Оглавление

Введение....2
Глава 1. Обзор законодательной и нормативно-правовой базы в области интернет-цензуры в Российской федерации....4
Федеральный закон № 398 от 28 декабря 2013 года....4
Федеральный закон № 149-ФЗ «Об информации, информационных технологиях и о защите информации»....4
Глава 2. Построение математической модели классификатора....7
Формальная постановка задачи....7
Обзор методов решения....8
Линейная регрессия....8
Методы на основе искусственных нейронных сетей....10
Метод Байеса....12
Метод опорных векторов....13
Предварительная обработка данных....15
Оценка качества классификации....19
Глава 3. Сбор информации....23
Получения списка запрещенных web-ресурсов....23
Подготовка списка разрешенных данных....27
Сбор данных с web-страниц....29
Получение запрещенной информации....31
Глава 4. Программная реализация классификатора....32
Подготовка данных....33
Выбор алгоритма классификации....36
Залючение....37

Введение

30 июля 2012 года вступил в силу ФЗ-139 от 28.07.2012 года, в соответствии с которым в РФ появился единый реестр запрещенных сайтов. Данный реестр представляет собой базу данных доменных имен, URL а также сетевых адресов содержащих информацию, распространение которой нежелательно в России. Некоторые государственные ведомства получили право на добавление записей в этот реестр, причем в некоторых случая без суда.

Но федеральный закон не предписывал порядок добавление записей в этот реестр. Этот фактор показал несовместимость с принципом WEB 2.0, а именно привлечение пользователей к наполнению ресурсов информационным материалом. Целые сайты начали подвергаться блокировкам, из-за информации размещенных на них отдельными пользователями. Например в августе 2015 года была заблокирована, а позже разблокирована интернет-энциклопедия «Wikipedia», что сопровождалось экономическими расходами. А 16 апреля 2018 года на территории РФ был заблокирован интернет-мессенджер Telegram, но техническая блокировка мессенджера оказалась неэффективной, и повлияло на работу многих сторонних сервисов, но практически не отразилась на доступности самого мессенджера.

Если от таких блокировок, как в случае блокировки Telegram защититься практически невозможно, возможно лишь наблюдать доступность ресурса, то целью данный дипломной работы будет создание системы «раннего реагирования», которая сможет защитить владельцев ресурсов от действия некоторых их пользователей, методом оценки контента.

Таким образом, возникает угроза нарушения доступности интернет ресурсов, из-за умышленных или неумышленных действий пользователей. Снизить риски могла бы премодерация контента, но это повлекло бы дополнительные материальные затраты, снизило бы оперативность доставки контента на ресурсы. Классификация информации активно используется в

различных областях, таких как фильтрация спама, dlp-системах, новостных агрегаторах. Таким образом, я считаю, возможно использование существующих наработок в области классификации информации для определения информации запрещенной к распространению в Российской Федерации.

Для получение практической ценности и достижение поставленной цели необходимо решить ряд зад:

- Подготовить выборки для обучения тестирования
- Разработать математическую модель классификатора информации
- Реализовать этот классификатор в качестве программного продукта
- Произвести обучение, используя обучающую выборку
- Произвести тестирование полученного классификатора
- Произвести анализ результатов тестирования

Глава 1. Обзор законодательной и нормативно-правовой базы в области интернет-цензуры в Российской федерации.

Согласно 4 и 5 пункту 29 статьи конституции Российской федерации, в стране запрещена цензура и каждому дано право свободно распространять информацию, за исключением государственной тайны.

Федеральный закон № 398 от 28 декабря 2013 года

Данный федеральный закон направлен на борьбу с экстремизмом в сети интернет. Сам федеральный закон не содержит определение экстремизма, определение экстремизма содержится в ФЗ №114 «О противодействии экстремистской деятельности», по определению экстремизм это:

Экстремизм (от фр. extremisme, от лат. Extremus – крайний) – приверженность к крайним взглядам и, в особенности, мерам.Федеральный закон от 2 июля 2013 года № 187-ФЗ «О внесении изменений в законодательные акты Российской Федерации по вопросам защиты интеллектуальных прав в информационно-телекоммуникационных сетях»

Данный федеральный закон предполагает блокировку ресурсов, распространяющих законную информации незаконным способом, а именно направлен на поддержку правообладателей информационного контента. Информация заблокированная исходя из нарушения данного закона не следует считать незаконной и следовательно не включать в выборку.

Федеральный закон № 149-ФЗ «Об информации, информационных технологиях и о защите информации»

Федеральный закон, реализацией которого является «Единый реестр доменных имен, указателей страниц сайтов в сети "Интернет" и сетевых адресов, позволяющих идентифицировать сайты в сети "Интернет", содержащие информацию, распространение которой в Российской Федерации

запрещено» определяет критерии информации, распространение которой должно быть запрещено.

В статье 15.1 определяется список критериев, по которому информационный ресурс будет помещен в этот реестр:

1. Материалов с порнографическими изображениями несовершеннолетних и (или) объявлений о привлечении несовершеннолетних в качестве исполнителей для участия в зрелищных мероприятиях порнографического характера;
2. Информации о способах, методах разработки, изготовления и использования наркотических средств, психотропных веществ и их прекурсоров, новых потенциально опасных психоактивных веществ, местах их приобретения, способах и местах культивирования наркосодержащих растений;
3. Информации о способах совершения самоубийства, а также призывов к совершению самоубийства;
4. Информации о несовершеннолетнем, пострадавшем в результате противоправных действий (бездействия), распространение которой запрещено федеральными законами;
5. Информации, нарушающей требования Федерального закона от 29 декабря 2006 года N 244-ФЗ "О государственном регулировании деятельности по организации и проведению азартных игр и о внесении изменений в некоторые законодательные акты Российской Федерации" и Федерального закона от 11 ноября 2003 года N 138-ФЗ "О лотереях" о запрете деятельности по организации и проведению азартных игр и лотерей с использованием сети "Интернет" и иных средств связи;
6. Информации, содержащей предложения о розничной продаже дистанционным способом алкогольной продукции, и (или) спиртосодержащей пищевой продукции, и (или) этилового спирта, и (или) спиртосодержащей непищевой продукции, розничная продажа которой

ограничена или запрещена законодательством о государственном регулировании производства и оборота этилового спирта, алкогольной и спиртосодержащей продукции и об ограничении потребления (распития) алкогольной продукции;

Глава 2. Построение математической модели классификатора

В данной главе будут рассмотрены существующие подходы к решению задач по классификации, также будет проведено построение модели классификатора. Классификация информации на разрешенную/запрещенную является задачей двухклассовой классификации, наиболее простой в техническом отношении случай, который служит основой для решения более сложных задач[1].

Задача классификации текстов, на первый взгляд не может быть решена заранее известным алгоритмом, по причине того, что не существует алгоритма, который мог бы генерировать тексты указанных классов. В связи с чем для создания алгоритма целесообразно применять алгоритмы машинного обучения. Машинное обучение – обширная и перспективная область исследований, использующая объемную теоретическую базу, а также имеющая обширную программную базу[2], в связи с чем невозможно рассмотреть все существующие подходы к решению подобных задач в рамках данной работы, и будут рассмотрены лишь основные подходы и возможности их комбинирования для данной прикладной цели.

Формальная постановка задачи

Пусть T – множество все текстов, $C=\{Запрещенный, Разрешенный\}$ – множество классов определяющие, содержит ли данный фрагмент текста запрещенную информацию, тогда, имея некоторую входную выборку размером N $D=\{(t_i,t_i), t_i \in T, c_i \in C, i \in \overline{1,N}\}$

Тогда задача сводится к нахождению $P(c|t)$, то есть условному распределению классов текста на данном тексте.

Будем считать тексты, условная вероятность которых $p(c_1|t)>\frac{1}{2}$ запрещенными к распространению в Российской Федерации. [3]

Обзор методов решения

Задача классификации является частным случаем обучения с учителем(supervised learning), популярными методами ее решения считаются линейная регрессия, машина опорных векторов(SVM), наивный Байес, деревья решений, искусственные нейронные сети, метод k-ближайших соседей.

