УДК 004.912, 004.89
DOI: 10.46548/21vek-2021-1054-0005

ОСНОВНЫЕ МЕТОДЫ АВТОМАТИЧЕСКОЙ ОБРАБОТКИ И МОДЕРАЦИИ ТЕКСТОВЫХ ДАННЫХ В СОЦИАЛЬНЫХ СЕТЯХ

Бутаев Михаил Матвеевич, доктор технических наук, профессор,
ученый секретарь научно-технического совета
ОАО «Научно-производственное предприятие «Рубин»
(440000, Россия, г. Пенза, ул. Байдукова, д. 2, e-mail: butaevmm@gmail.com)
Мартышкин Алексей Иванович, кандидат технических наук, доцент,
доцент кафедры «Вычислительные машины и системы»
Пензенский государственный технологический университет
(440039, Россия, Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11, e-mail: alexey314@yandex.ru)

Аннотация. В статье приводится обзор основных методов автоматической обработки и модерации текстовых данных в социальных сетях. Проводится исследование основных методов автоматической обработки и модерации текстовых данных в социальных сетях. Рассматриваются вопросы, связанные с возможностями искусственного интеллекта как технологии для решения задачи модерации текстового наполнения социальных сетей. Алгоритм, полученный с применением *TF-IDF* меры, определяет важность слов в текстовом сообщении и успешно борется с содержащими нецензурные слова и выражения блоками, но не всегда учитывает смысл. Данный подход пригоден для онлайн-обработки, по причине нетребовательности к вычислительным ресурсам. Алгоритм, построенный с применением этого метода, работает быстро и достигает приемлемой эффективности. Установлено, что корректно классифицировать сообщения оптимальнее всего получается у нейросетевых алгоритмов. Однако для этого требуется мощные вычислительные ресурсы и при этом затрачивается большое количество времени на обработку. По этой причине такие алгоритмы более подходят для модерации текстового наполнения социальной сети в офлайн-режиме. Сделаны выводы по проведенному исследованию.

Ключевые слова: токсичное сообщение, аугментация данных, онлайн режим, офлайн режим, *TF-IDF*, *Word2vec*, *ELMo.*

BASIC METHODS OF AUTOMATIC PROCESSING AND MODERATION OF TEXT DATA IN SOCIAL NETWORKS

Butaev Mikhail Matveyevich, doctor of technical sciences, professor,
scientific secretary of the scientific and technical council
JSC "Scientific production acceptance" Rubin "
(440000, Russia, Penza, Baidukova st., 2, e-mail: butaevmm@gmail.com)
Martyshkin Alexey Ivanovich, candidate of technical sciences, docent,
associate Professor of sub-department «Computers and systems»
Penza state technological University
(440039, Russia, Penza, BaydukovProyezd / Gagarin Street, 1a/11, e-mail: alexey314@yandex.ru)

Abstract. The article provides an overview of the main methods of automatic processing and moderation of text data in social networks. The main methods of automatic processing and moderation of text data in social networks are being investigated. The issues related to the possibilities of artificial intelligence as a technology for solving the problem of moderation of the text content of social networks are considered. The algorithm obtained using the TF-IDF measure determines the importance of words in a text message and successfully combats blocks containing obscene words and expressions, but does not always take into account the meaning. This approach is suitable for online processing, due to the low demand for computing resources. The algorithm built using this method works quickly and achieves acceptable efficiency. It is established that the best way to correctly classify messages is to use neural network algorithms. However, this requires powerful computing resources and requires a large amount of processing time. For this reason, such algorithms are more suitable for moderation of the text content of a social network in offline mode. In conclusion, the conclusions on the work done.

Keywords: toxic message, data augmentation, online mode, offline mode, TF-IDF, Word2vec, ELMo.

Введение. В XXI веке – веке информационных технологий социальные сети (СС) дают возможность пользователям общаться и взаимодействовать *online* даже если они территориально удалены друг от друга [1]. Однако, в последнее время пользователями СС все более осознается риск потенциального ущерба, который может быть нанесен вредоносными материалами, опубликованными в Интернет [2]. Все это позволяет пользователям СС размещать потенциально опасную информацию, которую, например, нежелательно воспринимать несовершеннолетним или психологически чувствительным людям. Такое наполнение ресурсов СС может иметь в своем составе в том числе и нецензурную лексику (мат, бранные слова и т.п.), а кроме

этого, велик риск пропаганды суицида, терроризма и т.п.

