Оригинальная статья / Original article

https://doi.org/10.21869/2223-1536-2024-14-3-22-35

УДК 004.042

Метод и алгоритм интеллектуальной обработки текстовой информации

С. В. Ефанов[1] ✉, Е. Н. Иванова[1], И. Е. Чернецкая[1]

[1] Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

✉ e-mail: nshysh@yandex.ru

Резюме

Цель исследований *заключается в разработке алгоритма интеллектуальной обработки для классификации текстовой информации. Поскольку количество информации растет каждый день, необходимо быстро и качественно отделять значимое от второстепенного содержимого. Поэтому разработка алгоритма интеллектуальной обработки для классификации текстовой информации является актуальной задачей.*

Методы. *Предложен метод для классификации текстовой информации, представленной на одном или нескольких естественных языках. В его основу входит 5 ключевых стадий: ввод задания, накопление очереди задач, обработка задачи, формирование результата обработки задания, вывод результата. Входное задание представлено в виде HTTP-запроса, в теле которого хранится объект файла. Если интенсивность входного потока больше скорости обработки, то происходит накопление задач. После выбора активного задания (по принципу FIFO) происходит его обработка. В результате преобразований происходит декодирование принятых данных в строку, используя кодировку UTF-8. Под обработкой понимается процесс рубрикации, когда происходит поиск шаблонов в строке. По завершению рубрикации происходит формирование результата по выбранному заданию. Из накопленного результата формируется ответ на исходный HTTP-запрос, в теле которого находится список найденных рубрик.*

Результаты. *Разработан метод и алгоритм обработки текстовых данных, позволяющие определить тематики, которые присутствуют во входном наборе данных. Алгоритм, реализованный программно, позволяет работать с текстовыми данными на различных языках.*

Заключение. *Программная разработка алгоритма классификации текстовых данных была выполнена на языке программирования C++ с использованием библиотек Qt версии 5.11. Данная реализация показала пропускную способность 1-5 Мб в секунду (на однородном входном текстовом наборе данных). Алгоритм позволяет корректно обрабатывать поврежденные форматы файлов.*

Ключевые слова: *рубрикация; автоматическая обработка; классификация данных; алгоритм определения естественных языков; n-граммный алгоритм; словарь шаблонов.*

Конфликт интересов: *Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.*

Для цитирования: Ефанов С. В., Иванова Е. Н., Чернецкая И. Е. Метод и алгоритм интеллектуальной обработки текстовой информации // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2024. Т. 14, № 3. С. 22–35. https://doi.org/10.21869/2223-1536-2024-14-3-22-35

Поступила в редакцию 18.07.2024 *Подписана в печать 20.08.2024* *Опубликована 30.09.2024*

Algorithm for intelligent procesing of text information

Sergei V. Efanov[1] ✉, Elena N. Ivanova[1], Irina E. Chernetskaya[1]

[1] Southwest State University
50 Let Oktyabrya Str. 94, Kursk 305040, Russian Federation

✉ e-mail: nshysh@yandex.ru

Abstract

The purpose of research. *The goal of the research is to develop an intellectual processing algorithm for classifying text information. As the amount of information grows every day, it is necessary to quickly and efficiently separate significant from unimportant content. Therefore, the development of an intellectual processing algorithm for classifying text information is an urgent task.*

Methods. *A method is proposed for classifying text information presented in one or more natural languages. It is based on 5 key stages: entering a task, accumulating a queue of tasks, processing the task, generating the result of processing the task, outputting the result. The input task is presented in the form of an http request, the body of which contains a file object. If the intensity of the input stream is greater than the processing speed, then an accumulation of tasks occurs. After selecting the active task (using the FIFO principle), it is processed. As a result of the transformations, the received data is decoded into a string using UTF-8 encoding. Processing refers to the process of categorization, when a search for patterns occurs in a line. Upon completion of rubrication, the result for the selected task is generated. From the accumulated result, a response to the original http request is formed, the body of which contains a list of found categories.*

Results. *A method and algorithm for processing text data has been developed to determine the topics that are present in the input data set. The algorithm, implemented in software, allows you to work with text data in various languages.*

Conclusion. *The software development of the text data classification algorithm was carried out in the C++ programming language using the Qt libraries version 5.11. This implementation showed a throughput of 1-5 MB per second (on a homogeneous input text data set). The algorithm allows you to correctly process damaged file formats.*

Keywords: *rubrication; automatic processing; data classification; natural language detection algorithm; n-gram algorithm; template dictionary.*

Conflict of interest: *The Authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.*

For citation: Efanov S.V., Ivanova E.N., Chernetskaya I.E. Algorithm for intelligent procesing of text information. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Serija: Upravlenie, vychislitel'naja tekhnika, informatika. Meditsinskoe priborostroenie = Proceedings of the Southwest State University. Series: Control, Computer Engineering, Information Science. Medical Instruments Engineering.* 2024;14(3):22–35. (In Russ.) https://doi.org/10.21869/2223-1536-2024-14-3-22-35

Received 18.07.2024 *Accepted 20.08.2024* *Published 30.09.2024*

***

Введение

Обработка больших данных стала ключевой задачей во многих отраслях, таких как медицина, банковское дело, производство, маркетинг и т. д. С ростом объема данных, которые собирают компании, возникает потребность в их обработке [1]. Однако традиционные методы классификации данных, основанные на анализе вручную и использовании статистических методов, ограничены по своей эффективности и не могут

стать применимыми при работе с большими объемами информации [2]. Поэтому проблема классификации текстовой информации актуальна как никогда ранее.

Представленный алгоритм реализует программа классификатора текстовых данных.

Классификация текстовой информации – это процесс обработки входного массива данных, когда происходит анализ наличия различных тем (рубрик) в тексте [3].

Рассмотрим схему методики проведения обработки входного файла (рис. 1).