Также для увеличения точности возможно использовать композиции алгоритмов для получения более точных результатов, примерами композиции алгоритмов могут служить голосования, бустинг и бэггинг.

Для подбора подходящего алгоритма необходимо также рассмотреть данные, их свойства и способы предварительной обработки.

Линейная регрессия

Введем вектор $\vec{Z}=(Z_0, Z_1, \dots, Z_n)$ факторов регрессии и вектор неизвестных параметров регрессии $\vec{V}=(V_0, V_1, \dots, V_n)$

Тогда функция линейной регрессии будет иметь вид: $f(\vec{Z})=(\vec{V}, \vec{Z})$

Пусть на i-ом эксперименте факторы регрессии принимают заранее известные значения $\vec{Z^i}=(Z_0^i, Z_1^i, \dots, Z_n^i)$

После $k \geq n$ экспериментов будет получен набор откликов: $F=\begin{bmatrix} f(\vec{Z^0}) \\ f(\vec{Z^1}) \\ f(\vec{Z^2}) \\ \vdots \\ f(\vec{Z^k}) \end{bmatrix}$

Значения факторов регрессии, полученных на экспериментах возможно записать в виде матрицы $Z=\begin{bmatrix} Z_0^0 & Z_1^0 & Z_2^0 & Z_3^0 & Z_4^0 & \cdots & Z_n^0 \\ Z_0^1 & Z_1^1 & Z_2^1 & Z_3^1 & Z_4^1 & \cdots & Z_n^1 \\ Z_0^2 & Z_1^2 & Z_2^2 & Z_3^2 & Z_4^2 & \cdots & Z_n^2 \\ \vdots & & & & & & \vdots \\ Z_0^k & Z_1^k & Z_2^k & Z_3^k & Z_4^k & \dots & Z_n^k \end{bmatrix}$

Алгоритм машинного обучения с использованием линейной регрессии состоит в нахождении неизвестных параметров регрессии $\vec{V}$, имея матрицу Z и вектор $\vec{F}$.

Для обучение линейной регрессии, а именно нахождение $\vec{V}$ используются различные методы обучения, такие как метод максимального правдоподобия, метод наименьших квадратов, метод моментов.

Метод наименьших квадратов

Один из наиболее известных и часто используемых на практике методов оценки неизвестных параметров регрессии, метод наименьших квадратов — метод нахождения оптимальных параметров линейных регрессии, таких что сумма квадратов отклонений минимальна. Суть метода в минимизации расстояния $|a*\vec{V}-V|$ между векторами.

Метод максимального правдоподобия

Одним из наиболее универсальных и эффективных методов оценивания неизвестных параметров распределений является метод максимального правдоподобия, который и приводит к оценкам максимального правдоподобия Этот метод получил распространение в 20-е гг. XX в. благодаря работам английского статистика Р. Фишера (хотя у него были и предшественники). [4]

Идея заключается в том, чтобы выбрать коэффициенты $X=(x_{1,}x_{2,}\cdots,x_n)$ таким образом, чтобы максимизировать вероятность совместного появления результатов выборки $(y^{(1)},y^{(2)},\cdots,y^{(n)})$ [5]

Таким образом функция правдоподобия будет иметь вид:

$$L(y^{(1)},y^{(2)},\cdots,y^{(n)}|X)=p(y^{(1)}|X)*p(y^{(2)}|X)*\cdots*p(y^{(n)}|X)$$

Формула 1

В свою очередь метод максимального правдоподобия заключается в нахождение максимума функции правдоподобия.

Очевидно, что если максимум функции правдоподобия достигается точке $\grave{X}$ то эта точка также будет являться максимумом логарифма функции правдоподобия.

Методы на основе искусственных нейронных сетей

Существует достаточно много различных разновидностей нейронных сетей, но основные из них это рекуррентные сети, сети прямого распространения, самоорганизующиеся карты и радиально-базисные функции.

Традиционные нейронные сети прямого распространения (Feed Forward Back Propagation, FFBP) состоят из входного, выходного и промежуточных слоев: сигнал идет последовательно от входного слоя нейронов по промежуточным слоям к выходному. Классическим примером FFBP является многослойный перцептрон. Классификация документа в FFBP это подача признаков на вход нейронной сети, полученные на выходе значения сигналов и будут результатом классификации. Классический метод обучения данной сети – метод обратного распространения ошибки. При получении ошибки на выходе нейронной сети, сигнал распространяется обратно по ребрам нейронной сети, стремясь минимизировать.

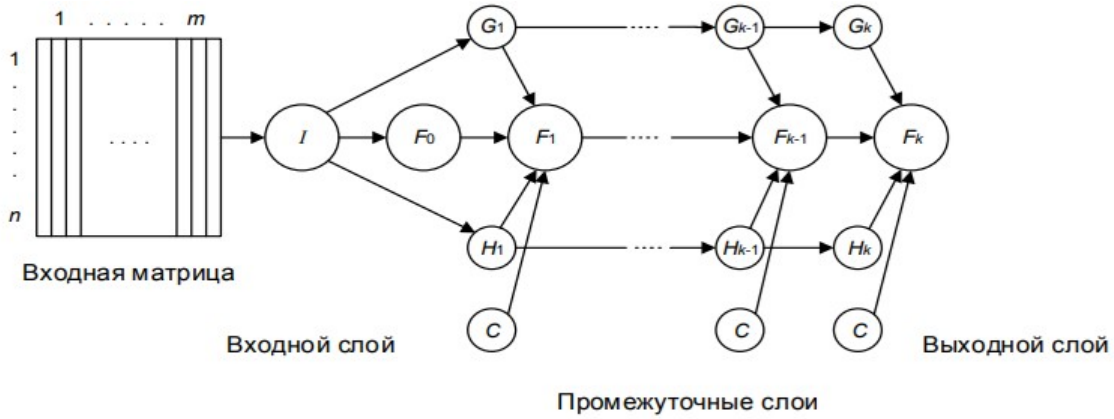

Рисунок 1 – Обобщенная схема нейронной сети с динамической архитектурой

Если количество промежуточных слоев нейронной сети не задано заранее, то данную архитектуру называют динамической. В таком случае будет

происходит создание слоев до тех пор, пока не будет достигнут необходимы уровень точности. Обобщенная схема DAN2 приведена на рисунке 1.

Элементы F_k являются функциями, которые содержат текущий элемент накопленных знаний (Current Accumulated Knowledge Element, CAKE), полученный на предыдущем шаге обучения. С это константы. Вершины G_k и H_k являются текущими нелинейными компонентами процесса, оставшиеся в остатке по передаточной функции нормализованной и взвешенной суммы входных сигналов (Current Residual Nonlinear Element, CRNE).

Сверточная нейронная сеть – это однонаправленная многослойная нейронная сеть с применением свертки. Свертка это операция при которой все подмножества входных данных умножается на ядро свертки поэлементно, а результат аккумулируется и записывается в соответствующую позицию на выходе. На рисунке 2 изображена обобщенная схема CNN.[6]

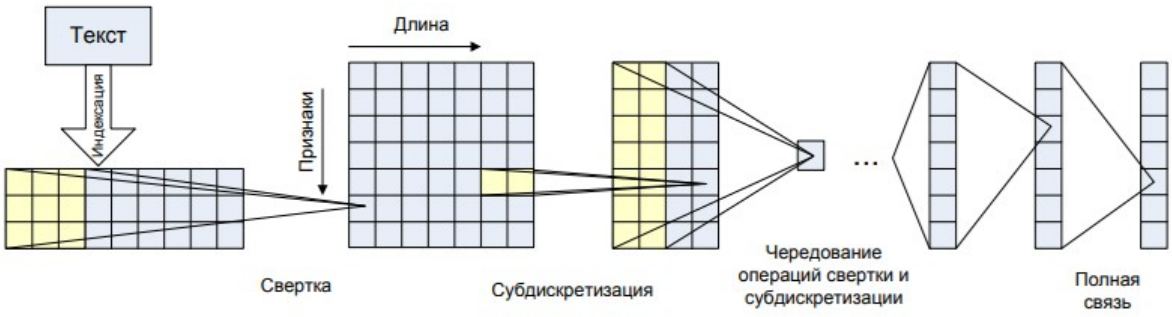

Рисунок 2 – Обобщенная схема сверточной нейронной сети

Рекуррентная нейронная это расширение многослойного перцептрона обратными связями. Сеть Элмана(рисунок 5), одна из наиболее распространенных типов рекуррентных нейронных сетей, имеет обратные связи не от выхода сети, а от выходов внутренних нейронов. Это позволяет добится сохранения предыстории наблюдаемых сигналов и накопления информации, способствующей

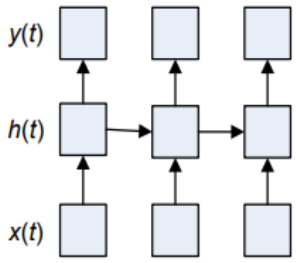

Рисунок 3 – Нейронная сеть Элмана(разновидность рекурентной сети)

правильному выбору стратегии обучения. Отличительной особенностью RNN является запоминание последовательностей сигналов.

