DOI 10.37882/2223-2966.2023.06.39

ПРОГРАММНОЕ СРЕДСТВО ЧАТ-БОТА ДЛЯ ПОИСКА НЕСТРУКТУРИРОВАННЫХ ТЕКСТОВЫХ ДАННЫХ НА ОСНОВЕ ТЕХНОЛОГИИ DOCKER

CHATBOT SOFTWARE TOOL FOR SEARCHING UNSTRUCTURED TEXT DATA BASED ON DOCKER TECHNOLOGY

V. Filatov
S. Koryagin
A. Kryazhin
A. Andreev
A. Rusakov

Summary: This article presents the architecture of a text analytics system for creating text bots, searching for unstructured data based on modern python libraries. A distinctive feature of the developed architecture is the use of docker with accompanying tools in the form of containers, which allow you to run this system on any platform, scaling to any task. The text visualization based on d3js library is also used, which allows to understand the meaning of this text and to describe the main topics considered in the text.

Keywords: morphological analysis, keyword extraction, unstructured data mining, text mining, Docker.

Филатов Вячеслав Валерьевич
кандидат технических наук, доцент,
Российский Технологический Университет МИРЭА
filv@mail.ru
Корягин Сергей Викторович
кандидат технических наук, доцент,
Российский Технологический Университет МИРЭА
dongenealog2003@mail.ru
Кряжин Александр Александрович
Российский Технологический Университет МИРЭА
8743808@gmail.com
Андреев Александр Григорьевич
Российский Технологический Университет МИРЭА
festagain123@gmail.com
Русаков Алексей Михайлович
старший преподаватель,
Российский Технологический Университет МИРЭА
rusal@bk.ru

Аннотация. В данной статье приводится архитектура системы текстовой аналитики для создания текстовых ботов, поиска неструктурированных данных на основе современных библиотек python. Отличительной особенностью, разрабатываемой архитектуры, является использование docker с сопутствующими инструментами в виде контейнеров, которые позволяют запускать данную систему на любой платформе, масштабируя под любую задачу. Также используется визуализация текста на основе библиотеки d3js, которая позволяет понять смысл данного текста и описать основные темы, рассматриваемые в тексте.

Ключевые слова: морфологический анализ, извлечение ключевых слов, поиск неструктурированных данных, интеллектуальный анализ текста, Docker.

Введение

С развитием интернета и ростом популярности автоматической обработки естественного языка (NLP) и компьютерной лингвистики, чат-боты стали активно применяться в разных сферах, таких как развлечения и бизнес. Повышение эффективности чат-ботов позволило им осуществлять рутинные задачи и круглосуточно отвечать на вопросы клиентов.

Одним из ключевых аспектов анализа и обработки естественного языка является извлечение ключевых слов из текста, что способствует быстрому пониманию его смысла. В связи с актуальностью проблемы информационного шума, программное средство, реализованное в данном проекте, направлено на ускорение процесса анализа больших объемов текста и извлечение наиболее важной информации для пользователя.

Далее будет рассматриваться разработка программного средства чат-бота для поиска неструктурированных текстовых данных и извлечения ключевых слов. Основные задачи, решаемые в ходе работы, включают обзор математических методов и алгоритмов для анализа текстов, разработку архитектуры системы текстовой аналитики и технологической структуры чат-бота [1].

Архитектура системы текстовой аналитики

Для морфологического анализа применяется библиотека pymorphy2, основанная на словаре OpenCorpora и структуре данных DAWG. Сначала токены получают грамматическую информацию, после чего обрабатываются для создания специальных классов.

