# Введение

Развитие социальных сетей и интернет-коммуникаций способствует появлению новых феноменов. В современной интернет-терминологии активно используются такие понятия, как «троллинг» и «тролль». С точки зрения социологии, данная проблема рассмотрена в нескольких источниках, где описано, что троллинг может воздействовать на общее моральное состояние, вплоть до проблем с психикой. Эта тема притягивает к себе особенное внимание психологов и общественности из-за отрицательного воздействия на будущее поколение и стабильное психологическое здоровье общества вцелом. В последнее время троллинг окончательно адаптировался в обществе и в сети Интернет, из-за чего перешел в уровень «нормы».

***Целью*** данного исследования является анализ сообщений пользователей с платформы Twitter для выявления провокационных текстов.

В рамках проекта ставятся следующие ***задачи***:

Предобработка данных: осуществить необходимые операции для подготовки данных к анализу, включая удаление выбросов и заполнение пропущенных значений. Это позволит обеспечить надежность и качество анализа на основе доступных данных.

Разведывательный анализ данных: провести исследование структуры данных, выявить закономерности и тренды в текстах сообщений. Визуализация данных и выявление основных характеристик провокационных сообщений помогут в выявлении троллей и снижении дезинформирующих сообщений.

# Обзор литературы по теме исследования

Некоторые исследователи уже пробовали изучить источники информации в разных сообществах. В одних [6] предлагаются модели оценивания авторитета пользователей. Они основаны на объеме лучших ответов пользователей. В других [7] авторы основываются на выборе вопросов, предпочитаемых пользователем для ответа. Эксперты всегда отвечают на вопросы, в которых больше всего компетентны. Есть авторы, которые предлагают более глубокие структуры, основанные на поведенческих и когнитивных условиях пользователей, для оценки не только точности, но и опыта пользователей, дающих информацию [8].

В одной из актуальных статей [1] авторы предлагают использовать теорию функций убеждения, учитывая большие объемы данных в различных интернет-сообществах, а также высокую вероятность неточности или неопределенности, как источника информации, так и самих данных. Этот метод направлен на обнаружение троллей в сообществах вопросов-ответов с использованием функций убеждений. Рассматриваемый подход, позволяет точнее определить особенность пользователя благодаря характеру опубликованных сообщений. Он основан на мере конфликта в теории функций убеждения, применяемой между сообщениями разных пользователей сообщества. Прежде всего, этот метод направлен на анализ сообщений относительно опубликованного вопроса или темы. Но обнаружения нерелевантного сообщения недостаточно, чтобы судить, является ли этот пользователь троллем. Таким образом, не только содержание сообщений характеризует троллей, но и их поведение.

# Методология исследования (данные, методы, технологии)

В данном исследовании используется набор данных (рисунок 1), состоящий из комментариев пользователей сети интернет, в количестве 14412. Второй характеристикой дата сета является бинарный показатель (принимает значения 0 или 1), который указывает, токсичный комментарий или нет.

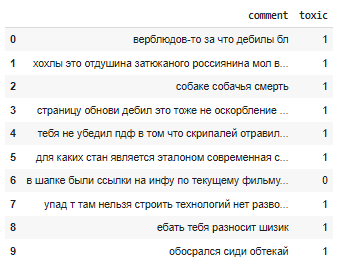


Рисунок 1. Первые 10 строк из набора данных

Перед тем, как набор данных можно будет использовать для создания модели машинного обучения, необходимо его подготовить. Предварительная обработка исходных данных играет важную роль для построения качественной модели. Первым делом посмотрим на целевую переменную «toxic». Как видно из рисунка 2, данный показатель имеет дисбаланс классов. Следовательно, при оценке модели мы сосредоточимся на показателях F1, точности и отзывчивости (precision and recall).



Рисунок 2. Распределение переменной на классы «токсичный» и «не токсичный»

Для подготовки данных при решении задач обработки естественного языка (англ. Natural Processing Language, NLP) в машинном обучении используются специальные обработчики исходного текста, которые позволяют разбить его на блоки, пригодные для подачи в модель. Такие обработчики называются токенизаторами (англ. Tokenize). Токенизация — самый первый шаг при обработке текста, результатом которого является набор (список) так называемых токенов (подстрок). Они могут быть предложениями, словами или даже отдельными символами. Предварительно необходимо привести данные к одному регистру (был выбран нижний регистр), а также очистить от лишних элементов: знаков препинания, букв других языков, табуляций и переходов на новую строку. В результате можем получить строки, в которых не останется данных, их стоит удалить. На рисунке 3 представлен топ 40 слов, которые встречаются в комментариях.

