КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА

УДК 004.912 ISSN 1995-5499
DOI: https://doi.org/10.17308/sait/1995-5499/2022/3/139-151
Поступила в редакцию 10.08.2022
Подписана в печать 30.09.2022

ПРИМЕНЕНИЕ МОДЕЛИ ДИСТИЛЛЯЦИЙ ЗНАНИЙ BERT ДЛЯ АНАЛИЗА НАСТРОЕНИЙ ТЕКСТА

© 2022 Н. Е. Косых✉

Петербургский государственный университет путей сообщения Императора Александра I
Московский пр., 9, 190031 Санкт-Петербург, Российская Федерация

Аннотация. Увеличение сложности архитектур нейронных сетей и увеличение объема обрабатываемых данных в процессе машинного обучения ставит вопрос о необходимости применения более производительных подходов, которые позволили бы оптимизировать процесс разработки моделей классификации текста для решения задач анализа настроений. Целью работы является обучение и оптимизация нейросетевой модели-трансформера для классификации данных в рамках решения анализа настроений русскоязычного текста. В рамках научного исследования предлагается применение предварительной обученных моделей двунаправленного кодирования BERT, а также модели дистилляции знаний ruBERT-tiny для выполнения мультиклассовой классификации текста для анализа настроений пользовательского текста. Применение этапа уплотнения данных для моделей дистилляции знаний позволяет оптимизировать этап обучения моделей классификации текста. Разработана программа на языке программирования Python с использованием библиотек машинного обучения. Техническое решение позволяет апробировать предобученные модели классификации данных, на основе которых создать оптимизированные модели классификации для анализа настроений пользовательских текстов с учетом специфики предметной области.

Ключевые слова: анализ настроений, классификация настроений текста, дистилляция, модель обучения, предварительная обработка данных, нормализация данных, BERT, ruBert, Python.

ВВЕДЕНИЕ

Анализ настроений, также известный как анализ мнений или сентиментный анализ, представляет собой алгоритм, используемый для определения мнения людей по определенной теме. С ростом социальных сетей, блогов, форумов с онлайн обзорами, крупные компании осознали, что знание настроений своей аудитории может помочь получить представление о поведении пользователей, и использовать полученную информацию для маркетинговой составляющей и/или повышения лояльности к бренду, мониторинга конкурентного поля. По состоянию на 2021 год в мире насчитывается 4,14 млрд пользователей социальных сетей (54 % населения мира), которые проводят в них в среднем 2,5 часа в день [1].

Основная задача анализа настроений заключается в классификации полярности текста на уровне документа или предложения, входящего в этот набор. Настроение текста — это отношение субъекту к некоторому объекту реального мира, другому субъекту или явлению. Настроение, как правило, выражается через слова, имеющих заранее определенную эмоциональную составляющую, характеристику или аспект.

✉ Косых Никита Евгеньевич
e-mail: nikitosagi@mail.ru

Анализ настроений — одна из многих задач обработки естественного языка, которая помогает компьютерам понимать, интерпретировать и использовать мировые языки. Анализ настроений текста превращает большое количество неструктурированного текста в упорядоченные данные, используя два типа представления:

– контекстно-свободное языковое представление: каждое слово в предложении уникально, вне зависимости от других слов;

– контекстное языковое представление: каждое слово представлено относительно других слов в предложении.

Поскольку контекстное языковое представление учитывает связь каждого слова с другими словами, появляется понятие о направленности:

– однонаправленные модели: предсказывают следующее слово на основе предыдущих слов;

– двунаправленные модели: предсказывают слово, основываясь на его контексте окружающих его слов.

Прежде чем решать задачу анализа настроений исследуемого текста или предложения, необходимо обозначить основные подходы к анализу настроений:

– подход на основе лексики и правил для анализа настроений;

– подход, основанный на машинном обучении [2].

Первый из обозначенных подходов для анализа настроений основан на пользовательском словаре с определенным значением эмоциональной окраски в числовом эквиваленте для каждого слова. Второй подход предполагает обучение компьютерных систем классифицировать объекты и события, определять взаимосвязи между ними, а также строить прогнозы при вводе данных. На основании второго подхода может быть сформирована модель машинного обучения для решения задачи классификации настроений.

Самый распространенный подход к решению задачи анализа настроения — это двоичная классификация — разбиение элементов некоторого множества, состоящего из предложений, на два класса в зависимости от их эмоциональной окраски — положительный и отрицательный. Проблема такого подхода состоит в неоднозначности и неточности оценки предложений.

Для решения этой неоднозначности, как правило, используются наборы данных, содержащих третий дополнительный класс предложений с нейтральной эмоциональной окраской. Применение таких наборов данных позволяет создать более точную и адекватную модель классификации текстовых данных.

В представленном подходе для анализа настроений текстов предварительной обученная модель BERT дообучается на пользовательских наборах данных, содержащих двоичное или троичное разделение данных на классы по эмоциональной направленности. Предложенные нейросетевые модели имеют неоспоримое преимущество перед моделями нейронных сетей с долгой краткосрочной памятью (ДКП), которые страдали от потери информации в длинной последовательности текстов. Новые модели могут легко адаптироваться в структуру документа и понять контекст некоторого слова в предложении на основе предыдущих и последующих слов благодаря двунаправленному подходу.

Целью данного исследования является применение вышеописанных моделей семейства BERT для создания моделей классификации данных в рамках решения задачи анализа настроений текстов.

1. ПОДХОДЫ К МАШИННОМУ ОБУЧЕНИЮ

Первый из подходов — обучение с учителем объединят в себе алгоритмы и методы построения моделей классификации данных на основе множества входных данных. Часть данных, которые подаются на вход для обучения уже имеют принадлежность к определенному классу настроений. Эта принадлежность обозначается «меткой» класса, и на практике записывается как 0, 1, что в задачах анализа настроений текста соответствует классам: «отрицательный» и «положительный». В процессе обучения машина запоминает существующие метки класса, устанавли-

вает некоторые закономерности и использует их для прогнозирования меток классов для новых данных. Подход чаще всего используется для задач классификации и регрессии.

