

СИСТЕМА ИЗВЛЕЧЕНИЯ И АНАЛИЗА ТЕКСТОВЫХ ДАННЫХ ИЗ СОЦИАЛЬНЫХ СЕТЕЙ ДЛЯ ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

К. А. Лехов¹, Д. Д. Сперанский²,
М. А. Митрохин³, Н. С. Карамышева⁴

^{1, 2, 3, 4} Пензенский государственный университет, г. Пенза, Россия

¹ kirill.lekhov@pnzgu.ru, ² dmitry.speransky@pnzgu.ru,
³ mmax83@mail.ru, ⁴ karamyshevans@yandex.ru

Аннотация. *Актуальность и цели.* Рассмотрены вопросы использования инструментов, предоставляемых социальными сетями и мессенджерами, для построения системы анализа больших данных для поддержки деятельности образовательной организации. *Материалы и методы.* В качестве материалов работы выступает текстовая информация социальной сети «ВКонтакте», а именно опубликованные в открытом доступе пользователями текстовые материалы. Рассматривается вопрос использования API социальной сети «ВКонтакте» в качестве инструмента извлечения данных. Для анализа текстовой информации предлагается применение методов обработки естественного языка (natural language processing). *Результаты.* Предложена архитектура системы анализа текстовых данных и построения социального графа на основе информации социальной сети «ВКонтакте», размещенной в общем доступе. *Выводы.* При реализации предложенной архитектуры система позволит осуществлять тематический мониторинг текстовой информации, определять тренды интересов и запросы обучающихся.

Ключевые слова: аналитика, обработка данных, API, автоматизация, социальные сети

Для цитирования: Лехов К. А., Сперанский Д. Д., Митрохин М. А., Карамышева Н. С. Система извлечения и анализа текстовых данных из социальных сетей для образовательного учреждения // Модели, системы, сети в экономике, технике, природе и обществе. 2021. № 1. С. 128–136. doi:10.21685/2227-8486-2021-1-11

SYSTEM FOR EXTRACTING AND ANALYZING TEXT DATA FROM SOCIAL NETWORKS FOR AN EDUCATIONAL INSTITUTION

K.A. Lekhov¹, D.D. Speranskiy²,
M.A. Mitrokhin³, N.S. Karamysheva⁴

^{1, 2, 3, 4} Penza State University, Penza, Russia

¹ kirill.lekhov@pnzgu.ru, ² dmitry.speransky@pnzgu.ru,
³ mmax83@mail.ru, ⁴ karamyshevans@yandex.ru

Abstract. *Background.* The paper discusses the use of tools provided by social networks and messengers to build a big data analysis system to support the activities of an educational organization. *Materials and methods.* The materials of the work are text information of the social network VKontakte, namely text materials published in the public domain by users. The question of using the API of the social network "VKontakte" as a data extraction tool is considered. For the analysis of text information, the use of natural language processing methods is proposed. *Results.* The article proposes the architecture of the

system for analyzing text data and building a social graph based on the information of the social network "VKontakte" posted in the public domain. *Conclusions.* Implementation of the proposed architecture will allow for the system thematic monitoring of text information and identification of trends of interests and requests of students.

Keywords: analytics, data processing, API, automation, social networks

For citation: Lekhov K.A., Speranskiy D.D., Mitrokhin M.A., Karamysheva N.S. System for extracting and analyzing text data from social networks for an educational institution. *Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve = Models, systems, networks in economics, technology, nature and society.* 2021;1:128–136. (In Russ.). doi:10.21685/2227-8486-2021-1-11

Введение

Социальные сети стали неотъемлемой частью жизни современного человека. В них люди работают, общаются, обучаются и т.д. Также социальные сети являются сосредоточением персональной информации об интересах людей, их компетенциях, активности. Компании, такие как Facebook, Google, Yandex, Mail.ru, в своих сервисах массово используют эту информацию в коммерческих целях для таргетирования рекламы [1], аналитики и маркетинговых мероприятий. Для обработки большого объема данных в автоматическом режиме компании разрабатывают собственные, не доступные для массового использования, инструменты. Помимо коммерческого применения (рекомендательные сервисы и таргетированная реклама) такие программные продукты позволяют в автоматическом режиме распознавать слова и фразы, категоризовать контент по тематикам и на основе полученных данных блокировать или удалять материалы (при наличии ненормативной лексики, оскорблений и т.д.).

