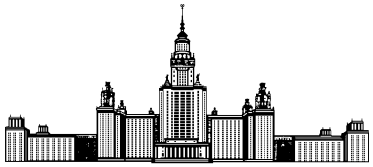
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра алгоритмических яхыков

КУРСОВАЯ РАБОТА СТУДЕНТА 324 ГРУППЫ

Тематическая классификация текста на основе нейронных сетей

Выполнил:

студент 3 курса 324 группы

Яндутов Алексей Владимирович

Научный руководитель:

д.т.н., в. науч. сотр. ВЦ РАН

Лукашевич Наталья Валентиновна

Москва, 2020

Содержание

1 Введение

Классификация текстов некоторой коллекции – отнесение каждого текста естественного языка к тематическим категориям(классам) из предопределённого набора. В последнее время данная задача осбенно актуальна, так как наблюдается значительный рост числа текстов и сложных документов, для которых требуется эффективно решать задачу тематической классификации в автоматическом режиме.

Задача построения автоматического тематического классификатора возникает во многих областях, где требуется работать с большим количеством данных, и ручная обработка экспертами невозможна в силу ограниченности человеческого ресурса. Примером таких областей могут служить:

- поисковые и рекомендательные системы
- системы документооборота
- фильтрация информации(например детектирование спама)

В течение долгого времени среди основных подходов к решению данной задачи доминировали традиционные методы машинного обучения, такие как метод опорных векторов и найивный байесовский классификатор. Однако данные модели не позволяют достичь глубокого понимания текстовой информации и учитывать многие особенности языка. В последнее время был достигнут некоторый успех в направлении обработки естественного языка в переходе от таких линейных моделей, действующих в разряженных пространствах с высокой размерностью, к нелинейным моделям нейронных сетей, использующих плотные векторные представления слов. На данный момент самые выдающиеся результаты в решении данной задачи были достигнуты при помощи моделей глубокого обучения.

В данной работе рассматриваются некоторые традиционные алгоритмы машинного обучения и основные нейросетевые модели. Кроме того, проводятся вычислительные эксперименты, показывающие эффективность работы нейронных сетей по сравнению с линейными методами машинного обучения на датасете Web of Science-11967. Построены классификаторы, сравнимые по точности на указанном датасете с результатами статьи[2].

2 Постановка задачи

В работе решается задача многоклассовой тематической классификации. [14] Задано конечное множество классов и множество объектов, для конечного подмножества которых известно, к какому классу они относятся. Такое подмножество называется обучающей выборкой. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества, то есть способный поставить каждому объекту в соответствие метку класса, к которому принадлежит данный объект.[6]

Формальная постановка задачи:

$D = \{d_1, d_2..., d_{|D|}\}$ - множество текстовых документов.

$C = \{c_1, c_2, ..., c_{|C|}\}$ - конечное множество меток классов.

Существует $y^* : D \to C$ - неизвестная целевая функция, значения которой известны только на объектах конечной обучающей выборки $D^m = \{(d_1, y_1), ..., (d_m, y_m)\}$.

Требуется построить алгоритм $a : D \to C$, аппроксимирующий целевую функцию y^* по заданной метрике.

3 Набор данных

Эксперименты проводилсь на наборах данных Web of Science и Reuters, представленные текстами на английском языке.

3.1 Web of Science

Web of Science содержит тексты из научных статей из следующих разделов: компьютерные науки, электротехника, психология, машиностроение, строительство, медицина и биохимия. Данный датасет доступен в трех вариациях: WOS-5736, WOS-11967, WOS-46985, где число в названии характеризует размер набора.

В силу оптимальности размера датасета был выбран WOS-11967, который включает 33 класса. Тексты в данном наборе имеют длину от 24 до 988 слов. Для обучающей выборки было взято 80%, для тестововй соотвественно 20% от общей выборки.

3.2 Reuters

Данный набор содержит тексты из новостных лент крупного международного агентства новостей и финансовой информации Reuters. Для эксперимента выбрана версия датасета Reuters-21578, которая содержит 10788 документов и 90 классов. Размеры категорий имеют сильный дисбаланс. Тексты имеют длину от 2 до 1861 слов. Набор данных в исходном варианте разбит на обучающую(7769 документов) и тестовую(3019) выборки.