Преимущества метода:

- при удачном подборе параметров алгоритма позволяет добиться высокого качества классификации информации;
- возможность инкрементного обучения.

Недостатки метода:

- вероятная расходимость или медленная сходимость, так как используемые градиентные методы нуждаются в больших объемах обучающей выборке;

Метод Байеса

Одним и самым популярным из вероятностных алгоритмов классификации является Метод Байеса (Naive Bayes, NB). Пусть P(ci|d) – вероятность того, что текст, вернее его отображение в вектор $d=(t_1,\ldots,t_n)$, классифицируется как c_i для $i=\overline{1\ldots|C|}$ Цель обучения данного классификатора заключается в том, чтобы выбрать такие значения c_i и d, при которых вероятность $P(c_i|d)$ достигает своего максимума:

$$CSV(D)=\arg\max_{c_i\in C} P(c_i|d)$$ *Формула 2*

Для вычисления значений $P(c_i|d)$ воспользуемся теоремой Байеса:

$$P(c_i|d)=\frac{P(c_i)*P(d|c_i)}{P(d)}$$ *Формула 3*

где $P(c_i)$ – априорная вероятность отнесения текста к категории c_i; $P(d|c_i)$ – вероятность найти текст, представленный кортежем $d=(t_1,\ldots,t_n)$, в категории c_i; $P(d)$ – вероятность того, что существует возможность представить в виде кортежа $d=(t_1,\ldots,t_n)$, произвольно взятый текст. По сути $P(c_i)$ является отношением количества текстов из обучающей выборки L,

отнесенных в категорию c_i , к количеству всех текстов из L . $P(d)$ не зависит от категории c_i , а значения $(t_1,\ldots,t_n)$, заданы заранее, поэтому знаменатель – это константа, не влияющая на выбор наибольшего из значений $P(c_i|d)$. Вычисление $P(d|c_i)$ представляет собой сложный процесс, в связи с большим количеством признаков $(t_1,\ldots,t_n)$, , поэтому допускают «наивное» предположение о том, что любые две координаты, рассматриваемые как случайные величины, статистически не зависимы. Следовательно возможно воспользоваться формулой:

$$P(d|c_i)=\prod_{k=1}^{n} P(t_k|c_i)$$

Формула 4

Далее все вероятности рассчитываются по методу максимального правдоподобия.

Преимущества метода:

- хорошее быстродействие;
- возможность обучать алгоритм инкрементно;
- относительно несложная программная реализация метода;

Недостатки метода:

- относительно низкое качество классификации
- невозможность классификации на основе сочетания признаков.

Метод опорных векторов

Одним из линейных способов классификации является метод на основе опорных векторов (Support Vector Machine, SVM). Данный способ классификации является одним из самых лучших в задачах классификации данных. Пусть дано множество тестов, подлежащих классифицированию. Определим биективное отображение данного множество во множество точек в пространстве размерности $|D|$.

Линейно разделимой называют выборку, которую можно разделить гиперплоскостью. Для классификации такой выборке достаточно провести плоскость так, чтобы по одну сторону оказались точки одного класса, по другую – точки другого класса.

В таком случае классификатором будет является уравнение гиперплоскости, то есть сопоставляя неизвестные элементы с данным уравнением, и определяя по какую сторону располагается отображенная точка, возможно сделать вывод о принадлежности объекта тому или иному классу.

Количество таких разбиений, в общем случае, бесконечно, но очевидно, что лучшим решением будет выбор плоскости, максимально удаленной от данных точек. В данном методе под расстоянием между прямой(гиперплоскостью) и множеством точек имеется ввиду расстояние между прямой и ближайшей к ней точкой множества. Метод SVM заключается в максимизации данного расстояния. Прямая, максимизирующая расстояние до двух параллельных прямых,

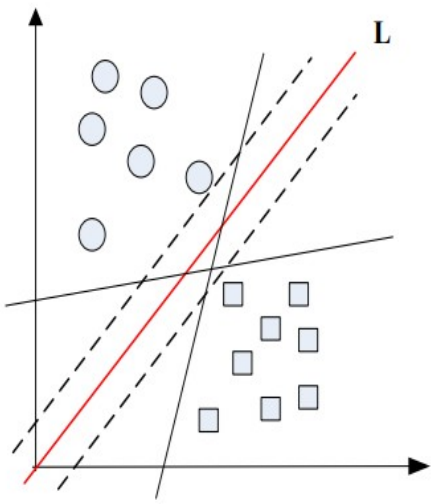

Рисунок 4 – Разделяющая гиперплоскость в методе опорных векторов

называется разделяющей (на рисунке 4 обозначена буквой L). Точки, ближайшие к параллельным гиперплоскостям называют опорными векторами, на рисунке 4 через них проходят пунктирные линии. Таким образом, данный алгоритм эксплуатирует допущение, что чем больше расстояние между данными гиперплоскостями, тем точнее будет работать полученный классификатор, так как увеличение расстояния между классами способствует более уверенной классификации.

В практических применения структура данных обычно неизвестна, и редко удается построить разделяющую гиперплоскость, а также невозможно обеспечить линейную разделимость выборки.

Существуют такие документы, которые алгоритм ложно-положительно отнесет к данному классу. Эти документы принято называть выбросами, они создают погрешность метода и их желательно игнорировать. Данная ситуация называется проблемой линейной неразделимости.

Для применения данного метода на линейно неразделимых множествах, необходимо от исходного пространства признаком к новому, таким образом, чтобы в этом новом пространстве обучающая выборка оказалась бы линейно разделимой, для этого скалярное произведение заменяется на некоторую функцию-ядро, данный переход позволит перейти к другому пространству признаков, где обучающие данные будут линейно разделимы. Таким образом проблема поиска разделяющей гиперплоскости сводится к задаче, эквивалентной поиску седловой точки функции Лагранжа с дополнительными условиями, такая система уравнений имеет простое решение, и это уже чисто вычислительная задача.

В данном случае говорят об алгоритме с мягким зазором (soft-margin SVM), и о жестком зазоре (hard-margin SVM) в линейно-разделимом случае.

Преимущества метода:

- является одним из качественнейших семейством алгоритмов;
- отсутствует необходимость в большом объеме обучающей выборки;

Недостатки метода:

- не очевидно, как понимать параметры алгоритма;
- низкая терпимость к выбросам во входных данных;

Предварительная обработка данных

Предобработка данных(data preprocessing) в задаче классификации текстов включает в себя ряд различных процедур, таких как токенизация,

удаление функциональных слов (семантически нейтральных слов, чаще всего это функциональные части речи). Следующей процедурой является морфологический анализ (разметка по частям речи и выделение основы слов). Эти процедуры приводят к редуцированию пространства признаков. После предобработки признаками будут являться все значимые слова и возможно их сочетания.

Индексация документов – процедура построения индекса, то есть удобного для обработки формата представления исходных данных.

Модель «мешка слов» (bag-of-words), к примеру, представляет документ многомерным вектором слов и их весов. То есть, каждый документ – это вектор в многомерном пространстве, координаты которого соответствуют индексам слов в «мешке», а координаты являются весом данного слова в документе.

Также существует распространенная модель индексации Word2vec, которая представляет каждое слово в виде вектора, содержащий информацию о сопутствующих словах.

