20. CVE-2022-23724. Available at: https://nvd.nist.gov/vuln/detail/CVE-2022-23724 (accessed 26 October 2024).
21. CVE-2023-27121. Available at: https://nvd.nist.gov/vuln/detail/CVE-2023-27121 (accessed 26 October 2024).

Статью рекомендовал к опубликованию д.т.н., профессор А.Н. Целых.

Машкина Ирина Владимировна – Уфимский университет науки и технологий; e-mail: profmashkina@mail.ru; г. Уфа, Россия; тел.: +79279277089; кафедра вычислительной техники и защиты информации; д.т.н.; профессор.

Уразаева Айгуль Маратовна – e-mail: ajgul.urazaeva2017@yandex.ru; кафедра вычислительной техники и защиты информации; студент.

Mashkina Irina Vladimirovna – Ufa University of Science and Technology; e-mail: profmashkina@mail.ru; Ufa, Russia; phone: +79279277089; the Department of Computer Science and Information Security; dr. of eng. sc.; professor.

Urazaeva Aigul Maratovna – e-mail: ajgul.urazaeva2017@yandex.ru; the Department of Computer Science and Information Security; student.

УДК 004.853 DOI 10.18522/2311-3103-2024-5-88-102

Е.М. Герасименко, Ю.А. Кравченко, Д.А. Шаненко

АЛГОРИТМ ПОИСКА И ПРИОБРЕТЕНИЯ ЗНАНИЙ НА ОСНОВЕ ТЕХНОЛОГИЙ ОБРАБОТКИ И АНАЛИЗА ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ*

Статья посвящена решению актуальной научной проблемы повышения эффективности обработки и анализа текстовой информации при решении задач поиска и приобретения знаний. Актуальность данной задачи связана с необходимостью создания эффективных средств обработки накапливаемого огромного количества слабо структурированных данных, содержащих важные, иногда скрытые знания, необходимые для построения эффективных систем управления сложными объектами различной природы. Предлагаемый автором алгоритм поиска и приобретения знаний при обработке и анализе текстовой информации, отличается применением низкоуровневых детерминированных правил, позволяющих провести качественное упрощение текста на основе исключения из текстовой информации слов, инвариантных к смыслу. Алгоритм опирается на доменную проработку, позволяющую сформировать списки доменно-специфичных слов, что позволяет обеспечить высокое качество упрощения текста. В данной задаче исходными данными являются потоки текстовой информации (описание профилей), извлеченных из онлайн платформ для рекрутинга, выходная информация представляется предложениями, сформированными в виде тройки «субъект-глагол-объект», отражающих гранулы знаний, полученных в процессе обработки текста. Использование данного порядка единиц, составляющих предложение, обусловлено тем фактом, что данный порядок наиболее распространен в русском языке, хотя в самих текстах возможны иные вариации порядка без потери общего смысла. Основная идея алгоритма заключается в разбиении большого корпуса текста на предложения с последующей фильтрацией полученных предложений на основании введенных пользователем ключевых слов. В последствии предложения разделяются на компоненты и упрощаются в зависимости от вида поступившей компоненты (глагольная, именная). В качестве примера в данной работе использовалась сфера маркетинга, а ключевыми словами выступили «социальные сети». Автором разработан алгоритм поиска и приобретения знаний на основе технологий обработки и анализа текстов на естественном языке, а также была выполнена программная реализация предложенного алгоритма. В качестве методов оценки эффективности использовался ряд метрик: индекс Флэша-Кинкейда; индекс Кол-

* Исследование выполнено за счет гранта Российского научного фонда № 22-71-10121, https://rscf.ru/project/22-71-10121/ в Южном федеральном университете.

ман-Лиау; автоматический индекс удобочитаемости. Проведенные вычислительные эксперименты подтвердили эффективность предложенного алгоритма по сравнению с аналогами, использующими нейронные сети для решение подобных задач.

Обработка текстовой информации; поиск знаний; извлечение знаний; обработка естественного языка; извлечение компонент; упрощение текста.

E.M. Gerasimenko, Yu.A. Kravchenko, D.A. Shanenko

ALGORITHM FOR SEARCHING AND ACQUISITION OF KNOWLEDGE BASED ON TECHNOLOGIES FOR PROCESSING AND ANALYZING TEXTS IN NATURAL LANGUAGE

The article is devoted the topical scientific problem of increasing the efficiency of processing and analyzing text information when solving problems of searching and acquiring knowledge. The relevance of this task is related to the need to create effective means of processing the accumulated huge amount of poorly structured data containing important, sometimes hidden knowledge that is necessary for building effective control systems for complex objects of different nature. The algorithm of search and knowledge acquisition in processing and analyzing textual information proposed by the author is characterized by the use of low-level deterministic rules that allow for qualitative text simplification based on the exclusion of words invariant to meaning from textual information. The algorithm relies on domain elaboration that allows to create lists of domain-specific words, which allows for high quality text simplification. In this task, the input data are streams of textual information (profile descriptions) extracted from online recruiting platforms; the output information is represented by sentences formed in the form of a triple "subject-verb-object", reflecting the granules of knowledge obtained during text processing. The use of this order of units constituting a sentence is due to the fact that this order is the most widespread in the Russian language, although other variations of the order are possible in the texts themselves without losing the general meaning. The main idea of the algorithm is to split a large corpus of text into sentences, then filter the resulting sentences based on the keywords entered by the user. Subsequently, the sentences are further split into components and simplified depending on the type of received component (verbal, nominal). The field of marketing was used as an example in this work, and the keywords were "social media". The author has developed an algorithm for for knowledge search and acquisition based on natural language text processing and analysis technologies, and a software implementation of the proposed algorithm has been performed. A number of metrics were used as efficiency evaluation methods: the Flash-Kincaid index; the Coleman-Liau index; and the automatic readability index. The conducted computational experiments have confirmed the effectiveness of the proposed algorithm in comparison with analogues that use neural networks to solve similar problems.

Text information processing; knowledge retrieval; knowledge extraction; natural language processing; component extraction; text simplification.

Введение. Задачи поиска знаний при обработке и анализе текстовой информации востребованы во многих типах прикладных информационных систем. Традиционные методы поиска знаний основаны на правилах, тезаурусах, машинном обучении с учителем и требуют наличия достаточных априорных знаний об исследуемой предметной области в виде лингвистических ресурсов, обработанных корпусов текста, словарей и грамматик.

Для создания правил и грамматик отдельных предметных областей разработано достаточное количество инструментов, которые постоянно применяются в информационных системах: CPSL [1]; Jape [2]; UIMA Ruta [3]; ABBYY Compreno [4] и др.

