Практическое применение функционального программирования и регулярных выражений в библиометрическом анализе

А.В. Пруцков, И.В. Сусанина

***Аннотация*—При библиометрическом анализе публикаций российских авторов в области арт-терапии на основе данных Научной электронной библиотеки, а также Российской государственной библиотеки за период с 2005 по 2021 гг. возникла задача нахождения наиболее часто встречающихся авторов работ в библиографическом указателе. Были предложены два решения этой задачи: с помощью табличного процессора и с помощью программирования на языке высокого уровня. Для обоих решений были оценены временны́е затраты на их выполнение и точность полученного результата. Решение с помощью табличного процессора требует значительных затрат ручного труда по заполнению таблицы размером 890×608, что приведет к ошибкам и снижению точности. Решение с помощью программирования на языке высокого уровня автоматизирует процесс подсчета, однако требует знания программирования и временны́х затрат на написание программы и ее отладки. Было выбрано второе решение. Оно включало следующие этапы: 1) подготовка исходных данных; 2) написание программы для решения задачи; 3) получение результата с корректировкой библиографических описаний. К программе было выдвинуто дополнительное требование: использовать только функциональное программирование. Разработанная программа позволила решить задачу нахождения наиболее часто встречающихся авторов работ в библиографическом указателе.**

***Ключевые слова*—Библиометрический анализ, функциональное программирование, регулярные выражения, язык программирования Java.**

I. Введение

При написании цикла статей, посвященных библиометрическому анализу публикаций российских авторов в области арт-терапии за период с 2005 по 2021 гг., возникли задачи статистической обработки данных. Большая их часть решались в табличном процессоре (например, Microsoft Excel или LibreOffice Calc). Одной из задач был нахождение наиболее часто встречающихся авторов работ в отдельных разделах библиографического указателя [1] (далее указателя). Текст основной части указателя представляет собой последовательность 890 библиографических описаний, некоторые из которых приведены на рис. 1.

…
14. Арт-терапия. Диалог: Россия — Великобритания: сб. ст. / под ред. Н.Ю. Жвитиашвили.—СПб., 2008.—С.153.
…
19. БАЛАБАСОВА, Н.В. Избранные техники арт-терапии / Н.В. Балабасова.—Краснодар, 2013.—279 с.
…
82. Использование элементов арт-терапии в оценке эффективности стоматологических профилактических программ у детей / О.И. Адмакин, Е.А. Скатова, А.М. Шломина [и др.] // Стоматология детского возраста и профилактика.—2014.—Т. 13, №2.—С.43—46.
…
132. ЛЕБЕДЕВА, Л.Д. Восстановительная арт-терапия эмоциональных состояний онкобольных / Л.Д. Лебедева // Клиническая и специальная психология.—2012. —№ 3.—С.60—70.
133. ЛЕБЕДЕВА, Л.Д. Ресурс восстановительной арт-терапии в эмоциональной адаптации больных раком молочной железы / Л.Д. Лебедева, В.В. Музяков // Развитие радиотерапевтической службы в России. Высокие технологии консервативного лечения: материалы VII Общерос. мед. форума в Приволжском федеральном округе совместно с VII Российской науч.-практ. онкологической конф. / отв. ред. В.В. Родионов. —М., 2012. —С.97—102.
…

Рисунок 1. Примеры библиографических описаний в библиографическом указателе

II. Цель работы

Целью работы автоматизация решения задачи нахождения наиболее часто встречающихся авторов работ в указателе с использованием существующих информационных технологий и применение полученного решения на практике.

III. Вопросы исследования

RQ1. Какое из возможных решений задачи нахождения наиболее часто встречающихся авторов работ в указателе потребует минимальных временны́х затрат и

Статья получена 18 апреля 2022.

Пруцков Александр Викторович, Рязанский государственный радиотехнический университет имени В. Ф. Уткина (РГРТУ), 390005, Российская Федерация, Рязань, Гагарина, 59/1; Рязанский государственный медицинский университет имени академика И. П. Павлова (РязГМУ), 390026, Российская Федерация, Рязань, ул. Высоковольтная, 9; Рязанский государственный университет имени С. А. Есенина (РГУ), 390000, Российская Федерация, Рязань, ул. Свободы, 46 (e-mail: mail@prutzkow.com).

