УДК: 025.4.03

Применение методов кластеризации для анализа неиндексируемых интернет-ресурсов

М.П. Сычев [1], А.В. Астрахов [1],
Д.И. Правиков [1], О.И. Тягунков [1]

[1] МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Представлены результаты сравнительного анализа двух алгоритмов кластерного анализа Lingo и STC. В качестве корпуса документов для оценки возможности кластеризации использован набор документов, полученных в ходе мониторинга сайтов определенной тематической направленности. Показано, что для корпуса документов указанной тематики алгоритм Lingo обеспечивает более высокое качество кластеризации.

E-mail: zi@bmstu.ru

Ключевые слова: *информационный поиск, извлечение знаний, кластеризация, сингулярное разложение, суффиксное дерево.*

Введение. Ввиду стремительного развития информационных ресурсов сети Интернет, их активного использования в различных областях деятельности человека объемы информации, которую необходимо обрабатывать, возросли многократно, что привело к бурному развитию технологии распределенного хранения сверхбольших объемов данных. Вместе с тем с ростом объемов информации должны быть усовершенствованы методы извлечения этой информации и представления ее пользователю [1 — 4]. Одним из направлений подобной обработки является кластеризация, которая призвана решить следующие задачи:

– разбиение исходного множества на группы схожих объектов и предоставление возможности работы с каждой группой в отдельности;

– сокращение объема хранимых данных путем оставления по одному представителю от каждого кластера;

– выделение нетипичных объектов, не подходящих ни к одному из кластеров (так называемые аномалии).

Особенности информационно-поисковых систем (ИПС) для обработки неиндексированных сайтов. Несмотря на то что для поиска информации в сети Интернет существуют поисковые системы Яндекс, Google, Yahoo, Mail, Rambler и др., значительный объем данных содержится в так называемом сером, темном или глубинном Интернете — сайтах, не проиндексированных каким-либо общедоступным поисковиком. Одним из направлений решения задачи обработки информации неидексированных сайтов является создание собственной ИПС с последующим ее совершенствованием. Упрощенная архитектура подобной ИПС показана на рис. 1.

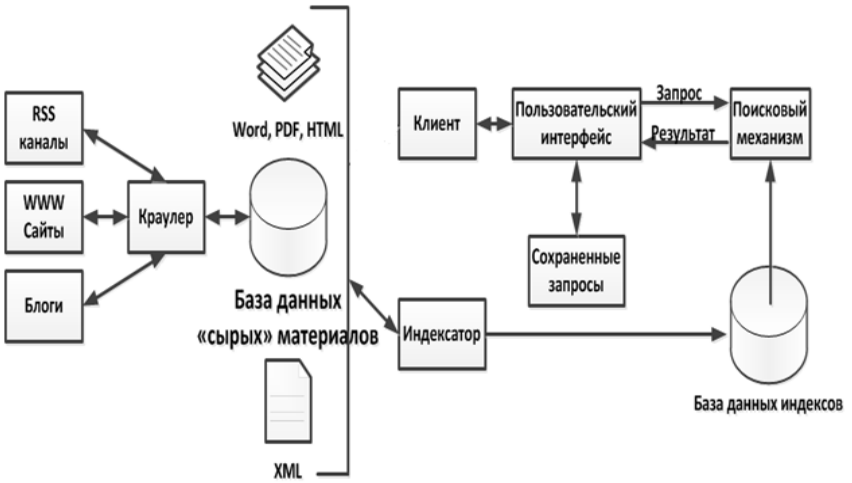

Рис. 1. Архитектура ИПС

В представленной ИПС можно выделить три подсистемы:

1) подсистема сбора данных — включает в себя специальную программу «краулер» («робот», «парсер»), которая проверяет текстовые материалы, изымает все возможные данные и записывает их в базу данных;

2) подсистема обработки и хранения данных — материалы из базы данных обрабатываются (морфологический разбор, нормализация слов) и индексируются (индексы заносятся в специальную базу индексов поисковой платформы);

3) подсистема взаимодействия с пользователем — пользовательский интерфейс API со строкой для ввода запроса и область вывода результатов поиска.

Для разработки ИПС с нуля на уровне лучших мировых образцов требуются серьезные финансовые (до 100 млн долл. в год) и временные (не менее 5 лет) затраты. Вместе с тем существует достаточно большой набор модулей, распространяемых как свободное программное обеспечение, с открытыми исходными кодами. Эти модули могут лечь в основу создания специализированной ИПС, среди них: AOT, MySQL fulltext, Xapian, PostgreSQL Textsearch, Apache Lucene, ApacheSolr.

Одной из наиболее глубоко проработанных с математической и программной точек зрения является платформа ApacheSolr, основанная на библиотеке Lucene [5]. Дополнительным преимуществом архитектуры ApacheSolr является возможность расширения ее функциональности по сравнению с базовой комплектацией.

