

Санкт-Петербургский государственный университет

Направление Математическое обеспечение и администрирование
информационных систем

Смирнов Александр Львович

Классификация текстового контента

Курсовая работа

Научный руководитель:
д. ф.-м. н., профессор Литвинов Ю. В.

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Alexander Smirnov

Classification of text content

Course Work

Scientific supervisor:
Associate Professor Yuri Litvinov

Saint-Petersburg
2019

Оглавление

Введение	4
1. Обзор существующих решений	7
2. Описание предлагаемого решения	8
2.1. Обучение модели	8
2.2. Расширение для Chrome и сервер	9
Заключение	11
Список литературы	12

Введение

Для изучения данной работы требуются знания предметной области, поэтому введем некоторые понятия и определения.

Понятия:

- Бинарная классификация контента — разделение контента на 2 условные группы
- Расширение браузера — компьютерная программа, которая в некотором роде расширяет функциональные возможности браузера
- Сервер — локальный компьютер, выполняющий обработку запросов
- GET-запрос запрашивает данные с сервера
- POST-запрос отправляет данные, подлежащие обработке, на указанный сервер
- Нейронная сеть - алгоритм машинного обучения, построенный по принципу организации и функционирования биологических нейронных сетей [1]

Подходы к классификации текста:

- Rule-based - подход, основанный на классификации по заданным заранее правилам. Например, по наличию или отсутствию тех или иных слов
- Machine Learning (ML) based - подход, основанный на алгоритмах машинного обучения [2]
- Hybrid Systems - подход, совмещающий в себе ML Based и Rule-based подходы

Характеристики сравнения эффективности:

- Accuracy — общая точность классификатора
- Recall — отношение заблокированных взрослых сайтов к общему количеству взрослых сайтов (% классифицированных взрослых сайтов)

- Precision — отношение заблокированных взрослых сайтов к числу всех заблокированных сайтов (точность блокировки)
- F1 Score — среднее гармоническое между Precision и Recall, для учёта и того, и другого в одной величине

Актуальность работы

Каждый родитель желает оградить своего ребенка от плохого влияния внешнего мира. Интернет несет в себе не только массу полезной информации, но и огромное количество негатива, которое может сформировать у ребенка неправильное мировоззрение или восприятие действительности. Существует еще масса «взрослых» сайтов, просмотр которых ребенку категорически запрещен. Поэтому у родителей возникает вопрос: «Как защитить ребенка от ненужных сайтов?»

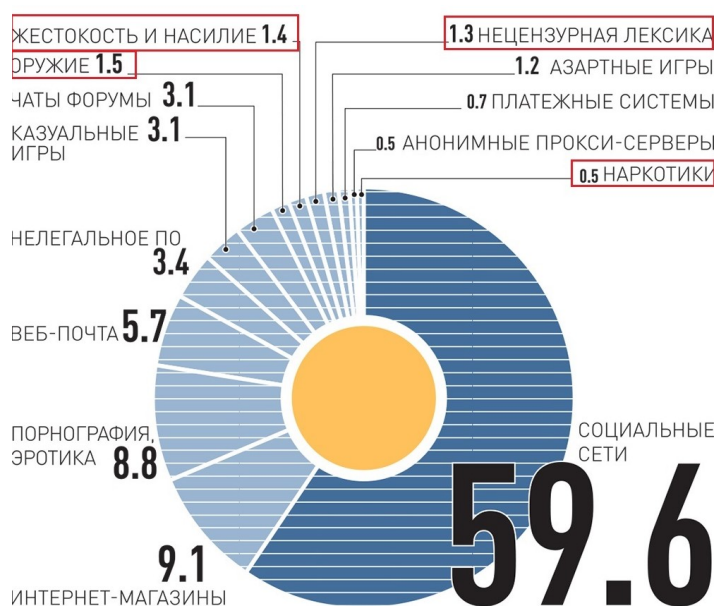


Рис. 1: Что интересует детей в интернете (Ист. – Лаборатория Касперского)

Цели

- Ограничить детей от взрослого текстового контента в интернете
- Получить опыт

- Бинарная классификация текста
- Написание расширения для Chrome
- Написание Python-сервера для приёма запросов

Задачи

- Провести анализ возможных решений для классификации текста
- Написать Python-сервер, использующий обученную модель для ответа на запросы от расширения
- Сделать расширение для Chrome, обращающееся к серверу

Реализация данных задач позволит полностью ограничить детей от негативного влияния интернета, так как программа будет блокировать конкретные страницы, содержащие недопустимый для детей контент.