Сусанина Ирина Владимировна, Рязанский государственный медицинский университет имени академика И. П. Павлова (РязГМУ), 390026, Российская Федерация, Рязань, ул. Высоковольтная, 9.

обеспечит максимальную точность?

RQ2. В случае программного решения можно ли решить задачу нахождения наиболее часто встречающихся авторов работ в указателе, используя только функциональное программирование?

IV. Возможные решения задачи нахождения наиболее часто встречающихся авторов работ в библиографическом указателе

A. *С помощью табличного процессора*

Были найдены и проанализированы два способа решения этой задачи.

Задача нахождения наиболее часто встречающихся авторов работ в указателе может быть решена в табличном процессоре. Заголовками столбцов таблицы будут авторы, а строк – библиографические описания. Если автор входит в библиографическое описание, то счетчик на пересечении соответствующей строки работы и столбца автора увеличивается на единицу. Далее необходимо подсчитать сумму счетчиков по столбцам и отсортировать полученные суммы по убыванию.

Этот способ неприемлем по следующим причинам:

1) доля ручного труда в этом способе решения значительна, что привело бы к высоким временны́м затратам;

2) размеры таблицы велики (890 описаний × 608 авторов), что может привести к ошибкам при ее заполнении и снижению точности результатов, а также дополнительным временны́м затратам на проверку правильности заполнения.

B. *Программирование решения задачи на языке высокого уровня*

Задача нахождения наиболее часто встречающихся авторов работ в указателе может быть решена с помощью программы на языке высокого уровня, например языке Java.

Этот способ требует выполнения следующих этапов:

1. Подготовка исходных данных
2. Написание программы для решения задачи.
3. Получение результата с корректировкой библиографических описаний.

Решение задачи с помощью программы на языке высокого уровня, несмотря на необходимость знания программирования и временны́х затрат на написание программы и ее отладки, имеет следующие преимущества:

– обеспечивает минимальные затраты ручного труда для получения результата, что снижает временны́е затраты, и максимальную точность;

– разработанная программа может использоваться для решений этой задачи для других исходных данных.

C. *Выбор решения задачи*

Из проведенного сравнения решений задачи был сделан вывод, что программирование решения задачи на языке высокого уровня потребует минимальных временны́х затрат и обеспечит максимальную точность (RQ1).

К этому решению задачи было выдвинуто требование: использовать только средства функционального программирования языка Java.

Рассмотрим перечисленные выше этапы решения задачи с помощью программирования на языке высокого уровня.

V. Подготовка исходных данных

A. *Последовательность подготовки исходных данных*

Подготовка исходных данных включала два этапа:

1. Получение текста библиографического указателя.
2. Разделение текста на отдельные библиографические описания.

Эти этапы состояли в следующем.

B. *Получение текста библиографического указателя*

Указатель был предоставлен в виде сканированного файла в формате PDF без текстового слоя. Текстовый слой был добавлен в файл с указателем с помощью специализированных Интернет-сервисов. Текстовый слой позволил получить текст указателя.

C. *Разделение текста на отдельные библиографические описания*

Этот этап выполнялся в текстовом процессоре Microsoft Word.

При добавлении текстового слоя в текст были вставлены символы конца абзаца там, где заканчивалась строка одного абзаца и начиналась другая. Написать регулярное выражение, удаляющие такие символы конца абзаца, в текстовом процессоре не удалось. Поэтому эти символы удалялись в два действия:

1) замена всех символов конца абзаца на символы пробела: «^p» → «•»; здесь и далее кавычки обозначают начало и конец строки и в строку не входят, а символ • обозначает пробел;

2) добавление символов конца абзаца перед номерами библиографических описаний; библиографические описания в указателе пронумерованы: номер стоит в начале описания; поэтому символ конца абзаца должен находиться перед номером; символ конца абзаца был добавлен заменой: «•([0-9]{1;3}.•)» → «^0013\1»; номер описан левой частью замены и выделен в группу 1 регулярного выражения; правая часть замены описывает добавление символ конца абзаца перед группой 1, то есть номером описания.

Полученные библиографические описания были сохранены в текстовом файле.

