**Вопросы 2**

1. **Автоматическая классификация текстов. Задачи классификации. Основные подходы к автоматической классификации (рубрицирования) текстов.**

07-01classf.ppt: 2-3

Классификация/рубрикация информации –

отнесение порции информации к одной или нескольким категориям из конечного множества рубрик

* Виды:
  + Тематическая классификация (рубрикация)
  + Жанровая классификация (новости – научные тексты – техническое задание)
  + Классификация по авторам
  + Фильтрация спама
  + Классификация по тональности текстов
* Должна быть задана система классов (категорий) для классификации
  + vs. кластеризация

Применение:

* + Навигация по коллекции документов
  + Замена сложного запроса
  + Иерархическое упорядочение знаний предметной области
    - Анализ распределения документов по тематике
  + Фильтрация потока текстов:
    - Тематический сбор новостей
    - Персонализированная фильтрация потока текстов
    - Тематический сбор информации из интернет

*Методы классификации:*

* Ручное рубрицирование
* Автоматическое
  + Инженерный подход (=методы, основанные на знаниях, экспертные методы)
  + Методы машинного обучения
* Полуавтоматическое

1. **Что такое инженерный метод классификации текстов? Плюсы и минусы инженерных методов классификации**

07-01classf.ppt: 7, 9

07-02rubr.ppt: 32-37, 57

Инженерный подход (=методы, основанные на знаниях, экспертные методы)

* Основное предположение: рубрикатор создается осмысленно, содержание рубрики можно выразить ограниченным количеством понятий в виде формулы
* Эксперты описывают смысл рубрики в виде булевских выражений, правил продукции

Расширение с помощью тезаурусов - представление рубрики в виде булевской формулы для небольшого числа ОПОРНЫХ концептов, затем автоматическое расширение с использованием иерархической структуры Тезауруса.

*Проблемы методов, основанных на знаниях:*

* Содержание рубрики сложнее, чем это выглядит по формулировке
* Лексическая многозначность

текст может быть отнесен не к той рубрике из-за того, что некоторые слова, сопоставленные рубрике, в конкретном тексте употреблены в таком значении, которое не соответствует данной рубрике

(морские суда, решение суда, старинное здание суда)

* Ложная корреляция
* Нестандартный контекст употребления терминов
* Упоминание терминов вне главной темы
* Неполнота описания рубрики (например, отсутствуют аббревиатуры)

*Методы, основанные на знаниях («инженерный» подход)*

* + высокая эффективность
  + «прозрачность» получаемых результатов
  + трудоемкость описания рубрик

1. **Укажите плюсы и минусы ручного рубрицирования.**

07-01classf.ppt: 8

07-02rubr.ppt: 21

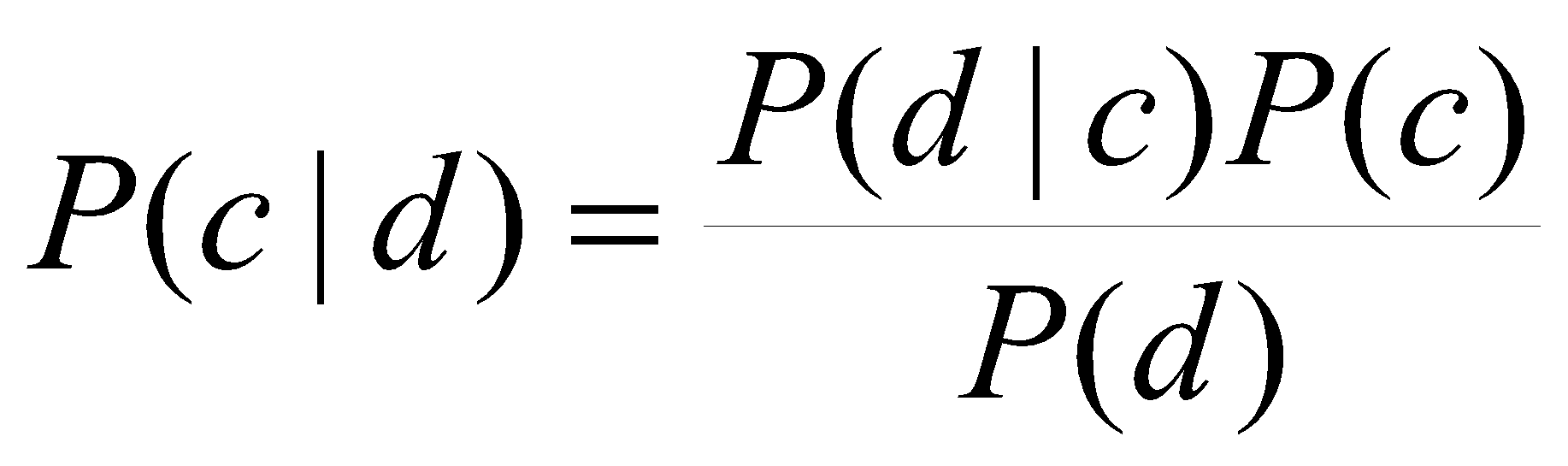
* Высокая точность рубрицирования
  + Обычно процент документов, в которых проставлена явно неправильная рубрика, чрезвычайно мал
* Низкая полнота рубрицирования
  + одна-две основных рубрики, характеризующие основное содержание документа, хотя документ может быть отнесен и к ряду других рубрик.
  + В результате получается, что
    - Процент совпадения результатов рубрицирования различных экспертов весьма низкий - 60 %.
    - В результате похожие документы могут получить достаточно разные наборы рубрик
    - Непоследовательность ручного рубрицирования
* Низкая скорость обработки документов

1. **Метод Байеса для автоматической классификации текстов**

07-01classf.ppt: 32-45

*Релевантность документов рубрике:*

* Пользователь (эксперт) – размечает документы, относящиеся к рубрике
  + Эта разметка производится оффлайн.
  + Может быть результатом ручной рубрикации
* Определение лучшего класса для документа
  + P (C|di) – условная вероятность: вероятность класса С при предъявлении документа di
  + Метод Байеса из курса теории вероятности



*Байес:*

Задача: Классифицировать новый документ *d на основе совокупности признаков в один из классов cj* ∈ *C*

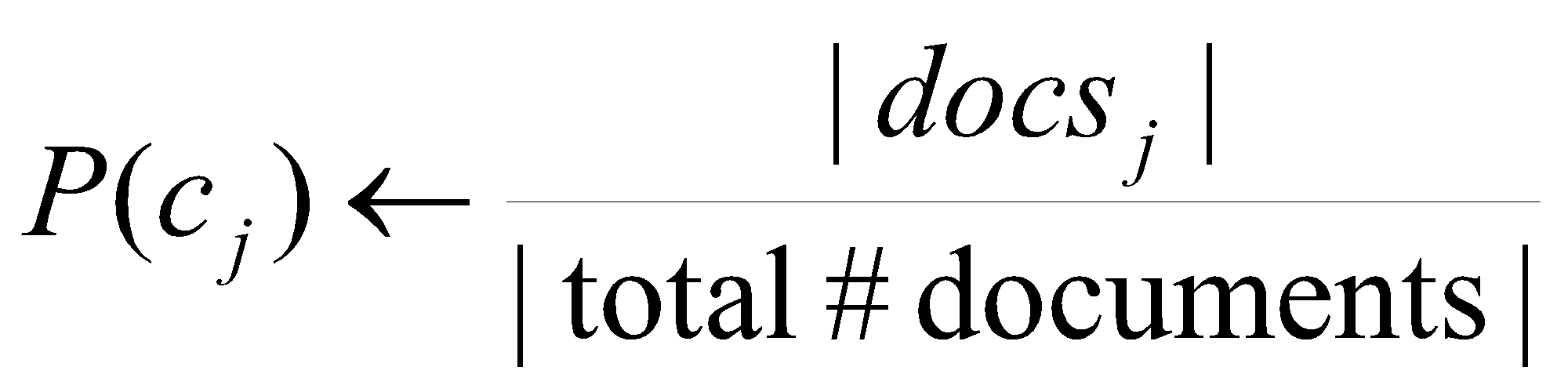
*Naive Bayes*

Предположение о независимости:

Предположим, что вероятность наблюдения конъюнкции атрибутов равна произведению индивидуальных вероятностей *P*(*xi*|*cj*).

