Раздел I. Алгоритмы обработки информации

УДК 004.912 DOI 10.18522/2311-3103-2024-4-6-14

В.В. Курейчик, П.С. Герасименко

ОСНОВНЫЕ ПОДХОДЫ К ИЗВЛЕЧЕНИЮ ТЕКСТОВОЙ ИНФОРМАЦИИ (ОБЗОР)

Данная статья посвящена обзору известных и современных подходов, методов и алгоритмов полнотекстового поиска. Описана краткая история решения задачи поиска в неструктурированных текстовых данных, её развитие и актуальность. Сформулирована основная задача поиска в текстовых данных. Приведено определение индекса базы данных. В общем виде определена целевая функция поисковой информационной системы и описаны возможные компромиссные вариации её параметров при решения различных прикладных задач. Приведена обобщённая архитектура современной поисковой информационной системы с разделением задачи поиска на две фазы: первичное извлечение релевантных записей и их последующее ранжирование для формирования окончательных результатов поиска. Даны базовые описания основных алгоритмов и методов полнотекстового поиска, таких как: поиск по термам (логический поиск), поиск с помощью деревьев и их разновидностей (B-деревья, UB-деревья, tries), поиск на основе n-грамм (в том числе поиск на основе частотного представления), использование векторной модели пространства (VSM), поиск на основе инвертированного (обратного) индекса, поиск с использованием аппарата нечёткой логики и биоинспирированных методов. Приведены основные достоинства и недостатки этих методов, описана их применимость в различных условиях, а также рассмотрены возможные методы оптимизации поиска текстовых данных для улучшения точности, скорости поиска и эффективности использования ресурсов. Представлены возможные перспективные направления в области решения задачи первичного извлечения информации. Приведены некоторые способы определения сходства текстовых записей для решения задачи ранжирования на основе аппарата нечёткой логики. Затронуты вопросы повышения релевантности первичного извлечения с помощью методов искусственного интеллекта, нейронных сетей, аппарата нечёткой логики и биоинспирированных методов, в частности методы расширения поискового запроса и/или расширения обрабатываемых текстовых записей. Описано влияние граничных условий построения поисковой системы на повышение её эффективности. В заключение статьи подводятся итоги обзора и обсуждаются перспективы дальнейшего развития различных методов полнотекстового поиска.

Полнотекстовый поиск; B-деревья; векторная модель пространства; обратный индекс; n-грамм индексирование; двухфазовый поиск текста; индексы; извлечение информации; ранжирование; нейронные сети; нечёткая логика; биоинспирированные алгоритмы.

V.V. Kureichik, P.S. Gerasimenko

BASIC APPROACHES TO EXTRACTING TEXTUAL INFORMATION (OVERVIEW)

This article is devoted to the review of known and modern approaches, methods and algorithms of full-text search. A brief history of the solution of the problem of search in unstructured text data, its development and relevance are described. The main task of search in text data is formulated. The definition of the database index is given. The target function of the search information system is defined in general terms and possible compromise variations of its parameters when solving various applied problems are described. A generalized architecture of a modern search information system is given with the division of the search problem into two phases: the primary extraction of relevant records and their subsequent ranking to form the final search results. The article provides basic descriptions of the main algorithms and methods of full-text search, such as: search by terms (logical search), search using trees and their varieties (B-trees, UB-trees, tries), search based on n-grams (including search based on frequency representation), use of the vector space model (VSM), search based on an inverted (reverse) index, search using the

apparatus of fuzzy logic and bioinspired methods. The main advantages and disadvantages of these methods are given, their applicability in various conditions is described, and possible methods for optimizing the search for text data to improve the accuracy, speed of search and efficiency of resource use are considered. Possible promising directions in the field of solving the problem of primary information extraction are presented. Some methods for determining the similarity of text records for solving the ranking problem based on the apparatus of fuzzy logic are given. The article touches upon the issues of increasing the relevance of primary extraction using artificial intelligence methods, neural networks, fuzzy logic and bioinspired methods, in particular methods for expanding the search query and/or expanding the processed text records. The influence of the boundary conditions of the search system construction on increasing its efficiency is described. In conclusion, the article summarizes the review and discusses the prospects for further development of various full-text search methods.

