ООО «Преферентум»

Система классификации «Preferentum.Class – интеллектуальный классификатор текстов» (Preferentum.Class)

Руководство по использованию

Листов 40

Москва

2019

Оглавление

Вводная информация о проблематике решения

В июле 2014 года известное агентство International Data Corporation (IDC) выпустило аналитический отчет «The Knowledge Quotient: Unlocking the Hidden Value of Information Using Search & Content Analytics»[1]. В нем компания опубликовала результат исследования, целью которого была попытка оценки влияния имеющихся в компании механизмов поиска и аналитической обработки информации на производительность работы сотрудников. Исследование проводилось на базе 2155 организаций частного и государственного секторов из 6 стран мира.

В отчете отмечено, что около 90% всей информации, которую приходится обрабатывать сотрудникам современных организаций, составляет неструктурированная текстовая информация и объем такой информации непрерывно и существенно возрастает. Из количественных оценок, приведенных в докладе, обращают на себя внимание следующие:

- Примерно 36% процентов рабочего времени офисный служащий тратит на сбор и преобразование данных из доступных ему информационных систем.
- Только 56% потраченного на поиск необходимой информации времени заканчиваются получением необходимого результата.

Отечественные специалисты[2] также подчеркивают важность эффективной обработки текстовых данных: «28% информации, значимой для принятия бизнес решений, находится в слабоструктурированных текстах. Информационное обеспечение, необходимое для обработки данной информации, должно включать в себя как информационный поиск, содержащий мультилингвистическую семантическую задачу, напрямую связанную с релевантностью результатов, – выявления семантических эквивалентов, так и методы <интеллектуальной обработки информации>».

Предлагаемое Вашему вниманию руководство является пособием по системе, реализующей одну из функций интеллектуального анализа текстовых данных – автоматизированное распределение текстовых документов по тематическим (смысловым) категориям.

[1] «Коэффициент знания: Как извлечь скрытую ценность информации используя механизмы поиска и анализа контента».

[2] Хайрова Н.Ф. Информационная технология применения семантически ориентированных методов классификации задачи opinion mining / Н.Ф. Хайрова, Н.В. Шаронова // International Journal Information Technologies & Knowledge. Volume 6/2012, Number 3. с. 273 – 282.

Для понимания специфики документа приводится краткий обзор технологического процесса. Далее приводится описание основных функций системы. Часть возможностей системы предполагает работу с ней хорошо подготовленного пользователя. В частности, использование SDK для интеграции системы в корпоративные системы пользователей требует высококвалифицированного разработчика со стороны пользователя, умеющего применять одну или несколько сетевых интеграционных технологий или привлечение персонала компании ООО «Преферентум» на основе договорных отношений. Определенный опыт требуется для настройки и машинного обучения системы классификации. Он приходит к пользователям значительно быстрее при взаимодействии с консультантами разработчика.

Определения и сокращения

Биграмма – в контексте данного документа, пара последовательных слов.

Именная группа – словосочетание, в котором имя существительное является вершиной, то есть главным словом, определяющим характеристику всей составляющей.

Индекс – тематический словарь системы, необходимый для проведения классификации в отдельно взятой области. Физически представляет собой набор файлов, создаваемых системой в отдельной заданной пользователем папке.

ИТ – информационные технологии.

Классификация (текста) – определение тематики (рубрики) отдельно взятого текста или документа.

Кластер – объединение, массив нескольких однородных документов, отличающийся какими-либо свойствами от других подобных массивов.

Неструктурированная (текстовая) информация – представленная в виде текста свободной формы. Физически это могут быть текстовые файлы, электронные документы, файлы электронной почты, информация с web-сайтов и так далее.

Проприетарный – являющийся частной собственностью авторов или правообладателей, закрытый.

Ранг (какой-либо рубрики) – величина от 0 до 1, показывающая вероятность отнесения текстовой информации к тематике данной рубрики. Например, если ранг рубрики равен 0.84, вероятность того, что тематика текста соответствует тематике рубрики, равна 84%.

Рубрика – то же самое, что тематика, – информационная направленность текста или документа.

Семантический – связанный со значением, смыслом чего-либо. Также может использоваться в значении «словарный».

Стоп-слова – слова (или отдельные символы), которые не учитываются при анализе текста, так как не являются информативными с точки зрения анализа данных. Примерами стоп-слов являются знаки препинания, числа, союзы, междометия, причастия, предлоги, местоимения и некоторые вводные слова.

СЭД – система электронного документооборота.

Терм – интуитивно понятное выражение (в русском языке обычно имя существительное или прилагательное), являющееся формальным именем объекта или именем формы.

1. Обзор технологии

Программное обеспечение Preferentum.Class – интеллектуальный классификатор текстов (далее Preferentum.Class) решает проблему распределения неструктурированной текстовой информации по заранее заданным тематическим рубрикам (классам). В профессиональной терминологии данная задача называется «классификация текстов с предварительным обучением».

