Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

Метод определения признаков авторского стиля для текстов на русском языке

Студент ИУ7-85Б (Группа)	(Подпись, дата)	**О.П. Кондрашова** (И.О.Фамилия)
Руководитель ВКР	(Подпись, дата)	**А.П. Ковтушенко** (И.О.Фамилия)
Консультант	(Подпись, дата)	**Л.Л. Волкова** (И.О.Фамилия)
Консультант	(Подпись, дата)	(И.О.Фамилия)
Нормоконтролер	(Подпись, дата)	**Ю.В. Строганов** (И.О.Фамилия)

2021 г.

РЕФЕРАТ

Расчётно-пояснительная записка, ?? с., ?? рисунков, ?? табл., ?? источников.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последние годы очень быстрыми темпами развивается область обработки естественных языков. Во многом это связано с тем, что с каждым годом объём текстовой информации, используемой человечеством, увеличивается, и растёт потребность в более эффективных алгоритмах обработки и анализа документов, написанных на естественных языках. Особо важную роль играет возможность определить автора текста, основываясь на его стилистических признаках.

Выявление признаков авторского стиля позволяет установить принадлежность текста определенному человеку. Актуальность данного направления в компьютерной лингвистике обусловлена необходимостью в выявлении плагиата или в создании рекомендательной системы для нахождения похожжих текстов. Для определения стиля автора необходимо выделить характерные признаки из принадлежащих ему текстов.

Целью данной работы является разработка метода определения признаков авторского стиля для текстов на русском языке.

Для достижения поставленной цели ставятся следующие задачи:

- изучение предметной области;
- анализ существующих решений определения авторства текста;
- анализ алгоритмов классификации;
- разработка метода определения признаков авторского стиля;
- разработка метода классификации текстов, использующего предложенный метод классификации текстов;
- проектирование структуры и реализация программного обеспечения, реализующего разработанный метод;
- подготовка входных данных и обучение классификатора;
- апробация предложенного метода.

Разрабатываемый метод позволит проводить классификацию текстов на русском языке по авторскому стилю на основании результатов морфологического анализа, не задействуя более трудоемкий и менее доступный синтаксический анализ.

1 Аналитический раздел

В данном разделе рассматриваются понятия авторского стиля и стилеметрических характеристик, этапы обработки и анализа текста, исследуются алгоритмы классификации, также приводится обзор существующих методов и решений для определения авторского стиля, анализируются их достоинства и недостатки. Также в этом разделе приводится описание последовательного решения задачи.

1.1 Описание предметной области

Под авторским стилем понимается совокупность характеристик, позволяющих установить авторство текста или выдвинуть предположение, кем может являться автор или к какой группе авторов он может принадлежать. К особенностям авторского стиля можно отнести грамматические конструкции, стилистические приемы, способы построения фраз и абзацев или любой другой набор признаков, который отличает конкретного автора от всех других.

Раздел лингвистики, занимающийся измерением стилевых характеристик с целью систематизации и упорядочения текстов, называется стилеметрией [1]. Объектом стилеметрии является текст, созданный конкретным автором. Предметом исследования являются элементы стиля, которые понимаются как особенности стиля конкретного автора. Стиль текста описывается как набор некоторых выделенных характеристик. В большинстве случаев в качестве характеризующих параметров текста выбираются статистические характеристики: количество использования определенных частей речи, знаков препинания, количество и длина предложений (измеренная в словах, слогах, знаках), количество полнозначных и служебных слов, средняя длина предложения в тексте и т.д.

Под авторским инвариантом понимаются стилеметрические характеристики, которые однозначно характеризует одного автора или небольшое число «близких авторов» [2]. Анализируемые характеристики должны принимать существенно разные значения для произведений разных групп авторов. Для успешного установления авторства нужно, чтобы число «разных групп» было достаточно велико, и при этом каждая группа должна включать в себя небольшое количество близких по стилю произведений, для которых точно определен автор.