Предсказатель pymorphy2 имеет два алгоритма предсказания, работающих совместно. Один из подходов заключается в отсечении префиксов, особенно если

```
text = """Турагенты не владеют средствами обслуживания и
выступают посредниками между предприятием туристского
обслуживания и покупателем туристской путевки, продвигая
и реализуя туристский продукт. С законодательством,
закрепляющим деятельность туристских агентств и туроператоров
в Российской Федерации, можно ознакомиться на сайте
федерального агентства по туризму. В России туристический
бизнес стал активно развиваться в 1990-х. В 1993 была
учреждена Российская ассоциация туристических агентств (PATA),
которая в 2002 была преобразована в Российский союз
туриндустрии (РСТ). Одна из проблем в сфере правового
регулирования взаимоотношений между страховыми компаниями и
туроператорами заключается в отсутствии законодательного
запрета на смену статуса компаний, осуществляющих реализацию
туристского продукта конечным потребителям (туристам). Это
означает, что компания, реализующая турпродукт, может
«перейти» из статуса туроператора в статус турагентства, то
есть вместо туроператорской деятельности (по формированию,
продвижению и реализации туристского продукта) начать
осуществлять турагентскую деятельность (по продвижению и
реализации туристского продукта). В силу действующего
законодательства, турагент не несет ответственности по страховым
полисам, проданным туроператором. Это означает, что на практике
не исключена ситуация, когда потребитель (турист),
приобретший турпродукт у подобного турагента («бывшего
туроператора»), при наступлении страхового случая столкнется с
трудностями при получении страхового возмещения."""
```

```
[
  "Турагенты",
  "не",
  "владеют",
  "средствами",
  "обслуживания",
  "и",
  "выступают",
  "посредниками",
  "между",
  "предприятием",
  "туристского",
  "обслуживания",
  "и",
  "покупателем",
  "туристской",
  "путевки",
  "продвигая",
  "и",
  "реализуя",
  "туристский",
  "продукт",
  "С",
  "законодательством",
  "закрепляющим",
  "деятельность",
  "туристских",
  "агентств",
  "и",
  "туроператоров",
  "в",
  "Российской",
  "Федерации",
```

Рис. 1. Токенизация текста

```
chunk: 33
Parse(word='российский', tag=OpencorporaTag('ADJF masc,sing,nomn'), normal_form='российский', score=0.85, methods_
Parse(word='союз', tag=OpencorporaTag('NOUN,inan,masc sing,nomn'), normal_form='союз', score=0.767857, methods_sta
Parse(word='туриндустрии', tag=OpencorporaTag('NOUN,inan,femn sing,gent'), normal_form='туриндустрия', score=0.2,

subchunk: 33.1
Parse(word='союз', tag=OpencorporaTag('NOUN,inan,masc sing,nomn'), normal_form='союз', score=0.767857, methods_sta
Parse(word='туриндустрии', tag=OpencorporaTag('NOUN,inan,femn sing,gent'), normal_form='туриндустрия', score=0.2,

subchunk: 33.2
Parse(word='туриндустрии', tag=OpencorporaTag('NOUN,inan,femn sing,gent'), normal_form='туриндустрия', score=0.2,
```

Рис. 2. Пример извлечения вложенных слов с помощью параметра nested

они являются известными словообразовательными префиксами [2].

Если слово не начинается с известного префикса, анализатор все равно пробует разобрать слово путем отсечения префикса. Алгоритм корректно работает только с не очень длинными префиксами и не очень короткими остатками.

Алгоритм извлекает данные последовательности в списки логических блоков, которые впоследствии, после их обработки, станут ключевыми словами. В работе данного алгоритма также заложена возможность извлечения вложенных термов с помощью специального аргумента nested.

Задача выделения ключевых слов преобразуется в задачу разметки последовательности с использованием статистической модели и ручной BIO-разметки. В множество допустимых признаков X входят группы признаков, такие как варианты разбора токена по словарю, признаки на основе написания токена, положения токена в предложении и наличие слов-триггеров.

Ключевые слова имеют вид: <adj>* <noun>+, где <adj>* означает встреченные ноль или несколько прилагательных (а также числительных или причастий), а <noun>+ означает одно или несколько существительных.

После извлечения ключевых слов производится расчет веса и сортировка по частоте употребления и количеству слов. При визуализации графов веса корректируются с использованием меры TF-IDF для одиночных ключевых слов. TF-IDF — это статистическая мера, ис-

пользуемая для оценки важности слова в тексте, учитывающая частоту термина и обратную частоту документа.

