для
этого самого документа.

Часто встречающиеся в языке слова (например, предлоги и междометия) получают низкий вес TF-IDF, а специфические для данной предметной области слова – высокий.

Принцип построения меры TF-IDF TF — это относительная частотность термина, показывающая, насколько часто термин встречается в документе.

TF термина a = (Количество раз, когда термин a встретился в тексте / количество всех слов в тексте)

IDF — это обратная частотность документов. Она измеряет непосредственно важность термина для текущего документа:
IDF термина a = log (Общее количество документов / Количество документов, в которых встречается термин a)

Итоговым знаечнием меры является произведение TF и IDF:
TF-IDF термина a = (TF термина a) * (IDF термина a)

Предположение: чем меньше угол, тем более похожи векторы.
Косинусная мера: чем больше косинус угла между векторами, тем более похожи документы.

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\cdot\frac{\vec{d}}{|\vec{d}|} = \frac{\sum q_i d_i}{\sqrt{\sum q_i^2}\sqrt{\sum d_i^2}}$$

qi – это tf-idf вес слова i в запросе,di - это tf-idf вес терма i в документе,
q – может быть не только запрос, но и другой документ.
cos(q,d) – это косинусное сходство между q и d или, эквивалентно, косинус угла между q и d.

Вопрос 16. Классическая процедура оценки качества информационно поиска

Классическая (Cranfield)
процедура оценки

• Составим список запросов и ограничим коллекцию документов
• Для каждой пары запрос/документ выставим экспертную оценку «релевантности»
• Будем рассматривать ответ системы не как последовательность документов, а как множество/последовательность оценок релевантности
• На полученной последовательности/множестве оценок релевантности построим метрики

Вопрос 17. Что такое РОМИП, какие задачи в нем решаются?

Конференции по тестированию методов

В России – РОМИП (Российский семинар по методам информационного поиска) (www.romip.ru):
– 2003 – по н.в

– Поддержан грантом РФФИ 04-07-90280-в (рук. Добров Б.В.)
– Участники: Яндекс, Рамблер, Mail.ru, Кирилл и Мефодий, корпорация Галактика, компания RCO, лаб. ЛАИР НИВЦ МГУ

РОМИП

– Соответствует
– Скорее соответствует
– Возможно соответствует – Не соответствует
– Не может быть оценен

РОМИП-2011

• Соревнование по автоматическому анализу тональности текста
• Тональность текста – эмоциональная оценка, выраженная в тексте

• ***Неожиданная*** *развязка и новые герои делают этот фильм* ***непохожим*** *на предшественника.* • Актуальность задачи
– Анализ мнений о политиках, партиях

– Извлечение и представление отзывов о товарах и услугах
– Имидж компании

РОМИП-2011:
анализ тональности
• Данные для обучения – отзывы с числовыми оценками пользователей
– Отзывы о фильмах и книгах с imhonet
– Отзывы о камерах с Яндекс. маркета
• Данные для тестирования
– Отзывы из блогов
– Поиск по блогам Яндекса (ППБЯ) по специальным запросам

– Отбор отзывов и ручная оценка (2 оценки на отзыв)