4 Традиционные методы машинного обучения для классификации текстов

Решение задачи классифкации текстов состоит из следующих этапов:

- Предварительная обработка текстов
- Перевод текстов в вещественное пространство признаков, где каждому документу будет сопоставлен вектор фиксированной длины
- Выбор алгоритма машиннного обучения и обучение классификатора

В данной работе рассмотрены алгоритмы: наивный байесовский классификатор и метод опорных векторов.

4.1 Предварительная обработка текстов

Предобработка текстов имеет большое влияние на эффективность традиционных методов машинного обучения.

Все естественные языки содержат большое количество слов, которые не несут полезной информации о тексте, в контексте которых они находятся. Такие слова называются стоп-словами. Примером в английском языке могут служить артикли, в русском языке предлоги, частицы и союзы. Кроме того тексты содержат пунктуационные знаки, цифры и прочие символы, которые могут зашумлять признаковое пространство и отрицательно влиять на эффективность классификатора.

Вторым этапом в предобработке является приведение каждого слово к некоторой основе, одинаковой для всех его грамматических форм. Это обосновывается тем, что слова, несущие один и тот же смысл, могут быть записаны в разной форме. Распространены следующие подходы:

- Стемминг - процесс нахождения основы для заданного исходного слова путем отсечения от слова окончаний и суффиксов таким образом, чтобы оставшаяся часть была одинаковой для всех грамматических форм слова.

- Лемматизация - процесс приведения словоформы к её нормальной(словарной) форме

При построении классификатора были использованы оба подхода, однако второй дал чуть более высокие результаты, поэтому все результаты, описанные ниже, представлены для лемматизации.

4.2 Перевод текста в векторное представление

Алгоритмы машинного обучения ориентированы на работу с признаковым описанием объектов, поэтому все тексты обычно переводят в вещественное пространство признаков. Наиболее известные способы, позволяющие осуществить данную операцию, основаны на статистической информации о словах. В работе применены следующие методы для различных $ngram$ при n = 1, 2, 3:

- Частотное кодирование - один из самых простых способов извлечения признаков из текста. Длиной вектора, сопоставляемого документу, является размер мешка слов коллекции, а на позиции i этого вектора стоит частота i-ого слова в данной документе.

- TF-IDF - статистическая мера, используемая для оценки важности в контексте документа и всей коллекции.

  $\text{TF-IDF}(d,t) = \text{TF}(d,t) \cdot \text{IDF}(d,t)$

  $\text{TF}(d,t) = \frac{n_t}{|d|}$ - частота слова, оценивающая важность слова t в пределах документа d, n_t - число вхождений слова t в данный документ.

IDF$(d,t) = log(\frac{N}{df(t)})$ - обратная частота документа, уменьшает TF-IDF в широко употребляемых в корпусе словах, где df(t) - число документов, в которых присутствует данное слово, N - размер корпуса.

При решении данной задачи наилучшим образом показали себя те модели, которые выделяли из текста последовательности из трех слов, оставляя только те, которые присутствуют в менее, чем $\frac{1}{2}N$ документах.

4.3 Наивный байесовский классификатор

В байесовском подходе предполагается, что обучающие объекты и ответы на них $\{(d_1, y_1), \ldots, (d_m, y_m)\}$ независимо выбираются из некоторого распределения $p(d, y)$, заданного на множестве D×C Данное распределение можно переписать как

$$p(d, c) = p(c)p(d|c)$$

По формуле Байеса можно записать апостериорное распределние на множестве ответов:

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)}$$

Данный алгоритм использует допущение о независимости признаков, поэтому ответом классификатора будет:

$C_{MAP} = \arg\max_{c \in \mathcal{C}} p(d|c)p(c) = \arg\max_{c \in \mathcal{C}} p(x_1, ..., x_n|c)p(c) = \arg\max_{c \in \mathcal{C}} p(x_1|c)...p(x_n|c)p(c)$

4.4 Метод опорных векторов

Традицонная версия метода решает задачу бинарной классифкации. Пусть $\mathbb{X} = R^n$ - пространство объектов, $Y = \{-1, 1\}$ - множество допустимых ответов. Линейная модель классификации определяется следующим образом:

$$a(x) = sign(\langle w, x \rangle + w_0)$$

где $w \in \mathbb{R}^n$ - вектор весов, $w_0 \in \mathbb{R}$ - сдвиг.

