Научная статья
УДК 004.891.2
DOI: 10.17072/1993-0550-2024-1-60-71

Создание алгоритма для предсказания наличия недостоверной информации в социальных сетях на русском языке

Александр Андреевич Черняев[1], Александр Григорьевич Ивашко[2]
[1,2]Тюменский государственный университет, Тюмень, Россия
[1]a.a.chernyaev@utmn.ru;
[2]a.g.ivashko@utmn.ru

Аннотация. Развитие способов передачи информации от пользователя к пользователю, таких как социальные сети, привело к тому, что количество недостоверной информации достигает рекордных показателей. Данная проблема касается не только обычных пользователей социальных сетей, но и средств массовой информации, которые в качестве источника информации могут обращаться к подобным сообщениям. Распространение ложной информации приводит как к проблемам финансовым, так и к опасности жизнедеятельности человека. Отследить данные сообщения вручную уже почти не представляется возможным, и в связи с этим требуется создать алгоритм, который способен выполнять этот процесс автоматически. Целью данной работы является попытка создать подобный алгоритм для русского языка методами машинного обучения. В качестве данных, на которых основаны модели, взята выборка данных, которая прошла процесс ручной аннотации. Выборка прошла процесс подготовки и балансировки. Из этой выборки были получены 29 атрибутов, которые можно разделить на 3 категории: пользователя, текста и распространения. Эти атрибуты и были применены для получения классифицирующих моделей, которые способны предсказывать с достаточно большой вероятностью. Результатом данной работы стал алгоритм для предсказания наличия недостоверной информации в сообщении социальной сети.

Ключевые слова: *машинное обучение; нейронные сети; анализ данных; лингвистический анализ; семантический анализ; социальные сети*

Для цитирования: *Черняев А. А., Ивашко А. Г.* Создание алгоритма для предсказания наличия недостоверной информации в социальных сетях на русском языке // Вестник Пермского университета. Математика. Механика. Информатика. 2024. Вып. 1(64). С. 60–71. DOI: 10.17072/1993-0550-2024-1-60-71.

Статья поступила в редакцию 08.12.2023; одобрена после рецензирования 12.02.2024; принята к публикации 18.03.2024.

Research article

An Algorithm Creating for Predicting the Inaccurate Information Presence in Social Networks in Russian Language

Alexander A. Chernyaev[1], Alexander G. Ivashko[2]
[1,2]Tyumen State University, Tyumen, Russia
[1]a.a.chernyaev@utmn.ru;
[2]a.g.ivashko@utmn.ru

Abstract. The development of user-to-user communication methods, such as social media, has resulted in the amount of inaccurate information reaching record levels. This problem affects not only regular users of social media, but also the media, which may refer to such messages as a source of information. The spread of false information leads to both financial and life-threatening problems. It is almost impossible to trace these messages manually and therefore it is required to create an algorithm that can perform this process

automatically. The purpose of this paper is to try to create such an algorithm for the Russian language using machine learning methods. The data on which the models are based is a sample of data that has undergone the process of manual annotation. The sample has undergone the process of preparation and balancing. From this sample, 29 attributes were obtained which can be divided into 3 categories: user, text and distribution. These attributes and were applied to obtain classification models that are able to predict with sufficiently high probability. The result of this work is an algorithm for predicting the presence of inaccurate information in a social network post.

Keywords: *machine learning; neural networks; data analysis; linguistic analysis; semantic analysis; social networks*

For citation: Chernyaev, A.A., Ivashko, A.G. (2024), "An Algorithm Creating for Predicting the Inaccurate Information Presence in Social Networks in Russian Language", *Bulletin of Perm University. Mathematics. Mechanics. Computer Science*, no. 1(64), pp. 60-71. (In Russ.). DOI: 10.17072/1993-0550-2024-1-60-71.

The article was submitted 08.12.2023; approved after reviewing 12.02.2024; accepted for publication 18.03.2024.

Введение

Актуальность данной темы обусловлена сильным ростом распространения недостоверной информации в сети Интернет. В современное время любой пользователь сети Интернет может быть самостоятельным средством массовой информации, что приводит к распространению непроверенной информации в настолько больших количествах, что модерация этого потока ручными методами просто не осуществима.

Увеличение количества недостоверной информации приводит к потерям, как финансовым, так и ухудшает состояние безопасности в обществе [1].

Чтобы решить данную проблему в последние года, было создано множество методов, которые позволяют автоматически определять некоторые виды некорректной информации, например, слухи. Но проблема этих решений в том, что созданы они, в основном, только для английского и китайских языков.

Целью данной работы является построение метода автоматического определения наличия недостоверной информации с особенностями для русского языка.

1. Методы

Методы, которые были использованы в данной области, можно разделить на несколько категорий:

1) применение различных методов машинного обучения. В основном такие работы содержат изучение внешних данных сообщения, а не самого текста, например, параметры, построенные вокруг данных пользователя или данные текста сообщения;

2) синтаксический и семантический анализ текста. Методы, основанные на глубоком изучении строения текста и его смысла;

3) графы распространения недостоверной информации. Построение графов взаимодействия пользователей в социальных сетях, например, для поиска первоисточника сообщения.

Среди данных категорий наибольшего успеха добились работы, применяющие машинное обучение с использованием различных параметров.

Данная работа будет комбинировать различные методы и подавать их результаты на модель искусственного интеллекта (модель, обученная на выборке данных, способная распознавать определенные образы).

