УДК 004.912, 004.89
DOI: 10.46548/21vek-2021-1054-0017

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ПРОТОТИПА МОДУЛЯ АВТОМАТИЧЕСКОГО ОТСЛЕЖИВАНИЯ КОНТЕНТА СОЦИАЛЬНЫХ СЕТЕЙ

Мартышкин Алексей Иванович, кандидат технических наук, доцент,
доцент кафедры «Вычислительные машины и системы»
Маркин Евгений Игоревич, аспирант кафедры «Вычислительные машины и системы»
Зупарова Валентина Владимировна, магистрант кафедры «Вычислительные машины и системы»
Пензенский государственный технологический университет
(440039, Россия, Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11,
e-mails: alexey314@yandex.ru, evgeniymarkin1@gmail.com, zuparova_vv@mail.ru)

Аннотация. В статье описаны программные модули автоматического отслеживания текстового содержимого социальных сетей. Рассмотрены возможные варианты работы системы регулирования и отслеживания содержимого социальных сетей – онлайн и офлайн режимы. Приведено описание алгоритмов обработки и анализа текстовых данных, описание архитектуры разрабатываемого модуля. Обоснованы выборы классификаторов сообщений на базе расчета логистической регрессии за счет проведения тестирования некоторых из известных современных классификаторов с их последующим сравнительным анализом. Тестирование предлагаемого прототипа проводилось на основе метода перекрестной проверки. В статье отражены примеры работы созданных алгоритмов анализа текстового контента в онлайн и офлайн режимах. Разработанные алгоритмы способны классифицировать входные сообщения и определять наличие в тексте токсичных блоков. Отмечено, если пользователи пытаются видоизменить контент с целью обхода системы модерации, алгоритмы смогут отличить токсичное содержимое от нетоксичного. Проведено тестирование для определения верхнего порога сервиса регулирования, работающего в офлайн-режиме, определяющего максимально возможное количество модераций в секунду. В заключении сформулированы основные выводы по проделанной работе.

Ключевые слова: токсичное сообщение, онлайн режим, офлайн режим, социальная сеть, прототип, программный модуль, *TF-IDF, ELMo*.

RESEARCH AND DEVELOPMENT OF A PROTOTYPE MODULE FOR AUTOMATIC TRACKING OF SOCIAL MEDIA CONTENT

Martyshkin Alexey Ivanovich, candidate of technical sciences, docent,
associate professor of sub-department «Computers and systems»
Markin Evgeniy Igorevich, postgraduate of sub-department «Computers and systems»
Zuparova Valentina Vladimirovna, master's student of sub-department «Computers and systems»
Penza state technological university
(440039, Russia, Penza, Baydukov Proyezd / Gagarin Street, 1a/11,
e-mails: alexey314@yandex.ru, evgeniymarkin1@gmail.com, zuparova_vv@mail.ru)

Abstract. The article describes software modules for automatic tracking of text content in social networks. Possible variants of the system for regulating and tracking the content of social networks – online and offline modes-are considered. The article describes the algorithms for processing and analyzing text data, and describes the architecture of the module being developed. The article substantiates the choice of message classifiers based on the calculation of logistic regression by testing some of the well-known modern classifiers with their subsequent comparative analysis. Testing of the proposed prototype was carried out on the basis of the cross-validation method. The article presents examples of the work of the created algorithms for analyzing text content in online and offline modes. The developed algorithms are able to classify input messages and determine the presence of toxic blocks in the text. It is noted that if users try to modify the content in order to bypass the moderation system, the algorithms will be able to distinguish toxic content from non-toxic content. Testing was conducted to determine the upper threshold of the regulation service operating in offline mode, which determines the maximum possible number of moderations per second. In conclusion, the main conclusions on the work done are formulated.

Keywords: toxic message, online mode, offline mode, social network, prototype, software module, TF-IDF, ELMo.