Для русского языка на текущий момент не создано корпуса с разметкой временных выражений, поэтому для тестирования статистической модели была выполнена ручная BIO-разметка. BIO-последовательность содержит метку для каждого токена текста: B (начало ключевого слова), I (внутри ключевого слова) и O (вне ключевого слова).

Следующим этапом является применение алгоритма BIO-разметки для извлечения ключевых слов. Задача выделения временного выражения сводится к задаче разметки последовательности, где y — скрытая последовательность переменных, x — последовательность наблюдаемых переменных.

Результатом алгоритма является извлечение ключевых слов из текста, которые могут быть использованы для анализа, визуализации и дополнительной обработки. Важность каждого ключевого слова оценивается на основе его частоты и веса, корректируемого с помощью меры TF-IDF.

Подбор современных инструментов для разработки чат-бота

Архитектура чат-бота основана на микросервисном подходе и включает 4 слабо связанных Docker-контейнера (рисунок 3):

Технология создания Telegram чат-ботов, разработанный на Python, прослушивает и обрабатывает запросы пользователей. После обработки текста и извлечения ключевых слов, результаты отправляются в NoSQL базу данных MongoDB, представленную другим микросервисом. Каждому запросу присваивается случайный идентификатор, по которому можно получить результат работы алгоритма через веб-браузер: чат-бот формирует ссылку на веб-страницу и отправляет ее пользователю в мессенджере.

База данных NoSQL MongoDB хранит ключевые слова, извлеченные из текста пользователя, в формате JSON-документа с уникальным идентификатором.

Фреймворк Eve предоставляет HTTP REST API для доступа к результатам обработки текста пользователя в базе данных MongoDB с использованием JavaScript XMLHttpRequest запросов.

Веб-сервер Nginx обеспечивает доступ к результатам обработки текста через веб-браузер. Запросы к REST API, написанному на Python-фреймворке Eve, обрабатываются базой данных MongoDB, которая извлекает данные и возвращает их JavaScript обработчику в браузере. JavaScript-алгоритмы обрабатывают полученные ключевые слова и отображают их пользователю в виде графиков с использованием JavaScript-библиотеки D3.js.

Схематичное изображение последовательности работы чат-бота представлено на рисунке ниже. Далее приведены следующие библиотеки Python, которые использовались для решения поставленной задачи:

Библиотека python-telegram-bot — это инструментарий для разработки ботов в мессенджере Telegram на языке Python. Она предоставляет простой и удобный интерфейс для создания и настройки ботов, а также позволяет легко взаимодействовать с Telegram API. С помощью python-telegram-bot вы можете создавать ботов для общения с пользователями, управления группами, отправки уведомлений и многого другого.

pymorphy2 — это библиотека морфологического анализа русских слов на языке Python [2]. Она позволяет лемматизировать и склонять слова, определять их части речи, падежи и другие характеристики. pymorphy2 используется в различных проектах, связанных с обработкой естественного языка, в том числе в поисковых системах и анализаторах текста.

pymongo — это библиотека для работы с базами данных MongoDB на языке Python [6]. Она предоставляет удобный интерфейс для взаимодействия с базой данных, позволяет выполнять запросы на чтение и запись данных, а также работать с индексами и агрегатами. pymongo поддерживает все основные функции MongoDB и может быть использована для разработки приложений, связанных с хранением и обработкой данных.

rutermextract — это библиотека для извлечения ключевых слов из текстов на русском языке [5]. Она используется для автоматической обработки больших объемов текста и позволяет выделять наиболее значимые слова и фразы. rutermextract основана на статистических методах анализа текста и может быть использована для анализа статей, отзывов, новостей и других типов текстов.

Eve — это фреймворк для создания RESTful API на языке Python. Он позволяет быстро и легко создавать API для веб-приложений, используя базу данных MongoDB в качестве хранилища данных. Eve предоставляет гибкую систему маршрутизации, аутентификации и авторизации, а также позволяет легко настраивать параметры API, такие как форматы ответов, фильтрация и сортировка данных. Eve может быть использован для создания различных приложений, связанных с обработкой данных и взаимодействием с внешними сервисами.

