Задачи информационного поиска в коллекциях текстовых документов: классические и современные подходы

Научный руководитель: проф. д.т.н Миркин Б.Г.
Аспирант: Фролов Д.С.

Содержание

- Задача информационного поиска
- Агрегированное представление документов для задач информационного поиска
- Текущее состояние области и результаты докладчика

Информационный поиск (Information retrieval) - наука о методах поиска неструктурированной информации, удовлетворяющей заданным параметрам. В широком смысле к информационному поиску относят и другие задачи анализа неструктурированной информации.

Типичная задача - поиск документов в коллекции, релевантных заданному запросу. В связи с растущим объемом текстовых документов различные задачи анализа текстов и, прежде всего, задачи их поиска, все чаще встречаются на практике. Они известны в разных постановках и широко освещены в литературе. Актуальность задачи автоматизации поиска документов в доказательстве не нуждается.

Первые упоминания information retrieval в научной литературе - начало 1950-х

Бурное развитие - с конца 1980-х в связи с появлением поисковых систем и Интернета

Классическая постановка задачи

Дано множество (коллекция) документов и множество поисковых запросов. Требуется для каждого запроса предоставить множество наиболее релевантных ему документов из коллекции.

Релевантность - степень смыслового соответствия документа поисковому запросу

Ранжирование - процесс назначения степени релевантности документам коллекции и их последующая сортировка

Поиск подстроки в строке

Существует масса алгоритмов поиска подстрок в строках, а также подстрок, удовлетворяющих регулярным выражениям

Ввиду объемов коллекций документов, рассматриваемых в задачах информационного поиска, этот подход едва ли применим на практике

Булев поиск

Исторически первый метод информационного поиска

Модель булева поиска: обрабатывается запрос, имеющий вид булева выражения: запроса, в котором термины (признаки документов) используются в сочетании с операциями: AND, OR, NOT

Использовался в коммерческих системах вплоть до начала 2000-х годов

Булев поиск

	GNU Emacs: введение	linux.org.ru	Введение в администрировани е систем
Emacs	1	1	0
FreeBSD	0	1	1
Настройка	1	0	1
Форум	0	1	0

Emacs AND FreeBSD AND NOT Настройка

Индексирование и ранжирование

Принципиально новое направление (конец 1980-х годов)

Серия публикаций D. Cutting, J. Pedersen (“Space Optimizations for Total Ranking” и др.)

Принципы, изложенные в серии, заложили основу для создания огромного количества свободных и проприетарных поисковых движков (Lucene, ElasticSearch и т.д.), развивающихся и по сей день

Индексирование

Классическое обратное индексирование:

1. Выделение признаков и весов их вхождения из документов
2. Объединение полученных признаков и создание указателей от признаков к документам, где они были обнаружены

Документы	Признаки	Обратный индекс
1. A B C 2. C D E 3. D E A	A, B, C, D, E	A: [1, 3] B: [2] C: [2] D: [2, 3] E: [2, 3]

Ранжирование

Важным является процесс назначения весов вхождения признаков в документы и последующей оценки релевантности запросу

Веса признаков:

1. Частота встречаемости в документе (TF)
2. TF-IDF
3. ...

Функции ранжирования

Существует огромное множество функций ранжирования

Задан запрос w = $w_1 \dots w_N$, определить релевантность документа d этому запросу

1. $S(d) = f(w_1, d) + \dots + f(w_N, d)$ - сумма весов признаков запроса в документе
2. BM25

$$score(q,d) = \sum_{i=1}^{|q|} idf(q_i) \cdot \frac{tf(q_i,d) \cdot (k_1+1)}{tf(q_i,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})}$$

3. ...

Задача нечеткого поиска

В случае, когда признак содержит неточность, он не будет найден в индексе. В этом случае решается задача нечеткого поиска.