Очевидно, что для обучающих и тестовых документов должен применяться один и тот же метод индексации.

Уменьшение размерности пространства признаков.

Быстродействие и вычислительная сложность алгоритмов классификации непосредственно зависит от размерности пространства признаков. Для эффективной работы классификатора целесообразно сократить количество признаков.

Также с уменьшением размерности пространства признаков уменьшается влияние эффекта переобучения.

Эффект переобучения — состояние алгоритма, при котором классификатор фокусируется на не значимых характеристиках учебных данных, а не на важности и значимости. Это приводит к тому, что классификатор показывает отличные результаты на обучающих данных и сильно деградирует

на тестовой выборке. Во избежания переобучения, размер обучающей выборки должен быть сопоставим с размером пространства признаков. Между размерностью пространства признаков и качеством классификации существует нелинейная взаимосвязь, например в некоторых случаях сокращение размерности пространства признаков в 10 или даже 100 раз может привести лишь к незначительной деградации качества классификации.

Существуют различные методы оценки веса признака документа. Одним из наиболее распространенных методов является вычисление TF*IDF. Основой оценки является отношение частоты появления слова в пределах документа к частоте его появления в других документах обучающей выборки.

Частота термина TF (term frequency) – оценка важности слова в пределах одного документа d, вычисляется по формуле:

$$TF = \frac{n_{t,d}}{n_d}$$ *Формула 5*

где $n_{t,d}$ – количество употреблений слова t в документе d; n_d – общее число слов в документе d. Обратная частота документа IDF (inverse document frequency) – показатель важности термина, определяет обратную величину к частоте вхождения данного термина в документах обучающей коллекции, таким образом уменьшает вес общеупотребительных слов:

$$IDF = \log\left(\frac{|D|}{D_t}\right)$$ *Формула 6*

где $|D|$ – количество документов в коллекции; D_t – количество документов, содержащих данный термин t.

Итоговый вес термина в документе относительно всей коллекции документов вычисляется по формуле:

$$V_{¿}t, d = TF * IDF$$ *Формула 7*

Очевидно, что формула 7 оценивает важность термина только со стороны вероятности его появления в документе и не затрагивает его сочетания с другими терминами.

Анализ специфичных особенностей предметной области

Произведя предварительный анализ данных запрещенных сайтов, было замечено, что некоторый конечный набор слов или фраз встречается в подавляющем большинтсве запрещенных данных.

Положим T_1 – множество всех текстов, принадлежащих к классу текстов содержащих запрещенную информацию.

Предложение – это единица языка, которая представляет собой грамматически организованное соединение слов (или слово), обладающее смысловой и интонационной законченностью. С точки зрения текста, предложение – некоторое последовательное подмножество слов, оканчивающееся некоторым, терминальным, знаком припинания.

Не нарушая общности, рассмотрим некоторый текст из

$$t_{1,i} \in T_{1,} w_j \in t_{1,i}$$

Где $w_{i,j}$ слово, встречающееся в тексте $t_{1,i}$ как минимум 1 раз. Текст $t_{1,i}$ можно представить в виде вектора.

$$t_{1,i} = (s_{1,} s_{2,} \ldots, s_k), s_m = (w_{i_1}, w_{i_2}, \ldots, w_{i_k})$$

Таким образом, мы определили предложение s_m

Положи $d = w_{1,1} \cup w_{1,2} \cup \ldots \cup w_{2,1}, w_{2,2} \cup w_{|T_1|,|W_i|} \setminus w_{1,1} \cup w_{1,2} \cup \ldots \cup w_{2,1}, w_{2,2} \cup w_{|T_2|,|W_i|}$ множе ство слов, встречающихся в запрещенных текстах и отсутствующих в некотором подмножестве разрешенных текстов.

Стоит отметить, что таким образом, множество не должно содержать синтаксически незначимых и нейтральных слов.

Мною была выдвинута следующая гипотеза, что так как большинство алгоритмов классификации текстов работает с разряженными данными, в случае класса запрещенных текстов, запрещенная информация плотно

размещается в некотором, небольшом, подтексте достаточно плотно. В качестве поддекстов возможно использовать синтаксические предложения, как самостоятельную часть языка.

Так как d – конечно, то взяв предложения s_m содержащие слова из множества d получим структуру, которую возможно использовать в качестве основы для построения признака.

Таким образом, применив один из классических алгоритмов, используя алгоритмы композиции, с алгоритмом направленым на поиск плотных данных в разряженом тексте, возможно увеличить качество работы прикладного классификатора.

Оценка качества классификации

Для обучения и оценки качества необходимы обучающая и тестовая выборка

$$\Omega = L \cup T$$

Прежде всего необходимо определить обучающую и тестовую выборки. Следующим шагом будет нахождения оптимальных признаков классификации и проверка их на тестовой выборке. Если находить оптимальные признаки по всей выборке, а потом оценивать качество, в таком случае отобранные признаки уже оптимизируют качество и оценка получиться оптимистичной. Для того, чтобы оценка качества работы алгоритма была объективной, необходимо определить соотношение обучающей и тестовой выборки. При выборе слишком маленькой обучающей выборки, оценка качества будет пессимистичной. При недостаточном объеме тестовой выборки, оценка будет неточна. Распространенная практика разделять обучающую и тестовую выборку в отношении 30% и 70%, то есть брать обучающую выборку в два раза больше тестовой.

Существуют более объективные способы оценки качества классификатора, например кросс-валидация. Принцип работы его заключается в

разбивание всего множества Ω на k частей, далее каждая по очереди выступает как тестовая. В данном способе большое значение имеет выбор k. Как правило, выбирают $k=5$ или $k=10$. Главным недостатком данного способа являются большие трудозатраты.

Точность и полнота – основные критерии оценки качества работы классификатора.

Точность (precision) классификации в пределах класса – это отношение количества текстов данного класса к количеству текстов, которых классификатор сопоставил с данным классом.

Полнота (recall) классификации – это доля найденных классификатором документов, действительно принадлежащих классу, относительно всех документов этого класса в тестовой выборке.

Полнота (recall) классификации – отношение количества текстов сопоставленных с данным классом к полному количеству текстов данного класса в пределах тестовой выборки.

Оценка качества работы классификатора производится на основе тестовой выборке также с привлечением эксперта(см. табл. 1).

Таблица 1:Оценка качества работы классификатора

Класс c_i		**Экспертная оценка**	
		Положительная	**Отрицательная**
Оценка системы	**Положительная**	TP	FP
	Отрицательная	FN	TN

В таблице приняты следующие условные обозначения:

- TP – true positive(истинно положительное решение);
- TN – true negative(истинно отрицательное решение);
- FP – false positive(ложно положительное решение);
- FN – false negative(ложно отрицательное решение).

Точность, по-определению, вычисляется следующим образом: $p=\frac{TP}{TP+FP}$

Полнота: $r=\frac{TP}{TP+FN}$

F-мера – характеристика более высокого порядка, основанная на полноте и точности: $F_B=\frac{(B^2+1)*pr}{B^2*p+r}$, где $0\le B<\infty$

При $0\le B<1$ точность имеет приоритет над полнотой.

При $B=1$ точность и полнота равноправны, тогда $F_B=\frac{2*p*r}{(p+r)}$

При $1<B<\infty$ решающий вес оценке придает полнота.

Также встречается другая формула для вычисления точности (accuracy). Также эту величину, часто называют правильностью или аккуратностью классификатора:

$$accuracy=\frac{TP+TN}{TP+TN+FP+FN}$$

Для сравнения алгоритмов классификации иногда могут использоваться специфические характеристики, например точка безубыточности, или сбалансированная точность.

Точка безубыточности (break even point, BEP) – заимствованная из экономики величина, при которой прибыль равна нулю, то есть доходы компенсируются расходами, и последующее увеличение реализации продукции будет приносить прибыль. В контексте классификации информации эта величина является оценкой качества алгоритма. Точка безубыточности вместе с F-мерой является комбинированной характеристикой точности и полноты.[7] .

Одной из характеристик алгоритмов классификации является быстродействие.