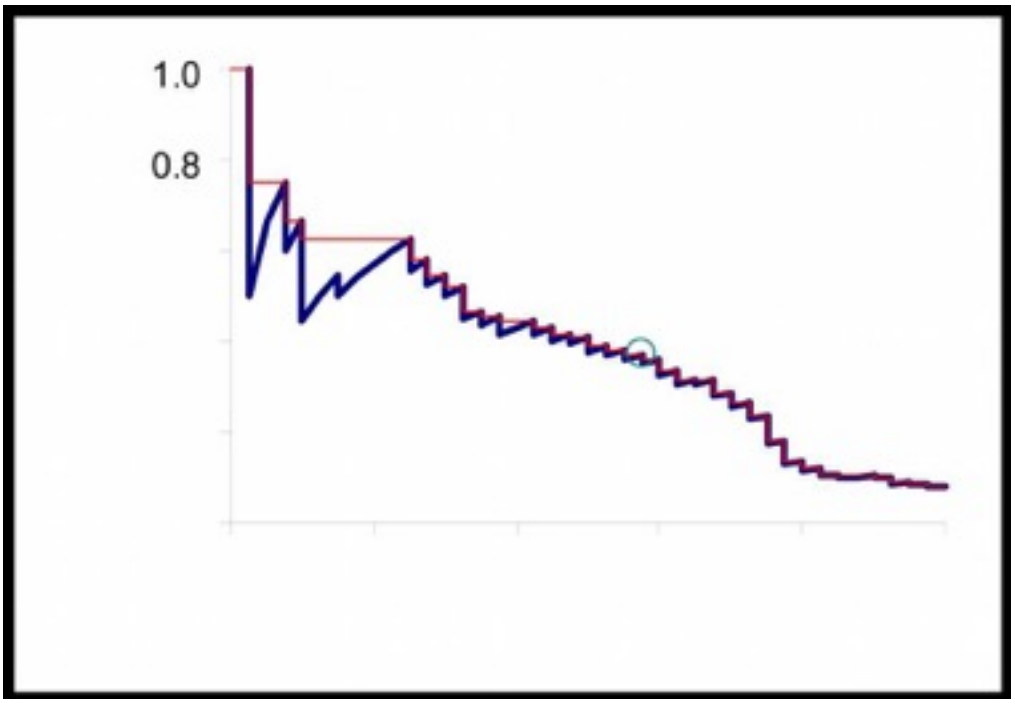
Вопрос 18. Кривая полнота-точность. 11-точечный график TREC?

Усреднение по запросам

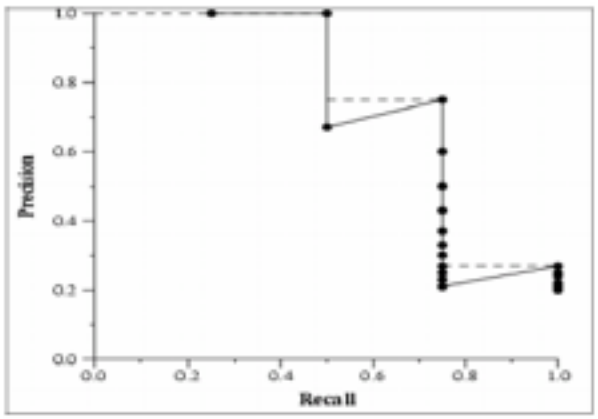
• Кривая полнота-точность для одного запроса не очень интересна

• Нужно построить кривую полнота- точность для совокупности запросов

– Пока Кривая – это совокупность точек – Как интерполировать?



11-точечный график TREC



- Значения полноты от 0 до 1 с шагом 0.5
- Интерполяция точности
  - если $r_i > \text{recall}(q_j)$ то

$$p(r_i, q_j) = 0$$

  - если $r_i \le \text{recall}(q_j)$ то

$$p(r_i, q_j) = \max_{n \ge \text{pos}(r_i, q_j)} (\text{precision}(n))$$

Вопрос 19. Что такое пулинг в информационном поиске? Сложности, связанные с пулингом

Пулинг vs. Полнота
Для каждого запроса:
• Собрать результаты систем участников глубины A
• Выбрать из полученных результатов B первых • Удалить дубликаты
• Проставить оценки релевантности
• Не оцененные документы считать нерелевантными
• Оценить весь ответ системы (с глубиной A)

Сложности, связанные с
пулингом
• Взаимное усиление систем
• Недооценка систем, не участвовавших в оценке
• Получаемая оценка – оценка снизу
• Но: участники относительно в равных условиях

Вопрос 20. Оценка качества в поисковых машинах Интернет.

• Полноту невозможно измерить
• K- первых документов
• Релевантные документы должны показываться раньше • NDCG (Normalized Cumulative Discounted Gain)
• Использование кликов пользователей – A/B testing

Вопрос 21. Шкалы оценок. Мера NDCG

Пусть есть запрос p.
Каждому предложению ставится оценка, в соответствии с его релевантностью запросу по шкале оценок.
g_i - оценка для i-го элемента выдачи
DCG_p - характеристика выдачи
С помощью NDCG можно сравнивать на качество различные поисковики.