Введение. Социальные сети (СС) дали определенный толчок к революции на современный взгляд и общение людей между собой [1, 2]. Сегодня СС являются чуть ли не основными источниками новостей и другой необходимой информации. Обычно любая размещаемая пользователями информация никак не проверяется на соблюдение норм и правил общения в сети Интернет. Такое содержимое (контент) может включать в себя нецензурную или бранную лексику и т.п. [3]. В статье приведены примеры работы прототипа отслеживания текстового наполнения в онлайн и офлайн режимах. Предлагаемые решения способны классифицировать входные сообщения, они могут определять наличие в тексте токсичных блоков.

Цель, преследуемая в ходе написания статьи, – исследование и разработка прототипа модуля автоматического отслеживания содержимого СС [4, 5] и сокращение временных затрат на модерацию за счет автоматического выполнения этого процесса. В статье решается задача экспериментального подтверждения эффективности предлагаемого прототипа и модуля отслеживания для высоконагруженной платформы. Входные данные включают в себя текстовые сообщения, необходимые проверить на токсичность. Отслеживание осуществляется в онлайн и офлайн режимах.

Материалы и результаты исследования. Объект исследования – текстовые сообщения, необходимые проверить на токсичность. В офлайн-режиме модерация должна иметь возможность корректировки действия модели на основе полученных в результате предыдущих запусков системы данных. Чтобы корректно предсказывать результат на контрольной выборке, модель должна напрямую зависеть от обучающей выборки, что достигается за счет ее корректного набора, который должен быть разносторонним и охватывать максимально возможные условия конкретной задачи. Прототип, работающий в офлайн-режиме, должен реализоваться на основе решений, в основе своей использующих нейронную сеть. Примем, что высоконагруженная платформа имеет микросервисную архитектуру и в качестве методов взаимодействия между микросервисами используется система удаленного вызова процедур с открытым исходным кодом (*gRPC*) [6].

Модерация в онлайн-режиме. При решении поставленной в работе задачи создан прототип программного модуля. Анализ текстовых данных в приоритете осуществляется на основе словаря запрещенных слов. В случае, если таких слов не найдено, рассчитываем *TF-IDF* меру для входного текстового сообщения и принять решение о принадлежности текста сообщения к токсичному или нетоксичному классу [7]. Прототип программного модуля реализован на языке *Python* [8]. Рисунок 1 отражает алгоритм работы прототипа в онлайн-режиме.