Методы анализа стиля можно разделить их на две большие группы – экспертные и формальные. Экспертные методы предполагают исследование текста профессиональным лингвистом-экспертом. К формальным относятся приемы из теории вероятностей и математической статистики, алгоритмы классификации, кластерного анализа и нейронных сетей.

1.2 Этапы обработки текста

В данном разделе будет выполнен обзор этапов обработки текстов на естественном языке.

1.2.1 Токенизация

Токенизация — это базовый этап в автоматической обработке текстов, суть которого заключается в разбиении непрерывной строки на отдельные «слова» (токены) [3].

На этапе токенизации также решается задача разбиения текста на предложения. При разделении текста на отдельные предложения недостаточно учитывать только наличие точки и большой буквы после нее. Например, под данное правило не попадают сокращения и инициалы («И.И. Иванов» — три разных предложения).

Токенизация текста состоит из нескольких этапов. В первую очередь текст полностью приводится к нижнему регистру. На данном этапе часть информации может быть утеряна. Например, «ООО» представляет собой аббревиатуру (общество с ограниченной ответственностью), в то время как «ооо» — способ выражения эмоций.

Следующий этап — это замена всех знаков препинания и прочих символов на пробелы. Текст на естественном языке состоит не только из букв, но и из символов: скобок, кавычек, тире. Если все символы, не являющиеся буквами или строками, заменить на пробелы, то слова, разделённые пробелами, можно объявить отдельными токенами. Однако при наличии сложных составных слов замена дефиса на пробел может привести к потере исходного смысла предложения. Например, слово «каком-либо» было бы правильно считать за единый токен, но при замене дефиса на пробел, полчаются два отдельных слова: «каком» и «либо».

Далее каждое слово, отделённое пробелом, объявляется отдельным токеном. На данном этапе стоит учитывать, что некоторые наборы слов должны рассматриваться как одно. Например, названия городов («Нижний Новгород»), или сокращения («к.т.н.», кандидат технических наук). Если рассматривать слова «Нижний» и «Новгород» независимо друг от друг, важная информация, определяющая конкретный текст, может быть утеряна, а разделенные на буквы сокращения и вовсе теряют какой-либо смысл [4].

1.2.2 Нормализация

Следующий этап после токенизации — это нормализация слов в тексте, то есть приведение каждого слова к его начальной форме.

Существует два основных подхода к нормализации: стемминг и лемматизация [4].

Стемминг — это подход, суть которого заключается в следующем: по некоторым правилам от каждого слова отрезается его окончание. Данный подход

может работать некорректно в том случае, если при изменении формы слово меняется целиком (например, «был», «есть», «будет»).

Суть подхода лемматизации заключается в использовании словаря, в который предварительно записано большое количество слов и их форм. В первую очередь слово проверяется по словарю. Если слово найдено в словаре, то с его помощью можно найти известные формы данного слова. В противном случае по определённому алгоритму выводится способ изменения слова, и на его основании делаются выводы о первоначальной форме слова.

Лемматизация дает более точный результат при работе с незнакомыми словами, но в связи с поиском в словаре и работой алгоритма приведения к нормальной форме, является более медленным подходом по сравнению со стеммингом.

1.3 Этапы анализа текста

Полный анализ текста включает в себя несколько этапов [3]. Сложность полного анализа заключается в том, что естественный язык неполон и не всегда формализован, также возникающие сложности связаны с тем, что на практике до сих пор не реализованы все теоретические положения, разработанные на данный момент. Выбор того, какой анализ текста необходимо провести, зависит от поставленной задачи и выбранного пути ее решения.

Графематический анализ – начальный анализ текста на естественном языке, обеспечивающий выделение синтаксических или структурных единиц. Входные данные могут представлять собой линейную структуру, содержащую единый фрагмент текста, но в более общем случае подаваемый на вход текст состоит из многих структурных единиц: основного текста, заголовков, вставок, комментариев и т.д. При машинном переводе ставится задача сохранить структуру исходного текста. Первый вариант – линейная структура текста без вставок – обычно встречается в диалоговых системах. Но и в этом случае

графематический анализ должен выделять синтаксические единицы: абзацы, предложения, отдельные слова и знаки препинания.