Пример шкалы оценок

1. В прошлом: TReC – бинарные ({0,1})
2. Сейчас TReC: ({0,1,2})
   - Высоко релевантный
   - Релевантный
   - Нерелевантный
3. РОМИП
   - Соответствует
   - Скорее соответствует
   - Возможно соответствует
   - Не соответствует

- Не может быть оценен

- Предположения
  - Лучше, если релевантные документы находятся в начале списка
  - Если есть несколько типов релевантных документов, то лучше, чтобы документы с высокими оценками были раньше в списке
- Существует наилучшее упорядочение расположения оценок от лучших к худшим(Ideal DCG)
- Cumulative gain

- Cumulative gain

$$CG_\lambda = \sum_{i=1}^{\lambda} g_i$$

- Discounted Cumulative Gain

$$DCG_\lambda = g_1 + \sum_{i=2}^{\lambda} \frac{g_i}{\log i}$$

- Нормализация DCG по отношению к лучшему упорядочению по данному запросу

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCGp}$$

Вопрос 22. Что такое информационно-поисковые тезаурусы? Зачем они нужны? Где применяются сейчас

- **Информационно-поисковый тезаурус** – нормативный словарь терминов предметной области, создаваемый для улучшения качества информационного поиска в данной предметной области
  - Нормативные ключевые слова
  - Национальные и международные стандарты
  - Используются в ряде международных организаций и парламентский организаций
    - Европейский парламент – EUROVOC
    - ООН – UNBIS Thesaurus

- Цели:
  - Перевод языка авторов на нормативный язык, используемый для индексации и поиска
  - Обеспечение последовательности в присваивании индексных терминов
  - Обозначение отношений между терминами
  - Облегчение информационного поиска
  - Расширение запроса, т.е. поиск не только по словам запроса, но и по близким по смыслу к словам запроса

Применение

- Информационный поиск
  - Корпоративные или предметно-ориентированные системы
  - Автоматическое расширение запроса
  - Визуализация выдачи
- Автоматическая рубрикация текстов

  - Несколько десятков рубрикаторов
- Автоматическая кластеризация текстов
- Автоматическое реферирование текстов
  - Одного документа, многих документов, составление аналитических отчетов
- Системы мониторинга

Вопрос 23. Назовите методы расширения запросов пользователей при информационном поиске.

  - Глобальные методы
    - Ручные тезаурусы
    - Автоматически порождаемый тезаурус
  - Локальные методы (по конкретному запросу)
    - Relevance feedback (обратная связь по релевантности)
    - Pseudo Relevance feedback (обратная связь по псевдорелевантности)

3.Ы. Pseudo-relevance алгоритм:
  - Строит поисковую выдачу по запросу
  - Предполагает, что первые k документов - релевантны
  - Выполняет relevance feedback

Вопрос 24. Алгоритм Роккио для *relevance feedback*

Алгоритм Роккио (Rocchio algorithm) — классический алгоритм для реализации метода RF. Он инкорпорирует модель обратной связи по релевантности в модель векторного пространства, описанную в разделе 6.3.

Теория. Мы хотим найти вектор запроса $\vec{q}$, максимально близкий к релевантным документам и минимально похожий на нерелевантные документы. Если C_r — это множество релевантных документов, а C_{nr} — нерелевантных, то целью наших поисков является следующий вектор.[1]

$$\vec{q}_{opt} = \arg\max_{\vec{q}} \left[sim(\vec{q}, C_r) - sim(\vec{q}, C_{nr}) \right] \quad (9.1)$$

- Проблема: мы не знаем все релевантные документы

Особенности:
- Соотношение α vs. β/γ : Если у нас много оцененных документов, то лучше более высокие β/γ.
- Некоторые веса в модифицированном векторе запроса становятся отрицательными
  - Отрицательные веса слов игнорируются (устанавливаются равными 0)

- На практике используется:

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

- D_r = множество известных релевантных doc векторов
- D_{nr} = множество известных нерелевантных doc векторов
  - Отличны от C_r *и* C_{nr}
- q_m = модифицированный вектор запроса; q_0 = исходный вектор запроса; α,β,γ: веса
- Новый запрос «сдвигается» по направлению к релевантным документам и «уходит» от нерелевантных документов

Вопрос 25. Назовите проблемы расширения запроса при помощи обратной связи по релевантности

Пользователь не хочет размечать документы!
Пользователям необходимо иметь достаточно знаний