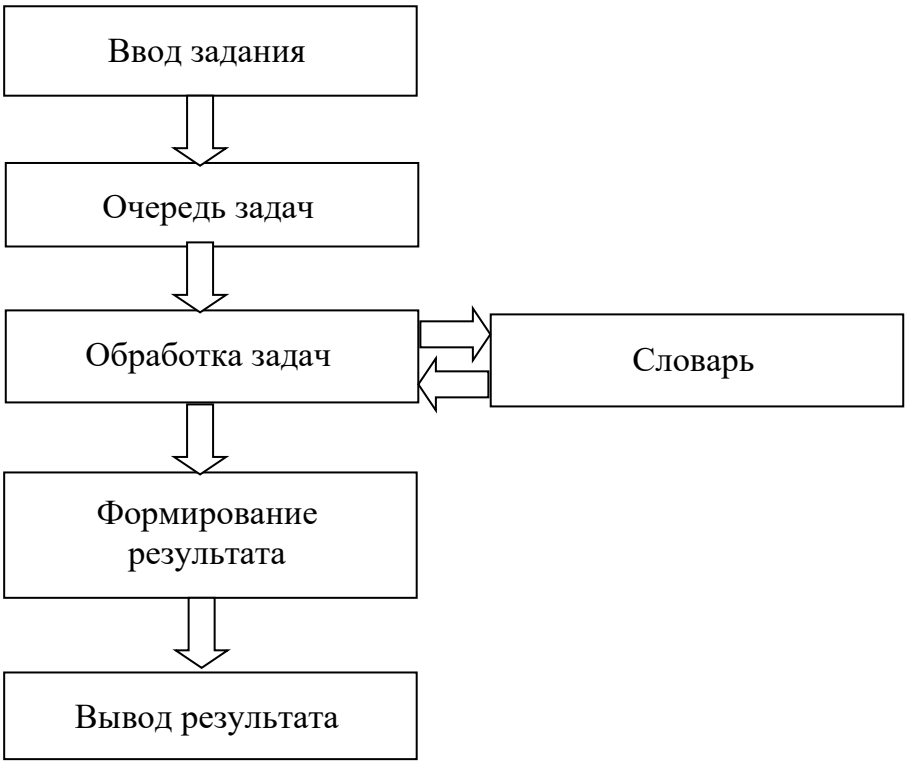

Рис. 1. Схема методики проведения обработки входного файла

Fig. 1. The method of processing the input file

Данная методика реализована программно с использование средств разработки на языке C++.

Входное задание представлено в виде HTTP-запроса, в теле которого хранится объект файла. Если интенсивность входного потока больше скорости обработки, то происходит формирование пула задач. После выбора активного задания (по принципу FIFO – «первым пришёл – первым ушёл») происходит его обработка.

При обработке задач выполняется определение списка естественных языков, которые находятся в тексте. На основе найденных языков подбираются соответствующие шаблоны, наличие которых будет проверяться в тексте [4]. Корпус найденных шаблонов сформирует список полученных рубрик, по которым можно будет классифицировать входной текст [5].

На качество обработки влияет точность определения списка естественных языков в тексте [6].

При распознавании языков происходит двухфазная обработка текста. Первая часть основана на анализе текста в качестве массива символов, где каждый элемент анализируется на предмет его соответствия из стандарта кодирования UTF-8. При нахождении связи делается вывод о том, что переданный символ принадлежит некоторой определенной смысловой группе, которая также связана с языком (кроме символов пунктуации). Так как большинство групп символов из UTF-8 принадлежит конкретному языку, можно сделать вывод о наличии того или иного языка в тексте [7]. Однако данный метод не является предельно точным, так как входной текст может содержать латинские символы, из-за чего данный подход способен определить лишь латинскую группу, а не конкретные языки. Поэтому для улучшения качества была реализована вторая фаза определения языков[1].

Вторая часть определителя языков основана на использовании n-граммного алгоритма, с помощью которого улучшается результат нахождения языков. В данном алгоритме подсчитываются частоты n-грамм (сочетаний символов или подстрок, длиной не более n) и предполагается, что примерно 300 самых часто используемых n-грамм сильно зависят от языка [8]. Главным языком текста считается тот, n-граммы которого встречаются чаще остальных языков. Языком тестируемого документа считается найденный в тестовом документе [9].

Если использовать только n-граммный алгоритм, то возможны потери присутствующих языков в коротких текстах. Результаты работы двухфазного определителя языков представлены ниже (рис. 2).

