САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Серебрякова Маргарита Владимировна

Дипломная работа

Обработка новостных сообщений в научной области

Научный руководитель,
ст. преподаватель
Попова С.В.

Санкт-Петербург
2016

Содержание

Введение

Важным аспектом в современном обществе является гонка технологий и постоянный рост темпов научного прогресса. Развитие существующего потенциала учёных, помощь в продвижении их идей, предоставление максимально комфортных условий для проведения исследований – все эти вопросы регулярно подымаются как отдельным государством, так и мировым сообществом в целом. С целью их разрешения формируется огромное количество фондов и программ, которые проводят всевозможные конкурсы и мероприятия с различными целевыми группами. Но встаёт проблема доступности данной информации для отдельного учёного. Объявления, как правило, публикуются на сайтах организаторов, т.е. данные весьма разрознены, и отдельному лицу сложно своевременно отслеживать новые публикации. При получении информации с различных источников результатом будет являться весьма большой объём документов, большая часть которых не будет интересна отдельному лицу.

Таким образом, видна актуальная задача сбора сообщений в научной сфере и их автоматического представления в виде удобном для быстрого фасетного поиска. Предполагается, что последнее позволит учёному настроить нужные фильтры и получить только тот набор объявлений, который интересен непосредственно ему.

Получение первичных результатов для разработки такой системы легло в основу данной дипломной работы, целью которой является создания аппарата автоматической классификации научных (в первую очередь конкурсных) объявлений по заданной системе классов ряда категорий. Для достижения выбранной цели решались следующие задачи:

- анализ значительного объёма данных для определения категорий, которые могут быть интересны пользователю (например, для кого сделано объявление, тип объявления, возрастная группы и т.д.) и выделение основных классов внутри категорий, например, по целевой группе: аспиранты, студенты, доктора наук, кандидаты наук и др.

- разработка тестовой и обучающей коллекций на основе определённых категорий и классов.
- изучение подходов к обработке естественно языка и задачи классификации, выбор стратегии обработки данных.
- изучение и имплементация двух алгоритмов машинного обучения, решающих задачу классификации
- оценить влияние использования различных подходов нормализации документов и значений ключевых параметров алгоритмов, определение лучших результатов.

В качестве материала по рассматриваемой теме были использованы объявления о конкурсах, которые ранее были получены с сайта УНИ СПБГУ. Данные были классифицированы и размечены вручную. Всего рассматривались четыре категории, количество документов в которых составило:

- Категория участников - 492
- Тип конкурс - 399
- Тип объявления - 329
- Масштаб конкурса - 297

В роли инструментария, который позволил написать необходимую программу для проведения исследования, использовалась библиотека алгоритмов машинного обучения Weka. В ходе работы были изучены такие общие принципы Weka, как область применения, какие задачи возможно решить при помощи данного пакета, доступные методы, структура входных данных, API (интерфейс программирования приложений).

Глава 1. Предметная область

1.1 Общие сведения о текстовом анализе

Основным экспериментальным материалом, использованном в работе являются документы, содержащие информацию на русском языке касательно проводимых мероприятий и новостей в научной сфере, а именно: объявления о самих конкурсах, общие сведения в данной области (например, проведение конференции или уведомления о внесённых изменения в организацию фондов), обращения к участникам мероприятий, подведение итогов конкурсов. Таким образом в качестве используемого материала в работе рассматривались тексты короткой длины, а задачей исследования являлся интеллектуальный анализ текста.

«Text mining» (текстовый анализ) является частью более общего раздела научных методов «Data mining» (извлечение данных, анализ данных). «Text mining» также можно свободно охарактеризовать как процесс обработки текста для извлечения информации, которая будет полезна для конкретных целей. По сравнению с типом данных, хранимых в базах данных, текст является неструктурированным, аморфным набором, с которым трудно работать алгоритмически, тем не менее, в современной культуре, текст является наиболее распространенным средством для официального обмена информацией.

В последние годы происходит сильный рост объёмов данных (в том числе и текстовых) как во всемирной паутине, так и в институциональных репозиториях. Именно поэтому важность автоматического извлечения конкретных данных из текстов, функция которых заключается в передаче и хранении фактической информации или мнений, не поддаётся сомнению, даже если результаты лишь частично успешные.

Главной задачей по факту является преобразование исходного текста к набору данных для дальнейшего анализа с помощью алгоритмов обработки данных. Значительную роль при этом играет способ представления

обрабатываемых документов, способы их предварительно обработки, определение требуемых мер и весовых функций.

Существует множество приложений текстовой обработки, включающие передовые исследования анализа и классификации новостных сообщений, электронных писем, фильтрации спама, иерархическое построение структуры топиков веб-страниц, автоматическое создание и обработки онтологии и конкурентной разведки. Каждое из этих приложений опирается на конкретное представление корпусов текста и множество достаточно надёжных, легко масштабируемых, не зависящих от языка алгоритмов. Вычислительные методы анализа больших текстовых корпусов можно разделить на две основные категории:

- статистические
- лингвистические

Статистические методы, как правило, строятся на базе статистической и вероятностной структуре и часто не принимают во внимание синтаксическую и семантическую структуру текста. Такие методы основаны на развитии математического представления текста.

Одним из самых популярных способов можно назвать матрицу слов («bag-of-word matrix»), когда каждый документ представляется вектором, содержащим частоту встречаемости каждого слова данного документа. В более общем виде данная матрица есть некоторое мультимножество слов, составленное без учёта грамматики и даже порядка слов, но сохраняя кратность.

Лингвистические методы, которые зачастую основаны на обработке естественного языка, пытаются разобрать документы на основе компьютерного представления человеческой речи. [14] Примерами могут послужить алгоритмы синтаксического анализа [8, 9, 5] и автоматической морфологической разметки [4]. Такой подход может потенциально привести к более точному представлению текста, лежащему в основе работы методов, что даёт дорогу широкому разнообразию приложений для обработки текста. Например, более детальная используемая структура текста может привести к

автоматическому выделению онтологий или обеспечить понятным для машины представлением контента.

Проведённое исследование опиралось на статистический метод, наряду с которым использовалась информация о частях речи слов для отбора слов во множество терминов.

1.1 Построение векторной модели

Векторная модель семантики (vector space model, VSM) была представлена Солтоном в 1975 г [13]. Новизна её состояла в том, чтобы использовать частоты слов в качестве ключевой информации для обнаружения семантической информации. Представление каждого компонента корпуса в качестве точки в многомерном пространстве (вектора в векторном пространстве) заключает в себе основную идею VSM. Здесь размерность пространства равна мощности множества признаков модели. Координатами являются значения этих признаков, которые рассчитываются определённым образом для каждого документа. Например, множеством признаков могут являться все слова документа, а за их значения приниматься частоты слов для конкретного документа. Семантически схожим текстовым документам соответствуют близко расположенные точки пространства.

Встречаются три наиболее популярных вида матриц [3]:

- Термин-документ. Показывает сходство между документами. Для этого каждое мультимножество слов документа представляется вектором значений, полученным следующим образом: рассмотрим мультимножество {a, a, b, c, c, c}, где буквы a, b, c отвечают за некоторое слово, тогда соответствующий вектор будет иметь вид {2, 1, 3}. То есть на первом месте стоит частота слова a, на втором слова b и на третьем - c. Порядок элементов в мультимножестве не играет роли, но при построении векторов последовательность слов-признаков должна быть постоянна. Тогда коллекцию документов можно представить матрицей, где строки относятся к

определённому термину, а столбцы соответствуют некоторому документу.

- Слово-контекст. Рассматривает схожесть между словами. Матрица подобна модели термин-документ, но здесь столбцами могут быть не обязательно документы, но и главы, абзацы, предложения.
- Пара-модель отслеживает схожесть отношений. Строками матрицы являются пары слов, а столбцами - различные отношения между парами слов.

В проведённой научный работе применялась матрица термин-документ.

2 Лингвистическая обработка

Пусть необходимо исследовать достаточно большой объём документов, написанных на естественном языке. Предварительным этапом к построению VSM будет предобработка текста, которую можно разделить на три типа [3]:

1. Токенизация – принятие решения о том, что будет являться терминами (признаками) и способе их извлечения из исходного текста
2. Нормализация – приведение всех слов к некоторой нормальной форме за счёт единого (как правило нижнего) регистра и стеммирования.
3. Комментирование (автоматическая морфологическая разметка и синтаксический анализ) - создание для каждого слова метки, указывающей на принадлежность его к определенной части речи.