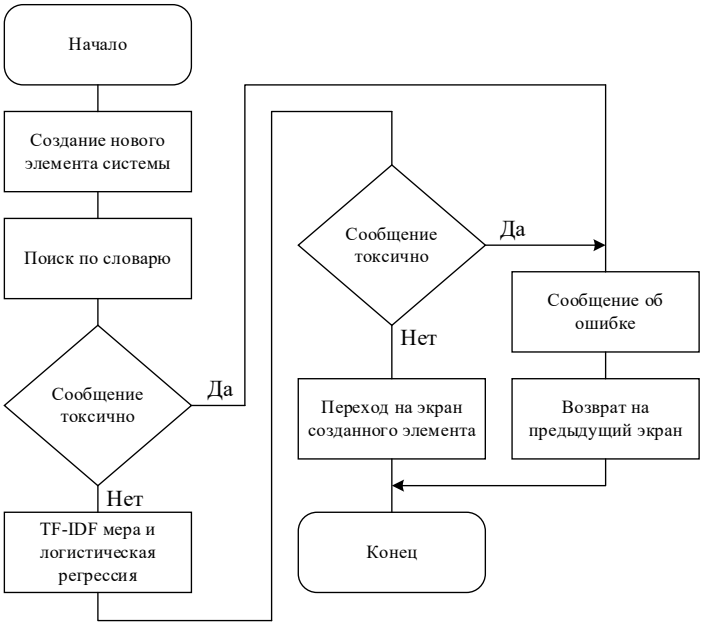

Рисунок 1 – Алгоритм работы прототипа модуля в онлайн-режиме

Рассмотрим алгоритм более подробно. Создавая элемент любого типа, пользователь попадает на экран создания элемента, где присутствуют различные текстовые поля, которые пользователь заполняет по своему усмотрению. Все инициализированные пользователем текстовые поля, в дальнейшем подлежат обработке и модерации на токсичность. Если после анализа текстового содержимого на наличие токсичных элементов, алгоритм не нашел токсичного текста, система отправляет запрос на создание элемента на сервер, ждет успешного завершения и отображает элемент пользователю. При обнаружении токсичного текста система отправляет модератору уведомление о попытке опубликования в системе токсичного сообщения.В программной реализации используется алгоритм Ахо-Корасика [9], основанный на построении конечного автомата, на вход которого подается подстрока, наличие которой необходимо проверить в строке поиска. Если автомат пришел в конечное состояние, то данная подстрока присутствует в строке поиска.

Расчет TF-IDF меры. Рассчитывая *TF-IDF* меру для отслеживания содержимого в онлайн-режиме, вместо привычных слов использовались *n*-граммы [10], что дает некоторые преимущества: появляется возможно обработки и корректной классификации слов и словосочетаний, изначально отсутствующих в обучающей выборке; возможно изменять минимальную длину символов одного *n*-грамма, регулируя точность распознавания и классификации сообщений [11, 12]. На рисунке 2 приведен пример разбиения слова «образование» на *n*-граммы с минимальной длиной одного *n*-грамма равной 3.

Аналогично можно регулировать верхний порог

длины *n*-грамм, но тогда для слов, длина которых выше этого порога, *TF-IDF* мера не рассчитается. Для корректной классификации полученных с помощью *TF-IDF* меры векторов признаков сообщений с максимальной эффективностью применен классификатор, базирующийся на расчете логистической регрессии [13]. Обоснованное решение об использовании именно данного классификатора принято после проведения тестирования на основе метода перекрестной проверки [14]. Необходимо правильно классифицировать сообщения тестовой выборки и определить процент верных ответов. Такой подход позволяет сбалансировано протестировать всю обучающую выборку. Помимо логистической регрессии тестировались также классификаторы на базе случайного леса [15, 16] и метода опорных векторов [17]. В качестве обучающей и тестовой выборки использовались собранные ранее данные.

Результаты проведенных экспериментов отражены на графиках, представленных на рисунке 3.

