ОБЗОР ПОДХОДОВ К КЛАССИФИКАЦИИ ТЕКСТОВ АКТУАЛЬНЫМИ МЕТОДАМИ

А.А. Казанцев, *Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, farvest.ax@yandex.ru;*
М.В. Прохоров*, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, prokhorov.m.v@mail.ru;*
П.С. Худякова, *Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, dobrovolina@mail.ru.*

УДК 004.421.6

Аннотация. В условиях роста количества информации и в то же время увеличения доступной вычислительной мощности все большее значение приобретает автоматическая классификация данных, в частности текстовых. В данной статье представлен обзор основных алгоритмов классификации текста и его этапов. Рассмотрены принципы работы четырех основных методов – наивного байесовского, *k*-ближайших соседей, метода опорных векторов и использования нейронных сетей. В статье описывается современное состояние текущих подходов, а также рассматриваются способы, которые позволяют уменьшить вычислительную сложность алгоритмов и увеличить точность классификации.

Ключевые слова: классификация текстов; обработка текстов; наивный байесовский классификатор; метод ближайших соседей; метод опорных векторов; нейронные сети.

AN OVERVIEW OF APPROACHES TO THE CLASSIFICATION OF TEXTS BY RELEVANT METHODS

Aleksey Kazantsev, *St. Petersburg state university of telecommunications n/a prof. M.A. Bonch-Bruevich;.*
Maxim Prokhorov*, St. Petersburg state university of telecommunications n/a prof. M.A. Bonch-Bruevich;.*
Polina Khudyakova*, St. Petersburg state university of telecommunications n/a prof. M.A. Bonch-Bruevich.*

Annotation. In the context of an increasing amount of information and, at the same time, an increase in available computing power, automatic classification of data, in particular text data is becoming increasingly important. This article provides an overview of the main algorithms for text classification and its stages. The principles of operation of the four main methods – Naive Bayesian, *k*-Nearest Neighbors, support vector machine, and the use of neural networks are considered. The article describes the current state of the current approaches, as well as the ways that can reduce the computational complexity of algorithms and increase the accuracy of classification.

Keywords: text classification; text processing; naive Bayes classifiers; *k*-nearest neighbors; support vector machine; neural network.

Классификация текста – это акт разделения набора входных документов на два или более классов, где каждый документ может быть отнесен к одному или нескольким классам. Огромный рост информационных потоков и, в особенности, взрывной рост интернета способствовали росту автоматизированной классификации текстов. Развитие компьютерной техники обеспечило достаточную вычислительную мощность, позволяющую использовать автоматизированную

классификацию текстов в практических приложениях. Классификация текста обычно используется для обработки спам-писем, тематического деления больших текстовых коллекций, а также в поисковых системах интернета.

Процесс классификации текста с точки зрения автоматических систем можно четко разделить на два основных этапа:

1. Этап поиска информации, когда данные извлекаются из текста и сохраняются в числовом виде.
2. Основной этап классификации, когда алгоритм обрабатывает эти данные, чтобы принять решение о том, к какой категории должен принадлежать текст [1].

К процессу классификации добавляются дополнительные этапы, чтобы уменьшить объем вычислений и обучить алгоритмы перед фактической классификацией (рис. 1).