Каждая тематическая рубрика представляет собой определенную область человеческих знаний. Например, в состав демонстрационного стенда входят рубрики «Театр», «Спорт», «Медицина», «Космос». Количество рубрик, с которыми работает система, теоретически может быть бесконечно большим. На практике их объем ограничен логикой и ограничениями решаемой задачи, требованиями к качеству классификации и производительности системы.

В системе реализованы следующие функции:

- Выявление тематических кластеров в массивах документов, имеющих отношение к заданной предметной области.
- Отнесение документа к одному или нескольким разделам тематических классификаторов.
- Автоматическая настройка каждого из элементов классификатора на наиболее значимые смысловые концепты.
- Поиск документов, похожих на заданный.

В реальной жизни задача классификации текстов возникает в таких областях, как:

- Сортировка обращений в службу ServiceDesk, когда необходимо определить тематику обращения и переслать его профильному департаменту/специалисту;
- Аналогично – при обработке письменных обращений граждан на государственные и муниципальные интернет-порталы;
- Сортировка документов в информационных хранилищах.

Для классификатора информация о рубриках и вся статистика хранится в так называемом индексе (словаре), представляющем собой множество файлов проприетарного формата в некоторой директории локального компьютера. Для того чтобы была возможна классификация, необходимо сначала «обучить» классификатор, подав ему на вход некоторое количество текстов с указанием рубрики для каждого текста. Система вносит соответствующую информацию в индекс классификатора, при этом сами обучающие тексты не сохраняются. В дальнейшем можно дообучать классификатор. ПО Preferentum.Class может работать с любым количеством индексов одновременно.

После обучения система способна оценивать переданный ей текст на соответствие внесенным в индекс рубрикам. Оценка производится путем присвоения каждой из рубрик ранга – величины от 0 до 1[3]. Чем ближе ранг рубрики к 1, тем с большей вероятностью тематика текста относится к этой рубрике. И наоборот, низкий ранг рубрики говорит о том, что тематика текста не соответствует данной рубрике. Текст может соответствовать как одной, так и нескольким рубрикам, внесенным в индекс. Во втором случае ранги рубрик будут примерно равны.

Качество классификации текста зависит от многих факторов, основными из которых является качество предварительного обучения, а также объем и семантическая однородность классифицируемого текста. Практические исследования показали, что качество классификации квалифицированного оператора-человека составляет 95 – 97% (от 3 до 5 ошибок на 100 документов). Поэтому на практике необходимо стремиться к качеству классификации системой хотя бы по одной из рубрик не ниже 95%, в этом случае результат классификации может считаться приемлемым для дальнейшего использования. В противном случае необходимо устранить причины неудовлетворительной классификации: провести дополнительное обучение системы, изменить настройки, при возможности, изменить состав рубрик.

Настройки системы позволяют регулировать чувствительность алгоритма классификации к внесенным в индекс тематикам. Регулировкой настроек можно повысить качество классификации по отдельным рубрикам, при этом часть обрабатываемых файлов будет отнесена системой в область неуверенной классификации. Файлы, исключенные из автоматической классификации, подлежат ручной обработке человеком-оператором. Объем отнесенных на ручной разбор текстов позволяет грубо оценить экономию трудозатрат на операции классификации. Например, если 91.2% документов классифицируются с качеством выше 99.2%, а 8.8% относятся на ручной разбор (пример взят из реального случая), это означает, что автоматизированная система способна заменить более 91% объема ручной работы по разнесению поступающей информации[4].

[3] Очевидно, что в процентном отношении этот диапазон выражается величинами от 0 до 100%.

[4] В данной оценке не учтены эргономические факторы, повышающие результативность системы, такие как возможность круглосуточной работы автоматизированной системы и отсутствие фактора усталости. С другой стороны, при расчете совокупной оценки экономической эффективности необходимо учитывать стоимость самой системы, затраты на ее внедрение и сопровождение.

2. Описание работы

2.1.Дистрибутив системы

Дистрибутив системы Preferentum.Class представляет собой архив с именем виде Class.Sdk.N.M.rar, где N,M –целые числа, обозначающие номер версии и номер релиза системы. В дистрибутив входят следующие папки:

1 Папка «Samples» содержит примеры текстов, которые распределены по четырем категориям: «Космос», «Медицина», «Спорт», «Театр». Это - демонстрационная обучающая выборка, которая используется для создания простейшего классификатора. Её можно сохранить на диск в файловой структуре в любом удобном месте. Выборка очень маленькая, там всего 8-10 коротких примеров в каждой категории, поэтому классификатор на ее основе получится не очень точный.

2 Папка «TestDesk» содержит приложение «Class.TestDesk.exe», которое является непосредственно демонстрационным стендом системы. Для данного приложения необходимо вручную создать ярлык запуска на рабочем столе.

3 Папки «Server» и «WebServer» содержат серверные варианты реализации стенда классификации.

Дистрибутив ПО Preferentum.Class обновляется на сайте компании ООО «Преферентум» (http://preferentum.ru) по мере выхода свежих релизов. Разработчиками учитываются замечания и предложения пользователей системы, и они реализуются в свежих релизах, с обеспечением преемственности. Поддержка старых версий, без наличия договорных отношений, не предполагается.