Full-text search; B-trees; vector space model; inverse index; n-gram indexing; two-phase text search; indexes; information extraction; ranking; neural networks; fuzzy logic; binned algorithms.

Введение. Поиск похожей текстовой информации в виде отдельных документов или записей в базах данных является актуальной задачей и представляет всё больший интерес по мере увеличения объёмов хранимых и постоянно генерируемых данных. В данной области ведутся интенсивные исследования, направленные как на оптимизацию и усовершенствование уже хорошо зарекомендовавших себя решений, так и на поиск новых подходов и алгоритмов.

Основная задача систем извлечения информации – определить и получить ту информацию, которая наилучшим образом связана с запросом пользователя. Поскольку релевантными могут быть несколько записей, результаты часто ранжируются в соответствии с их оценкой релевантности запросу пользователя.

Решение этой задачи развивалось от традиционного поиска (когда к тексту относятся как к набору символов, букв или строк различной длины) к осуществлению поиска по смыслу и до современных систем семантического поиска [1] (направленных на сопоставление смыслов текста и поискового запроса, расширенного контекстом самого пользователя). Отдельно выделим гипотезу о статистической семантике (statistical semantics hypothesis) [2]: статистические зависимости употребления слов человеком могут быть использованы для нахождения заложенного в них смысла, так как она лежит в основе ряда широко используемых алгоритмов (например, VSM или TF-IDF).

В конечном счёте большинство подходов к поиску по текстовым данным сводится к преобразованию исходного текста к какой-либо форме, которая позволит вычислить меру близости двух записей или документов. Лучше всего это заметно на попытках представить текст в виде вектора, где существует целый ряд правил представления текста в виде вектора, при этом важно, чтобы близкие по смыслу и содержанию документы преобразовывались в близкие векторы [3]. В таких подходах постоянно предлагают новые методы приведения текста к какой-либо другой, удобной для вычислений, форме и сами способы вычисления меры схожести этих представлений.

Основная цель информационного поиска – предоставить конечному пользователю информацию наиболее релевантную его запросу, поэтому любая задача неточного поиска подразумевает ранжирование первичных результатов по релевантности. Как правило, критерий релевантности задаётся отдельно. Некоторые исследования [4] показывают перспективность глубокого разделения поиска кандидатов для ранжирования и самого ранжирования, так если мы можем выделить небольшую группу из n кандидатов (где n существенно меньше, чем общее число N всех объектов поиска) с использованием максимально простого и эффективного алгоритма, то задача ранжирования выполняется уже на этой небольшой группе, что открывает перспективы для применения более ресурсоемких, но более точных алгоритмов. Обобщённая схема поисковой информационной системы с разделением фазы первичного отбора и ранжирования представлена на рис. 1.

Рассмотрим более подробно некоторые подходы к организации поиска в текстовых данных. Отметим, что большинство из них может быть использовано как для решения задачи первичного извлечения, так и ранжирования данных.

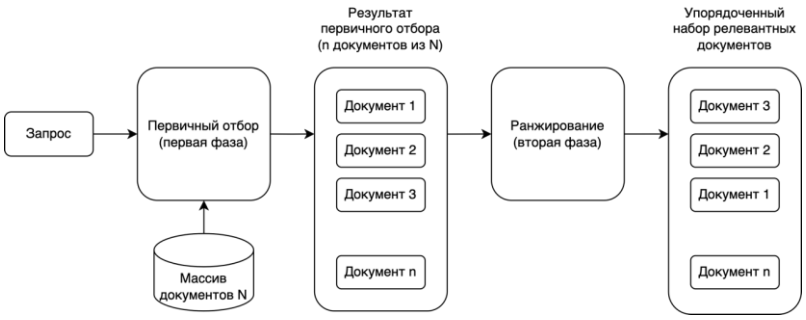

Рис. 1. Архитектура поисковой информационной системы

1. Анализ методов извлечение информации. Оценить качество системы извлечения информации можно с помощью целевой функции от трёх переменных: размер представления текстовой записи в памяти *s*, качество метода определения сходства двух (или более) записей *q* и время, затрачиваемое на работу, *t*.