Морфологический анализ обеспечивает для каждой словоформы в тексте определение ее нормальной формы, от которой данная словоформа образована, а также набора характеризующих ее параметров. Морфологический анализ является основой для нормализации текста и проводится для того, чтобы в дальнейшем ориентироваться только на нормальную форму. Нормализация требует обязательного морфологического анализа текста, распознающего части речи с учетом контекста и многочисленных правил согласования (без него нормализация будет давать значительное количество ошибочных результатов).

Предсинтаксический анализ отвечает за две противоположные задачи: объединение отдельных лексических единиц в одну синтаксическую или, наоборот, ее разделение на несколько. В единую синтаксическую единицу объединяются изменяемые неразрывные словосочетания, т.к. в данном случае при использовании слов по отдельности теряется первоначальный смысл, заложенный в конкретное словосочетание. К задачам предсинтаксического анализа также относится проведение синтаксической сегментации, которая заключается в разметке линейного текста на фрагменты, привязанные правилам следующего этапа синтаксического анализа. Синтаксический анализ является задачей с экспоненциальным ростом сложности, поэтому для ускорения его работы часть задач переносится на этап предсинтакисческиго анализа.

Синтаксический анализ – самая трудоемкая часть анализа текста. На данном этапе необходимо определить роли слов в предложении и их связи между собой. Необходимо сопоставить линейную последовательность лексем (слов, токенов) языка с его формальной грамматикой. Результатом этого этапа является набор синтаксических деревьев, которые наглядно показывают выделенные в тексте связи и являются материалом для дальнейшем обработки.

На этапе постсинтаксического анализа требуется уточнить смысл, заложенный в слова и выраженный при помощи различных средств языка (предлоги, префиксы, аффиксы и т.д.). Необходимо учесть, что одна и та же

мысль может быть выражена различными конструкциями языка. В связи с этим полученное на предыдущем этапе синтаксическое дерево необходимо нормализовать, т.е. конструкция, выражающая некоторое действие различным образом для различных языков или ситуаций, должна быть сведена к одному и тому же нормализованному дереву.

Семантический анализ проводится для анализа текста «по смыслу». Семантический анализ уточняет связи, которые не смог уточнить постсинтаксический анализ, так как многие роли выражаются не только при помощи средств языка, но и с учетом значения слова. Словарь для поддержки семантического анализа должен оперировать смыслами и, следовательно, описывать свойства и отношения понятий, а не слов.

1.4 Методы и решения определения авторского инварианта

В данном разделе проанализированы существующие методы и решения для определения признаков авторского стиля.

1.4.1 Метод полного синтаксического анализа

Под синтаксическим анализом в области обработки текстов на естественном языке понимается построение такой структуры, которая позволяет приблизиться к некоторому формализованному представлению смысла текста и установить связи между словами в предложении.

Большинство моделей синтаксической структуры предложения опираются либо на грамматику составляющих, предложенной в работах Ноама Хомского [5], либо на грамматику зависимостей, для которой основополагающими считаются работы Люсьена Теньера [6] и Игоря Мельчука [7].

Грамматика составляющих предполагает, что предложение на естественном языке может быть представлено в виде иерархии составляющих.

Данная иерархия предполагает выделение синтаксических групп, которые не должны пересекаться между собой, но которые в свою очередь состоят из более мелких групп, вплоть до атомарных единиц – слов предложения. Такую иерархическую структуру называют деревом составляющих. Дерево составляющих удовлетворяет свойству проективности, согласно которому при представлении дерева графически связи между структурными единицами не должны пересекаться.