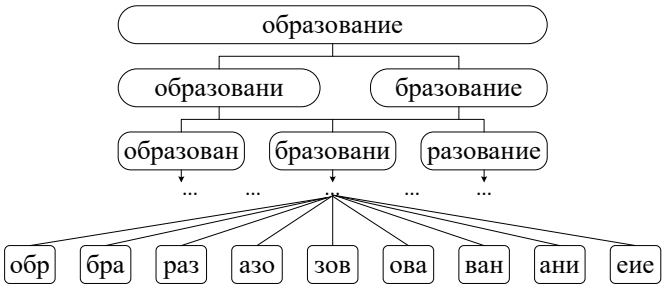

Рисунок 2 – Разбиение слова на n-граммы

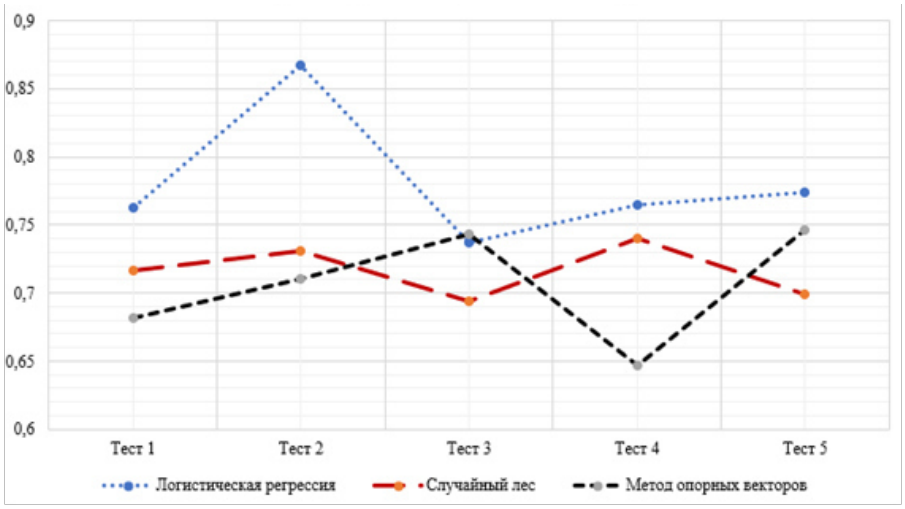

Рисунок 3 – Результаты экспериментов по определение оптимального классификатора сообщений в онлайн-режиме

Опираясь на результаты, заключим, что классификатор с принципами функционирования на основе логистической регрессии, является одним из самых эффективных способов классификации сообщений и проверки их на токсичность.

Модерация в офлайн-режиме. Работа в офлайн-режиме подразумевает, что содержимое уже опубликовано в СС, то есть запрос на сервер уже пришел, выполнился и создался новый элемент системы. Теперь в фоновом режиме следует проверить текстовое наполнение созданного элемента на содержание токсичных составляющих. Логика работы прототипа в офлайн-режиме основана на анализе и обработке текста с помощью предварительно обученных языковых моделей (*ELMo*) [18, 19]. Стоит обратить внимание на обучающую выборку. Часто нейросетевые алгоритмы показывают хорошие результаты на тестовой выборке, но при обработке реальных данных от пользователей возможны сбои [20]. Если алгоритм принял за токсичные данные, которые таковыми не являются, модератор может добавить их в обучающую выборку и переобучить алгоритм.

Для классификации сообщений, как и в случае с онлайн-режимом, выбран классификатор на основе логистической регрессии. Аналогичным образом проведены эксперименты для определения оптимального классификатора для работы алгоритма в офлайн-режиме. В качестве обучающей и тестовой выборки использовались собранные ранее данные. Результаты тестирования отражены на графиках, представленных на рисунке 4.

Отметим, что для алгоритмов, работающих в онлайн и офлайн режимах, использовались разные обучающие выборки ввиду того, что данные режимы решают разные задачи.

Алгоритм на базе *TF-IDF* меры не способен в полной мере оценивать токсичность сообщения по его смыслу и содержанию. Получено, что использование в качестве классификатора сообщений логистической регрессии дает прирост в точности определения

класса сообщения порядка 6%, в отличие от случая использования классификатора на основе метода опорных векторов. Поэтому этот алгоритм выбран в качестве основного.

В качестве примера работы прототипа модуля модерации содержимого СС в онлайн-режиме рассмотрим вариант создания элемента в системе, содержащего токсичные блоки данных. При попытке создать контакт с токсичным названием модуль модерации обнаруживает токсичный контент и блокирует данную попытку еще до отправки запроса на сервер. Пользователь получает сообщение об ошибке, а модератор системы получает уведомление о попытке публикации в СС элемента с токсичным содержимым.