$$F(s,q,t) = \begin{cases} S(s) \to min \\ Q(q) \to max \\ T(t) \to min. \end{cases}$$

Заметим, что время работы также можно разделить на составляющие: добавление, обновление, удаление, извлечение записей. При этом в идеале поисковая система работает с минимально возможным представлением *s*, которое находит наиболее похожие записи за минимальное время *t.* Тем не менее, для каждой конкретной практической задачи на переменные целевой функции *F* вводятся дополнительные ограничения. Например, для компактных устройств с ограничениями по памяти её размер может быть существенно ограничен. Однако для систем нахождения максимально релевантной (ценной) информации на первый план выходит максимизация качества определения сходства записей *q*. При этом для коммерческих решений важным является минимальное время нахождения необходимых записей *t*.

Совокупность функций преобразования и сравнения используется для построения индекса данных – компактного и удобного для обработки представление массива текстовых записей [5]. Индекс в большинстве случаев существенно меньше по размеру, чем исходные данные и существенно эффективнее с точки зрения обработки.

Создание индексов для полнотекстового поиска в базах данных является ключевым моментом для оптимизации производительности поиска. Рассмотрим достоинства и недостатки основных подходов к поиску в текстовых данных.

1.1. Обычный поиск по термам. Классический поиск по термам, также известный как логический поиск, представляет собой традиционный подход к поиску информации, который сопоставляет содержимое запроса содержимому документов [5, 6]. Это простой и легкий в реализации метод, но имеет некоторые ограничения, такие как неспособность обрабатывать синонимы, многозначность и контекст. Несмотря на эти ограничения, он широко используется в практике, особенно в небольших системах поиска.

1.2. Деревья и их разновидности. Использование различных вариантов деревьев (не всегда похожих на классические деревья, именно поэтому они называются в иностранной литературе *tries*) применяется в задачах поиска текста достаточно давно [5]. Безусловно современные деревья значительно сложнее и эффективнее, чем изначально предложенные B-деревья [7, 8]. Под деревом понимается такая структура данных, которая позволяет извлекать данные посредством весьма короткого пути, построенного на основе поискового запроса.

При использовании деревьев поиск данных происходит очень быстро и практически не зависит от размера набора данных, поскольку индекс и, следовательно, время поиска имеет логарифмическую сложность. Основание этого логарифма очень велико, по мень-

шей мере 1000 для современных архитектур хранения данных. Также дерево легко адаптируется к изменениям в хранимых данных, поскольку оно представляет собой самоорганизующуюся структуру, которая реорганизуется при каждой вставке или удалении данных. Таким образом, это позволяет осуществлять постоянную непрерывную обработку без перерывов на реорганизацию.

Один из современных вариантов реализации поиска с помощью деревьев представлен в [8]. Он демонстрирует высокие показатели по скорости поиска и обновлению самого дерева для данных, которые умещаются в оперативной памяти. Также современные деревья предлагают поиск не чувствительный к незначительным опечаткам/ошибкам в пользовательском запросе.

К основным недостаткам деревьев относят чрезмерное использование памяти и сложности с поддержанием оптимальной структуры дерева при интенсивных операциях вставки/обновления/удаления записей.

1.3. Инвертированное индексирование (обратный индекс). Инвертированный индекс – это структура данных, используемая в извлечении информации для эффективного и быстрого нахождения документов, содержащих определенные слова или термины. Идея инвертированных индексов для поиска была предложена в работе [9] и с тех пор этот подход активно используется в поисковых системах и постоянно модифицируется [10]. Идея инвертированного индекса состоит в следующем: для каждого слова/терма есть список документов, в которых это слово/терм используется. Он состоит из двух основных частей: словаря и списка обратных ссылок.

В словаре каждому уникальному слову или термину из всей коллекции документов присваивается уникальный идентификатор или токен. Этот словарь хранит отображение между словами и списками документов, в которых они встречаются.

Для каждого термина создается список обратных ссылок, который содержит идентификаторы (или другую информацию) всех документов, где встречается данный термин. Эти списки обеспечивают прямой доступ к документам, содержащим интересующие слова или термины.