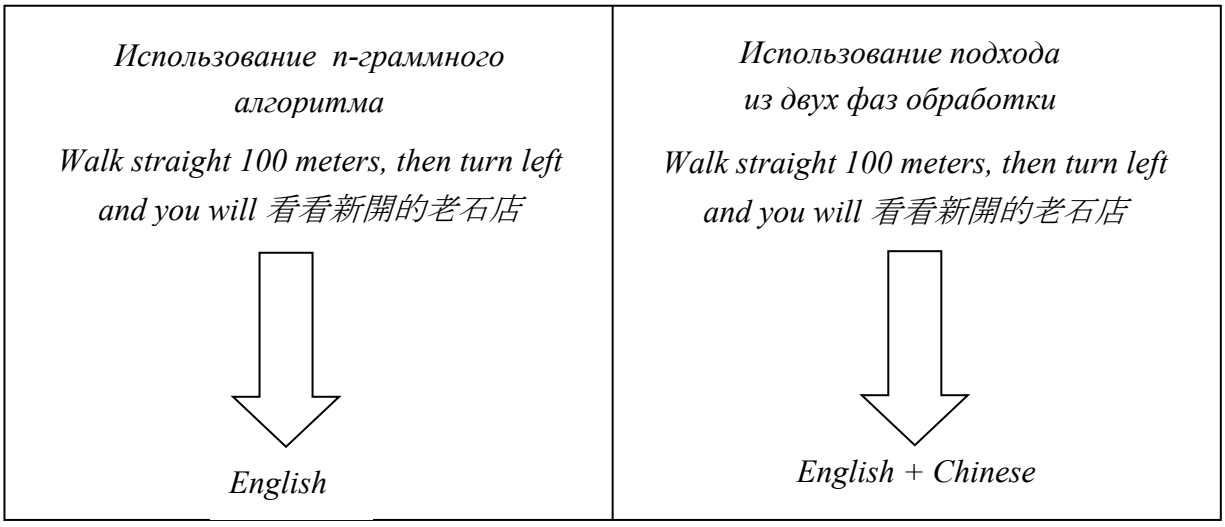

Рис. 2. Результаты использования подхода из двух фаз обработки

Fig. 2. Results of using a two-phase approach

[1] Давыдова Ю. В. Методы текстового поиска и обработки информации в социальных сетях при управлении деятельностью правоохранительных органов: дис. … канд. техн. наук. Белгород, 2021. 146 с.

Использование двухфазной обработки позволяет обеспечить корректность нахождения языков в тексте, уменьшая пропуски языков.

Однако не всегда текст может быть сразу передан на определение языков. Уровень качества поиска зависит от структуры организации текста (например, для файлов в формате json) [10]. Данная ситуация является частным случаем при определении языков [11].

Для решения этой задачи происходит анализ структуры, при этом не имеет значения, является ли она полноценная, не содержащая смысловых разрывов. Обработка json основана на подходе, который не зависит от структуры «ключ – значение». В процесс обработки подобных файлов находятся значения (value), которые накапливаются. Из них далее формируется буфер, который и будет передан на определение языков. Рассмотрим результаты обработки файла с json-структурой, которая имеет смысловой разрыв (рис. 3).

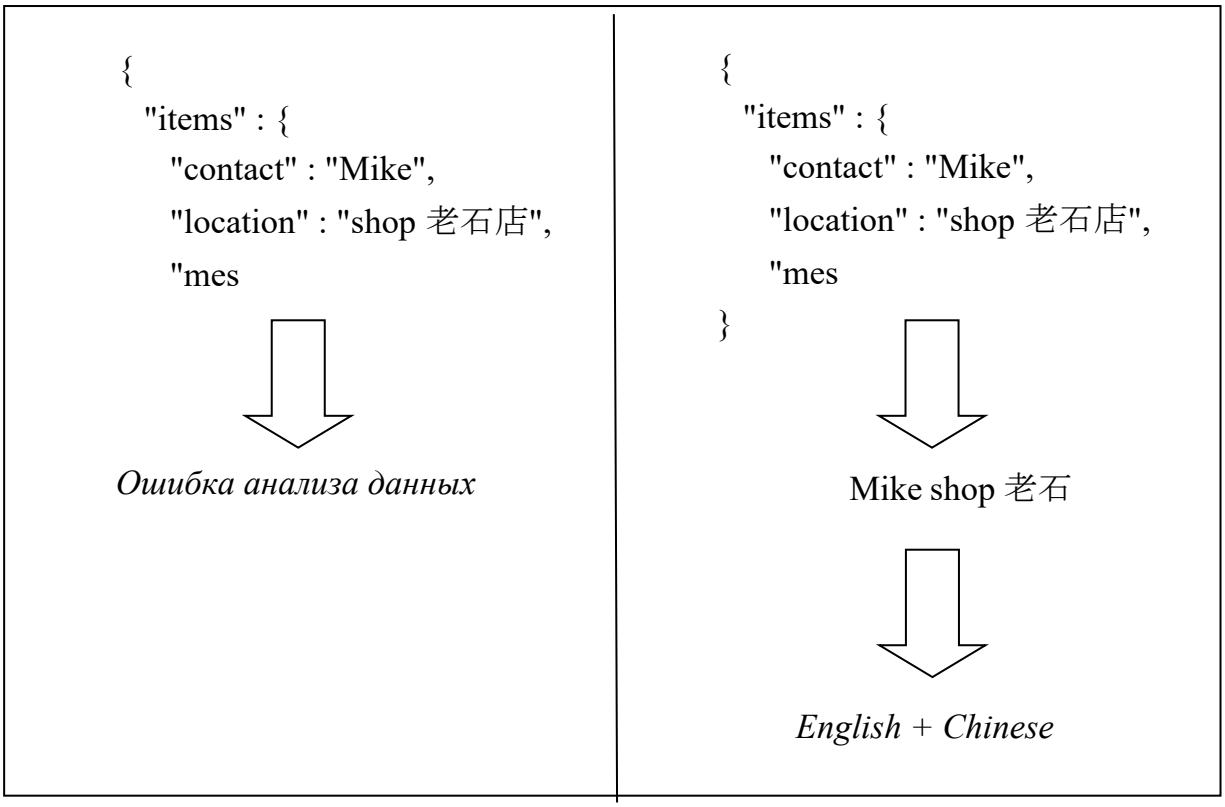

Рис. 3. Результаты обработки файла с json-структурой

Fig. 3. Results of processing a file with a json-structure

Использование данного подхода позволяет обрабатывать файлы с «битыми» структурами json.

При выполнении двухфакторной обработки увеличивается скорость анализа задания, так как на вход второй фазы подается список присутствующих в тексте языков, а не все доступные языки (количество которых 100 единиц). Данный список всегда меньше

инициализируемого количества языков (всего доступно 100 языков). Поэтому нет необходимости искать нерелевантные языки в тексте. С помощью первой фазы определитель уменьшает количество предполагаемых языков, и тем самым увеличивается скорость обработки. Графики, описывающие данную ситуацию, можно увидеть ниже (рис. 4).