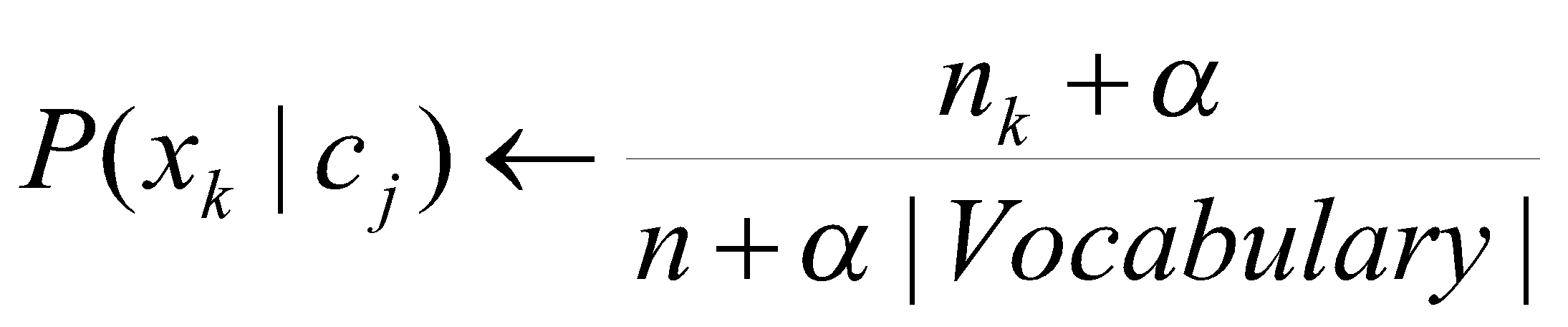
Нужно сглаживание, чтобы добавить тем признакам, которые не встречались, немного вероятности (чтобы не занулялось все).

*Наивный Байес. Обучение:*

* Из обучающего корпуса извлекаем *Vocabulary – совокупность разных слов*
* Вычисляем *P*(*cj*)and *P*(*xk | cj*)
  + Для каждого *cj* in *C* do
    - *docsj* ← *множество документов, в которых проставлен класс - cj*
    - 

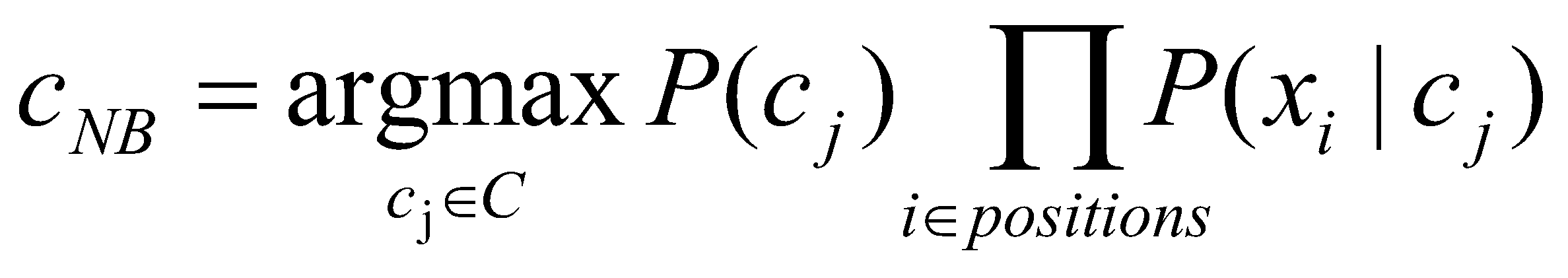
*Textj* ← документ, содержащий все l *docsj класса Сj*

Для каждого слова *xk* in *Vocabulary*

* + - *nk* ← number of occurrences of *xk* in *Textj*
    - 

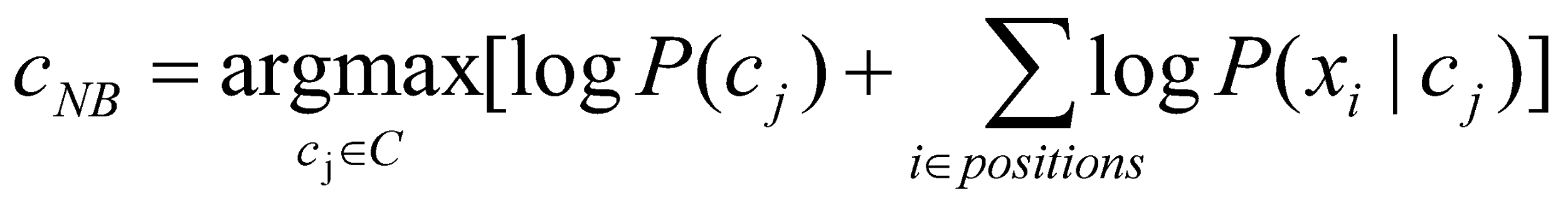
*Наивный Байес. Классификация:*

Return *cNB*, where



*Можно сделать переход к логарифмам:*

* Перемножение большого числа вероятностей от 0 до1 может привести к проблемам типа floating-point underflow.
* Так как log(*xy*) = log(*x*) + log(*y*), лучше выполнять все операции, суммируя логарифмы
* Класс с максимальным логарифмом – является наиболее вероятным



* Заметим, что полученная модель классификации – это просто сумма весов

1. **Байесовский классификатор в задаче обнаружения поискового спама**

07-01classf.ppt: 46-51

* Применяется во многих системах распознавания спама
  + «убийца спама»
* **Спам** - рассылка коммерческой и иной рекламы или иных видов сообщений лицам, не выражавшим желания их получать

*Извлечение признаков:*

* Заголовок: получатель, отправитель, доменные имена
* Текст
  + Слова, фразы, строки символов
  + Могут быть бинарными или числовыми
  + URL, HTML tags, картинки

*Байес:*

* Два класса: спам – неспам
* Вычисление вероятностей
  + Вероятности классов: доля каждого класса в обучающей выборке
  + *Вероятности признаков P*(*xk | cj*): *– количество вхождений в классе/число признаков в классе*
* Могут быть варианты формул, связанные с тем, что пропустить спам менее опасно, чем отправить нормальное письмо в спам
* Используется во многих системах фильтрации и в настоящее время
  + Возможно, что предположение о независимости признаков, больше соответствует действительности

1. **Метод Роккио для автоматической классификации текстов**

07-01classf.ppt: 19-24

* Методы relevance feedback могут быть адаптированы к классификации текстов
* Использование tf-idf векторов для представления документов
* Для обучения вычисляется вектор-центроид, суммированием векторов в обучающей коллекции для каждой категории
  + Prototype = centroid of members of class
* Тестовые документы присваиваются категории, вектор которой наиболее близок по косинусной мере
* Центроид:



где *Dc* is множество всех документов, которые принадлежат классу *c* and *v*(*d*) is векторное представление документа *d.*

Выводы:

* Простое представление для каждого класса: центроид-прототип
* Классификация основана на близости центроиду
* Мало используется вне классификации текстов
  + Может быть эффективна для классификации текстов
  + Но обычно хуже, чем Naïve Bayes
* Простое обучение и тестирование

1. **Метод Knn для автоматической классификации текстов**

07-01classf.ppt: 25-31

* kNN = k Nearest Neighbor
* Чтобы классифицировать документ *d* в класс c нужно:
* определить *k*-окружение as *k* ближайших соседей *d*
* Посчитать число документов в N, которые относятся к c
* Оценить P(c|*d*) as i/k
* Выбрать максимальный класс argmaxc P(c|*d*) [ = majority class]

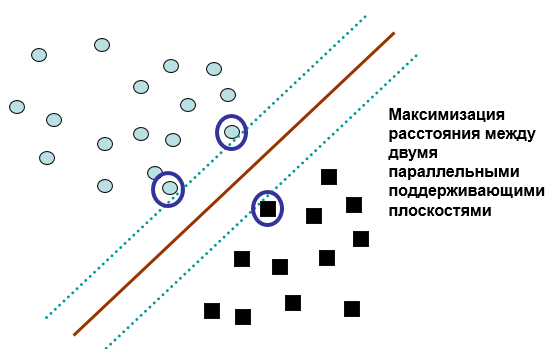
Выводы:

* Использование только одного ближайшего соседа (1NN) ведет к ошибкам из-за:
  + нетипичных примеров
  + ошибок в ручной привязке единственного обучающего примера.
* Более устойчивой альтернативой является *k* наиболее похожих примеров и определение большинства
* Величина *k* is типично нечетная: 3, 5 (наиболее распространенные величины)

1. **Поясните основной принцип метода SVM для автоматической классификации текстов**

07-02rubr.ppt: 9-17

* Метод опорных векторов (SVM) находит оптимальное решение
  + Максимизирует расстояние между гиперплоскостью и трудными точками, близкими к границе раздела
  + Интуитивно: если нет точек около границы раздела, то нет и сложных (неопределенных) примеров



* Интересует максимальное разделение, т.е. две параллельные гиперплоскости. Они могут быть описаны равенствами
* w\*x-b=1
* w\*x-b=-1
* Нужно максимизировать расстояние между плоскостями, которое d=2/|w|, т.е. минимизировать |w|
* В итоге задача квадратичной оптимизации:
  + |w|-> min, при условиях
  + w\*x-b>>1, ci=1
  + w\*x-b<<-1,ci=-1

*Классифицирующая линейная функция*

f(x) = Σαici<xi,x> + b

* где xi – это опорные вектора
* <xi,x> - скалярное произведение опорного вектора на вектор текущего документа
* сi – это {-1,1}
* αi – вес опорных векторов

*Нелинейный СВМ:*

Общая идея: исходное пространство признаков может быть отображено в пространство большей размерности так, что обучающее множество стало разделимо

1. **Плюсы и минусы методов машинного обучения для классификации текстов**

07-02rubr.ppt: 27, 57

Проблемы:

* размер рубрикатора больше 300-500 рубрик, обычно со сложной иерархией
* трудно обеспечить достаточную по качеству и количеству обучающую коллекцию, субъективизм ручного индексирования (обучающей коллекции) значительно возрастает
* сложные задачи решаются на основе инженерных подходов или с помощью частичной автоматизации

*Машинное обучение*

* + Эффективно при наличии качественно размеченной обучающей коллекции
  + низкая эффективность при большом числе рубрик
  + трудно интерпретируемые результаты («черный ящик»)

1. **Особенности применения методов машинного обучения при классификации текстов в зависимости от размера обучающей коллекции**

07-02rubr.ppt: 51-56

***Нет данных:***

* Пишем правила для категорий:
  + If (wheat or grain) and not (whole or bread) then
    - Categorize as grain
* На практике правила гораздо более громоздкие
  + Можно использовать веса tf или tf.idf
* Можно добиться высоких результатов
  + Construe: 94% recall, 84% precision over 675 categories (Hayes and Weinstein 1990)
* Большой объем работы
  + Примерно 2 дня на класс … плюс поддержание

***Мало данных:***

* If you’re just doing supervised classification, you should stick to something high bias
  + There are theoretical results that Naïve Bayes should do well in such circumstances (Ng and Jordan 2002 NIPS)
* The interesting theoretical answer is to explore semi-supervised training methods:
  + Bootstrapping, EM over unlabeled documents, …
* The practical answer is to get more labeled data as soon as you can
  + How can you insert yourself into a process where humans will be willing to label data for you??

***Разумное количество данных:***

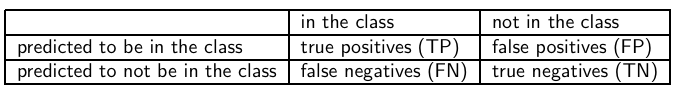
* Perfect!
* We can use all our clever classifiers
* Roll out the SVM!
* But if you are using an SVM/NB etc., you should probably be prepared with the “hybrid” solution where there is a Boolean overlay
  + Or else to use user-interpretable Boolean-like models like decision trees
  + Users like to hack, and management likes to be able to implement quick fixes immediately

***Очень много данных:***

* This is great in theory for doing accurate classification…
* But it could easily mean that expensive methods like SVMs (train time) or kNN (test time) are quite impractical
* Naïve Bayes can come back into its own again!
  + Or other advanced methods with linear training/test complexity like regularized logistic regression (though much more expensive to train)

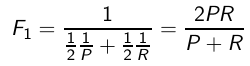
1. **Метрики качества для систем автоматической классификации (рубрикации) текстов. Микро и макро усреднение**

07-01classf.ppt: 54-56



*P* = *TP* / ( *TP* + *FP*)

*R* = *TP* / ( *TP* + *FN*)



* + Посчитали меру оценки (*F*1) for one class.
  + Как агрегировать оценки F1 для многих классов.
  + Macroaveraging - макроусреднение
    - Посчитать *F*1 для каждого из *C* классов
    - Среднее арифметическое для этих *C* чисел
  + Microaveraging - микроусреднение
    - Посчитать TP, FP, FN для каждого из *C* classes
    - Суммировать эти С чисел для каждого показателя
    - Посчитать *F*1 для суммированных TP, FP, FN

1. **Что такое кластеризация текстов? Чем она отличается от классификации (рубрикации) текстов? Типы методов кластеризации**

08-01clust.ppt: 2, 9

* Имеется текстовая коллекция
* Нужно разбить коллекцию на классы близких документов
* Могут быть созданы иерархические классы

Кластеризация (обучение без учителя) отличается от классификации тем, что неизвестно множество классов, на которое коллекция разбивается.

*Алгоритмы кластеризации:*

* Плоские алгоритмы
  + Обычно начинаются со случайного разбиения
  + Итеративное уточнение
    - *K средних (К* means)
* Иерархические алгоритмы
  + Снизу-вверх, аггломеративный
  + (Сверху-вниз, разбиение)

1. **Метод K-means для кластеризации текстов**

08-01clust.ppt: 11-18

*Основные идеи:*

* Рассматривает документы как вектора с вещественными значениями
* Кластеры базируются на понятии центроидаточек в кластере, *c*:



* Присваивание документов к кластеру базируется на сходстве с текущими центроидами кластеров
* Число кластеров задается заранее.

*Алгоритм:*

1. Выберем *K* случайных документов {*s*1, *s*2,… *sK*} как исходное множество (seeds) – это как бы центроиды будущих кластеров.
2. До тех пор пока кластеризация не сойдется (или другой критерий остановки):
   1. Для каждого документа *di*:

*Присваиваем* *di* к кластеру *cj*такому, что similarity(*xi*, *s*j) - максимально.

* 1. Затем обновляем множество {*s*1, *s*2,… *sK*}, заменяем на центроиды текущих кластеров

Для каждого кластера *cj: s*j = μ(*cj*)

*Условие останова:*

Несколько возможностей

* + Фиксированное число итераций
  + Не меняется разделение по документов
  + Не меняется позиция центроидов

*Выводы:*

K-means работает, если

* + Кластеры сферические
    - Что происходит, если кластеры вытянутые
  + Кластеры хорошо разделены
  + Кластеры похожего объемы
    - Что будет, если кластеры разного объема
  + Кластеры имеют сходное количество элементов
    - Что будет, если элементов одного кластера намного больше, чем другого.

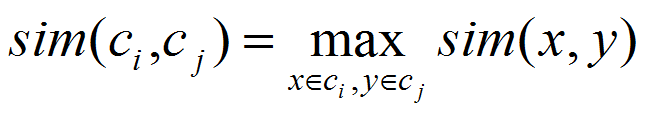
1. **Аггломеративная кластеризация – основной принцип и подвиды**

08-01clust.ppt: 22-27

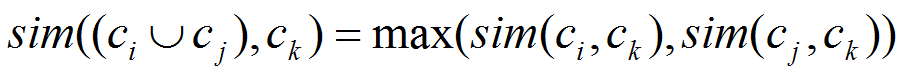
* Начинает с рассмотрения документов как отдельных кластеров
  + итеративно объединяет ближайшую пару кластеров, до тех пор пока не останется один кластер.
* История объединения и образует бинарное дерево или иерархию

Виды:

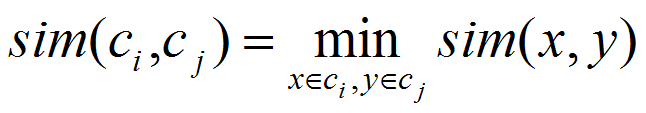
* Много способов определения, что такое наиболее сходная пара кластеров
* **Single-link**
  + Сходство по наиболее похожим документам (длинные и тонкие кластеры-цепочки)
  + Использует максимальное сходство пар:



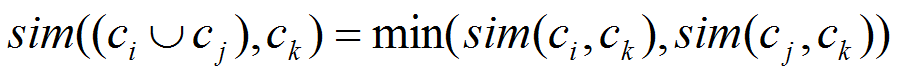
* После склеивания ci и cj, сходство результирующего кластера к другому кластеру, ck,:



* **Complete-link**
  + Сходство по наиболее непохожим документам (более «плотные», сферические кластеры)
  + Использует наименее сходные пары:



* + После склеивания ci и cj, сходство результирующего кластера с другим кластером, ck, :



* **Центроид**
  + Сходство по наиболее похожим центроидам
* **Average-link**
  + Средний косинус между парами элементов двух кластеров

1. **Методы тестирования автоматической кластеризации**

08-01clust.ppt: 28-33

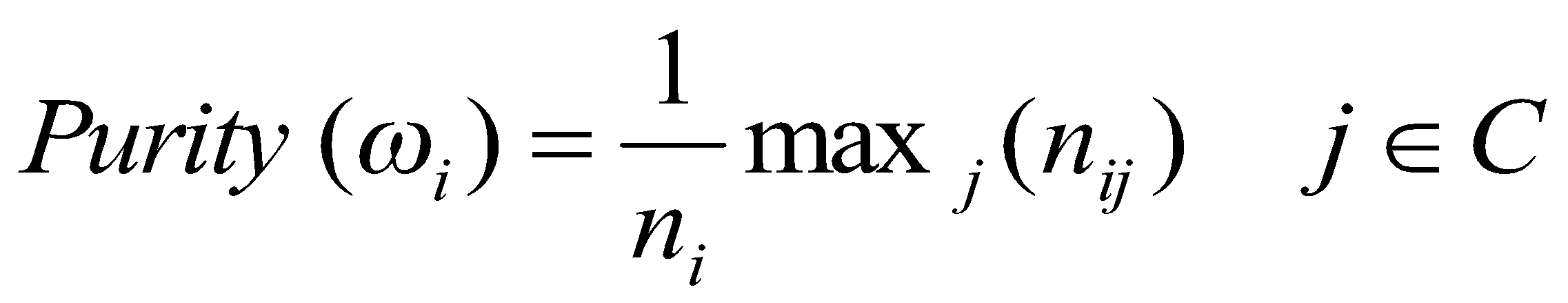
*Внутренний критерий:*

Хорошая кластеризация производит качественные кластеры, в которых:

* Внутри кластера сходство высокое
* Между классами – сходство низкое
* Измеряемое качество кластеризации зависит и от документа, и от меры сходства

*Внешние критерии качества кластеризации:*

* Качество измеряет способность кластеризации обнаруживать скрытые классы объектов в эталонных данных (gold standard)
* Оценивает кластеризацию по отношению к «истинным» кластерам (ground truth) … требует *размеченных данных*
* Предположим, что имеются документы с *C* правильными кластерами, тогда как наш алгоритм порождает *K* кластеров, ω1, ω2, …, ω*K* с *ni* элементами.
* Простая мера: purity, Отношение между доминантным классом в кластере πi и размером кластера ωi



* Мера: Rand

|  |  |  |
| --- | --- | --- |
| Число документов | Тот же кластер в кластеризации | Разные кластеры в кластеризации |
| Тот же кластер в эталоне | A | B |
| Другой кластер в эталоне | C | D |

+ точность, полнота, f-мера

1. **Особенности кластеризации потока новостей в реальном времени**

08-01clust.ppt: 34,41

*Особенности обработки новостного потока:*

* Корпус документов постоянно пополняется
* Временное окно (24-72-120 часов)
* Разные размеры
* Наличие дубликатов, определение первоисточника
* Ошибки при сборе новостных сообщений:
  + ошибки очистки
  + ошибки датировки
* Спамерские технологии источников

*Требования к кластеризации:*

* Минимизация времени работы (максимально 15-30 минут)
* Актуальность (главное сообщение)
* Публикация всех кластеров (не только больших, но и малых)
* Учет перепечаток
* Точность важнее полноты
* Эволюционность кластеризации
* Учет ручного вмешательства:
  + корректировка кластеризации
  + корректировка представления на портале

1. **Автоматическое аннотирование. Виды автоматических аннотаций.**

08-02sum.ppt: 3-6

*Автоматическое аннотирование документа* (совокупности близких по смыслу документов) - автоматическая технология, передающая в краткой форме основное содержание документа (совокупности документов)

* + Аннотирование отдельного документа,
  + Аннотирование совокупности документов -построение обзорного реферата

Назначение: быстрое ознакомление с содержанием документа (совокупности документов)

*Типы аннотаций:*

* Абстракты vs. Экстракты
  + Экстракты получаются извлечением фрагментов исходных текстов (обычно предложений) – основная применяемая технология
  + Абстракты порождаются – экспериментальные технологии
* Типы по основному содержанию
  + Индикативные – собственно содержание
  + Информативные – упор на цифры, данные
  + Оценочные – упор на мнения
* По фокусу
  + Общее содержание
  + В ответ на запрос (Query-based - сниппет) – контекстная аннотация, тематически-ориентированная аннотация
* По структуре
  + Связная аннотация
  + Аннотация ключевыми словами
* Одного документа или многих документов
* На том же языке, на другом языке

1. **Методы и признаки для отбора предложений в экстрактивном методе автоматического аннотирования**

08-02sum.ppt: 8, 11-18

*Извлечение предложений:*

* Обработка документа – разделение на предложения
* Подсчет веса предложения на основе некоторых характеристик
* Упорядочение предложений по мере снижения веса
* Отбор предложений с максимальным весом для аннотации

*Подходы:*

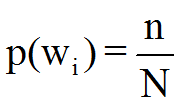
* Частотные подходы:
  + самое важное должно быть частотно
* Подходы, основанные на графах
  + Центральность предложения
* Подходы, основанные на машинном обучении
  + Признаки предложения и их комбинирование
* Подходы, основанные на методах оптимизации

*Частотные подходы:*

* Интуиция
  + Слова, которые часто повторяются в документе, имеют отношение к основной теме документа
  + Предложения, которые часто повторяются в разных документах, выражают основную тему документа
* Проблемы: разнообразие лексики

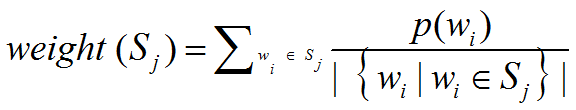
Например, *SumBasic*

* Основная идея: наиболее частотные слова исходного кластера с большей вероятностью должны оказаться в аннотации кластера:



n – число вхождений слова, N – общее число токенов

* Итеративный метод. На каждой итерации происходит расчет вероятностей слов, отбирается предложение с максимальной средней вероятностью слов:



* После отбора предложения происходит пересчет вероятностей для слов из отобранного предложения: вес слова резко снижается
* Таким образом достигается снижение повторов в аннотации

*Обогащение пословного представления: лексические цепочки*

* Использование информации об объектах и связях между ними, описанной в тезаурусах
  + Лексические цепочки – совокупности близких по смыслу слов
    - Возможность, учета частотности цепочки
    - Снижение повторяемости
    - Увеличение связности

*Использование tf.idf:*

* Частотное слово в исходном документе (наборе документов) может быть недостаточно тематическим,
* оно всегда частотно в данной коллекции
* Поэтому:
  + Tf – в исходном документе
  + Idf – в корпусе, из которого извлечен этот документ

*Графовые методы:*

* Вершины – предложения
* Дуги – сходство между предложениями
  + Косинусная мера
* Задается порог для сходства
* Центральность предложения – предложение при заданном пороге связано с максимальным числом других предложений. Это предложение и берется в аннотацию

*Методы на основе разных характеристик и машинного обучения*  
Характеристики для извлечения предложения:

* Частотные слова текста
* Слова получают веса,
* вес предложения зависит от весов слов
* Учет заголовка
* Вес предложения увеличивается, если в нем присутствуют слова заголовка
* Присутствие ключевых слов и конструкций
* Специальный список: *подчеркнем, основным результатом*
* Позиция в тексте
* Предложения в начале более важны
* Связность с предыдущим
* Новизна информации

1. **Метод MMR автоматического аннотирования**

08-02sum.ppt: 19-20

* Итеративный метод – локальная оптимизация
* На каждой итерации производится ранжирование предложений-кандидатов
* В итоговую аннотацию отбирается одно с самым высоким рангом
  + Максимизировать сходство с исходным документов (набором документов)
  + Минимизировать сходство с уже отобранными в аннотацию предложениями
* При использовании в контекстно-зависимой аннотации – максимизируется сходство с запросом

Пусть:

Q – запрос к системе

S – множество предложений кандидатов

s – рассматриваемое предложение кандидат

Е – множество выбранных предложений

Тогда:



1. **Метрика Rouge для тестирования автоматических аннотаций**

08-02sum.ppt: 46-47, 51

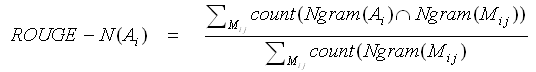
**ROUGE** или **Recall-Oriented Understudy for Gisting Evaluation** – набор метрик и комплекс программ для оценки автоматического аннотирования и машинного перевода текстов.

Основная идея – сравнение генерированного текста с “эталонным”, сделанным человеком.

Существуют различные формы метрики, сравнивающие:

* 1. n-граммы (ROUGE-N)
  2. минимальные общие подстроки (ROUGE-L и ROUGE-W)
  3. униграммы и биграммы (ROUGE-1 and ROUGE-2)

**Общая формула:**



*Ai* – оцениваемая обзорная аннотация *i*-того кластера.

*Mij* – ручные аннотации *i*-того кластера.

*Ngram(D)* – множество всех n-грамм из лемм соответствующего документа *D.*

*Плюсы и минусы:*

+ Малое участие человека, лёгкость применения

- Отсутствие оценки читабельности, результат не всегда идеален с точки зрения человека

1. **Метод пирамид для тестирования автоматических аннотаций**

08-02sum.ppt: 48-49, 51

* Разработан в 2005 году Колумбийским университетом.
* Эксперты выделяют из «эталонных» аннотаций «информационные единицы» - Summary Content Units (SCUs).
* Каждый SCU получает вес, равный количеству «эталонных» аннотаций, где она встречалась.
* Оценка – суммарный вес входящих SCU.
* Неоднократное вхождение SCU в автоматическую аннотацию не поощряется.

Пример:

* **Итоговый результат:**
* **Пример SCU:**

Мини-субмарина попала в ловушку под водой.

* + 1. мини-субмарина... была затоплена... на дне моря...
    2. маленькая... субмарина... затоплена... на глубине 625 футов.
    3. мини-субмарина попала в ловушку... ниже уровня моря.
    4. маленькая... субмарина... затоплена... на дне морском...

*Плюсы и минусы:*

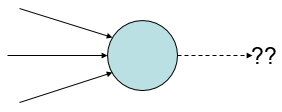
+ Наиболее объективная оценка содержания аннотации

- Отсутствие оценки читабельности, большое участие человека

1. **Что такое PageRank? Зачем нужен, как вычисляется**

09-01link.ppt: 16-39

*Вес PageRank:*

* Представим, что пользователь случайно бродит по страницам:
  + Начинает на случайной странице
  + На каждом шагу переходит на следующую по ссылку с равной вероятностью
* В пределе каждая страница получит рейтинг посещений – можно использовать как вес страницы
* Недостаточно, т.к. поскольку есть много тупиковых страниц.
  + В которых остановится случайное блуждание.
  + Это обессмысливает рассуждения о рейтинге посещений.  
    