В статье рассматривается потенциал искусственного интеллекта (ИИ) для решения задачи модерации текстовой наполняемости СС. Актуальность проводимого исследования в том, что сегодня еще нет эффективных и бесплатных приложений, по своему функционалу автоматически способных отслеживать и проводить модерацию текстовых данных в рассчитанной на десятки и сотни миллионов пользователей, формирующих свои запросы-транзакции к СС.

Цель исследования – провести сравнительный анализ основных методов автоматической обработки и модерации текстовых данных в СС. Среди решаемых задач можно отметить: выделение токсичных сообщений и подготовка обучающей выборки; анализ современных алгоритмов обработки текстовых данных в онлайн и офлайн режимах. Объектом исследования выступает текстовое содержимое сообщений пользователей СС. В работе необходимо достичь поставленной цели в двух режимах: онлайн, когда текстовое содержимое необходимо обработать максимально быстро, чтобы пользователь не заметил нарушений в структуре работы логики приложения, и офлайн, когда временных ограничений на время обработки нет, поэтому нужно стремиться реализовать модель с максимальным качеством модерации данных.

Материалы и результаты исследования. Задача модерации текстовой начинки СС – классификация, при которой нужно правильно отнести сообщение к токсичному или нетоксичному классу. Допустим, у нас в тексте сообщения в СС встречаются сообщения, разделенные на два класса: токсичные (Определенно, этот человек выглядит мерзко (I). Белые люди всегда будут командовать черными (II)) и нетоксичные (Победителей не судят. Посмотри на себя: ты весь черный, как мой автомобиль (II)). Сравнение приведенных данных показывает, что токсичные сообщения могут либо содержать нецензурную лексику в явном виде, либо могут не содержать ее, но все же оскорблять кого-то или задевать чувства людей (I). Здесь решающее значение имеет контекст, где употребляется слово. Известно, что одно и то же слово, употребленное в разных контекстах, может заключать в себе разный смысл и, в результате, по-разному классифицироваться (II). Для того, чтобы правильно классифицировать сообщения, нужно располагать набором данных (корпусом текстов), где сосредоточены и токсичные, и нетоксичные сообщения [3]. В качестве такого набора данных принято использовать открытый корпус ненормативной лексики русского языка, состоящий из 7351 слова [4].

Для модерации текстового содержимого СС используют классифицированную по определенным правилам обучающую выборку. Для решения такой задачи подойдет открытый набор токсичных комментариев на русском языке [3], к достоинству которой можно отнести большое количество документов, содержащихся в ней (более 14000). Увы, применить этот набор данных непосредственно без его коррекции не удается, т.к. часть документов может быть размечена неверно – токсичные сообщения могут помечаться как нетоксичные и наоборот. Тогда необходимо проводить ручной анализ и обработку данных. Кроме того, можно дополнительно добавить в обучающую выборку негативные комментарии из неразмеченного открытого корпуса *RuTweetCorp* [5, 6], что поможет качественнее дать классификацию обучающей выборке, однако потребует колоссальных затрат времени на ручную обработку и классификацию.

Определение словаря используемых при общении слов и терминов – одна из главных задач при составлении обучающей выборки. Встречаются слова, похожие на плохие, но только на первый взгляд (например, слово «оглобля»). Необходимо четко разделять такие слова и относить их к нетоксичным. Сегодня, несмотря на высокий уровень развития сферы *IT*, к сожалению, идеальных систем не существует, и недобросовестные пользователи могут попытаться обмануть системы модерации содержимого с тем, чтобы алгоритм принимал завуалированное токсичное сообщение за нетоксичное, а человек понимал смысл слов в контексте. Получается замкнутый круг, в котором разработчики синтезируют новые программные средства, а пользователи тестируют их, находя всевозможные уязвимости [7]. Для обхода систем модерации содержимого широкое распространение получили варианты изменения данных, включающие в себя, например, введение дополнительных пробелов, отсутствие пробелов, использование транслитерации и т.п. На самом деле всевозможных вариантов изменения данных намного больше. Для того чтобы система была устойчива к такому набору входных данных, требуется вручную генерировать изменения и добавлять их в обучающую выборку.