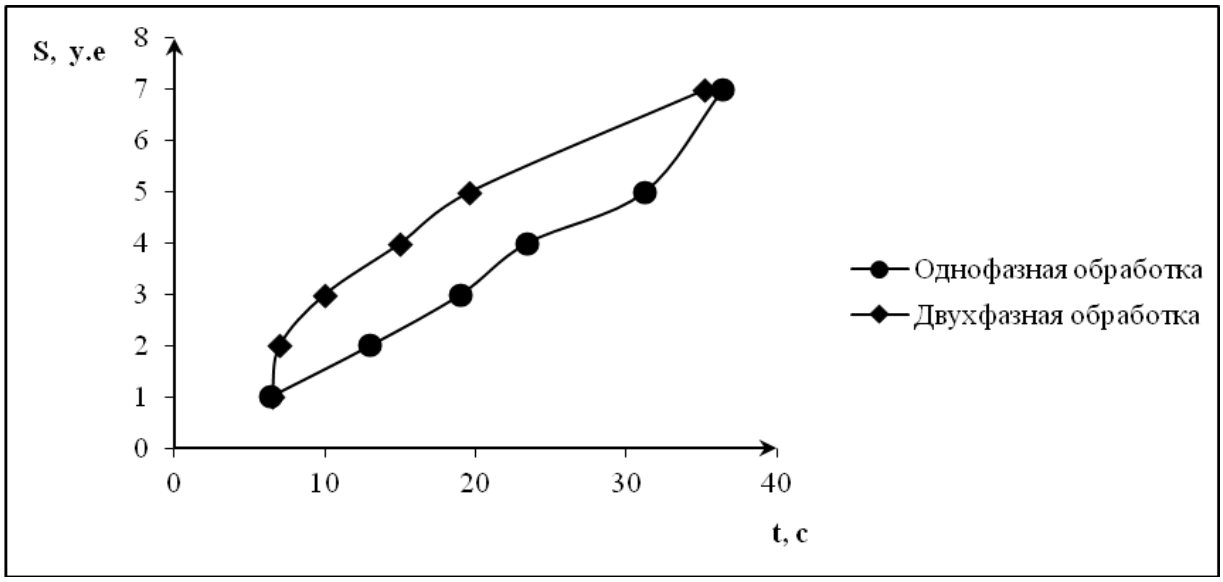

Рис. 4. Экспериментальные результаты измерения времени работы двух алгоритмов

Fig. 4. Experimental results of measuring the running time of two algorithms

За счет сокращения количества языков, подлежащих к поиску в тексте, удается достичь ускорения определения финальных языков.

Материалы и методы

Обработка данных происходит на основе созданного заранее словаря элементов, которые являются поисковыми шаблонами. Поиск базируется на основе анализа текста на предмет вхождения одной или нескольких рубрик.

Каждая тема хранит в себе набор связей шаблонов и языков [12]. В данном контексте под шаблонами понимается слово или словосочетание на естественном языке. Также шаблон может иметь сокращение [13].

Анализ текста основан на использовании TF-IDF-векторов, которые позволяют рассчитать коэффициент сходства терма (одной единицы текста) с шаблоном [14]. Для нахождения атрибутов тематики внутри текста используется поиск ключевых связей векторов [15].

Модель с образцами, которые будут применяться в поисковых операциях, загружается при старте программы. Рассмотрим модель с организацией данных в виде словаря элементов (рис. 5).

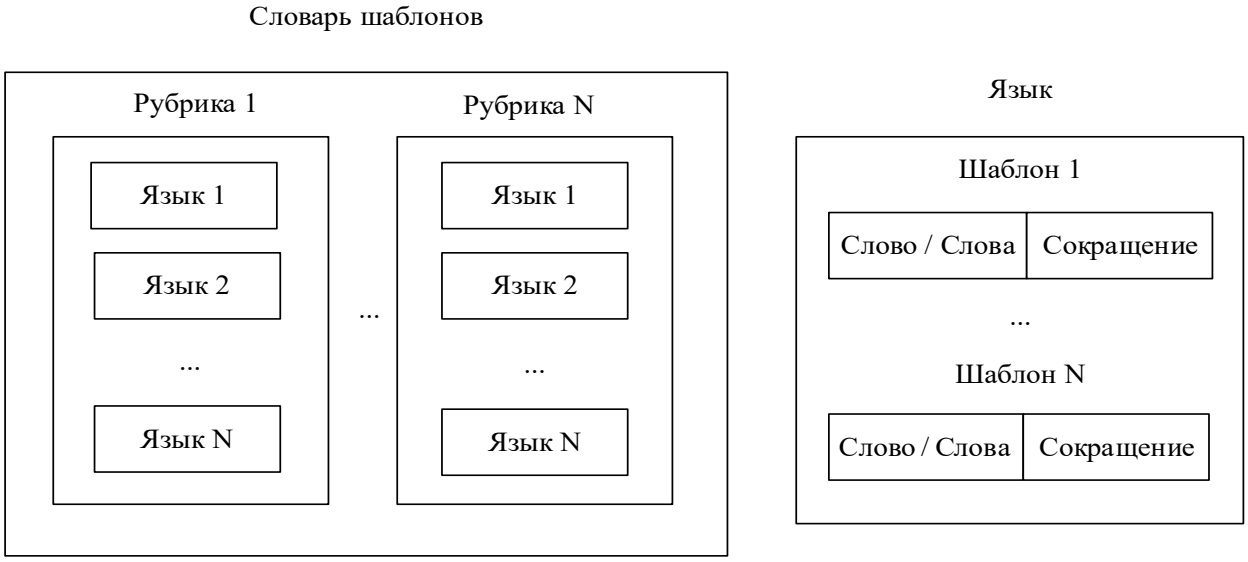

Рис. 5. Модель словаря

Fig. 5. Dictionary Model

Поисковые операции выполняются последовательно над каждым выбранным шаблоном из словаря рубрик. Этапы алгоритма, который описывает реализацию процесса обработки потока текстовых данных, представлены ниже (рис. 6, 7).