1. Обзор существующих решений

Существует два типа решения поставленной задачи:

- Ограничения на поиск
 - Семейный поиск Яндекс
 - Безопасный поиск Google
- Контентная фильтрация
 - Traffic Inspector
 - Интернет Цензор

Данные решения проблемы не являются оптимальными. В случае ограничения на поиск фильтрация происходит по ключевым словам, что позволяет просматривать непристойный для детей контент, переходя напрямую по ссылкам. Обратная же ситуация с контентной фильтрацией - ограничение накладывается конкретно на определенные ссылки. К тому же большинство таких программ имеют абсолютно не дружелюбный интерфейс, не понятный обычному пользователю (2).

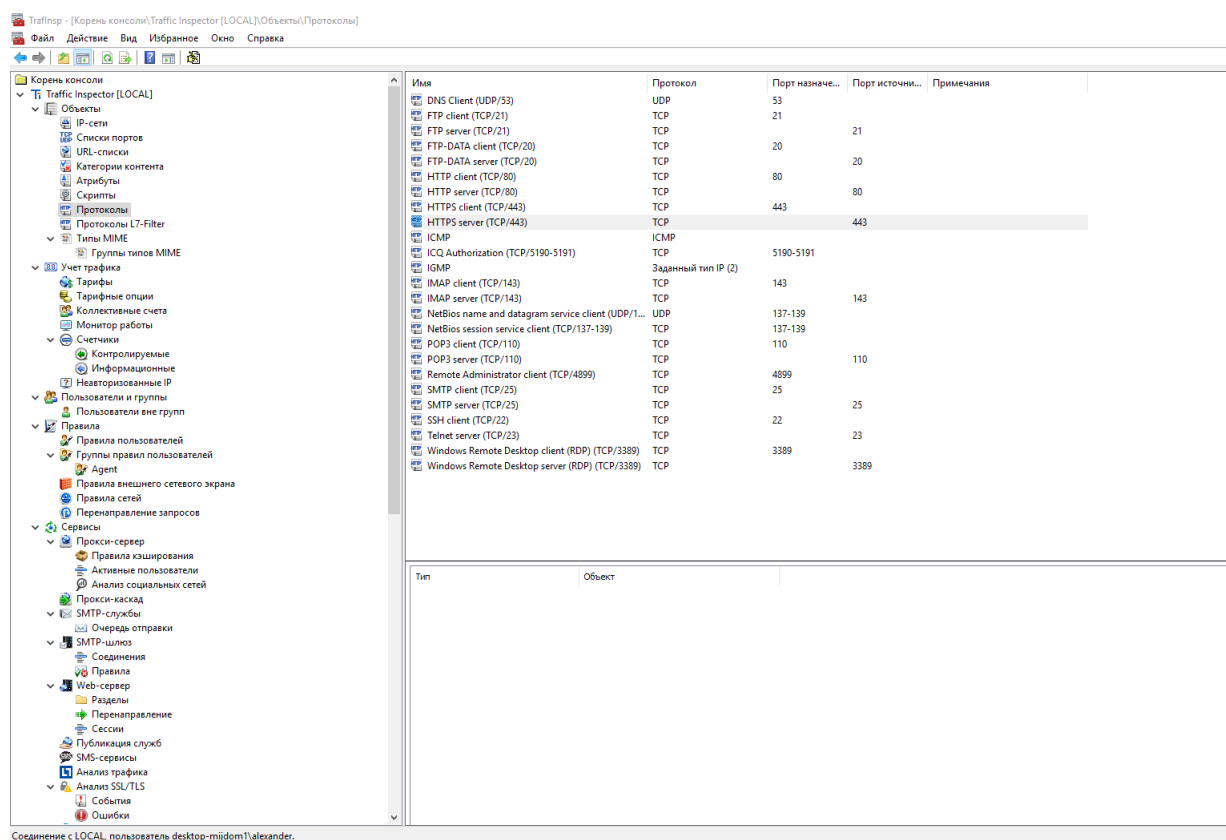


Рис. 2: Интерфейс программы Traffic Inspector

2. Описание предлагаемого решения

Подход, предлагаемый в данной работе, заключается в том, чтобы классифицировать страницы на взрослые и детские в зависимости от текстового контента на них.

Подход классификации веб-страниц по содержимому хорош тем, что в отличие от подходов, основанных на блокировке по URL, нам не нужно иметь огромную базу адресов, подлежащих блокировке, которую, к тому же, нужно постоянно поддерживать в актуальном состоянии. Также, данный подход имеет преимущество над блокировкой результатов в поисковой выдаче в том, что невозможно будет напрямую попасть на страницу, зная её домен.

Будем применять метод машинного обучения – нейронные сети. Для этого нам нужно подготовить данные для обучения, построить модель, обучить модель на размеченных данных и использовать обученную модель для классификации содержимого сайта.

2.1. Обучение модели