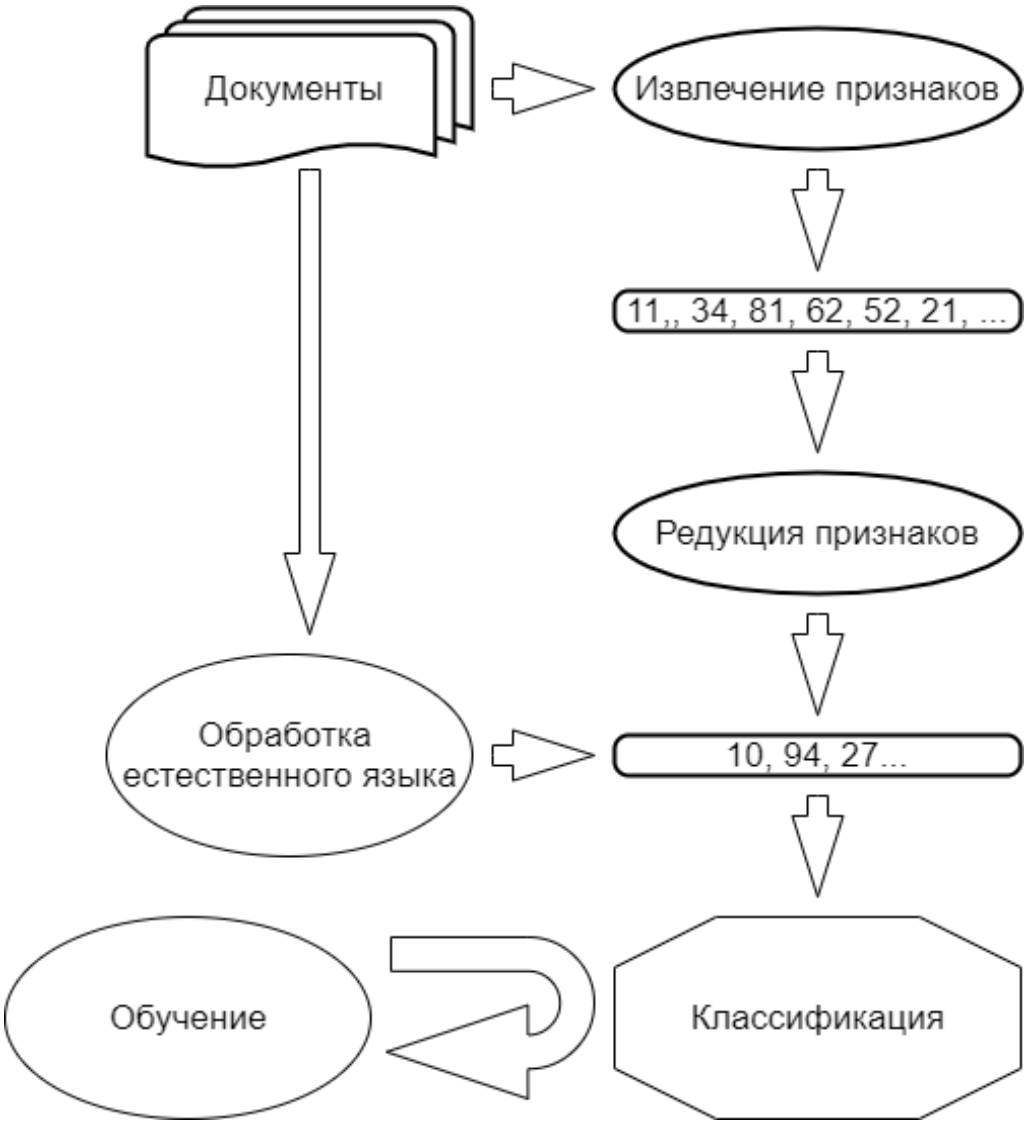

Рисунок 1

В данной статье будет рассмотрено современное состояние всех этих этапов, рассмотрены методы и алгоритмы, используемые для снижения вычислительной сложности и повышения точности процесса классификации текстов. Эта статья

направлена на то, чтобы дать простой и понятный обзор основных практических моментов этого процесса.

Практическое применение

Одной из самых главных проблем является большое количество «спам-сайтов», размещающих на своих страницах рекламу из сети *Google AdSence* или других подобных рекламных сетей. Большую часть времени на таких страницах реклама лишь незначительно связана с содержанием страницы, и люди, которые нажимают на такие объявления, имеют гораздо меньшую вероятность «конверсии», то есть покупки рекламируемых товаров или услуг. Однако, как только пользователь нажимает на рекламу, рекламируемый сайт должен заплатить (через *Google*, в случае *Google AdSence*) фиксированную плату за переход на сайт, где размещена реклама. Поэтому спам-сайты, заманивающие посетителей рекламой товаров, которые эти посетители не собираются покупать, являются для рекламодателя пустой тратой денег.

1. Предобработка текста

1.1 Извлечение признаков

При автоматической классификации текста возникает ряд сложностей, и первая проблема будет заключаться в том, что все математические методы, которые могут быть использованы для решения задачи классификации, работают только на числах, а не на длинных, неструктурированных отрывках текста, поэтому, прежде чем выполнять какие-либо математические операции с текстом, необходимо использовать определенные алгоритмы для извлечения некоторой числовой информации (признаков) классифицируемого текста, имеющих отношение к классификации (рис. 2) [3].