Образовательным учреждениям информация социальных сетей также может помочь в их основной деятельности: для целевой работы с абитуриентами, определения интересов учащихся и вовлечения их во внеучебную деятельность в соответствии с этими интересами, выявления негативных проявлений в общении и коммуникациях, отслеживания трудовых успехов выпускников. На основе полученной информации также могут составляться аналитические отчеты по контингенту.

Материал и методика

Для анализа данных, полученных из социальных сетей, в настоящее время доступен ряд технологий. Из-за значительного объема данные хранятся в распределенных базах данных, это позволяет обеспечить надежное хранение и масштабируемость. В обработке таких данных также доминирует распределенный подход, реализуемый в рамках парадигмы *MapReduce* [2]. Одним из популярных наборов инструментов для распределенного хранения и обработки данных является фреймворк *Apache Hadoop* [3], который включает набор программных компонент для работы с данными. В качестве методологической базы при работе с текстовой информацией, полученной из социальных сетей, используется *Natural Language Processing* – набор методов, который позволяет определять смысловую нагрузку, выявлять тематическую и интонационную окраску текста [4].

В качестве материалов исследования выступают данные социальной сети «VKontakte», а именно опубликованные в открытом доступе пользователями текстовые материалы.

Описание предлагаемой архитектуры системы

В целях анализа информации для образовательных учреждений может быть создан программный продукт для автоматизации работы с контентом социальных сетей. Предполагаемый функционал продукта рассмотрим на примере архитектуры программы анализа данных пользователей социальной сети «ВКонтакте».

Программа включает графический пользовательский интерфейс, базу данных, тематический словарь и модуль сканирования, который объединяет функционал сбора и анализа данных и модуль построения социального графа (рис. 1).

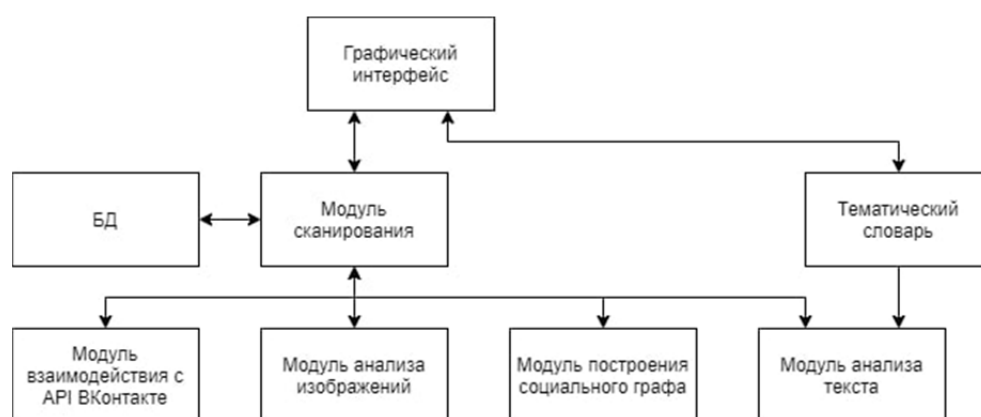


Рис. 1. Схема взаимодействия модулей

Для эффективного взаимодействия с социальной сетью необходимо применять инструменты, предоставленные компанией-разработчиком. Для сбора данных возможно использование библиотеки VK_API [5], которая позволяет применять почти все возможности API социальной сети «ВКонтакте» [6]. В таблице 1 представлены наиболее востребованные методы API «ВКонтакте» для обсуждаемой системы.

Таблица 1

Методы API «ВКонтакте»

Метод	Возвращаемые данные
vk.users.get	ID профиля «ВКонтакте» по «никнейму», основные данные на странице пользователя (город, школа, образование, дата рождения и т.д.)
vk.friends.get	Список друзей
vk.wall.get	До 100 последних записей на стене пользователя
vk.photos.getAll	Все фотографии в альбомах пользователя

Данные, которые подлежат анализу, являются открытыми и общедоступными. Библиотека работает через объект класса VkApi. При инициализации объекта ему передаются логин и пароль от аккаунта, с которого будет производиться «сканирование». Затем у объекта класса VkApi вызывается метод «get_api», он возвращает класс «VkApiMethod».

Графический интерфейс целесообразно разрабатывать с использованием библиотеки PyQt 5 [7]. Библиотека обладает всем необходимым набором инструментов для создания качественного пользовательского интерфейса и является кроссплатформенной.