*Телепортация*:

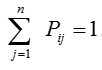
* В тупиковой странице – переход на случайную страницу
  + Для любой не тупиковой страницы - с вероятностью 10%, переходим на случайную страницу
  + С оставшейся вероятностью (90%) – переход по одной из исходящих ссылок ( с равной вероятностью)
  + 10% - параметр
* Теперь можно говорить о посещаемости страницы как о ее рейтинге

*Формула:*

PR(A) = d + (1-d)(PR(T1)/C(T1) +…+PR(Tn)/C(Tn))

* PR(Tn) – исходная значимость страницы
* C(Tn) – количество исходящих ссылок со страницы
* PR(Tn)/C(Tn) – значимость страницы равномерно распределяется по исходящим ссылкам и переносится в значимость страницы А по входящим в нее ссылкам
* d – например, 0.15 значимость страницы, без входящих ссылок (коэффициент телепортации)
* (1-d)(…) – 0.85

*Математическая основа - марковские цепи:*

* Цепь маркова состоит из n состояний, плюс nхn матрица вероятностей переходов P.
* На каждом шаге, мы в одном из состояний
* Для , элемент матрицы Pij означает вероятность перехода в j (следующее состояние), при условии, что сейчас состояние - i.
* 

*Алгоритм построения матрицы переходов:*

* Составить матрицу NxN по количеству страниц
* Для каждой страницы (строчки) исходящие ссылки обозначить 1 и поставить в соответствующих столбцах
* Нормализовать в каждой строке единицы, поделив на количество единиц
* Единицы умножить на коэффициент сглаживания (= (1-d), где d - коэффициент телепортации)
* Ко всем элементам строчки добавить коэффициент телепортации, поделенный на N (т.е. d/N)
* Если со страницы не было ссылок, то по всем столбцам ставим 1/N

*Итеративный подсчет PageRank:*

* X - случайный вектор начальных состояний
* Подсчитываем матрицу P (матрица переходов с телепортациями)
* Нужно выполнить умножение xP, затем …
* Достаточно быстро достигается сходимость

1. **Алгоритм HITS**

09-01link.ppt: 41-52

*Hyperlink-Induced Topic Search (HITS)*

* В ответ на запрос вместо упорядоченного множества страниц найдем два множества взаимосвязанных страниц:
  + ***Hub pages*** *– хорошие списки ссылок по теме*.
    - “Bob’s list of cancer-related links.”
  + ***Authority pages*** – часто упоминаются на страницах хабов
* Хорошо работает на широких тематических запросах

*Хабы и авторитеты:*

* Хорошая хаб-страница (посредник) для какой-то темы указывает на многие авторитетные страницы для этой темы
* Хорошая авторитетная страницы по теме указывается большим количеством хороших хабом по этой теме
* Для каждой страницы рекурсивно вычисляется ее значимость как посредника и как авторитета (автора)

**Основная схема:**

* Извлечь исходное множество (базовый набор) потенциально хороших хабов или авторитетов
* Из них формируем небольшой топ-лист хабов или авторитетов
  + Итеративный алгоритм

**Базовый набор:**

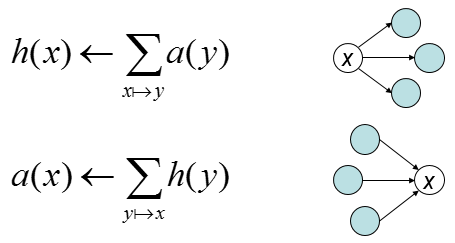
* Дан текстовый запрос (например, ***браузер***), получаем страницы, содержащие слово ***браузер.***
  + Это **корневой набор** страниц.
* Добавляем любую страницу, которая
  + указывает на страницу из корневого набора или
  + На которую есть ссылка со страницы корневого множества.
* Это **базовый** набор

**Разделение хабов и авторитетов:**

* Вычисляем для каждой страницы в базовом наборе hub score *h(x)* и authority score *a(x).*
* Инициализация: для всех *x, h(x)←1; a(x) ←1*;
* Итеративно пересчитываем *h(x), a(x)*;
* В результате итераций
  + Выдать страницы с наивысшими *h()* как топ-хабы
  + С наивысшими *a()* scores как топ-авторитеты

**Итеративный пересчет:**

* Повторяем следующий пересчет для всех *x*:



* Таким образом,
  + оценка авторитетности страницы вычисляется как сумма значений оценок посреднических страниц, которые указывают на эту страницу.
  + посредническая оценка страницы вычисляется как сумма значений оценок авторитетности страниц, на которых она ссылается.
* Рост значений авторитетности и посредника – необходима нормализация.
* Значения, полученные в результате этого процесса, в конечном итоге сходятся.
* Обычно требуется около 5 итераций

**Недостатки HITS:**

* Сдвиг темы (Topic drift)
  + Нерелевантные документы могут вызвать сдвиг темы
* Нерелеватные страницы на первых позициях выдачи
  + приводят к ошибочным результатам
* Взаимное усиление страниц, ссылающихся друг на друга
* Поисковая оптимизация SEO: создание искусственного множества ссылок

1. **Особенности использования кликов пользователя в качестве фидбека от пользователя. Каскадная модель при обработке кликов.**

09-02logs.ppt: 37-46

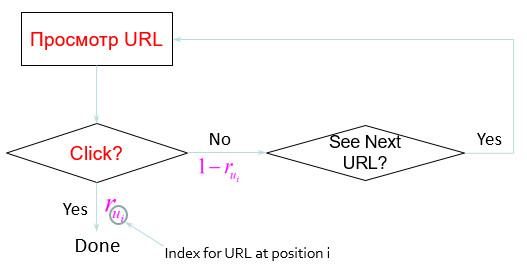
* Более высокие позиции получают больше кликов пользователя, чем более низкие позиции (eye fixation).
* Это справедливо, даже есть выдачу переставить наоборот
* “Клики информативны, но смещены (biased)”.

*Гипотеза о «наблюдении»:*

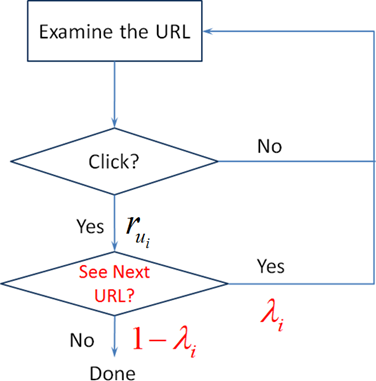
* Документ должен быть прочитан перед кликом.
* Условная вероятность клика после прочтения зависит от релевантности документа
* Вероятность клика делится на две части
  + Глобальный компонент: вероятность увидеть – зависит от позиции документа
  + Локальный компонент: зависит от пары (запрос, документ)
* Это основа любой современной модели

*Каскадная модель:*

* Первый документ всегда просматривается
* Дальше модель Маркова
  + Просмотр на позиции i+1 зависит от просмотра и клика на позиции i
* Просмотр идет линейно
* Объединяем две гипотезы:
* Формальная спецификация модели:
  + - Гипотеза просмотра
  + – Каскадная гипотеза
  + – Моделирование клика
* Блок схема поведения пользователя:



Обобщение для 1+ кликов:



1. **Классификация запросов по цели. Зачем нужна. Особенности обработки разных типов запросов**

09-02logs.ppt: 11-16

1. Навигационные запросы

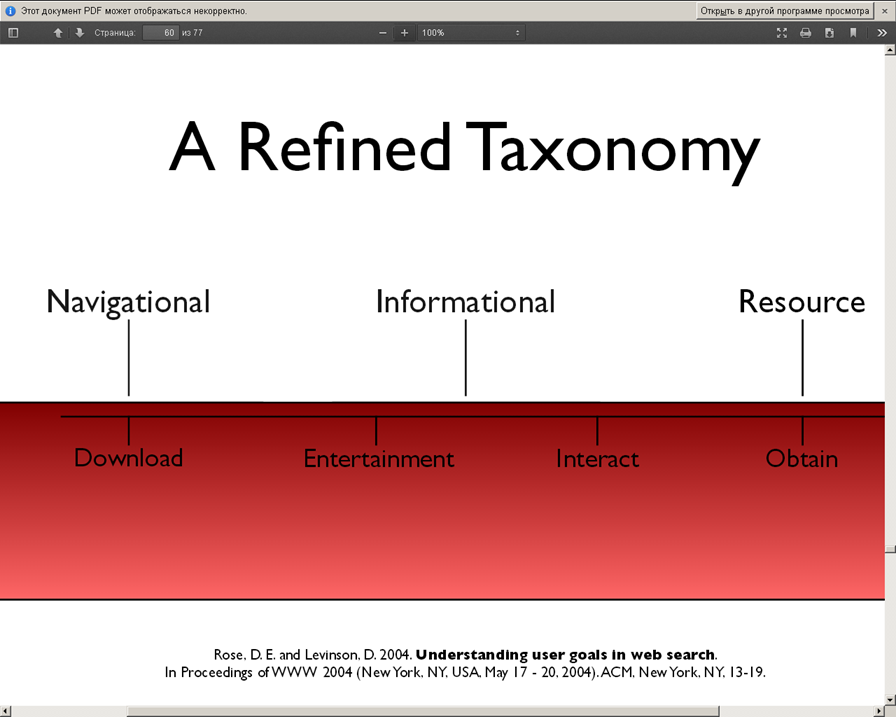
* Поиск конкретной страницы (american airlines, google, cnn)
* 20-25% от всех запросов