2. Источник данных

Так как в методах машинного обучения нельзя обойтись без качественных обучающих данных, то необходимо их найти и обработать.

В качестве обучающих данных были выбраны сообщения пользователей в социальной сети Twitter.

Полученные результаты могут быть развернуты и на другие социальные сети с небольшими правками. Так как социальные сети имеют разные особенности.

На момент получения данных социальная сеть Twitter была удобна многими особенностями, такими как:

1. Открытое API с возможностью получения данных как исторических, так и за недавний срок;

2. Количество символов ограниченно 480 символами, что уменьшает время на обработку синтаксиса и семантики текста;

3. Разнообразие предоставляемых данных, как о пользователе, так и о самом сообщении;

4. Простое и разнообразное взаимодействия между пользователями позволяет легко строить графы связей; и многие другие.

Но самое важное для данной работы – это обширная русскоязычная аудитория.

Через API, на момент получения данных, можно было получить 54 характеристики, которые описывали пользователя, сообщение и взаимодействие пользователей. Но не все эти характеристики являются необходимыми для построения модели.

Среди обязательных можно выделить следующие:

1. ID-пользователя – уникальный идентификатор пользователя в социальной сети.
2. Дата создания аккаунта.
3. Количество сообщений, который оставил пользователь за все время пользования с даты регистрации.
4. Количество понравившихся сообщений.
5. Количество подписчиков.
6. Количество подписок на других пользователей.
7. Метка о достоверности аккаунта.
8. ID-сообщения – уникальный идентификатор данного сообщения.
9. Текст данного сообщения.
10. Дата создания сообщения.
11. Количество отметок "Нравится".
12. ID-сообщения, на который поступил ответ.
13. ID-пользователей, которые сделали репост (копирование оригинального сообщения с ссылкой на оригинального пользователя) данного сообщения.
14. ID-пользователя на сообщение которого поступил ответ.

Сообщение в данной социальной сети может быть как оригинальным сообщением пользователя, ответом на другое сообщение, так и комментарием в дополнении к репосту. Таким образом формируется взаимодействие пользователей в социальной сети.

Сообщение может содержать ссылки, хэштеги (тег, ссылочного типа, который начинается с символа #), изображения, видео- и другие медиа-элементы. Исследование этих элементов поможет создать алгоритм для автоматического определения наличия недостоверной информации в социальных сетях, однако, на данный момент, ведется работа только с текстовыми данными.

3. Построение выборки для обучения моделей

Описанные выше характеристики необходимо получить для каждого примера, который будет добавлен в обучающую выборку.

Сами примеры были получены следующими способами:

a) поиск данных в социальной сети и сбор данных вручную;

b) поиск заранее собранных, но не размеченных данных.

В случае первого варианта (a) были собраны ID-сообщений и получены описанные выше данные через API.

В случае второго варианта (b) удалось найти большую коллекцию сообщений [2], которая содержала множество миллионов записей пользователей на тему "Короновирус". В данную выборку попали данные и от пользователей, которые писали сообщения на русском языке. Стоит отметить, что изначально данная выборка содержала только ID-сообщения, дату и регион, в котором было оставлено сообщения. При помощи этих параметров удалось отсортировать по RU региону и применяя ID-сообщения собрать остальную информацию через API. Всего в итоговую выборку, после фильтрации данных и ручной разметки, попало 10150 записей, которые содержат набор вышеописанных характеристик и поле, определяемое следующим образом:

1 – если сообщение несет в себе некорректную или ложную информацию;

0 – иначе.

Например, сообщение

"*Солист Rammstein Тилль Линдеманн попал в больницу в Берлине с COVID-19, сообщают немецкие СМИ*" на момент написания, было ложным, что было опровергнуто самой группой. Такому сообщению была поставлена метка 1.

А сообщение

"*#coronavirus #коронавирус В Москве выявлен 6-й человек с диагнозом коронавирус*", было написано официальными СМИ и такому сообщению была поставлена метка 0.

Процесс разметки включал чтение текста сообщения, поиск информации и оценка.

4. Реализация алгоритма

Для реализации алгоритма был выбран язык Python, так как данный язык наиболее приспособлен для работы с моделями машинного обучения. Собранная выборка помещается в реляционно-графовую базу данных EdgeDB, где каждая запись воспринимается в качестве объекта, что упрощает и ускоряет работу в Python относительно стандартных реляционных баз данных.

Сам алгоритм состоит из двух частей (рис. 1):