Токенизация

Токенизация не всегда представляет собой простую задачу. Инструмент-разметчик должен знать, как работать с пунктуацией, распознавать переносы, а также в отдельных задачах уметь распознавать такие составные термины, как «генетический алгоритм» или «нейронная сеть». Зачастую используются различные списки «стоп-слов» для отсеивания терминов с высокой степенью встречаемости в языке и не несущих важной информации (предлоги, союзы, местоимения).

Нормализация

Важность нормализации обусловлена тем, что различные части текста могут иметь один и тот же смысл. В связи с этим фактом приведение всех слов к единой форме играет столь значимую роль.

Во-первых, все символы приводятся к одному регистру. Однако стоит учитывать, что встречаются ситуации, когда слова разного регистра имеют разный смысл. Примером может послужить аббревиатура СТО (Специальная теория относительности), которая после понижения регистра будет означать числительное «сто».

Необходимо также учесть тот факт, что слово имеет различные морфологические формы, которые не меняют его смысл, а лишь связывают его с другими словами в предложении. Поэтому замена флективных форм в документе на “нормальные” (например, именительный падеж, единственное число, мужской род для существительных) положительно сказывается на качестве классификации. Простейшим примером служит процедура стемминга (англ. «stemming»). Данный метод заключается в приведении исходного слова к некоторой форме (стему) путём отбрасывания максимального по длине окончания, которое находится из заранее составленного набора. В виду того, что объём указанного набора окончаний невелик, алгоритмы стемминга показывают хорошее время работы. Недостаток метода заключается в возникновении большого числа ошибок, при которых не все словоформы приводятся к одному стему (understemming) или разные по смыслу слова приравниваются к одинаковой нормальной форме (overstemming). Алгоритмы стемминга изучаются в информатике приблизительно со второй половины 20 века.

Исследования показали, что стемминг, а значит и нормализация, приводит к увеличению полноты (Recall) поиска и одновременно уменьшает его точность (Precision) [2, 16]. За счёт того, что различные слова по факту расцениваются системой как синонимы, то возрастает степень схожести между документами, благодаря чему находится большее число релевантных

классу документов и полнота повышается. Обратный эффект заключается в том, что при указанном упрощении слово может потерять изначально заложенный смысл, что приводит к понижению точности. Данная проблема решается путем применения более сложных алгоритмов, таких как лемматизация или методов, основанных на правилах словообразования (например, алгоритм Портера для английского языка [6])

Основная идея лемматизации, которая применяется в данной работе с помощью сервиса MyStem [19], заключается в применении словаря, где каждой флективной форме слова указана “нормальная” (лемма). Сложность заключается в составлении таких словарей и достаточно большом времени поиска нужного слова.

Комментирование

Общая схема данного этапа представляет собой добавление меток с информацией о частях речи, всевозможных значениях слова и синтаксическом анализе предложения. Что позволяет разрешать случаи смысловой неоднозначности и определяются зависимости между словами.

3 Математическая обработка

Все тексты прошли лингвистическую предобработку в процессе которой были токенизированы, удалены стоп-слова, все полученные слова лемматизированы, приведены к нижнему регистру. Следующим шагом является составление матрицы частот. Затем рассчитываются веса её компонент, так как часто встречаемые слова (с высокой кратностью) несут малый объём информации. Далее имеющуюся матрицу можно «сгладить»: уменьшить количество «шумов» и нулевых элементов (в проведённом исследовании не применяется).

Построение частотной матрицы

Каждый элемент матрицы частот представляет собой числовое значение v, соответствующее возникновению определённого события: конкретный

объект (термин, слово) встретился в определённой ситуации (документе, контексте) *v* раз. В теории построение матрицы частот является простой задачей подсчета числа появления событий.

Взвешивание признаков.

Имеется множество подходов для определения весов признаков. Общим моментов является то, что по некоторому алгоритму подсчитываются для каждого слова числовые «веса», которые показывают насколько информативен данный термин с точки зрения решения какой-либо задачи.

Одним из самых распространённых способов оценки веса терма для матриц термин-документ является TF-IDF (частота термина × обратная частота документа) семейство весовых функций (Спарк Джонс, 1972).

Формулы для расчёта TF (term frequency) И IDF (inverse document frequency):

$$TF(t,d)=\frac{n_d}{\sum_D n_i}$$

где t – какое-либо слово в документе d, D – множество всех документов корпуса, n_d и n_i – сколько раз слово появилось в рассматриваемом и *i*-ом документах соответственно.

Для каждого уникального слова в пределах одного корпуса документов существует единственное значение IDF. Значение IDF находится из следующего выражения:

$$IDF(t,D)=\log\frac{|D|}{|(d_i \ni t)|}$$

где

$|D|$ – число документов в коллекции;

$\left|(d_i \ni t)\right|$ – сколько всего документов содержат *t*.

Таким образом, мера TF-IDF получается перемножением двух сомножителей:

$$TF-IDF(t,d,D)=TF(t,d)\times IDF(t,D)$$

Элемент получает высокий вес, когда соответствующее слово часто фигурирует в рассматриваемом документе (т.е. достаточно большое значение TF), но этот термин редко встречается в корпусе в целом (т.е. DF мала, и, таким образом, получается высокая IDF). Учёт IDF уменьшает вес широкоупотребительных слов.

Солтон и Бакли (1988) определили большое семейство TF-IDF весовых функций и оценивали их по результатам решения задач поиска информации, демонстрируя, что TF-IDF взвешивание может принести значительные улучшения по сравнению с обычной частотой.

Другим видом взвешивания, часто применимым в сочетании с TF-IDF, является нормализация длины (англ. length normalization) [17]. В информационном поиске, если подобная оптимизация отсутствует, поисковые системы, как правило, имеют уклон в пользу более длинных документов. Нормализация длины корректирует это смещение.

Оптимизация матрицы

Самый простой способ для повышения производительности поиска информации является ограничение количества компонентов вектора. Сохранение только элементов, представляющих наиболее часто встречающиеся слова контента, является одним из таких способов. Эвристики сглаживания матриц, которые основаны на свойствах весовых функций, представленных на предыдущем этапе математической обработки, позволяют не только сохранить смысловую дискриминацию, но и повысить производительность вычислений.

В 1990 г. Был найден способ улучшить измерение подобия математической операцией на матрице термин-документ (*Y*) на основе знаний

из линейной алгебры, который заключается в применении компактного сингулярного разложения. Данный метод позволяет представить Y как произведение матриц UEV^T. Используя k сингулярных чисел можно получить матрицу ранга k максимально аппроксимирующую исходную. Тем самым уменьшается размерность, а также понижает шум и степень разрежённости матрицы Y [10].

2.1 Задача классификации данных

Одной из задач машинного обучения является классификация данных наряду с кластеризацией, регрессией, ранжированием, поиском ассоциативных правил и др. [19]. Необходимость классификации возникает в различных сферах человеческой деятельности. В самом широком смысле данный термин может подразумевать любой случай, в котором принято решение или сделан прогноз на основе имеющейся информации, а процедура классификации есть соответствующий метод, который позволяет производить подобные рассуждения в новых ситуациях. В этой работе мы будем рассматривать более строгое толкование. Будем считать, что проблема классификации касается построения процедуры, которая будет применяться к последовательности случаев, в которых каждый новый случай должен быть отнесен к одному из набора предопределенных классов на основе наблюдаемых признаков или особенностей.

Разработка процедуры классификации по набору данных, для которых классы заранее известны называется обучением с учителем. Указанный подход концептуально отличается от неконтролируемого обучения или кластеризации, в котором классы (кластеры) извлекаются из самих данных.

Примерами, для которых задача классификации является фундаментальной, могут послужить процедуры для сортировки писем на основе машинного чтения почтовых индексов, определение кредитоспособности лиц на основе финансовой и личной информации, а также предварительный диагноз болезни пациента для того, чтобы выбрать немедленное лечение в ожидании окончательных результатов осмотра. На самом деле, некоторые из наиболее актуальных проблем, возникающих в науке, промышленности и торговли можно рассматривать как задачу классификации или поиска решения с использованием сложных и зачастую весьма обширных данных. Можно выделить три основных направления

исследований по классификации данных: статистические анализ, машинное обучение и нейронные сети. [11]

В проведённом исследовании применялись инструменты машинного обучения, изучающего построение алгоритмов, которые могут обучаться и делать в дальнейшем прогнозы для вновь полученных данных.