2.2. Сценарии использования

ПО Preferentum.Class предусматривает использование в решениях, работающих как на Windows, так и на Linux системах. Решение может встраивается в систему через собственный API или в виде готовой библиотеки «dll».

Описание API системы классификации Preferentum.Class представлено в соответствующем разделе настоящего руководства.

Лицензия для ознакомления с функционалом системы не требуется. Поддержка сторонних разработчиков приложений с использованием Preferentum.Class оказывается на договорной основе. Коммерческое использование системы предполагает лицензирование, зависящее от объема

предоставляемых возможностей. Замечания и предложения по продукту принимаются на preferentum@it.ru.

2.3. Демонстрационный стенд системы – TestDesk

Для ознакомления с основными принципами и возможностями работы ПО Preferentum.Class предлагается установить демонстрационный стенд системы – TestDesk.

TestDesk является рабочим полнофункциональным вариантом работы ПО Preferentum.Class. Данное руководство предполагает использовать его в качестве демонстрационного стенда в силу простоты установки и настройки.

2.3.1. Общие возможности

TestDesk может использоваться в целях тестирования алгоритма классификации, для проведения практических исследовательских работ, а также для оценки работоспособности системы с точки зрения замещения функций человека-оператора.

Система реализована как программный модуль на платформе Windows. Стенд поддерживает различные варианты представления данных в обучающей и тестовой выборках: в виде CSV-файла либо в виде структуры файловых папок, содержащих файлы с текстовой информацией (форматы TXT, DOC, DOCX, HTML, PDF, ODT и др.).

Поддерживается возможность экспорта результатов автоматической классификации в виде CSV-файла с целью последующего анализа качества обучающей выборки.

Реализована возможность автоматической «чистки» обучающей выборки, что позволяет устранить из процесса обучения классификатора человеческие ошибки и повысить качество классификации.

Поддерживается визуальное отображение графа типовых ошибок классификации для выявления проблемных рубрик и их последующего дополнительного обучения.

Возможна тонкая настройка алгоритмов классификации для настройки порога неуверенной классификации и передачи неуверенно классифицируемых текстов на ручную обработку оператору, настройки параметров индексирования текстов обучающей выборки.

Поддерживается раздельная настройка порогов неуверенной классификации по каждой отдельной рубрике, что позволяет гибко учитывать неравномерность распределения текстов в обучающей выборке по категориям.

Реализована возможность расширения используемого лексического словаря администратором для учета специфики деятельности организации: отраслевой терминологии, аббревиатур, сленговых выражений и т.п.

Поддерживается возможность ручной корректировки весовых коэффициентов в обученном индексе с защитой от их изменения при последующем дополнительном обучении классификатора. Это позволяет администратору системы при необходимости вручную добавлять правила классификации «поверх» автоматически обученного индекса.

2.3.2. Минимальные технические требования к системе

2.3.2.1. Десктоп-версия

Реализована в целях разработки, тестирования и настройки параметров индексов. TestDesk.exe, представляющий собой толстый клиент под Windows. Данный модуль алгоритмически аналогичен серверам, но сам не имеет API для внешнего использования.

- Компьютер: Процессор x86 или x64 с тактовой частотой от 2 ГГц;
- Операционная система: Microsoft® Windows® 10 / 8.1 / 8 / 7;
- Объем оперативной памяти: не менее 2 ГБ (рекомендовано – от 4 ГБ);
- Свободное место на диске: от 150 МБ;
- Видеокарта и монитор с разрешением не менее 1024x768 точек;
- Клавиатура, мышь или другое указательное устройство;
- Опционально – пакет офисных приложений Microsoft Office версии 2003 и выше.

2.3.2.2. Версия клиент-сервер

- Компьютер: Процессор i7/4 ядра;
- Операционная система: WinServ2012 R2 или выше;
- Объем оперативной памяти: не менее 4 ГБ;
- Свободное место на диске: 30 ГБ;

2.3.3. Установка системы

Установка системы на компьютере пользователей требует разархивирования дистрибутива в произвольную папку. Лицензия для ознакомления с функционалом системы не требуется.

2.3.4. Старт приложения. Интерфейс системы

Для старта приложения необходимо запустить файл «Class.TestDesk.exe», расположенный в папке «TestDesk». Если приложение запускается не в первый раз, система запоминает последний индекс, с которым производилась работа и при запуске системы предлагает открыть его см. Рис. 1.

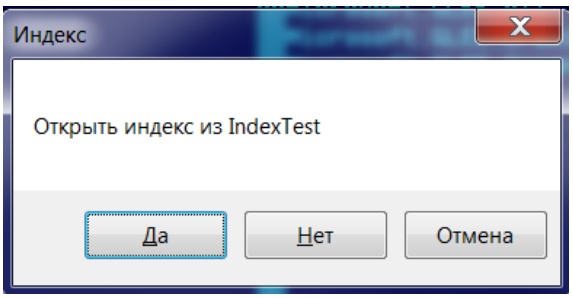


Рис. 1- Запрос на открытие индекса

Запрос можно игнорировать (кнопка «Нет»), если предполагается работать с новым или отличным от предложенного системой индексом.