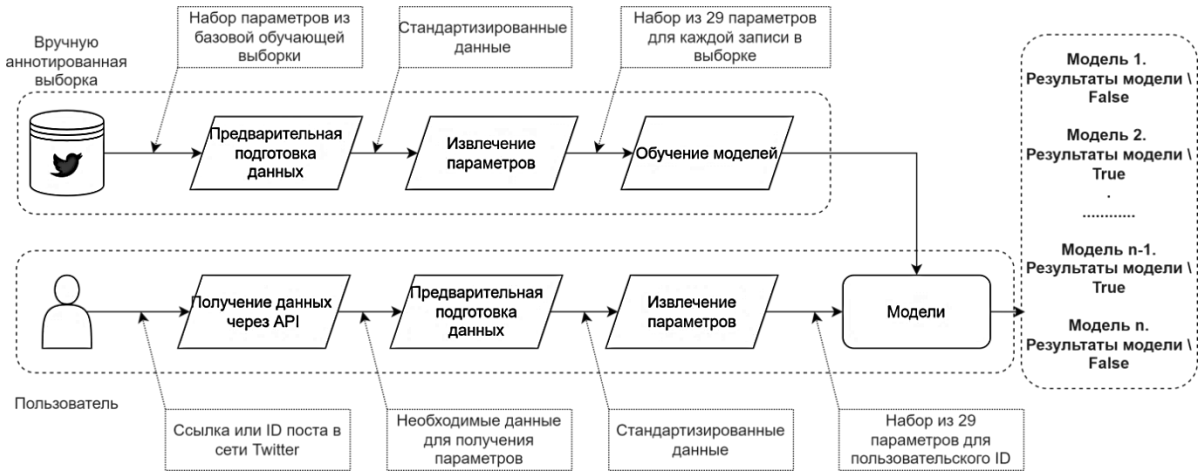

Рис. 1. *Диаграмма основных частей из которых состоит алгоритм*

1. Подготовка моделей, включающая создание и обучение моделей на собранных данных через манипулирование параметрами моделей. На вход поступают вручную аннотированные данные, а на выходе – модели, обученные на атрибутах, полученных из данных;

2. Работа с данными, которые подаются на модели, пользователем. Это либо ссылка на сообщение, либо ID-сообщения, которые пользователь предоставляет на вход. На выходе пользователь получает предсказание от нескольких моделей и их предсказания в процентах.

Предварительная подготовка данных включает очистку данных, приведение текста и даты к единому формату, очистку от специальных символов и другие. Обновленные данные хранятся отдельно от оригинальных записей, так как некоторые атрибуты применяют оригинальный текст, а другие – обновленный.

Для получения атрибутов моделей применяются различные методы манипулирования данными, таких как: построение деревьев, преобразования данных в табличные, регулярные выражения. Все это сделано путем разработки дополнительных функций.

Далее рассмотрим сами атрибуты, которые необходимо получать из данных.

5. Получение атрибутов модели

Атрибуты можно разделить на три категории. Часть этих атрибутов свойственна для всех языков, а оставшаяся часть является подобранной для русского языка.

Рассмотрим данные атрибуты:

Первая категория – *атрибуты пользователя*. При помощи данных атрибутов происходит изучение поведения пользователя и его действия в социальной сети [3], [4].

Этими атрибутами являются:

1. *Количество дней между регистрацией и сообщением*. Этот атрибут указывает, насколько стар данный аккаунт. Если аккаунт достаточно новый, то можно сделать вывод, что текст сообщения был сгенерирован и отправлен ботом.

2. *Вовлечение* – показатель того, насколько пользователь погружен в систему социальной сети (4.1):

$$ENG = \frac{\sum(R + L + F + FS)}{Y}, \qquad (4.1)$$

где R – ответы, L – количество лайков, F – количество подписчиков, FS – количество подписок, Y – количество лет со дня создания аккаунта с округлением в большую сторону.

3. *Влияние* – количество пользователей, на которое пользователь может распространять свои сообщения (4.2):

$$\inf = F, \qquad (4.2)$$

где F – количество подписчиков.

4. *Оригинальность* – показывает, насколько данный пользователь оригинален. Если доля оригинальных сообщений больше или равна 50 %, то можно сказать, что пользователь достаточно оригинален, иначе не оригинален (4.3):

$$ORG = \frac{T}{T+R}, \qquad ((4.3)$$

где T – количество сообщений, R – количество репостов.

5. *Роль* – атрибут, который определяет роль пользователя в социальной сети (4.4). Если количество подписчиков больше, чем подписок, то данный пользователь получает роли – распространитель, иначе получатель:

$$RL = \frac{F}{FS}, \qquad (4.4)$$

где F – количество подписчиков, FS – количество подписок.

6. *Доверие* – для этого используется специальная отметка "Verified", которую можно получить через API. Она указывает на то, что пользователь проверен системой социальной сети и является официальным аккаунтом.

Вторая категория – это *категория распространения и связи сообщений в сети*. Для построения атрибутов данной категории необходимо построить дерево связей [5], [6]. Построение данного дерева включает сбор данных о пользователях, которые каким-то образом реагировали на рассматриваемое сообщение (диффузия сообщений), но изначально API возвращает значения, в которых все связи идут от оригинального сообщения. То есть, если пользователь сделал репост репоста, то ссылка будет ввести к оригинальному сообщению в обоих случаях.

Чтобы это исправить, необходимо получить список всех репостов, подписчиков и подписок для каждого пользователя, которые взаимодействовали с оригинальным сообщением, и время публикации каждого сообщения.

При помощи манипулирования этой информацией можно построить дерево взаимодействия.

Из этого дерева можно получить следующие атрибуты:

7. *Наиболее связанный компонент (НСК).* Отображает глубину полученного дерева, т. е. НСК является наибольший путь от корня дерева к листьям (4.5):

$$LCP = \frac{LCC}{AC}, \qquad (4.5)$$

где LCC – количество узлов в НСК, AC – общее количество узлов в дереве.

8. *Доля изолированных сообщений*. Не каждый узел в дереве мог получить какую-то реакцию (репост, ответ, лайк). В данном атрибуте необходимо найти все эти узлы и определить их долю относительно всех узлов в дереве (4.6):

$$IP = \frac{I}{AC}, \qquad (4.6)$$

где I – количество изолированных узлов в дереве, AC – общее количество узлов в дереве.