2.2 Алгоритмы классификации

В данном исследовании были использованы два алгоритма машинного обучения с учителем: C4.5 и Наивный байесовский метод.

5 Метод C4.5

C4.5 - это один из алгоритмов построения дерева решений, который представляет собой алгоритм ID3 [7] с дополнительными полезными возможностями, а именно: возможна работа с числовыми атрибутами, есть функция обрезки ветвей, не несущих полезной информации, допускается неполная информация об объектах (значения признаков могут быть пустыми) и др. Тот факт, что C4.5 может работать с числовыми признаками, позволил применить C4.5 для решения поставленной задачи. В пакете Weka данный алгоритм реализован через класс J48.

Для имплементации C4.5 исходные данные и их структура должны отвечать некоторым параметрам:

1. Объекты из предметной области должны быть описаны через конечное число признаков (атрибутов). Необходимо, чтобы набор атрибутов оставался постоянным для всех примеров из тренировочной выборки. Все признаки обязательно должны иметь либо дискретное, либо, как уже было упомянуто выше, числовое значение.
2. Входные данные обязаны быть полными относительно классов. То есть для каждого объекта в выборке должен быть однозначно (не

допускается вероятностная оценка) указан класс, к которому он относится.

3. Также важно, чтобы множество классов имело конечное число значений.
4. Количество признаков должно быть существенно больше числа классов.

Алгоритм построения дерева принятия решений.

Допустим, что исходные данные содержат множество T объектов-примеров, а также набор атрибутов A. C – множество классов, c_i – элемент данного множества, $i=\overline{1,k}$.

Процесс построения дерева происходит сверху вниз, т.е. сначала находится корень дерева, затем его потомки и так далее. На начальном этапе мы имеем только корень, с которым связано всё множество T. Затем выделяется атрибут, который лучше всего классифицирует примеры (с максимальной информационной выгодой). По его значениям распределяются объекты исходного множества. Таким образом в результате разбиения мы получаем n узлов-потомков T_j , где n – число значений атрибута (в случае числового признака используются пороги значений или диапазоны).

Далее процесс повторяется рекурсивно для всех полученных подмножеств и т.д. Данная процедура прерывается, если у вновь полученного узла все относящиеся к нему примеры принадлежат одному классу, тогда он становится листом, а данный класс указывается в качестве решения. Также происходит, если на некотором шаге после разделения множества по некоторому признаку среди потомков оказалось пустое множество (то есть ни один из объектов не попал в узел после проверки), то ассоциированный с ним узел помечается как лист, а решением является наиболее вероятный класс для его предка.

Классификация нового объекта заключается в обходе дерева начиная с корня. В результате проверок классификатор попадает в некоторый лист, а связанное с ним решение указывается в качестве класса для объекта.

Критерий выбора атрибута, по которому происходит разбиение.

Пусть рассматривается некоторый атрибут $\hat{a}$ (всего их в выборке m штук), который принимает n значений $v_1, v_2 \ldots v_n$. Соответственно после распределения объектов по данной проверке будут получены n новых узлов T_1 , T_2 … T_n . Для принятия решения о выделении «лучшей» проверки для текущего множества в распоряжении имеется лишь информация о распределении классов в исходном узле и полученных потомках по выбранной проверке.

Вероятность того, что произвольно выбранный объект из некоторого множества M будет относиться к классу c_i рассчитывается по классической формуле:

$$P(c_i|M) = \frac{num(c_i, M)}{|M|},$$

где $num(c_i, M)$ – количество документов из множества M, принадлежащих классу c_i .

Далее используется одна из версий формулы Хартли, которая гласит, что информационный размер сообщения о каком-либо событии непосредственно зависит от вероятности возникновения данного события [1].

$$I = \log_2(\frac{1}{P})$$

Тогда оценку количества информации, необходимой для установления класса объекта из исходного множества T, можно представить формулой энтропии выбранного множества:

$$¿ \frac{num(c_i, T)}{¿} ¿ T \vee ¿ * \log_2 ¿ \frac{num(c_i, T)}{¿} ¿ T \vee ¿$$

$$I(T) = -\sum_{i=1}^{k} ¿$$

Для полученных после разбиения по проверке $\hat{a}$ подмножеств T_j применяется следующее выражение:

¿ $T_j \vee \frac{¿}{¿ T \vee ¿ * I(T_j)}$

¿

$$I_{\hat{a}}(T) = \sum_{j=1}^{n} ¿$$

Тогда итоговое значение «information gain» критерия выбора проверки рассчитывается для всех атрибутов по формуле

$$IG(\hat{a}) = I(T) - I_{\hat{a}}(T)$$

В виду того, что энтропия увеличивается с приближением распределения классов к равновероятным событиям, для узла T в качестве проверки выбирается тот признак (атрибут), который максимизирует значение данного выражения, т.к. необходимо разбить элементы таким образом, чтобы один из классов имел существенно большую вероятность относительно других (понизить неопределённость данных).

$$a = \underset{\hat{a} \in A}{argmax}\, IG(\hat{a})$$

В данной работе все атрибуты являются числовыми, поэтому необходимо выбрать порог значений, по которому все элементы будут делиться на два множества. Т.к. количество значений признака конечно, то можно допустить, что случайно взятый числовой признак $\hat{a}$ принимает значения $\{v_1, v_2 \ldots v_n\}$. Требуется расположить значения в порядке возрастания или убывания. Далее последовательно рассматривается пара $v_i u v_{i+1}$, и их среднее значение th_i используется для разбиения всех объектов на две группы $T^1_{th_i} u T^2_{th_i}$: примеры, у которых значение выбранного атрибута больше th_i, и те, у которых оно меньше. Для каждого признака находится порог, по которому получаются наиболее определённые подмножества:

¿ $T^2_{th_i} \vee \frac{¿}{¿ T \vee ¿ * I(T^2_{th_i})}$

$$IG(a, th_i) = ¿ I(T) - \frac{|T^1_{th_i}|}{|T|} * I(T^1_{th_i}) - ¿$$

$$th_a = \underset{i=\overline{1,n-1}}{argmax} ¿$$

На последнем шаге находится непосредственно атрибут, дающий самое высокое число «information gain».

$$a = \underset{\hat{a} \in A}{argmax}\, IG(\hat{a}, th_{\hat{a}})$$

Основным достоинством алгоритма C4.5 является его простота, но имеется и ряд недостатков, а именно [12]:

- Не гарантирует оптимальность решения, т.к. может привести лишь к локальной оптимизации, т.е. метод относится к «жадным алгоритмам» (англ. greedy algorithm).
- Достаточно часто происходит перенасыщение метода: отличные показатели для объектов из тренировочной выборки, но плохая классификация новых случаев. То есть алгоритм, не обучается, а лишь запоминает исходные примеры.

6 Наивный байесовский метод.

Данный метод основан на теореме Байеса, которая заключается в формуле вычисление апостериорной вероятности.

$$P(c \vee d) = \frac{P(d \vee c)P(c)}{P(d)}$$

- $P(c|d)$ - вероятность что документ d принадлежит классу c, которую необходимо найти.
- $P(d|c)$ - вероятность встретить документ d среди всех документов класса c
- $P(c)$ - безусловная вероятность встретить документ класса c.
- $P(d)$ - безусловная вероятность документа d в корпусе документов.

Суть метода заключается в поиске максимума функции апостериорной вероятности.

То есть решением будет наиболее вероятный класс.

$$C = \underset{c \in C}{argmax} \frac{P(d \vee c)P(c)}{P(d)}$$

Вероятность $P(d)$ не зависит от выбранного класса, поэтому исследуемая функция сводится к

$$C = \underset{c \in C}{argmax} P(d \vee c)P(c)$$

«Наивность» классификатора заключается в том, что при расчёте используется аппроксимация вероятности $P(d|c)$, которая представляет собой произведение условных вероятностей всех слов из данного документа.

$$P(d \vee c) \approx \prod_{i=1}^{n} P(w_i \vee c)$$

где w_i соответствует некоторому признаку (слову).