Входной файл передается в теле HTTP-запроса. После считывания запроса происходит извлечение и сохранение файла в памяти программы.

Далее происходит анализ полученного файла с целью определения формата (MIME-типа) [16]. Относительно данного типа выбирается нужный парсер. Под парсером в данном случае подразумевается специальная сущность способная, обработав файл, извлечь текстовое содержимое.

Выбор парсера основан на анализе так называемый magic bytes (байтовую последовательность, которая однозначно указывает на заданный формат файла) [17]. Все используемые парсеры можно разделить на две группы. Первая группа парсеров отвечает за обработку файлов, которые являются контейнерами, а вторая группа – для неархивных файлов. Например, форматы, связанные с парсерами из первой группы (rar, uue, gz) и относительно второй группы (html, txt, pdf и т. д.). На данном этапе важно корректно определить нужный обработчик, так как каждый парсер ориентирован только на свой заданный формат файла [18].

После выбора парсера происходит обработка нужного файла, а именно извлечение контента. Под контентом подразумевается текстовое содержимое, которое очищено от структур и тегов. Например, при обработке файла, имеющего формат html, будет получена структура, которая хранит в себе контент, размеченный гипертекстовыми тегами [19].

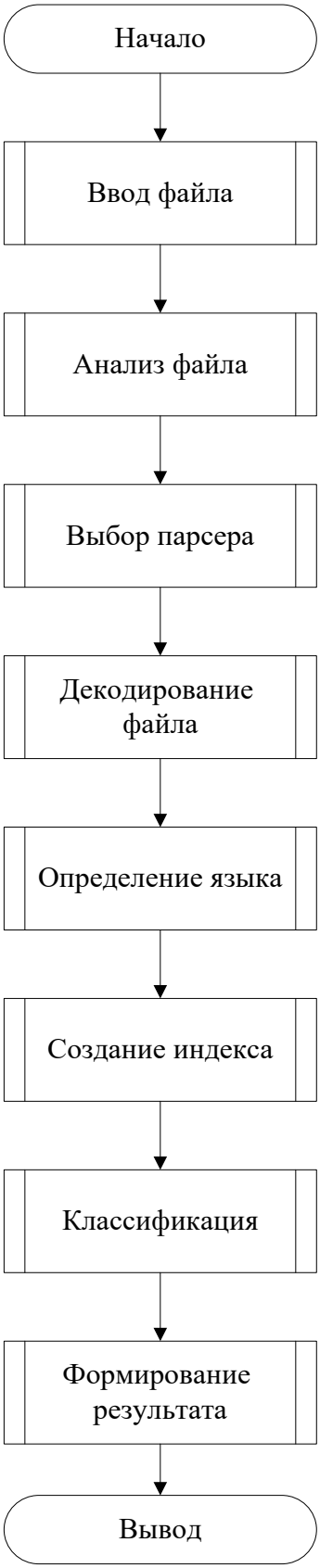

Рис. 6. Алгоритм обработки потока данных

Fig. 6. Data stream processing algorithm

Для корректной работы с данным содержимым необходимо очистить его от всевозможных сопутствующих тегов, которые не имеют смысла относительно цели обработки. Процесс анализа итеративен, происходит несколько циклов определения и очистки тегов, чтобы в итоге получить контент, который можно будет передать на определитель языков.

После того как текст, который был получен из файла, был модернизирован и стал полноценным контентом, происходит анализ для получения вхождений естественных языков. Данная процедура основана на 3 стадиях.

В первой стадии происходит анализ всех символов контента, чтобы найти главные группы из кодировки UTF-8. В тексте могут присутствовать символы из латинской или греческой группы. Некоторые группы однозначно связаны со своим языком, так как они и являются основой для данного языка. Например, символы из группы GREEK (греческие символы) представляют собой базис для греческого алфавита. И именно эти символы однозначно указывают на то, что данный язык присутствует в тексте. В свою очередь группа латинских символов не дает такой конкретики. Она лишь косвенно может указать на список из преобладающих языков. Существует множество языков, алфавиты которых используют латинские символы как основные при письме. Поэтому относительно данной группы нельзя однозначно сделать вывод о том, какой именно язык присутствует в тексте.

Обработчик может лишь косвенно передать список языков, которые связаны с данной символьной группой из кодировки UTF-8.

На второй стадии выбираются символы групп, которые не дают однозначности (например, латиница или кириллица) и из которых формируется отдельный текст. Это делается для того, чтобы исключить другие группы символов. В итоге получается текст, внутри которого символы только одной группы (например, латиница). Остальные группы, которые однозначно указывают на принадлежность к конкретному алфавиту, позволяют заранее сделать вывод о нужном языке. И поэтому эта запись с языком попадает в результирующий список.

На третью стадию передаются тексты из символьных групп, которые не обладают сильной связью с каким-либо конкретным алфавитом. Для каждого текста происходит анализ, который находит n-граммы слов и подсчитывает коэффициент TF-IDF. Данный коэффициент нужен для получения специального числового значения, которое отражает, насколько часто встречается тот или иной терм в тексте. Определение языка строится на статистическом подходе, когда получаются значения частот термов и сравниваются с уже сохраненными значениями. За основу взята мысль, что примерно 300 самых часто используемых n-грамм сильно зависят от языка. В ходе анализа происходит выявление минимального отклонения тестируемого значения n-граммной статистики с ее заданной величиной. После этого языком тестируемого контента считается язык, величина отклонения которого меньше, чем заданная статистика.

Все языки, которые были получены, сводятся в один список, который будет использоваться при поисковых запросах.

Далее происходит создание поискового инвертированного индекса, который представляет собой специальную структуру данных для хранения термов из текста. Сохраняется связь между словом из текста и документом. Для каждого слова из коллекции текстов ставится в соответствие список всех документов, в которых данный терм встречался. Эта структура предоставляет быстрый доступ к документам по соответствующим термам.