Применение алгоритмов кластеризации для расширения возможностей специализированной ИПС. Кластерный анализ как самостоятельный раздел математики сложился уже достаточно давно. Вместе с тем задача применения кластерного анализа к результатам

поисковых выдач ИПС требует дополнительных исследований. Так, в ИПС, распространяющихся как специальное программное обеспечение (СПО), в частности в ядре ИПС ApacheSolr (и соответственно в библиотеке Lucene) отсутствуют собственные инструменты кластеризации результатов поиска. В ряде крупных коммерческих проектов, например в продуктах фирмы Oracle, данная возможность существует, однако закрытость исходных кодов ограничивает их применение в собственных разработках.

Для расширения функциональности реализованной на базе ApacheSolr специализированной ИПС была поставлена задача сравнительного анализа методов кластеризации с последующей интеграцией наиболее подходящего из методов. Проанализировав публикации с точки зрения перспектив алгоритмической реализации и возможности интеграции в специализированную ИПС выбраны два алгоритма: STC (suffix tree clustering) — алгоритм, в котором кластеры образуются в узлах специального вида дерева (суффиксное дерево, строящееся из слов и фраз входных документов [6]), и Lingo — алгоритм, основанный на интеграции метода сингулярного разложения и *k*-средних [7].

На рис. 2 приведен пример, заимствованный из работы [6], построения суффиксного дерева и принципа метода кластеризации для строк: «кошка ест сыр», «мышь тоже ест сыр», «кошка тоже ест мышь». На основе построенного дерева формируются базовые кластеры: кошка, сыр, мышь, тоже ест. Если базовые кластеры пересекаются более чем по половине содержащихся в них слов, то происходит объединение кластеров.

В отличие от STC, алгоритм Lingo основан на сингулярном разложении терм-документной матрицы [7].

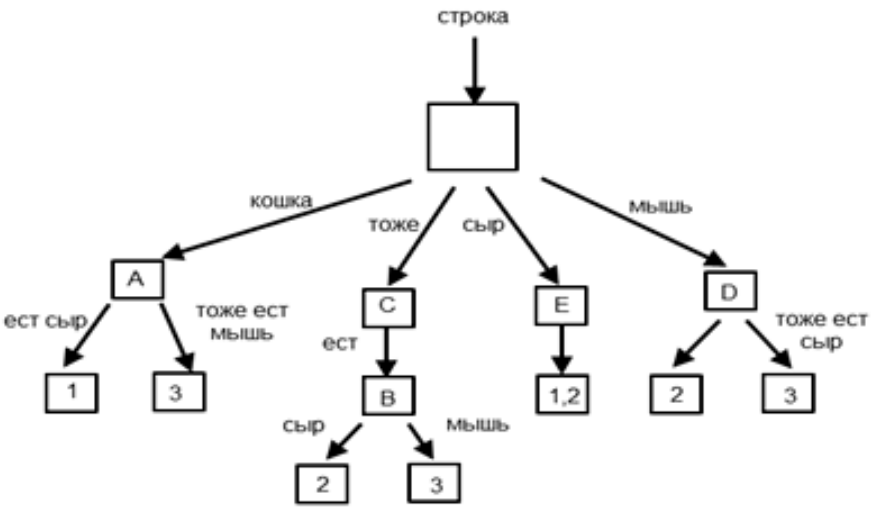

Рис. 2. Алгоритм [6] построения суффиксного дерева

При сингулярном разложении терм-документная матрица A размерностью $t \times d$ разлагается на матрицы U, S и V так, чтобы $A = USV^{-1}$. Здесь U — ортогональная матрица размерностью $t \times t$, где столбцы называют левыми сингулярными векторами матрицы A; V — ортогональная матрица размерностью $d \times d$, где строки называют правыми сингулярными векторами матрицы A; S — диагональная матрица размерностью $t \times d$ с диагональными элементами, упорядоченными по убыванию:

$$\sigma_1 \geq \sigma_1 \geq ... \geq \sigma_{\min(t,d)}.$$

При этом ранг r_A матрицы A равен числу ненулевых элементов.

Алгоритм Lingo предусматривает следующие этапы:

1-й — подача на вход системы множества документов;

2-й — предварительная обработка документов;

3-й — выделение часто встречающихся выражений (термов);

4-й — поиск имен кластеров;

5-й — заполнение кластеров документами;

6-й — сортировка кластеров для отображения;

7-й — вывод результатов кластеризации.

Следует отметить, что в рамках предварительной обработки документов проводятся следующие процедуры:

– очистка текста от HTML-тэгов и специальных символов;

– определение языка текста;

– морфологическая обработка;

– сегментация;

– приведение слов к нормальной форме;

– удаление стоп-слов;

– выделение часто встречающихся выражений.

Суть метода Lingo заключается в том, что сингулярному разложению подвергается терм-документная матрица A, а в полученной матрице U столбцы представляют абстрактные понятия, встречающиеся в исходных документах.

Поиск абстрактных понятий и фраз проводится с помощью классической функции косинусного расстояния. При этом определяется, насколько хорошо фраза представляет собой абстрактное понятие:

$$M = U^{\mathrm{T}} P,$$

где P — матрица размерностью $t \times (t + p)$ построена посредством обработки фраз как псевдо-документов и с использованием TF-IDF схемы взвешивания для оценки важности слова в контексте документа.