Грамматика зависимостей предполагает, что предложения текста представляют собой деревья зависимостей, в которых слова связаны ориентированными дугами, обозначающими синтаксическое подчинение. Считается, что подход с деревом зависимостей более точно отражает специфику языков с произвольным порядком слов. Грамматика зависимостей допускает наличие непроективных связей, нарушающих свойство проективности дерева

Синтаксический анализ разделяют на глубокий (полный) анализ и поверхностный [8, 9]. Задачей глубокого синтаксического анализа является построение полного синтаксического дерева предложения с максимальной связанностью с учетом дальних связей, а также определение грамматических функций слов предложения (подлежащее, сказуемое, обстоятельства места, времени и т.д.). Понятие поверхностного синтаксического анализа объединяет в себе различные подходы, направленные на построение неполной (частично связанной) синтаксической структуры текста разной сложности. Поверхностный синтаксический анализ охватывает такие задачи как разделение предложения на рекурсивно невложенные синтаксические группы, сегментацию (выделение в предложении различных оборотов и простых предложений в составе сложного), построение поверхностного синтаксического дерева.

Полный синтаксический разбор гарантирует высокое качество анализа, поскольку устанавливает все существующие зависимости при разборе предложения. Но т.к. в русском языке в предложении зачастую встречается свободный порядок слов (в отличие, например, от английского языка), данный метод зачастую требует корректирующего вмешательства эксперта вдобавок к

машинному синтаксическому разбору. В силу этих особенностей языка метод синтаксического разбора является очень затратным по времени. К тому же большая часть программного обеспечения, выполняющего синтаксический анализ, является коммерческими проектами и закрыта для использования.

1.4.2 Метод энтропийной классификации

Метод классификации текстов для задачи определения авторства, предложенный в работе [10], основан на применении алгоритмов сжатия данных.

Суть рассматриваемого метода заключается в том, чтобы добавить текст, автор которого неизвестен, к корпусу текстов, характеризующему конкретного автора, проанализировать, насколько хорошо сжимается полученная после добавления выборка текстов, и сравнить полученные результаты для различных авторов. В рамках энтропийного подхода правильным исходным классом текста является тот класс, на котором получены наилучшие результаты сжатия.

Рассмотрим на примере задачу классификации текста T относительно текстов $S_1, \dots, S_n$, характеризующих n классов. Выбор источника текста T должен осуществляться согласно оценке, рассчитываемой по формуле:

$$q(T) = argmin_i \, H(T|S_i), \qquad (1)$$

где $H(T|S)$ – характеристика энтропии текста T по отношению к тексту S

Существенным преимуществом метода энтропийной классификации является отсутствие необходимости в предварительной обработке текста.

1.4.3 Рекуррентные нейронные сети

Для решения задачи классификации текстов используют в том числе и нейронные сеть. Рекуррентные нейронные сети – подкласс нейронных сетей с обратными связями, которые используют предыдущие состояния сети для

вычисления текущего. Обычно данный класс нейронных сетей используется в задачах обработки последовательностей нефиксированной длины. В частности, рекуррентные нейронные сети показывают результаты лучше других методов в задачах классификации текста [11].

Рекуррентные нейронные сети позволяют использовать текст в его исходном виде, то есть учитывается порядок слов. Сеть последовательно получается на вход слова. При этом в рекуррентной сети есть скрытый слой h, который обновляется после появления каждого нового слова или токена (например, буквы). Слой обновляется, учитывая и своё предыдущее состояние, и полученное слово. После того как скрытое состояние обновлено, на его основе генерируется выход y для данного токена. Процесс повторяется до тех пор, пока через сеть не пройдут все слова в тексте.

Данная сеть описывается двумя формулами. Первая — это обновление скрытого состояния. Оно обновляется путём некого усреднения предыдущего значения и нового входа:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1}), \quad (2)$$

где h_{t-1} – состояние слоя в предыдущий момент времени;

x_t – слово, которое было подано на вход;

f – некая нелинейная функция активации.