После формирования индекса происходит поиск шаблонов. При поиске шаблона, исходя из сохраненного оператора, составляется запрос внутри специального анализатора, именуемого QueryParser. Его задача произвести анализ переданной строки, а также сформировать нужный запрос, именуемый Query. Каждый Query имеет свой тип и параметры, которые определяют его назначение. При создании Query анализируются все особенности переданной строки на наличие операторов, отражающих тип запроса. В данном случае под операторами понимаются специальные символы, которые явно указывают на то, какой механизм поиска будет применен

для заданного шаблона. Для обработки шаблонов внутри переданной строки в QueryPaser применяется языковой анализатор Analyzer для работы с текстовыми данными. Analyzer выполняет функцию разметки запроса для анализа входной строки, именуемой Input string. Отметка Index in memory говорит о том, что созданный индекс сохраняется в оперативную память. Входная строка (Input string) преобразуется в поисковую строку (search string) посредством замены знаков препинания на разделитель (в данном случае пробел).

Рассмотрим общую схему, отражающая организацию работы анализатора запросов QueryParser (рис. 7).

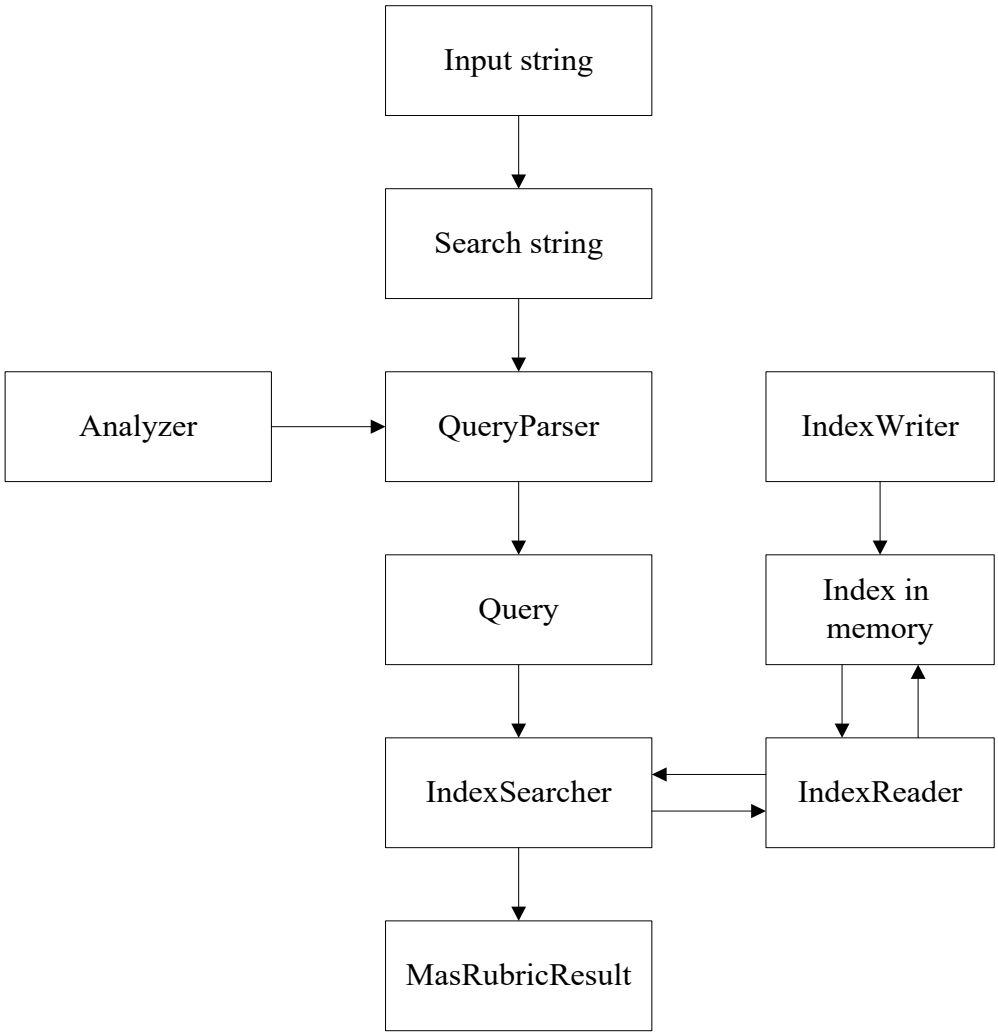

Рис. 7. Организация работы анализатора запросов

Fig. 7. Organizing the work of the query analyzer

После обработки статистика по найденным шаблонам сохраняется в обработчике (MasRubricResult), который хранит в себе результат рубрикации. Результат обработки файла формируется на основе списка статистик по всем найденным рубрикам.

Результаты и их обсуждение

Описанные метод и алгоритм интеллектуальной обработки позволяют оптимизировать процесс автоматической классификации текстовых данных. Предложенные метод и алгоритм применимы в области автоматической рубрикации, когда важно произвести поиск шаблонов на естественном языке в текстовом наборе данных.

Выводы

Предложен метод для классификации текстовой информации. В его основу входит 5 ключевых стадий: ввод задания, накопление очереди задач, обработка задачи, формирование результата обработки задания, вывод результата. Файл попадает на вход программы. После того как файл прочитан, происходит формирование задания на классификацию. Сформированное задание сохраняется в очередь, из которой выбирается активное задание по принципу FIFO для классификации. После того как обработчик завершит рубрикацию данных, произойдет формирование результата по выбранному заданию.