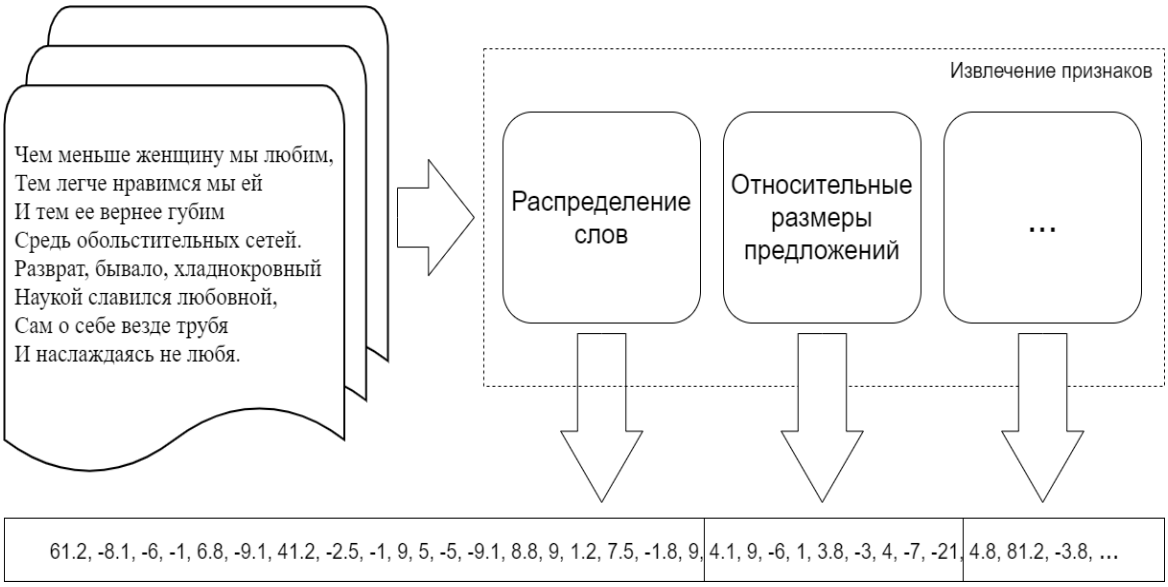

Рисунок 2

Наиболее распространенной используемой характеристикой является распределение частот слов. При подготовке к извлечению информации алгоритм просматривает все доступные тексты (обычно ограниченные набором обучающих документов) и создает словарь всех слов, встречающихся в этих текстах. Затем каждому слову в словаре отводится место в выходном векторе признаков. Когда объекты документа извлекаются, это значение в векторе будет представлять

количество раз, когда это слово встречалось в документе. Другая вариация этого метода включает в себя только добавление «1» в вектор, если конкретное слово есть в документе, и «0», если его нет, или деление числа раз, когда слово встречалось в этом документе, на среднее число раз, когда оно встречалось за всю коллекцию документов [5]. Некоторые методы рекомендуют учитывать распределение слов только в начале документа.

Относительно новым развитием в общей классификации текстов является использование частот символов или последовательностей слов вместо частот слов. В распределении последовательностей символов частота определенной комбинации символов отслеживается через тексты. Распределение последовательностей слов выводит это на другой уровень, отслеживая последовательности словосочетаний. При таком методе повторения фраз «основная часть классификации текста» и «небольшая часть текста» не будут иметь никакого сходства с текстами, так как последовательности слов совершенно различны [3].

Из классифицируемого текста могут быть извлечены дополнительные признаки, однако характер таких признаков должен сильно зависеть от характера проводимой классификации. Если требуется разделить веб-сайты на спам- и не-спам- сайты, то распределение частот слов или онтология мало пригодны для классификации при намеренном смешивании текстов с обычных веб-сайтов при создании спам-сайтов.