Когда пользователь вводит поисковый запрос, система использует инвертированный индекс для быстрого определения документов, содержащих все или часть терминов из запроса. Это позволяет значительно ускорить процесс поиска и повысить его качество, особенно при работе с большими объемами текстовой информации.

Этот вид индекса широко используется в популярных поисковых системах, таких как Sphinx, Manticore, Lucene (ElasticSearch). Это широко используемый, эффективный и масштабируемый для больших объёмов данных подход. К основным недостаткам инвертированных индексов относят интенсивное использование памяти, обслуживание и обновление при этом являются ресурсоёмкими. Этот вид индекса хорошо работает для поиска точных совпадений, но для поиска по сложным фразам необходимо применять специальные техники.

1.4. Индексирование на основе n-грамм. Индексирование на основе n-грамм – это метод, используемый в информационном поиске и обработке естественного языка для эффективного поиска и извлечения текстовых документов на основе подстрок [11]. Термин «n-грамма» относится к последовательности из n элементов обрабатываемого текста. В контексте индексации эти «элементы» обычно представляют собой отдельные слова. Поиск на основе n-грамм состоит из четырёх этапов [12]:

1. Токенизация текста, разбивка его на отдельные слова или токены. В этом процессе как правило удаляются знаки пунктуации, слова приводятся к одному регистру, удаляются стоп слова и обрабатываются специальные случаи, такие как сокращения и слова с дефисами.

2. Генерация n-грамм. Текст разделяется на n-граммы. N-грамма – это последовательность из n токенов, если $n = 1$, то n-граммы – это отдельные слова, если $n = 2$, то n-граммы – пары последовательных слов, и так далее. Например, для предложения «Быстрый поиск текстовых данных», при $n = 2$, 2-граммы будут «Быстрый поиск», «поиск текстовых», «текстовых данных».

3. Индексация. Каждая n-грамма индексируется вместе с документами, в которых она встречается. Это может быть сделано с использованием структур данных, таких как хеш-таблицы или инвертированные индексы. Индекс присваивает каждой n-грамме идентификаторы всех документов, которые её содержат.

4. Обработка запроса. Когда пользователь вводит запрос, применяется та же самая токенизация и генерация n-грамм (1-3 этапы). После этого происходит поиск в индексе документов, содержащих n-граммы из запроса, и получается список всех документов, которые содержат предложенные набор *n*-грамм.

Индексация *n*-грамм имеет преимущество в точности перед поиском отдельных слов, так как позволяет работать со связками слов. Однако индексация n-грамм также имеет некоторые ограничения, такие как увеличенные требования к хранению и возможность появления шума от часто встречающихся n-грамм.

Эти подходы постоянно подвергаются модификациям и улучшениям [11, 12].

Ещё одним развитием этого метода индексирования является преобразование исходного текста к «частотному» виду [13]. Модификация метода состоит в разделении текста на n-граммы, для построения «спектрограммы» текстового документа с помощью преобразования Фурье.

1.5. Векторная модель пространства (Vector Space Model). Модель векторного пространства (Vector Space Model, VSM) – это математическая модель, используемая для информационного поиска в области обработки естественного языка и текстового анализа [3]. Она представляет документы и запросы в виде векторов в многомерном пространстве, где каждое измерение соответствует термину в словаре. Обобщённо процесс извлечения информации с помощью векторной модели пространства можно представить в виде следующих этапов [14]:

1. Создание словаря слов на основе документов. Для этого используют токенизацию текста, удаление стоп слов, приводят слова к базовым формам.

2. Каждый документ и поисковый запрос представляются в виде вектора в векторном пространстве. Размерности этих векторов соответствуют количеству терминов в словаре, а значение каждой размерности представляет собой некоторую меру важности или частоты соответствующего термина в документе или запросе. Перед построением векторов обычно применяются схемы взвешивания терминов к матрице документ-термин. Одна из популярных схем – TF-IDF (частота термина-обратная частота документа) [14], которая придает больший вес терминам, часто встречающимся внутри документа, но редко встречающимся во всем массиве документов. В [4, 15] приведены другие подходы к приведению текста к векторам.