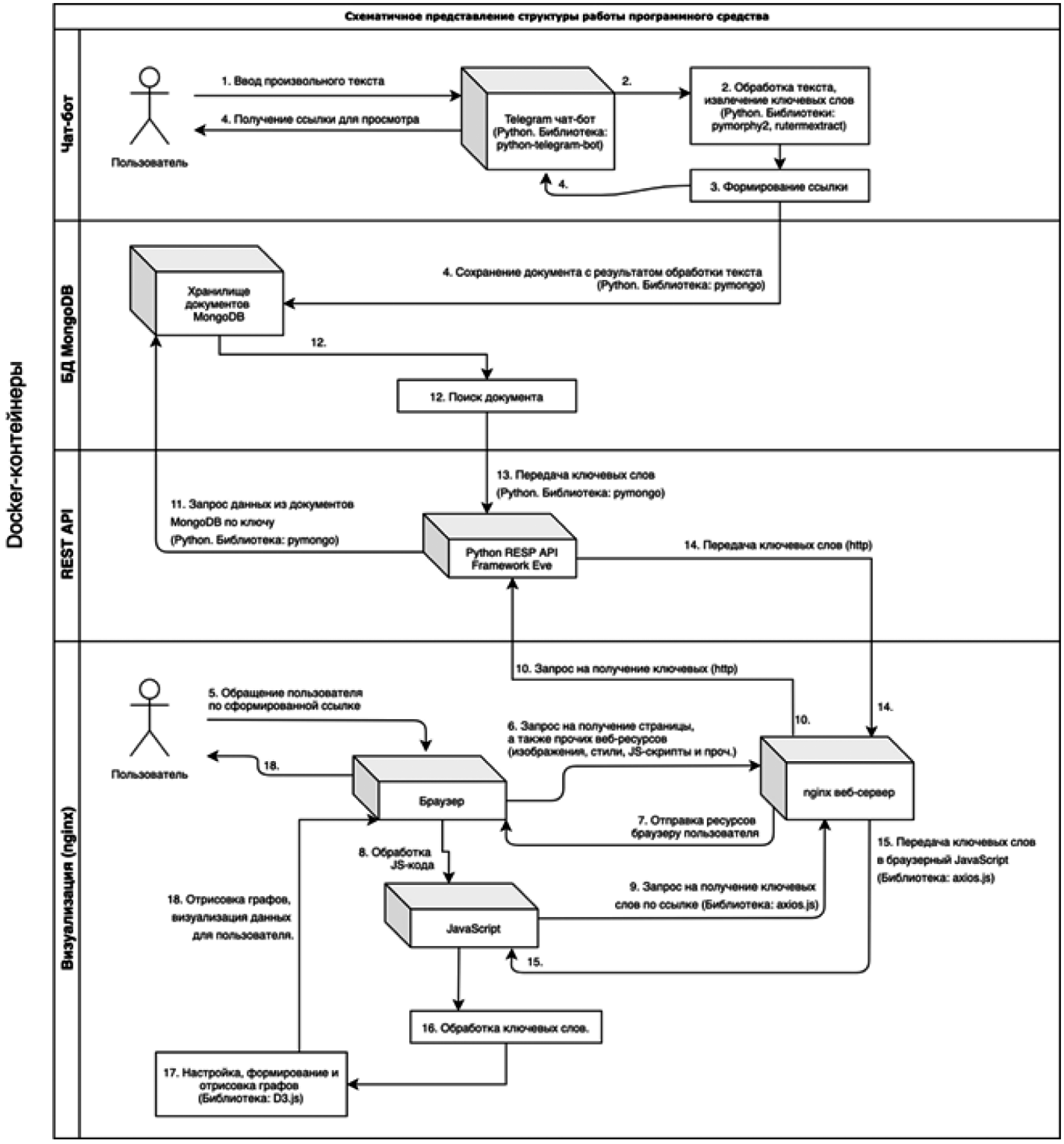

Рис. 3. Схематичное представление структуры работы программного средства через Docker-контейнеры

Тестирование и отладка программного средства

Для тестирования работы программного средства проводилась серия экспериментов, в результате чего было сделано заключение, что разработанное программное средство чат-бота может применяться для решения поставленной задачи.