Вторая формула, которой описывается сеть, — это значение на выходе:

$$y_t = g(W_{hy}h_t), \quad (3)$$

где h_t – обновлённое состояние скрытого слоя;

g – функция активации.

Рекуррентная нейронная сеть представляет собой сеть прямого распространения, длина которой зависит от числа токенов, которые подаются на вход. При этом параметры, матрицы W_{hh}, W_{xh}, W_{hy} являются общими и не

меняются от шага к шагу. Изменяются только скрытые состояния h_t, входы x_t, выходы y_t.

Для обучения рекуррентных нейронных сетей используется алгоритм обратного распространения ошибки по времени (backpropagation through time) [12], который является вариантом алгоритма обратного распространения ошибки (backpropagation), используемого для нейронных сетей прямого распространения сигнала.

1.4.4 Метод анализа длин слов

Для поставленной задачи определения авторства текста существует также решение анализа стилистики текста, основанное на длине слов. Данное решение легло в основу программы «Худломер» [13].

«Худломер» — это метод автоматической классификации функционального стиля текста на основе спектров длин слов. Программа позволяет определять следующие стили: разговорный стиль, стиль художественной литературы, газетно-информационный стиль, научно-деловой стиль.

1.4.5 Применения методов из теории вероятности и математической статистики

Предлагаемый в работе [14] метод основывается на учете статистики употребления пар элементов любой природы, идущих друг за другом в тексте (букв, морфем, словоформ и т. п.), т. е. на формальной математической модели последовательности букв (и любых других элементов) текста как реализации цепи Маркова. По произведениям, для которых достоверно определен автор, вычисляется матрица переходных частот употребления пар элементов (букв, грамматических классов слов и т. п.). Для каждого автора строится матрица переходных частот и оценивается вероятность того, что именно он написал

исследуемый текст. Автором неизвестного текста считается тот, для кого вычисленная оценка вероятности больше.

Примеры того, в виде каких пар можно представить текст:

а) пары букв в словах (в той форме, в которой они употреблены в тексте) и пробелах между ними;

б) пары букв в словах, приведенных к начальной форме;

в) пары наиболее обобщенных грамматических классов слов в их последовательностях в предложениях текста. К таким классам слов относят части речи и некоторые условные категории вроде «конец предложения», «сокращение» и др.;

г) пары менее обобщенных грамматических классов слов. К ним относятся такие семантико-грамматические разряды, как одушевленные и неодушевленные существительные, прилагательные качественные, относительные, притяжательные и т. п.

1.4.6 Метод выделения N-грамм

N-грамма — это последовательность из N идущих подряд слов в тексте. Использование N-грамм характерно тем, что позволяет учитывать порядок слов.

Для предложения «Наборы подряд идущих токенов» существуют следующие N-граммы:

- униграммы (N = 1): наборы, подряд, идущих, токенов;
- биграммы (N = 2): наборы подряд, подряд идущих, идущих токенов;
- триграммы (N = 3): наборы подряд идущих, подряд идущих токенов.

N-грамма имеет в основе своего использования математическую модель и определяется следующим образом: «N-граммой на алфавите V называют произвольную цепочку длиной N, например последовательность из N букв алфавита V одного слова, одной фразы, одного текста или, в более интересном случае, последовательность из грамматически допустимых описаний N подряд стоящих слов» [15].

Чем больше N, до которого будут найдены N-граммы, тем больше будет получено признаков. При этом увеличение параметра N может привести к переобучению. Например, если N — это длина текста, то у каждого документа будет уникальный признак, и алгоритм сможет переучиться под обучающую выборку.

N-граммы можно использовать не только на словах. В качестве токенов можно рассматривать отдельные символы в предложении. После нахождения таких токенов для них можно также вычислять N-граммы (буквенные N-граммы). Данный подход позволяет учитывать известные слова в незнакомых формах. Часто буквенные N-граммы используют вместе с N-граммами по словам.