9. *Низкая – высокая диффузия*. Во время выполнения репоста может возникнуть ситуация, когда пользователь у которого больше подписчиков, сделал репост сообщения пользователя, у которого меньше подписчиков. Данная ситуация называется низкая–высокая диффузия. Необходимо определить долю таких ситуаций относительно всего дерева (4.7):

$$LHDP = \frac{LHD}{AC-1}, \qquad (4.7)$$

где LHD – количество ситуаций с низкой – высокой диффузией в дереве, AC – общее количество узлов в дереве.

10. *Доля сообщений со ссылками*. Для данного атрибута необходимо найти долю узлов, которые содержат ссылки в сообщении. Определить наличие ссылки можно при помощи API или регулярного выражения. Количество таких узлов делится на общее количество узлов в дереве (4.8):

$$URLP = \frac{URL}{AC}, \qquad (4.8)$$

где URL – количество узлов со ссылками в дереве, AC – общее количество узлов в дереве.

Последняя категория – это *атрибуты текста сообщения*. Данная категория содержит уникальные для русского языка атрибуты. Для большинства атрибутов этой категории были собраны коллекции слов, которые применялись в регулярных выражениях.

11. *Слова, выражающие мнение*. Это набор таких слов и словосочетаний как: *считаю*, *вроде*, *кажется* и т. д. [7] (4.9):

$$MN = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \qquad (4.9)$$

12. *Вульгарные выражения*. Под вульгарными словами понимает, как мат, так и грубые слова [8] (4.10):

$$V = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.10)$$

13. *Слова, выражающие уверенность*. Данный атрибут указывает на то, что пользователь уверен [9] в том, что он пишет. Например, *знаю*, *уверен*, *точно* и т. д. (4.11):

$$S = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.11)$$

14. *Слова, выражающие условность.* Для определения наличия условности [10] в сообщении пользователя применяются подобные слова: *если*, *когда бы*, *как* и т. д. (4.12):

$$IF = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.12)$$

15. *Местоимения*. Параметр определяет наличие местоимений. Это слова: *я, ты, твой* и т. д. (4.13):

$$PR = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.13)$$

16. *Количество чисел в тексте*. В тексте сообщения могут присутствовать числа. Для данного атрибута мы проверяем долю чисел относительно всех слов в тексте (4.14):

$$NC = \frac{C}{N}, \quad (4.14)$$

где *C* – количество чисел в тексте, *N* – количество слов в тексте.

17. *Слова, описывающие величину*. Пользователь в сообщении может сделать предположение о каком-то количестве, которое можно описать такими словами: *несколько*, *минимум*, *количество* и т. д [11] (4.15):

$$CN = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.15)$$

18. *Слова, описывающие относительное время*. Пользователь в тексте может применять такие слова, как *только что*, *с тех пор как*, *вчера* и т. д. [12], которые указывают на относительность времени (4.16):

$$TM = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.16)$$

19. *Слова, описывающие чувства*. Атрибут определяет наличие слов, указывающих на одно из 5 чувств. Это слова: увидел, почувствовал, услышал и т. д. (4.17):

$$FL = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.17)$$

20. *Перечисления*. В данном атрибуте необходимо определить наличия перечисления. Это можно сделать как при помощи слов: *во-первых, это раз, во-вторых,* так и при помощи цифр *1,2,3 (4.18):*

$$CNT = \begin{cases} 0, \textit{слова отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.18)$$

21. *Правописание*. Чтобы определить наличие ошибок в тексте используются библиотеки, которые используются в OpenOffice или LibreOffice для проверки правописания. Применить эти библиотеки можно при помощи python расширения enchant [13] (4.19):

$$P = \frac{M}{N}, \quad (4.19)$$

где *M* – количество слов с ошибкой, *N* – количество слов в тексте.

22. *Настроение*. Параметр отвечает за получение семантического разбора текста. Для получения данного параметра используется модель RuBert, переобученная на корпусе заранее размеченных данных. Модель на выходе определяет текст к одной из следующих категорий:

Neutral – нейтральное значение, т.е. текст не является ни негативным, ни позитивным.

Skip – модель не смогла определить эмоцию.

Negative – модель определила текст как имеющий негативную коннотацию.

Positive – модель определила текст как имеющий позитивную коннотацию.

23. *Смайлики*. Каждый смайлик имеет свой уникальный код и их наличие можно найти без проблем. Для поиска была подключена дополнительная библиотека, написанная на Python [14], способная выполнять поиск самостоятельно и выдавать результат о наличии или отсутствии (4.20).

$$E = \begin{cases} 0, \textit{смайлики отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.20)$$

24. *Хэштеги*. Хэштег – средство для упрощенного поиска записей в сети. Его применяют в тех случаях, когда хотят, чтобы данное сообщение было проще найти в социальной сети [15] (4.21):

$$H = \begin{cases} 0, \textit{хэштеги отсутствуют} \\ 1, \textit{иначе} \end{cases}. \quad (4.21)$$