VI. Написание программы для решения задачи

A. *Программа для решения задачи*

Был составлен следующий алгоритм обработки данных, реализованный в программе на языке Java:

1) считать библиографические описания из текстового файла (см. листинг, строки 8-9);

2) извлечь инициалы и фамилии авторов из библиографического описания с помощью регулярного выражения (строки 10-13);

3) подсчитать количество вхождений авторов с помощью карты отображения (интерфейс Map) (строки 14-24).

Программа решения задачи использует только средства функционального программирования языка Java (строки 8-27). Данные преобразовывались следующим образом (таблица I). Приведены типы данных потока языка Java и подсчеты по фрагменту данных из 5 описаний.

Таким образом, при программировании решения задачи использовались только средства функционального программирования языка Java (RQ2).

Листинг. Решение задачи подсчета наиболее часто встречающихся авторов работ в указателе с помощью функционального программирования на языке Java

```
1  final String extractAuthorRegex = " ([^/]+/\\s*)((([А-Я][.]){1,2}\\s[А-Я][а-яё-]+(,\\s)?)+)(\\s*([.;\\[]|(/{2})).+\\.)";
2  final String splitAuthorsRegex = ",\\s*";
3
4  final Pattern extractAuthorPattern = Pattern.compile(extractAuthorRegex);
5  final Pattern splitAuthorsPattern = Pattern.compile(splitAuthorsRegex);
6
7  try {
8      Map<String, Long> sortedAuthors = Files
9              .lines(Paths.get(filename), Charset.forName("windows-1251"))
10             .map(extractAuthorPattern::matcher)
11             .filter(Matcher::find)
12             .map(m -> m.group(2))
13             .flatMap(s -> splitAuthorsPattern.splitAsStream(s))
14             .collect(
15                     Collectors.groupingBy(Function.identity(),
16                             Collectors.counting()))
17             .entrySet()
18             .stream()
19             .sorted(Map.Entry.comparingByValue(Comparator
20                     .reverseOrder()))
21             .collect(
22                     Collectors.toMap(Map.Entry::getKey,
23                             Map.Entry::getValue, (oldV, newV) -> oldV,
24                             LinkedHashMap::new));
25
26     sortedAuthors.forEach((k, v) -> System.out.printf("%s: %d\n", k, v));
27     System.out.println(sortedAuthors.keySet().size());
28 } catch (IOException e) {
29     System.err.println("Error");
30 }
```

↳

B. Регулярное выражение для извлечения инициалов и фамилий авторов

Перед составлением регулярного выражения для извлечения инициалов и фамилий авторов был проведен анализ структур библиографических описаний (типовые описания приведены на рисунке).

Проведенный анализ показал, что инициалы и фамилии авторов находятся между одинарной наклонной чертой и двумя наклонными чертами (см. рисунок, библиографические описания 132, 133), или точкой (библиографическое описание 19), или открывающейся квадратной скобкой (библиографическое описание 82), или точкой с запятой.

Регулярное выражение (см. листинг, строка 1) состоит из трех основных частей (таблица II).

Таким образом, для извлечения инициалов и фамилий авторов необходимо выделить группу 2 регулярного выражения.