В таких ситуациях целесообразно проконсультироваться с экспертами и извлечь некоторые из их знаний для использования на этапе извлечения признаков. В примере с веб-спамом можно обнаружить рекламу на странице и вычислить, сколько пикселей на экране компьютера будет использоваться рекламой для отображения содержимого страницы. Часто можно различить сайты, ориентированные на смысловой контент (скорее всего, не спам), и сайты, ориентированные на рекламу (скорее всего, спам), просто взглянув на пропорции контента и рекламы на первых экранах страницы.

1.2 Обработка естественного языка

Подходы к обработке естественного языка могут быть применены как к этапам извлечения признаков, так и к этапам редукции признаков в процессе классификации текста. Лингвистические признаки могут быть извлечены из текстов и использованы как часть их векторов признаков. Например, части текста, которые были написаны в прямой речи, использование различных типов склонений, длина предложений, соотношение различных частей речи в предложениях могут быть обнаружены и использованы в качестве вектора признаков. Обработка естественного языка также может быть использована способами, охватывающими как извлечение признаков, так и их редукцию [3]. Можно показать, что извлечение информации из таких форм исходных документов может уменьшить размерность входного вектора без снижения эффективности классификации. Извлечение признаков последовательности слов, описанное выше, а также большинство методов сокращения количества слов, описанных в следующем разделе, используют знания из области лингвистики и обработки естественного языка. Однако еще большее сотрудничество между лингвистикой и учеными, занимающимися классификацией текстов, могло бы привести к появлению новых способов включения лингвистических знаний в разработку механизмов извлечения признаков, образования признаков и даже классификаторов в направлении классификации текстов.

1.3 Редукция признаков

В типичном подходе для классификации текстовых статей частота слов используется в качестве основной части вектора признаков. Это обычно приводит к созданию векторов признаков с размерностью порядка десятков тысяч измерений. Вычислительная сложность любых операций с такими векторами признаков будет пропорциональна размеру вектора признаков, поэтому любые методы, которые уменьшают размер вектора признаков, не оказывая существенного влияния на производительность классификации, очень приветствуются в любом практическом применении. Кроме того, некоторые конкретные слова в конкретных языках только добавляют шум к данным, а удаление их из вектора признаков фактически повышает эффективность классификации [3].

Набор операций сокращения признаков включает в себя комбинацию трех общих подходов:

1. Стоп-слова.
2. Стемминг.
3. Статистическая фильтрация.

В любом языке есть множество слов, которые передают мало смысла или вообще ничего не значат, но требуют грамматической структуры языка; такие слова называются «стоп-словами». В качестве примера в русском языке стоп-словами считаются такие слова, как: «ну», «и», «а» и многие другие. Общепринятой практикой является исключение стоп-слов из вектора признаков. Можно использовать списки стоп-слов или определять стоп-слово по их частоте, что считается более эффективным и независимым от языка [1].

Второй способ традиционной редукции признаков – это использование стемминга для уменьшения частоты слов с общим корнем до одного признака, например, если бы документ содержал семь экземпляров слова «дом», три экземпляра слова «дома» и два экземпляра слова «доме», то после стемминга эти три отдельных признака были бы сведены только к одному, который описывал бы, что слова с корнем, похожим на «дом», встречались в документе 12 раз (7+3+2).

Методы статистической фильтрации используются для отбора тех слов, которые имеют более высокую статистическую значимость. Много различных статистических методов исследуются и используются для векторной фильтрации признаков, но основное различие между этими методами заключается в том, как много информации об исходных данных используется. Можно вычислить обобщенную статистическую значимость слова в зависимости от того, насколько различна частота его употребления в документах, но более сложные алгоритмы также учитывают предложенную классификацию указанных документов и вычисляют статистическую значимость слов в конкретных категориях [1].

2 Классификация

2.1 Статистическая классификация

Наиболее широко используемый тип классификатора – Наивный байесовский классификатор. Среди других статистических классификаторов байесовский классификатор является самым простым, но все же очень эффективным и, благодаря своей простоте, является также единственным наиболее исследованным классификатором. Наивный байесовский классификатор предполагает, что контекст входного вектора признаков статистически независим [2]. В базовой форме для двух возможных классов (например, спам и не-спам-тексты) байесовская вероятность того, что текст является спамом, равна

вероятности нахождения его компонентов вектора признаков в спам-тексте, умноженной на вероятность того, что любой текст является спам-текстом (т. е. отношение спам-текстов в коллекции, деленное на общую вероятность того, что определенная компонента вектора признаков встречается в тексте [4]. Во время обучения, если известно, что текст является спамом, то его особенности добавляются как к вероятностям спама, так и к общим вероятностям текстовых признаков. Особенности не спам-текстов добавляются только к общим вероятностям текстовых функций.