Для начала нам необходимо научиться представлять рассказы в виде, в котором мы можем их обрабатывать. Делать это мы будем с помощью словаря наиболее популярных в русском языке слов следующим образом: каждому рассказу в предложении мы будем сопоставлять список из 10000 элементов (размер словаря), в котором каждым элементом будет являться значение 1 либо 0 (в зависимости от наличия или отсутствия данного слова в словаре). Далее мы попробуем и сравним 4 архитектуры:

- Random model – случайный выбор блокировать/не блокировать
- Rule-based model – блокировка по списку непотребных слов
- Classifier – 3-х слойная обычная сеть
- Upgraded Classifier – Classifier, из словаря которой были исключены самые частые слова и добавлена ненормативная лексика

Получили следующие результаты:

	F1 Score	Accuracy	Recall	Precision
Random model	—	0.51	—	—
Rule-based model	0.06	0.41	0.03	1.0
Classifier	0.90	0.88	0.93	0.87
Upgraded Classifier	0.91	0.90	0.92	0.91

Можем видеть, что наиболее выгодной моделью, как и можно было предположить, является Upgraded Classifier, которую мы и будем использовать в дальнейшем.

Также взглянем на рисунки (3) и (4):

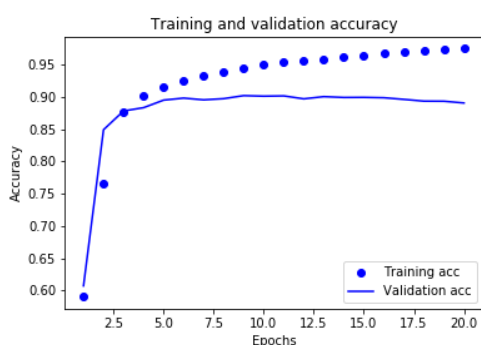


Рис. 3: Зависимость точности от количества эпох

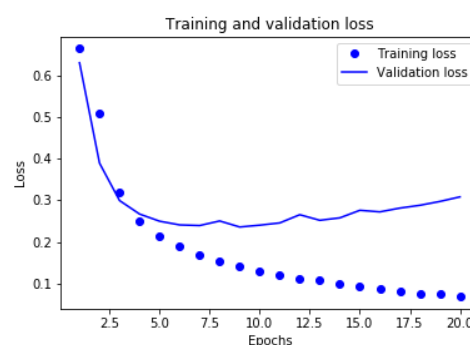


Рис. 4: Зависимость потерь от количества эпох

Из зависимости на (3) можно сделать вывод, что после 10-ти эпох точность предсказаний модели на данных, которых она ещё не видела, не растёт, вследствие чего дальнейшее обучение не имеет смысла. На (4) видно, что после 7-й итерации обучения функция потерь начинает расти, что свидетельствует о переобучении модели и необходимости преостановить обучение.