ТАБЛИЦА I. ПОСЛЕДОВАТЕЛЬНОСТЬ ПРЕОБРАЗОВАНИЯ ДАННЫХ ПОТОКОМ

Строка	Результат выполнения
8-9	Stream<String> – все библиографические описания: 14. Арт-терапия. … / под ред. Н.Ю. Жвитиашвили. … С.153. 19. БАЛАБАСОВА, Н.В. Избранные … / Н.В. Балабасова. … 279 с. 82. Использование элементов … / О.И. Адмакин, Е.А. Скатова, А.М. Шломина [и др.] // … С.43—46. 132. ЛЕБЕДЕВА, Л.Д. Восстановительная … / Л.Д. Лебедева // … С.60—70. 133. ЛЕБЕДЕВА, Л.Д. Ресурс … / Л.Д. Лебедева, В.В. Музяков // … С.97—102.
10-11	Stream<String> – библиографические описания только с инициалами и фамилиями авторов: 19. БАЛАБАСОВА, Н.В. Избранные … / Н.В. Балабасова. … 279 с. 82. Использование элементов … / О.И. Адмакин, Е.А. Скатова, А.М. Шломина [и др.] // … С.43—46. 132. ЛЕБЕДЕВА, Л.Д. Восстановительная … / Л.Д. Лебедева // … С.60—70. 133. ЛЕБЕДЕВА, Л.Д. Ресурс … / Л.Д. Лебедева, В.В. Музяков // … С.97—102.
12	Stream<String> – инициалы и фамилии авторов из библиографическими описаниями: Н.В. Балабасова О.И. Адмакин, Е.А. Скатова, А.М. Шломина Л.Д. Лебедева Л.Д. Лебедева, В.В. Музяков
13	Stream<String> – список авторов: Н.В. Балабасова О.И. Адмакин Е.А. Скатова А.М. Шломина Л.Д. Лебедева Л.Д. Лебедева В.В. Музяков
14-16	Map<String, Long> – карта отображения с авторами и количество описаний с ними: [Н.В. Балабасова, 1], [О.И. Адмакин, 1], [Е.А. Скатова, 1], [А.М. Шломина, 1], [Л.Д. Лебедева, 2], [В.В. Музяков, 1]
17-24	Map<String, Long> – карта отображения, отсортированная по количеству описаний с авторами по убыванию: [Л.Д. Лебедева, 2], [Н.В. Балабасова, 1], [О.И. Адмакин, 1], [Е.А. Скатова, 1], [А.М. Шломина, 1], [В.В. Музяков, 1]

ТАБЛИЦА II. РАЗБОР РЕГУЛЯРНОГО ВЫРАЖЕНИЯ ДЛЯ ВЫДЕЛЕНИЯ АВТОРОВ ИЗ БИБЛИОГРАФИЧЕСКОГО ОПИСАНИЯ

Номер группы	Группы регулярного выражения	Описание
1	([^/]+/\\s*)	Часть описания до инициалов и фамилий авторов: один или несколько символов, не являющихся наклонной чертой; наклонная черта; ноль или несколько разделителей слов (пробел)
2	((([А-Я][.]){1,2}\\s[А-Я][а-яё-]+(,\\s)?)+)	Инициалы и фамилии авторов: один или два инициала и фамилии авторов через запятую с разделителем слов; запятая с разделителем, которые могут отсутствовать
3	(\\s*([.;\\[]|(/{2})).+\\.)	Часть описания после инициалов и фамилий авторов: возможно разделитель слов; точка, точка с запятой, открывающаяся квадратная скобка или две наклонных черт; любые символы, заканчивающиеся точкой

VII. ПОЛУЧЕНИЕ РЕЗУЛЬТАТА С КОРРЕКТИРОВКОЙ БИБЛИОГРАФИЧЕСКИХ ОПИСАНИЙ

Была написана вспомогательная программа, которая выводит описания, не соответствующие регулярному выражению. Одни описания действительно не соответствовали регулярному выражению, другие – были неправильно распознаны в текстовом слое: отсутствовали запятые между инициалами, фамилиями авторов, пробелы между инициалами и фамилией, присутствовали пробелы в фамилии и др.

Полученный список авторов был проверен вручную с целью выявления однофамильцев авторов и проверки правильности подсчетов.

VIII. ОТВЕТЫ НА ВОПРОСЫ ИССЛЕДОВАНИЯ

RQ1. Какое из возможных решений задачи нахождения наиболее часто встречающихся авторов работ в указателе потребует минимальных временны́х затрат и обеспечит максимальную точность?

Ответ: Решение с помощью программы на языке высокого уровня потребует минимальных временны́х затрат и обеспечит максимальную точность по сравнению с решением задачи в табличном процессоре.

RQ2. В случае программного решения можно ли решить задачу нахождения наиболее часто встречающихся авторов работ в указателе, используя только функциональное программирование?

Ответ: Да, можно.

IX. ЗАКЛЮЧЕНИЕ

В работе получены следующие результаты.

1. Найдены и оценены параметры двух способов решения задачи нахождения наиболее часто встречающихся авторов работ в указателе: с помощью табличного процессора и программированием решения

задачи на языке высокого уровня.