Для анализа текстовой информации применяются три библиотеки: Rymorphy2 [8], PyEnchant [9], Python-Levenshtein [10]. Первая используется для приведения слов к инфинитиву и отбора имен собственных. Две остальные – для исправления орфографических ошибок в тексте.

Задача анализа текста – определить тематику сообщений.

Вначале текст приводится к нижнему регистру, убираются пробелы перед первым словом и после последнего слова. Текст разбивается на слова. Каждое из них «очищается» от сторонних символов, если таковые имеются. Это позволяет обрабатывать даже искаженные слова (например, часто букву «а» заменяют символом «@», используется транслит и т.д.).

В результате очистки может получиться несколько вариантов одного слова. В таком случае средствами библиотеки rymorphy2 вычисляется вероятность того, что слово написано правильно, и выбирается слово с наибольшей вероятностью.

Далее каждое слово проверяется на орфографические ошибки. В случае наличия ошибки ищутся возможные варианты замены слова. Библиотека pyenchant предоставляет список возможных замен, которые обрабатываются двумя методами: поиском среднего значения на основании расстояния Левенштейна [11] и нахождением наиболее подходящего слова методом сходства Джаро – Винклера.

Расстояние Левенштейна – метрика определения минимального количества посимвольных операций (удаление, вставка, замена), необходимых для превращения цепочки символов s_1 с количеством символов n в другую, s_2 с количеством символов m [11]:

$$L_{s_1 s_2}(i, j) = \min(L_{s_1 s_2}(i-1, j) + 1, L_{s_1 s_2}(i, j-1) + 1), L_{s_1 s_2}(i-1, j-1) + \varphi(s_{1i}, s_{2j}),$$

$$\varphi(s_{1i}, s_{2j}) = \begin{cases} 1, & \text{если } s_{1i} \neq s_{2j}, \\ 0, & \text{если } s_{1i} = s_{2j}, \end{cases}$$

где $i = 1 \dots n, j = 1 \dots m$; $L_{s_1 s_2}$ – расстояние Левенштейна между цепочками символов s_1, s_2 ; i – количество шагов удаления символов из первой цепочки; j – количество шагов вставки в первую цепочку.

На основании вычисленного расстояния поиск приближенной обобщенной медианной строки осуществляется с помощью «жадного алгоритма» [10].

Сходство Джаро – Винклера – мера схожести строк для измерения расстояния между двумя цепочками символов, которая вычисляется на основании расстояния Джаро [12].

Вычислить расстояние Джаро между двумя цепочками символов s_1 и s_2 можно следующим образом:

$$\begin{cases} d_j = 0, & \text{когда } m = 0, \\ d_j = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right), & \text{в остальных случаях,} \end{cases}$$

где $|s_i|$ – длина строки s_i ; m – число совпадающих символов; t – половина числа транспозиций.

Метрика Джаро – Винклера вычисляется по формуле

$$d_w = d_j + (lp(1 - d_j)),$$

где d_j – расстояние Джаро для строк s_1 и s_2 , l – длина общего префикса от начала строки до максимум четырех символов, p – постоянный коэффициент масштабирования для корректировки оценки в сторону повышения для выявления наличия общих префиксов.

Коэффициент p не должен превышать 0,25, потому что в ином случае расстояние может стать больше 1. Стандартное значение константы: $p = 0,1$.

Для каждого предложенного на замену слова вычисляется расстояние Джаро – Винклера. Из всех вариантов выбирается тот, у которого наибольшая метрика. Если по двум метрикам результат совпадает, то возвращается предложенное слово.

Все очищенные и проверенные на орфографию слова соединяются в сплошной текст. В этом тексте ищутся слова, которые находятся в тематическом словаре. Тематический словарь задает ключевые слова, сгруппированные по тематикам текстов. Частота появления слов из соответствующих тематических групп определяет тематическую окраску текста и позволяет выявлять требуемый контент. Добавлять слова в словарь можно через пользовательский интерфейс или путем загрузки из файла. Слова в словарь добавляются в форме инфинитива.

Модуль построения социального графа используется для получения целевых пользователей для поэтапного анализа (рис. 2).