Второй из подходов — обучение без учителя предполагает, что в процессе обучения машина должна сама устанавливать закономерности и зависимости. Когда зависимость установлена происходит группировка данных на классы по одному или нескольким признакам. Чаще всего результаты такого обучения непредсказуемы, и этот подход не рекомендуется использовать для решения задач классификации каких-либо данных.

В качестве инструмента реализации подходов обучения используются нейронные сети, основные из которых рассмотрены в последующих разделах.

2. ПОСТАНОВКА ЗАДАЧИ ОБУЧЕНИЯ С УЧИТЕЛЕМ

В основе работы большинства нейронных сетей лежит подход к обучению с учителем, один из подходов, который используется для решения задач классификации.

Далее приведена формальная постановка задачи для обучения с учителем.

Дано:

– множество объектов $X = \{x_1, x_2, ..., x_m\}$ и множество меток (классов) $Y = \{y_1, y_2, ..., y_l\}$;

– размеченные данные вида $X_l, Y_l =< (x_{1:l}, y_{1:l}) >$, создающие совокупность обучающей выборки;

– неразмеченные данные вида $X_u = \{x_{l+1:m}\}$, данные не принимающие участие в обучении, образующие тестовую выборку;

– неизвестная целевая зависимость $f : X \to Y$.

Каждый элемент из входного набора данных описывается кортежем $< y_i, x_i >$, где x_i — строка из массива входных данных X и y_i — целевое значение (метка класса), принадлежащее множеству допустимых ответов Y. Входной набор данных разделяется на несколько подмножеств: обучающее и тестовое. Обучающий набор содержит в себе текстовые данные с эталонными значениями классов, а в тестовом наборе значения меток классов отсутствуют. Требуется построить такой алгоритм $a : X \to Y$, который приближал бы неизвестную целевую зависимость f как на элементах выборки, так и на все множестве X [3].

Результатом работы нейронных сетей является создание модели классификации данных, способной отнести единицу пользовательских данных к определенному классу и присвоить значение метки класса в зависимости от поставленной задачи. В процессе обучения модель пытается найти закономерности, которые можно использовать для классификации данных, не входящих в исходных обучающий набор. Проверка обученной модели на тестовом наборе данных дает представление о качестве обученной модели [18].

3. КОНФИГУРАЦИИ НЕЙРОННЫХ СЕТЕЙ

Искусственные нейронные сети (ИНС) создаются с помощью взаимосвязанных компонентов обработки данных, которые по своей структуре напоминают устройство человеческого мозга, имитируя работа биологической нейронной сети. ИНС состоят из нейронов — узлов сети, которые образуют слои, необходимые для обработки и передачи данных другим узлам сети. Узлы, как вершины в графах соединены ребрами, с обозначением веса, который влияет на силу сигнала и значение конечных данных сети.

Рекуррентные нейронные сети (РНС) до последнего времени являлись основным типом искусственной нейронной сети, которые использовались для обработки естественного языка. РНС распознают последовательные характеристики данных и используют закономерности для прогнозирования следующих наиболее вероятных значений. Такие сети наиболее эффективны в случаях, когда необходимо понимать контекст данных. Они отличаются от других типов нейронных сетей тем, что в них используются циклы обратной связи для обработки последовательностей данных, на основе которых формируется конечный результат. Такие циклы обратной связи дают возможность сохранятся инфор-

мации, что позволяет запоминать контекст исследуемых данных.

В некоторых случаях ИНС обрабатывают информацию в одном направлении — от входа к выходу, их называют сети прямого распространения сигнала. К таким сетям относятся сверточные нейронные сети (СНС), лежащие в основе систем распознавания образов. С другой стороны, РНС могут быть модифицированы, дополнительным уровнем вложенности для возможности обрабатывать информация в двух направлениях. РНС используют петли обратной связи, таких как обратное распространение ошибки во времени для возврата информации в сеть. Это связывает входы сети, что позволяет обрабатывать последовательные и изменяющиеся во времени данные.

Двунаправленные РНС (ДРНС) — еще один тип РНС, которая изучает прямое и обратное направление потоков данных, в рамках задачи обработки естественного языка можно рассматривать контекст спереди и позади слова. В обычной нейронной сети прямого распространения прямой проход используется для предсказания будущих значений [4], а обратный проход — для оценки прошлых значений. Однако такие проходы не выполняются одновременно как ДРНС.

Наиболее распространенной проблемой при работе с РНС является проблема исчезающего градиента. К градиентам относятся ошибки, возникающие в процессе обучения нейронной сети, а точнее вектор частных производных функций потерь по весам нейронной сети.

Обычные РНС, использующие градиентный метод обучения, показывают убывающую эффективность по мере роста и усложнения структуры сети. Одним из решений этой проблемы является применение сетей долгой краткосрочной памяти (ДКП). РНС, построенные с использованием ДКП, распределяют данные по краткосрочным и долгосрочным ячейкам памяти и позволяет определить какие данные можно забыть, а какие необходимо запомнить и вернуть в сеть.

Для дальнейшего улучшения качества обучения было предложено использовать механизм self-attention [5] (самовнимание), описанный в разделе 4.1.

4. ТРАНСФЕРНЫЕ МОДЕЛИ ОБУЧЕНИЯ

Недавние достижения в области обработки естественного языка показали, что трансферное обучение помогает достичь самых передовых результатов для решения задач путем настройки предварительно обученных моделей, вместо обучения моделей с нуля, как это происходит в классических РНС. В данном исследовании рассматривается один из подходов к трансферному обучению BERT (Bidirectional Encoder Representations from Transformers — двунаправленные представления кодировщика от трансформеров) — метод машинного обучения, разработанный Google на основе механизма Transformers (блоки трансформера) [6] для обработки каждого слова из исходного набора данных в полном контексте всех слов до и после. Главная задача блоков трансформеров — установить, взаимосвязь слов из текста, поданного на вход.

4.1. Механизм самовнимания

Базовая модель семейства BERT достаточно долго обучается на огромных корпусах текстов, пропуская через себя миллионы документов и постепенно осваивая язык, грамматику и сущность слов. В дальнейшем модель можно дообучить на пользовательских наборах данных для выполнения конкретной прикладной задачи, таких как анализ настроений, классификации комментариев.