Разработан метод и алгоритм обработки текстовых данных, позволяющие определить тематики, которые входят во входной набор данных. Алгоритм, реализованный программно, позволяет работать с текстовыми данными на различных языках. Программная разработка алгоритма классификации текстовых данных была выполнена на языке программирования C++ с использованием библиотек Qt версии 5.11. Данная реализация показала пропускную способность 1–5 Мб в секунду (на однородном входном текстовом наборе данных). Алгоритм позволяет корректно обрабатывать поврежденные форматы файлов.

Список литературы

1. Кобышев К. С., Молодяков С. А. Анализ и классификация алгоритмов извлечения отношений из текстовых данных // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2021. № 5. С. 71–79. https://doi.org/10.37882/2223-2966.2021.05.15. EDN KXLLZK

2. Поляков А. А., Фетисов М. В. Классификация алгоритмов предварительной обработки текстовых данных для машинного обучения // Технологии инженерных и информационных систем. 2021. № 4. С. 70–79. EDN QROXYD

3. Разработка перспективных методов поиска и классификации текстовой информации из открытых источников сети Интернет / М. А. Сазонов, А. В. Яковлев, М. О. Кожанчиков, А. А. Мазниченко // Системы управления и информационные технологии. 2023. № 2 (92). С. 92–95. EDN QOMDON

4. Баранчиков А. И., Федосова Е. Б. Применение методов data Mining для анализа и выявления закономерностей в реляционных базах данных // Радиотехнические и телекоммуникационные системы. 2023. № 2 (50). С. 40–45. https://doi.org/0.24412/2221-2574-2023-2-40-45. EDN CIBVDW

5. Набиуллин Д. А., Кононова В. В., Новикова С. В. Метод автоматизированной разметки больших данных с использованием нейронных сетей // Вестник Технологического университета. 2021. Т. 24, № 6. С. 103–107. EDN PJNLIK

6. Методы интеллектуального анализа текстовых данных для служб экстренного реагирования / А. А. Сабитов, Р. Н. Минниханов, М. В. Дагаева [и др.] // Математические методы в технике и технологиях – ММТТ. 2020. Т. 7. С. 84–87. EDN NMCBWD

7. Ломакина Л. С., Субботин А. Н. Классификация потоковых данных на основе байесовского критерия // Моделирование, оптимизация и информационные технологии. 2020. Т. 8, № 1 (28). С. 18. https://doi.org/10.26102/2310-6018/2020.28.034. EDN ULSSNK

8. Андреев А. В. Искусственный интеллект и его роль в обработке больших данных // Умная цифровая экономика. 2023. Т. 3, № 1. С. 65–69.

9. Лейн Х., Хапке Х., Ховард К. Обработка естественного языка в действии. СПб.: Питер, 2020. 576 с.

10. Баулина А. Р., Ресан М. Т., Янаева М. В. Системы текстового поиска, обработки и анализа естественного языка // Обществознание и социальная психология. 2022. № 9 (39). С. 101–104.

11. Иванова Г. С., Мартынюк П. А. Анализ методов извлечения информации из текстовых данных / Г. С. Иванова // Нейрокомпьютеры: разработка, применение. 2022. Т. 24, № 3. С. 18–28. https://doi.org/10.18127/j19998554-202203-02

12. Кадермятова Л. М., Тутубалина Е. В. Анализ моделей векторных представлений слов в задаче разметки семантических ролей в русскоязычных текстах // Электронные библиотеки. 2020. Т. 23, № 5. С. 1026–1043.

13. Анализ данных / С. В. Лейхтер, С. Н. Чуканов, И. С. Чуканов, И. В. Широков. Омск: Омский государственный университет им. Ф. М. Достоевского, 2022. 108 с. EDN WHSYZW

14. Phat H. N., Anh N. T. M. Vietnamese text classification algorithm using long short term memory and word2vec // Informatics and Automation. 2020. Vol. 19, N 6. P. 1255–1279. https://doi.org/10.15622/ia.2020.19.6.5. EDN MFDPBK

15. Огарок А. Л., Жаворонкова О. Г. Методы семантической обработки неструктурированной текстовой информации // Информатизация и связь. 2022. № 6. С. 44–48. https://doi.org/10.34219/2078-8320-2022-13-6-44-48

16. Огарок А. Л. Математическая модель процесса семантической обработки текстовой информации // Информатизация и связь. 2021. № 6. С. 87–91. https://doi.org/10.34219/2078-8320-2021-12-6-87-91

17. Попов О. Р., Гребенюк Е. В. Алгоритмы построения интеллектуальных систем обработки текстовой информации для задачи анализа мнений // Интеллектуальные ресурсы – региональному развитию. 2021. № 2. С. 104–110.

18. Казанцев А. А., Прохоров М. В., Худякова П. С. Обзор подходов к классификации текстов актуальными методами // Экономика и качество систем связи. 2021. № 1 (19). С. 57–67. EDN ZUJEVN

19. Семина Т. А. Анализ тональности текста: современные подходы и существующие проблемы // Социальные и гуманитарные науки. Отечественная и зарубежная литература. Серия 6: Языкознание. 2020. № 4. С. 47–63.

References

1. Kobyshev K.S., Molodyakov S.A. Analysis and classification of algorithms for extracting relations from text data. *Sovremennaya nauka: aktual'nye problemy teorii i praktiki. Seriya: Estestvennye i tekhnicheskie nauki = Modern Science: Current Problems of Theory and Practice. Series: Natural and Technical Sciences*. 2021;(5):71–79. (In Russ.) https://doi.org/10.37882/2223-2966.2021.05.15. EDN KXLLZK

2. Polyakov A.A., Fetisov M.V. Classification of algorithms for preliminary processing of text data for machine learning. *Tekhnologii inzhenernykh i informatsionnykh sistem = Technologies of Engineering and Information Systems*. 2021;(4):70–79. (In Russ.) EDN QROXYD