На Рис. 2 представлены основные элементы интерфейса системы. Меню системы имеет 3-уровневую структуру:

- Меню 1-го уровня позволяет создать новый тематический словарь (кнопка «Создать индекс») либо выбрать ранее созданный («Открыть существующий индекс»).
- Меню 2-го уровня реализует основные функции системы: обучение с возможностью последующей классификации текста. Подробнее об этих функциях будет рассказано ниже.
- Состав меню 3-го уровня зависит от выбранной в меню 2-го уровня функции. Таким образом, меню 3-го уровня являются контекстно-зависимыми.
- Зеленым контуром выделена кнопка основных настроек системы.
- Синим контуром на Рисунке 2 обведены окна ввода-вывода информации. Состав и расположение окон контекстно привязаны к выполняемой системой процедуре.
- Помимо основных настроек, в системе возможны тонкие настройки выполняемых процедур. Тонкие настройки вынесены в окна ввода-вывода информации.

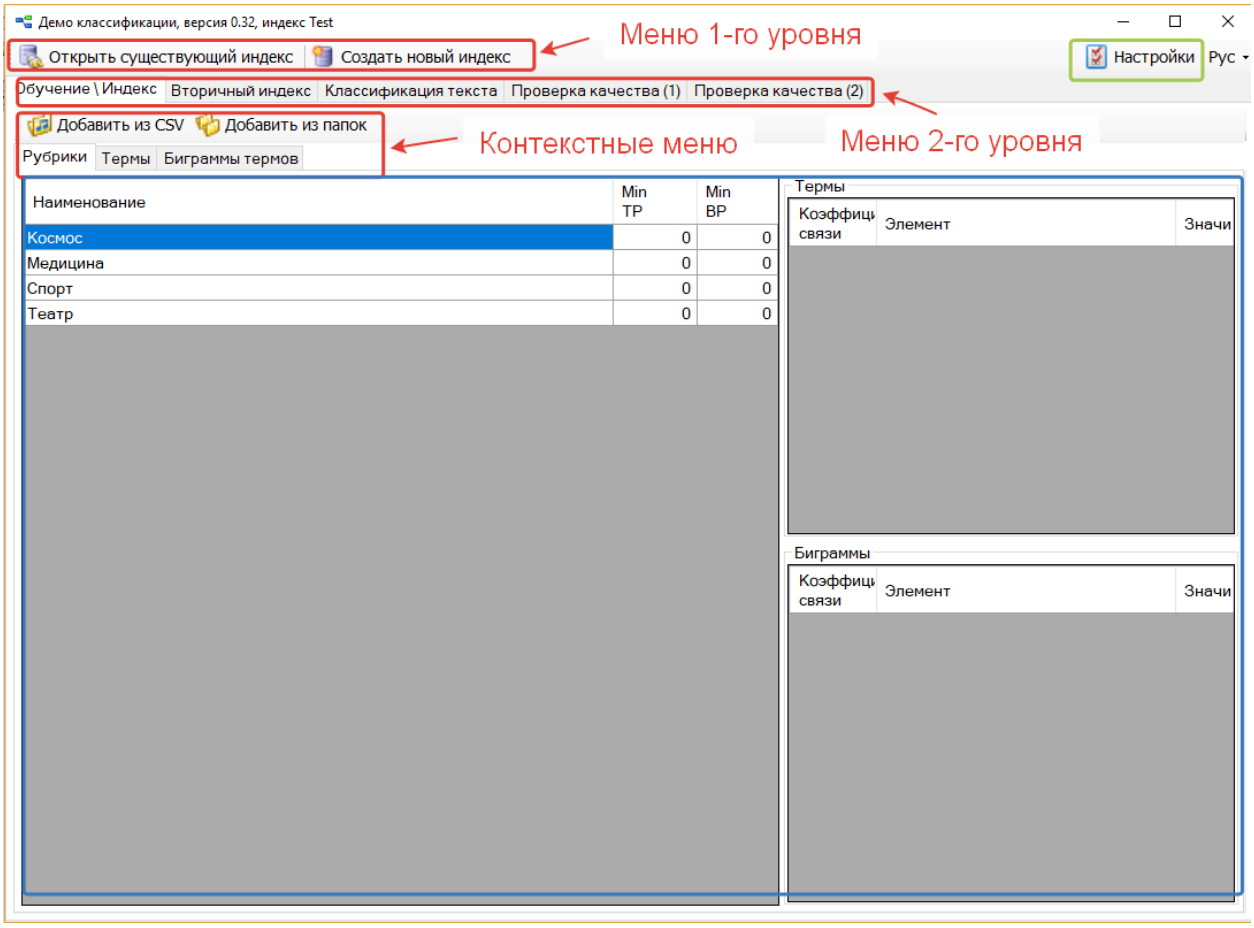


Рис. 2- Интерфейс системы

При запуске системы в первый раз необходимо выбрать «Создать новый индекс», указав чистую папку. Для работы с ранее созданными индексами необходимо выбрать «Открыть существующий индекс».