Подобные сетей основаны на архитектуре «трансформер», которая используется для моделирования задач понимания языка, полагаясь на механизмы самовнимания для построения глобальных зависимостей между входами и выходами. Далее рассматривается принцип работы механизма самовнимания.

Пусть на вход сети передано предложение, состоящие из слов $x_1, x_2, ..., x_n$, при это уже в векторном представлении. Предположим, что необходимо установить зависимость x_4 от всех остальных слов. Обозначим зависимость как y_4, которая вычисляет по выражению 1.

$$y_3 = \sum_{i=1}^{n} w_{4i} \cdot x_i, \quad (1)$$

где w_{4i} — веса семантической близости слова, получены как скалярное произведение слова x_4 со словом x_i; x_i — векторное представление некоторого слова, входящего в состав предложения.

При этом можно установить, каким образом слово x_i оказывает «внимание» на слово x_4. Операция самовнимания входит в блок трансформера, и с каждым блоком происходит переход на абстракцию более высокого уровня относительно исходных слов, что позволяет лучше устанавливать связь слов друг с другом.

В чистом виде блок трансформера состоит из двух компонентов — кодировщика и декодера. Первый компонент считывает входные данные (рис. 1), а второй выполняет задачу предсказания. Внутренняя архитектура блока состоит из следующих элементов:

– сначала текст разбивается на слова, а потом слова сопоставляются с их векторными представлениями;

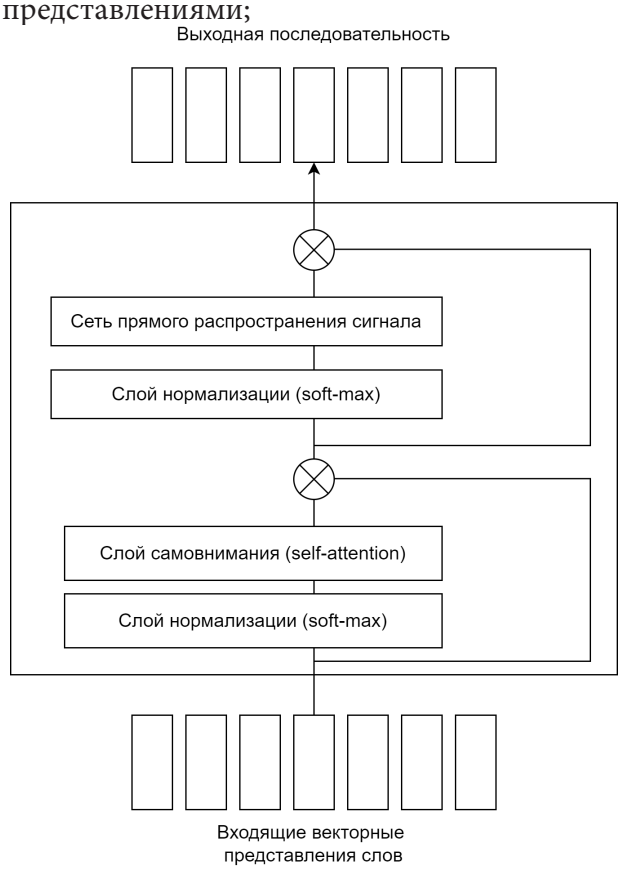

Рис. 1. Внутреннее устройство блока трансформера для кодера
[Fig. 1. Internal structure of the transformer unit for the encoder]

– позиционные кодировщики вводят информацию о позиции входного слова;

– уровень самовнимания кодирует информацию о входной последовательности с учетом контекста;

– слой прямого распространения сигнала, который работает как статическая память, одна из его выходных последовательностей является константой.

– перекрестное внимание декодирует выходную последовательность различных входов и модальностей.

Слой самовнимания принимает n входов и возвращает n выходов. Слой самовнимания создает три вектора для каждого входящего числового представления слова: вектор запроса, вектор ключа и вектор значения. Эти векторы создаются с помощью перемножения входящего вектора на три матрицы, которые были получены при обучении. В итоге после перемножения мы получаем проекции W^Q, W^K, W^K для каждого слова в составе входящего предложения. Далее рассчитываются коэффициенты самовнимания для каждого слова, входящего в предложения, путем скалярного произведения вектора запроса на вектор ключа этого слова. Полученные скалярные величины делятся на квадратный корень размерности вектора ключа — $\sqrt{64}$, а затем полученный результат пропускается через функцию нормализации. После применения функции скалярные величины представляются вещественным числом в интервале [0,1] и их сумма равна 1. Каждый вектор значения умножается на коэффициент нормализации. Далее взвешенные векторы значения складываются, образуя выход слоя самовнимания для этого слова.

После всех вычислений мы получаем вектор, который можно дальше передать в сеть прямого распространения сигнала. На практике для ускорения вычислений используются матрицы вместо векторов.

Таким образом появляется возможность учитывать при обучении позицию слова в контексте предложения. Для этого необходимо добавить в начала векторного представления слова вектор-позицию (рис. 2) такой же размерности.

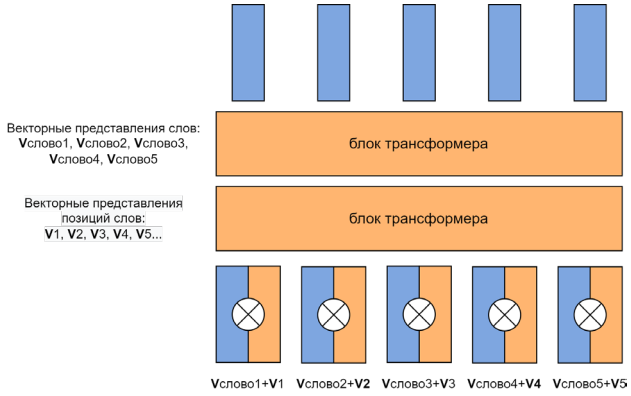

Рис. 2. Входные векторы предложений, учитывающие порядок слов
[Fig. 2. Input vectors of sentences considering the word order]

В результате для каждого слова генерируется вектор, заключающий в себе значения слова и номер позиции в предложении. Похожие по смыслу слова имеют близкие числовые значения внутри векторов.