- Примеры:
  - Неправильное написание: Brittany Speers.
  - Многоязыковой информационный поиск (hígado).
  - Несоответствие словаря пользователя и словаря коллекции
    - Cosmonaut/astronaut

Не работает если релевантные документы не локализованы среди нерелевантных
Т.е. нарушено

- Распределение слов в релевантных документах сходно
- Распределение слов в нерелевантных документах отлично от распределения слов в релевантных документах
  - 1) Все релевантные документы похожи на один прототип
  - 2) Имеется несколько прототипов, но у них значительное пересечение по составу
  - Сходство между релевантными и нерелевантными документами относительно небольшое

Вопрос 26. Вопросно-ответные системы: постановка задачи. основные компоненты, особенности тестирования.

Выдать ответ на вопрос, а не документ(ы) в которых он содержится.

Вопросно-ответные системы могут быть:

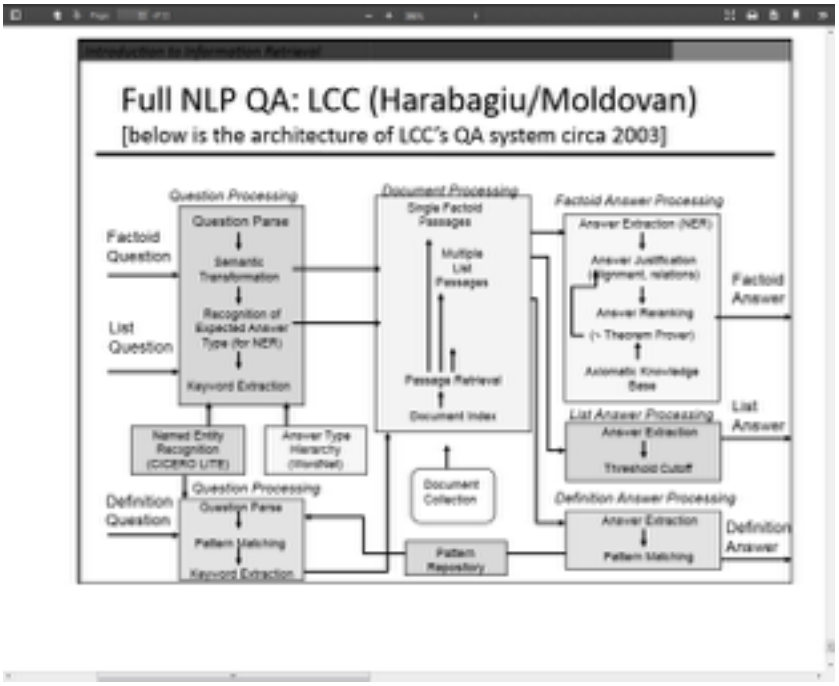
- основанные на ответах по текстовым коллекциям
  - Т.е. заранее набранные ответы на частые вопросы
- основанные на знаниях и гибридные системы
  - IBM Watson, Apple Siri, Wolfram Alpha

Основные компоненты

Для фактоидных:

- Обработка вопроса
  - Семантическая обработка
    - лишние слова
    - учет синтаксической структуры
    - и пр.
  - Определение типа вопроса
  - Выделение ключевых слов
- Поиск в индексе(среди набранных заранее)
- Поиск в коллекции
- Генерация гипотез(отрывков)
- Доказательство гипотез(выбор ответа)

(Подробнее на слайде ниже)



Особенности тестирования

- **Глубокий семантический анализ предложения**
- **Кто был первым лауреатом Нобелевской премии по физике**

  Он стал первым после Михаила Горбачева российским лауреатом Нобелевской премии с 1990 года и первым россиянином, заслужившим Нобелевку по физике, после академика Капицы, который получил ее в 1978 году.

- **Ответ содержится в нескольких разных текстах.**
  - Для создания ответа необходимо уметь выполнять автоматическое аннотирование по многим документам (Multidocument summarization)
  - Тестирование в конференции DUC

Типы вопросов

- **"Factoid" вопрос** - вопрос, основанный на каком-то факте. Предполагает короткий ответ, хранящийся в одном документе.
  (Сколько калорий в Биг Маке?)
- **"List" вопрос** - вопрос, ответ на который состоит из нескольких частей. Информация для ответа хранится во множестве различных документов.
  (Список наименований газировки?)
- **"Other" вопрос** - "а скажи, что я еще не знаю…?"