На приведенных ниже скриншотах был произведен тест программного средства с использованием произвольного текста из произведения Ф.М. Достоевского «Братья Карамазовы».

Был начат диалог с ботом. Боту был отправлен произвольный текст (рисунок 4);

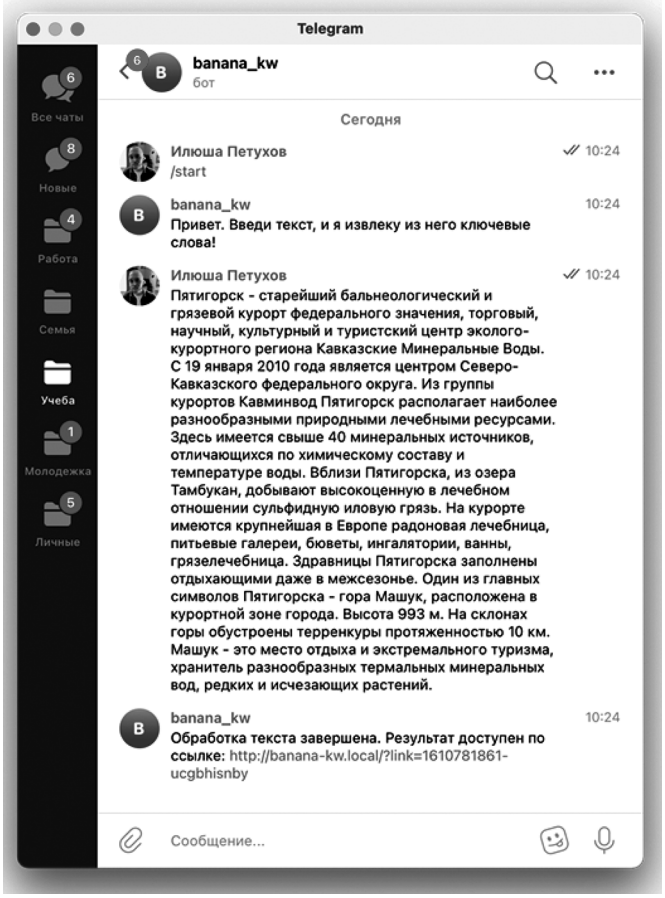

Рис. 4. Запрос на обработку текста

1. Пользователь перешел по ссылке для просмотра результата обработки текста в виде графа ключевых слов (рисунок 5);

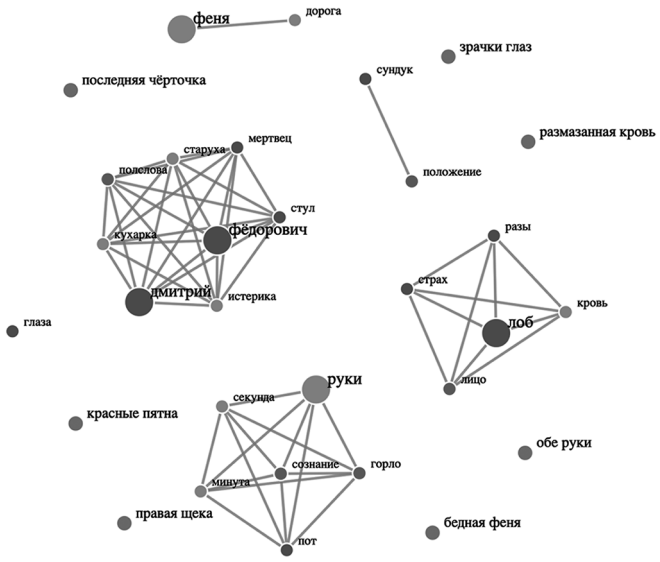

Рис. 5. Получение визуализированного результата в виде графов со списком ключевых слов

2. Был произведен более подробный разбор ключевых слов с помощью дополнительной пользовательской настройки в интерфейсе страницы (рисунок 6).