Геометрически данный линейный классификатор соответствует гиперплоскости с вектором нормали w. Величина скалярного произведения $\langle w, x\rangle$ пропорциональна расстоянию от гиперплоскости до точки x. Таким образом линейный классификатор разделяет пространство объектов на две части с помощью гиперплоскости, и при этом одно полупространство относит к положительному классу, а другое - к отрицательному.

Метод опорных векторов с линейным ядром строит классификатор, линейно разделяющий обучающую выборку, и при этом имеющий максимальный отступ объектов выборки от разделяющей гиперплоскости. То есть, в случае разделимости выборки решается следующая оптимизационныя задача:

$$\begin{cases} \frac{1}{2}\|w\| \to \min\limits_{w,b} \\ y_i(\langle w, x_i\rangle + b) \geq 1, \ i = 1, \ldots, m \end{cases}$$

Для решения задачи многоклассовой классификации в работе был использован метод "один против всех". Пусть каждый объект относится к одному из K классов: $Y = \{1, ...K\}$. Обучим K классификаторов $b_1(x), ..., b_K(x)$, каждый из которых будет отличать i-й класс от всех остальных, то есть решать бинарную задачу принадлежности объекта к классу i. Итоговый классификатор будет выдавать класс, соответствующий самому уверенному из бинарных алгоритмов.

$$a(x) = \arg \max_{i \in \{1,...,K\}} b_i(x)$$

4.5 Проблемы традиционных подходов

- высокая размерность пространства
- большой объем данных
- раряженность пространства
- необходимо извлечь из текста правильные признаки

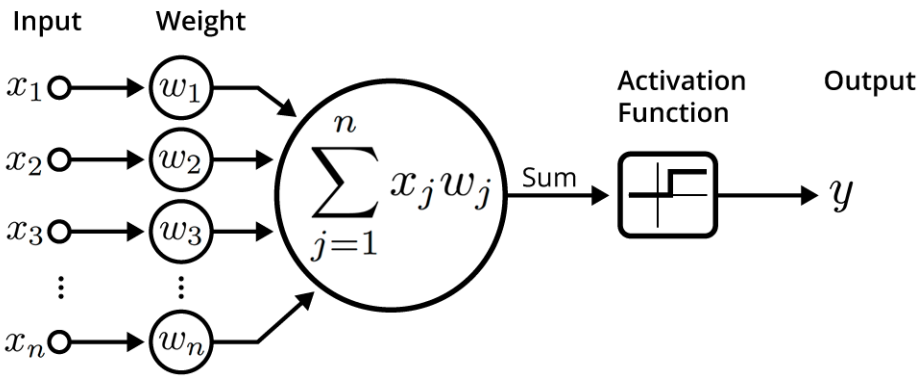


Рис. 1: Искусственный нейрон

5 Нейросетевые методы

Понятие искусственной нейронной сети было предложено У. Маккалоком и У.Питтсом в 1943 год. В частности, ими была предложена модель искусственного нейрона, которая строится следующим образом. Нейрон получает входные сигналы(сигналы других нейронов сети либо сигналы, поступающие на вход сети), умножает их на вес, соответствующий каждому из каналов, суммирует и применяет нелинейную функцию активации. Таким образом вычисляется величина активации нейрона, которая далее подается на выход.

На рис.1 приведена модель искусственного нейрона.

5.1 Функции активации

Существует множество видов нелинейных функций активации.В настоящее время нет устоявшегося подхода относительно того, какую нелинейность применять в каких условиях, поэтому выбор правильной нелинейности для задачи является по большей части эмпирическим вопросом[1]. В данной работе в нейронных моделях были использованы следующие функции активации:

- Сигмоида: $f(x) = \frac{1}{1+e^{-x}}$ - представляет собой S-образную функцию, преобразующую каждое значение аргмента в диапазон [0, 1]. Данная функция была канонической нелинейностью для нейронных сетей с момента их создания, однако в настоящее время считается устаревшей для использования в внутренних слоях нейронной сети.

- ReLU: $f(x) = max(0, x)$ - несмотря на простоту, хорошо справляется со многими задачами.

- Softmax: $f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$ для $i = 1, \ldots, N$ - используется на выходе сети для моделирования распределения вероятностей по возможным выходным классам.