1. Информационные запросы

* Поиск информации (Transfinite Numbers, Escher)
* 40-45% от всех запросов

1. Транзакционные запросы

* Покупка чего-то (Hotels Saint Petersburg, Tickets for Hermitage)
* 30-35% от всех запросов



1. **Вероятностная модель информационного поиска: основная идея, различие с векторной моделью**

10-01prob.ppt: 2-14

Вероятностные модели

* + Попытка найти математическое обоснование
  + Классическая модель BIM (Binary Independence Model)
  + Модель BM25
  + Языковая модель информационного поиска

Вероятностная модель пытается оценить вероятность, что пользователь оценит документ dj какрелевантный посредством отношения

*P(dj relevant to q)/P(dj nonrelevant to q)*

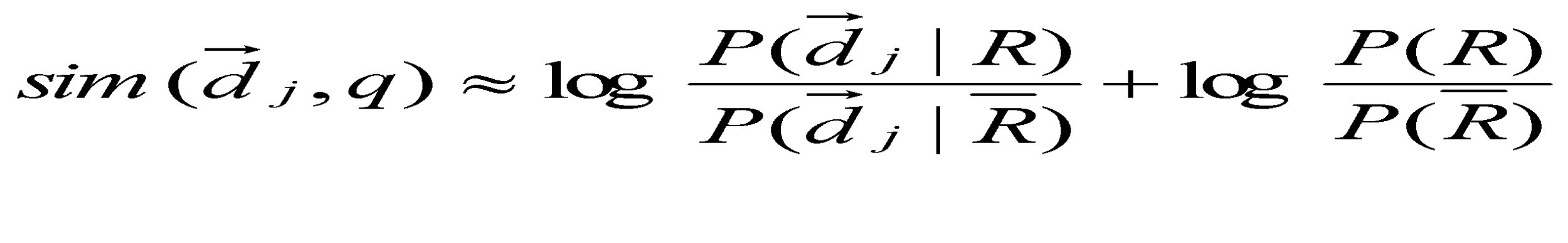
* Определения
  + Все веса слов - бинарные, т.е. *wi,j* ∈ {0,1}
  + Пусть *R* – множество документов, про которые известно, что они релевантны запросу *q*
  + Пусть - оставшиеся документы
  + - это вероятность, что документ *dj* релевантен запросу *q*
  + - вероятность, что документ *dj* нерелевантен запросу *q*
* *Сходство sim(dj,q) документа dj с запросом q определяется как отношение*



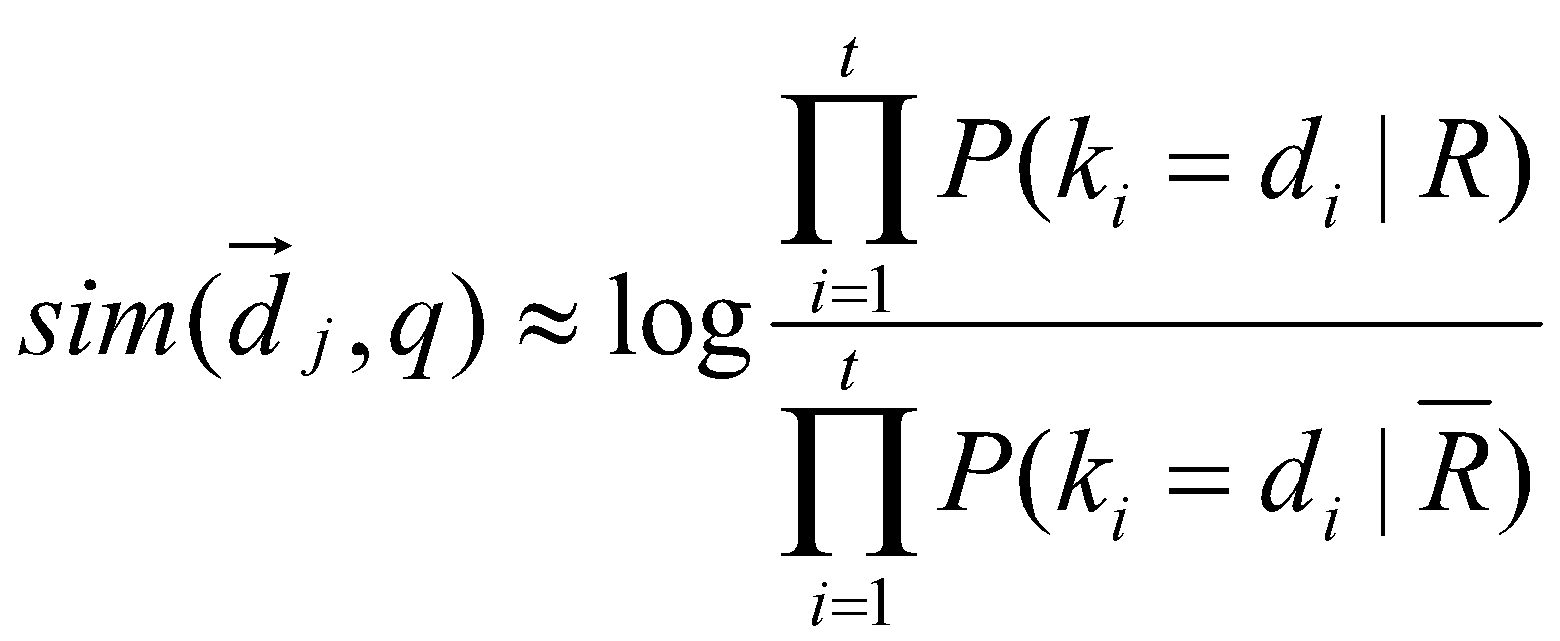
* Используя правило Байеса



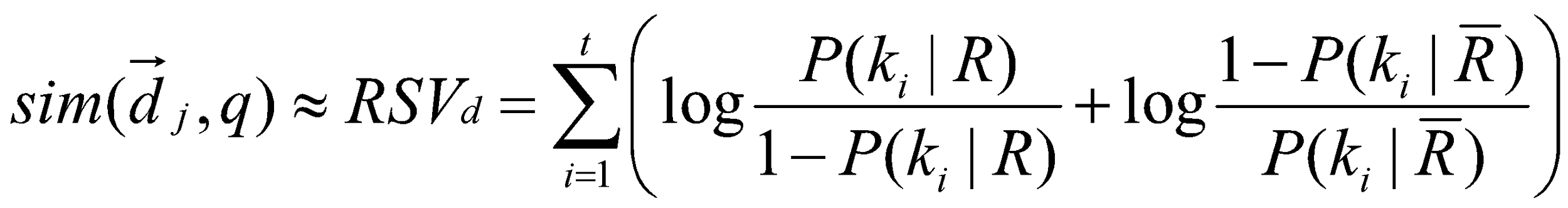
* + *P(R)* – вероятность случайно выбрать релевантный документ в коллекции
  + - вероятность случайно выбрать документ *dj* из множества *R* релевантных документов



Предполагая независимость слов и :



* + P (*ki* |*R*) – это вероятность, что слово *ki* присутствует в документе из случайно выбранного документа в множестве релевантных документов R
  + - это вероятность того, что *ki* не присутствует в документе, случайно выбранного из множества релевантных документов R
  + В результате оказывается:



**Различие между векторными и вероятностными:**

* + - В обоих случаях поисковая система строится похожим образом
    - Различия: в вероятностном информационном поиске сходство между запросов и документом считается не косинусной мерой и tf-idf в векторном пространстве, а несколько другой формулой, мотивированной теорией вероятности

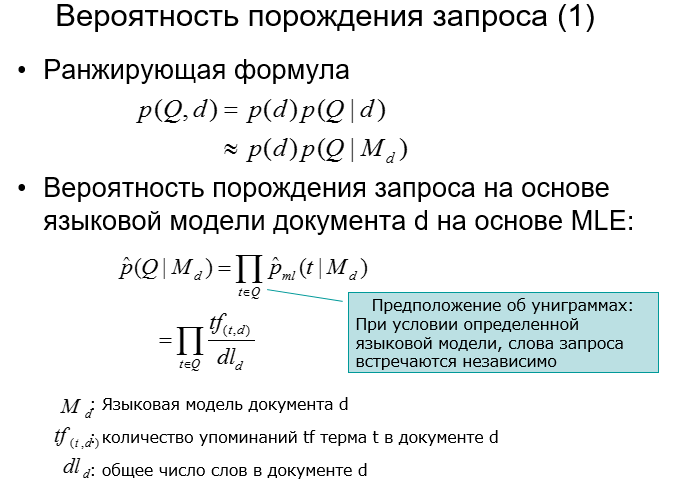
1. **Языковые статистические модели. Сглаживание. Где применяются**