Некоторые методы нечеткого поиска:

1. Расширение множества признаков
2. Фрагментное индексирование (N-граммы)
3. Сигнатурные хэширования
4. Метрические деревья (Burkhard-Keller и др.)

Расширение множества признаков

Исторически первый и самый простой метод нечеткого поиска

1. Простое перечисление всех вариантов использования признака
   emacs : [emacs, emax, imacs, emaks, ...]

2. Формирование признаков с “джокером”:
   emacs : [$macs, e$acs, ...]

3. Алгоритмы семейства Soundex:
   emacs -> [e=i, cs=x] : [imax]

N-граммные индексы

Признаки в индексе заменяются на цепочки N подряд идущих символов (фрагменты)

N = 3, 4

При очень большом количестве фрагментов множество отсекается по квантилю

Хэширование по сигнатуре

Метод описан в публикациях Л. М. Бойцова (“Поиск по сходству в документальных базах данных: хеширование по сигнатуре”, 2001 и др.)

E	_ _ _ 1 _ _ _
M, K	_ 1 _ _ _ _ _
C, S	_ _ _ _ _ _ 1
...	...

EMACS -> 0101001

Агрегированное представление

Одно из направлений развития - переход от множеств исходных документов к агрегированному представлению.

Существует множество подходов, среди которых:

- Вероятностное тематическое моделирование (вероятностное латентно-семантическое индексирование (PLSI), скрытое размещение Дирихле (LDA) и модификации) - используется признаковое описание документов
- Метод аннотированного суффиксного дерева (АСД) - используется фрагментное описание документов
- Хеширование - как правило, фрагментное или символьное описание

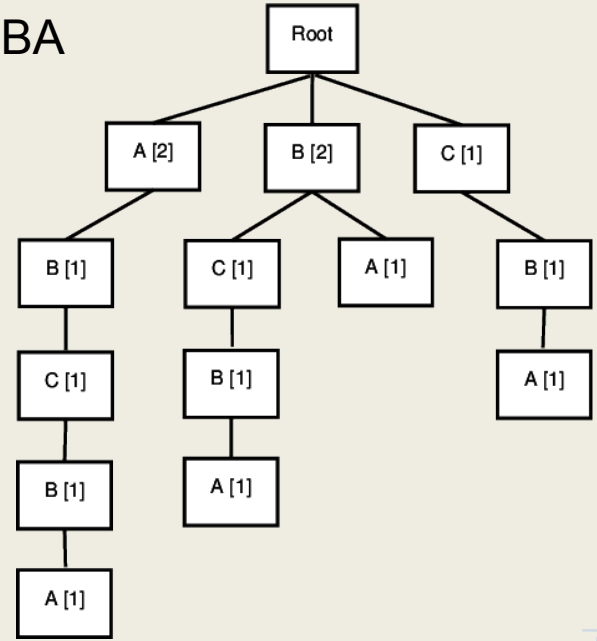
- Агрегированное представление документов описано и использовано (например, для задач построения таксономии) в работах Б. Г. Миркина, Р. Пампапати, Е. Л. Черняк

- Агрегированные представления коллекций документов (PLSI и LDA) описаны, например, в работах К.В. Воронцова, а также в исследованиях ВЦ РАН и ИСП РАН (А. Коршунов, А. Гомзин и др.)

Аннотированное суффиксное дерево (АСД)

Структура данных, позволяющая хранить фрагменты текста (суффиксы) вместе с их частотами

Пример для строки ABCBA



Алгоритм построения АСД

[Демонстрация]

Вычисление степени вхождения строк в АСД

Условная вероятность p(u) узла u (f(u) - его аннотация):

$$p(u) = \frac{f(u)}{\sum_{v \in T: ancestor(v) = ancestor(u)} f(v)}$$

Для строки $x = x_1 \ldots x_N$ степень вхождения S(x, T) в АСД T определяется следующим образом:

$$S(x,T) = \frac{1}{N} \sum_{k=0}^{N} s(x_k, T)$$

$$s(x_k, T) = \frac{1}{k_{max}} \sum_{i=0}^{k_{max}} p(x_i^k)$$

k_{max} - длина наибольшего совпадения суффикса с АСД

Ускорение метода АСД

Если каждый документ коллекции представлен АСД, можно проверять степень вхождения строки-запроса не во все документы, а только в специальным образом отобранные кандидаты