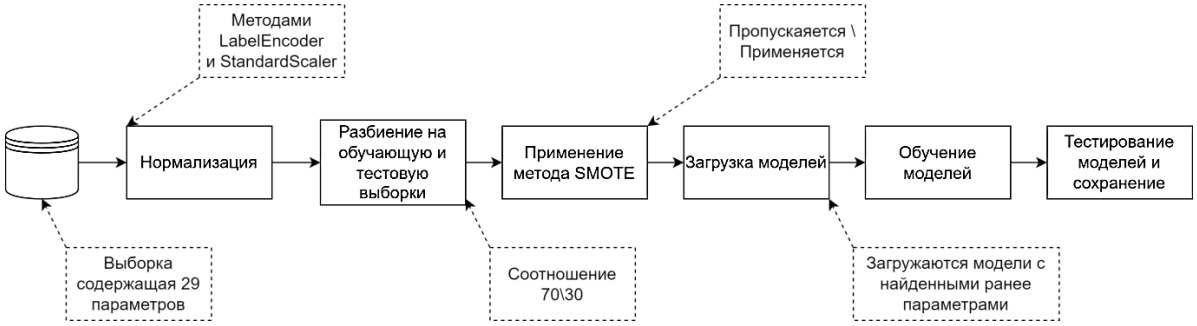

Рис. 2. *Диаграмма процесса обучения моделей*

25. *Ссылки* – сообщение может содержать в себе ссылки на дополнительные источники информации [16]. Например, пользователь во время обсуждения, какой-то темы может привести пример, найденный в интернете. Получаем данный атрибут через бинарную функцию (4.22):

$$U = \begin{cases} 0, ссылки\ отсутствуют \\ 1, иначе \end{cases}. \quad (4.22)$$

26. *Повторяющиеся слова*. Повторяющиеся слова [15] могут указывать на генерацию текста при помощи моделей (4.23):

$$RW = \begin{cases} 0, пов.слова\ отсут. \\ 1, иначе \end{cases}. \quad (4.23)$$

27. *Повторяющиеся символы*. Аналогично с повторяющимися словами повторяющиеся символы [15] могут указывать на ошибки генерации текста (4.24):

$$RC = \begin{cases} 0, пов.\ символы\ отсут. \\ 1, иначе \end{cases}. \quad (4.24)$$

28. *Восклицательные знаки* указывают на эмоциональную реакцию на какое-то сообщение или явление. Такие сообщения могут помочь с определением (4.25):

$$EP = \begin{cases} 0, вос.знаки\ отсут. \\ 1, иначе \end{cases}. \quad (4.25)$$

29. *Количество слов в тексте* [20]. Простой атрибут для оценки использования доступного количества символов в сообщении (4.26):

$$WCN = \sum WD, \quad (4.26)$$

где *WD* – слова в сообщении.

Таким образом количество атрибутов, которые применяются, необходимо получить для обучения модели равно 29.

Например, для сообщения "*Вторая неделя карантина, говорят, самая сложная. И в плане изоляции дома, и в плане новых случаев заболевания коронавирусом (подходит к концу инкубационный период у многих, кто общался с больными)*". получились следующие атрибуты: 1: 4292, 2: 12626.223, 3: 3830, 4: 0, 5: 0.53577, 6: true, 7: 0, 8: 0, 9: 0, 10: 0, 11: 1, 12: 0, 13: 0, 14: 0, 15: 0, 16: 0, 17: 0, 18: 0, 19: 0, 20: 0, 21: 0, 22: Neutral, 23: 0, 24: 0, 25: 0, 26: 0, 27: 0, 28: 0, 29: 29.

Описанные выше атрибуты необходимо получить для каждой записи в обучающей выборке данных.

Важным замечанием является то, что модель строится относительно данных атрибутов, а не самих данных, которые были получены после разметки.

6. Создание тестовых моделей

Полученные атрибуты необходимо рассчитать для каждой записи в собранной выборке данных.

В связи с тем, что собранные данные оказались плохо сбалансированными, необходимо применить один из методов балансировки. В ходе анализа доступных алгоритмов на тестовых моделях лучшего всего себя показал метод SMOTE (Synthetic Minority Oversampling) [17]. Алгоритм SMOTE позволяет сгенерировать дополнительные примеры в выборку и таким образом изначальная выборка данных, состоящая из 10150 записей, получила новые записи и стала размером в более 13 тыс. примеров.

В качестве тестирования полученных атрибутов были выполнены проверки на пяти моделях (Support Vector Classification (SVC), Multi-layer Perceptron (MLP), K – ближайших соседей (KNN), наивный байесовский классификатор (CNB) и дерево – решений (Tree)) с применением данных оригинальных и включающих сгенерированные примеры.

Оценка моделей основывается на следующих методах:

$$accuracy(a,X)=\frac{1}{k}\sum_{i=1}^{k}[a(x_i)=y_i], \qquad (5.1)$$

где $a(x_i)$ – предсказанная метка класса, y_i – ожидаемая метка класса.

Данная оценка (5.1) определяет долю верно определенных меток класса относительно всех примеров в тестовой выборке:

$$recall(a,X)=\frac{TP}{TP+FN}, \qquad (5.2)$$

$$precision(a,X)=\frac{TP}{TP+FP}, \qquad (5.3)$$

$$F_1=\frac{2\cdot precision\cdot recall}{precision+recall}, \qquad (5.4)$$

где

TP – true positive (предсказание оказалось правильным с меткой класса True),

FP – false positive (предсказание оказалось ошибочным с меткой класса True),

FN – false negative (предсказание оказалось ошибочным с меткой класса False),

TP, *FP* и *FN* определяются из матрицы ошибок.

Оценка F_1 (5.4) позволяет проанализировать: получилась ли модель переобученной, то есть показывают хорошие результаты на обученных данных, но плохие на новых данных.

Для построения данных моделей применялась python библиотека scikit-learn, где описанные выше модели уже реализованы и необходимо только определить параметры и подготовить данные, которые поступают на вход в качестве обучающих и тестирующих.