10-01prob.ppt: 21-28

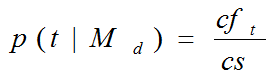
* Статистические модели: определение вероятности предложений, последовательностей слов
* Языковая модель – математическая модель, которая вычисляет вероятность последовательности слов или условную вероятность следования слова в контексте

Языковая вероятностная модель:

* Статистическая модель порождения текста определяет вероятности строк для данного языка
* Моделируют вероятность порождения строк в языке



*Нехватка данных:*

* Нулевая вероятность 
* Модель не должна приписывать нулевую вероятность запросу, который содержит слово, отсутствующее в документе: сглаживание
* Общий подход в информационном поиске
* Не встречавшееся в документе слово возможно, но не должно быть более вероятно, чем случайно встреченное в коллекции
* Если  то 

где cft – количество вхождений слова в коллекцию

cs – количество слов в коллекции

*Сглаживание:*

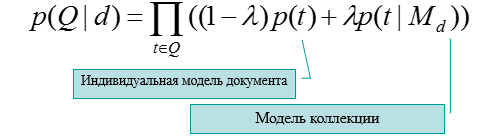
* Необходимо уйти от нулевых вероятностей => техники сглаживания
* Уже упоминались при обсуждении байесовской классификации
* Методы
* Ранее использовалось: добавить 1, ½ или к частотам упоминания
* Здесь: смешивание статистики от документа и коллекции

1. **Языковая модель информационного поиска**

10-01prob.ppt: 24-29

* Трактуем каждый запрос как случайный процесс
* Подход:
* Насчитывает языковые модели для каждого документа
* Выводим вероятность порождения запроса на основе модели каждого документа
* Ранжируем документы в соответствии с этими вероятностями
* Обычно используются униграммы

**Общая формулировка языковой модели для информационного поиска**



* Пользователь имеет документ в уме и порождает запрос из этого документа
* Равенство выражает вероятность, что документ, который имел в виду пользователь именно этот
* Соединяет вероятность слова в документе и вероятность в коллекции целиком
* Необходимо корректный подбор
* Высокое значение lambda делает поиск более похожий на булевский (требуется упоминание всех слов в запросе), больше подходит для коротких запросов
* Низкое значение – больше подходит для длинных запросов
* Можно настраивать для оптимизации качества поиска

1. **Методы приблизительного вычисления сходства документов в реальных поисковых системах**

10-02fullsysst.ppt: 4-7(Думаю 2-3 не надо)

Неточная векторная модель: общий подход

* Найти А кандидатов K(лучшие)<A<<N
  + А не обязательно содержит все К лучших документов, но содержит много документов из K
* Выдать К лучших документов из А
* Методы:
  + Рассмотреть только слова с высокими idf
  + Рассмотреть только документы с большим количеством терминов запроса

Слова (термы) с высоким idf:

* Пример: “Над пропастью во ржи” – рассмотрим только слова пропасть и рожь, так как интуитивно над и во мало вносят в вес документа и не сильно изменят ранжирование
* Преимущества подхода:
  + Частотные слова содержатся в большом количестве документов
  + Резкое сокращение просмотра документов

Документы с большим количеством терминов запроса

* Либо любой документ, содержащий хотя бы один терм запроса – кандидат в выдачу
* Либо (для многословных запросов) считаем только документы, в которых содержится несколько слов запроса (например 3 из 4)
  + Называют мягкой конъюнкцией, можно часто встретить в интернет поиске

1. **Обработка фразовых запросов и запросов с указанием близости слов в поисковых системах**

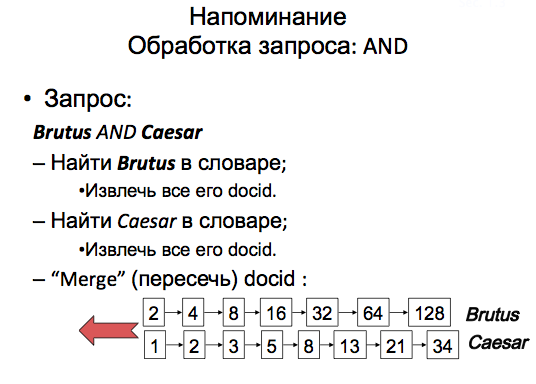
10-02fullsysst.ppt:18-28

Фразовые запросы

* Когда мы хотим сохранить порядок слов в запросе
* Например: «Stanford University» (в кавычках) (“I went to university of Stanford не подходит”)
* Много запросов пользователей внутри себя содержат фразовые запросы
* Не достаточно хранить только матрицу (индекс): терм-документ
* Подходы:
  + Биграммный индекс
  + Позиционный индекс
    - Для каждого терма храним его позиции в документе
    - Для сопоставления используется алгоритм слияния на уровне документа

Биграммный индекс

* Нарезаем тексты на пары слов, строим на парах индекс (Например, “I went”, “went to”…)
* Длинные запросы также нарезаем на пары слов
* Получаем запрос вида – Stanford university AND university palo AND palo alto



* Тем не менее могут встретиться документы, которые содержат пары слов, но не содержат фразу
* Проблемы:
  + Сверхбольшой индекс
  + Проблемы со случайным вхождением слов внутрь биграммы
  + Не является стандартным решением

Позиционный индекс:

* Храним позиции терма в документе
* Для сопоставления используется алгоритм merge на уровне документа

Запросы с указанием близости

* Пример запроса “Москва /3 университет
* Для таких запросов может использоваться только позиционный индекс, биграммный не может
* Позиционный индекс сейчас стандартно используется для обработки запросов на близость и фразовых запросов
* Возможно комбинирование биграммных и позиционных индексов

*Близость слов запроса:*

* Запросы на естественном языке: набор терм, набиваемых в поисковую строку
* Пользователь предпочитает, чтобы термины запроса встречались недалеко друг от друга
* Пусть w – минимальное окно, содержащее все слова запроса, например, для запроса “увольнение директора” минимальное окно в документе “Директора ждало неожиданное увольнение” – 4 слова
* Как учесть в скоринговой функции?

*Разборщик запросов (колдунщик)*

Текстовый запрос пользователя может исполниться посредством нескольких запросов к системе, например, запрос: “повышение оплаты труда”

* Исполняем запрос как фразовый запрос
* Если <K документов содержат фразу “повышение оплаты труда”, то исполняются два фразовых запроса “повышение оплаты” и “оплата труда”
* Если все еще меньше К документов, то запрос на векторное пространство “повышение оплаты труда”
* Эти операции исполняет разборщик запросов
* Скоринговая функция должна учитывать разные факторы, включая минимальное окно, в котором содержится запрос

1. **Позиционный индекс в поисковой системе. Зачем нужен, как обрабатывается**

10-02fullsysst.ppt: 23-25

* Храним позиции терма в документе  
  <be: 993427;  
  1 док.: 7, 18, 33, 72, 86, 231…  
  2 док.: 3, 149…  
  4 док.: 17, 191, 291, 430, 434  
  5 док.: 363, 367
* В каком документе может содержаться фраза «to be or not to be»?
* Для сопоставления используется «merging алгоритм», на уровне документа

Обработка фразовых запросов:

* Извлекаем инвертированный индекс для слов to, be, or, not  
  to:

2:1,17,74,222,551; **4:8,16,190,429,433**; 7:13,23,191; ...  
be:  
1:17,19; **4:17,191,291,430,434**; 5:14,19,101; …

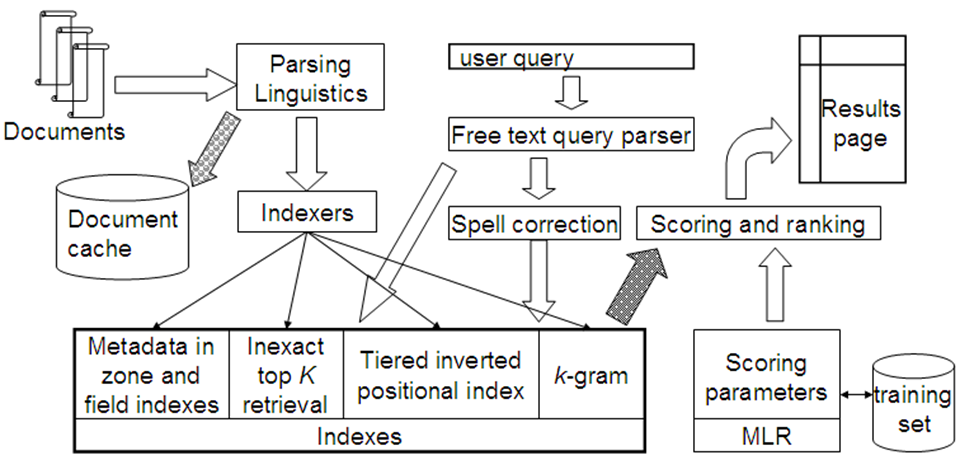
* Москва /3 университет  
  Только позиционный индекс может использоваться для такой обработки, биграммный не может
* Позиционный индекс сейчас стандартно используется для обработки запросов на близость и фразовых запросов  
  Возможно комбинирование биграммных и позиционных индексов