Следовательно, получаем следующую формулу:

$$C = \underset{c \in C}{argmax} \left(P(c) \prod_{i=1}^{n} P(w_i | c) \right)$$

То есть предполагается, что вероятности слов не связаны друг с другом, что является абсолютно неверным предположением для естественного языка.

Из формулы видно, что при большом объёме документа перемножается много маленьких чисел. Поэтому вполне возможна ситуация,когда порядок полученной вероятности выйдет за пределы разрядной сетки. Чтобы этого избежать можно прологарифмировать обе части уравнения:

$$C = \underset{c \in C}{argmax} \left(\ln P(c) + \sum_{i=1}^{n} \ln P(w_i | c) \right)$$

Данная формула справедлива на основании монотонности логарифмической функции. Маленькие значения перейдут в отрицательные, но их абсолютные значения будут значительно больше, что предотвратит арифметическое переполнение. В данном случае взят наиболее встречаемый натуральный логарифм, но при решении задачи основание логарифма роли не играет.

Расчёт $P(c)$ и $P(\ w_i\ |c)$ осуществляется по тренировочной коллекции.

$$P(c) = \frac{D_c}{D}$$

где D_c - мощность класса c (количество документов данного класса), а D – общее количество документов в коллекции.

В случае взвешенных признаков, имеющих численные значения (например, вес TF-IDF, который применялся в проведённом исследовании) $P(\ w_i\ |c)$ принимает несколько иной вид $P(\ w_i\ =\ \widehat{w}_i\ |c)$, то есть вероятность того, что признак w_i примет значение $\widehat{w}_i$ для документов класса c.

В используемом программном пакете Weka эта вероятность рассчитывается двумя способами, которые также рассматриваются при проведении эксперимента:

$$P(w_i = \widehat{w}_i | c) = g(x; \mu_c; \sigma_c),$$

где $g(x,; \mu_c; \sigma_c) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$ – функция плотности нормального распределения,

параметры μ_c и σ_c находятся из обучающей выборки.

Второй способ заключается в использовании функции ядра (kernel function) [15]

В данной работе документы представляются вектором значений по множеству слов-признаков, составленном по обучающей выборке. То есть при классификации нового документа слова, ранее не встречающиеся в тестовой коллекции, не учитываются. Поэтому необходимость сглаживания Лапласа или любого другого отпадает.

Преимуществами Наивного байесовского классификатора служат простота реализации и низкие вычислительные. Главный недостаток вытекает из фигурирующей гипотезы о независимости признаков классификации - относительно низкое качество классификации в большинстве реальных задач.

6.1 Метрики оценивания качества

С целью выявить наиболее удачные инструменты классификации, такие как используемый алгоритм с меняющимися параметрами и предварительную обработку документов (нормализация, отбрасывание стоп-слов), полученные результаты были оценены по следующим критериям:

- Точность (англ. precision) – показывает, какая часть от тех документов, которых классификатор посчитал соответствующими рассматриваемому классу действительно ему принадлежат.
- Полнота (англ. recall) – характеризует способность классификатора находить как можно больше объектов, относящихся к классу.
- F-мера – является объединением первых двух характеристик, представляет собой среднее гармоническое точности и полноты.

Данные метрики также применяются в области информационного поиска для оценки качества работы поисковых систем.

Если рассматривать документы на принадлежность некоторому классу X, то все полученные результаты категоризации можно представить в виде

таблицы, которая называется «таблицей сопряжённости» или «матрицей неточностей» (confusion matrix):

		Ожидалось	
		1	**0**
Получили	**1**	**tp** (true positive)	**fp** (false positive)
	0	**fn** (false negative)	**tn** (true negative)

Табл.1 Матрица неточностей для класса X

Здесь 1 означает, что элемент принадлежит X, 0 – не принадлежит.

- Истинно-положительный (**true positive**) - классификатор принял верное решение о том, что данный объект(документ) относится к классу.
- Ложно-положительный (**false positive**) - получена некорректная информация о принадлежности документа классу X.
- Ложно-отрицательный (**false negative**) - объект соответствует классу, но на выходе получили обратный результат
- Истинно-отрицательный (**true negative**) - классификатор правильно определил документ как не относящийся к X.

Например, при оценивании возвращаемых поисковой системой результатов по некоторому запросу классом X являются релевантные документы. Данный подход использовался и в проведённой работе, который будет более подробно рассмотрен в третьей главе.

Формулы, по которым рассчитываются метрики «точность» и «полнота» в виду введённых обозначений:

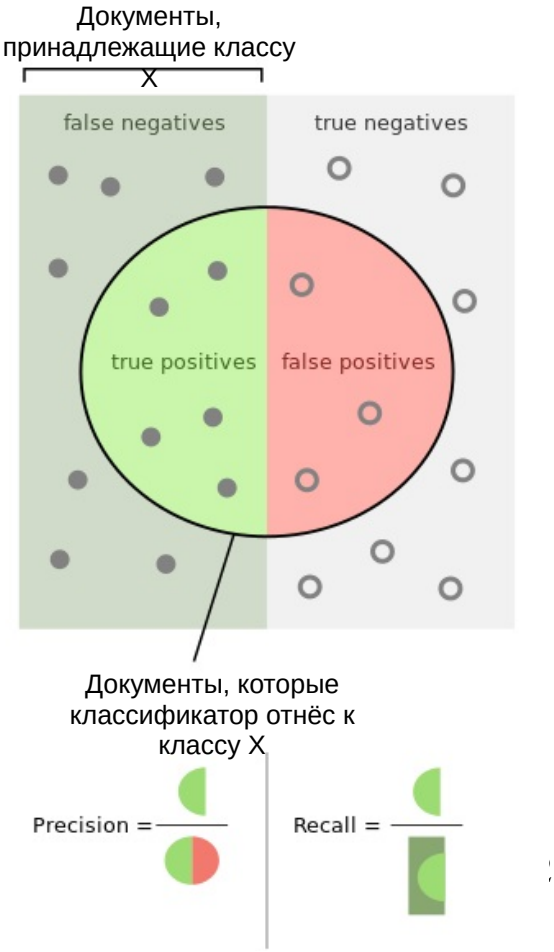

Рисунок 3.1 Точность и полнота

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$F_{measure} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

3.1 Разработка схемы классификации

В виду того, что в качестве метода достижения поставленной цели принята классификация с учителем, первым шагом необходимо подготовить обучающую коллекцию. В первую очередь требовалась схема классификации, которая была бы наиболее близка к реальной картине восприятия объявлений о конкурсах в научной сфере человеком. Объектами классификации являются текстовые документы небольшого размера. После изучения исходных данных, полученных предварительно с сайта УНИ СПбГУ было выделено 4 категории:

1) категория участников
   - другое: организации, журналисты, работники компаний, субъекты малого предпринимательства, эксперты, учащиеся школ/колледжей и т.п.
   - доктора наук
   - кандидаты наук
   - молодые учёные, исследователи, преподаватели
   - молодые доктора наук
   - молодые кандидаты наук
   - аспиранты
   - студенты
2) тип конкурса
   - другое: проекты проведения различных научных мероприятий (конференций, симпозиумов, экспедиций, школ), творческие конкурсы (эссе, дизайна, видеороликов и др. форматов) и т.п.
   - научных/исследовательских проектов, гранты на проведение исследований
   - премии и стипендии (именные, правительства РФ и др.), а также конкурсы уже выполненных научных работ.
   - научная мобильность (обучение и выполнение исследовательской работы за рубежом, стажировки, школы и конференции)

- инновационных проектов, стартапов.

3) тип объявления
   - о конкурсе
   - объявление результатов
   - общая информация: изменения в организации различных фондов, ФЦП, объявления о научных мероприятиях не на конкурсной основе (конференции, научные школы и др.)
   - информация для участников или победителей конкурсов, руководителей проектов, грантодержателей
4) масштаб конкурса
   - не указано
   - международный
   - российский
   - внутри вузовский
   - региональный, городской

3.2 Организация обучающего и тестового множеств

Первым шагом составлялась тестовая коллекция следующим образом: из множества объявлений последовательно отбирался и размечался набор из 150 текстов. Последовательный выбор производился с целью максимального приближения к реальной ситуации и текстам, которые нужно будет классифицировать. Тем не менее, для некоторых классов документов оказалось недостаточное количество (за минимум было взято 15 объектов). Поэтому они были дополнены документами, найденными по ключевым словам и проверенными вручную на принадлежность классу. Например, для первой категории класса «5» ключевыми словами будут: «аспирант», «аспирантура» и «phd». Аналогичным последнему способу методом разрабатывалась обучающая выборка.