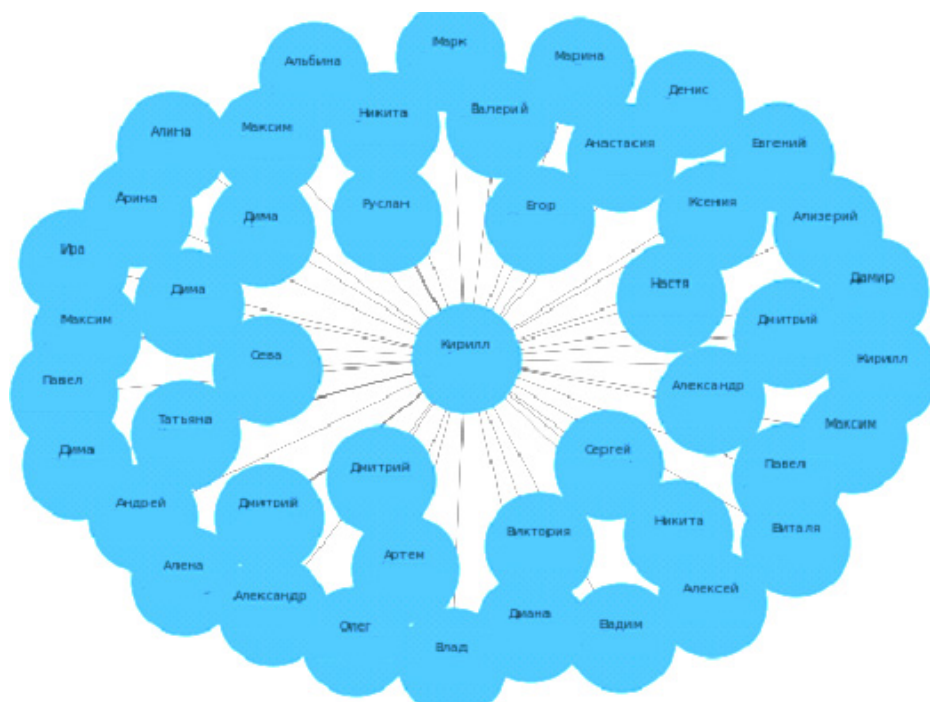


Рис. 2. Пример построения социального графа

На начальном этапе работы программы ей необходимо явно указать профили для анализа. В дальнейшем она с помощью социального графа проанализированных профилей определяет наиболее релевантные задачам анализа профили и добавляет их в очередь для последующего анализа.

Вариант реализации

Для реализации предлагаемого программного обеспечения возможно использование языка программирования Python 3.8 и IDE PyCharm Community Edition 2020.2. Достоинством этих инструментов является их свободное распространение и кроссплатформенность.

Реализация программного обеспечения позволит автоматизировать деятельность по анализу контента социальных сетей для решения задач менеджмента контингента вузов и других образовательных организаций. Алгоритм может определить тематику текста по наличию в нем слов, заранее заданных оператором.

Предлагаемый вид главного меню пользовательского интерфейса показан на рис. 3.

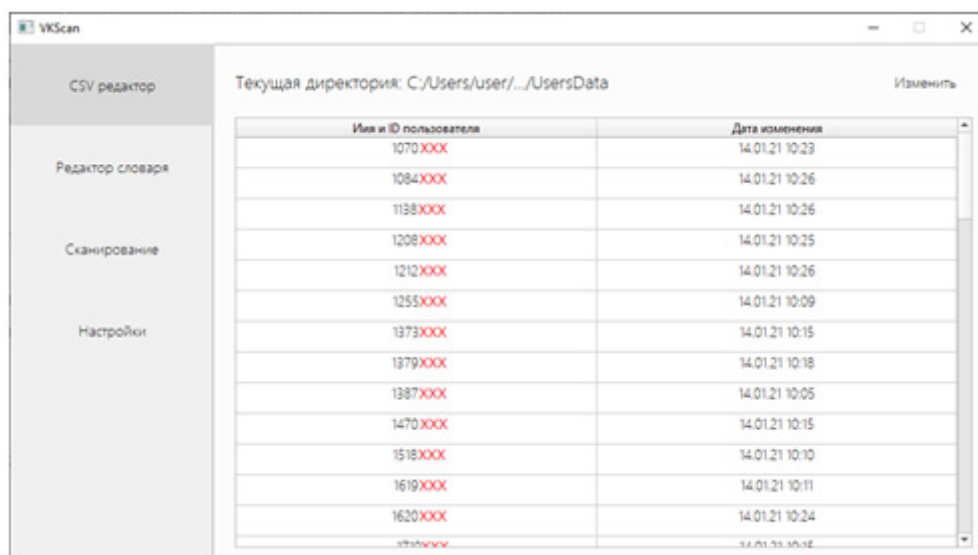


Рис. 3. Главное меню пользовательского интерфейса

В четырех основных вкладках представляется первичная информация о параметрах и результатах анализа:

- «CSV-редактор» для просмотра и редактирования текстовых данных;
- «Редактор словаря» для изменения в списке слов, которые ищутся в текстовых данных;
- «Настройки» для установки или смены аккаунта «ВКонтакте», при помощи которого происходит «сканирование».