Меню 2-го уровня имеет 4 основные закладки: «Обучение\Индекс», «Вторичный индекс», «Классификация текста», «Проверка качества (1)», «Проверка качества (2)». На закладке «Обучение\Индекс» производится формирование индекса (обучение), после проведения обучение можно система показывает структурный состав и элементы внутри индекса. На закладке «Классификация текста» можно классифицировать один текст. Закладка «Проверка качества (1)» служит для оценки качества обучения текущего индекса (подробнее процедура оценки рассмотрена ниже). Закладка «Проверка качества (2)» функционально полностью соответствует закладке «Проверка качества (1)». Она сделана для удобства, чтобы можно было сравнивать результаты на разных прогонах выборки. Качество обучения индекса – ключевой параметр для проведения качественной классификации.

Закладка «Вторичный индекс» служит для построения сложных обучающих индексов и может использоваться опытными пользователями системы, хорошо разбирающимися в

технологии машинной классификации. В рамках данного руководства мы не будем подробно останавливаться на данной функции.

2.3.5. Формирование тематического классификатора

Перед проведением обучения необходимо подготовить выборку расклассифицированных человеком текстов в одном из указанных ниже форматов и разбить ее на обучающую часть и проверочную (тестовую) часть. Документы в тестовой выборке уже отнесены человеком к определенной категории, т.е. для них известны «правильные ответы». После автоматической классификации тестовой выборки можно сравнить результат, данный человеком, с результатом работы программы и оценить точность классификации. Тестовая часть по объему обычно меньше или равна обучающей. Обычное соотношение обучающей и тестовой части в процентном отношении – 80/20, 70/30, 50/50. Тестовая часть в дальнейшем потребуется для отладки индекса и повышения качества классификации. Пропорциональное отношение между обучающей и тестовой частями зависит от конкретной задачи и выбирается с целью достижения максимального качества обучения[5].

Обучение классификатора возможно 2-мя способами.

1. Наиболее простым вариантом является загрузка текстового словаря из папок. Для этого необходимо нажать кнопку «Добавить из папок» и указать базовую папку, подпапками которой являются тематические рубрики, а уже внутри подпапок находятся обучающие тексты в файлах произвольных форматов (txt, doc, docx, rtf, html, pdf и др.). Вариант с загрузкой информации из папок может быть удобнее и в случае, если классифицируются большие по объёму документы.

В состав тестового стенда входит демонстрационная папка «Samples» с набором файлов по дисциплинам «Космос», «Медицина», «Спорт», «Театр». Последовательность действий при обучении системы на примере демо-стенда см. Рис. 3.

[5] На практике лучше разбить массив данных на четыре-пять частей. При настройке классификатора одну из частей можно будет выбрать в качестве тестовой выборки, а остальные использовать для обучения. Начать обучение можно будет с одной части, а затем, последовательно увеличивая объем обучающей выборки, можно будет проследить зависимость качества классификации от объема доступной для обучения информации.