4.2. Этап предварительной подготовки данных BERT

Для каждого кодировщика BERT существует своя модель предварительной обработки данных — набор операторов для приведения исходного текста в числовые представления, ожидаемые кодировщиком на входе. Каждая из моделей поставляется со своими заранее сформированным словарем и связанной с ним логикой нормализации текст, на практике не требует точной настройки параметров обучения.

Обработка текста условно разделена на шесть этапов [7], как показано на рис. 3. Первый этап — разбиение слов на токены. В базовой модели BERT использует словарь в размере 30552 слов. Процесс разбиения на слова (токенизация) включает в себя разбиение входного потока текстовых данных на список слов [11], доступных в словаре. Отсутствующие слова постепенно разбиваются на морфемы, а затем представляются группой морфем. Поскольку морфемы являются частью словаря, можно получить векторное представление этих морфем, а контекст слова — это просто комбинации этих морфем. Далее все предложения усекаются до единой длины — это еще одно из важных условий для успешного обучения.

На выходе блока предварительной подготовки данных остается числовое представление вектора с указанием индекса слов в словаре с учетом их семантической близости.

Инкапсулированные базовые операции этапа предварительной подготовки текста внутри моделей BERT позволяют на выходе получить более чистый и лаконичный программный код при реализации этапов обучения и прогнозирования пользовательских значений в реализуемых задачах.

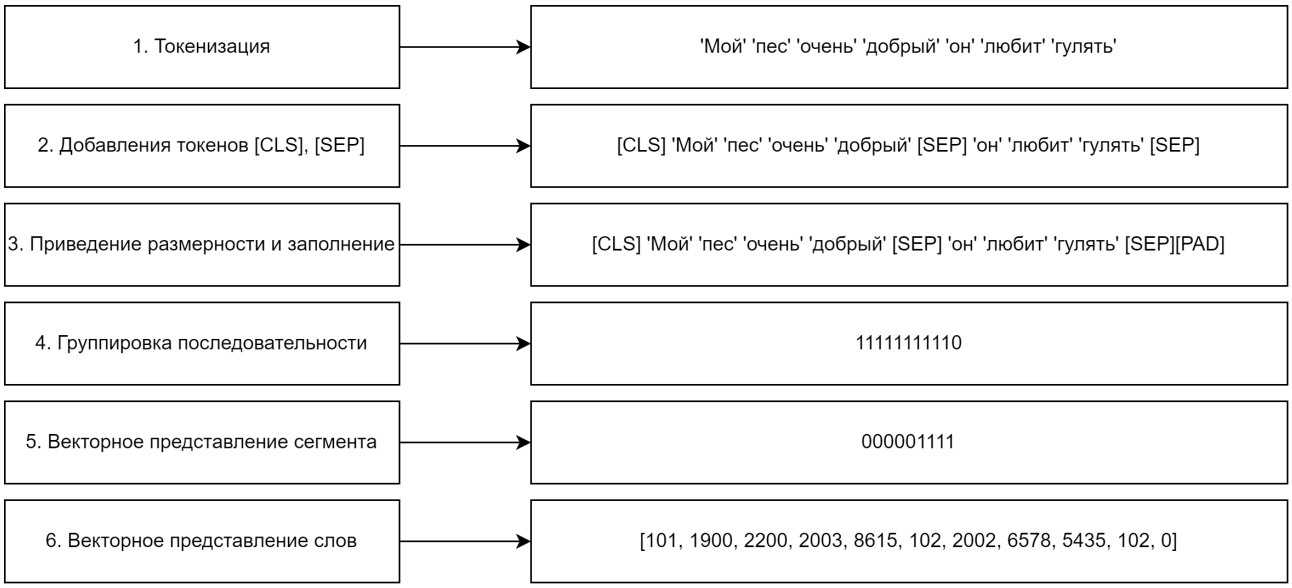

Рис. 3. Стек операции метода предварительной подготовки текста
[Fig. 3. The operation stack of the text preprocessing method]

5. ОБУЧЕНИЕ МОДЕЛЕЙ СЕМЕЙСТВА BERT

Основная задача состоит в том, чтобы протестировать наиболее производительные модели из семейства BERT для выполнения задачи классификации данных в рамках анализа тональности русскоязычного текста.

Вся техническая реализация выполнена в интерактивно облачной среде Google Colab, которая позволяет объединить в одном документе исполняемый код и форматированный текст. Среда позволяет выполнять код Python, используя вычислительные мощности графических процессоров (GPU). Такой подход позволяет заниматься исследованием данных: разрабатывать и тестировать новые модели машинного обучения, а также визуализировать полученные результаты внутри среды.

Техническая часть исследования будет разделена на два этапа. Первый этап относится к выбору подходящего набора данных для выполнения поставленной задачи и его первичная обработка (рис. 4). Этап включает в себя загрузку пользовательских данных и библиотек для работы с ними, предварительная подготовка данных для обучения модели, включая дополнительные этапы нормализации (лемматизация данных) и маскирования слов.

Следующий этап включает в себя работу с предобученными моделями на базе механизма трансформеров. Алгоритм работы с моделями включает себя следующие под этапы: выбор необходимой модели с сайта-репозитория, выгрузка данных в рабочую среду, векторное представление входных последовательностей (токенизация), обучение выбранной модели, тестирование моделей [8]. После тестирования модели необходимо убедиться, удовлетворяет ли модель необходимым критериям качества для решения задачи классификации данных, если нет, то есть несколько вариантов дальнейшего развития событий — или возвращаемся на этап выбора моделей, для выбора модели, обученной на другом корпусе текстов или переходим к этапу точной настройки параметров обучения модели.