Основным недостатком перечисленных инструментов являются значительные трудозатраты на разработку правил и грамматик.

Поиск и извлечение знаний играют решающую роль в области искусственного интеллекта, позволяя компьютерам понимать данные на уровне, сравнимом с человеком [5]. Также они помогают извлекать ценные сведения, обнаруживать скрытые закономерности и придавать смысл большим объемам данных. Все это может быть использовано в различных отраслях, таких как здравоохранение, финансы, производство и др., для стимулирования инноваций, совершенствования процессов и получения конкурентоспособных преимуществ.

Извлечение знаний включает в себя ряд этапов, направленных на извлечение и преобразование данных в значимые единицы знаний. Эти этапы обычно включают предварительную обработку данных, выбор функций, интеллектуальный анализ данных (data mining) и представление знаний.

Спектр методов, используемых для решения задач извлечения знаний, достаточно обширен [6–14]. Это относительно новая тенденция, возникшая в связи с необходимостью масштабирования систем извлечения текстовой информации до очень больших объемов данных, которые представляют собой смесь огромного количества различных тем.

Актуальность работы связана с необходимостью создания эффективных средств обработки накапливаемого огромного количества слабо структурированных данных, содержащих важные, иногда скрытые знания, необходимые для построения эффективных систем управления сложными объектами различной природы.

1. Постановка задачи. При обработке естественного языка важно выявить и извлечь наиболее релевантные и значимые слова (или фразы) из заданного текста. В данном процессе главную роль играют парсеры, которые анализируют синтаксическую структуру текста и помогают выявить ключевые компоненты, отражающие важные понятия.

Синтаксический анализ структуры позволяет сосредоточиться на конкретных синтаксических единицах при извлечении знаний из текстов. Подобные единицы часто несут в себе важную информацию о субъекте, объекте или действии [15].

Синтаксический анализ зависимостей устанавливает отношения между подлежащим, сказуемым и дополнением, а также находит другие значимые синтаксические зависимости, которые вносят вклад в общий смысл предложения (например, в сложноподчиненных предложениях придаточные дополняют главное) [16–21]. Из данных зависимостей можно извлечь знания, особенно если слова играют важную роль в структуре предложения.

Упрощение предложений. Синтаксический анализ позволяет получить сведения о конкретных синтаксических единицах, но зачастую они могут быть нагружены связанными с ними «дополнениями» (например, прилагательными или наречиями) [22–27]. Попытка работать с полноценным предложением, извлечь из него элементы знаний, занимает больше времени, по сравнению с упрощенными компонентами.

Поэтому для повышения эффективности обработки текстовой информации требуется представить следующие решения:

1) определить наиболее подходящую структуру алгоритма и реализацию модулей алгоритма;

2) реализовать алгоритм поиска и приобретения знаний, наиболее подходящий под условия дальнейшей эксплуатации.

Оценка эффективности работы алгоритма будет проводиться при помощи следующих метрик:

- Индекс Флэша-Кинкейда.
- Индекс Колман-Лиау.
- Автоматический индекс удобочитаемости.

Опишем основные метрики более подробно.

Индекс Флэша-Кинкейда. Применяется для измерения уровня сложности текста на основе длины слов и предложений.

Значения варьируются от 0 до 18, где 18 соответствует наиболее сложному тексту. Расчет индекса производится по формуле:

$$F_1 = FKGLF = 0.39\frac{total\ words}{total\ sentences} + 11.8\frac{total\ syllables}{total\ words} - 15.59, \quad (1)$$

где total words – общее количество слов в тексте, total sentences – общее количество предложений в тексте, a total syllables – общее количество символов текста.

Индекс по шкале принято распределять следующим образом:

а) 0 – 1 – текст очень легко читается;

б) 1 – 5 – текст легко читается;

в) 5 – 11 – достаточно легко читается. Стандартный разговорный язык;

г) 11 – 18 – текст сложен для восприятия;

д) >18 – текст очень трудно читается.

Индекс Колман-Лиау. Основывается на анализе количества букв в слове и слов в предложении, опирается на количество символов, а не слогов в слове, поскольку символы легче и точнее подсчитываются компьютерными программами, чем слоги. Расчет индекса производится по формуле:

$$F_2 = CLI = 0.0588\mathrm{L} - 0.296S - 15.8, \quad (2)$$

где L – среднее количество букв на 100 слов, а S – среднее количество предложений на 100 слов.

Индекс по шкале принято распределять следующим образом:

а) 0 – 1 – текст очень легко читается;

б) 1 – 5 – текст легко читается;

в) 5 – 8 – достаточно легко читается. Стандартный разговорный язык;

г) 8 – 11 – текст сложен для восприятия;

д) >11 – текст очень трудно читается.

Автоматический индекс удобочитаемости. В отличие от индекса Колман-Лиау, подсчитывает не слоги, а символы. От количества символов зависит сложность слова, помимо этого ведется подсчет предложений. Расчет индекса производится по формуле:

$$F_3 = ARI = 4.71 \times \frac{C}{W} + 0.5 \times \frac{W}{S} - 21.43, \quad (3)$$

где C – среднее количество букв и цифр в тексте, W – количество слов в тексте, S – количество предложений в тексте.

Индекс по шкале принято распределять следующим образом:

а) 0 – 1 – текст очень легко читается;

б) 1 – 5 – текст легко читается;

в) 5 – 8 – достаточно легко читается. Стандартный разговорный язык;

г) 8 – 11 – текст сложен для восприятия;

д) >11 – текст очень трудно читается.

Целевой функцией алгоритма является минимизация значений описанных мер читабельности текста.

$$F_i \rightarrow min\,,$$

где i – номер метрики.

Каждая из представленных мер читабельности имеет порог, после которого текст становится очень трудным для восприятия (нечитабельным). Показатели удобочитаемости являются алгоритмическими эвристиками, используемыми для оценки удобочитаемости. Стоит отметить, что удобочитаемость не является показателем качества письма и что эти эвристические методы являются лишь оценкой понятности отрывка.

2. Обобщенная схема процесса поиска и приобретения знаний. Для проведения вычислительных экспериментов была выбрана предметная область, связанная с установлением деловых контактов. Предложенная авторами версия алгоритма представляет собой набор модулей для использования в информационно-поисковой системе в сфере маркетинговых исследований.

Обобщенная схема процесса обработки текста алгоритмами представлена на рис. 1.