3. После того как документы и запросы представлены в виде векторов, следующим шагом является расчет сходства между ними. Самая часто используемая метрика сходства – косинусное сходство, которое измеряет косинус угла между двумя векторами [3]. В [16, 17] приведены другие подходы к определению меры сходства.

4. Чтобы извлечь соответствующие документы для данного запроса, сходство между вектором запроса и каждым вектором документа вычисляется с использованием меры сходства. В результате мы получаем набор ранжированных документов, наиболее соответствующих запросу.

В целом, модель векторного пространства предоставляет гибкий подход для представления текстовых данных и извлечения соответствующей информации на основе мер сходства. Она широко используется в поисковых системах и для кластеризации документов. К недостаткам относят сложности с лексической неоднозначностью и вариативностью (например, омонимы всегда будут иметь ненулевое сходство, а документы со схожим значением, но с разным словарем терминов не будут связаны), также возникают сложности с выделением слов, специфичных для предметной области, так как они существенно реже встречаются в тексте и поэтому им приходится увеличивать вес при сравнении с помощью дополнительных методов.

1.6. Использование нейросетей, глубокое обучение. Широкое применение для решения задач семантического поиска в последнее время приобрели различные виды нейронных сетей [18–20]. Они показывают очень хорошие результаты в области поиска похожих по смыслу текстов даже с минимальными пересечениями на уровне отдельных слов. Зачастую нейросети используют как сложный классификатор, который позволяет отнести пользовательский запрос к какому-либо классу и сравнить его с классом интересующего нас документа/записи. Кроме того, с помощью нейронных сетей можно расширить поисковый запрос и/или добавляемую в хранилище запись [21–23], что позволит включить в результаты поиска в том числе записи, которые не содержат слова поискового запроса, но отвечают его смыслу.

Одним из подходов к семантическому поиску является представление текстовых документов и запросов в непрерывном векторном пространстве (аналогично векторной модели, описанной выше) с помощью свёрточных (англ. convolutional neural network, CNN) и рекурентных нейронных сетей (англ. recurrent neural network, RNN) [23].

Кроме того, механизмы, основанные на внимании, такие как архитектура трансформер [24], используются для улучшения способности систем выделять и обрабатывать важные части как самого запроса, так и документов. В ряде работ [25] было показано, что использование предварительно обученных языковых моделей, таких как BERT и GPT-4, повышает производительность систем извлечения информации, обеспечивая лучшее понимание семантики и контекста на естественном языке.

Обработка текста с помощью нейронных сетей является одним из самых перспективных направлений семантического сравнения. Однако, одним из недостатков применения нейронных сетей является необходимость их обучения, то есть решения дополнительных весьма непростых задач по подбору обучающих данных и самой стратегии обучения.

1.7. Нечёткий поиск. Нечётка логика может быть применена в системах информационного поиска для работы с неточными или двусмысленными запросами и улучшения релевантности результатов поиска [26]. Большой интерес представляет использование нечёткой логики для определения меры сходства документов независимо от формы их представления (вектор, n-граммы, инвертированные индексы). Введение нечёткой логики позволяет улучшить ранжирование результатов поиска, учитывая степень схожести между терминами запроса и содержимым документа. Вместо бинарных совпадений (точное совпадение или его отсутствие), использование нечёткой логики позволяет получать частичные совпадения, где документы, содержащие термины, сходные с запросом, получают более высокие оценки релевантности, что позволяет извлекать документы, тесно связанные с информационными потребностями пользователя, но не обязательно точно соответствующие терминам запроса [27]. Также в [27] предлагается заменить метрики требованием к нечеткому сходству, удовлетворяющему свойству транзитивности с настраиваемым нечетким конъюнктором.

Также нечёткую логику используют для расширения запросов, учитывая синонимы, родственные термины или даже приблизительные совпадения с исходными терминами запроса [28]. Это особенно эффективно при работе с полисемией или синонимией, так как позволяет извлекать семантически связанные документы, не содержащие точных терминов, указанных в поисковом запросе.