Модерация текстового содержимого СС в онлайн-режиме. Базовую фильтрацию сообщений можно реализовать при помощи словаря запрещенных слов, когда если слово из словаря встречается в сообщении, такое сообщение однозначно можно отнести к токсичному классу. Тогда необходимо при любом взаимодействии пользователя с системой проверять загружаемую им информацию на наличие в ней слов из заранее сформированного словаря. Эта операция предполагает обработку в синхронном режиме, основная проблема будет состоять в том, что поиск данных в исходной строке должен осуществляться быстро. Наилучшим решением в этих условиях оказывается алгоритм Ахо-Корасика [8], реализующий поиск множества подстрок из словаря в конкретно заданной строке.

Как правило, при операции поиска слов в корпусе текстов, данные разбивают на термы – слова, из которых состоит текст, а также другие элементы текста [9]. Для оценки важности конкретного слова в пределах текущего документа требуется рассчитать *TF-IDF* [10]. Благодаря применению *IDF*-характеристики уменьшается вес слов, распространенных и часто

встречающихся в исследуемом тексте.

$$TF_{(t)} = \frac{m_t}{\sum_k m_k}, \quad (1)$$

где m_t – количество попаданий слова t в сообщение, $\sum_k m_k$ – суммарное количество слов в сообщении.

$$IDF_{(t)} = \frac{|D|}{|\{d_i \in D, t \in d_i\}|}, \quad (2)$$

где: $|D|$ – количество документов в имеющейся коллекции, $|\{d_i \in D, t \in d_i\}|$ – количество документов из имеющейся коллекции $|D|$, в которую входит слово t.

Для каждого встречающегося в сообщении слова, определив *TF-IDF* меру, получаем векторное представление сообщения. Итоговое произведение *TF×IDF* вычисляется так, что самый большой вес будет задан для слов, специфичных для конкретного документа.

К достоинствам этого метода отнесем высокую скорость вычисления векторов для сообщений, к недостаткам – большую размерность выходного вектора, что возникает по причине анализа сообщения, которое может иметь большой объем.

Модерация текстового содержимого СС в офлайн-режиме. Если нужно понимать токсичность сообщения по его смыслу, то выше рассмотренных алгоритмов недостаточно. Сегодня известны разнообразные технические решения, базой которых являются нейронные сети. Всю область современных решений анализа текста можно разделить на несколько составляющих: работа с готовыми предобученными моделями эмбеддингов (работа с *Word2Vec* или *FastText*) [11]; разработка и реализация собственных нейронных сетей и вычисление на их основе векторов слов с дальнейшим выбором множества признаков для обучения; использование новых предобученных моделей (эмбеддинги из языковых моделей (*ELMo*) [12]. Рассмотрим основные алгоритмы подробнее.

Можно сказать, что алгоритм *Word2Vec*, впервые предложенный в 2013 году Т. Миколовым [13], – одно из самых существенных достижений последнего времени в области обработки естественных языков. Цель названного алгоритма – построение векторных представлений слов, не только включающих набор определенных цифр в заданном порядке, но и несущих конкретный смысл для решения задач в области анализа и обработки текста. Отметим, что каждый словесный вектор может иметь множество измерений, поэтому каждому уникальному слову в корпусе текстов присваивается вектор в пространстве. Достоинством этого алгоритма является возможность определять токсичность сообщения по его смыслу. Однако готовая модель обучается на ограниченном наборе слов, поэтому для слов, не входящих изначально в корпус текстов, не будут вычислены векторы (эмбеддинги), что негативно отразится на работе с данными. Для модели с максимальным качеством обработки и модерации, важно получить максимально большую обучающую выборку, что порождает другую проблему – недостатка вычислительных мощностей и времени обработки такого большого объема данных на компьютере.

Известно две основные реализации алгоритма *Word2Vec: Continuous Bag-Of-Words* (*CBOW*), пытающийся предсказать слово, исходя из соседних слов; *Continuous Skip-gram* (*SG*), пытающийся предсказать контекст, где употребляется целевое слово. Здесь основная гипотеза в том, что контекст, определяющий целевое слово, находится на ближайшем расстоянии и в ближайших словах. Ввиду этого, проводя анализ соседних слов в сообщении, можно понять, о каком целевом слове идет речь [14]. Рисунок 1 демонстрирует варианты архитектур нейронных сетей, поддерживаемые в алгоритме *Word2Vec* [15].