На обработку поступает корпус текста. В самом начале текст разбивается на отдельные предложения, после этого происходит фильтрация получившихся предложений по заданным ключевым словам. После фильтрации на дальнейшую обработку поступают только отобранные на предыдущем шаге предложения. Если предложения сложные (имеют в себе более одной пары подлежащего и сказуемого), то они дополнительно разделяются на простые предложения с сохранением отношений (сохраняется информация о том, что предложение было сложносочиненными или сложноподчиненными). После этого предложения разделяются на ключевые и дополняющие компоненты. На заключительном этапе происходит упрощение компонент (вырезаются прилагательные, наречия). После этого полученные результаты работы алгоритма вносятся в базу данных и доступны для дальнейшей работы.

Рис. 1. Обобщение процесса обработки текстов

На рис. 2 представлена схема модуля разбиения текста на предложения.

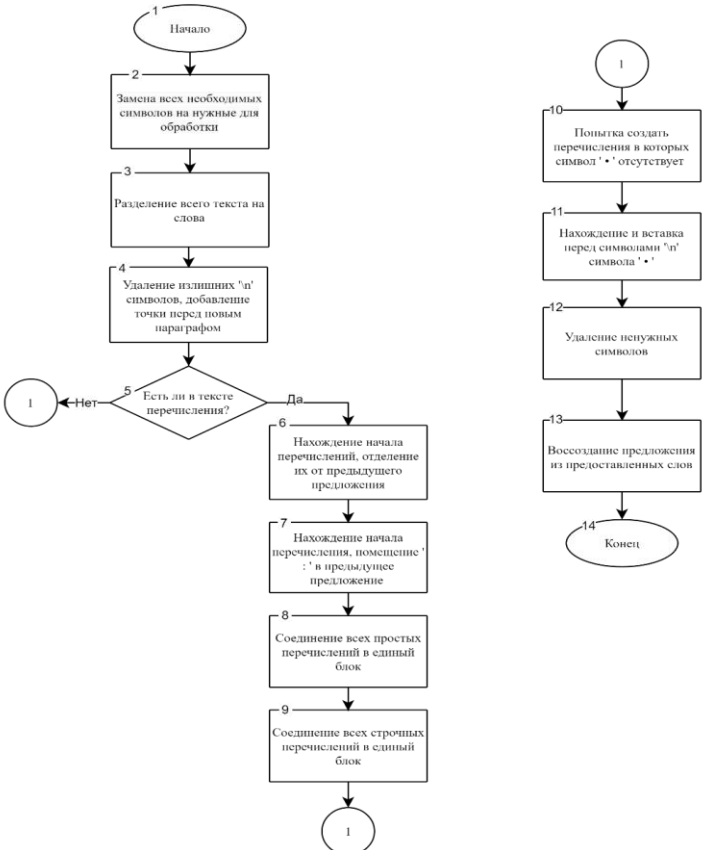

Рис. 2. Схема разбиения текста на предложения

Модуль фильтрации предложений принимает в качестве входных данных список текстов (предложений), список ключевых слов и параметр to_lowercase (необязателен, установлено значение «Истина» по умолчанию для перевода слов в нижний регистр). Он перебирает список текстов и перечисляет их (т.е. присваивает каждому тексту идентификаторы). Для каждого текста вызывает функцию поиска ключевых слов, чтобы попытаться получить список ключевых слов. Если ключевые слова найдены, создает экземпляр специального файла, передает ему соответствующую информацию (текст, найденные ключевые слова и идентификатор текста) и возвращает список экземпляров специального файла.

На рис. 3 представлена схема модуля фильтрации предложений. Модуль разбиения сложных предложений опирается на полученные от spaCy отношения зависимостей. Отношения зависимостей, используемые для идентификации и разделения предложений: cconj для сложносочинённых и acl:relcl, acl, xcomp, ccomp для сложноподчиненных предложений.

Рис. 3. Схема фильтрации предложений

Сначала определяется корень предложения (подразумевается, что этим словом является глагол), а затем проверяется наличие зависимости между корнем и другими частями предложения. В случае если корнем предложения стала другая часть речи, проводится обработка неправильного корня слова, в процессе которой проверяется наличие у корня слова зависимостей вида acl, relcl или conj. Функция работает рекурсивно: сначала предложения разделяются на одном уровне (разбиваются по ширине), а затем происходит переход на более глубокий уровень (разбиваются по глубине). Рекурсия вызвана тем фактом, что сложные предложения также могут быть составными (сложносочиненное предложение будет главным для зависимого).

На рис. 4 представлена схема модуля разбиения сложных предложений.

Модуль деления на компоненты разделяет части простого предложения, полученного на предыдущем этапе, на определенные грамматические компоненты на основе зависимостей spaCy. Поскольку предложение представлено в виде глагола и его зависимых элементов, задача рассматривается как присвоение определенных меток зависимым от глагола элементам.

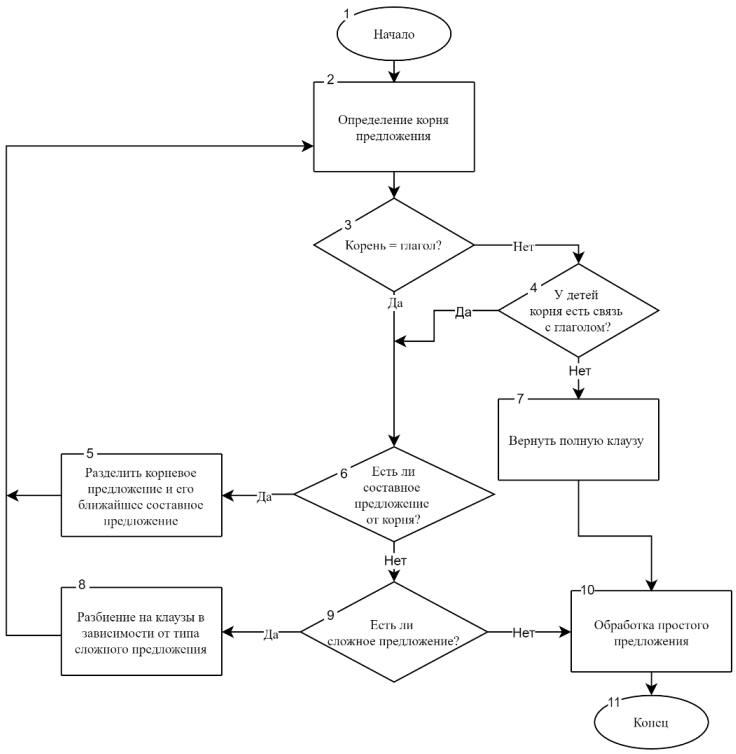

Рис. 4. Схема разбиения сложных предложений

На рис. 5 представлена схема модуля деления на компоненты предложений.