В окне «Сканирование» (рис. 4) представлена статистика анализа данных. Также в этом окне можно вручную указать профили на сканирование или загрузить их из текстового файла.

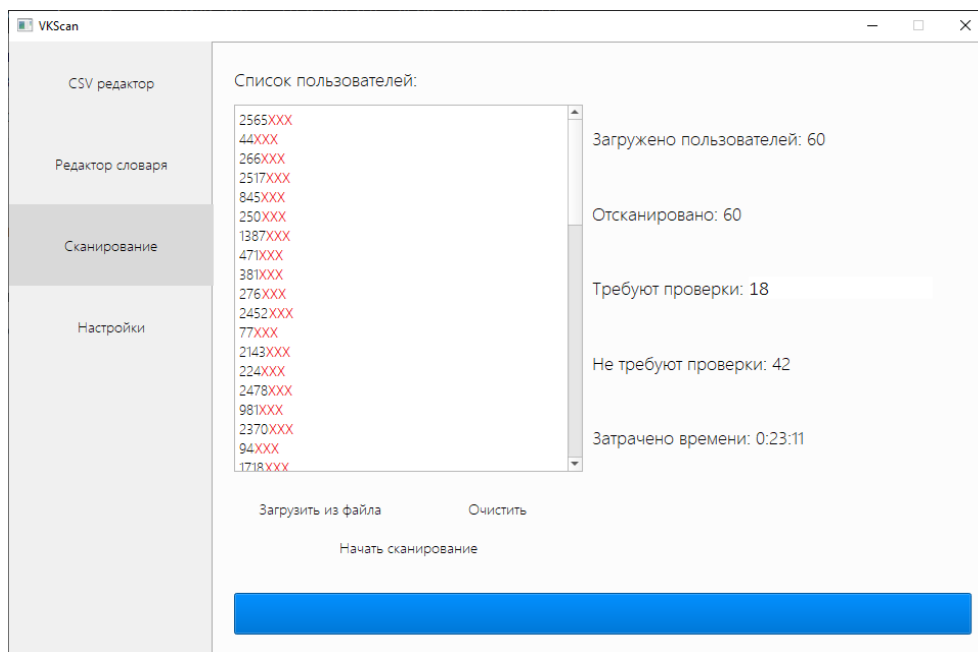


Рис. 4. Окно сканирования

Для тестирования программы было создано 60 тестовых профилей, сообщения которых были сформированы заранее обозначенными тематиками. Некоторые посты 18 профилей содержали измененные путем замены символов слова ненормативной лексики. Тематический словарь содержал 65 уникальных слов, которые алгоритм должен был распознать в записях на страницах тестовых пользователей. По результатам проверки «чистыми» оказались 42 профиля пользователей, тематика сообщений которых была правильно определена в 80 % случаев. При обработке сообщений 18 остальных профилей алгоритм распознал наличие нецензурных выражений. Общее время на проверку всех профилей 23 мин и 11 с.

Заключение

Предложена архитектура системы для сбора и анализа данных пользователей социальных сетей. На основе предложенной архитектуры возможен анализ данных из социальных сетей для решения задач образовательной организации – определения интересов учащихся, выявления негативных проявлений в общении и коммуникациях, отслеживания трудовых успехов выпускников.

Разработаны приложение, позволяющее в автоматическом режиме анализировать профили пользователей социальной сети «ВКонтакте» на предмет наличия тематического контента, а также утилита для построения социального графа, дающая возможность визуализировать взаимосвязи между людьми.

Предложенную программу можно преобразовать в клиент-серверное приложение. Все вычисления (сбор, анализ данных) будут производиться на сервере, в то время как клиенты смогут получать готовый результат. Функционал программы возможно расширить: усовершенствовать модуль анализа текста за счет использования нейронных сетей, добавить модуль анализа графической информации.