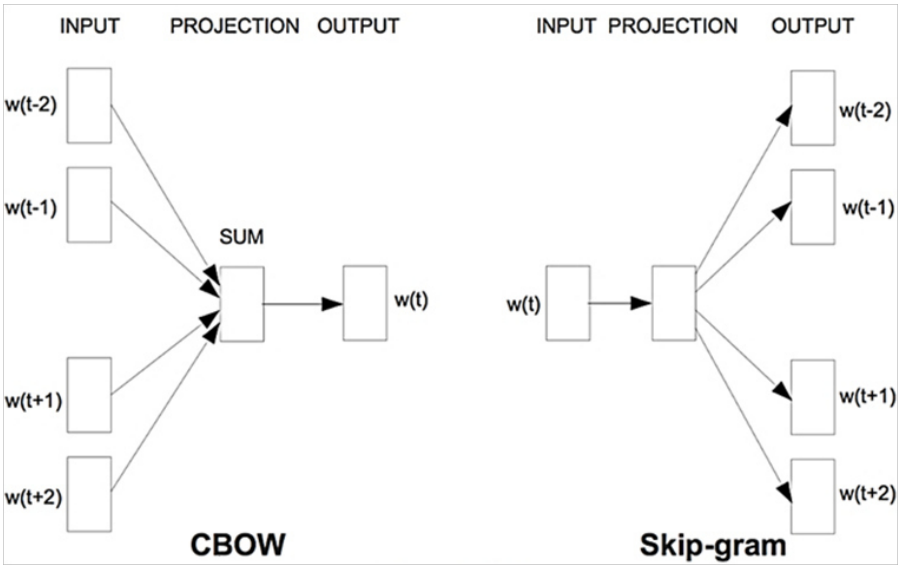

Рисунок 1 – Архитектуры нейронных сетей, поддерживаемые в алгоритме Word2Vec

Архитектура *Skip-gram* неплохо работает на обучающей выборке малого объема и позволяет качественно характеризовать редкие слова и фразы. В то же время архитектура *CBOW* позволяет в разы быстрее обучить и повысить точность классификации для самых часто встречающихся слов в корпусе. Архитектура *Skip-*

gram обучается предсказывать контекст по заданному целевому слову, то, если два слова (соответственно, редко и часто встречающиеся) помещаются рядом, оба будут иметь одинаковое отношение, по той причине, что каждое слово рассматривается двояко: и как целевое, и как контекстное [16]. В архитектуре *CBOW* редко встречающееся слово будет составной частью используемого для предсказания целевого слова контекста. Учитывая последовательность обучающих слов $w_1, \ldots w_t$, цель *Skip-gram* – максимизация средней логарифмической вероятности [17].

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c,\ j\ne 0}\log p(w_{t+j}\mid w_t), \qquad (3)$$

где: c – размер обучающей выборки.

Большая обучающая выборка приводит к увеличению количества обучающих примеров и, как следствие, – к повышению точности классификации [18].

$$p(w_0\mid w_I)=\frac{\exp(\grave{\vartheta}w_0^{T}\vartheta w_I)}{\sum_{w=1}^{W}\exp(\grave{\vartheta}w^{T}\vartheta w_I)}, \qquad (4)$$

где: $\grave{\vartheta}w$ и ϑw_I – входное и выходное векторное представление w, W – количество слов в корпусе текстов.

ELMo. Алгоритм вычисления эмбеддингов на основе языковых моделей, представляющий собой двунаправленную языковую модель, отличающуюся от традиционных тем, что для каждого полученного слова вычисляется вектор, который по сути своей является функцией, зависящей от всего входного предложения. В основу алгоритма заложена рекуррентная двунаправленная нейронная сеть с архитектурой *LSTM* (долгая краткосрочная память). Алгоритм предназначен для обучения на задаче языкового моделирования [19, 20].