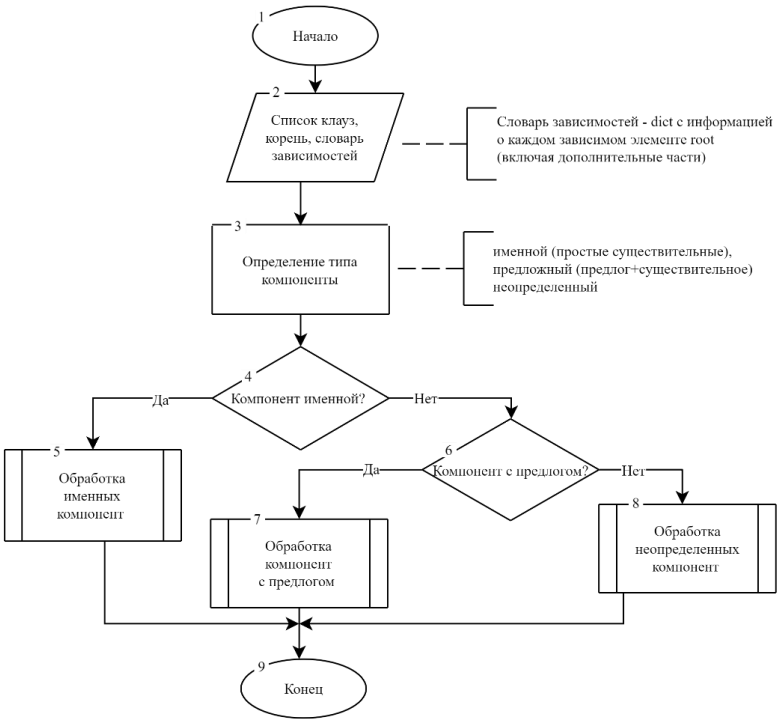

Рис. 5. Схема деления на компоненты

Модуль упрощения предназначен для упрощения глагольных и именных компонент текста. Он оценивает каждый компонент на основе его длины, типа и специфических лингвистических особенностей, а затем классифицирует составляющие его лексемы на основные (ядро) и дополнительные (детали) элементы.

На рис. 6 представлена схема общего процесса упрощения компонент предложения.

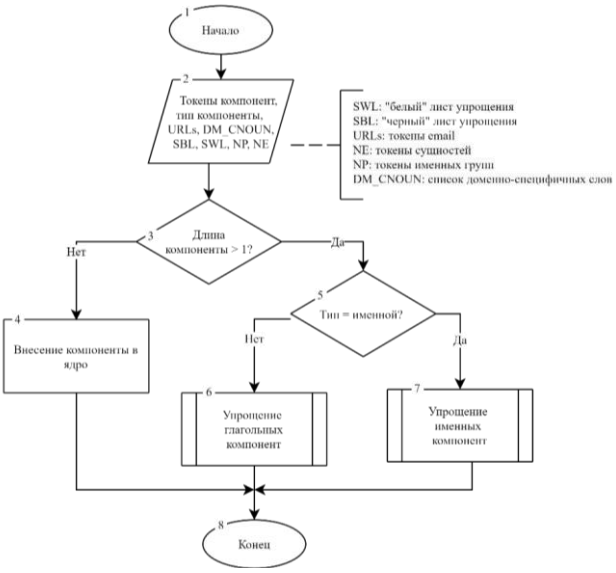

Рис. 6. Схема общего процесса упрощения компонент

Сначала определяется длина компоненты. Если длина больше 1, то упрощение не выполняется. Если компонента имеет тег части речи «глагол», производится переход к упрощению глагольного компонента; в противном случае переход к упрощению именного компонента.

На рис. 7 представлена схема упрощения именных компонент, а на рис. 8 – глагольных компонент.

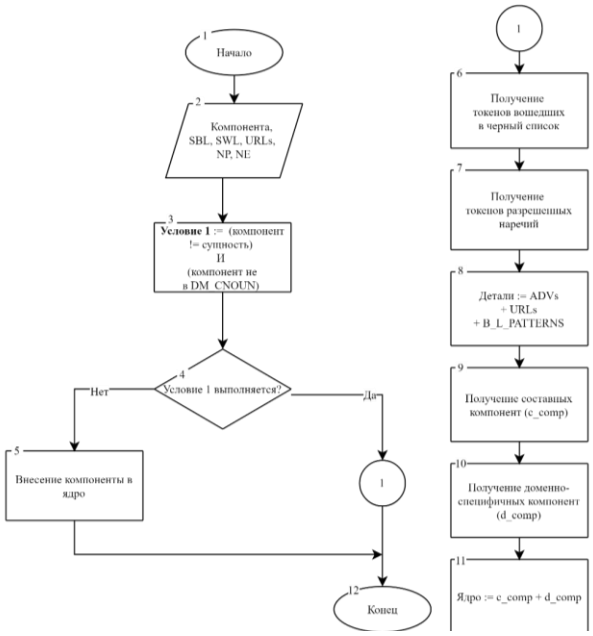

Рис. 7. Схема упрощения именных компонент

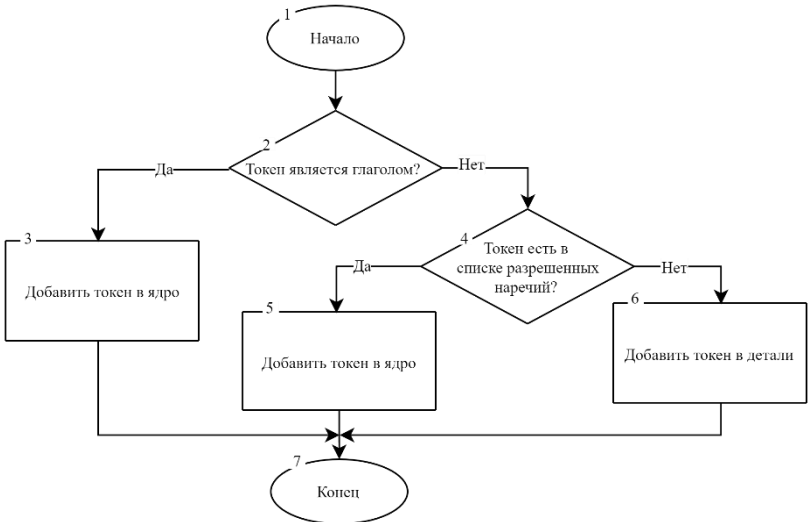

Рис. 8. Схема упрощения глагольных компонент

При упрощении именного компонента выполняется ряд проверок: если компонент не является сущностью, не внесен в «белый» список для упрощения, то выполняется упрощение именных компонент при помощи фильтрации неважных для передачи смысла предложения токенов на основе заранее определенных критериев.