Отметим, что данный подход позволяет совмещать нескольких источников доказательств сходства документов, таких как текстовое содержание, метаданные, контекстная информация и любые другие. Агрегируя доказательства из разных источников с использованием нечётких операторов, система может принимать более обоснованные решения о релевантности документов по отношению к заданному запросу, что потенциально приводит к более точным и всесторонним результатам поиска.

1.8. Биоинспирированные алгоритмы. Существует ряд работ, посвящённых использованию биоинспирированных методик для решения задач извлечения информации. Так, например, генетические алгоритмы используют для поиска наилучшего набора документов, связанных с ключевыми словами из запроса пользователя. В [29] приведён обзор применения различных биоинспирированных методов для решения задач классификации и кластеризации текстовых данных.

Ещё одним направлением использования биоинспирированных методов является расширение (оптимизация) запросов [30] – добавление, удаление или изменение исходных запросов. Они применяются для устранения неточностей в системах поиска информации, которые возникают из первоначального запроса, предоставленного пользователями. Тем не менее, в некоторых работах отмечается отсутствие значимых улучшений показателей поиска при использовании генетических алгоритмов для расширения поисковых индексов.

2. Направления оптимизации поиска и улучшения его качества. В ряде случаев разбиение задачи поиска на две фазы: поиск документов для ранжирования и непосредственно ранжирование, позволяет повысить эффективность поисковых систем. Основной задачей первой фазы является нахождение максимально полного подмножества записей релевантных запросу. При этом основной задачей второй фазы является максимально точное ранжирование результатов первой фазы. Безусловно, у этого подхода есть недостаток, так как чрезмерное упрощение предварительного поиска может привести к исключению из выдачи записей, которые могут быть близкими по смыслу, но существенно различаться по лексическому составу.

Актуальным является поиск новых форм и способов приведения текстовых документов к максимально удобному для поиска и извлечения виду. Также по мере развития аппаратной базы совершенствуются структуры данных, используемых для решения задач извлечения информации.

Также для улучшения качества поиска применяют различные техники расширения как самих анализируемых текстов, так и запросов пользователя. Под расширением понимается добавление новых слов в исходные данные по определённым правилам. Эти слова не встречаются в исходном тексте или запросе, но являются схожими, синонимичными или релевантными по другим критериям. Такие подходы позволяют находить релевантные записи по смыслу даже если они не содержат слова из запроса пользователя.

Кроме того, на этапе ранжирования сравнительно небольшой группы документов применяются алгоритмы машинного обучения, нейронные сети, учитывающие предпочтения конкретного пользователя поисковой системы и конкретную практическую задачу её владельца.

При некоторых граничных условиях, накладываемых на предметную область, добиваются существенных преимуществ с точки зрения быстродействия и размеров индексов. Яркий пример – разграничение двух видов поиска: поиск в документах (как правило достаточно больших текстовых массивах) и поиск в записях баз данных (в большинстве случаев текстовые данные в них представлены относительно короткими фрагментами). Если строить индекс исходя из работы только с небольшими фрагментами текста, то требования к нему снижаются, что позволяет получить существенный выигрыш в быстродействии.

Заключение. В данной работе представлен обзор и анализ современных моделей, методов и алгоритмов полнотекстового поиска. Рассмотрена обобщённая архитектура поисковой информационной системы, предложена целевая функция. В результате проведенного обзора и анализа выявлены основные достоинства и недостатки этих методов. Представлены возможные перспективные направления в области первичного извлечение информации. Предложены направления для повышения релевантности первичного извлечения на основе методов искусственного интеллекта и нейронных сетей, а также методы расширения поискового запроса и/или расширения обрабатываемых текстовых записей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Guha R.V., McCool R., Miller E.* Semantic search, *Proceedings of the 12th International Conference on World Wide Web, Budapest,* 2003, pp. 700-709.
2. *Weaver W.* Machine Translation of Languages. MIT Press, Cambridge, MA, Reprinted from a memorandum written by Weaver in 1949, 1955.
3. *Salton G., Wong A., Yang C.S.* A vector space model for automatic indexing, *Communications of the ACM,* 1975, Vol. 18, No. 11, pp. 613-620.
4. *Rygl J., Pomikálek J., Řehůřek R., Růžička M.* Semantic Vector Encoding and Similarity Search Using Fulltext Search Engines, *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017.