Skip-граммы — это чуть более расширенный подход к использованию N-грамм. Более точно они называются k-skip-n-граммы, это наборы из N токенов, причём расстояние между соседними должно составлять не более K токенов [16]. Данный метод позволяет учитывать больше различных N-грамм в предложении, и, в связи с этим, как правило, используется в языковом моделировании в сочетании с другими подходами.

1.5 Извлечение признаков из текста

В данном разделе рассматриваются различные подходы к извлечению признаков из текста и определению значимости полученных признаков.

1.5.1 Счётчики слов

Данный подход подразумевает, что текст рассматривается как неупорядоченный набор слов. Более формально описать его можно следующим образом. Пусть всего в выборке N различных слов: $w_1, \dots, w_N$. В этом случае каждый текст кодируется с помощью N признаков, причём признак j — это доля

вхождений слова w_j среди всех вхождений слов в документе. Таким образом, текст кодируется вектором признаков, а сумма значений всех признаков составляет единицу.

Используя такой подход, имеет смысл обращать внимание на два нюанса. Первый — это стоп-слова. Это популярные слова, встречающиеся в каждом тексте (например, предлоги или союзы) и не несущие в себе никакой информации, которая могла бы выделить определенный текст среди всех остальных. Их стоит удалять ещё при предобработке, например, на этапе токенизации. Кроме того, имеет смысл удалять редкие слова. Если какое-то слово входит только в 1 или 2 текста, то, скорее всего, не получится его значимо учесть в модели и оценить, какой вклад оно вносит в целевую переменную.

1.5.2 TF-IDF

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте текста, являющегося частью корпуса текстов. Как и в подходе с использованием счетчика слов, считается, что если слово часто встречается в тексте, и оно не является стоп-словом, то, скорее всего, оно важно. Второе утверждение, которое лежит в основе TF-IDF: если слово встречается в других документах реже, чем в данном, то и в этом случае, скорее всего, оно важно для текста. По этому слову можно отличить этот текст от остальных. Если учесть описанные выше соображения, в результате получится подход TF-IDF (TF — term frequency, IDF — inverse document frequency) [17].

TF — частотность термина, которая измеряет, насколько часто термин встречается в документе. TF измеряется как отношение числа вхождений некоторого слова к общему числу слов документа. IDF — инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Таким образом, мера TF-IDF является произведением двух сомножителей TF и IDF.

Значение признака для слова w и текста x вычисляется по следующей формуле:

$$TF\text{–}IDF(x,w) = n_{dw} \log \frac{l}{n_w}, \quad (4)$$

где n_{dw} – доля вхождений слова w в документ d;

l – общее количество документов;

n_w – число документов в выборке, в которых слово w встречается хотя бы раз.

Если отношение $\frac{l}{n_w}$ велико, слово редко встречается в других документах, значение признака будет увеличиваться. Если слово встречается в каждом тексте, то значение признака будет нулевым ($\log \frac{l}{l} = 0$).

1.6 Классификация текстов

В данном разделе будут рассмотрены алгоритмы классификации текстов.

1.6.1 Метод «наивной» байесовской классификации

Наивный байесовский классификатор [18] основан на применении теоремы Байеса со строгими (наивными) предположениями о независимости и оперирует условными вероятностями. Данный алгоритм называют наивным по причине того, что он использует наивное допущение о том, что входящие в текст слова не зависят друг от друга.

Для построение наивного байесовского классификатора необходимо выбрать закон, по которому распределены данные. Обучение данного классификатора заключается в вычислении параметров распределения по примерам из тестового набора данных.

Если предположить, что данные распределены по закону Бернулли, то класс c^*, к которому относится текст t, вычисляется по следующей формуле:

$$c^* = argmaxP(c)\sum_{i=1}^{m} P(x|c)^{x_i(t)}, \quad (5)$$

где x – характеристики, по которым оцениваются тексты;

m – количество текстов;

$x_i(t)$ – величины, показывающие наличие i-ой характеристики в тексте;

c – метка класса;

$P(c), P(x|c)$ – параметры, вычисленные при обучении классификатора.