Для каждого токена в глагольной компоненте выполняется следующая проверка: если токен является глаголом, то происходит добавление его в ядро. В противном случае – добавление токена в раздел «Детали».

3. Компонентная архитектура разработанного программного приложения и вычислительный эксперимент. На рис. 9 показана архитектура модулей разработанного программного приложения.

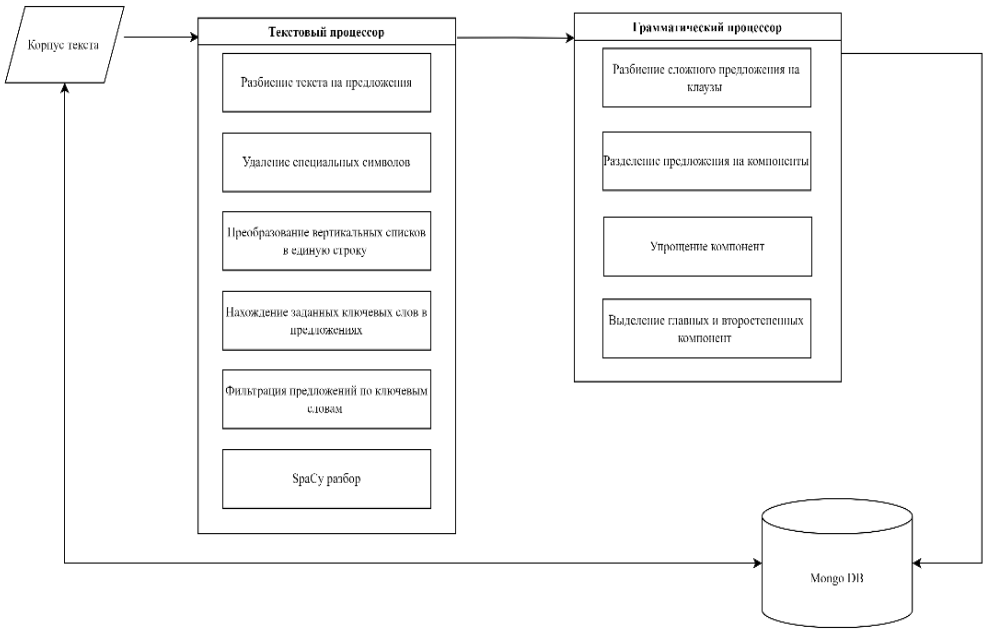

Рис. 9. Схема архитектуры модулей

Алгоритм направлен на упрощение предложений для увеличения скорости передачи и обработки информации: поступающий на вход корпус текста будет проходить через предварительный этап обработки, где поступивший текст будет разбит на отдельные предложения, из которых удаляются специальные символы. После этого будут отбираться те предложения, которые подходят под заданные ключевые слова.

Предложения последовательно будут отправляться на обработку spaCy. Далее предложение будет разбито на клаузы (клауза – группа слов, состоящая из подлежащего и глагола; элементарное предложение), а после этого каждая клауза будет разделена на компоненты. После этого будет запущен модуль упрощения. Здесь будут выделяться важные элементы внутри компонент, поскольку именно они будут представлять извлеченные знания. Все прилагательные и наречия, не попавшие в списки допуска, будут помечаться как детали и подразумеваться как слова, не имеющие критической важности для смысла.

После этого предложения с упрощенными элементами будут сохранены в БД и доступны для дальнейшей работы.

Для исследования работы алгоритма будем проверять качество упрощения предложений с использованием сразу нескольких метрик:

- индекса Флэша-Кинкейда, в котором значения варьируются от 0 до 18, где 18 соответствует наиболее сложному тексту;
- индекса Колман-Лиау, в котором значения варьируются от 0 до 11, где 11 соответствует наиболее сложному тексту;
- автоматического индекса удобочитаемости, в котором значения варьируются от 0 до 11, где 11 соответствует наиболее сложному тексту.

Проверим сложность чтения до работы алгоритма и после, а также сравним с результатами аналогов. Поскольку предложения являются сложными, то значения метрики будут превышены для всех рассматриваемых экспериментов. Цель – максимально приблизиться к верхним пороговым значениям.

Тестироваться будет следующее предложение:

«Social Island является рекламным агентством, которое использует силу психологического воздействия для своих любимых клиентов, чтобы обеспечить мгновенную вирусность в социальных сетях».

Авторское решение разделяет предложения на клаузы и работает с ними, поэтому результат будет отражен в виде последовательности полных клауз, полученных в результате работы. Результаты упрощения показаны в табл. 1.

Таблица 1

Результаты упрощения предложения

Сервис «Robotext»	Сервис «Neural Writer»	Авторское решение
Social Island – рекламное агентство, которое использует силу психологического воздействия для своих любимых клиентов, чтобы обеспечить мгновенную вирусность в социальных сетях.	Social Island использует психологическое воздействие, чтобы реклама стала вирусной в социальных сетях.	Social Island является рекламным агентством; агентство использует силу воздействия для клиентов; агентство обеспечить вирусность в социальных сетях.

Сервис Neural Writer полностью изменил предложение, не сохранив его структуру, а вот Robotext практически не изменил предложение.

Учитывая, что объем информации растет экспоненциально, не всегда системы позволяют получить ответ за допустимое время. Проверим насколько изменилась читабельность тестового предложения, а также сравним время отклика при помощи формул (1), (2), (3). Результаты сведены в табл. 2.

Таблица 2

Оценка сложности предложений и времени обработки

Метрика	Изначальное предложение	Сервис «Robotext»	Сервис «Neural Writer»	Авторское решение
Индекс Флэша-Кинкейда	21.77	21.68	21.27	17.57
Индекс Колман-Лиау	23.25	22.91	22.84	18.73
Автоматический индекс удобочитаемости	25.83	25.28	25.04	22.58
Скорость работы, сек.	–	1.8	2.4	1.9

Согласно индексу Флэша-Кинкейда, авторский алгоритм сумел упростить предложение до индекса 16, что соответствует тексту, который сложно читается в отличие от начального индекса, который показал, что предложение очень трудное для восприятия. Индекс Колмана-Лиац, а также автоматический индекс читаемости все еще превышают верхние пороговые значения показателей.