5. *Baeza-Yates R., Ribeiro-Neto B.* Modern Information Retrieval. ACM Press New York, 1999.
6. *Manning C.D., Raghavan P., Sch"utze, H.* Introduction to Information Retrieval. Cambridge University Press, 2008.
7. *Yan W., Zhang X.* A Concise Concurrent B + -Tree for Persistent Memory, *ACM Transactions on Architecture and Code Optimization,* 2023, Vol. 21 (2).
8. *Leis V., Kemper A., Neumann T.* The adaptive radix tree: ARTful indexing for main-memory databases, *2013 IEEE 29th International Conference on Data Engineering (ICDE), Brisbane, QLD, Australia,* 2013, pp. 38-49.
9. *Salton G.* Information Retrieval: Data Structures and Algorithms. Reading, Massachussetts: Addison-Wesley, 1989.
10. *Giulio Ermanno P., Rossano V.* Techniques for Inverted Index Compression, *ACM Computing Surveys,* 2020, Vol. 53, pp. 1-36.
11. *Miller E., Shen D., Liu J., Nicholas C*. Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System, *Journal of Digital Information,* 2000, Vol 1, No 5.
12. *Srinivasa K., Devi B.* GPU Based N-Gram String Matching Algorithm with Score Table Approach for String Searching in Many Documents, *Journal of The Institution of Engineers (India) Series B,* 2017, Vol. 98, pp. 467-476.
13. *Kudryavtsev K.* Search in text documents based on N-grams and Fourier window transformation, *Computer Science,* 2014, Vol. 65, pp. 871-880.
14. *Jones K.* A Statistical Interpretation of Term Specificity in Retrieval, *Journal of Documentation,* 2004, Vol. 60, pp. 493-502.
15. *Chebil W., Soualmia L.* Improving semantic information retrieval by combining possibilistic networks, vector space model and pseudo-relevance feedback, *Journal of Information Science,* 2023.
16. *Eminagaoglu M.* A new similarity measure for vector space models in text classification and information retrieval, *Journal of Information Science,* 2020, Vol. 48.
17. *Singh A., Yadav A., Rana A.* K-means with Three different Distance Metrics, *International Journal of Computer Applications,* 2013, Vol. 67, pp. 13-17.
18. *Nogueira R., Yang W., Lin J., Cho K.* Document expansion by query prediction, *arXiv preprint arXiv:1904.08375,* 2019.
19. *Mao Y., He P., Liu X., Shen Y., Gao J., Han J., Chen W.* Generation-augmented retrieval for open-domain question answering, *arXiv preprint arXiv:2009.08553,* 2020.
20. *Yan M., Li C., Bi B., Wang W., Huang S.* A unified pretraining framework for passage ranking and expansion, *Proceedings of the AAAI Conference on Artificial Intelligence,* 2021, Vol. 35, pp. 4555-4563.
21. *Efron M., Organisciak P., Fenlon K.* Improving retrieval of short texts through document expansion, *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval,* 2012, pp. 911–920.
22. *Agirre E., Arregi X., Otegi A.* Document expansion based on WordNet for robust IR, *Coling 2010: Posters,* 2010, pp. 9-17.
23. *Shen Y., He X., Gao J., Deng L.* Latent semantic models with deep neural networks for information retrieval, *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM,* 2014, pp. 269-278.
24. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I.* Attention Is All You Need, *Advances in Neural Information Processing Systems,* 2017, pp. 6000-6010.
25. *Devlin J., Chang M.W., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* 2019, pp. 4171–4186.
26. *Chimah J., Ude F.* Current trends in information retrieval systems: review of fuzzy set theory and fuzzy Boolean retrieval models, *Journal of Library Services and Technologies,* 2020, Vol. 2, pp. 48-56.
27. *Bartos T., Eckhardt A., Skopal T.* Fuzzy approach to non-metric similarity indexing, *Proceedings - 4th International Conference on SImilarity Search and APplications, SISAP 2011,* 2011, pp. 115-116.
28. *Mounira C., Jedidi A., Gargouri F.* Semantic/Fuzzy Information Retrieval System, *International Journal of Information Technology and Web Engineering,* 2017.
29. *Abualigah L., Hanandeh E.* Applying Genetic Algorithms to Information Retrieval Using Vector Space Model, *International Journal of Computer Science, Engineering and Applications,* 2015, Vol. 5, pp. 19-28.
30. *Russell A.* A Genetic Algorithm for Query Optimization, *Department of Computer and Information Sciences University of Strathclyde, Glasgow August 26,* 2019.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Курейчик Владимир Викторович – Южный федеральный университет; e-mail: vkur@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; кафедра систем автоматизированного проектирования им. В.М. Курейчика; зав. кафедрой; д.т.н.; профессор.