5.2 Функция потерь

Пусть $y_i = a(x_i)$ - предсказание нейронной сети, y_i^* - истиннный результат целевой функции на обучающем объекте x_i. При обучении нейронной сети, как и при обучении линейного классификатора, строится функция потерь $L(y, y^*)$, показывающая ошибку предсказания алгоритма на некотором объекте. Формальной целью обучения является минимазация потерь на объектах обучающей выборки.

При решении задачи многоклассовой классификации на выходе нейронной сети необходимо получить вероятность принадлежности объекта каждому из классов. В этом случае в качестве функции потерь обычно используется кросс-энтропия:

$$L_{cross-entropy}(y_i, y_i^*) = -\sum_{j=1}^{K} y_{ij}^* \log y_{ij}$$

где K - количество классов в задаче.

В процессе обучения нейронная сеть при помощи метода обратного рапространения ошибки настраивает веса нейронов W, минимизируя функцию потерь.

В данной работе для классификации текстов с помощью нейронных сетей используется описанная выше функция потерь.

5.3 Сеть прямого распространения

Нейроны, связанные друг с другом, образуют сеть: выход нейрона может поступать на вход одного или нескольких нейронов. Такая сеть, в которой нет циклов, и все нейроны предыдущего слоя связаны с нейронами следующего, называется сетью прямого распространения. Было показано, что даже сеть с одним скрытым слоем с сигмоидной функцией и линейной функцией на выходном слое способна приблизить с любой точностью любую непрерывную функцию.

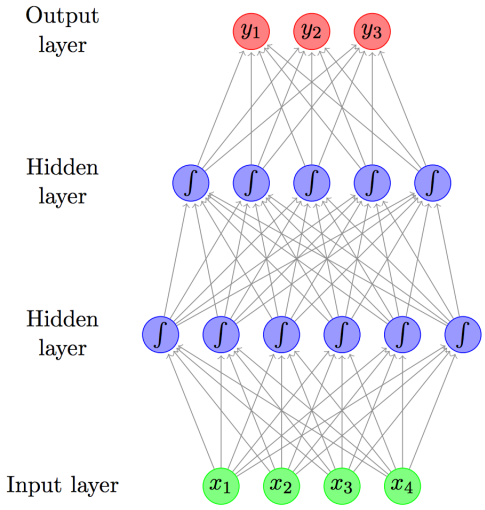


Рис. 2: Сеть прямого распространения с двумя скрытыми слоями

Типичная нейронная сеть с прямой связью может быть изображена так, как на рисунке 2. Каждый круг представляет собой нейрон, причем входящие стрелки являются входами нейрона, а исходящие стрелки - выходами нейрона. Нижний слой не имеет входящих стрелок и является входом в сеть. Самый верхний слой не имеет исходящих стрелок и является выходом сети. Другие слои считаются скрытыми. Слой, в котором каждый нейрон соединен со всеми нейронами на предыдущем уровне, называется полносвязным.

5.4 Сверточная нейронная сеть

Сверточная нейронная сеть обычно представляет собой чередование сверточных слоев (convolution layers), субдискретизирующих слоев (subsampling layers) и при наличии полносвязных слоев (fully-connected layer) на выходе[6]. Все три вида слоев могут чередоваться в произвольном порядке.

В сверточном слое нейроны, которые используют одни и те же веса, объединяются в карты признаков (feature maps), а каждый нейрон карты признаков связан с частью нейронов предыдущего слоя. При вычислении сети получается, что каждый нейрон выполняет свертку некоторой области предыдущего слоя (определяемой множеством нейронов, связанных с данным нейроном).

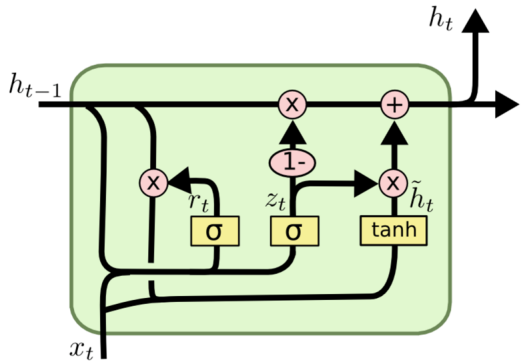


Рис. 3: Ячейка сети GRU

Слои субдискретизирующих слоев выполняют уменьешние размерности карты. Это можно делать разными способами, но зачастую используется метод выбора максимального элемента(max-pooling) в каждой из карты предыдущего слоя.