Сам процесс подготовки данных и моделей к обучению можно наблюдать на рис. 2.

Рассмотрим главные этапы данного процесса:

1. Нормализация. Так как данные для обучения содержат 29 разнообразных атрибутов, то их необходимо привести к общему виду, например, в числовой формат от -1 до 1. Для этого применяются методы LabelEncoder для преобразования атрибута *Настроение*, который содержит в себе четыре разных текстовых значения, в числовой формат. И StandartScaler, который преобразовывает числовые атрибуты в единый формат. Оба метода имеются в библиотеке scikit-learn.

2. Полученная выборка делится на две части: обучающая и тестирующая в соотношении 70/30. Процесс обучения строится на обучающей выборке, а затем проверяется при помощи тестовой выборки.

3. Для эксперимента модели будут обучаться на данных, которые получили дополнительные примеры при помощи SMOTE и без них.

После этих шагов выполняется загрузка моделей с заранее полученными параметрами и начинается процесс обучения. Всего было получено 30 моделей, основанных на стандартных методах. Представленные результаты получены для данных с дополнительно сгенерированными примерами при помощи алгоритма SMOTE, так как полученные модели, на оригинальной выборке данных, получились переобученными.

После проведения тестирования моделей получились следующие результаты [18]:

Таблица 1. *Результаты тестовых моделей*

Название модели	Accuracy	F1
SVC	0.7954	0.8841
MLP	0.778	0.8738
KNN	0.8314	0.9079
CNB	0.6837	0.8018
Tree	0.8684	0.9208

Можно сделать вывод, что полученные стандартные модели показывают неплохой результат в классификации на полученных данных.

7. Создание модели нейронной сети

Далее рассмотрим возможность создания новой модели нейронной сети. В качестве инструмента для создания модели были выбраны Tensor-Flow \ Keras.

В качестве основы, на которой будет создаваться новая нейронная сеть, была выбрана модель: простая Sequential модель. Данная модель содержит в себе стандартный набор слоев, включая вход, выход, скрытые слои, слои нормализации и слои исключения, для предотвращения переобучения.

Каждый слой имеет набор параметров, которые необходимо подобрать, и так как этот процесс может занимать продолжительное время в случае ручного подбора, то было решено применить метод Роя Частиц (Particle Swarm Optimization) [19], [20], который позволяет в короткие сроки определить самые оптимальные настройки сети.

Процесс получения оптимальных настроек состоит в поэтапной оценке множества параметров, когда в ходе каждого этапа получаются оценки, которые сравниваются друг с другом.

В сетку параметров попали следующие настройки: решающая функция для скрытых слоев, функция оптимизации, количество нейронов в каждом слое.

Итоговая модель содержит 7 слоев:

1. Входной слой;
2. Слой с функцией нормализации – Adamax;
3. Скрытый слой с следующими параметрами:
   a) число нейронов: 128;
   b) преобразование входов Kernel: 29x128;
   c) функция активации: SELU.
4. Скрытый слой с следующими параметрами:
   a) число нейронов: 128;
   b) преобразование входов Kernel: 128x128;
   c) функция активации: RELU.
5. Dropout;
6. Скрытый слой с следующими параметрами:
   a) число нейронов: 1
   b) преобразование входов Kernel: 1x128;
   c) функция активации: линейная.
7. Выходной слой.

Процесс обучения такой же, как и для стандартных моделей. Имеется обучающая выборка и тестирующая. В дополнение к этому необходимо указать количество итераций, которые применяются для обучения моделей. На каждой итерации происходит взвешивание оценки моделей, а данные подаются кусками по 256 записей.

В ходе экспериментов было обнаружено, что процесс обучения входит в стагнацию примерно на 500-й итерации.

В итоге новая модель получила следующие оценки:

Таблица 2. *Значения метрик для модели нейронной сети*

Метрика	Значение
Accuracy	0.9162
Precision	0.9340
Recall	0.8957
F1	0.91

По полученным результатам (табл. 2) можно сделать вывод, что данная модель показала себя лучше, чем стандартные модели.

В итоге все описанные модели (табл. 1 и табл. 2) применяются в разработанном приложении, то есть на выходе пользователь получает оценку не от одной модели, а сразу от нескольких, а также их результаты, что позволяет получить более качественную классификацию.

В дальнейшем имеется возможность добавления новых моделей в алгоритм для улучшения качества предсказаний.

Заключение

Таким образом, можно сделать вывод, что создание алгоритма для автоматизации поиска недостоверной информации на основе текстового анализа сообщений и анализа распространения новости в социальной сети – возможно.

Описанный алгоритм, применяющий предложенные модели, дает возможность выполнить оценку недостоверной информации с вероятностью более 90 %.

Высокий уровень вероятности оценки недостоверной информации позволяет прогнозировать востребованность предлагаемого алгоритма у пользователей различных социальных сетей.

Список источников

1. *Pennycook G.* The Psychology of Fake News. Trends in Cognitive Sciences. 2021. Vol. 25. P. 321–357. DOI: 10.1016/j.tics.2021.02.007.
2. *Banda Juan M., Tekumalla Ramya, Wang Guanyu, Yu, Jingyuan Liu, Tuo Ding, Yuning, Artemova, Katya Tutubalina, Elena & Chowell Gerardo.* A large-scale COVID-19 Twitter chatter dataset for open scientific research - an international collaboration (Version 67) [Data set]. Zenodo. DOI10.5281/zenodo.5000423.