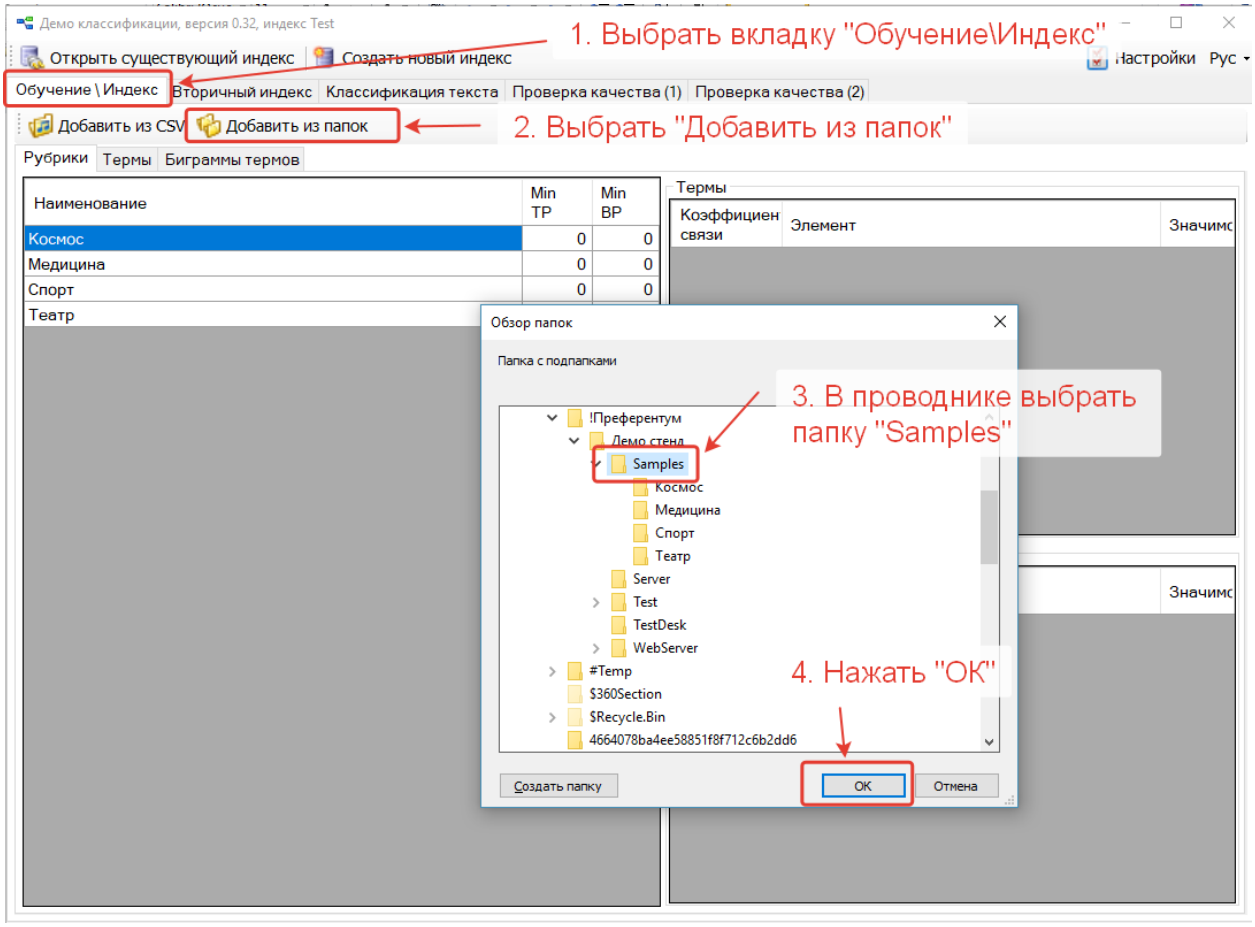


Рис. 3 - Выбор папки для обучения индекса

На заданный системой вопрос «Загрузить … файлов из … папок-рубрик?» необходимо ответить «Да». Индексирование займет около 5-10 секунд. После обучения на вкладке «Рубрики» Вы увидите четыре категории – «Космос», «Медицина», «Спорт», «Театр».

Примечание. Для проверки качества сформированного индекса необходимо заранее подготовить (как было указано выше) или сформировать на данном этапе папку с проверочными файлами (назвать ее, например, «Samples (проверка качества)») с таким же набором подпапок и файлов внутри, как в папке с обучающим набором файлов. Набор файлов внутри проверочных папок должен соответствовать тематике папок, но отличаться по составу от обучающего набора. Подробнее о процедуре проверки качества см. раздел «Проверка качества индекса».

2. Вторым вариантом является загрузка заранее подготовленного текстового файла (например, формата CSV), в котором в каждой строке находится тематическая рубрика, затем после точки с запятой идёт в одну строку текст классификатора, соответствующий теме рубрики. Такой формат может быть также удобен в случае, если классифицируемые документы представляют собой относительно короткие тексты, например, обращения на горячую линию. Тогда каждое отдельное обращение преобразуется в отдельную строку файла.

Кодировка файла: UTF-8 или Windows-1251. Если в тексте обращения есть символы конца абзаца, возврата каретки и перехода на новую строку – их надо заменить пробелами (разумеется, кроме последнего в строке). Символы «/» и «\» также нужно заменить на пробелы или символ подчеркивания.

Пример (жирным шрифтом выделена тематика рубрики (на практике необязательно), тексты классификатора обрезаны (на практике должны быть полными), знак «;» обязателен в качестве разделителя):

Здравоохранение; Благодаря безразличию и непрофессионализму …

Жилищно-коммунальное хозяйство; Здравствуйте, в ноябре месяце было 5 раз отключение…

Реклама; Адрес нарушения: улица Шаболовка д.69/32. Сообщение: Несколько месяцев назад…

Записей для одной рубрики может быть сколько угодно, рубрики могут идти вперемежку. Загрузка в один индекс может происходить множество раз – данные будут накапливаться. После обучения возможна классификация.

После обучения интерфейс системы будет выглядеть как на Рис. 4. При переходе между категориями в окне «Рубрики» (слева) в окнах «Термы» и «Биграммы» (справа) будут отображаться значимые термы и биграммы, связанные с данной категорией (списки отсортированы по убыванию силы связи). Классификатор обучен и готов к работе.