1.6.2 Метод k ближайших соседей

В основе метода k ближайших соседей [19] лежит гипотеза компактности векторного пространства, которая гласит, что: документы одного класса образуют в пространстве терминов компактную область, причём области разных классов не пересекаются [4]. Согласно рассматриваемому методу, для нахождения категории, соответствующей документу d, классификатор должен сравнить d со всеми документами из обучающей выборки L, то есть для каждого $d_z \in L$ вычисляется расстояние $\rho(d_z, d)$. Далее из обучающей выборки выбираются k документов, ближайших к d. Считается, что документ d принадлежит тому классу, который является наиболее распространенным среди соседей данного документа, то есть для каждого класса c_i вычисляется функция ранжирования:

$$CSV(d) = \sum_{d_z \in L_k(d)} \rho(d_z, d) \cdot \Phi(d_z, c_i), \quad (6)$$

где $L_k(d)$ – k документов из L ближайшие к d;

$\Phi(d_z, c_i)$ – известные величины, уже расклассифицированные по категориям документы из обучающей выборки.

1.6.3 Метод деревьев решений

Подход, использующий деревья решений, относится к символьным (то есть нечисловым) алгоритмам.

Для каждого дерева принятия решений основе обучающего множества узлами являются термины документов, листьями – метки классов, а на ребрах отмечены веса терминов [4]. Каждое дерево представляет собой ациклический граф.

Процессе классификации заключается в последовательных переходах от одного узла к другому в соответствии со значениями признаков объекта. Классификация считается завершенной, когда достигнут один из листьев дерева. Значение этого листа определяет тот класс, которому принадлежит рассматриваемый документ. На практике как правило используют бинарные деревья решений, в которых принятие решения перехода по ребрам осуществляется в зависимости от результата проверки наличия признака в документе [19].

Лесом решений [20] называют комитет (ансамбль) из нескольких деревьев решений. Использование нескольких деревьев приводит к улучшению качества прогнозирования и к лучшему пониманию закономерностей исследуемого явления. Разные деревья могут быть получены различными методами (или одним методом, но с различными параметрами работы), по разным выборкам наблюдений за одним и тем же явлением, путем привлечения различных характеристик. Для задачи классификации решение принимается по большинству результатов, выданных деревьями решений, а в задаче регрессии — по их среднему значению.

1.6.4 Метод опорных векторов

Метод опорных векторов [21] основан на разбиении векторного пространства документов разделяющей поверхностью на подпространства, где каждому подпространству соответствует только один класс. Алгоритм работает в предположении, что чем больше зазор классификации (расстояние между найденной поверхностью и ближайшей точкой данных), тем меньше будет средняя ошибка классификатора.

Для данного метода обязателен этап предобработки данных для преобразования текстовые данных в числовые векторы признаков. В процессе обучения осуществляются преобразования над пространствами, которые производятся с помощью оператора ядра. Такие преобразования приводят данные к такому виду, чтобы их можно было разделить гиперплоскостями на подпространства таким образом, чтобы большая часть объектов из обучающей выборки соответствовала своему классу, располагающемуся в конкретном подпространстве. Операторы ядра подразделяются на линейные и нелинейные. Далее при классификации нового текста его также необходимо перевести в векторное представление, после чего присвоить ему класс, найденный в зависимости от того, в каком подпространстве находится полученный вектор.

1.7 Выводы

В данном разделе были рассмотрены этапы обработки и анализа текстов на естественном языке. Проведен обзор существующих методов и решений определения авторского стиля. Были проанализированы алгоритмы классификации текстов.

2 Конструкторский раздел

В данном разделе рассматривается проектирование структуры программного обеспечения и требования к нему.