Герасименко Петр Сергеевич – e-mail: pege@sfedu.ru; тел.: 88634371651; кафедра систем автоматизированного проектирования им. В.М. Курейчика; аспирант.

Kureichik Vladimir Victorovich – Southern Federal University; e-mail: vkur@sfedu.ru; Taganrog, Russia; phone: +78634371651; the department of Computer Aided Design; head of department; dr. of eng. sc.; professor.

Gerasimenko Petr Sergeevich – e-mail: pege@sfedu.ru; phone: +78634371651; the department of Computer Aided Design; post graduate student.

УДК 004.896 DOI 10.18522/2311-3103-2024-4-14-30

Б.К. Лебедев, О.Б. Лебедев, М.А. Ганжур

БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ ПЛОТНОЙ УПАКОВКИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАСКРОЯ ПОЛУОГРАНИЧЕННОЙ ПОЛОСЫ

Предлагается архитектура и методология раскроя-упаковки полуограниченной полосы на основе методов биоинспирированного поиска. В основе подхода к декомпозиции обшей задачи упаковки и методологии формированию карт раскроя лежат эвристики уровневого подхода к упаковке полосы. Архитектура сформирована на основе декомпозиции общей задачи и включает 5 основных секций: управление процессом поиска; формирования блоков; формирование контейнеров; компакция контейнеров; заполнения полосы контейнерами. Упаковка ориентирована на двухуровневый раскрой полосы. На первом уровне путем гильотинного разреза выполняется раскрой на контейнеры. На втором уровне два варианта раскроя: путем гильотинного или путем не гильотинного разреза выполняется раскрой контейнеров на детали (элементы прямоугольной формы). Упаковка выполняется путем последовательного заполнения уровней полосы контейнерами. В основу методологии раскроя-упаковки полуограниченной полосы положен иерархический подход снизу вверх. Задача, решаемая на первом уровне иерархии, заключается в формировании множества блоков B одинаковой ширины на базе исходного набора A прямоугольников, включаемых в блоки. Для решения поставленной задачи авторами разработан биоинспирированный алгоритм одномерной упаковки элементов в одинаковые блоки. На втором уровне иерархии решается задача распределения блоков по контейнерам. Все контейнеры и блоки имеют одинаковую ширину D, равную ширине полосы. В каждом контейнере помещаются два блока. Задача распределения блоков по контейнерам сведена к задаче нахождения максимального паросочетания минимальной стоимости. В отличие от канонической метаэвристики муравьиного алгоритма в работе агентом на графе поиска решений строится максимальная клика, которая является интерпретацией решения. На третьем уровне иерархии решается задача компакции контейнеров. Процесс распределения блоков по контейнерам сопровождается процедурой сжатия каждой пары блоков, назначаемых в один контейнер. Целью компакции является минимизация общей площади контейнера путем плотного размещении блоков. Компакцию последовательно проводят во всех контейнерах. На четвертом уровне иерархии решается задача заполнения полосы контейнерами. В качестве модели для представления решения на графе поиска решений служит клика. Разработана база данных коллективной эволюционной памяти. Разработана методика формирования феромоновых точек и структур данных коллективной эволюционной памяти. Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и сети Интернет. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 3-5%. Временная сложность алгоритма, полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ($BCA \approx O(n^2)$).

Раскрой-упаковка; полуограниченная полоса; декомпозиция; методология; задача об упаковке в контейнеры; компакция; поисковая оптимизация; роевой интеллект; муравьиная колония; адаптивное поведение.