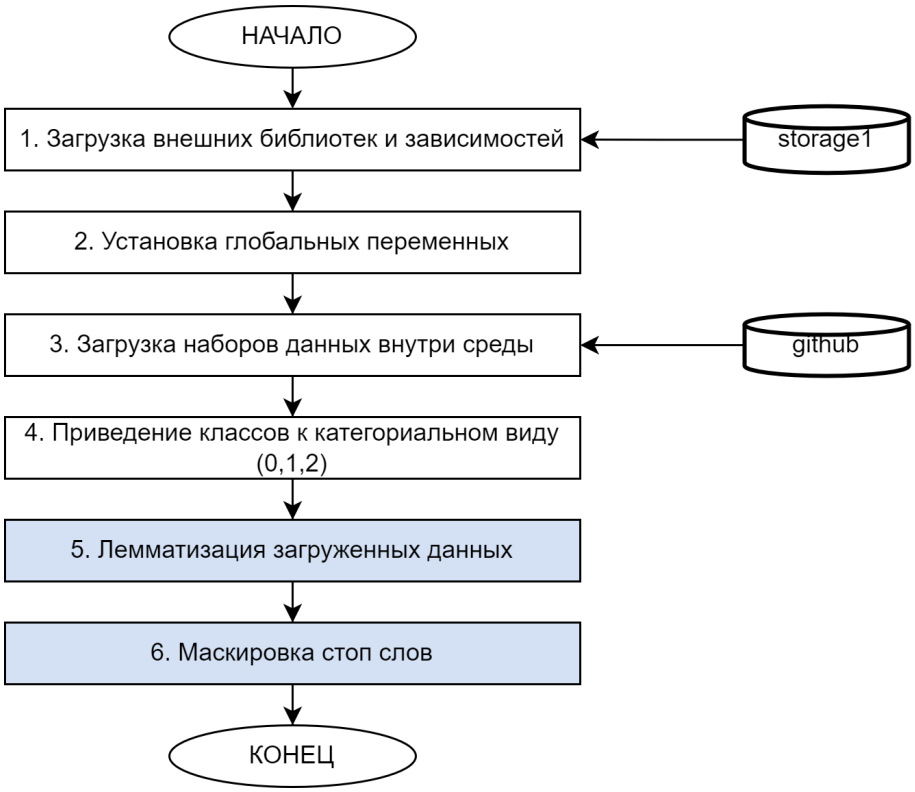

Рис. 4. Этап предварительной подготовки данных; блок 6 — дополнительный шаг маскировки стоп-слов
[Fig. 4. Data preparation stage; block 6 — additional stop-word masking step]

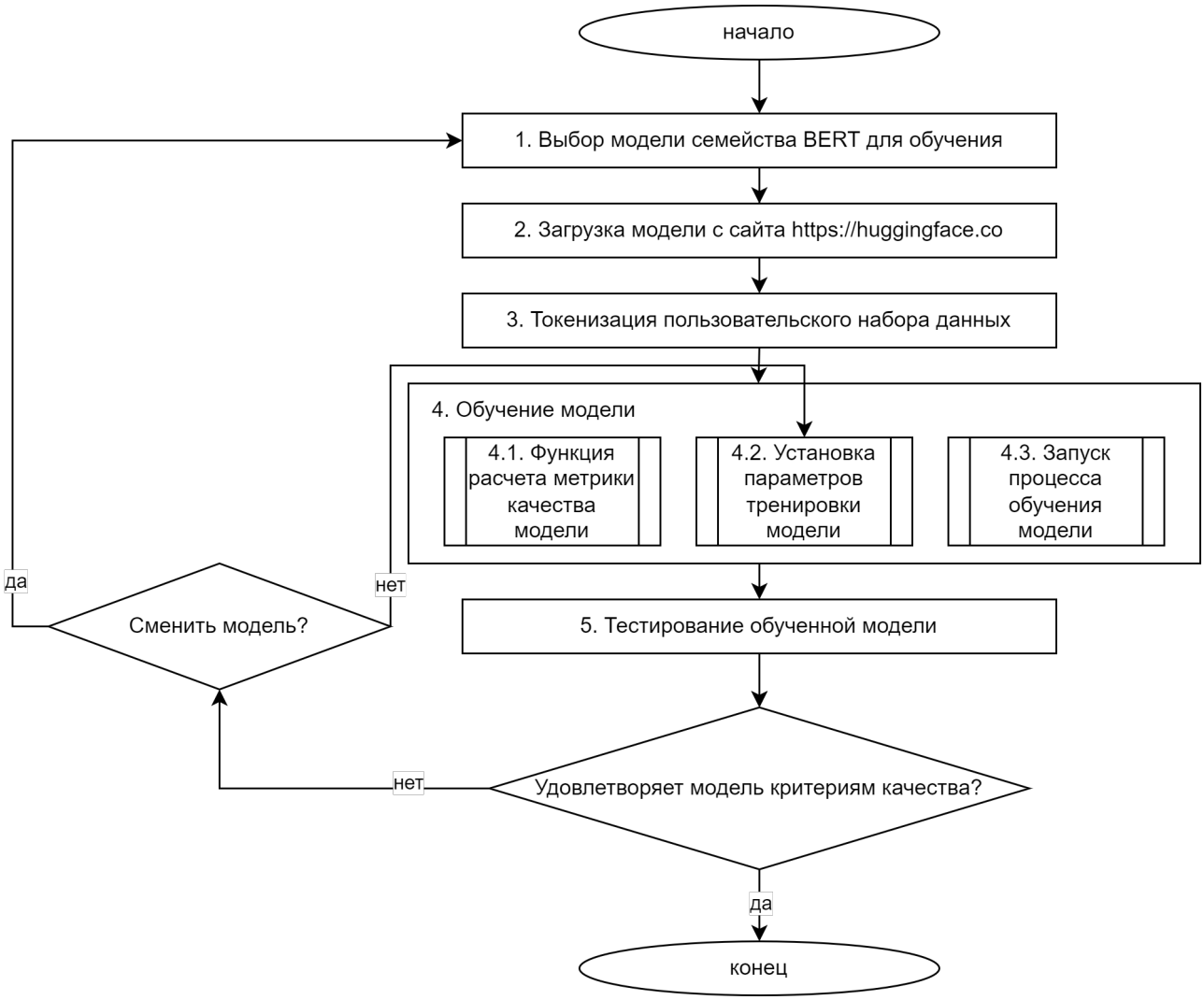

Рис. 5. Алгоритм обучения модели BERT
[Fig. 5. BERT training algorithm]

Как показывает практика, то многообразие настроек обучения (рис. 5 блок 4.2–4.3) сводится к выбору количества эпох обучения сети, размера обучающей выборки и выбор метода оптимизации нейронных сетей.