Вероятность того, что любой текст является спамом, является оценочным параметром алгоритма – чем больше эта вероятность, тем больше будет количество текстов, классифицированных как спам. Увеличение этого числа уменьшит количество ложно-отрицательных срабатываний (т. е. спам-текстов, классифицированных как не спам), но также увеличит количество ложных срабатываний (т. е. не спам-текстов, классифицированных как спам).

Структура наивного байесовского классификатора позволяет легко закодировать некоторые экспертные знания в обучающий набор данных – например, если эксперты согласны с тем, что слово «Виагра» появляется в спаме гораздо чаще, чем в не-спамовых текстах, то мы можем непосредственно увеличить вероятность спама этого слова.

2.2 Функциональная классификация

Если рассматривать каждое число в векторе признаков как координату в измерении, то каждый документ можно представить в виде точки в многомерном пространстве, где число измерений равно числу объектов в векторе признаков [10]. Такая интерпретация позволяет использовать геометрические способы классификации, которые также могут быть более легко представлены визуально.

Одним из простейших геометрических (или функциональных) классификаторов является классификатор k -ближайших соседей (kNN). Идея этого классификатора очень проста – в многомерном пространстве находится точка, представляющая классифицируемый документ, и исследуется область вокруг, чтобы узнать, какие еще точки находятся поблизости [6]. Учитывается только k ближайших соседей. Если все они принадлежат к одной и той же категории, то новый документ также будет отнесен к этой категории. В противном случае распределение определяет вероятность появления документа принадлежность к категории [8]. Другими словами, если из пять ближайших соседей четырех относятся к классу А, а один к классу В, то новый документ классифицируется к классу А с 80%-й вероятностью.

В настоящее время одним из наиболее активно исследуемых классификаторов является метод опорных векторов (SVM). Если визуализировать два класса классификации как два набора точек и представить пограничную область между классами, то можно идентифицировать документы, которые являются граничными примерами каждого класса [4]. Если найти вектор, который проходит через точки в пространстве, представляющие эти граничные документы, так что все документы категории находятся на одной стороне этого вектора, то это будет опорный вектор. Математически, усредняя два вектора поддержки из двух категорий, мы можем определить вектор, который будет лежать примерно посередине между категориями и который может быть использован для классификации новых документов, основываясь на том, на какой стороне этого вектора находится новый документ [7].

2.3 Классификация с использованием нейронных сетей

По своей сути использование нейронной сети для любой задачи классификации является простым процессом: данные вектора признаков подаются на входы сети, а на выходах происходит категоризация. Каждому выходу непосредственно присваивается категория – если, например, самый сильный сигнал поступает из нейронной сети на выход номер 3, то классифицируемый объект относится к третьей категории. Разница в силе между самым сильным выходным сигналом и другими выходными сигналами указывает на уверенность сети в этой классификации. Если выход не является достаточно сильным, то классификация может быть отклонена для повышения надежности результата [9].

Однако фундаментальная проблема в использовании нейронной сети заключается в том, чтобы определить фактический дизайн сети. Теоретически можно построить нейронные сети любой сложности, но очень трудно математически предсказать, сможет ли данная нейронная сеть преуспеть в конкретной задаче классификации. Учитывая эту сложность, исследователи сосредоточились на простых и предсказуемых нейросетевых конструкциях для решения практических задач в области классификации текста и используют только более сложные конструкции в новых и более сложных областях распознавания изображений и речи.

3. Обучение и оценка

Классификатор сам по себе не обладает знанием. Любое знание, необходимое для классификации, должно быть получено классификатором либо путем прямого перевода экспертных знаний, либо из обучения. Два основных типа обучения – это обучение с учителем и без него.