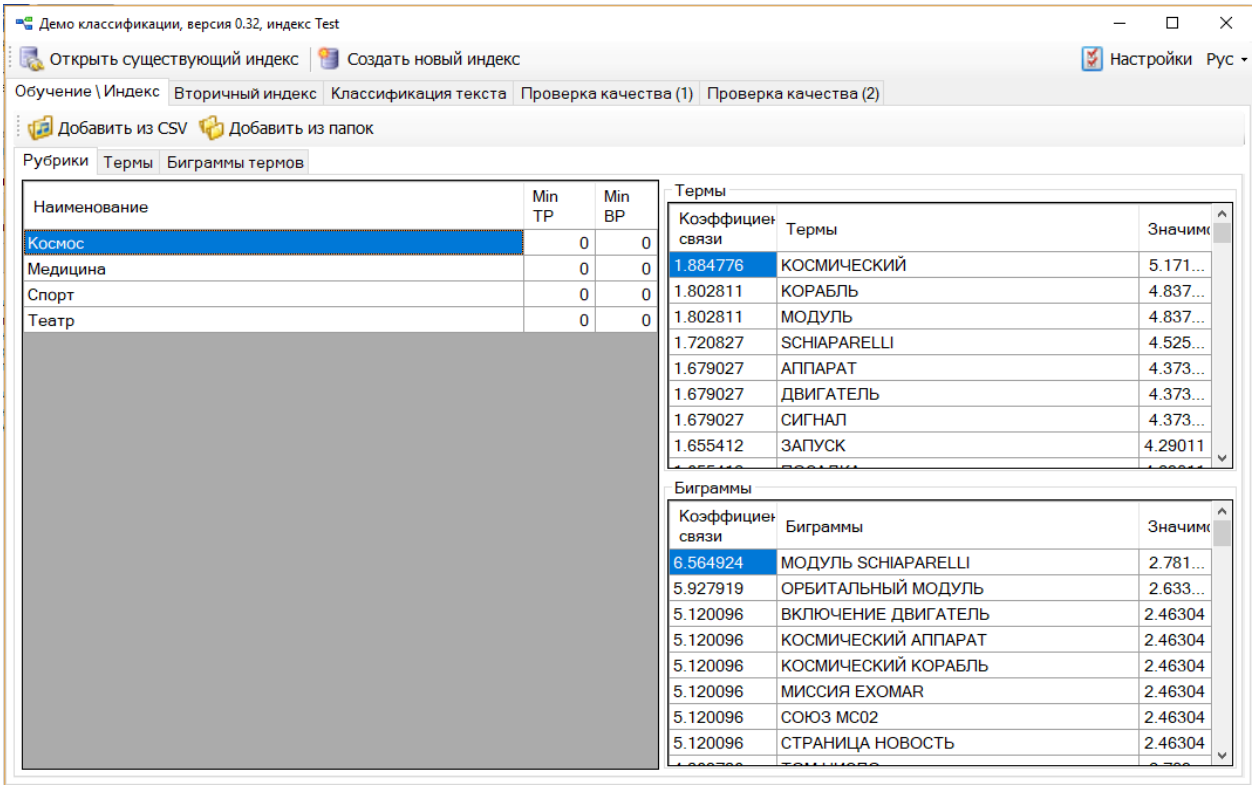


Рис. 4 - Интерфейс системы после обучения

2.3.6. Проверка качества индекса

Проверочная часть должна быть подготовлена в одном из указанных выше форматов. Для загрузки проверочной части необходимо открыть вкладку «Проверка качества (1)» и нажать, в зависимости от формата проверочной части, одну из кнопок «Классифицировать из CSV» или «Классифицировать из папок» см.Рис. 5. Выводится общая статистика, граф типовых ошибок классификации и статистика по отдельным рубрикам. Для анализа результата и последующей отладки обучения служат кнопки «Сохранить», «Расщепить на файлы», «Расщепить на CSV».