Вопрос 27. Классификация вопросов в вопросо-ответных системах. Типы вопросов и типы ответов

Вопросы делятся на:

- Фактоидные вопросы
  - краткий ответ
  - *Who is the first American in space?*
- Нефактоидные (нарративные)вопросы
  - Ответ включает предложение или несколько предложений**.**
  - *Расскажите, пожалуйста, о туристических и транзитных визах в США. Что собой представляют визы, выдаваемые супругам, и визы, связанные с обучением? Сколько стоит оформление визы?*

Типы вопросов

- Вопрос типа почему
  - по какой причине, на каком основании, с какой целью, какова причина, какова цель, зачем, для чего, из-за чего, благодаря чему, и т.п. все предлоги из типа фрагм.
  - что + (вызвать/ вызывать/ обуславливать/ обусловить/ определить/ определять/ повлечь/ повлиять/ породить/ порождать/ привести/ приводить/ способствовать)
- Вопрос типа когда
  - в какое время (_ / случиться / происходить / быть / произойти / совершиться / случаться / приключиться / приключаться ; с какого времени, до какого времени / к какому сроку / к какому времени)
- Вопрос типа «где»
- Вопрос типа «как»
- Вопрос типа «кто»
- Вопрос типа «что такое»
- Вопрос типа «куда»
- Вопрос типа «сколько»
- Вопрос типа «какой параметр»
- Вопрос типа «что»
- Вопрос типа «как называется»
- Вопрос на сравнение

Типы ответов

- Имя (вопрос Кто?)
  - Имя человека
  - Название организации
  - Название товара
- Выражение из списка
  - Список месяцев
  - Список государств
- Число
  - Календарные даты
  - Число с единицами измерения

- Фрагменты предложения
  - Именные группы (из-за болезни)
  - Обороты (потому что …)
- Ответ типа ФРАГМЕНТ
  - в начале
  - в конце
  - в то время как …
- Ответ типа СПИСОК
  - день
  - ночь
  - Утро…
- Ответ типа ЧИСЛО
  - Даты, точное время

Вопрос 28. Исправление несловарных ошибок на основе применения правила Байеса

Порождение кандидатов

- 80% ошибок находятся на расстоянии 1
- Почти все ошибки на расстоянии 2
-
- Также позволяется вставка пробела или дефиса
  - `thisidea → this idea`
  - `inlaw → in-law`
  -

  Из множества кандидатов нужно отобрать лучшего кандидата

- Для каждого слова порождается множество кандидатов:
  - С похожим произношением
  - С похожим написанием

- Текущее слово *w* включается в множество

- Выбор лучшего кандидата
  - Подход зашумленного канала
  - Использование контекста
  - *Flying form Heathrow to LAX* → *Flying from Heathrow to LAX*

- Мы видим неправильно написанное слово *x*
- Найдем правильное слово $\hat{w}$

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w|x)$$

$$= \underset{w \in V}{\operatorname{argmax}} \frac{P(x|w)P(w)}{P(x)}$$

$$= \underset{w \in V}{\operatorname{argmax}} P(x|w)P(w)$$

1) Ищем вероятность P(w)

- Пусть *C(w)* = # количество вхождений (буквы) *w*
- T - количество элементов(букв) в корпусе

$$P(w) = \frac{C(w)}{T}$$

2) Ищем P(x|w)

- *P(x|w)* = Вероятность перехода (редактирования)
  - удаления/вставки/замена/перестановки

$$P(x|w) = \begin{cases} \frac{\text{del}[w_{i-1}, w_i]}{\text{count}[w_{i-1}w_i]}, & \text{if deletion} \\ \frac{\text{ins}[w_{i-1}, x_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$

3) Выбираем лучший вариант.

P.S.

- Для униграмм P(*w*) всегда ненулевое
  - Поскольку наш словарь построен на текстовой коллекции
- Но P(w_k|w_{k-1}) может быть нулевым.
- Нужно сглаживание
  - Можно применить сглаживание add-1

Вопрос 29. Исправление ошибок перехода в другое словарное слова на основе применения правила Байса

Короче, тоже самое, что и для несловарных, только еще рассматривается вариант, что ошибки нет, то бишь добавляется P(w|w) которая обычна близка к 1.