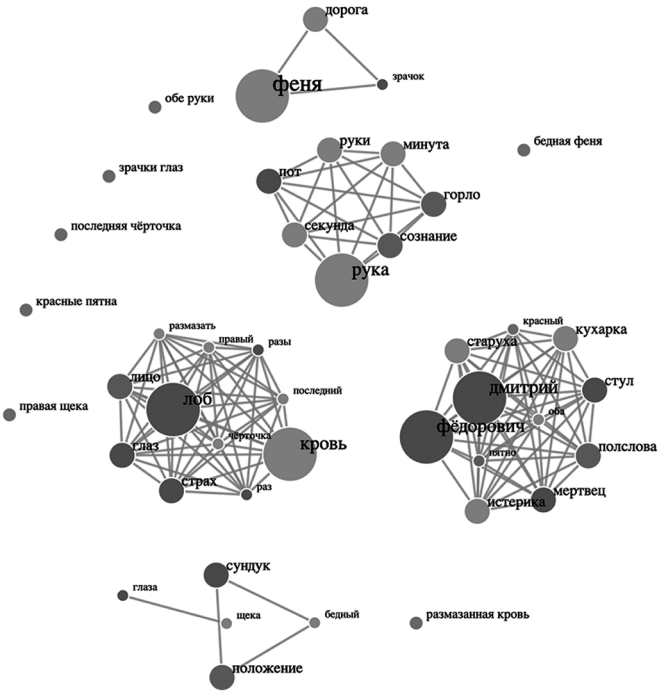

Рис. 6. Более подробный разбор ключевых слов

Выводы

В данной работе была разработана система текстовой аналитики на основе микросервисной архитектуры, представленная в виде чат-бота для мессенджера Telegram. Для решения поставленной задачи были использованы следующие библиотеки и инструменты: python-telegram-bot, pymorphy2, rutermextract, pymongo, Eve. Система осуществляет морфологический анализ текста, извлечение ключевых слов, а также оценку их важности на основе меры TF-IDF. Разработанное программное средство было протестировано на примере произвольного текста из произведения Ф. М. Достоевского «Братья Карамазовы». Результаты работы системы были успешно визуализированы в виде графа ключевых слов и могут быть использованы для анализа, визуализации и дополнительной обработки текстов на русском языке.

ЛИТЕРАТУРА

1. Зильберман Н.Н. Технологии виртуальных собеседников и формы речевого взаимодействия //Гуманитарная информатика. — 2009. — №. 5. — С. 80–85.
2. Иомдин Л.Л. и др. Синтаксический анализатор системы ЭТАП: современноесостояние //Annual International Conference» Dialogue. — 2012.
3. Казенников А.О. Сравнительный анализ статистических алгоритмов синтаксического анализа на основе деревьев зависимостей //Компьютерная лингвистика и интеллектуальные технологии. По материалам ежегодной Международной конференции «Диалог. — 2010. — №. 9. — С. 16.
4. Rad B.B., Bhatti H.J., Ahmadi M. An introduction to docker and analysis of its performance //International Journal of Computer Science and Network Security (IJCSNS). — 2017. — Т. 17. — №. 3. — С. 228.
5. Korobov M. Morphological analyzer and generator for Russian and Ukrainian languages //Analysis of Images, Social Networks and Texts: 4th International Conference, AIST 2015, Yekaterinburg, Russia, April 9–11, 2015, Revised Selected Papers 4. — Springer International Publishing, 2015. — С. 320–332.
6. Banker K. et al. MongoDB in action: covers MongoDB version 3.0. — Simon and Schuster, 2016.
7. Мальковский М.Г., Арефьев Н.В. Сочетаемостные ограничения в системе автоматического синтаксического анализа //Программные продукты и системы. — 2012. — №. 1. — С. 28–31.
8. Горелов А.И. Обзор развития технологий виртуальных собеседников //Научно-практические исследования. — 2018. — №. 6. — С. 74–80.

© Филатов Вячеслав Валерьевич (filv@mail.ru), Корягин Сергей Викторович (dongenealog2003@mail.ru);
Кряжин Александр Александрович (8743808@gmail.com); Андреев Александр Григорьевич (festagain123@gmail.com);
Русаков Алексей Михайлович (rusal@bk.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»