3. *Черняев А.А.* 2019. Математическое моделирование оценки достоверности слухов в средствах массовой информации / А.А. Черняев, А.Г. Ивашко // Вестник Тюменского государственного университета. Физико-математическое моделирование. Нефть, газ, энергетика. 2019. Т. 5, № 4(20). С. 181–199. DOI 10.21684/2411-7978-2019-5-4-181-199. EDN SQYEWN.
4. *Chernyaev A. Spryiskov A. Ivashko A., Bidulya Y.* A Rumor Detection in Russian Tweets. 2020. P. 108–118. DOI: 10.1007/978-3-030-60276-5_11.
5. *Eismann K.* Diffusion and persistence of false rumors in social media networks: implications of searchability on rumor self-correction on Twitter. Journal of Business Economics. 2021. Vol. 91. P. 1299–1329. DOI: 91. 10.1007/s11573-020-01022-9.
6. *Vosoughi S.* Automatic detection and verification of rumors on Twitter. 2015. P. 1–147.
7. *Иванова Г.Ф.* О мнениях и оценках / Г.Ф. Иванова // Известия Российского государственного педагогического университета им. А.И. Герцена. 2007. Т. 8, № 41. С. 25–31. EDN JXKQIX.
8. *Емельянова О.Н.* Бранная и вульгарная лексика в толковых словарях русского языка // Вестник Красноярского государственного педагогического университета им. В.П. Астафьева. 2015. № 4(34). С. 126–130. EDN VDKKMN.
9. *Рамазанова Р.З.* Вводно-модальные слова как средство выражения уверенности в современном русском языке // Филология и культура. 2020. № 2(60). С. 77–82. DOI 10.26907/2074-0239-2020-60-2-77-82. EDN PWAYJW.
10. *Селезнёва Е.В.* Сложноподчиненное предложение с придаточным условия: содержание и объем понятия // Филология на стыке научных эпох: сб. статей памяти доктора филол. наук, проф. Анатолия Михайловича Ломова / Автономная некоммерческая организация по оказанию издательских и полиграфических услуг. Воронеж: "Наука–Юнипресс", 2020. С. 158–164. EDN HESCYX.
11. *Шульга М.В.* 2002. Количественная оценочность в газетно-публицистическом тексте // Вестник МГУЛ – Лесной вестник. 2002. № 3. URL: https://cyberleninka.ru/article/n/kolichestvennaya-otsenochnost-v-gazetno-publitsisticheskom-tekste (дата обращения: 22.02.2023).
12. *Туманова А.Б.* Категория времени в современной науке: анализ и интерпретация / А.Б. Туманова, Т.В. Павлова, Н.Ю. Зуева // Неофилология. 2019. Т. 5, № 18. С. 131–138. DOI 10.20310/2587-6953-2019-5-18-131-138. EDN EAONIK.
13. *Lachowicz D.* Библиотека для Python Enchant. URL: https://abiword.github.io/ enchant/ (дата обращения: 22.02.2023).
14. *Vicenzi A.* Библиотека для Python Emojis. URL: https://emojis.readthedocs.io/en/latest/ (дата обращения: 22.02.2023).
15. *Jahanbakhsh-Nagadeh Z., Feizi-Derakhshi MR., Ramezani M.* A model to measure the spread power of rumors. J Ambient Intell Human Comput. 2022. DOI: 10.1007/s12652-022-04034-1.
16. *Castillo C., Mendoza M., Poblete B.* Information credibility on Twitter. Proceedings of the 20th International Conference on World Wide Web. 2011. P. 675–684. 10.1145/1963405.1963500.
17. *Chawla N., Bowyer K., Hall L., Kegelmeyer P.* Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002. Vol. 16. P. 321–357, DOI: 10.1613/jair.953.
18. *Черняев А.А., Ивашко А.Г.* Математическое моделирование оценки доверия к сообщению в социальных сетях на русском языке // Прикладная информатика. 2023. Т. 18, № 4. С. 121–132. DOI: 10.37791/2687-0649-2023-18-4-121-132.
19. *Kumar A., Sangwan S.R., Nayyar A.* Rumour veracity detection on twitter using particle swarm optimized shallow classifiers. Multimed Tools Appl 78, 2019. Vol. 78. P. 24083–24101. DOI: 10.1007/s11042-019-7398-6.
20. *Kennedy J., Eberhart R.* Particle swarm optimization. Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948 Vol. 4, DOI: 10.1109/ICNN.1995.488968.

References

1. Pennycook, G. (2021), "The Psychology of Fake News", *Trends in Cognitive Sciences*, vol. 25, pp. 321-357. DOI: 10.1016/j.tics.2021. 02.007.
2. Banda Juan M., Tekumalla Ramya, Wang Guanyu Yu, Jingyuan Liu, Tuo Ding, Yuning, Artemova Katya Tutubalina, Elena & Chowell Gerardo. A large-scale COVID-19 Twitter chatter dataset for open scientific research - an