Главное отличие алгоритма от *Word2Vec* в том, что модель не рассчитывает векторное представление для каждого слова в корпусе текстов, а кодирует входное сообщение посимвольно. Стоит сказать, что символов в языке много меньше, чем слов, поэтому каждому символу присваивается свой вектор представления, после чего рассчитанные векторные представления подаются на вход нейронной сети, где проходят через два слоя – агрегирующий и сверточный. На данном этапе входная последовательность описывается с помощью определенных признаков и передается на вход так называемой «скоростной сети». Здесь необходимо получить представления, в дальнейшем отправляемые на вход глубокой двунаправленной нейронной сети с архитектурой *LSTM* [21], которая, как правило, использует обучение без учителя. Допустим, имеется последовательность предыдущих слов, поэтому возможна попытка предсказания следующего встречающегося слова. Размер входной последовательности возможен быть произвольным, т.к. используется рекуррентная нейронная сеть, поэтому ограничение в виде фиксированного скользящего окна контекста, свойственного традиционным моделям, здесь снимается [22].

FastText. Алгоритм *FastText* является развитием *Word2Vec* [23, 24]. Главный недостаток предобученной модели эмбеддингов *Word2Vec* состоит в том, что обработка, обучение и построение результирующих векторов выполняется на словах, описанных только в корпусе текстов. Поэтому для слов, отсутствующих в начальной обучающей выборке, векторы рассчитаны не будут, что ухудшает конечную точность. Разработчики из *Facebook* предложили разбивать слова на *n*-граммы для учета морфологических единиц исследуемого языка [22]. С семантической точки зрения *n*-грамма – последовательность звуков, слогов, слов или букв заданной длины [25]. По этой причине стало возможным рассматривать при обработке не только слова, но и их составные части – корни, приставки, суффиксы и окончания, тем самым закрывая проблему для слов, не входящих в начальный корпус текстов. Итак, когда рассматриваемое слово разбито на n-граммы и для каждого из них рассчитано векторное представление, то для получения векторного представления всего слова целиком достаточно сложить полученные вектора.

Заключение. В статье рассмотрены современные механизмы решения задачи автомодерации текстовых сообщений. Отмечены сильные и слабые стороны существующих алгоритмов и показаны ситуации, когда предпочтительнее применять тот или иной алгоритм.

При выборке и дальнейшем исключении запрещенных слов по словарю результаты зависят от выбора алгоритма поиска в заданном сообщении. Применяя такой подход возможно быстро определять некоторые токсичные блоки данных и блокировать их до передачи в основной алгоритм модерации. Но всё-таки данный метод может модерировать только слова, не вникая в смысл сообщений. Для первичной обработки текста в онлайн-режиме данный подход подойдет вполне.

Алгоритм, полученный с применением статистической *TF-IDF* меры, качественно определяет важность слов в документе и успешно борется с нецензурными словами и выражениями, содержащимися в сообщениях. Этот алгоритм целесообразно использовать для онлайн-обработки, ввиду его небольших требований к вычислительным ресурсам и эффективной работе.

Корректно классифицировать сообщения, основываясь на их смысле, содержании и семантических связях, лучше всего получается у нейросетевых алгоритмов и предобученных моделей эмбеддингов. Однако их реализация требует значительных вычислительных мощностей и времени на обработку информации. Поэтому данные алгоритмы целесообразно использовать для модерации текстового контента в офлайн-режиме.

Что касается существующих средств автоматической модерации текстового контента, то такие, обычно платные средства, либо имеют ограничения по количеству транзакций, обрабатываемых в единицу времени, либо не способны поддерживать обработку русского языка [26].

СПИСОК ЛИТЕРАТУРЫ:

1. Социальные сети и виртуальные сетевые сообщества / отв. ред. Верченов Л. Н., Ефременко Д. В., Тищенко В. И. – М.: ИНИОН РАН, 2013. – 360 с.

2. Мкртчян, Л.М. Риски и угрозы социальной безопасности личности в сетевом коммуникативном пространстве: к постановке проблемы // Приоритетные научные направления: от теории к практике. – 2013. – № 8. – С. 149-155.

3. Определение токсичных комментариев на русском языке [Электронный ресурс]. – URL: https://habr.com/ru/company/mailru/blog/526268/ (дата обращения: 18.04.2021).

4. Корпус ненормативной лексики русского языка для нужд NLP [Электронный ресурс]. – URL: https://github.com/odaykhovskaya/obscene_words_ru/blob/master/obscene_corpus.txt (дата обращения: 15.04.2021).

5. Рубцова Ю. В. Автоматическое построение и анализ корпуса коротких текстов (постов микроблогов) для задачи разработки и тренировки тонового классификатора // Инженерия знаний и технологии семантического веба. – 2012. – Т. 1. – С. 109-116.