Изначально предполагалось, что один текст в некоторой категории может принадлежать нескольким классам. Как показал результат ручной разметки, это характерно только для первой категории. Для остальных количество документов с множественным наследованием не превысило семи процентов. То есть, к примеру, лишь три документа из восьмидесяти, относящихся к классу «конкурс проектов, грантов» категории «тип

конкурса», также соответствовали другому классу этой категории. В результате в тестовой и тренировочной коллекции лишь для классов категории участников были сформированы две дополнительные группы «принадлежит» и «не принадлежит».

Тренировочная выборка представляет собой иерархическую систему текстовых файлов, в которой на верхнем уровне размещены папки-категории, каждая из которых содержит папки all и classes. В каталоге all хранятся все примеры данной категории, файлы не повторяются и имеют уникальные имена, которые в дальнейшем используются для подсчёта TF-IDF для каждого слова и построению VSM документа. В папке classes расположен уровень папок-классов, включающие в себя (лишь в случае первой категории) два подкласса: 1 – документы соответствуют указанному классу, 0 – не соответствуют. В каталогах 0 и 1 собрано приблизительно равное (разница менее 5 процентов от общего числа) количество файлов-примеров (имена совпадают с папкой all), чтобы у классификатора не было некорректной информации о количественном преобладании одного класса над другим. Все файлы имеют расширение .txt и кодировку UTF-8.

Структура тестовой выборки отличается лишь отсутствием папки all в первой категории (т.к. при расчёте весов слов для выбранного документа учитываются лишь объекты–примеры из обучающего множества, поэтому взвешивание термов происходит непосредственно перед определением класса), а также отсутствием жёсткой связи между подклассами 0 и 1, они заполняются по результатам ручной сортировки файлов.

Далее приведены некоторые сведения по полученным обучающему и тестовому множествам.

7 Общие сведения по полученным выборкам

Нормализация	Средняя количество слов в документах	Количество уникальных слов
Отсутствует	217	53418

MyStem (все слова)	236	12195
MyStem (сущ. & прил. & глаголы)	188	8643
MyStem (сущ. & прил)	162	6897

Таблица 3.1 Тренировочная коллекция. Общие сведения

Категория	Количество документов
категория участников	492
тип конкурса	399
тип объявления	329
масштаб конкурса	297

Таблица 3.2 Тренировочная коллекция. Количество документов по категориям

Категория	Количество документов
категория участников	100
тип конкурса	145
тип объявления	158
масштаб конкурса	134

Таблица 3.3 Тестовая коллекция. Количество документов по категориям

8 Информация по полученным классам в тестовой выборке

Класс	Количество документов в подклассе 1 (принадлежит)
другое	31
доктора наук	16
кандидаты наук	15
молодые учёные	39
молодые д.н.	20
молодые к.н	25
аспиранты	39
студенты	31

Таблица 2.4 Первая категория. Количество документов по классам

Класс	Количество документов
другое	23
проекты & гранты	41
премии, стипендии & выполненные работы	37
научная мобильность	29
Стартапы & инновационные проекты	15

Таблица 3.5 Вторая категория. Количество документов по классам

Класс	Количество документов
объявление о конкурсе	111
объявление результатов	16
общая информация	16
информация для участников	15

Таблица 3.6 Третья категория. Количество документов по классам

Класс	Количество документов
не указано	20
международный	41
российский	43
внутривузовский (СПбГУ)	15
городской & региональный	15

Таблица 3.7 Четвёртая категория. Количество документов по классам

9 Информация о размерах классов в обучающей выборке

Класс	Количество документов

	в подклассе 1 (принадлежит)
другое	168
доктора наук	47
кандидаты наук	42
молодые учёные	219
молодые д.н.	95
молодые к.н	105
аспиранты	206
студенты	195

Таблица 3.8 Первая категория. Количество документов по классам

Класс	Количество документов
другое	76
проекты & гранты	61
премии, стипендии & выполненные работы	75
научная мобильность	97
Стартапы & инновационные проекты	90

Таблица 3.9 Вторая категория. Количество документов по классам

Класс	Количество документов
объявление о конкурсе	154
объявление результатов	63
общая информация	41
информация для участников	71

Таблица 3.10 Третья категория. Количество документов по классам

Класс	Количество документов
не указано	61
международный	74
российский	61
внутривузовский (СПбГУ)	45
городской & региональный	46

Таблица 3.11 Четвёртая категория. Количество документов по классам

Глава 4 Проведение экспериментов

10.1 Описание шагов предобработки данных

После того как были организованы обучающая и тестовая коллекция, все документы, которые в них содержатся следовало предварительно обработать и привести к одному виду. Первым шагом проводилась токенизация, затем лемматизация слов и определение частей речи с использованием сервиса MyStem [20], который показал сравнительно неплохие результаты на корпусе текстов на русском языке - ruscorpora. Данный программный продукт проводит морфологический анализ текста на русском языке, а также присутствует возможность построения гипотетических разборов для слов, не входящих в словарь.

Рассматривались 3 случая представления документов:

1. Все слова в исходном виде
2. Все слова после лемматизации
3. Лемматизированные существительные, прилагательные и глаголы

Эксперимент должен был позволить определить лучший подход к представлению документов.

Далее программа составляет вектор уникальных слов корпуса документов, который будет использоваться классификатором в качестве множества признаков. Затем для каждого слова всех документов рассматриваемой категории рассчитывается вес TF-IDF, тем самым получаем следующее представление объекта выборки (документа): вектор значений, где на *i*-ой позиции стоит подсчитанный вес TF-IDF слово, соответствующее этой позиции вектора признаков. Были подготовлены различные наборы стоп-слов: Яндекса, стандартный и расширенный списки Вордстата, а также пустой список (слова не отбрасывались)

10.2 Результаты экспериментов

Исходным шагом для проведения экспериментов стало определение принципа выбора признаков векторной модели документов. Для этого выставлялся минимальный порог длины слова, а также использовались различные списки стоп-слов (Яндекс, Вордстат, расширенный Вордстат) Опытным путём не было выявлено подхода, который бы стабильно показывал лучшие результаты. Показатели варьировались от класса к классу и при применении различных алгоритмов классификации. Тем не менее удалось установить, нижний порог длины слова. Было принято решение использовать список стоп-слов Яндекса и в качестве признаков использовать слова, длины которых более двух символов.

Следующим шагом следует определить лучший случай представления документов. Рассматриваются все категории и оба метода классификации.

Первая категория

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,432	*0,613*	0,507	0,556	0,484	**0,517**
доктора наук	0,302	*0,813*	**0,441**	0,600	0,188	0,286
кандидаты наук	0,306	*0,733*	**0,431**	1,000	0,200	0,333
молодые учёные	0,708	0,447	0,548	0,543	*0,658*	**0,595**
молодые д.н.	0,257	0,450	0,327	0,300	*0,900*	**0,450**
молодые к.н	0,387	0,480	**0,429**	0,279	*0,680*	0,395
аспиранты	0,788	0,667	**0,722**	0,566	*0,769*	0,652
студенты	0,792	0,613	**0,691**	0,571	*0,645*	0,606
Среднее по категории	0,497	0,602	**0,51**	0,552	0,566	0,479

Табл. 4.1 Первая категория. Без нормализации

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,538	*0,677*	**0,600**	0,533	0,516	0,525
доктора наук	0,326	*0,938*	**0,484**	0,333	0,188	0,240

кандидаты наук	0,302	*0,867*	**0,448**	0,800	0,267	0,400
молодые учёные	0,781	0,658	**0,714**	0,600	*0,711*	0,651
молодые д.н.	0,320	0,400	0,356	0,297	*0,950*	**0,452**
молодые к.н	0,406	0,520	**0,456**	0,305	*0,720*	0,429
аспиранты	0,707	0,744	**0,725**	0,579	*0,846*	0,688
студенты	0,759	*0,710*	**0,733**	0,559	0,613	0,585
Среднее по категории	0,517	*0,689*	**0,565**	0,5	0,601	0,496