2.1 Функциональная модель метода определения признаков авторского стиля

2.2 Функциональная модель метода классификации

3 Технологический раздел

3.1 Выбор средств программной реализации

3.2 Формат входных данных

3.3 Интерфейс пользователя

4 Экспериментальный раздел

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мартыненко Г. Я. Стилеметрия: Возникновение и становление в контексте междисциплинарного взаимодействия. Часть 2. Первая половина XX

в.: Расширение междисциплинарных контактов стилеметрии. СТРУКТУРНАЯ И ПРИКЛАДНАЯ ЛИНГВИСТИКА, № 11, 2015, С. 9-28.

2. Фоменко В.П., Фоменко Т.Г. Авторский инвариант русских литературных текстов. Предисловие А.Т. Фоменко // Фоменко А.Т. Новая хронология Греции: Античность в средневековье. Т. 2. М.: Изд-во МГУ, 1996. С. 768-820.

3. Ермакович М.В. Автоматическое определение границ слова в русском тексте с помощью комплекса лингвистических правил. Компьютерная лингвистика и интеллектуальные технологии: по материалам международной конференции «Диалог 2017» Москва, 31 мая — 3 июня 2017.

4. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В.

5. Chomsky N. Three models for the description of language // IRE Transactions on Information Theory. — 1956. — Vol. 2, no. 3. — P. 113–124.

6. Tesnière L. Elements de syntaxe structurale. — Editions Klincksieck, 1959.

7. Mel’cuk I. A. Dependency syntax: theory and practice. — ŠUNY Press, 1988. — P. 428.

8. Abney S. P. Parsing by chunks // Principle-Based Parsing. — Kluwer Academic Publishers, 1991. — P. 257–278.

9. Federici S., Montemagni S., Pirrelli V. Shallow parsing and text chunking: a view on underspecification in syn- tax // Cognitive science research paper-university of Sus- sex CSRP. — 1996. — P. 35–44.

10. Хмелев Д.В. Классификация и разметка текстов с использованием методов сжатия данных. Краткое введение [электронный ресурс] - режим доступа: URL: http://compression.ru/download/articles/classif/intro.html (дата обращения: 11.12.2020).

11. Lai S., Xu L., Liu K., Zhao J. Recurrent Convolutional Neural Networks for Text Classification. // AAAI, 2015. P. 2267 – 2273.

12. Werbos P.J. Backpropagation through time: what it does and how to do it. // Proceedings of the IEEE 78, Harvard, 1990. Issue 10. P. 1550 – 1560.

13. Программы анализа и лингвистической обработки текстов. Русская виртуальная библиотека [электронный ресурс] - режим доступа: URL: https://rvb.ru/soft/catalogue/c01.html (дата обращения: 11.12.2020).

14. Кукушкина О. В., Поликарпов А. А., Хмелев Д. В. Определение авторства текста с использованием буквенной и грамматической информации // Проблемы передачи информации. М.: Наука, 2001. Т. 37, № 2. С. 96–108.

15. Гудков В. Ю., Гудкова Е. Ф. N-граммы в лингвистике // Вестник Челябинского государственного университета. 2011. № 24 (239). Филология. Искусствоведение. Вып. 57. С. 69–71.

16. Guthrie David, Allison Ben, Liu Wei, Guthrie Louise, Wilks Yorick. (2006). A Closer Look at Skip-gram Modelling. Proc. of the Fifth International Conference on Language Resources and Evaluation.

17. Shahzad Qaiser, Ramsha Ali. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications (0975 – 8887) Volume 181 – No.1, July 2018

18. Murphy K.P. Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series). The MIT Press, 2012. ISBN: 0262018020.

19. Батура Т.В. Методы автоматической классификации текстов // Программные продукты и системы. 2017. №1.

20. Hastie, T., Tibshirani R., Friedman J. Chapter 15. Random Forests // The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p. — ISBN 978-0-387-84857-0.

21. Tong S., Koller D. Support vector machine active learning with applications to text classification // The Journal of Machine Learning Research, 2002. Issue 2. P. 45 – 66.