5.1. Выбор и оптимизация модели семейства BERT

Опытное использование предварительно обученных языковых моделей BERT значительно улучшило производительность многих задач обработки естественного языка. Однако, обучение таких моделей под решение конкретных задач обычно требует больших вычислительных ресурсов. Поэтому обучение сложно выполнять на устройствах с ограниченными ресурсами. Чтобы ускорить вывод и уменьшить размер модели при минимальных потерях в качественных характеристиках, было предложено использовать модель на основе блоков трансформеров — ruBert-tiny [9], созданную по технологии дистилляций, т.е. путем перекладывания знаний из одной модели в другую. В качестве основы была выбрана мультиязычная модель BERT, словарь которой содержал 120.000 токенов, был урезан до 30.000 самых часто встречающихся слов. Размер векторного представления по сравнению со своей старшей моделью уменьшен с 768 до 312, а число слове с 12 до 3 соответственно. В качестве учителей для дистилляции модели были выбраны те, которые идеально подходят для задач классификации

текста RuBert [10]. Насколько эффективно использовать урезанную модель, обученную для выполнения разных задач? Эффективность использования модели сильно зависит от количества обрабатываемых предложений за единицу времени. В табл. 1 приведены сравнения в скорости обработки данных для других общедоступных моделей BERT [9], понимающих русский язык. Скорость указана в расчете на одной предложение из Лейпцигского веб-корпуса русского языка.

Все расчёты были выполнены на Colab (Intel(R) Xeon(R) CPU @ 2.00GHz и Tesla P100-PCIE) [13] с размером партии обучения — 1. Как можно заметить по данным таблицы, модель ruBert-tiny работает раз в 20 быстрее своих тяжеловесных соседей и легко может уместиться на бюджетные хостинги, а также обучаться с использованием облачных ресурсов от Google Colab на пользовательских наборах данных.

5.2. Выбор набора данных для обучения

Для этапа обучения был выбран набор данных, собранный из отзывов о товарах из категории «Женская одежда и аксессуары» сайта одного из крупнейших российских онлайн магазинов. Набор данных состоит из 90000 автоматически классифицированных отзывов. Согласно полученным данным, отзывы содержат категориальные оценки по 5-бальной шкале. Как правило, разное количество классов и разное количество экземпляров классов приводит к потере качества для модели классификации. Все 1 и 2-бальные отзывы были объединены в единый в класс — негативные отзывы, 4 и 5-бальные отзывы в класс положительных отзывов и 3-бальные образовали дополнительный класс нейтральный отзывов. Таким образом, получен сбалансированный набор данных для обучения, содержащий по 30000 элементов каждого класса.

Первым делом необходимо привести признаки классов из обучающей выборки из текстового в числовое представление, где нейтральные метки классов заменяются на — 0, положительные — 1, отрицательные — 2. Далее полученный набор необходимо разделить на тренировочную и тестовые выборки данных в соотношении 1 к 9.

5.3. Обучение модели на пользовательских данных

Следующим шагом определяются пользовательские функции для обучения модели, расчета метрики качества, задания параметров обучения модели. Далее вызывается функция, запускающая процесс обучения модели.

Модель ruBert-tiny была обучена на различных срезах из пользовательского набора данных размерами: 3000, 9000, 27000 предложений. Для каждого из срезов данных проводилось обучение с изменением количества эпох обучения от 1 до 2. Результаты процесса обучения будут оценены по нескольким показателям, таким как величина функции потерь, точность обученной модели [17]. Показатели качества модели сильно зависят от размера исходного набора данных, выбранного для обучения модели, а также стоит отметит, что применение нескольких эпох обучения дает небольшой прирост к увеличению показателей качества модели. Точность классификации данных оценена на тестовом на-

Таблица 1. Сравнение моделей семейства BERT
[Table 1. Comparison of BERT family models]

Модель	Скорость (CPU)	Скорость (GPU)	Вес на диске
ruBert-tiny	6 мс	3 мс	45 мб
BERT-Base	125 мс	8 мс	680 мб
DeepPavlov (RuBert-Base-Cased-Sentensed)	110 мс	8 мс	680 мб
LaBSE	120 мс	8 мс	1.8 гб
sBert-Large	420 мс	16 мс	1.6 гб

боре данных для 2 эпох обучения, составила 76,4 % против 75,8 % для 1 эпохи обучения. При этом стоимость функции потерь практически не изменяется и зависит в основном от размера обучающей выборки.

ЗАКЛЮЧЕНИЕ

На сегодняшний день протестированная модель используется для получения самых точных результатов на различных сложных задачах в области обработки естественного языка. По сравнению с объемными моделями как Tatyana-ruBert [14], ruBert-base [15] можно получить приближенную по качеству модель классификации данных с огромным преимуществом в скорости обучения и тестирования.

Пиковая точность модели Tatyana-ruBert составила 77.1 %, модели ruBert-base 74,2 % и модели ruBert-tiny 71 %. Для большей наглядности к графикам было применено экспоненциальное скользящее среднее сглаживание. Однако, стоит отметить, что результаты небольшого преимущества в точности классификации данных у моделей ruBert-base и Tatyana-ruBert сводятся на нет из-за временной сложности работы моделей, обучения, превышающих в 10 раз временную сложность работы ruBert-tiny (рис. 6).

Как видно из графика системное время, затраченное на обучение моделей Tytyana-ruBert и ruBert-base более чем в 10 раз, превышает время, затраченное на обучение модели ru-Bert-tiny с учетом изменения числа эпох обучения.

Такие легковесные дистиллированные модели крайне удобны для обучения в условиях ограниченных аппаратных ресурсов, а также обеспечивают необходимый уровень качества для внедрения в готовые решения классификации данных.

БЛАГОДАРНОСТИ

Исследования по теме проводились в рамках реализации Федеральной программы поддержки университетов «Приоритет 2030».

КОНФЛИКТ ИНТЕРЕСОВ

Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Колошина, В. С.* Анализ влияния социальных сетей в политической сфере общества: от мировой к Российской практике (на

Рис. 6. Оценка вычислительной сложности моделей семейства BERT
[Fig. 6. Evaluating the computational complexity of BERT family models]

анализе социальных сетей российских политических партий) / В. С. Колошина, М. Е. Родионова // Наука и образование в наши дни: фундаментальные и прикладные исследования: Материалы XLIII Всероссийской научно-практической конф. (Ростов-на-Дону, 23 декабря 2021 г). – Ростов-на-Дону, 2021. – С. 412–420.