Проверим сложность 100 текстовых предложений, выведем средние показатели сложности в табл. 3.

Таблица 3

Средняя оценка сложности 100 предложений

Метрика	Изначальное предложение	Сервис «Robotext»	Сервис «Neural Writer»	Авторское решение
Средний индекс Флэша-Кинкейда	21.77	21.69	19.54	18.38
Средний индекс Колман-Лиау	23.25	20.64	19.42	16.31
Средний автоматический индекс удобочитаемости	25.83	23.8	21.72	18.32
Средняя скорость работы на одном предложении, сек	–	2.08	2.35	2.12

Табл. 3 демонстрирует, что упор на проработку домена позволяет добиться лучших показателей упрощения текста при допустимых временных рамках.

Временная сложность определяет, сколько времени требуется алгоритму для решения задачи при увеличении объема входных данных. Обычно она оценивается по числу операций, которые необходимо выполнить алгоритму.

Нотация Big-O позволяет показать эффективность (или быстродействие) алгоритма. Big-O позволяет легко сравнивать скорости работы алгоритмов и дает общее представление о том, сколько времени потребуется алгоритму для выполнения всех требуемых от него действий [28]. Термин «Big-O» обычно используется для описания общей производительности, но на самом деле описывает «наихудшую» (т.е. самую низкую) скорость, с которой может работать алгоритм.

Поскольку алгоритм использует различные операции (вложенные циклы, операция sort(), рекурсия), согласно [29], временная сложность алгоритма составляет в худшем случае: $O(n^2)$, т.е. представляет квадратичную сложность.

Заключение. В данной статье описана разработка алгоритма поиска и приобретения знаний. Основной целью работы являлось повышении эффективности алгоритмов поиска и приобретения знаний на основе технологий обработки и анализа текстов на естественном языке.

Основными результатами проведенного исследования являются следующие:

1. Представлена формализованная постановка решаемой задачи.

2. Разработан алгоритм поиска и приобретения знаний, которая отличается от известных аналогов использованием доменной проработки и применением детерминированных низкоуровневых правил, позволяющих обеспечить высокий уровень упрощения анализируемой текстовой информации.

Для оценки эффективности предложенного алгоритма разработано программное приложение и проведен вычислительный эксперимент. Полученные результаты проведенных экспериментальных исследований подтверждают эффективность предложенного алгоритма поиска и приобретения знаний на основе технологий обработки и анализа текстов на естественном языке. Временная сложность представленного алгоритма является квадратичной.

Данный алгоритм повышает эффективность обработки текстовой информации в различных доменах, требующих обработки больших объёмов данных, позволяя уменьшить сложность текста.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Appelt D.E.* The common pattern specification language // Technical report / SRI International, Artificial Intelligence Center. – 1998.
2. *Cunningham H., Maynard D., Bontcheva K., Tablan V.* A framework and graphical development environment for robust NLP tools and applications // Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. – 2002. – P. 168-175.
3. *Kluegl P., Toepfer M., Beck P.D. et al.* UIMA Ruta: Rapid development of rule-based information extraction applications // Natural Language Engineering. – 2016. – Issue 22, No. 1. – P. 1-40.
4. *Starostin A.S., Smurov I.M., Stepanova M.E.* A production system for information extraction based on complete syntactic semantic analysis // Papers from the Annual International Conference "Dialogue". – 2014. – P. 659-667.
5. *Куршев Е.П., Сулейманова Е.А., Трофимов И.В.* Роль знаний в системах извлечения информации из текстов // Программные системы: теория и приложения. – 2012. – Т. 3, № 3. – С. 57-70.
6. *Blanko M., Cafarella M.J., Soderland S. et al.* Open information extraction from the web // Proceedings of the 20th International Joint Conference on Artifical Intelligence. – 2007. – P. 2670-2676.
7. *Banko M., Etzioni O.* The tradeoffs between open and traditional relation extraction // Proceedings of ACL-08: HLT. – 2008. – P. 28-36.
8. *Zhu J., Nie Z., Liu X. et al.* StatSnowball: a statistical approach to extracting entity relationships // Proceedings of the 18th international conference on World wide web. – 2009. – P. 101-110.
9. *Wu F., Weld D.S.* Open information extraction using Wikipedia // Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. – 2010. – P. 118-127.
10. *Fader A., Soderland S., Etzioni O.* Identifying relations for open information extraction // Proceedings of the Conference on Empirical Methods in Natural Language Processing. – 2011. – P. 1535-1545.
11. *Etzioni O., Fader A., Christensen J. et al.* Open information extraction: The second // Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. – 2011. – P. 3-10.
12. *Schmitz M., Bart R., Soderland S. et al.* Open language learning for information extraction // Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. – 2012. – P. 523-534.
13. *Angeli G., Johnson Premkumar M.J., Manning C.D.* Leveraging linguistic structure for open domain information extraction // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. – 2015. – P. 344-354.
14. *Nakashole N., Weikum G., Suchanek F.* PATTY: A taxonomy of relational patterns with semantic types // Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. – 2012. – P. 1135-1145.
15. *Амурская О.Ю., Егорова А.Д.* Синтаксический парсинг как анализ структуры предложения (синтагмы) // Филология и культура. – 2022. – № 4 (70). – С. 14-23.