Более формально:

Вопрос 30. Учет контекста при исправлении ошибок написания

- Учет условных вероятностей появления одного слова после другого

P($w_1 \ldots w_n$) = P(w_1)P(w_2|w_1)…P(w_n|w_{n-1})

- Вероятности насчитываются на некотором корпусе. Нужно насчитать:
  - P(*wi*) – вероятность униграмм
  - P(w_n|w_{n-1}) – вероятность биграмм
- Для униграмм P(*w*) всегда ненулевое

- Поскольку наш словарь построен на текстовой коллекции
  - Для каждого предложения (фразы, запроса …)
    - Порождение списка кандидатов
      - Само слово
      - Все существующие слова на небольшом редакционном расстоянии (1-2)
      - Слова, близкие по звучанию
      - Все это считается заранее
    - Выбор лучшего кандидата
      - Noisy channel model

- Но $P(w_k|w_{k-1})$ может быть нулевым.
- Нужно сглаживание
  - Можно применить сглаживание add-1
  - Можно применить другой вид сглаживания:

$$P_{li}(w_k|w_{k-1}) = \lambda P_{uni}(w_к) + (1-\lambda)P_{bi}(w_k|w_{k-1})$$

$$P_{bi}(w_k|w_{k-1}) = C(w_k|w_{k-1}) / C(w_{k-1})$$

Пример:

- "a stellar and versatile **acress** whose combination of sass and glamour…"
- Частоты из корпуса современного американского английского со сглаживанием add-1
- P(actress|versatile)=.000021
  P(whose|actress) = .0010
- P(across|versatile) =.000021
  P(whose|across) = .000006
- **P("versatile actress whose") = .000021*.0010 = 210 x10^{-10}**
- P("versatile across whose") = .000021*.000006 = 1 x10^{-10}

Noisy Channel(предполагаем, что ошибки всюду в запросе)

Noisy channel для исправления ошибочных словарных слов

- Дано предложение $w_1, w_2, w_3, \ldots, w_n$
- Множество кандидатов для каждого слова w_i
  - Candidate(w_1) = {w_1, w'_1, w''_1, w'''_1,…}
  - Candidate(w_2) = {w_2, w'_2, w''_2, w'''_2,…}
  - Candidate(w_n) = {w_n, w'_n, w''_n, w'''_n,…}
- Нужно выбрать последовательностьW, которая максимизирует P(W)

Но так выходит слишком много вариантов, поэтому...

Упрощение: Одна ошибка на предложение

- Все возможные предложения с заменой одного слова
  - w_1, **w''_2**, w_3, w_4 two **off** thew
  - w_1, w_2, **w'_3**, w_4 two of **the**
  - **w'''_1**, w_2, w_3, w_4 **too** of thew
  - …
- Нужно выбрать последовательностьW, которая максимизирует P(W)

То есть на формулу вероятности навешиваются произведения вероятности следования слов и максимум ищется уже среди

- P('off''|'two')*p('thew'|'off')*<вероятность off при of>
- P('of''|'two')*p('thew'|'of')*<вероятность of при of>
- P('of''|'two')*p('them'|'of')*<вероятность them при thew>
- …

Задачи

Задача 1. Точность, полнота, F-мера – меры качества

Меры качества: – Точность – Полнота – F-мера

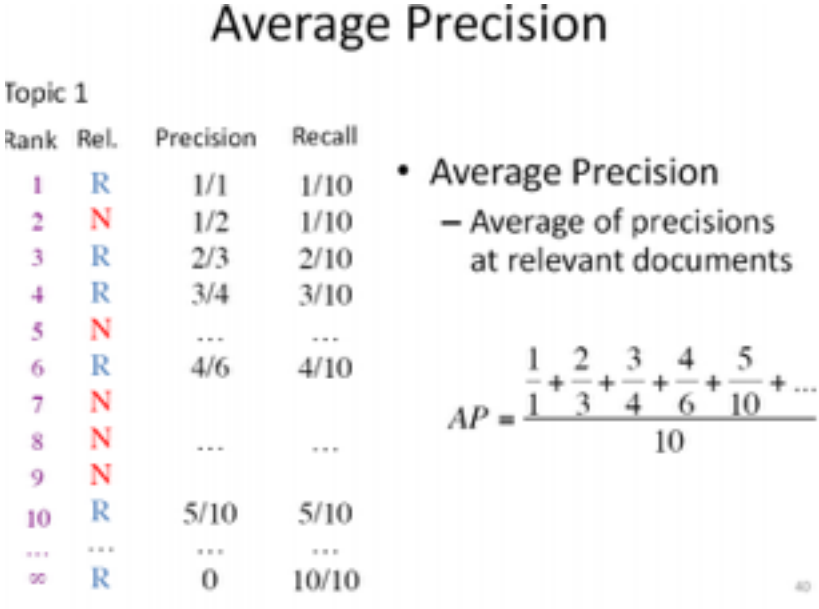
F1 = 2 / (1/P + 1/R)