Матрица M размерностью $k \times (t + p)$ является матрицей косинусов для каждой пары абстрактное понятие — фраза.

Выбирается одно слово или фраза как имя кластера с наибольшим значением для каждого абстрактного понятия.

Заполнение кластеров документами проводится в соответствии с формулой

$$C = Q^{\mathrm{T}} A,$$

где Q — матрица размерностью $t \times k$, которая формируется после отбора релевантных k столбцов из матрицы P.

Каждый элемент C_{ij} в матрице размерностью $k \times d$ показывает величину соответствия j-го документа i-му кластеру. Документ будет добавлен в кластер, если соответствующее значение будет больше порогового.

Кроме того, эти значения могут быть использованы для сортировки документов в их кластерах, таким образом, наиболее подходящий из них будет легче идентифицировать. Регулируя пороговое значение, можно контролировать количество документов, попадающих в каждый кластер.

Сравнительный анализ данных алгоритмов дает следующие результаты [8]. В табл. 1 представлены данные по времени выполнения процесса кластеризации каждого алгоритма при обработке разного числа документов.

Таблица 1

Время выполнения кластеризации

Алгоритм	Время выполнения, с		
	100 документов	200 документов	400 документов
Lingo	0,16	0,17	0,31
STC	0,01	0,02	0,06

Согласно данным табл. 1, можно сделать вывод о том, что алгоритм Lingo требует на 0,15 с больше времени выполнения по сравнению с алгоритмом STC.

Несмотря на этот незначительный недостаток, алгоритм Lingo обеспечивает получение большего количества кластеров, что подтверждают результаты проведенного анализа (табл. 2).

Таблица 2

Количество полученных кластеров по одному запросу для разного числа документов

Алгоритм	Количество полученных кластеров, шт.		
	100 документов	200 документов	400 документов
Lingo	23	63	63
STC	16	16	16

Таким образом, исходя из полученных данных сравнительного анализа, видно, что алгоритм Lingo обнаруживает значительно больше кластеров, обеспечивая высокое качество кластеризации документов.

Интеграция данных алгоритмов в специализированную ИПС была достигнута путем подключения библиотеки с открытым исходным кодом Carrot2 SearchResultsClusteringEngine.

Тестирование на реальном корпусе документов. Тестирование проведено на текстовом корпусе, состоящем из документов, которые получены в результате мониторинга сайтов определенной тематической направленности, неиндексируемых общеизвестными поисковыми системами. Результаты кластеризации методом Lingo поисковой выдачи по избранной тематике представлены в табл. 3.

Таблица 3

Результаты кластеризации поисковой выдачи по запросу *«Интернет»*

Число документов	Алгоритм Lingo	Алгоритм STC
50	1. Сети Интернет 2. Социальные сети …….. 6. Которые 7. Другие темы	1. Может 2. Интернет …… 10. РФ 11. Другие темы
100	1. Интернет 2. России 3. Сети Интернет …….. 25. Которых 26. Данным 27. Другие темы	1. Газеты 2. ОБСЕ 3. Данными интернет-опросов и интернет-голосований …… 13. Сети 14. Сети Интернет 15. Декларацию ОБСЕ о свободе 16. Другие темы
200	1. Сети Интернет …….. 60. России 61. Которой 62. Многие 63. Другие темы	1. Газеты 2. Информация 3. Соответствии с данными Интернет-опросов …… 14. Многие другие интернет-форумы 15. Модели китайских товарищей 16. Другие темы

В заключение можно сделать вывод, что проблема информационного поиска при больших информационных потоках решаема с помощью поисковой платформы ApacheSolr, а прозрачность и открытость исходного кода предоставляет возможность разработчикам и программистам использовать сторонние инструменты кластерного анализа. При этом установлено, что для корпуса документов избранной тематики, полученных мониторингом ресурсов «серого» Интернета, алгоритм кластеризации Lingo обеспечивает наиболее высокое качество кластеризации. Таким образом, можно констатировать, что подтверждена целесообразность использования технологий кластерного анализа для извлечения знаний применительно к большим массивам информации.

СПИСОК ЛИТЕРАТУРЫ

1. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск. М.: Вильямс, 2011. 528 с.
2. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы. М.: Финансы и статистика, 2004. 424 с.
3. Технологии анализа данных. DataMining, VisualMining, TextMining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. СПб.: БХВ-Петербург, 2007. 384 с.
4. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии. М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. 304 с.
5. The Apache Lucene project develops open-source search software. URL: http://lucene.apache.org/
6. Oren Zamir, Oren Etzioni Grouper: a dynamic clustering interface to Web search results // Networks: The International Journal of Computer and Telecommunications Networking. 1999. Vol. 31, issue 11–16. P. 1361–1374.
7. A survey of Web clustering engines / C. Carpineto, S. Osiński, G. Romano, D. Weiss // ACM Computing Surveys (CSUR). 2009. Vol. 41, issue 3 (July), Article No 17.
8. Summary of clustering algorithms that work within the Carrot2 framework. URL: http://project.carrot2.org/algorithms.html.

Статья поступила в редакцию 25.10.2012