Табл. 4.2 Первая категория. Все леммы

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,395	*0,548*	0,459	0,471	0,516	**0,492**
доктора наук	0,359	*0,875*	**0,509**	0,417	0,313	0,357
кандидаты наук	0,294	*1,000*	**0,455**	0,571	0,267	0,364
молодые учёные	0,786	0,579	**0,667**	0,574	*0,711*	0,635
молодые д.н.	0,290	0,450	0,353	0,300	*0,900*	**0,450**
молодые к.н	0,452	0,560	**0,500**	0,300	*0,720*	0,424
аспиранты	0,763	0,744	**0,753**	0,564	*0,795*	0,660
студенты	0,786	*0,710*	**0,746**	0,571	0,645	0,606
Среднее по категории	0,516	*0,683*	**0,555**	0,47	0,608	0,499

Табл. 4.3 Первая категория. Сущ. & прил. & глаголы

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,467	*0,677*	**0,553**	0,516	0,516	0,516
доктора наук	0,308	*0,750*	**0,436**	0,385	0,313	0,345
кандидаты наук	0,298	*0,933*	**0,452**	0,462	0,400	0,429
молодые учёные	0,774	0,632	**0,696**	0,571	*0,737*	0,644
молодые д.н.	0,314	0,550	0,400	0,300	*0,900*	**0,450**
молодые к.н	0,421	0,640	**0,508**	0,322	*0,760*	0,452
аспиранты	0,750	0,769	**0,759**	0,681	*0,821*	0,744
студенты	0,767	*0,742*	**0,754**	0,545	0,581	0,563

Среднее по категории	0,512	*0,712*	**0,57**	0,473	0,629	0,518

Табл. 4.4 Первая категория. Сущ. & прил.

Полученные значения F-score для рассмотренных выше случаев сведем в результирующие таблицы

		НБК			С 4.5			
класс	Ненорм	все	с+п+г	с+п	ненорм	все	с+п+г	с+п
другое	0,517	0,525	0,492	0,516	0,507	0,6	0,459	**0,553**
доктора наук	0,286	0,24	0,357	0,345	0,441	0,484	**0,509**	0,436
кандидаты наук	0,333	0,4	0,364	0,429	0,431	0,448	**0,455**	0,452
молодые учёные	0,595	0,651	0,635	0,644	0,548	**0,714**	0,667	0,696
молодые д.н.	0,45	**0,452**	0,45	0,45	0,327	0,356	0,353	0,4
молодые к.н	0,395	0,429	0,424	0,452	0,429	0,456	0,5	**0,508**
аспиранты	0,652	0,688	0,66	0,744	0,722	0,725	0,753	**0,759**
студенты	0,606	0,585	0,606	0,563	0,691	0,733	0,746	**0,754**
Взвешенное среднее по категории	0,479	0,496	0,499	0,518	0,51	0,565	0,555	**0,57**

Табл. 4.5 Результирующая таблица по первой категории

Анализ таблицы показывает, что для ряда классов удаётся достичь приемлемых результатов, для отдельных классов результаты получены достаточно плохие. В частности, из классов для которых получены приемлемые результаты выделим следующие: студенты, аспиранты, молодые ученные. Это связано в том числе с тем, что в объявлениях, относящихся к этим классам, зачастую в явном виде указываются и студенты, и аспиранты и молодые ученные (либо указывается, что ученные младше определённого возраста), причем в половине случаев объявления одновременно ориентированы и на аспирантов, и на молодых ученных. Хуже всего обстоят дела с классификацией для классов: кандидаты наук, молодые кандидаты наук, молодые доктора наук. В первую очередь это можно связать с тем, что в объявлениях по этим классам, часто лишь косвенно (без явного указания)

можно определить, что объявление относится к этим классам участников. К тому же набор таких объявлений и типы объявлений более разнообразен, чем типы объявлений для студентов и аспирантов.

Анализируя в целом работу классификаторов отметим следующее:

Практически во всех случаях (кроме молодых докторов наук) алгоритм C4.5 показывает более высокие результаты, для случаев, когда слова были приведены к леммам. Также наиболее хорошие результаты получены в случае представления текстов с помощью лемм существительных и прилагательных документа. Исключения составляют класс молодые ученные, когда предпочтительнее представлять текст с помощью всех лемм документа, а также классы кандидатов и докторов наук, когда помимо существительных и прилагательных полезным является использование глаголов.

Проанализируем работу классификаторов, для случаев, когда в терминах F-score были получены не высокие результаты. В первую очередь это молодые доктора наук и кандидаты. Для обоих классов, для случая, когда получены максимальные значения F-score мы наблюдаем высокую полноту (1,000 для кандидатов наук и 0,950 для молодых докторов наук) и низкую точность. Это говорит о том, что среди полученных объявлений в этих классах будут почти все нужные (то есть мы практически не потеряем объявлений о конкурсах в этих классах), но при этом в эти классы попадет много лишних объявлений. Тем не менее, с точки зрения практики высокое значение полноты нам кажется более важным, чем точности. Последнее сглаживает невысокие результаты, полученные с точки зрения оценки F-score.

Вторая категория

Без нормализации

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,263	*0,652*	**0,375**	0,300	0,130	0,182
проекты & гранты	0,583	0,341	**0,431**	0,500	0,024	0,047
премии, стипендии & выполненные работы	0,815	*0,595*	**0,688**	0,900	0,243	0,383

научная мобильность	0,800	0,414	**0,545**	0,292	*0,966*	0,448
Стартапы & инновационные проекты	0,545	0,800	0,649	0,519	*0,933*	**0,667**
Взвешенное среднее по категории	0,631	*0,517*	**0,533**	0,531	0,379	0,298

Табл. 4.6 Тип конкурса. Без нормализации.

Номера классов	1	2	3	4	5
1	15	4	1	0	3
2	22	14	2	0	3
3	8	3	22	3	1
4	11	2	1	12	3
5	1	1	1	0	12

Табл. 4.7 Тип конкурса. С4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	3	1	0	15	4
2	7	1	0	26	7
3	0	0	9	26	2
4	0	0	1	28	0
5	0	0	0	1	14

Табл. 4.8 Тип конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены все слова

	С4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,395	*0,652*	**0,492**	0,429	0,130	0,200
проекты & гранты	0,792	*0,463*	**0,585**	0,667	0,049	0,091
премии, стипендии & выполненные работы	0,793	*0,622*	**0,697**	0,929	0,351	0,510
научная мобильность	0,759	0,759	**0,759**	0,315	*1,000*	0,479
Стартапы & инновационные проекты	0,440	0,733	0,550	0,448	*0,867*	**0,591**
Взвешенное среднее по категории	0,686	*0,621*	**0,630**	0,603	0,414	0,345

Табл. 4.8 Тип конкурса. Все леммы

Номера классов	1	2	3	4	5
1	15	3	0	0	5
2	9	19	3	4	6
3	8	0	23	3	3
4	4	1	2	22	0
5	2	1	1	0	11

Табл. 4.9 Тип конкурса. С4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	3	1	0	14	5
2	4	2	1	25	9
3	0	0	13	22	2
4	0	0	0	29	0
5	0	0	0	2	13

Табл. 4.10 Тип конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные, прилагательные и глаголы.

	С4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,356	*0,696*	**0,471**	0,429	0,130	0,200
проекты & гранты	0,750	*0,293*	**0,421**	0,667	0,049	0,091
премии, стипендии & выполненные работы	0,741	*0,541*	**0,625**	0,929	0,351	0,510
научная мобильность	0,568	0,724	**0,636**	0,330	*1,000*	0,496
Стартапы & инновационные проекты	0,400	0,533	0,457	0,455	*1,000*	**0,625**
Взвешенное среднее по категории	0,612	*0,531*	**0,528**	0,606	0,428	0,351

Табл. 4.11 Тип конкурса. Сущ. & прил. & глаголы

Номера классов	1	2	3	4	5
1	16	2	0	0	5
2	13	12	3	9	4
3	8	1	20	5	3
4	5	0	3	21	0

5	3	1	1	2	8

Табл. 4.12 Тип конкурса. C4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	3	1	0	13	6
2	4	2	1	24	10
3	0	0	13	22	2
4	0	0	0	29	0
5	0	0	0	0	15

Табл. 4.13 Тип конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные и прилагательные.