Быстродействие – время затрачиваемое алгоритмом на отнесение данного текста к какому-либо классу. Касательно задач классификации текстов под быстродействием понимается процессорное время, необходимое для

классификации. Измерение времени производятся на обучающей выборке для оценки скорости обучения, и на тестовой выборке для оценки быстродействия классификации. Высокие затраты на обучение оправдываются в дальнейшем, так как это обычно единоразовая процедура, в отличии от классификации. Очевидно, что увеличение точности приводит к снижению быстродействия алгоритма, а увеличение скорости обычно черевато понижением точности классификации.

Глава 3. Сбор информации

Важным требованием, для получения качественного классификатора, является хорошо подготовленная, репрезентативная выборка. Выборка – некоторый массив уже классифицированных данных.

В случае бинарной классификации, которой является классификация на запрещенную и разрешенную информацию, важным моментом является выбор источников данных. Оптимальным способом получения выборки является сбор данных web-страниц.

Для сбора информации необходимо решить следующие задачи:

- Сбор списка содержащих список заблокированных web-ресурсов.
- Преодолеть ограничения, наложенные интернет-провайдерами на доступ к запрещенным ресурсам.
- Подготовить ресурсы с разрешенной информации
- Собрать web-страницы и извлечь из них текстовые данные, а также отфильтровать лишнюю информацию.

Получения списка запрещенных web-ресурсов

Официальный сайт https://eais.rkn.gov.ru/, предоставляет выгрузку списка только оператором связи, имеющим квалифицированную электронно-цифровую подпись, выданную любым удостоверяющим центром, из числа аккредитованных Минкомсвязи России. Таким образом тсановиться невозможно использовать официальный источник. Однако подобную выгрузку можно получить со сторонних источников, одним из таких источников является сайт общественной организации РосКомСвобода. Выборка доступна в различных форматах, для дальнейшей работы будет использоваться удобный для парсинга формат csv.

Структура csv:

ip-адреса	Доменные	Url-страниц	Гос. Орган	Номер	Дата внесения

	имена			постановления	в реестр

Для обработки данных будут использоваться инструменты свободной СУБД Postgresql.

Для начала, необходимо создать таблицу и импортировать в нее данные

```
create table black_list(id SERIAL, ip TEXT, domain TEXT, url TEXT, dep TEXT, number TEXT, date TEXT);

COPY black_list FROM '/path/to/csv' WITH (FORMAT csv);
```

Для начала стоит разобрать органы, добавляющие в реестр:

```
select dep, count(*) as count from black_list group by dep order by count;
```

Таблица 2:Распределение количества заблокированных ресурсов по гос. органам

Гос. орган	**Количество записей**
Роспотребнадзор	366
Росалкогольрегулирование	609
ФСКН	1930
Минкомсвязь	3784
Мосгорсуд	5837
Роскомнадзор	9572
МВД	10176
Генпрокуратура	16863
суд	27459
ФНС	52789

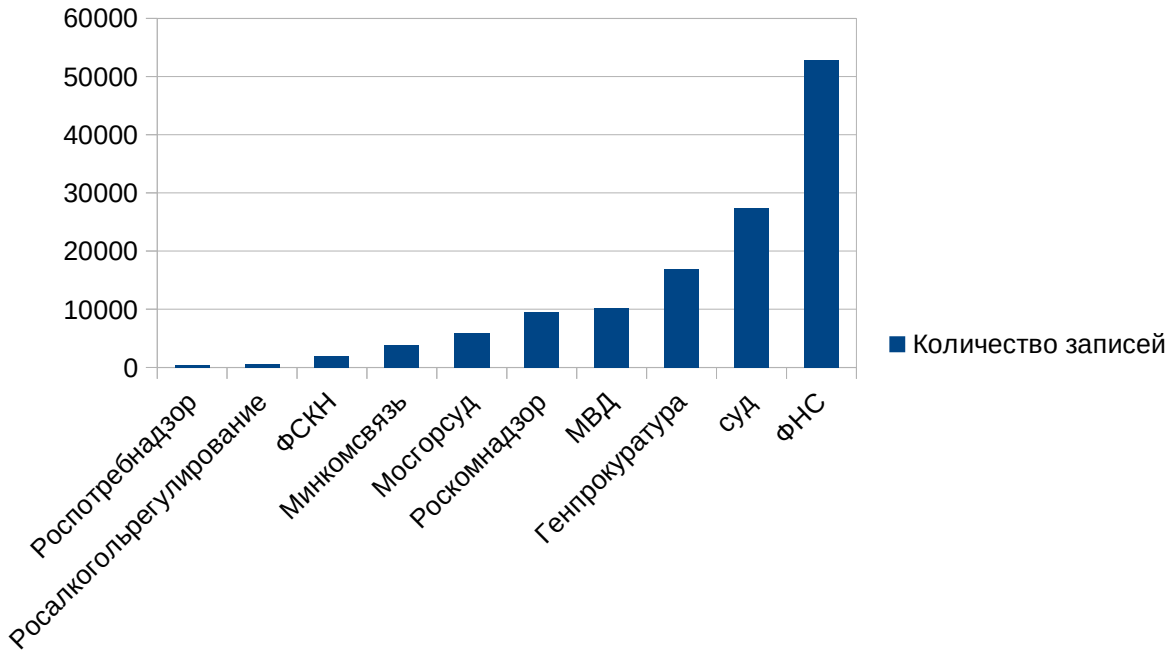

Рисунок 5 – Диаграмма распределений

Стоит отметить, что у различных органов различная юрисдикция, так, например основные цели блокировок ФНС(федеральной налоговой службы) это различные нелегальные букмекерские конторы и интернет-казино. При этом данные ресурсы не всегда содержат запрещенную информацию, возможно просто нарушение законодательства(например налогового).

Самые интересные данные для сбора, это адреса web-страниц(url), для получения их отфильтруем выборку:

```
DELETE FROM black_list where length(url) = 0;
```

Стоит заметить, что после такой обработки изменяться и распределения заблокированных ресурсов. Название государтсвенного органа также может служить признаком текста, который может быть использован для обучения, но так как для тестовой выборки по-идее он недоступен, то использование его в таком ключе бессмысленно, но можно использовать его для оценки вероятностей принадлежности к той или иной группе при предварительной обработке.

После применения фильтра статистика выглядит следующим образом:

Таблица 3:Отфильтрованное распределение количества заблокированных ресурсов по гос. органам

Гос. орган	Количество записей
Росалкогольрегулирование	94
Роспотребнадзор	366
ФСКН	1691
МВД	3451
Мосгорсуд	3696
ФНС	7190
Роскомнадзор	8323
суд	11709
Генпрокуратура	12019

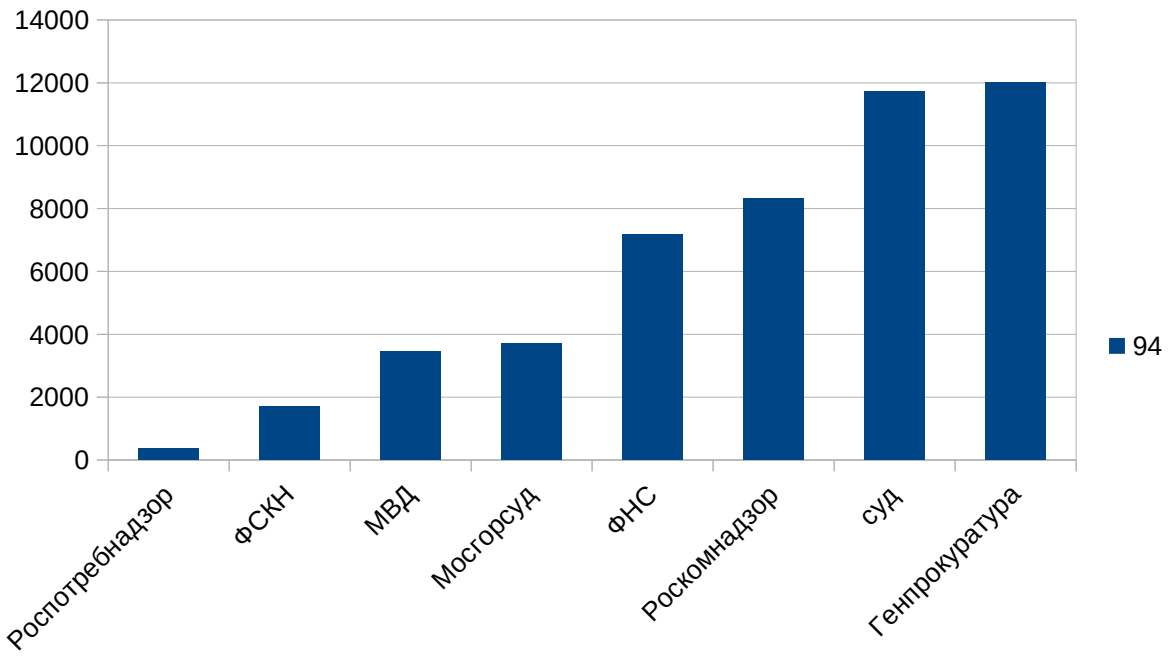

Рисунок 6 – Диаграмма распределений(отфильтрованные данные)

Таким образом, есть предположение, что задав не статичные вероятности (0, 1), а на основе данных о добавившем запись органе, возможно улучшение качества классификации. При обучении классификатора будет произведено сравнение на основе accuracy.