2. *Черкасов, Д. Ю.* Машинное обучение / Д. Ю. Черкасов, В. В. Иванов // Наука, техника и образование. – 2018. – № 5(46). – С. 85–87.

3. *Мальчиц, В. С.* Применение методов машинного обучения для классификации новостей / В. С. Мальчиц // Молодежь XXI века: шаг в будущее: Материалы XX региональной научно-практической конференции: в 3 томах, Благовещенск, 23 мая 2019 года. – Благовещенск: Амурский государственный университет, 2019. – С. 208–209.

4. *Видмант, О. С.* Прогнозирование финансовых временных рядов с использованием рекуррентных нейронных сетей LSTM / О. С. Видмант // Общество: политика, экономика, право. – 2018. – № 5(58). – С. 63–66. – DOI 10.24158/pep.2018.5.12.

5. *Галеев, Д. Т.* Экспериментальное исследование языковых моделей «трансформер» в задаче нахождения ответа на вопрос в русскоязычном тексте / Д. Т. Галеев, В. С. Панищев // Информатика и автоматизация. – 2022. – Т. 21. – № 3. – С. 521-542. – DOI 10.15622/ia.21.3.3.

6. *Костерин, М. А.* Нейросетевая классификация русскоязычных предложений по тональности на четыре класса / М. А. Костерин, И. В. Парамонов // Моделирование и анализ информационных систем. – 2022. – Т. 29. – № 2. – С. 116–133. – DOI 10.18255/1818-1015-2022-2-116-133.

7. *Ярушкина, Н. Г.* Применение языковых моделей word2vec и BERT в задаче сентимент-анализа текстовых сообщений социальных сетей / Н. Г. Ярушкина, В. С. Мошкин, А. А. Константинов // Автоматизация процессов управления. – 2020. – № 3(61). – С. 60–69. – DOI 10.35752/1991-2927-2020-3-61-60-69.

8. *Березин, С. А.* Рекомендательная система для мероприятий на основе языковой модели BERT / С. А. Березин // МНСК-2020: Материалы 58-й Международной научной студенческой конференции, Новосибирск, 10 апреля – 2020 года. – Новосибирск: Новосибирский национальный исследовательский государственный университет, 2020. – С. 154.

9. *Дале, Д. С.* Маленький и быстрый BERT для русского языка / Д. С. Дале. – Текст: электронный // Режим доступа: https://habr.com/ru/post/562064/. – (Дата обращения: 01.08.2022).

10. *Blinov, P.* Predicting Clinical Diagnosis from Patients Electronic Health Records Using BERT-Based Neural Networks / P. Blinov, M. Avetisian, V. Kokh [et al.] // Lecture Notes in Computer Science. – 2020. – Vol. 12299 LNAI. – P. 111–121. – DOI 10.1007/978-3-030-59137-3_11.

11. *Бессмертный, И. А.* Методы квантового формализма в информационном поиске и обработке текстов на естественных языках / И. А. Бессмертный, А. В. Васильев, Ю. А. Королева [и др.] // Известия высших учебных заведений. Приборостроение. – 2019. – Т. 62. – № 8. – С. 702–709. – DOI 10.17586/0021-3454-2019-62-8-702-709.

12. *Карпович, С. Н.* Корпус текстов русского языка для тестирования алгоритмов тематического моделирования / С. Н. Карпович // Интеллектуальные технологии на транспорте. – 2018. – Т. 1. – №. 13. – С. 11–19.

13. *Cui X.* Performance Evaluation of the NVIDIA Tesla P100: Our Directive-Based Partitioning and Pipelining vs. NVIDIA's Unified Memory // Matrix. – Т. 40. – P. 50.

14. RuBERT for Sentiment Analysis. – Текст: электронный // Режим доступа: https://huggingface.co/Tatyana/rubert-base-cased-sentiment-new (дата обращения: 01.08.2022).

15. *Кропанев, Н. Д.* BERT для анализа тональности длинных текстов на примере Kaggle Russian News Dataset / Н. Д. Кропанев, А. В. Котельникова // Общество. Наука. Инновации (НПК-2021). – 2021. – С. 256–259.

16. *Shazeer N., Stern M.* Adafactor: Adaptive learning rates with sublinear memory cost // International Conference on Machine Learning. – PMLR, 2018. – P. 4596–4604.

17. *Косых, Н. Е.* Оценка гиперпараметров при анализе тональности русскоязычного

корпуса текстов / Н. Е. Косых // Интеллектуальные технологии на транспорте. – 2020. – № 3(23). – С. 41–44.

18. *Большаков, М. А.* Сравнительный анализ методов машинного обучения для оценки качества ИТ-услуг / М. А. Большаков, И. А. Молодкин, С. В. Пугачев // Защита информации. Инсайд. – 2020. – № 4(94). – С. 36–43.

Косых Никита Евгеньевич — аспирант кафедры «Информационные и вычислительные системы». Петербургский государственный университет путей сообщения Императора Александра I. E-mail: nikitosagi@mail.ru ORCID iD: https://orcid.org/0000-0002-3814-7097

DOI: https://doi.org/10.17308/sait/1995-5499/2022/3/139-151 ISSN 1995-5499
Received 10.08.2022
Accepted 30.09.2022

SENTIMENT ANALYSIS OF USER TEXTS BASED ON THE TUNING OF THE TRAINING PARAMETERS OF A DISTILLED MODEL OF THE BERT FAMILY

© 2022 N. E. Kosykh✉

Emperor Alexander I St. Petersburg State Transport University
9, Moskovsky Avenue, 190031 Saint Petersburg, Russian Federation