5.5 Рекуррентная нейронная сеть

Рекуррентные нейронные сети — вид нейронных сетей, в которых имеется обратная связь. Это значит, что нейроны элементов последующих слоев имеют соединения с нейронами предшествующих слоев. Такая архитектура приводит к возможности учета результатов преобразования нейронной сетью информации на предыдущем этапе для обработки входного вектора на следующем этапе функционирования сети. Рекуррентные нейронные сети активно применяются для решения задачах автоматической обработки текстов, таких как моделирование языка, распознавание речи, перевод.

GRU(Gated Recurrent Units) - особый тип рекуррентных нейронных сетей. На рис. 3 представлена архитектура GRU ячейки, где

$\sigma(x) = \frac{1}{1+e^{-x}}$ - функция активации сигмоида

$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ - функция активации гиперболический тангенс

x_t - входной вектор в текущую ячейку

h_{t-1} - выход предыдущей ячейки

6 Вычислительные эксперименты

Для датасета Reuters в качестве тренировки были реализованы только традиционные методы машинного обучения. Для набора данных Web of Science кроме указанных были также реализованы нейросетевые модели. Ниже приведены конфигурации и описания моделей, достигших лучших результатов на наборе данных Web of Science.

6.1 Модели машинного обучения

Для методов машинного обучения была проведена предобработка текстов, описанная в разделе 3. В качестве методов машинного обучения были взяты: наивный байесовский классификатор вместе с частотным кодированием(NBC), метод опорных векторов вместе с частотным кодированием(SVM) и TF-IDF преобразованием(SVM(TF-IDF)). Для выбора лучшей модели была использована кросс-валидация, в ходе которой вся обучающая выборка разбивалась на 5 случайных подвыборок. Обучение происходило на каких-либо четырех, контроль соответственно на одной оставшейся. Для подбора оптимальных параметров модели был использован grid search, результаты которого представлены в таблицах 1 и 2.

Обозначения в таблице:

- ngrame_range - длина(в количестве слов) ngram, взятых для признака
- max_features - ограничение сверху на количество признаков(None - нет ограничений)
- max_df - ограничение сверху на частотность признаков в документах корпуса
- min_df - ограничение снизу на частотность признаков в документах корпуса

Данные параметры используются для отбора более информативных признаков и снижения размерности признакового пространства.

6.2 Нейросетевые модели

В качестве нейросетевых моделей были реализованы следующие: сеть прямого распространения(DNN), сверточная нейронная сеть(CNN), рекуррентная нейронная

Параметр	NBC	SVM	SVM(TF-IDF)	Тестируемые значения
ngram_range	3	3	3	1, 2, 3
max_features	None	None	None	None, 3000, 5000
max_df	0.5	0.3	0.5	0.3, 0.5, 0,75, 0.9
min_df	1	3	3	1, 3

Таблица 1: Перебор параметров на WOS

Параметр	NBC	SVM	SVM(TF-IDF)	Тестируемые значения
ngram_range	1	3	3	1, 2, 3
max_features	3000	5000	None	None, 3000, 5000
max_df	0.3	0.9	0.9	0.3, 0.5, 0,75, 0.9
mi_df	1	3	3	1, 3

Таблица 2: Перебор параметров на Reuters

сеть(RNN). Для нейросетевых моделей предварительная обработка документов качество не улучшила(в том числе и только удаление стоп-слов), поэтому в таблице для них указаны результаты экспериментов без предобработки. На вход первой сети подавались вектора, полученные TF-IDF преобразованием предобработанных текстов как в предыдущем разделе. На вход сверточной и рекуррентной сети подавались векторы предварительно обученной модели GLoVE, взятой с nlp.stanford.edu/projects/glove/. Архитектуры DNN и RNN приведены на таблице 3 и 4. Архитектура сверточной сети взята с github.com/kk7nc/Text_Classification#convolutional-neural-networks-cnn и достаточно громоздка, чтобы представить ее здесь. Идея архитектуры RNN была взята с https://github.com/kk7nc/Text_Classification#gated-recurrent-unit-gru и изменена для данной задачи.

7 Результаты

Результаты экспериментов были сравнены со результатами статьи [2], в которой приведены результаты тестирования описанных методов машинного обучения и нейросетевых моделей.