Подготовка списка разрешенных данных

Также для обучения классификатора необходимо собрать выборку, отнесенную к разрешенному классу информации. Для выборки возможно использовать различные источники, такие как новостные порталы, интернет энциклопедии, и тд. Для сбора информации с каждого ресурса требуется свой подход, но в общем случае его можно представить в следующем виде:

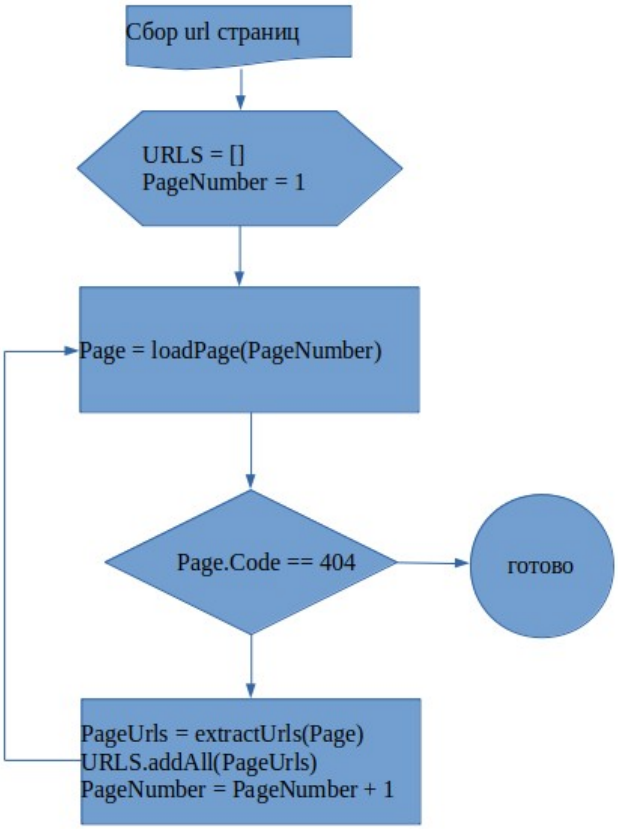

Рисунок 7 ЖБлок-схема

В качестве примера возможно рассмотреть получение списка url с новостного сайта mk.ru. В качестве инструмента для получения страниц и выполнения запросов к ним будет использоваться платформа Node.JS и Puppeteer

```
const fetchUrls = async (fromDate, toDate) => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  const loadPageUrls = async ({year, month, day}) => {
    await page.goto(`https://www.mk.ru/news/${year}/${month}/${day}/`);
    // ждем появления селектора, на случай ajax
    await page.waitForSelector('.news_list');
    return await page.evaluate(() => {
      // выполнение запроса в контексте страницы
      return [...document.querySelectorAll('ul.news_list.news_list_big > li > a')]
            .map(({href}) => href);
    })
  }
  // на сайте mk.ru пагинация работает через даты, формируем множество дат
  const range = Array.from(moment.range(fromDate, toDate).by('days'));
  const urls = uniqBy(flatMap(await Promise.all(range.map(momentToObject).map(loadPageUrls))));
  await browser.close();
  return urls;
};
```

Рисунок 8 – Получение списка url с сайта mk.ru

Аналогичный подход можно применить ко многим ресурсам, с разницей в способе формирование url и селектору ссылок. В блок схеме изображенной на рисунке 7 условием выхода является получение 404 статус-кода от web-сервера, что чаще будет избыточным, так как будут собраны все url, а при большом количестве страниц это создаст слишком большую выборку и вероятно получение блокировки от web-сайта из-за слишком большого трафика, поэтому целесообразней загружать диапазон страниц.

Сбор данных с web-страниц

Следующим шагом, после получения списка ресурсов является сбор с них текстовой информации. Если в случае url-ов разрешенных web-ресурсов задача является тривиальной, а именно сводится к выполнению селектора в теле документа, то для запрещенного контента сложность задачи сильно возрастает.

Основная сложность состоит в том, что большинство web-страниц содержат очень большое количество информационного шума, который негативно скажется на данных тестовой выборки.

Для решения данной проблемы были изучены различные информационные материалы на тему избавление страницы от информационного шума. Например в работе Криницкого А.И., Мартыненко Т.В. [8] был разработан алгоритм редукции информационного шума на основе

ключевых слов, переданных из поисковой системы. А в статье Кровецкого Александра был представлен алгоритм на основе машинного обучения с учителем в виде человека—эксперта, оценивающего важность тех или иных информационных блоков.[9] Очевидно, что эти алгоритмы не подходят для поставленной задачи. Позже было найдено решение – инструмент readability от компании arc90, он показал достаточную эффективность в удаление информационного шума, а также простую в использование работу с Puppeteer.

Для получения текстовой информации используется библиотека оптического распознавания символов tesseract и обвязка node-tesseract для Node.js.

Алгоритм получения текста web-страницы получился следующий:

1. Загружаем в headless-chrome через Puppeteer целевую web-страницу
2. Применяем readability;
3. Отправляем страницу в tesseract;
4. Получаем текст из tesseract.

```
const loadText = async (url) => {
  const page = await pagePool.acquire();
  const wordsPath = await wordsPool.acquire();
  const path = await filePool.acquire();
  try {
    await page.goto(url);
    // ждем появления селектора, на случай ajax
    const words = await page.evaluate(() => document.body.innerText.replace(/[^a-zA-ZA-Яа-я ]/g, '').replace(/ +/g, '\n').toLowerCase());
    page.evaluate(script => eval(script), script);
    await page.waitForSelector('#rdb-footer-wrapper');

    await writeFile(wordsPath, words);
    await page.screenshot({ path, fullPage: true });
    let brText = await page.evaluate(() => new Promise(resolve => {
      (function () {
        function selectText(element) {
          const selection = window.getSelection();
          const r = document.createRange();
          r.selectNodeContents(element);
          selection.removeAllRanges();
          selection.addRange(r);
          return selection.toString();
        }

        const text = selectText(document.body);
        window.setTimeout(() => resolve(text), 200);
      })();
    }));
    const text = mode === 'ocr' && await tesseract.recognize(path, {
      ...config, 'user-words': wordsPath
    });
    return mode === 'ocr' ? text : brText;
  }
  catch (e) {
    return null;
  }
  finally {
    wordsPool.release(wordsPath);
    filePool.release(path);
    pagePool.release(page);
  }
};
```

Рисунок 9 – Получение текстовой информации с web-страницы

Решение с использованием оптического было решено использовать в связи с тем, что readability манипулирует стилями для получения читаемой веб-страницы, а также возможность извлечения текстового содержания находящегося на изображениях, но в добавок был реализован и метод с получением видимого текста на web-странице(рис. 7).

Также в headless-chrome было установлено расширение adblock для удаления рекламных блоках, что также положительно сказалось на уменьшении информационного шума.

Получение запрещенной информации

В рамках данной работы в научно-исследовательских целях, потребовался доступ к запрещенной в РФ информации, и появилась задача в доступе в запрещенным ресурсам.