Список литературы

1. Полякова А. В., Бородина И. П. Роль таргетинга в интернет-маркетинге // Таврический научный обозреватель. 2015. № 5-1. С. 116–118.
2. Lammel R. Google's MapReduce Programming Model – Revisited // Science of Computer Programming. 2008. Vol. 70, № 1. P. 1–30.
3. Apache Hadoop. URL: <https://hadoop.apache.org/> (дата обращения: 06.03.2021).
4. Young T., Hazarika D., Poria S., Cambria E. Recent Trends in Deep Learning Based Natural Language Processing. arXiv:1708.02709v8 2018.
5. Документация vk_api. URL: <https://vk-api.readthedocs.io/en/latest/> (дата обращения: 25.01.2021).
6. Список методов | Разработчикам | ВКонтакте. URL: <https://vk.com/dev/methods> (дата обращения: 25.01.2021).
7. What is PyQT 5? URL: <https://www.riverbankcomputing.com/software/pyqt/> (дата обращения: 06.03.2021).
8. Морфологический анализатор pymorphy2. URL: <https://pymorphy2.readthedocs.io/en/stable> (дата обращения: 06.03.2021).
9. PyEnchant. URL: <https://pyenchant.github.io/pyenchant/> (дата обращения: 06.03.2021).
10. Python-Levenshtein master doc. URL: <https://rawgit.com/ztane/python-Levenshtein/master/docs/Levenshtein.html> (дата обращения: 17.01.2021).
11. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии наук СССР. 1965. Т. 163. С. 845–848.
12. Winkler W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi–Sunter Model of Record Linkage. 1990. P. 354–359.

References

1. Polyakova A.V., Borodina I.P. The role of targeting in Internet marketing. *Tavrisheskiy nauchnyy obozrevatel' = Tavrichesky Scientific Reviewer*. 2015;5-1:116–118. (In Russ.)
2. Lammel R. Google's MapReduce Programming Model – Revisited. *Science of Computer Programming*. 2008;70(1):1–30.
3. *Apache Hadoop*. Available at: <https://hadoop.apache.org/> (accessed 06.03.2021).
4. Young T., Hazarika D., Poria S., Cambria E. *Recent Trends in Deep Learning Based Natural Language Processing*. arXiv:1708.02709v8 2018.
5. *Dokumentatsiya vk_api = vk_api documentation*. Available at: <https://vk-api.readthedocs.io/en/latest/> (accessed 25.01.2021). (In Russ.)
6. *Spisok metodov | Razrabotchikam | VKontakte = List of methods | To Developers | VKontakte*. Available at: <https://vk.com/dev/methods> (accessed 25.01.2021). (In Russ.)
7. *What is PyQT 5?* Available at: <https://www.riverbankcomputing.com/software/pyqt/> (accessed 06.03.2021).
8. *Morfologicheskiy analizator pymorphy2 = Morphological analyzer pymorphy2*. Available at: <https://pymorphy2.readthedocs.io/en/stable> (accessed 06.03.2021). (In Russ.)
9. *PyEnchant*. Available at: <https://pyenchant.github.io/pyenchant/> (accessed 06.03.2021).
10. *Python-Levenshtein master doc*. Available at: <https://rawgit.com/ztane/python-Levenshtein/master/docs/Levenshtein.html> (accessed 17.01.2021).
11. Levenshteyn V.I. Binary codes with correction of drop-outs, inserts and substitutions of characters. *Doklady Akademiy nauk SSSR = Reports of the USSR Academy of Sciences*. 1965;163:845–848. (In Russ.)
12. Winkler W.E. *String Comparator Metrics and Enhanced Decision Rules in the Fellegi–Sunter Model of Record Linkage*. 1990:354–359.

Информация об авторах / Information about the authors

Кирилл Александрович Лехов

студент,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: kirill.lekhov@pnzgu.ru

Kirill A. Lekhov

Student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Дмитрий Денисович Сперанский

студент,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: dmitry.speransky@pnzgu.ru

Dmitriy D. Speransky

Student,
Penza State University
(40 Krasnaya street, Penza, Russia)

Максим Александрович Митрохин

доктор технических наук, доцент,
заведующий кафедрой вычислительной
техники,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: mmax83@mail.ru

Maxim A. Mitrokhin

Doctor of technical sciences,
associate professor,
head of sub-department
of computer engineering,
Penza State University
(40 Krasnaya street, Penza, Russia)

Надежда Сергеевна Карамышева

кандидат технических наук, доцент
кафедры вычислительной техники,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: karamyshevans@yandex.ru

Nadezhda S. Karamysheva

Candidate of technical sciences,
associate professor of sub-department
of computer engineering,
Penza State University
(40 Krasnaya street, Penza, Russia)