16. *Кобзарева Т.Ю.* Лингвистический базис анализа поверхностно-синтаксических связей сегментов в русском предложении // Вестник РГГУ. Серия: История. Филология. Культурология. Востоковедение. – 2008. – № 6. – С. 157-170.
17. *Кобзарева Т.Ю.* Иерархия задач поверхностно-синтаксического анализа русского предложения // Научно-техническая информация. Серия 2: Информационные процессы и системы. – 2007. – № 1. – С. 23-35.
18. *Кобзарева Т.Ю., Лахути Д.Г., Ножов И.М.* Сегментация русского предложения: поверхностно-синтаксический анализ как самостоятельный модуль анализа текста // Матер 5-ой международной конференции «Информационное общество, информационные ресурсы и технологии телекоммуникации». Секция «Интеллектуальные системы автоматизированной поддержки научных исследований». – М.: ВИНИТИ. НТИ, 2000. – С. 31-34.
19. *Кобзарева Т.Ю.* Проблема кореференции в рамках поверхностно-синтаксического анализа русского языка // Компьютерная лингвистика и интеллектуальные технологии: Тр. Международной конференции «Диалог 2003». – М.: Наука, 2003. – С. 278 -284.
20. *Кобзарева Т.Ю.* Принципы сегментационного анализа русского предложения // Московский лингвистический журнал. – 2004. – Т. 8, № 1. – С. 31-80.
21. *Кобзарева Т.Ю.* Поиск хозяина предложной группы в русском предложении // Компьютерная лингвистика и интеллектуальные технологии: Тр. Международной конференции «Диалог-2010». – 2010. – Вып. 9 (16). – С. 186-191.
22. *Rets I., Astruc L., Coughlan T., Stickler U.* Approaches to simplifying academic texts in English: English teachers' views and practices // English for Specific Purposes. – 2022. – Issue 68. – P. 31-46.
23. *Crossley S.A.*, *Allen D., McNamara D.S.* Text simplification and comprehensible input: A case for an intuitive approach // Language Teaching Research. – 2012. – Issue 16, No. 1. – P. 89-108. – DOI: 10.1177/1362168811423456.
24. *Soemer A., Schiefele U.* Text difficulty, topic interest, and mind wandering during reading // Learning and Instruction. – 2019. – Issue 61, No. 1. – P. 12-22.
25. *Allen D.* A study of the role of relative clauses in the simplification of news texts for learners of English // System. – 2009. – Issue 37, No. 4. – P. 585-599.
26. *Long M.H.* Optimal input for language learning: Genuine, simplified, elaborated, or modified elaborated? // Language Teaching. – 2020. – Issue 53, No. 2. – P. 169-182. – DOI: 10.1017/S0261444819000466.
27. *Tickoo M.L.* Simplification: Theory and Application. Anthology Series 31 // ERIC Clearinghouse. – 1993. – P. 254.
28. Big-O // Big-O.io. – URL: https://big-o.io (дата обращения: 03.07.2024).
29. Complexity Cheat Sheet for Python Operations // GeeksforGeeks. – URL: https://www.geeksforgeeks.org/complexity-cheat-sheet-for-python-operations/ (дата обращения: 03.07.2024).

REFERENCES

1. *Appelt D.E.* The common pattern specification language, *Technical report,* SRI International, Artificial Intelligence Center, 1998.
2. *Cunningham H., Maynard D., Bontcheva K., Tablan V.* A framework and graphical development environment for robust NLP tools and applications, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics,* 2002, pp. 168-175.
3. *Kluegl P., Toepfer M., Beck P.D. et al.* UIMA Ruta: Rapid development of rule-based information extraction applications, *Natural Language Engineering,* 2016, Issue 22, No. 1, pp. 1-40.
4. *Starostin A.S., Smurov I.M., Stepanova M.E.* A production system for information extraction based on complete syntactic semantic analysis, *Papers from the Annual International Conference "Dialogue",* 2014, pp. 659-667.
5. *Kurshev E.P., Suleymanova E.A., Trofimov I.V.* Rol' znaniy v sistemakh izvlecheniya informatsii iz tekstov [The role of knowledge in systems of information extraction from texts], *Programmnye sistemy: teoriya i prilozheniya* [Software systems: theory and applications], 2012, Vol. 3, No. 3, pp. 57-70.
6. *Blanko M., Cafarella M.J., Soderland S. et al.* Open information extraction from the web, *Proceedings of the 20th International Joint Conference on Artifical Intelligence,* 2007, pp. 2670-2676.
7. *Banko M., Etzioni O.* The tradeoffs between open and traditional relation extraction, *Proceedings of ACL-08: HLT,* 2008, pp. 28-36.
8. *Zhu J., Nie Z., Liu X. et al.* StatSnowball: a statistical approach to extracting entity relationships, *Proceedings of the 18th international conference on World wide web,* 2009, pp. 101-110.
9. *Wu F., Weld D.S.* Open information extraction using Wikipedia, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* 2010, pp. 118-127.

10. *Fader A., Soderland S., Etzioni O*. Identifying relations for open information extraction, *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 2011, pp. 1535-1545.
11. *Etzioni O., Fader A., Christensen J. et al.* Open information extraction: The second, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence,* 2011, pp. 3-10.
12. *Schmitz M., Bart R., Soderland S. et al.* Open language learning for information extraction, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* 2012, pp. 523-534.
13. *Angeli G., Johnson Premkumar M.J., Manning C.D.* Leveraging linguistic structure for open domain information extraction, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* 2015, pp. 344-354.
14. *Nakashole N., Weikum G., Suchanek F.* PATTY: A taxonomy of relational patterns with semantic types, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* 2012, pp. 1135-1145.
15. *Amurskaya O.Yu., Egorova A.D.* Sintaksicheskiy parsing kak analiz struktury predlozheniya (sintagmy) [Syntactic parsing as an analysis of sentence structure (syntagma)], *Filologiya i kul'tura* [Philology and Culture], 2022, No. 4 (70), pp. 14-23.
16. *Kobzareva T.Yu.* Lingvisticheskiy bazis analiza poverkhnostno-sintaksicheskikh svyazey segmentov v russkom predlozhenii [Linguistic basis for the analysis of superficial-syntactic connections of segments in a Russian sentence], *Vestnik RGGU. Seriya: Istoriya. Filologiya. Kul'turologiya. Vostokovedenie* [Bulletin of the Russian State University for the Humanities. Series: History. Philology. Cultural Studies. Oriental Studies], 2008, No. 6, pp. 157-170.
17. *Kobzareva T.Yu.* Ierarkhiya zadach poverkhnostno-sintaksicheskogo analiza russkogo predlozheniya [Hierarchy of tasks of superficial syntactic analysis of Russian sentences], *Nauchno-tekhnicheskaya informatsiya. Seriya 2: Informatsionnye protsessy i sistemy* [Scientific and technical information. Series 2: Information processes and systems], 2007, No. 1, pp. 23-35.
18. *Kobzareva T.Yu., Lakhuti D.G., Nozhov I.M.* Segmentatsiya russkogo predlozheniya: poverkhnostno-sintaksicheskiy analiz kak samostoyatel'nyy modul' analiza teksta [Segmentation of Russian sentence: surface-syntactic analysis as an independent module of text analysis], *Mater. 5-oy mezhdunarodnoy konferentsii «Informatsionnoe obshchestvo, informatsionnye resursy i tekhnologii telekommunikatsii». Sektsiya «Intellektual'nye sistemy avtomatizirovannoy podderzhki nauchnykh issledovaniy»* [Proceedings of the 5th international conference "Information society, information resources and telecommunication technologies". Section "Intelligent systems of automated support of scientific research"]. Moscow: VINITI. NTI, 2000, pp. 31-34.
19. *Kobzareva T.Yu.* Problema koreferentsii v ramkakh poverkhnostno-sintaksicheskogo analiza russkogo yazyka [he problem of coreference in the framework of superficial syntactic analysis of the Russian language], *Komp'yuternaya lingvistika i intellektual'nye tekhnologii: Tr. Mezhdunarodnoy konferentsii «Dialog 2003»* [Computer linguistics and intellectual technologies: Proceedings of the International Conference "Dialogue 2003"]. Moscow: Nauka, 2003, pp. 278 -284.
20. *Kobzareva T.Yu.* Printsipy segmentatsionnogo analiza russkogo predlozheniya [Principles of segmentation analysis of Russian sentences], *Moskovskiy lingvisticheskiy zhurnal* [Moscow Linguistic Journal], 2004, Vol. 8, No. 1, pp. 31-80.
21. *Kobzareva T.Yu.* Poisk khozyaina predlozhnoy gruppy v russkom predlozhenii [Search for the owner of a prepositional group in a Russian sentence], *Komp'yuternaya lingvistika i intellektual'nye tekhnologii: Tr. Mezhdunarodnoy konferentsii «Dialog-2010»* [Computer linguistics and intellectual technologies: Proceedings of the International Conference "Dialogue-2010"], 2010, Issue 9 (16), pp. 186-191.
22. *Rets I., Astruc L., Coughlan T., Stickler U.* Approaches to simplifying academic texts in English: English teachers' views and practices, *English for Specific Purposes,* 2022, Issue 68, pp. 31-46.
23. *Crossley S.A.*, *Allen D., McNamara D.S.* Text simplification and comprehensible input: A case for an intuitive approach, *Language Teaching Research,* 2012, Issue 16, No. 1, pp. 89-108. DOI: 10.1177/1362168811423456.
24. *Soemer A., Schiefele U.* Text difficulty, topic interest, and mind wandering during reading, *Learning and Instruction,* 2019, Issue 61, No. 1, pp. 12-22.
25. *Allen D.* A study of the role of relative clauses in the simplification of news texts for learners of English, *System,* 2009, Issue 37, No. 4, pp. 585-599.
26. *Long M.H.* Optimal input for language learning: Genuine, simplified, elaborated, or modified elaborated?, *Language Teaching,* 2020, Issue 53, No. 2, pp. 169-182. DOI: 10.1017/S0261444819000466.