Провайдеры связи блокируют доступ к ресурсу из реестра запрещенных сайтов. Таким образом простой сбор информации становится невозможным.

Решить эту проблему возможно следующими способами:

- Прокси сервер за пределами РФ
- Выделенный сервер за пределами РФ

Фактически, эти способы отличаются лишь тем, где находиться конечный потребитель информации, в данном случае програмный клиент, собирающий обучающую выборку данных. В случае прокси-сервера этот клиент расположен в пределах РФ и не имеет доступа к сайтам из реестра.

В случае выделенного сервера, клиент расположен за пределами РФ и на него не распространяются ограничения провайдеров Российской Федерации.

В рамках сбора информации более эффективным является второй способ, так как информация не проходит через посредников, а также потому, что на выделенном сервере более широкий канал, поэтому это дает выигрыш во времени. Далее все процессы по сбору информации проходили с выделенного сервера, расположенного за пределами РФ.

При сборе информации возникла следующая проблема, большое количество ссылок в реестре запрещенной информации либо устарело, либо является фотографией без текстового содержания, а также ссылками на фильмы и музыку, что, очевидно, не годится для обучения классификатора. В процессе сбора данных, с целью улучшения качества собранной обучающей выборки, были произведены фильтрация url по расширениям, например фильтрация страниц, оканчивающихся на «.jgp», так как очевидно, что данные страницы представляют собой фотографии, а также ссылок содержащие некоторые ключевые слова, например «audio» или «film», это позволило отсеять часть неподходящей информации, также часть информации была отсеяна по ключевым словам в странице, таким как «404» или «фильм».

В дальнейшем произведена ручная обработка обучающей выборки, что дало в общей сумме порядка 600 текстов для дальнейшего обучения.

Глава 4. Программная реализация классификатора

В данной главе будет рассмотрена реализация классификатора, а также инструментов с помощью которых будет реализован данный классификатор.

На данный момент машинное обучение является одной из самых быстроразвивающихся областей компьютерных наук, для него разработано множество различных программных и даже аппаратных инструментов.

Предварительный список инструментов, которые планируется использовать для построения классификатора:

- Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода, наиболее популярный инструмент в области машинного обучения.
- NumPy – библиотека для python, предназначенная для поддержки работы с тензорами и алгоритмами линейной алгебры.
- scikit-learn – высокоуровневая библиотека реализованных инструментов машинного обучения, в том числе различных классификаторов.
- Keras – высокоуровневый API нейросетей, предоставляющий библиотеку глубокого обучения для Python. Это один из лучших инструментов для тех, кто начинает свой путь в качестве специалиста по машинному обучению.С ним могут работать такие популярные фреймворки Python, как TensorFlow, CNTK или Theano.
- TensorFlow – это нейронная сеть, которая учится решать задачи путем позитивного усиления и обрабатывает данные на различных уровнях (узлах), что помогает находить верный результат.

Условно разработку классификатора можно разбить на несколько этапов: подготовку данных, выбор и подбор параметров классификатора.

Подготовка данных

Большое значение для задач классификации текстов имеет предварительная обработка выходных данных. Сырые тексты неудобны в обработке классическими алгоритмами классификации. Для обработки текстов необходима его токенизировать. Токен – последовательность символов, соответствующая лексеме, лексема это понятная для транслятора последовательность символов. В задачах обработки естественных языков имеет значение в первую очередь слова и знаки препинания.

Для построения токенизатора удобно и логично использовать модель автомата с магазинной памятью. Данный автомат находясь в некотором состоянии может выполнять действия на основе вершины стека, и входящего символа, а также может выполнять манипуляции со стеком.

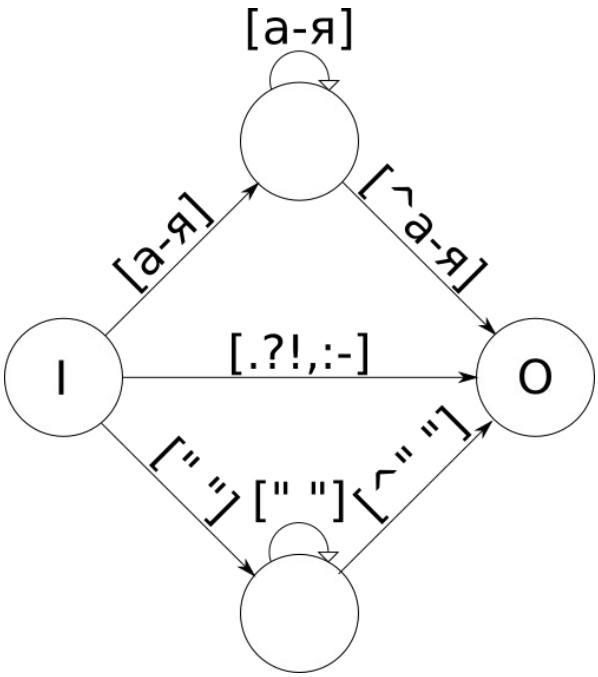

Рисунок 10 – Упрощенное изображение автомата

Диаграмма переходов в данном автомате представляет собой условные переходы для токенов типа слово, разделитель и знак препинания. Данная схема токенизации текста в будущем позволит упростить манипуляции с текстом и создаст основу для построения численной модели представления текста

В высокоуровневом языке программирования Python существует удобный инструмент для программирования автоматов – генераторы.

```
def tokenize(chars):
    state = State.uninitialized
    state_data = None

    def make_token(state, before_state_data):
        if state == State.word:
            yield Token(token_type=TokenType.word, data=before_state_data)
        if state == State.delimiter:
            yield Token(token_type=TokenType.delimiter, data=before_state_data)
        if state == State.punctuation:
            yield Token(token_type=TokenType.punctuation, data=before_state_data)

    for c in chars:
        for new_state, new_state_data, emmit in state_processor[state](c, state, state_data):
            before_state = state
            before_state_data = state_data
            if before_state != new_state or emmit:
                yield from make_token(before_state, before_state_data)
            state = new_state
            state_data = new_state_data
    yield from make_token(state, state_data)
```

Рисунок 11 - Токенизация текста

Все строковые объекты являются итерируемыми, следовательно создать возможно создать токенизатор работающий с потоком символов, что позволяет эффективно использовать аппаратные ресурсы.

Следующем этапом в обработке текста является извлечение основы слов, после токенизации является извлечение основы слов или стемминг. В общем случае стемминг является нетривиальной задачей, но существуют готовые решения для стемминга, например производительный snowball, реализующий алгоритм для распространенных европейских языков, но данный алгоритм показывает плохие результаты для русского языка, показывая разные основы для слов с общим корнем. Для русского языка существует морфологический анализатор pymorphy2, который в рамках проекте AOT реализует морфологический анализ слов, также он имеет функцию нормализации слов, что возможно использовать для стемминга, а также предсказание части речи, что может быть использовано для фильтрации служебных частей речи.

Для проверки предположения о том, что запрещенная информация содержится в ограниченных фразах содержащих небольшой набор слов, потребуется написать собственный алгоритм векторизации текста.

Рейтинг слова(RW) – разность количества вхождения данного слова в колеции соответствующие разным классам. Таким образом:

$$RW = N_D(W,0) - N_D(W,1)$$

Где N количественная функция показывающая количество вхождений данного слова в обучающую выборку данного класса.

В результате такой обработки каждому слову сопоставляется его рейтинг, наименьший рейтинг получают слова содержащиеся в запрещенных текстах, отсортировав полученный мешок слов, и взяв некоторое количество K слов получим список «плохих» слов BW , то есть слов, чье наличие в тексте свидетельствует о возможной запрещенной информации. Есть предположение, что манипулируя числом K , возможно эмпирическим путем менять число ложно-положительных срабатываний классификатора.

В дальнейшем имеет смысл перейти от работы со словами к работе с предложениями, упрощенно, предложение это последовательность слов, оканчивающаяся определенным знаком препинания.

Для начала необходимо дать определение фразе. Фраза – последовательность слов в пределах предложения. Делая допущение о том, что фраза, содержащая в себе запрещенную информацию содержит каждое слово один раз, возможно создать векторизатор.