2. Сделано вывод о том, что решение этой задачи в табличном процессоре имеет высокую долю ручного труда и может привести к ошибкам при ее заполнении.

3. Сделано вывод о том, что решение этой задачи с помощью программы на языке высокого уровня обеспечивает минимальные затраты ручного труда для получения результата и позволяет использовать программу в дальнейшем для решения таких задач. Поэтому для решения задачи выбран этот способ.

4. Выделены этапы решения задачи нахождения наиболее часто встречающихся авторов работ в указателе, включающие подготовку исходных данных, написание программы для решения задачи и получение результата с корректировкой библиографических описаний.

5. Разработана программа решения задачи с использованием только функционального программирования, составлено регулярное выражение для извлечения инициалов и фамилий авторов.

6. Выполнены этапы решения задачи и получен результат, использованный при написании статей по библиометрическому анализу публикаций российских авторов в области арт-терапии за период с 2005 по 2021 гг.

Наиболее полно функциональное программирование в языке Java описано в [2]. При составлении регулярного выражения для извлечения инициалов и фамилий авторов использовалась в качестве справочника книга [3]. Познакомиться с регулярными выражениями в текстовом процессоре Microsoft Word можно в [4]. Возможности табличных процессоров наилучшим образом представлены в [5].

Библиография

[1] Дрешер Ю.Н. *Арт-терапия:* библиогр. указ. (2005-2016 гг.). – Казань: Медицина, 2016. – 132 с.

[2] Урма Р.-Г., Фуско М., Майкрофт А. *Современный язык Java. Лямбда-выражения, потоки и функциональное программирование*: пер. с англ. – СПб.: Питер, 2020. – 592 с.

[3] Фридл Дж. *Регулярные выражения*: пер. с англ. – 3-е изд. – СПб.: Символ-Плюс, 2008. – 608 с.

[4] Morgado, F. *Microsoft Word Secrets. The Why and How of Getting Word to Do What You Want*. Apress, 2017.

[5] Маликова Л.В., Пылькин А.Н. *Практический курс по электронным таблицам MS Excel*: учеб. пособие для вузов. – М.: Горячая линия-Телеком, 2004. – 244 с.

Practical Application of Functional Programming and Regular Expressions in Bibliometric Analysis

Alexander Prutzkow, Irina Susanina

***Abstract*— In a bibliometric analysis of publications by Russian authors in the field of art therapy based on the library database of the Scientific Electronic Library, as well as the Russian State Library for the period from 2005 to 2021 we to solve a problem of the most frequently encountered authors of works in the bibliographic index. We propose two solutions to the problem: using a spreadsheet and using programming in a high-level language. For the solutions, the time costs for implementation and the reliability of the result obtained were compiled. The spreadsheet solution requires significant manual labor to populate an 890x608 table, resulting in errors and speed display. The high-level language programming solution automates the counting process but requires programming knowledge and time spent on writing and debugging the program. We chose the second solution. There are three stages of the solution: 1) preparation of initial data; 2) writing a program to solve the problem; 3) result with correction of bibliographic descriptions. An addition was put forward to the program: the use of only functional programming. The developed program revealed deviations from the most frequently encountered authors of works in the bibliographic index.**

Keywords: **Bibliometric Analysis, Functional Programming, Regular Expressions, Java Programming Language.**

REFERENCES

[1] Dresher Ju.N. *Art-terapija* [Art-Therapy]: bibliogr. ukaz. (2005-2016 gg.). – Kazan': Meditsina, 2016. – 132 pp.

[2] Urma, R.-G., Fusko M., Mycroft, A. *Modern Java in Action. Lambdas, Streams, Functional and Reactive Programming*. Manning, 2018.

[3] Friedl, J. *Mastering Regular Exspressions*, 3rd ed. O'Reilly, 2006.

[4] Morgado, F. *Microsoft Word Secrets. The Why and How of Getting Word to Do What You Want*. Apress, 2017.

[5] Malikova L.V., Pylkin A.N. *Prakticheskij kurs po elektronnym tablitsam MS Excel* [Practical Course of the Microsoft Excel Electronic Spreadsheet]: ucheb. posobie dlja vuzov. – M.: Gorjachaja linija-Telekom, 2004. – 244 pp.