Обратный индекс по фрагментам f_i :

$$\begin{cases} f_1 : \lfloor n_{11,}\ldots,n_{1m_1} \rfloor \\ f_2 : \lfloor n_{21,}\ldots,n_{2m_2} \rfloor \\ \ldots \\ f_K : \lfloor n_{K1},\ldots,n_{Km_K} \rfloor \end{cases}$$

Алгоритм метода поиска на основе АСД

Итоговый алгоритм поиска релевантных документов по заданному запросу:

1. Разделить поисковый запрос на фрагменты и выбрать “документы-кандидаты” из хэш-таблицы.

2. Рассчитать степени вхождения поискового запроса в АСД, построенные для выбранных документов.

3. Отсортировать полученные значения по релевантности.

Текущие исследования в области АСД

1. M. Levene, B. Mirkin, R. Pampapathi “A suffix tree approach to anti-spam email filtering” - 2006
2. Черняк Е. Л., Миркин Б. Г. Использование ресурсов Интернета для построения таксономии - 2013
3. Черняк Е. Л., Миркин Б. Г. Меры релевантности строка-текст в проблеме рубрикации научных статей - 2013
4. Mirkin B. G., Chernyak E. L. An AST method for scoring string-to-text similiarity in semantic text analysis - 2014
5. Chernyak E. L., Mirkin B. G. “Refining a Taxonomy by Using Annotated Suffix Trees and Wikipedia Resources” - 2015

Экспериментальная апробация - тестовая коллекция №1

Использовалась для измерения метрик качества

Документы: данные каталогов товаров интернет-магазина Ozon.ru, раздел “книги” (90 тыс. документов)
3 группы запросов, получены с помощью сервиса Yandex.Wordstat

1. Явные	*“вожди Атлантиды”, “Властелин Колец”*
2. Название подкатегории	*“русское фэнтези”, “мистика”*
3. “Неявные”	*“книги толкена”, “хобит книга купить”*

Метрики качества поиска

Исследованы следующие качественные характеристики:

Точечные оценки:

- точность на уровне N документов и полнота

Графические характеристики:

- TREC11 и ROC кривые

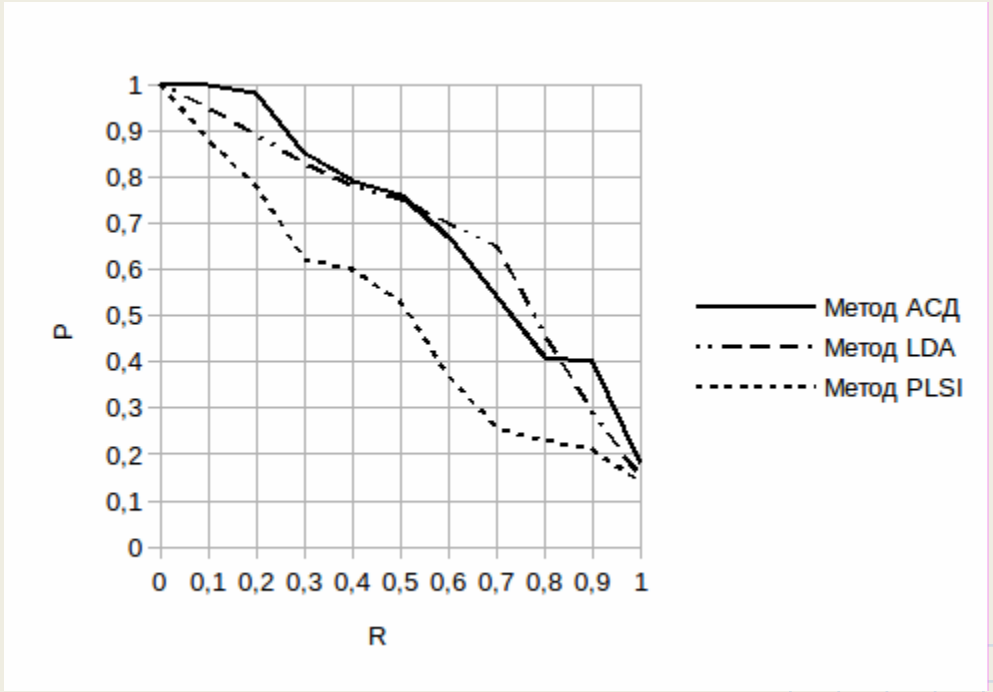
Исследование метрик качества

Средняя точность исследуемых методов на уровне 10 документов

Группа запросов	АСД	PLSI	LDA	LDA с биграммами	полнотекст. поиск MongoDB
№1	0.85	0.70	0.85	0.86	0.5
№2	0.84	0.68	0.81	0.86	0.5
№3	0.79	0.41	0.43	0.55	0.2
Среднее	0.82	0.56	0.70	0.76	0.4

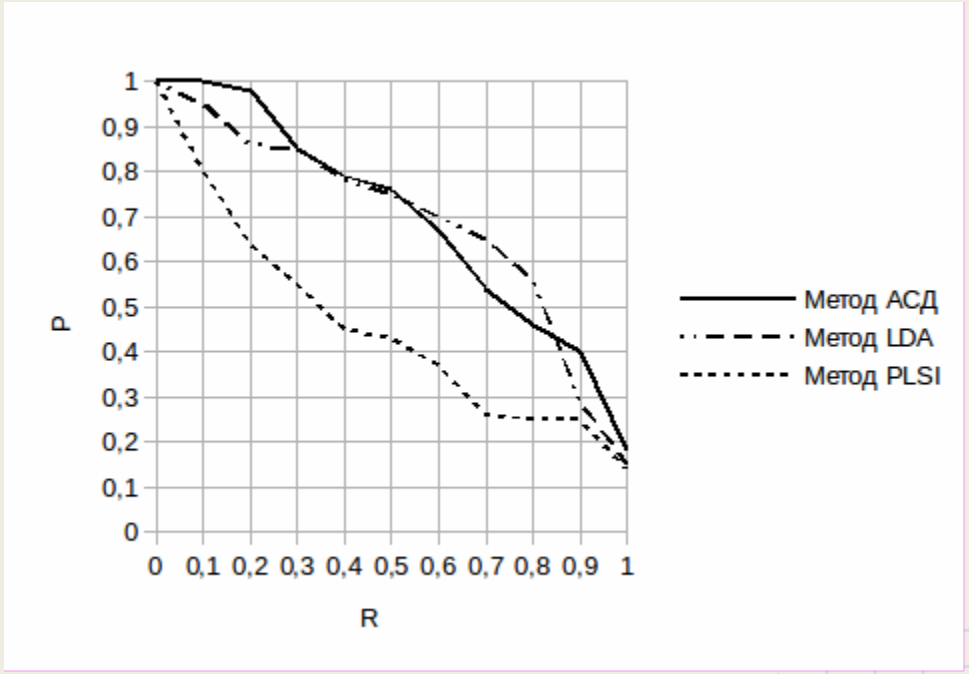
Кривые TREC11

Группа запросов №1 (“явные” запросы)