Annotation. The increasing complexity of neural network architectures and the increasing volume of processed data in machine learning raises the question of the need to apply more productive approaches that would optimize the development of text classification models to solve the tasks of sentiment analysis. The aim of this work is to train and optimize a approaches for data classification as part of the solution of sentiment analysis of the Russian-speaking text. This research proposes the application of pre-trained BERT bidirectional coding models as well as the ruBERT-tiny knowledge distillation model to perform multiclass text classification for sentiment analysis of user text. The application of the data compaction step for knowledge distillation models allows to optimize the training phase of the text classification models. A program is developed in the Python using machine learning libraries. The technical solution allows to test the pre-trained models of data classification, on the basis of which to create optimized models of classification for the analysis of the sentiment of the user texts, taking into account the specifics of the subject area.
Keywords: sentiment analysis, text sentiment classification, distillation, learning model, data preprocessing, data normalization, BERT, ruBert, Tatyana, Python.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. *Koloshina V. S. and Rodionova, M. E.* (2021) Analiz vliyaniya social'nyh setej v politicheskoj sfere obshchestva: ot mirovoj k rossijskoj praktike (na analize social'nyh setej rossijskih politicheskih partij) [Analysis of the impact of social networks in the political sphere of society: from world to russian practice (on the analysis of social networks of russian political parties)]. *Science and Education Today: Basic and Applied Research*. P. 412–420. (in Russian)

2. *Cherkasov D. and Ivanov V. V.* (2018) Mashinnoe obuchenie [Machine learning. Science]. *Technology and Education*. No. 5(46). P. 85–87. (in Russian)

3. *Malchits V. S.* (2019) Primenenie metodov mashinnogo obucheniya dlya klassifikacii novostej [Application of machine learning methods for news classification] *Youth in the 21st century: a step into the future*. P. 208–209. (in Russian)

✉ Kosykh Nikita E.
e-mail: nikitosagi@mail.ru

4. *Widmant O. S.* (2018) Prognozirovanie finansovyh vremennyh ryadov s ispol'zovaniem rekurrentnyh nejronnyh setej LSTM [Financial time series forecasting using LSTM recurrent neural networks]. *Society: politics, economics, law*. (5). P. 63-66. (in Russian).

5. *Galeev D. T. and Panischev V. S.* (2022) Eksperimental'noe issledovanie yazykovyh modelej" transformer" v zadache nahozhdeniya otveta na vopros v russkoyazychnom tekste. [Experimental study of linguistic "transformer" models in the task of finding an answer to a question in Russian-language text]. I*nformatics and Automation*. 21(3). P. 521–542. (in Russian)

6. *Kosterin, M. A. and Paramonov I. V.* (2022) Nejrosetevaya klassifikaciya russkoyazychnyh predlozhenij po tonal'nosti na chetyre klassa [Neural network classification of Russian sentences by tonality into four classes]. *Modeling and analysis of information systems.* 29(2). P. 116–133. (in Russian)

7. *Yarushkina N. G., Moshkin V. S. and Konstantinov A. A.* (2020) Primenenie yazykovyh modelej word2vec i bert v zadache sentiment-analiza tekstovyh soobshchenij social'nyh setej [Application of word2vec and bert language models in the task of sentiment analysis of social network text messages]. *Automation of management processes*. (3). P. 60–69. (in Russia)

8. *Berezin S. A.* (2020) Rekomendatel'naya sistema dlya meropriyatij na osnove yazykovoj modeli BERT [A recommendation system for events based on the BERT language model]. *INSC 2020*. P. 154–154. (in Russian)

9. *Dale D. S.* (2021) Malen'kiy i bystryy BERT dlya russkogo yazyka [Small and fast BERT for the Russian language] – Text: electronic. http://antiled66.ru/images/instrukcii/pasport_TSP02.pdf. Accessed 01 August 2022

10. *Blinov P., Avetisian, M., Kokh V., Umerenkov D. and Tuzhilin A.* (2020) Predicting clinical diagnosis from patient's electronic health records using BERT-based neural networks. *International Conference on Artificial Intelligence in Medicine*. P. 111–121.

11. *Bessmertny I. A., Vasiliev A. V., Koroleva Y. A., Platonov A. V. and Poleshchuk E. A.* (2019). Metody kvantovogo formalizma v informacionnom poiske i obrabotke tekstov na estestvennyh yazykah [Methods of quantum formalism in information retrieval and text processing in natural languages]. *Proceedings of higher education institutions. Instrumentation*. 62(8). P. 702–709. (in Russian)

12. *Karpovich S. N.* (2018) Korpus tekstov russkogo yazyka dlya testirovaniya algoritmov tematicheskogo modelirovaniya [A corpus of Russian texts for testing thematic modeling algorithms]. *Intelligent Technologies in Transport*. 1(13). P. 11–19. (in Russia)

13. *Cui, X., Scogland T. R., de Supinski B. R. and Feng W. C.* Performance Evaluation of the NVIDIA Tesla P100: Our Directive-Based Partitioning and Pipelining vs. *NVIDIA's Unified Memory. Matrix*. 40. P. 50.

14. RuBERT for Sentiment Analysis (2021) – Text: electronic. https://huggingface.co/Tatyana/rubert-base-cased-sentiment-new. Accessed 01 August 2022

15. *Kropanev N. D. and Kotelnikova A. B.* (2021) BERT dlya analiza tonal'nosti dlinnyh tekstov na primere Kaggle Russian News Dataset [BERT for analyzing the tonality of long texts on the example of Kaggle Russian News Dataset]. Society. Science. Innovations (NPC-2021). P. 256–259.

16. *Shazeer N. and Stern M.* (2018) Adafactor: Adaptive learning rates with sublinear memory cost. *In International Conference on Machine Learning*. P. 4596–4604.

17. *Kosykh N. E.* (2020) Ocenka giperparametrov pri analize tonal'nosti russkoyazychnogo korpusa tekstov [Hyperparameter estimation in tone analysis of Russian-language text corpus]. *Intelligent Technologies in Transport*. 3 (23). P. 41–44. (in Russain)

18. *Bolshakov M. A., Molodkin I. A. and Pugachev S. B.* (2020) Sravnitel'nyj analiz metodov mashinnogo obucheniya dlya ocenki kachestva IT-uslug [Comparative analysis of machine learning methods for IT service quality assessment]. *Information protection. Insider.* (4). P. 36–43.

Nikita Evgenievich Kosykh — post-graduate student of the Department of In-formation and Computing Systems. Petersburg State University of Communications Emperor Alexander I.
E-mail: nikitosagi@mail.ru ORCID iD: https://orcid.org/0000-0002-3814-7097