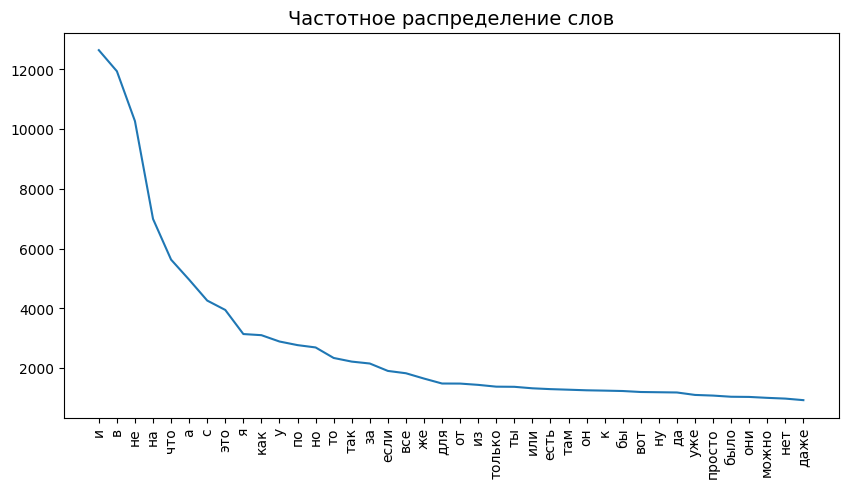


Рисунок 3. Частотное распределение слов

Как видно из рисунка 3, чаще всего встречаются предлоги и союзы. Убрав слова длиной в 1 и 2 символа, получаем более информативную картину (рисунок 4).

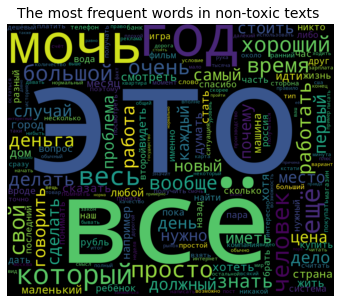
 

Рисунок 4. Частота слов в токсичных комментариях:

слева – учитывая союзы и предлоги, справа – без их учета

Подобно тому, как с помощью диаграмм плотности распределения можно найти мелкие признаки разграничения и размытости, мы можем исследовать степень сходства документов по всем признакам. Для этой цели часто используется очень популярный метод t-распределенного стохастического вложения соседей(t-distributed Stochastic Neighbor Embedding, t-SNE). Библиотека Scikit-Learn реализует метод разложения t-SNE в виде преобразователя sklearn.manifold.TSNE. Метод t-SNE способен сгруппировать схожие документы путем разложения векторных представлений документов с большим числом измерений в два измерения (с использованием распределений вероятностей из оригинальной и разложенной размерностей). Разложением на два или на три измерения можно построить диаграммы рассеяния [4].

К сожалению, метод t-SNE требует слишком много вычислительных ресурсов, поэтому на первых этапах чаще применяются более простые методы разложения, такие как SVD (сингулярное разложение) или РСА (метод главных компонент). В библиотеке Yellowbrick есть класс TSNEVisualizer, который создает внутренний конвейер преобразования, применяющий сначала разложение SVD c 50 компонентами по умолчанию, а затем выполняющий вложение t-SNE [4]. Применим данный метод к набору данных.