6. Рубцова Ю. В. Построение корпуса текстов для настройки тонового классификатора // Программные продукты и системы. – 2015. – №1(109). – С.72-78.

7. Гленфорд Майерс, Том Баджетт, Кори Сандлер. Искусство тестирования программ, 3-е издание = The Art of Software Testing, 3rd Edition. – М.: «Диалектика», 2012. – 272 с.

8. Алгоритм Ахо-Корасик - Алгоритмика [Электронный ресурс]. – URL: https://algorithmica.org/ru/aho-corasick/ (дата обращения: 19.04.2021).

9. McCormick, C. (2017, January 11). Word2Vec Tutorial Part 2 – Negative Sampling. [Электронный ресурс]. – URL: http://www.mccormickml.com (дата обращения: 15.04.2021).

10. Jones K. S. A statistical interpretation of term specificity and its application in retrieval (англ.) // Journal of Documentation: журнал. – MCB University: MCB University Press, 2004. – Vol. 60, no. 5. – PP. 493-502.

11. Обзор четырёх популярных NLP-моделей [Электронный ресурс]. – URL: https://proglib.io/p/obzor-chetyreh-populyarnyh-nlp-modeley-2020-04-21 (дата обращения: 17.04.2021).

12. Передача обучения с использованием ELMO Embeddings [Электронный ресурс]. – URL: https://www.machinelearningmastery.ru/transfer-learning-using-elmo-embedding-c4a7e415103c/ (дата обращения: 16.04.2021).

13. Курс по теоретическому глубокому машинному обучению Deep Learning в NLP. Лекции 1–5. [Электронный ресурс]. – URL: https://github.com/deepmipt/tdl (дата обращения: 15.04.2021).

14. Church K. W. Word2Vec // Natural Language Engineering. – 2017. – Т. 23. – №. 1. – С. 155-162.

15. Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space [Электронный ресурс]. – URL: http://icybcluster.org.ua:34145/technology-documents/Efficient_Estimation_of_Word_Representations_in_Vector_Space.pdf (дата обращения: 16.04.2021).

16. Skip-Gram: алгоритм прогнозирования слов контекста НЛП [Электронный ресурс]. – URL: https://www.machinelearningmastery.ru/skip-gram-nlp-context-words-prediction-algorithm-5bbf34f84e0c/ (дата обращения: 18.04.2021).

17. Levy O., Goldberg Y. Linguistic regularities in sparse and explicit word representations // Proceedings of the eighteenth conference on computational natural language learning. – 2014. – С. 171-180.

18. Levy O., Goldberg Y. Dependency-based word embeddings // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). – 2014. – Т. 2. – С. 302-308.

19. Pennington J., Socher R., Manning C. Glove: Global vectors for word representation // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – С. 1532-1543.

20. Bojanowski P. et al. Enriching word vectors with subword information // Transactions of the Association for Computational Linguistics. – 2017. – Т. 5. – С. 135-146.

21. Двунаправленные (bidirectional) рекуррентные нейронные сети [Электронный ресурс]. – URL: https://proproprogs.ru/neural_network/bidirectional-rekurrentnye-neyronnye-seti (дата обращения: 16.04.2021).

22. Short text classification in twitter to improve information filtering / B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, M. Demirbas // Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM. 2010. Geneva, Switzerland. P. 841–842.

23. Как работает FastText и где ее применять [Электронный ресурс]. – URL: https://sysblok.ru/nlp/kak-rabotaet-fasttext-i-gde-ee-primenjat/ (дата обращения: 19.04.2021).

24. Анализ настроений FastText для твитов: простое руководство. [Электронный ресурс]. – URL: https://www.machinelearningmastery.ru/fasttext-sentiment-analysis-for-tweets-a-straightforward-guide-9a8c070449a2/ (дата обращения: 19.04.2021).

25. N-грамма кратко [Электронный ресурс]. – URL: https://intellect.icu/n-gramma-9505 (дата обращения: 16.04.2021).

26. Построение автоматической системы модерации сообщений [Электронный ресурс]. – URL: https://habr.com/ru/post/454628/ (дата обращения: 18.04.2021).

Статья поступила в редакцию 13.05.2021
Статья принята к публикации 16.06.2021