2.2. Расширение для Chrome и сервер

Следующим шагом является написание сервера, задача которого обрабатывать POST/GET запросы от расширения. Работает он следующим образом: Получаем POST-запрос от расширения с текущей html-страницей, выбираем из него все русские слова, классифицируем страницу по заранее обученной модели и записываем результат в файл. Да-

лее мы получаем GET-запрос и отправляем расширению полученный результат из файла (5).

Работает это следующим образом:

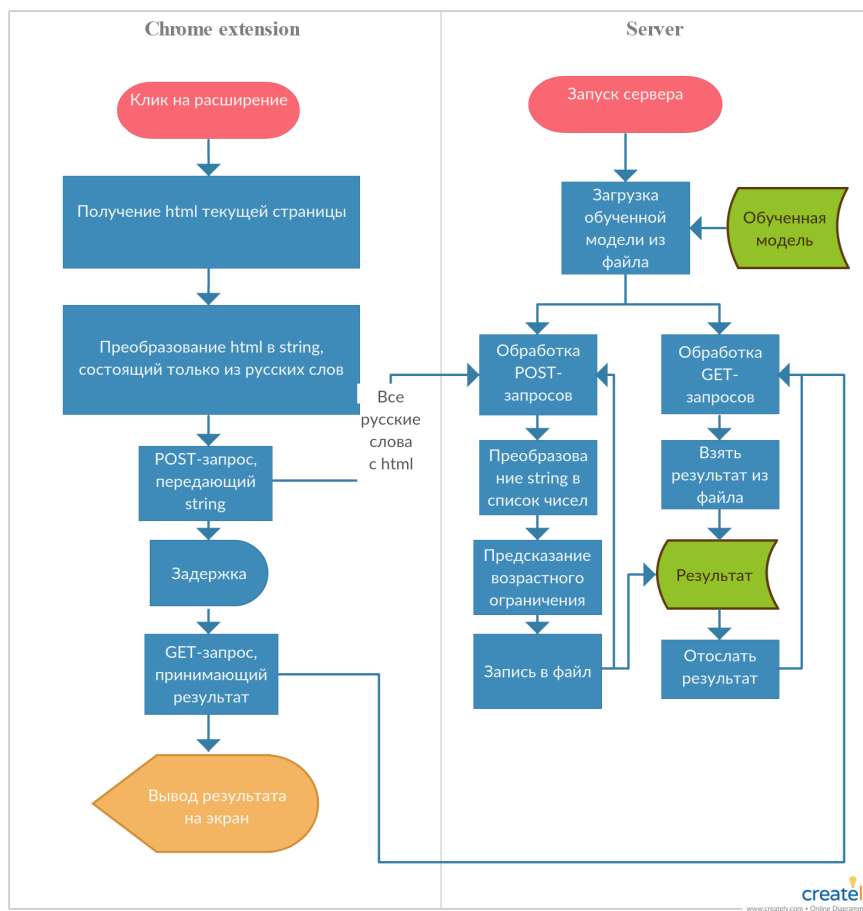


Рис. 5: Диаграмма взаимодействия расширения и сервера

Заключение

- Сделано расширение для Chrome
- Сделан сервер на Python, принимающий POST/GET запросы
- Выбрана и обучена оптимальная модель

Список литературы

- [1] Wikipedia. Нейронная сеть // Википедия, свободная энциклопедия. — 2011. — URL: <https://clck.ru/AQEDJ> (дата обращения: 05.06.2019).
- [2] Wikipedia. Машинное обучение // Википедия, свободная энциклопедия. — 2012. — URL: <https://clck.ru/ERPm2> (дата обращения: 05.06.2019).