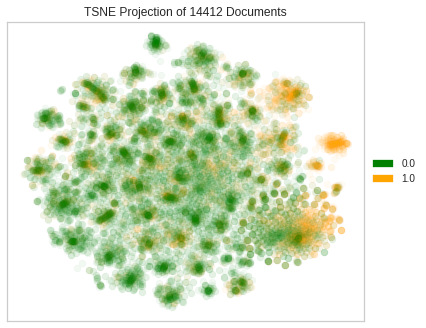
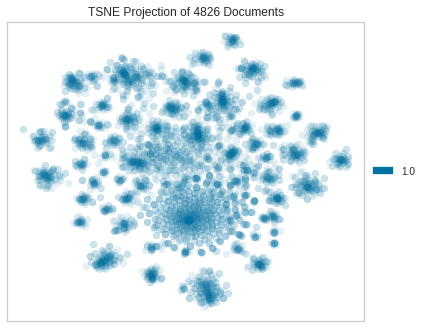


Рисунок 5. t-SNE проекция: слева – слова, имеющие значение класса «toxic» равное 1,  
справа – наложение групп слов, имеющих значение класса «toxic» равное 0 и 1

Как видно из рисунка, элементы классов достаточно плотно группируются и сами классы накладываются друг на друга. Есть два небольших скопления (желтое / токсичное – права, зеленое / не токсичное - вверху) где классы не накладываются друг на друга.

Последним шагом перед обучением модели будет лемматизация – приведение всех слов текста к их нормальным формам с учетом контекста. Одним из передовых решений является библиотека PyMystem3 от Яндекса. Теперь, используя токенизатор keras framework, превратим текст в последовательность целых чисел. После этого выбираем максимальную длину комментария, все более длинные комментарии будут обрезаны, а те, что короче, будут дополнены.

Для обучения выбрана модель Sequential из библиотеки Keras. Несмотря на частое объединение с фреймворками глубокого обучения, такими как TensorFlow, Caffe, Theano и PyTorch, библиотека Keras предлагает более обобщенный программный интерфейс для глубокого обучения. Первоначально библиотека Keras разрабатывалась для работы с фреймворком Theano, но после открытия исходных текстов и взрывного роста популярности TensorFlow библиотека Keras быстро стала одним из основных инструментов для многих пользователей TensorFlow и была включена в ядро TensorFlow в начале 2017 года. В Keras все сущее является объектом, что делает ее особенно удобным инструментом для прототипирования. Чтобы получить аналог многослойного классифицирующего перцептрона, определим функцию, которая создает экземпляр модели Sequential из Keras и добавляет 4 скрытых слоя [4]. Получаем следующую модель, представленную на рисунке 6.

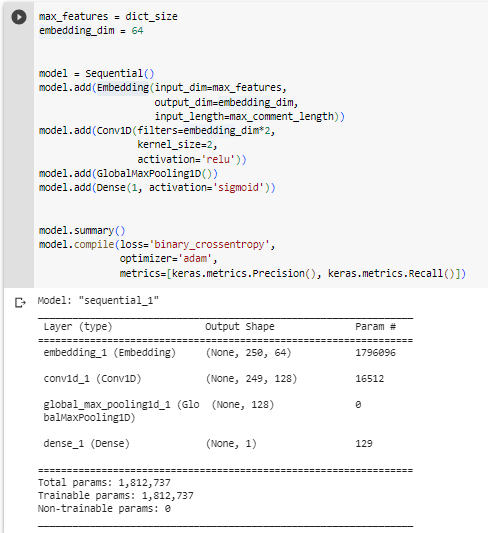


Рисунок 6. Фрагмент кода: создание экземпляра модели Sequential из библиотеки Keras

Проводим обучение модели, установив параметр количество эпох равным 50. По окончании обучения оценим изменение сетевой ошибки по эпохам. Как видно из рисунка 7 после 4ой эпохи данный показатель начинает расти. Поэтому как окончательную модель выбираем ту, которая имеет параметр количество эпох равный 4.

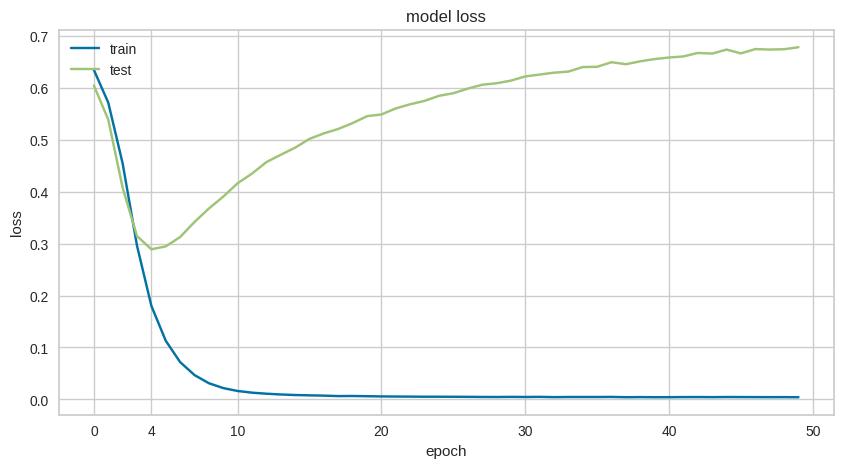


Рисунок 7. График изменения сетевой ошибки по эпохам

# Результаты

В ходе исследования были получены следующие результаты:

Был проведен анализ и предобработка данных, включая удаление лишних элементов и коротких слов, приведение к одному регистру и токенизация. Это позволило обеспечить надежность и точность анализа. В процессе исследования было принято решение не удалять стоп-слова, они вносят серьезную смысловую нагрузку в контексте комментариев, что повышает точность модели. Получена модель, точность которой составляет 0,867.

Для данной модели также был разработан веб интерфейс, на котором можно оценить работу модели.

# Выводы

Троллинг в сети – деструктивное явление, которое приводит к усложнению получения информации пользователями. Существуют рейтинговые системы саморегуляции [9], однако даже они не обходятся без модерации. В ходе исследования получена модель, с помощью которой можно точнее определить особенность пользователя благодаря характеру опубликованных сообщений. Для дальнейшего улучшения модели необходимо расширять датасет и количество токенов. Существует большое количество сленговых выражений, слов записанных с ошибками (случайно или намеренно), что мешает точности прогнозирования модели. В данном исследовании не было проведено сравнение результатов разных моделей, что могло бы выявить методы с большей точностью.

# Список источников:

1. Метод выявления "троллей" в сетевых сообществах / Е. В. Измайлова, Д. А. Алексеев, В. В. Свечникова, А. В. Сорокина // Философские проблемы информационных технологий и киберпространства. – 2022. – № 2(22). – С. 4-17. – DOI 10.17726/philIT.2022.2.1. – EDN TSCIRG.
2. Lin Miao, Mark Last, and Marina Litvak. Detecting Troll Tweets in a Bilingual Corpus. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 6247–6254, [Электронный ресурс]. 2020. URL: https://aclanthology.org/2020.lrec-1.766/ (дата обращения: 19.06.2023)
3. Стивен А. Рейнс и другие , Взаимодействие с партизанскими твитами российских троллей во время президентских выборов в США в 2016 г.: перспектива социальной идентичности, Journal of Communication , том 73, выпуск 1, февраль 2023 г., страницы 38–48, [Электронный ресурс]. 2023. URL: https://academic.oup.com/joc/article-abstract/73/1/38/6965707?redirectedFrom=fulltext (дата обращения: 19.06.2023)
4. Бенгфорт Б., Билбро Р., Охеда Т Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. — СПб.: Питер, 2019. — 368 с.:ил. — (Серия «Бестселлеры O’Reilly»).
5. Ганегедара, Т. Обработка естественного языка с TensorFlow : монография / Т. Ганегедара ; пер. с анг. В. С. Яценкова. - Москва : ДМК Пресс, 2020. - 382 с. - ISBN 978-5-97060-756-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1094940 (дата обращения: 20.06.2023). – Режим доступа: по подписке.
6. Smets P., Kennes R. The Transferable Belief Model // Artificial Intelligence. – 1994. – Vol. 66. – P. 191-234.
7. . Cambria E., Chandra P., Sharma A., Hussain A. Do not feel the trolls // Proceedings of the 9th International Semantic Web Conference, 2010. URL: http://ceur-ws.org/Vol‑664/paper1.pdf
8. Ortega F., Jose A., Fermin L., Carlos G., Fernando E. Propagation of trust and distrust for the detection of trolls in a social network // Computer Networks. – 2012. – Vol. 56. – P. 2884-2895.
9. Затонский А. В., Иванова Е.В. Методы формализации самооценки на примере научно-исследовательской работы студентов // Информатизация образования и  науки.  – 2011.  – № 3 (11).  – С.  110-116. (Zatonskij A.V., Ivanova E.V. Methods of formalization ofself-assessment on the example ofstudents’research work // Informatizaciya obrazovaniya i nauki. – 2011. – № 3 (11). – P. 110-116.)