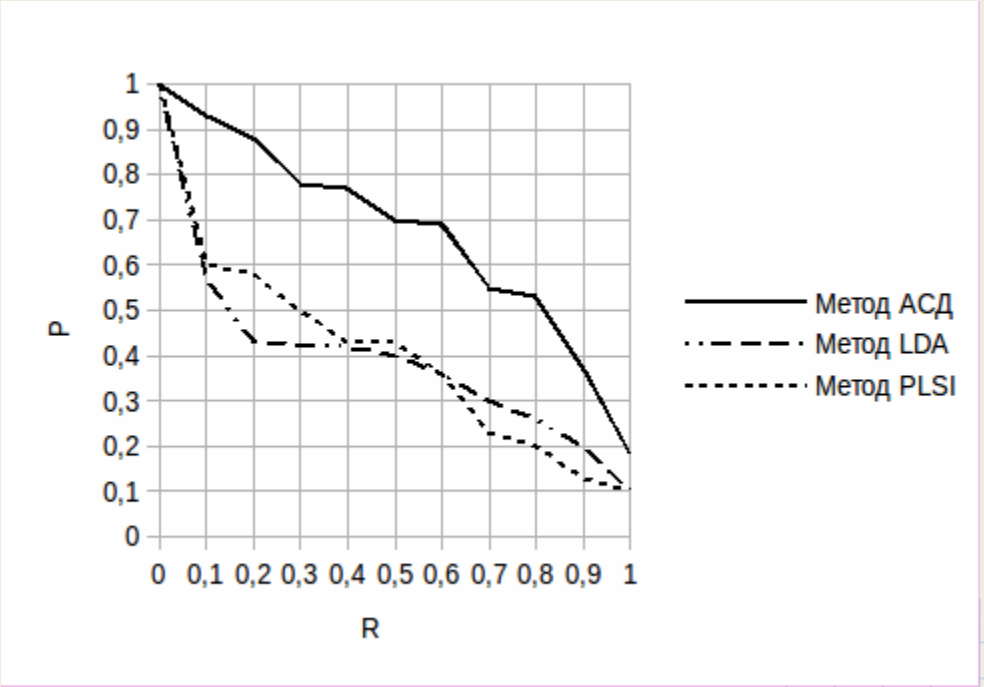
Кривые TREC11

Группа запросов №2 (название подкатегории)



Кривые TREC11

Группа запросов №3 (“неточные” запросы)



Время выполнения запроса

Среднее время поиска для исследуемых методов (с)

АСД	PLSI	LDA	LDA с биграммами	полнотекст. поиск MongoDB
0.43	0.23	0.25	0.29	8.55

Тестовая коллекция №2

Использовалась для измерения метрик производительности (время выполнения запросов)

Документы: 2000 больших (15-40 тыс. символов) статей сайта habrahabr.ru

Запросы: фрагменты документов

Время выполнения запроса

Среднее время поиска для исследуемых методов (с)

АСД	PLSI	LDA	LDA с биграммами	полнотекст. поиск MongoDB
0.15	0.04	0.05	0.08	3.09

Время выполнения запроса

Влияние длины строки в АСД на время обработки запроса (с)

АСД, 1 слово на строку	АСД, 2-3 слова на строку	АСД, 5-6 слов на строку
0.12	0.15	0.21

Текущие результаты по работе

- Подготовлена статья “Метод аннотированного суффиксного дерева для агрегированного представления текста в задачах поиска в коллекциях текстовых документов” (проходит рецензирование в “Бизнес-Информатике”)
- Подготовлена статья “Aggregate Text Representation for Information Retrieval in Collections of Text Documents” для доклада на конференции RuSSIR 2015 (август 2015, участие подтверждено комитетом по рассмотрению заявок, уточняется форма участия)

Дальнейшие задачи исследования

- модифицировать и улучшить метод, основанный на АСД, применив идеи оптимизации индексирования и безындексный подход
- разработать распределенные варианты этих алгоритмов
- адаптировать разработанные алгоритмы для работы с динамически изменяющимися коллекциями документов
- провести экспериментальное сравнение предложенных методов с существующими аналогами

Спасибо!