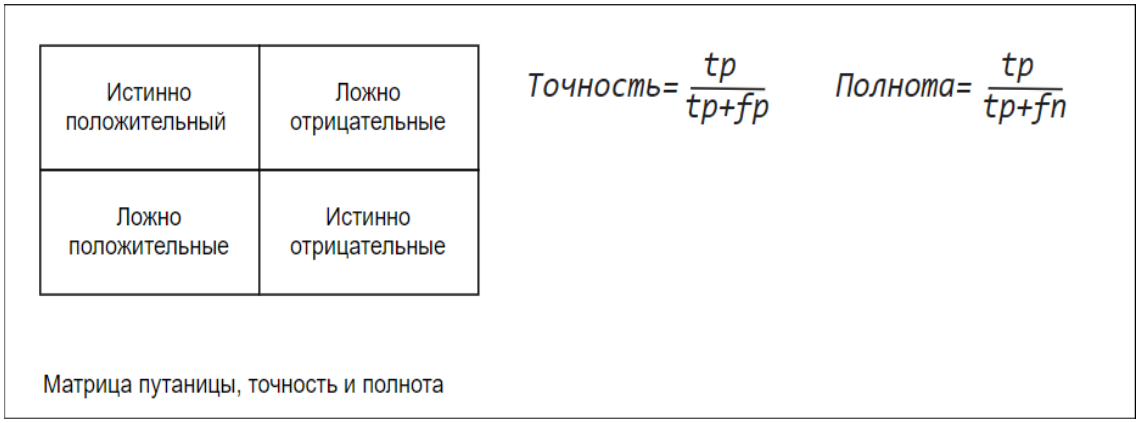

Рисунок 3

Сколько примеров нужно и как долго необходимо тренировать сеть, пока она не будет удовлетворять задаче? Прежде всего, очевидно, что чем больше присутствует примеров, тем лучше будут результаты, если предоставить все возможные входные векторы и корректные классификации для них, то это наилучший возможный сценарий, но в этом случае доступны гораздо более эффективные способы хранения и восстановления этой информации (классификация на основе памяти), и нет необходимости использовать более сложные классификаторы. Там, где классификация текста на основе нейронных сетей показывает наилучшие результаты, это возможность обобщения – способ обеспечить результат классификации для входных данных, которые классификатор не видел во время обучения. Для измерения успеха в этих областях используются

две основные меры: точность и ошибка [9]. Сначала вычисляется матрица путаницы. Для простого случая из двух категорий это матрица 2x2, в которой тестовые случаи распределяются следующим образом: первая ячейка – это количество тестовых случаев, которые были правильно отнесены к первой категории (истинно положительные), вторая – количество тестовых случаев, которые должны были быть отнесены к первой категории, но были классифицированы как принадлежащие ко второй (ложно отрицательные), и третья и четвертая ячейки соответственно ложно положительные и ложно отрицательные (рис. 3).

В простейшем случае контролируемого обучения все данные примера (пары входных векторов и правильные выходные векторы) случайным образом делятся на три части – обучающие, тестовые и проверочные наборы данных. Затем начинается обучение – сети показываются примеры из обучающего набора в случайном порядке и любые ошибки исправляются методом обратного распространения. Процесс проходит через все данные обучающего набора несколько раз, пока результат на последней итерации не превысит заданный минимальный порог. Иногда используется фиксированное число итераций [9].

После завершения обучения начинается процесс тестирования. В процессе тестирования все примеры из тестового набора данных передаются классификатору и вычисляется точность по сравнению с тестовыми данными. Если эта точность ниже заданного минимального порога обобщения (обычно около 70-80%), то система возвращается на стадию обучения (обычно на фиксированное число итераций).

Заключительный этап – это этап проверки. Процесс аналогичен этапу тестирования – вычисляется средняя ошибка по всему набору данных. От тестирования стадия проверки отличается ее результатом. Мера точности (часто мера $F1$), вычисляемая на этапе верификации, является конечной точностью классификатора и представляет собой каноническую цифру, по которой можно сравнивать различные системы классификации. Если значение хорошее, то сеть готова к использованию и обучение завершено. Однако если точность верификации сети неудовлетворительна, то никакое обучение ей не поможет – для улучшения результата необходимо изменить структуру системы классификаторов.