Оценка неранжированного поиска
• Precision (точность): доля релевантных документов в выданных: P(relevant| retrieved)
• Recall (полнота): доля выданных документов среди релевантных докуменов = P(retrieved| relevant)

Relevant Nonrelevant Retrieved tp fp Not Retrieved fn tn

• Precision P = tp/(tp + fp) • Recall R = tp/(tp + fn)

Задача 2. Мера качества упорядочения: средняя точность



Задача 3. Нахождение близости между запросом и документом по векторной модели, языковой модели

ВЕКТОРНАЯ:
df_t – подокументная частотность *t*: количество документов, содержащих *t*

df_t – это обратная мера информативности *t*
$df_t <= N$
Мы определяем idf (обратная подокументная частота) *t*

$idf_t = \log_{10}(N / df_t)$

Вес tf-idf это произведение tf на idf

Могут использоваться всякие модификации tf и idf:

1) $w_{t,d} = tf * idf$, tf = count

2) $w_{t,d} = log(1 + tf_{t,d}) * log_{10}(N / df_t)$
Tf- это количество вхождений терма в документ

Заметим, что “-” tf-idf – это не минус –Альтернативные имена: tf.idf, tf x idf

---

Увеличивается при росте количества упоминаний в документе Увеличивается для редких терминов в коллекции

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{\left|\vec{q}\right|\left|\vec{d}\right|} = \left|\frac{\vec{q}}{\left|\vec{q}\right|}\cdot\frac{\vec{d}}{\left|\vec{d}\right|}\right| = \frac{\sum q_i d_i}{\sqrt{\sum q_i^2}\sqrt{\sum d_i^2}}$$

qi – это tf-idf вес слова*i* в запросе

di - это tf-idf вес терма i в документе
q – может быть не только запрос, но и другой документ
cos(q,d) – это косинусное сходство между q и d или эквивалентно, косинус угла между q и d.

Если вектора уже нормализованы по длине, то косинус вычисляется:

$$\cos(\vec{q},\vec{d}) = \vec{q}\cdot\vec{d} = \sum q_i d_i$$

ЯЗЫКОВАЯ:

Задача 4. Мера упорядочения: NDCG

Оценки для не бинарного случая релевантности

- Cumulative gain

$$CG_\lambda = \sum_{i=1}^{\lambda} g_i$$

- Discounted Cumulative Gain

$$DCG_\lambda = g_1 + \sum_{i=2}^{\lambda} \frac{g_i}{\log i}$$

NDCG (Normalized Cumulative Discounted Gain)

Нормализация DCG по отношению к лучшему упорядочению по данному запросу

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCGp}$$

Задача 5. Расширение запроса методов relevance feedback.

Пользователь оценивает документы в поисковой выдаче
– Пользователь задает относительно простой, короткий запрос
– Затем пользователь размечает часть результатов как релевантные и нерелевантные
– Система вычисляет улучшает соответствие документов запросу на основе пользовательской разметки
– Процедура может выполняться итеративно.
Отрицательные веса слов игнорируются (устанавливаются равными 0)

Rocchio 1971 алгоритм (SMART)

На практике используется:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

D_r = множество известных релевантных doc векторов

D_{nr} = множество известных нерелевантных doc векторов

– Отличны от C_r и C_{nr}

q_m = модифицированный вектор запроса; q_0 = исходный вектор запроса; α,β,γ: веса

Новый запрос «сдвигается» по направлению к релевантным документам и «уходит» от нерелевантных документов