Номер слоя	Слой	Размер выхода
1	Полносвязный слой(relu)	512
2	Dropout(0.5)	512
3	Полносвязный слой(softmax)	33

Таблица 3: Архитектура DNN

Номер слоя	Слой	Размер выхода
1	Полносвязный слой	(400, 50)
2	GRU(двунаправленный)	(400, 1024)
3	Dropout(0.2)	(400, 1024)
4	GRU(двунаправленный))	1024
5	Dropout(0.2)	1024
6	Полносвязный слой(relu)	512
7	Полносвязный слой(softmax))	33

Таблица 4: Архитектура RNN

Результаты классификации оценивались с помощью метрики accuracy, т. е. считалась доля верно классифицированных объектов к общему количеству объектов. В таблице 5 приведены итоговые результаты построенных классификатров по каждой из моделей и результаты из статьи [2]. В таблице синим цветом выделено лучшее качество среди каждого из методов, красным - лучшее качество по всем методам.

Модель	Экспериментальные результаты	Результаты из статьи[1]
NBC	75.6	68.8
SVM	84.16	80.65
SVM(TF-IDF)	84.54	83.16
DNN	83.90	80.02
CNN	84.70	83.29
RNN	85	83.96

Таблица 5: Результаты экспериментов (Accuracy в процентах) на WOS-11967

Модель	Экспериментальные результаты	Результаты из статьи[1]
NBC	85.66	68.8
SVM	88.60	80.65
SVM(TF-IDF)	89.73	83.16

Таблица 6: Результаты экспериментов (Accuracy в процентах) на Reuters-21578

Обозначения моделей:

- NBC - наивный байсеовский классификатор
- NBC(TF-IDF) - наивный байесовский классификатор с TF-IDF
- SVM - метод опорных векторов
- SVM(TF-IDF) - метод опорных векторов с TF-IDF
- DNN - сеть прямого распространения с полносвязными слоями
- CNN - сверточная нейронная сеть.

- RNN - реккурентная нейронная сеть.

Видим, что на всех моделях достигнута точность, превосходящая той, которая указана в статье [2].

Также обратим внимание на то, что с помощью сверточной и рекуррентной сети получилось достичь чуть большей точности, чем в лучшей модели среди методов машинного обучения(SVM + TF-IDF), хотя разница не очень большая. В таблице указаны наилучшие результаты среди несольких попыток обучения сети. Так как нейронная сеть может получать немного разные результаты в различных попытках обучения, то для полноты картины необходимо усреднить результат нейронной сети на нескольких попытках обучения сети.

Код всех классификаторов можно найти в репозитории github.com/iden-alex/coursework-msu-3rd-course.

8 Заключение

В данной работе проводилось исследование основных методов классификации текстов, в том числе и методов с использованием нейронных сетей.

В процессе выполнения работы было выполнено следующее:

- Для задачи тематической классифкации реализованы: метод опорных векторов, наивный байесовский классификатор, сеть с прямым распространением, сверточная и рекуррентная нейронные сети
- Получены результаты, превосходящие результаты статьи [2]
- Показано, что в рассматриваемой задаче на наборе данных WOS-11967 сверточная и рекуурентная сети достигают чуть большей точности, чем традиционные методы машинного обучения.

Список литературы

[1] *Goldberg Yoav.* A primer on neural network models for natural language processing // *CoRR.* — 2015. — Vol. abs/1510.00726.

[2] Rmdl: Random multimodel deep learning for classification / Kamran Kowsari, Mojtaba Heidarysafa, Donald E. Brown et al. // Proceedings of the 2nd International Conference on Information System and Data Mining. — ICISDM '18. — New York, NY, USA: Association for Computing Machinery, 2018. — P. 19–28.

[3] *Sebastiani Fabrizio.* Machine learning in automated text categorization // *CoRR.* — 2001. — Vol. cs.IR/0110053.

[4] Text classification algorithms: A survey / Kowsari, Jafari Meimandi, Heidarysafa et al. // *Information.* — 2019. — Apr. — Vol. 10, no. 4. — P. 150.

[5] *Kowsari Kamran, Meimandi Kiana Jafari, Heidarysafa Mojtaba et al.* — Text Classification Algorithms: A Survey(code), 2019.

[6] *Анастасия Рысьмятова.* — Использование сверточных нейронных сетей для задачи классификации текстов, 2016.

[7] *Воронцов К. В.* — Курс лекций по машиному обучению, 2015.