Альтернативные способы обучения предполагают более эффективное использование примеров данных. Это важно, потому что даже ручная классификация тысячи примеров (небольшое число для обучения нейронной сети) является очень трудоемкой задачей. Кроме того, только квалифицированный специалист-человек может выполнить эту задачу, что может привести к большим денежным затратам на процедуру. Поэтому получение максимального эффекта от относительно небольшого числа примеров является острой проблемой. Один из способов сделать это – обучить несколько классификаторов одновременно с различными разбиениями одних и тех же обучающих данных и выбрать тот, который имеет наилучшие результаты проверки.

В неконтролируемом обучении нет учителя и, следовательно, нет прямой обратной связи по действиям классификатора. Вместо этого классификатор пытается сделать хорошее представление входного вектора на выходе, и для определения качества такого представления используется независимая от задачи мера [9]. Например, в веб-кластеризации, если имеется большое количество документов, но неизвестна строгая структурная полной классификации, в которую необходимо их классифицировать, можно использовать неконтролируемое обучение. Неконтролируемое обучение группирует веб-страницы с похожим содержанием или темой. Мерой качества может быть достоверность

классификации при фиксированном числе используемых категорий. Однако было показано, что неконтролируемые нейронные сети извлекают шаблоны там, где их нет, поэтому необходимо внимательно относиться к проблемам валидации, чтобы избежать переобучения.

Обучение без учителя может быть объединено с контролируемым обучением в целях редукции признаков перед классификацией с помощью классической нейронной сети обратного распространения ошибки или для компенсации отсутствия примеров негативной классификации в сценарии классификации двух классов. Обучение без учителя сочетает в себе известные классификации и неизвестные классификации для расширения диапазона возможных классов.

Еще один способ – это обучать несколько классификаторов последовательно, причем каждый следующий классификатор фокусируется на примерах, на которых предыдущие классификаторы работали плохо [9]. Таким образом формируется набор классификаторов, причем различные классификаторы поддерживают друг друга.

Заключение

Существующие алгоритмы классификации текста работают более эффективно, если лучше понимать методы извлечения признаков и способы их правильной оценки. В настоящее время основными алгоритмами для классификации текстов являются методы наивного байеса, *k*-ближайших соседей, метод опорных векторов и нейронные сетей. Данные методы были рассмотрены в статье, однако предварительная обработка текста и документов, а также некоторые улучшения алгоритмов могут помочь повысить точность и надежность оценки.

Литература

1. Батура Т.В. Методы автоматической классификации текстов //Программные продукты и системы, 2017. – Т. 30. – № 1.
2. Басалаева А. Ю., Гареева Г. А., Григорьева Д. Р. Web-scraping и классификация текстов методом наивного Байеса //Инновационная наука, 2018. – Т. 2. – № 5.
3. Поляков И.В. и др. Проблема классификации текстов и дифференцирующие признаки, 2015.
4. Абдурахманова Н.Н. и др. Сравнительный анализ методов Наивного Байеса и SVM алгоритмов при классификации текстовых документов // Молодой ученый, 2019. – № 29. – С. 8-10.
5. URL: https://docplayer.ru/45424867-Naivnyy-bayesovskiy-klassifikator.html.(дата обращения: январь 2021).
6. Стрюков Р.К., Шашкин А. И. О модификации метода ближайших соседей // Вестник ВГУ, 2015. – № 1. – С. 114-120.
7. Демидова Л.А., Соколова Ю.С. Классификация данных на основе SVM-алгоритма и алгоритма k-ближайших соседей // Вестник Рязанского государственного радиотехнического университета, 2017. – Т. 62. – С. 119.
8. Гришанов К.М., Белов Ю.С. Метод классификации k-nn и его применение в распознавании символов // В сборнике: Фундаментальные проблемы науки сборник статей Международной научно-практической конференции: Тюмень НИЦ АЭТЕРНА, 2016. – Т. 317.
9. Ха Л.М. Сверточная нейронная сеть для решения задачи классификации // Труды Московского физико-технического института, 2016. – Т. 8. – № 3 (31).
10. Пайвин Д.Н., Глазкова А. В. Исследование методов векторного представления естественного языка на примере классификации коротких текстов

//Математическое и информационное моделирование: материалы Всероссийской конференции молодых ученых, г. Тюмень, 18 апреля 2019 г. – Изд-во Тюм. гос. ун-та, 2019.