Пусть M некоторое максимальное расстояние между словами в некоторой фразе. Тогда:

$$V = (a_{1,}a_{2,}\dots,a_K)$$

Где a_i индекс i-ого слова в данной фразе, при a_i<0 понимаем, что данного слова не содержится в данной фразе. Таким образом сам алгоритм векторизации будет способствовать выделению последовательностей слов, и позволит значительно сократить размерность пространства признаков, таким

образом сняв часть ответственности алгоритима классификации. Пример векторизации с использованием данного алгоритма:

$BW = [купить, наркотик]$

$M = 2$

$T =$ в городе москве купить недорого наркотик

$PHRASE = купить\ недорого\ наркотик$

$V = (0, 2)$

Очевидно, что в некотором тексте будут встречаться несколько запрещенных фраз. Для работы классификатора необходимо получить один вектор признаков, в некотором фиксированном пространстве. Самым простым решением для этого будет использование некоторой функции-оценки данного вектора и выбор вектора соответствующего максимальной оценки. Так как вектор представляет собой индексы слов, то возможно использовать меру разброса – дисперсию для оценки вектора, предварительно убрав отрицательные индексы или, например, коэффициент эксцесса. Применение дисперсии покажет фразы с наименьшим разбросом слов, в то время коэффициент эксцесса покажет вектор с наиболее плосковершинным распределением, то есть в контексте фраз означает большое количеством слов в совокупности с их взаимным расположением.

Выбор алгоритма классификации

После приведенных преобразований, мы сфокусировали условную вероятность между признаком – классом и множеством входных признаков, так как изначально оценили слова по нахождению их в том или ином классе, таким образом алгоритмы слабо учитывающие взаимную вероятность признаков могут дать лучшую оценку, такие как классификатор байеса или его модификация – квадратичный дискриминант.

Тестирование и анализ алгоритма классификации

Тестирование классификатора и сравнение результатов, с другими, широко используемыми алгоритмами классификации позволит понять эффективность классификатора и дать представление о его дальнейшем развитии.

В целях тестирования вся исходная выборка поочередно разбивалась на обучающую и тестирующую выборку и отправлялась в классификатор, далее оценивалась аккуратность метода.

Таблица 4: Результаты испытаний классификатора

dt(tf-idf)	0,87	0,87	0,86	0,91	0,87	0,87
mlp(tf-idf)	0,91	0,92	0,91	0,92	0,92	0,91
rf(tf-idf)	0,90	0,89	0,92	0,88	0,74	0,86
gpc(tf-idf)	0,86	0,86	0,93	0,93	0,92	0,90
svc(sp[max_distance=1])	0,87	0,89	0,88	0,85	0,85	0,87
svc(sp[max_distance=2])	0,82	0,77	0,79	0,88	0,82	0,82
svc(sp[max_distance=4])	0,80	0,84	0,89	0,86	0,85	0,85
nb(sp[max_distance=1])	0,88	0,96	0,92	0,91	0,89	0,91
nb(sp[max_distance=2])	0,88	0,88	0,89	0,92	0,82	0,88
nb(sp[max_distance=4])	0,88	0,87	0,86	0,85	0,82	0,85
gpc(sp[max_distance=1])	0,88	0,88	0,85	0,83	0,83	0,85
gpc(sp[max_distance=2])	0,85	0,80	0,71	0,72	0,83	0,78
gpc(sp[max_distance=4])	0,75	0,70	0,77	0,72	0,80	0,75
mlp(sp[max_distance=1])	0,86	0,78	0,86	0,84	0,87	0,84
mlp(sp[max_distance=2])	0,70	0,83	0,83	0,86	0,72	0,78
mlp(sp[max_distance=4])	0,88	0,75	0,79	0,84	0,84	0,82
qda(sp[max_distance=1])	0,93	0,88	0,83	0,92	0,91	0,90
qda(sp[max_distance=2])	0,90	0,90	0,87	0,82	0,82	0,86
qda(sp[max_distance=4])	0,83	0,88	0,78	0,88	0,87	0,85

dt(sp[max_distance=1])	0,74	0,74	0,77	0,70	0,68	0,73
dt(sp[max_distance=2])	0,74	0,76	0,74	0,83	0,74	0,76
dt(sp[max_distance=4])	0,82	0,83	0,77	0,82	0,77	0,80

Произведя анализ данной таблицы, можно прийти к выводу, что данный классификатор показывает не уступает в аккуратности классическим классификаторам на основе tf-id, но при этом обладает значительно меньшем пространство признаков, что в общем ограничивает применимость данного классификатора, например он будет менее эффективен для классификации настроения текста, но будет эффективен в похожих задачах.

Данный алгоритм показывает наилучшее качество при использовании с наивным классификатором Байеса, а нетребовательность алгоритма к памяти, за счет намного меньшего объема признаков уменьшает вычислительную сложность алгоритма. Таким образом можно считать, что для задачи поиска запрещенной информации данный алгоритм показал хороший результат.

Заключение

Активное участие в развитии информационных технологий – одна из последних тенденций в законодательной сфере РФ. Развитие механизмов ограничения доступа граждан к противоправной информации, одно из самых значительных и спорных направлений. На данный момент реестр содержит более 140000 записей, не считая доменов и ip-адресов.

Большое количество записей показывает, что противозаконная информация достаточно распространена в русскоязычном сегменте всемирной паутины. Одним из перспективных направлений в данной области это применение инструментов машинного обучения для поиска противозаконной информации. Специфика области создает прямую угрозу доступности web-ресурсов. Средства машинного обучения могут использоваться как в качестве инструмента проверки пользовательского контента, так и в составе активных поисковиков противозаконной информации.

Произведя анализ запрещенной информации была выдвинута гипотеза, о структуре такой информации и одной из целей данной работы являлось проверка данной гипотезы. Для проверки гипотезы были разработаны специальные программные средства, а также произведен ручной отбор информации. В процессе был создан и протестирован алгоритм векторизации текстов на основе рейтинга слов.

Данный алгоритм показал хорошие результаты, при существенно меньших вычислительной сложности. В будущем это позволило бы реализовать поисковую систему для запрещенной информации, с целью последующей проверки правоохранительными органами.

Было реализовано программное обеспечение для сбора информации с web-страниц с использованием средств автоматизации, а также по для классификации, которое поможет предотвращать противоправные действия.

Разработанный алгоритм классификации, хоть и показал на тестах результаты не хуже, чем классические алгоритмы все же нуждается в более глубоком изучении, например проанализировать методы оценки векторов и подборе оптимальных параметров, также возможно включение в результирующий вектор значений рейтинга слов, также обобщение на номинальную классификацию.

Таким образом, в рамках данной работы был разработан и протестирован классификатор, и система сбора информации. Была выдвинута и подтверждена гипотеза и можно считать все поставленные задачи выполненными.

Список литературы

1: , Задача классификации,

2: В.В.Вьюгин, МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИМАШИННОГО ОБУЧЕНИЯ ИПРОГНОЗИРОВАНИЯ, 2013

3: , Probabilistic interpretaton of classical Machine Learning models, 2017

4: Ивченко Г.И., Медведев Ю.И., Введение в математическую статистику., 2010

5: Виктория Федотова, Введение в машинное обучение,

6: Ghiassi M., Olschimke M., Moon B., Arnaudo P., Automated text classification using a dynamic artificial neural network
model, 2012

7: Zhang X., Zhao J., Character-level Convolutional Networks for Text Classification,

8: Tarasov D. S., Deep Recurrent Neural Networks for Multiple Language Aspect-Based Sentiment Analysis. ,

9: Ju R., Ju R. An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis, 2015

10: Поляков Игорь Викторович, Соколова Татьяна Владимировна, Чеповский Александр Андреевич, Чеповский Андрей Михайлович, ПРОБЛЕМА КЛАССИФИКАЦИИ ТЕКСТОВ И ДИФФЕРЕНЦИРУЮЩИЕ ПРИЗНАКИ,

11: Yang Y., An evaluation of statistical approaches to text categorization,

12: Криницкая А.И., Мартыненко Т.В., Разработка адаптационного алгоритма очистки веб-страниц от информационного шума,

13: Краковецкий А., Очищаем веб-страницы от информационного шума, 2009