27. *Tickoo M.L*. Simplification: Theory and Application. Anthology Series 31, *ERIC Clearinghouse,* 1993, pp. 254.
28. Big-O, *Big-O.io*. Available at: https://big-o.io (accessed 03 July 2024).
29. Complexity Cheat Sheet for Python Operations, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/complexity-cheat-sheet-for-python-operations/ (accessed 03 July 2024).

Статью рекомендовал к опубликованию д.т.н. В.Ф. Лубенцов.

Герасименко Евгения Михайловна – Южный федеральный университет; e-mail: egerasimenko@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; кафедра систем автоматизированного проектирования им В.М. Курейчика; доцент.

Кравченко Юрий Алексеевич – e-mail: yakravchenko@sfedu.ru; тел.: 88634371651; кафедра систем автоматизированного проектирования им В.М. Курейчика; д.т.н.; профессор.

Шаненко Дарья Андреевна – e-mail: shanenko@sfedu.ru; тел.: 88634371651; кафедра систем автоматизированного проектирования им В.М. Курейчика, ассистент.

Gerasimenko Evgeniya Mihailovna – Southern Federal University; e-mail: egerasimenko@sfedu.ru; Taganrog, Russia; phone: +78634371651; the Department of Computer Aided Design named after V.M. Kureychik; associate professor.

Kravchenko Yury Alekseevich – e-mail: yakravchenko@sfedu.ru; phone: +78634371651; the Department of Computer Aided Design named after V.M. Kureychik; dr. of eng. sc.; professor.

Shanenko Daria Andreevna – e-mail: shanenko@sfedu.ru; phone: +78634371651; the Department of Computer Aided Design named after V.M. Kureychik; assistant.

УДК 004.056.55 DOI 10.18522/2311-3103-2024-5-102-118

С.В. Поликарпов, В.А. Прудников, К.Е. Румянцев

СИНТЕЗ ПСЕВДО-ДИНАМИЧЕСКИХ ФУНКЦИЙ PD-sbox-ARX-32

Целью работы является разработка метода синтеза оптимальных псевдо-динамических функций PD-sbox-ARX-32, размерностью 32 бита, в соответствие с противоречивыми требованиями к криптографическим характеристикам, рассматриваемой структуры. Рассмотрены методы синтеза классических sbox, в том числе с использованием эволюционного и генетического методов. Представлены требования к криптографическим характеристикам, как к функциям PD-sbox, так и к их составным элементам (классические sbox и ARX-функции). Предложен метод синтеза псевдо-динамических функций PD-sbox-ARX-32, включающий два этапа: 1) эвристический поиск структуры, соответствующей противоречивым требованиям к результирующим криптографическим характеристикам, потребляемым программным и аппаратным ресурсам, а также скорости работы представленной функции; 2) поиск оптимальных параметров основного элемента PD-sbox-ARX-32 – ARX-функций, при помощи эволюционного метода, суть которого заключается в подборе значений циклических сдвигов в ARX-функциях. В результате получен набор четырёх ARX-функций для псевдо-динамического преобразования PD-sbox-ARX-32, имеющего вес линейных характеристик равный 2^{-13} и разностных характеристик равный 2^{-32} (при этом эмпирический вес составляет 2^{-26}). Для определения весов криптографических характеристик в работе применены методы на основе использования SAT-решателей. Приведены выводы о том, что подобранная структура 32-битной ARX-функции в составе PD-sbox позволяет обеспечить критический путь (максимальное количество последовательных операций сложения по модулю 2^{16}) в четыре раза меньше чем 8-итерационная 32-битная Alzette-подобная структура, при двухкратном увеличении количества операций и при сопоставимых максимальных значениях весов разностных и линейных характеристик. Аналогичный результат получается при сравнении 32-битной ARX-функции с 8-итерационным 32-битным преобразованием из блочного криптоалгоритма Speck32. Предложенный метод синтеза параметров 32-битной ARX-функции позволяет минимизировать количество затрачиваемых ассемблерных инструкций на операции циклического сдвига при реализации на малоресурсных 8-битных микроконтроллерах AVR (например, ATmega328P).

Криптографические свойства; псевдо-динамическая функция; PD-sbox-ARX-32; синтез псевдо-динамических функций.