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
другое	0,356	*0,696*	**0,471**	0,429	0,130	0,200
проекты & гранты	0,750	*0,293*	**0,421**	0,667	0,049	0,091
премии, стипендии & выполненные работы	0,750	*0,568*	**0,646**	0,938	0,405	0,566
научная мобильность	0,656	0,724	**0,689**	0,354	*1,000*	0,523
Стартапы & инновационные проекты	0,458	0,733	0,564	0,405	*1,000*	**0,577**
Взвешенное среднее по категории	0,639	*0,559*	**0,555**	0,608	0,441	0,366

Табл. 4.14 Тип конкурса. Сущ. & прил.

Номера классов	1	2	3	4	5
1	16	2	0	0	5
2	17	12	3	4	5
3	6	1	21	7	2
4	4	0	3	21	1
5	2	1	1	0	11

Табл. 4.15 Тип конкурса. C4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	3	1	0	13	6
2	4	2	1	21	13
3	0	0	15	19	3
4	0	0	0	29	0
5	0	0	0	0	15

Табл. 4.16 Тип конкурса. НБК. Матрица неточностей

Приведём результирующую таблицу значений F-score.

	НБК				с 4.5			
Класс	ненорм	все	с+п+г	с+п	ненорм	все	с+п+г	с+п
Другое	0,182	0,2	0,2	0,2	0,375	**0,492**	0,471	0,471
проекты & гранты	0,047	0,091	0,091	0,091	0,431	**0,585**	0,421	0,421
премии, стипендии & выполненные работы	0,383	0,51	0,51	0,566	0,688	**0,697**	0,625	0,646
научная мобильность	0,448	0,479	0,496	0,523	0,545	**0,759**	0,636	0,689
Стартапы & инновационные проекты	**0,667**	0,591	0,625	0,577	0,649	0,55	0,457	0,564
Взвешенное среднее по категории	0,298	0,345	0,351	0,366	0,533	**0,63**	0,528	0,555

Табл. 4.17 Результирующая таблица по второй категории

Анализ полученных результатов показывает, что алгоритм C4.5 больше подходит для решения задачи классификации для данной категории, причем лучшие результаты достигаются, если все слова текста были приведены к нормальной форме и не использовался отбор терминов по частям речи. Исключение составляет класс «Стартапы & инновационные проекты» для которого лучшие результаты показал Наивный байесовский классификатор, причем для случая, когда слова в тексте не приводились к нормальной форме и не было отбора терминов по частям речи.

Третья категория

Без нормализации

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
объявление о конкурсе	0,870	0,847	**0,858**	0,703	*1,000*	0,825
объявление результатов	0,435	*0,625*	**0,513**	0,000	0,000	0,000
общая информация	0,444	*0,250*	**0,320**	0,000	0,000	0,000
информация для участников	0,389	*0,467*	**0,424**	0,000	0,000	0,000
Взвешенное среднее по категории	0,737	*0,728*	**0,728**	0,494	0,703	0,580

Табл. 4.18 Тип объявления. Без нормализации

Номера классов	1	2	3	4
1	94	8	4	5
2	6	10	0	0
3	3	3	4	6
4	5	2	1	7

Табл. 4.19 Тип объявления. C4.5. Матрица неточностей

Номера классов	1	2	3	4
1	111	0	0	0
2	16	0	0	0
3	16	0	0	0
4	15	0	0	0

Табл. 4.20 Тип объявления. НБК. Матрица неточностей

Проведена лемматизация, оставлены все слова

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
объявление о конкурсе	0,832	0,847	**0,839**	0,703	*1,000*	0,825

объявление результатов	0,500	*0,500*	**0,500**	0,000	0,000	0,000
общая информация	0,353	*0,375*	**0,364**	0,000	0,000	0,000
информация для участников	0,583	*0,467*	**0,519**	0,000	0,000	0,000
Взвешенное среднее по категории	0,726	*0,728*	**0,726**	0,494	0,703	0,580

Табл. 4.21 Тип объявления. Все леммы

Номера классов	1	2	3	4
1	94	6	7	4
2	8	8	0	0
3	8	1	6	1
4	3	1	4	7

Табл. 4.22 Тип объявления. C4.5. Матрица неточностей

Номера классов	1	2	3	4
1	111	0	0	0
2	16	0	0	0
3	16	0	0	0
4	15	0	0	0

Табл. 4.23 Тип объявления. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные, прилагательные и глаголы.

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
объявление о конкурсе	0,817	0,802	0,809	0,703	*1,000*	**0,825**
объявление результатов	0,429	*0,563*	**0,486**	0	0,000	0,000
общая информация	0,333	*0,250*	**0,286**	0	0,000	0,000
информация для участников	0,375	*0,400*	**0,387**	0	0,000	0,000
Взвешенное среднее по категории	0,489	*0,503*	0,492	0,494	0,703	**0,580**

Табл. 4.24 Тип объявления. Сущ. & прил. & глаголы.

Номера классов	1	2	3	4
1	89	6	8	8
2	7	9	0	0
3	7	3	4	2
4	6	3	0	6

Табл. 4.25 Тип объявления. C4.5. Матрица неточностей

Номера классов	1	2	3	4
1	111	0	0	0
2	16	0	0	0
3	16	0	0	0
4	15	0	0	0

Табл. 4.26 Тип объявления. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные и прилагательные.

	C4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
объявление о конкурсе	0,838	0,793	0,815	0,707	*1,000*	**0,828**
объявление результатов	0,462	*0,750*	**0,571**	0,000	0,000	0,000
общая информация	0,250	*0,063*	**0,100**	0,000	0,000	0,000
информация для участников	0,391	*0,600*	**0,474**	1,000	0,067	0,125
Взвешенное среднее по категории	0,698	0,696	**0,685**	0,592	*0,709*	0,594

Табл. 4.27 Тип объявления. Сущ. & прил.

Номера классов	1	2	3	4
1	88	10	2	11
2	3	12	1	0
3	10	2	1	3
4	4	2	0	9

Табл. 4.28 Тип объявления. C4.5. Матрица неточностей

Номера классов	1	2	3	4
1	111	0	0	0

2	16	0	0	0
3	16	0	0	0
4	14	0	0	1

Табл. 4.29 Тип объявления. НБК. Матрица неточностей

Приведём результирующую таблицу.

	НБК				С4.5			
Класс	ненорм	все	с+п+г	с+п	ненорм	все	с+п+г	с+п
объявление о конкурсе	0,825	0,825	0,825	0,2	**0,858**	0,839	0,809	0,815
объявление результатов	0	0	0	**0,828**	0,513	0,5	0,486	0,571
общая информация	0	0	0	0	0,32	**0,364**	0,286	0,1
информация для участников	0	0	0	0	0,424	**0,519**	0,387	0,474
Взвешенное среднее по категории	0,58	0,58	0,58	0,125	**0,728**	0,726	0,492	0,685

Табл. 4.30 Результирующая таблица по третьей категории

Анализ полученных результатов показывает, что алгоритм С4.5 лучше подходит для решения задачи классификации для данной категории, причем лучшие результаты достигаются, если все слова текста были приведены к нормальной форме и без использования отбора терминов по частям речи. Исключение составляет класс «объявление о конкурсе» для которого более хорошие результаты получены при представлении текста с помощью его слов без нормализации. Также исключение составляет класс «объявление результатов» для которого лучший результат показал наивный байесовский классификатор для случая, когда слова в тексте приводились к нормальной форме и использовались только существительные и прилагательные.