1. **Какие факторы помимо веса tf.idf учитываются в поисковых моделях, как создаются многофакторные модели в информационно поиске**

10-02fullsysst.ppt: 29-45

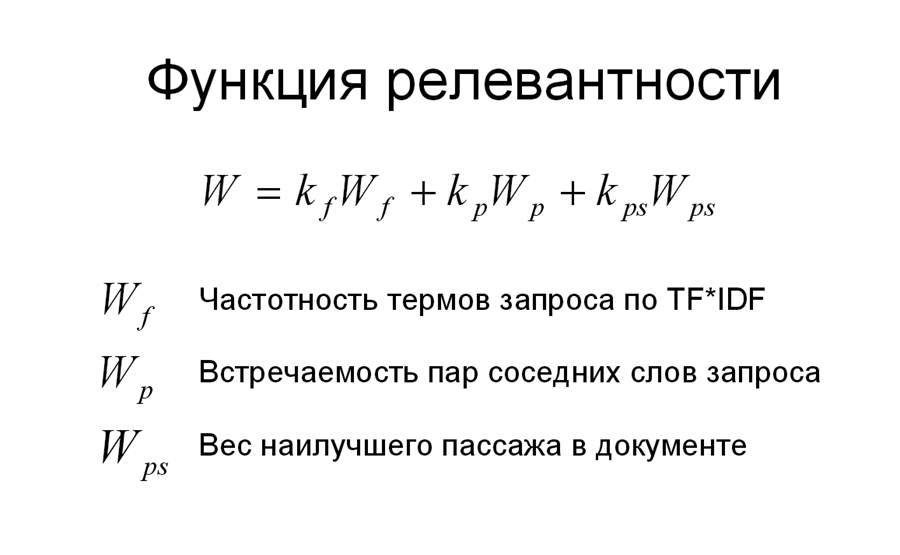
Много разных факторов:

* Скалярное произведение
* Близость слов запроса
* Ссылки
* Соответствие тематики запроса и документа



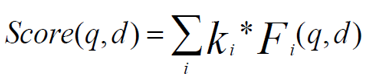
*Mail.Ru (2005):*

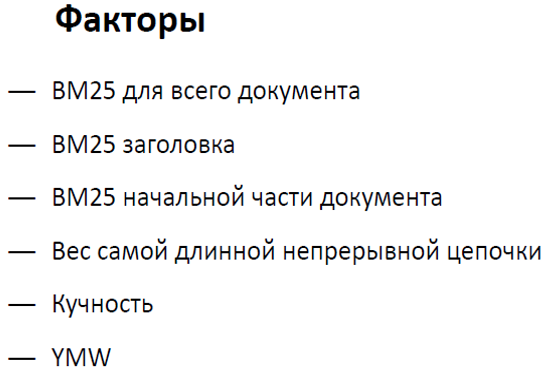
Комплексная функция релевантности

* Частота термов
* Совместная встречаемость термов
* Зоны документа
* Неполное вхождение термов в запрос
* Особая обработка длинных документов  
    
  

Основные характеристики пассажей:   
- полнота, длина, порядок слов, зона документа, близость к началу.

*Алгоритм Сафронова*:



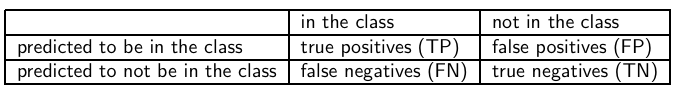
 Yandex Minimum Window

Алгоритмы яндекса:

1. Магадан
2. Арзамас
3. Снежинск

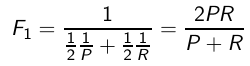
**Задачи на следующие темы:**

1. **Байесовская модель классификации текстов**
2. **Макро- и микро- усреднение при оценке качества автоматической рубрикации**



*P* = *TP* / ( *TP* + *FP*)

*R* = *TP* / ( *TP* + *FN*)



* + Посчитали меру оценки (*F*1) for one class.
  + Как агрегировать оценки F1 для многих классов.
  + Macroaveraging - макроусреднение
    - Посчитать *F*1 для каждого из *C* классов
    - Среднее арифметическое для этих *C* чисел
  + Microaveraging - микроусреднение
    - Посчитать TP, FP, FN для каждого из *C* classes
    - Суммировать эти С чисел для каждого показателя
    - Посчитать *F*1 для суммированных TP, FP, FN

**Задача:**

Система рубрикации должна классифицировать поток документов по двум рубрикам.  
Эксперт отнес к первой рубрике 75 документов, ко второй рубрике – 50 документов.  
Система отнесла:  
- к первой рубрике 100 документов, из них 50 правильно.  
- ко второй рубрике 40 документов, из них 30 правильно.  
Найти макро-характеристики качества классификации (точность, полноту, F-меру) - и микро- характеристики (точность, полноту, F-меру).

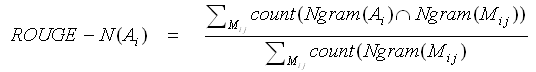
**Решение:**

Макроусреднение:

1. 1 рубрика:
   1. P = 50 / 100 = 1/2
   2. R = 50 / 75 = 2/3
   3. F = 2 \* 1/2 \* 2/3 / (1/2+2/3) = 2/3 / 7/6 = 4/7
2. 2 рубрика:
   1. P = 30 / 40 = 3/4
   2. R = 30 / 50 = 3/5
   3. F = 2 \* 3/4 \* 3/5 / (3/4+3/5) = 2/3
3. Усреднение:
   1. P = (1/2+3/4)/2 = 5/8
   2. R = (2/3+3/5)/2 = 19/30
   3. F = (4/7+2/3)/2 = 13/21

Микроусреднение:

1. TP = 50 + 30 = 80
2. FP = 50 + 10 = 60
3. FN = 25 + 20 = 45
4. Усреднение
   1. P = 80 / 140 = 4/7
   2. R = 80 / 125 = 16/25
   3. F = 2 \* 4/7 \* 16/25 / (4/7+16/25) = 32/53
5. **Кластеризация текстов**
6. **Метрика Rouge оценки качества автоматического аннотирования**

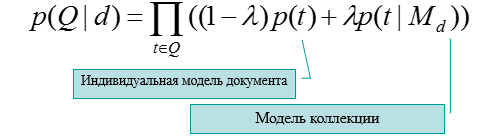


*Ai* – оцениваемая обзорная аннотация *i*-того кластера.

*Mij* – ручные аннотации *i*-того кластера (ЭКСПЕРТ).

*Ngram(D)* – множество всех n-грамм из лемм соответствующего документа *D.*

1. **Вычисление близости документа к запросу по языковой модели информационного поиска**



**Задача:**

Запрос к поисковой системе состоит из двух слов: q = (a b)  
В коллекции имеются следующие документы:  
   
d1 = a b c d  
d2 = a a a   
d3 = b b c   
d4 = a b b c  
   
Других документов в коллекции нет.   
Примените языковую модель к этой коллекции.

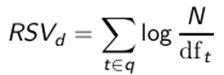
**Решение:**

Итоговое ранжирование:

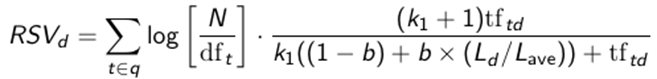
1. **Вычисление PageRank**
2. **Объяснение формулы BM25 (вероятностный информационный поиск)**

10-01prob.ppt: 15-17

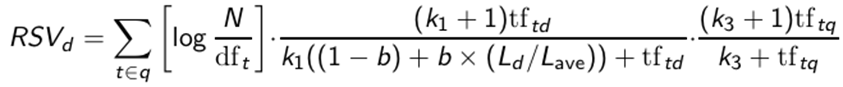
* Вероятностная модель BIM была изначально создана для поиска по записям в коротких каталогах сопоставимой длины – и работала прилично в этих условиях
* Для современного полнотекстового поиска, модель должна учитывать частоту термина в документе и длину документа
* BestMatch25 (BM25 или Okapi), развитие модели BIM, учитывает эти величины
* С 1994 до наших дней, модель BM25 – это одна из наиболее распространенных и устойчивых моделей информационного поиска
* Простейшая форма веса для документа d – это просто суммирование idf слов запроса, которые присутствуют в этом документе:



Это формула «исправляется» учетом частоты слова в документа и длины документа:



* : частота слова в документе d
* : длина документа d (средняя длина документа в коллекции)
* : параметр, контролирующий учет частоты слова
* : параметр, контролирующий учет длины документа
* Если запрос длинный, то можно учитывать похожее взвешивание для слов запроса



* : частота слова в запросе q
* : параметр, контролирующий частоту термина в запросе
* Нет нормализации запроса по длине (поскольку поиск делается для фиксированного запроса)
* Параметры нужно настраивать на коллекции
* Если оптимизация не выполнялась, то в экспериментах получено, что величины и должны иметь значения в промежутке [1.2, 2], b = 0.75