international collaboration (Version 67) [Data set]. Zenodo. DOI10.5281/zenodo.5000423.
3. Chernyaev, A.A., Ivashko, A.G. (2019), "Mathematical modeling estimates of the reliability of rumors in mass media", *Vestnik Tyumenskogo gosudarstvennogo universiteta. Fiziko-matematicheskoe modelirovanie. Neft', gaz, energetika = Tyumen State University Herald. Physical and Mathematical Modeling.Oil, Gas, Energy*, vol. 5, no. 4(20), pp. 181-199. DOI: 10.21684/2411-7978-2019-5-4-181-199.
4. Chernyaev, A., Spryiskov, A., Ivashko, A., Bidulya, Y.A. (2020), "Rumor Detection in Russian Tweets", pp. 108-118. DOI: 10.1007/978-3-030-60276-5_11.
5. Eismann, K. (2021), "Diffusion and persistence of false rumors in social media networks: implications of searchability on rumor self-correction on Twitter", *Journal of Business Economics*, vol. 91, pp. 1299-1329. DOI: 91. 10.1007/s11573-020-01022-9.
6. Vosoughi, S. (2015), Automatic detection and verification of rumors on Twitter, pp.1-147.
7. Ivanova, G.F. (2007), "About opinions and evaluations", *Izvestiya Rossiyskogo gosudarstvennogo pedagogicheskogo universiteta im. A. I. Gertsena = Herzen university journal of humanities & sciences*, no. 41, pp. 25-31. EDN JXKQIX.
8. Emel'yanova, O.N. (2015), "Abusive and vulgar vocabulary in defining dictionaries of the russian language", *Vestnik Krasnoyarskogo gosudarstvennogo pedagogicheskogo universiteta im. V.P. Astaf'eva = Bulletin of Krasnoyarsk state pedagogical university named after V. P.*, no. 4(34), pp. 126-130. EDN VDKKMN.
9. Ramazanova, R.Z. (2020), "Parenthetic words as a means of expressing certitude in the modern russian language", *Filologiya i kul'tura = Philology and Culture*, no. 2(60), pp. 77-82. DOI 10.26907/2074-0239-2020-60-2-77-82. EDN PWAYJW.
10. Selezneva, E.V. (2020), "A complex subordinate sentence with a subordinate condition: the content and scope of the concept", *Filologiya na styke nauchnykh epokh: Sbornik statey pamyati doktora filologicheskikh nauk, professora Anatoliya Mikhaylovicha Lomova = Philology at the junction of scientific epochs: Collection of articles in memory of Doctor of Philological Sciences, Professor Anatoly Mikhailovich Lomov*, pp. 158-164. EDN HESCYX.
11. Shul'ga, M.V. (2002), "Quantitative evaluation in the newspaper and journalistic text", *Vestnik MGUL – Lesnoy vestnik = Forestry bulletin*, no. 3.
12. Tumanova, A.B. (2019), "The category of time in modern science: analysis and interpretation", *Neofilologiya =Neophilology*, vol. 5, no. 18, pp. 131-138. DOI 10.20310/2587-6953-2019-5-18-131-138. EDN EAONIK.
13. Lachowicz, D. Python Library Enchant. URL: https://abiword.github.io/enchant/ (accessed: 22.02.2023).
14. Vicenzi, A. (2018). Python Library Emojis. URL: https://emojis.readthedocs.io/en/latest/ (accessed: 22.02.2023).
15. Jahanbakhsh-Nagadeh, Z., Feizi-Derakhshi, MR., Ramezani, M. (2022), "A model to measure the spread power of rumors", *J Ambient Intell Human Comput*. DOI: 10.1007/s12652-022-04034-1.
16. Castillo, C., Mendoza, M., Poblete, B. (2011), "Information credibility on Twitter", *Proceedings of the 20th International Conference on World Wide Web*, pp. 675-684. 10.1145/1963405.1963500.
17. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, P. (2002), "Smote: synthetic minority over-sampling technique", *Journal of artificial intelligence research*, vol. 16, pp. 321-357. DOI: 10.1613/jair.953.
18. Chernyaev, A., Ivashko, A. (2023), "Mathematical modeling of the assessment of credibility in a message in social networks on Russian language", *Prikladnaya informatika=Journal of Applied Informatics*, vol.18, no. 4, pp. 121-132 DOI: 10.37791/2687-0649-2023-18-4-121-132.
19. Kumar, A., Sangwan, S.R., Nayyar, A. (2019), "Rumour veracity detection on twitter using particle swarm optimized shallow classifiers", *Multimed Tools Appl 78*, vol. 78, pp. 24083-24101. DOI: 10.1007/s11042-019-7398-6.
20. Kennedy, J., Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia*, vol. 4, pp. 1942-1948. DOI: 10.1109/ICNN.1995. 488968.

Информация об авторах:

А. А. Черняев – аспирант, ассистент, инженер-исследователь кафедры программной и системной инженерии Института математики и компьютерных наук Тюменского государственного университета (625003, Россия, г. Тюмень, ул. Володарского, 6), AuthorID: 1234543;

А. Г. Ивашко – доктор технических наук, профессор кафедры, программной и системной инженерии Института математики и компьютерных наук Тюменского государственного университета (625003, Россия, г. Тюмень, ул. Володарского, 6), AuthorID: 250554.

Information about the authors:

Alexander A. Chernyaev – Postgraduate, Assistant, Engineer-Researcher, Institute of Mathematics and Computer Sciences, Tyumen State University (Volodarskogo, 6, Tyumen, Russia, 625003), AuthorID: 1234543;

Alexander G. Ivashko – Doctor of Technical Sciences (Eng.), Professor of the Institute of Mathematics and Computer Sciences, Tyumen State University, (Volodarskogo, 6, Tyumen, Russia, 625003), AuthorID: 250554.