3. Sazonov M.A., Yakovlev A.V., Kozhanchikov M.O., Maznichenko A.A. Development of promising methods for searching and classifying text information from open sources on the Internet. *Sistemy upravleniya i informatsionnye tekhnologii = Control Systems and Information Technologies*. 2023;2(92):92–95. (In Russ.) EDN QOMDON

4. Baranchikov A.I., Fedosova E.B. Application of data mining methods for analyzing and identifying patterns in relational databases. *Radiotekhnicheskie i telekommunikatsionnye sistemy = Radio Engineering and Telecommunication Systems*. 2023;2(50):40–45. (In Russ.) https://doi.org/10.24412/2221-2574-2023-2-40-45. EDN CIBVDW

5. Nabiullin D.A., Kononova V.V., Novikova S.V. Method of automated tagging of big data using neural networks. *Vestnik Tekhnolo-gicheskogo universiteta = Bulletin of the Technological University*. 2021;24(6):103–107. (In Russ.) EDN PJNLIK

6. Sabitov A.A., Minnikhanov R.N., Dagaeva M.V., et al. Methods for intellectual analysis of text data for emergency response services. *Matema-ticheskie metody v tekhnike i tekhnologiyakh – MMTT = Mathematical Methods in Engineering and Technology – MMTT*. 2020;(7):84–87. EDN NMCBWD

7. Lomakina L.S., Subbotin A.N. Classification of streaming data based on the Bayesian criterion. *Modelirovanie, optimizatsiya i informatsionnye tekhnologii = Modeling, Optimization and Information Technologies*. 2020;8(1):18. (In Russ.) https://doi.org/10.26102/ 2310-6018/2020.28.034. EDN ULSSNK

8. Andreev A.V. Artificial intelligence and its role in processing big data. *Umnaya tsifrovaya ekonomika = Smart Digital Economy*. 2023;3(1):65–69. (In Russ.)

9. Lane H., Hapke H., Howard C. Natural language processing in action. St. Petersburg: Peter; 2020. 576 p.

10. Baulina A.R., Resan M.T., Yanaeva M.V. Systems for text search, processing and analysis of natural language. *Obshchestvoznanie i sotsial'naya psikhologiya = Social Science and Social Psychology*. 2022:9(39):101–104. (In Russ.)

11. Ivanova G.S., Martynyuk P.A. Analysis of methods for extracting information from text data. *Neirokomp'yutery: razrabotka, primenenie = Neurocomputers: Development, Application*. 2022;24(3):18–28. (In Russ.) https://doi.org/10.18127/j19998554-202203-02

12. Kadermyatova L.M., Tutubalina E.V. Analysis of models of vector representations of words in the problem of marking semantic roles in Russian-language texts. *Elektronnye biblioteki = Electronic Libraries*. 2020;23(5):1026–1043. (In Russ.)

13. Leichter S.V., Chukanov S.N., Chukanov I.S., Shirokov I.V. Data analysis. Omsk: Omskii gosudarstvennyi universitet im. F.M. Dostoevskogo; 2022. 108 p. (In Russ.) EDN WHSYZW

14. Phat H.N., Anh N.T.M. Vietnamese text classification algorithm using long short term memory and word2vec. *Informatics and Automation*. 2020;19(6):1255–1279. https://doi.org/10.15622/ia.2020.19.6.5. EDN MFDPBK

15. Ogarok A.L., Zhavoronkova O.G. Methods of semantic processing of unstructured text information. *Informatizatsiya i svyaz' = Informatization and Communication*. 2022;(6):44–48. (In Russ.) https://doi.org/ 10.34219/2078-8320-2022-13-6-44-48

16. Ogarok A.L. Mathematical model of the process of semantic processing of text information. *Informatizatsiya i svyaz' = Informatization and Communication*. 2021;(6):87–91. https://doi.org/ 10.34219/2078-8320-2021-12-6-87-91

17. Popov O.R., Grebenyuk E.V. Algorithms for constructing intelligent systems for processing text information for the problem of opinion analysis. *Intellektual'nye resursy – regional'nomu razvitiyu = Intellectual Resources for Regional Development*. 2021;(2):104–110.

18. Kazantsev A.A., Prokhorov M.V., Khudyakova P.S. Review of approaches to text classification using current methods. *Ekonomika i kachestvo sistem svyazi = Economics and Quality of Communication Systems*. 2021;1(19):57–67. (In Russ.) EDN ZUJEVN

19. Semina T.A. Analysis of text sentiment: modern approaches and existing problems. *Sotsial'nye i gumanitarnye nauki. Otechestvennaya i zarubezhnaya literatura. Seriya 6: Yazykoznanie = Social and Humanitarian Sciences. Domestic and Foreign Literature. Episode 6: Linguistics*. 2020;4:47–63. (In Russ.)

Информация об авторах / Information about the Authors

Ефанов Сергей Валерьевич, аспирант, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: nshysh@yandex.ru, ORCID: 0009-0009-3117-5595

Sergei V. Efanov, Post-Graduate Student, Southwest State University, Kursk, Russian Federation, e-mail: nshysh@yandex.ru, ORCID: 0009-0009-3117-5595

Иванова Елена Николаевна, кандидат технических наук, доцент кафедры вычислительной техники, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: verksel@mail.ru, ORCID: 0009-0003-4466-5928

Elena N. Ivanova, Candidate of Sciences (Engineering), Associate Professor of the Department of Computer Science, Southwest State University, Kursk, Russian Federation, e-mail: verksel@mail.ru, ORCID: 0009-0003-4466-5928

Чернецкая Ирина Евгеньевна, доктор технических наук, заведующий кафедрой вычислительной техники, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: white731@yandex.ru, ORCID: 0009-0009-8254-9606

Irina E. Chernetskaya, Doctor of Sciences (Engineering), Head of the Department of Computer Science, Southwest State University, Kursk, Russian Federation, e-mail: white731@yandex.ru, ORCID: 0009-0009-8254-9606