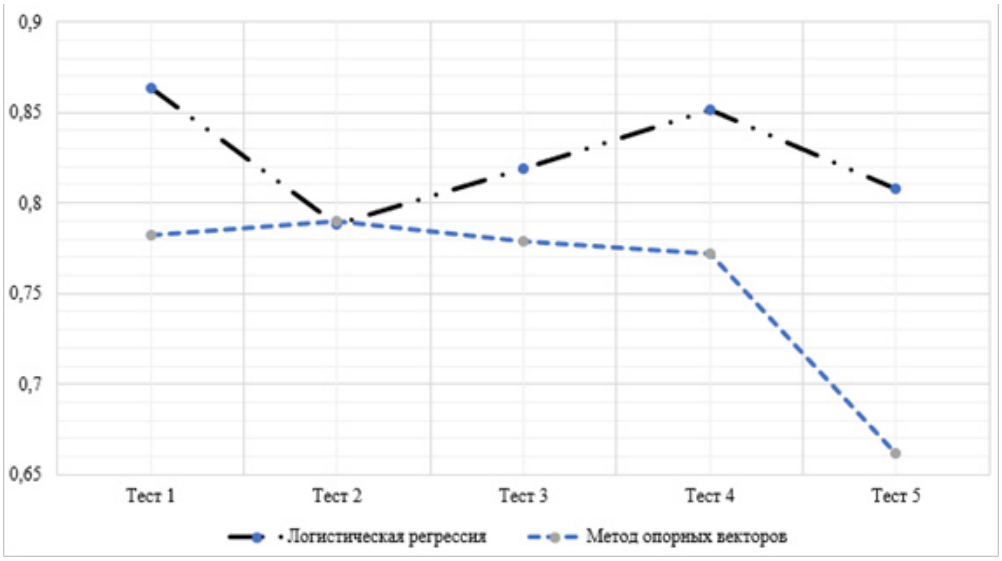

Рисунок 4 – Результаты экспериментов по определение оптимального классификатора сообщений в офлайн-режиме

В завершении исследования проведено нагрузочное тестирование для анализа работы алгоритма. Для моделирования работы пользователей в системе использовалась утилита для проведения нагрузочного тестирования *JMeter* [21]. Суть тестирования в следующем: пользователи регистрируются в системе и со случайной вероятностью создают элементы с токсичным содержимым. В качестве второго примера работы прототипа модуля модерации контента СС в офлайн-режиме рассмотрен вариант, когда содержащий токсичный текст элемент уже опубликован. В данном случае это заметка с токсичным блоком данных. Отслеживание в онлайн-режиме не нашло токсичного содержимого ввиду того, что анализировался только заголовок структурного элемента. Отслеживание и проверка содержимого осуществлялись алгоритмом в офлайн-режиме. В примере использовались сообщения, приобретающие токсичный окрас только в специфичном контексте. Более того, возможны попытки видоизменения данных с целью обхода системы модерации. После обнаружения токсичного содержимого текстовые данные внутри заметки обновляются и элемент отображается в нетоксичном виде. Модератору также отправляется уведомление об обнаружении элемента, нарушающего правила сообщества. Из проведенных тестов получено, что максимальная нагрузка, которую способен обработать сервис в офлайн-режиме, порядка 600 операций в секунду. Далее система переходит в состояние насыщения, и все вновь поступающие на модерацию запросы помещаются в очередь и ожидают обработки. Полученный показатель нагрузки является вполне приемлемым уровня предлагаемого прототипа.

Заключение. В статье представлен прототип модуля автоматической модерации текстового содержимого СС. В работе рассмотрены решения, на основе которых предложены методы обработки и классификации текстовых данных. Предложенный алгоритм функционирует в онлайн (модель модерирует текстовое содержимое на основе словаря запрещенных слов, рассчитывая *TF-IDF* меру для выявления текстового содержимого с ненормативной лексикой) и офлайн (текст обрабатывается с помощью предобученной языковой модели, с помощью которой возможно модерировать содержимое текстовых сообщений) режимах. Результаты экспериментальной проверки свидетельствуют о том, что прототип справляется со своей задачей на тестовых данных. Для более подробного исследования и получения результатов будут проведены дополнительные исследования.

СПИСОК ЛИТЕРАТУРЫ:

1. Реутов Е.В. Социальные сети в региональном сообществе: монография / Реутов Е.В.,Колпина Л.В., Реутова М.Н., Бояринова И.В.; отв. ред. Е.В. Реутов. Белгород: Константа, 2011. – С. 80–88.

2. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Изд-во физико-математической литературы, 2010. – 228 с.

3. Arefyev N. Evaluating Three Corpus-Based Semantic Similarity Systems for Russian. / Arefyev N., Panchenko A., Lukanin A., Lesota O., Romanov P. // In Proc. of the 21st International Conference on Computational Linguistics and Intellectual Technologies Moscow, Russia. RGGU. – 2015. – pp. 106-118.

4. Что такое модерация: определение понятия, виды и способы [Электронный ресурс]. – URL: https://www.calltouch.

ru/glossary/moderatsiya-chto-eto-takoe-i-gde-primenyaetsya/ (дата обращения: 15.04.2021).

5. Батура Т. В. Методы анализа компьютерных социальных сетей // Вестник НГУ. Серия: Информационные технологии. – 2012. – Т. 10. – вып. 4. – С. 13-28.

6. gRPC | Microsoft Docs [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/dotnet/architecture/cloud-native/grpc (дата обращения: 15.04.2021).

7. Михайлов Д. В., Козлов А. П., Емельянов Г. М., Выделение знаний и языковых форм их выражения на множестве тематических текстов: подход на основе меры TF-IDF // Компьютерная оптика. – Т. 39. – № 3. – 2015. – С. 429–438.

8. Маккинни, У. Python и анализ данных / У. Маккинни; перевод с английского А. А. Слинкина. – 2-ое изд., испр. и доп. – Москва: ДМК Пресс, 2020. – 540 с.

9. Алгоритм Ахо-Корасик - Алгоритмика [Электронный ресурс]. – URL: https://algorithmica.org/ru/aho-corasick/ (дата обращения: 15.04.2021).

10. Автоматизация выявления модификаций в образе договорных документов с помощью модели N-грамм / Блог компании Smart Engines / Хабр [Электронный ресурс]. – URL: https://habr.com/ru/company/smartengines/blog/500310/ (дата обращения: 17.04.2021).

11. Бершадская Е.Г., Видясова Л.А. Автоматизированный инструмент опинион-майнинг как средство обработки текстов // Современные методы и средства обработки пространственно-временных сигналоВ: сборник статей XIV Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2016. – С. 46–50.

12. Бершадская Е.Г., Назиров Р.Р. Проблемы сбора и представления неструктурированной информации из открытых источников // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XVI Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2018. – С. 64–68.

13. Как легко понять логистическую регрессию / Блог компании .io / Хабр [Электронный ресурс]. – URL: https://habr.com/ru/company/io/blog/265007/ (дата обращения: 17.04.2021).

14. Что такое перекрестная проверка | Data Science [Электронный ресурс]. – URL: http://datascientist.one/cross-validation/ (дата обращения: 15.04.2021).

15. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес / Блог компании Open Data Science / Хабр [Электронный ресурс]. – URL: https://habr.com/ru/company/ods/blog/324402/ (дата обращения: 15.04.2021).

16. 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).

17. Метод опорных векторов – Supported Vector Machine (SVM) [Электронный ресурс]. – URL: http://statistica.ru/branches-maths/metod-opornykh-vektorov-supported-vector-machine-svm/ (дата обращения: 10.04.2021).

18. 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.

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

20. Маркин Е.И., Подопригора И.А., Зоткина А.А., Бершадская Е.Г. Методы разработки искусственных нейронных сетей // Вестник современных исследований. – 2018. – № 3.1 (18). – С. 49–52.

21. jMeter - Краткое руководство - CoderLessons.com [Электронный ресурс]. – URL: https://coderlessons.com/tutorials/java-tekhnologii/vyuchi-jmeter/jmeter-kratkoe-rukovodstvo (дата обращения: 15.04.2021).

Статья поступила в редакцию 13.05.2021
Статья принята к публикации 16.06.2021