Четвёртая категория

Без нормализации

	С4.5	НБК

Класс	Precision	Recall	F-score	Precision	Recall	F-score
не указано	0,318	*0,350*	**0,333**	0,667	0,200	0,308
международный	0,638	0,732	**0,682**	0,521	*0,927*	0,667
российский	0,594	0,442	0,507	0,667	*0,698*	**0,682**
внутривузовский (СПбГУ)	0,813	*0,867*	**0,839**	1,000	0,533	0,696
городской & региональный	0,824	*0,933*	**0,875**	1,000	0,133	0,235
Взвешенное среднее по категории	0,616	*0,619*	**0,613**	0,697	0,612	0,573

Табл. 4.31 Масштаб конкурса. Без нормализации

Номера классов	1	2	3	4	5
1	7	9	4	0	0
2	3	30	8	0	0
3	11	8	19	2	3
4	1	0	1	13	0
5	0	0	0	1	14

Табл. 4.32 Масштаб конкурса. С4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	4	12	4	0	0
2	0	38	3	0	0
3	2	11	30	0	0
4	0	5	2	8	0
5	0	7	6	0	2

Табл. 4.33 Масштаб конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены все слова

	С4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
не указано	0,333	*0,450*	**0,383**	0,667	0,200	0,308
международный	0,652	0,732	0,690	0,638	*0,902*	**0,747**
российский	0,704	0,442	0,543	0,567	*0,791*	**0,660**

внутривузовский (СПбГУ)	0,778	*0,933*	**0,848**	1,000	0,400	0,571
городской & региональный	0,813	*0,867*	**0,839**	1,000	0,267	0,421
Взвешенное среднее по категории	0,653	0,634	**0,631**	0,700	0,634	0,598

Табл. 4.34 Масштаб конкурса. Все леммы

Номера классов	1	2	3	4	5
1	9	5	5	1	0
2	8	30	3	0	0
3	10	10	19	2	2
4	0	0	0	14	1
5	0	1	0	1	13

Табл. 4.35 Масштаб конкурса. С4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	4	7	9	0	0
2	0	37	4	0	0
3	2	7	34	0	0
4	0	6	3	6	0
5	0	1	10	0	4

Табл. 4.36 Масштаб конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные, прилагательные и глаголы.

	С4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
не указано	0,357	*0,500*	**0,417**	0,667	0,200	0,308
международный	0,714	0,732	0,723	0,607	*0,902*	**0,725**
российский	0,759	0,512	0,611	0,579	*0,767*	**0,660**
внутривузовский (СПбГУ)	0,778	*0,933*	**0,848**	1,000	0,467	0,636
городской & региональный	0,824	*0,933*	**0,875**	1,000	0,200	0,333

Взвешенное среднее по категории	0,695	*0,672*	**0,672**	0,695	0,627	0,588

Табл. 4.37 Масштаб конкурса. Сущ. & прил. & глаголы.

Номера классов	1	2	3	4	5
1	10	6	3	1	0
2	7	30	4	0	0
3	11	6	22	2	2
4	0	0	0	14	1
5	0	0	0	1	14

Табл. 4.38 Масштаб конкурса. С4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	4	8	8	0	0
2	0	37	4	0	0
3	2	8	33	0	0
4	0	6	2	7	0
5	0	2	10	0	3

Табл. 4.39 Масштаб конкурса. НБК. Матрица неточностей

Проведена лемматизация, оставлены существительные и прилагательные.

	С4.5			НБК		
Класс	Precision	Recall	F-score	Precision	Recall	F-score
не указано	0,387	*0,600*	**0,471**	0,667	0,200	0,308
международный	0,714	0,732	0,723	0,621	*0,878*	**0,727**
российский	0,769	0,465	0,580	0,596	*0,791*	**0,680**
внутривузовский (СПбГУ)	0,789	*1,000*	**0,882**	1,000	0,533	0,696
городской & региональный	0,875	*0,933*	**0,903**	1,000	0,333	0,500
Взвешенное среднее по категории	0,709	*0,679*	**0,677**	0,705	0,649	0,620

Табл. 4.40 Масштаб конкурса. Сущ. & прил.

Номера классов	1	2	3	4	5

1	12	5	2	1	0
2	7	30	4	0	0
3	12	7	20	2	2
4	0	0	0	15	0
5	0	0	0	1	14

Табл. 4.41 Масштаб конкурса. C4.5. Матрица неточностей

Номера классов	1	2	3	4	5
1	4	8	8	0	0
2	0	36	5	0	0
3	2	7	34	0	0
4	0	5	2	8	0
5	0	2	8	0	5

Табл. 4.42 Масштаб конкурса. НБК. Матрица неточностей

Приведём результирующую таблицу.

	НБК				C 4.5			
Класс	ненорм	все	с+п+г	с+п	ненорм	все	с+п+г	с+п
не указано	0,308	0,308	0,308	0,308	0,333	0,383	0,417	**0,471**
международный	0,667	**0,747**	0,725	0,727	0,682	0,69	0,723	0,723
российский	**0,682**	0,66	0,66	0,68	0,507	0,543	0,611	0,58
внутривузовский (СПбГУ)	0,696	0,571	0,636	0,696	0,839	0,848	0,848	**0,882**
городской & региональный	0,235	0,421	0,333	0,5	0,875	0,839	0,875	**0,903**
Взвешенное среднее по категории	0,573	0,598	0,588	0,62	0,613	0,631	0,672	**0,677**

Табл. 4.43 Результирующая таблица по четвёртой категории

Анализ полученных результатов показывает, что более удачным является использование алгоритма C4.5. В тех случая, когда с помощью этого алгоритма получены результаты выше чем у Наивного байесовского классификатора они значительно выше, а вот в обратной ситуации результаты C4.5 не сильно уступают лучшим результатам НБК. В случае использования

алгоритма C4.5 оптимальным является представление текста с помощью лемматизированных существительных и прилагательных.

По итогам экспериментов, было выявлено, работа какого классификатора и при каком способе представления документов получены наилучшие результаты для каждой категории в отдельности.

Заключение

В ходе данной работы разрабатывался инструмент для автоматической классификации текстовых документов, содержащих информацию из научной сферы. Решались такие задачи, как: разработка обучающего и тестового множеств, выбор модели представления документа, анализ возможностей выбранного программного пакета Weka, изучение двух алгоритмов машинного обучения – дерева построения решений и Наивного байесовский метода.

Рассмотрены различные подходы, влияющие на качество классификации. По результатам проведённого исследования для каждой категории данных были определены параметры, при которых были получены наилучшие результаты.

Статья в журнале

1. Hartley, R.V.L., Transmission of Information. // Bell Systems Technical Journal, 7 July 1928, pp 535-563
2. Hull, D.A.: Stemming Algorithms - A Case Study for Detailed Evaluation in Journal of the American Society for Information Science 47(1), 1986, pp 70-84,
3. Pantel P., Turney P. Kantrowitz, M: Vector Space Models of Semantics // Journal of Artificial Intelligence Research 37, 2010, pp 141-188

Книга одного автора

4. DeRose, Steven J. Stochastic Methods for Resolution of Grammatical Category Ambiguity in Inflected and Uninflected Languages. 1990. P 566
5. Miyao Y. From Linguistic Theory to Syntactic Analysis: Corpus-oriented Grammar Development and Feature Forest Model. PHD thesis, University of Tokyo. 2006.
6. Porter M.F. An algorithm for suffix stripping / M.F. Porter // Program. - 1980. - Volume 14, № 3. - P. 130-137.
7. Quinlan J. Ross. C4.5 : programs for machine learning. San Mateo, Calif. :Morgan Kaufmann Publishers, c1993. P. 302

Книга нескольких авторов

8. Ceriel, J. Grune, D. Parsing Techniques. A Practical Guide, 2007 P. 662
9. Green G. M., Morgan J. L., Practical guide to Syntactic analysis. 2001. P 14
10. Golub G. van Loan C. Matrix computations. Johns Hopkins University Press; 3rd edition (October 15, 1996) P. 728
11. Michie D., Spiegelhalter D.J., Taylor C.C.. Machine Learning, Neural and Statistical Classification. February 17, 1994. P. 290
12. Rokach L., Maimon O. Data Mining with Decision Trees. 2007. P264

13. Salton G., Wong A., Yang C.S., From Frequency to Meaning for automatic indexing
14. Srivastava A., Sahami M.. Text Mining: Classification, Clustering, and Applications. 2009. P. 328.

Статья в сборнике

15. G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, 1995, pp. 338–345

16. Kantrowitz, M. Stemming and its effects on TFIDF ranking / M. Kantrowitz, B. Mohit, V. Mittal // In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. - 2000. - NY, USA: ACM Press. - P . 357-359.

17. Singal A., Salton G., Mitra M., Buckley C. Document Lenght Normalization. Information Processing and Management. Technical Report TR95-1529, Department of Computer Science, Cornell University, Ithaca, New York, July 1995.

18. Ilya Segalovich, A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine

Ссылка в интернете

19. Машинное обучение (курс лекций, К.В.Воронцов) http://www.machinelearning.ru/
20. Сервис MyStem https://tech.yandex.ru/mystem/