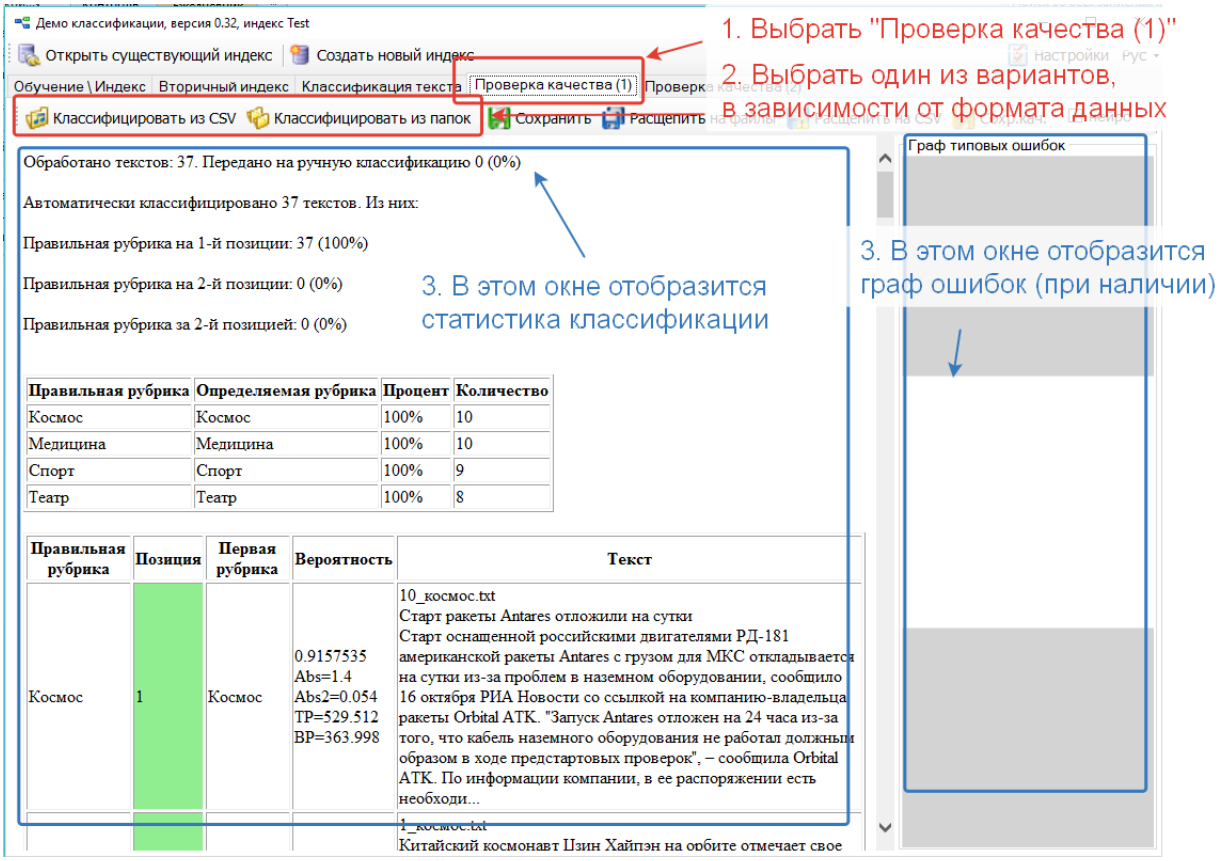
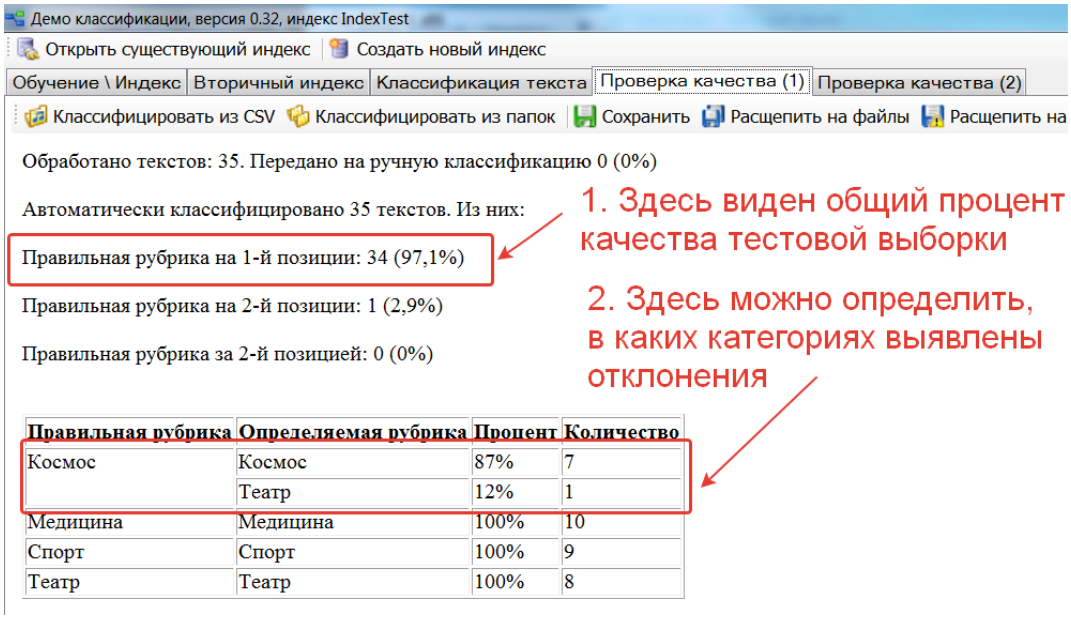


Рис. 5 - Проверка качества индекса

Интерфейс системы предоставляет следующий варианты работы с данными:

- Кнопка «Сохранить» позволяет сохранить результаты проверки в CSV-файл, далее открыть его в Excel и провести анализ данных файла средствами Excel.
- Кнопки «Расщепить на файлы» («Расщепить на CSV») позволяют выделить из проверяемых данных файлы (либо строки CSV), неправильно отнесенные к рубрикам. На Рисунке 6 по результатам проверки качества видно, что 97.1% проверочных файлов соответствуют заданным тематикам, а 2.9% некорректно отнесены к своей тематике. Ниже выводится детализация классификации, по которой можно определить категории с некорректно отнесенными файлами. В приведенном на Рис. 6 примере один из восьми файлов, находящихся в папке «Космос», по информационной наполненности более относится к категории «Театр».

Необходимо отметить, что операция расщепления, примененная к обучающей выборке, позволяет удалить из нее ошибочную информацию и таким образом повысить качество обучающей выборки.



Правильная рубрика	Определяемая рубрика	Процент	Количество
Космос	Космос	87%	7
	Театр	12%	1
Медицина	Медицина	100%	10
Спорт	Спорт	100%	9
Театр	Театр	100%	8

Рис. 6 - Пример классификации текста

2.3.7. Процедура классификации

Классификация текста производится на одноименной закладке «Классификация текста». Текст можно загрузить из файла произвольного формата (txt, doc, docx, rtf, html, pdf и др.) либо скопировать в окно «Текст для классификации».

После загрузки текста нажать кнопку «Классифицировать». В правом окне отображается результат классификации – перечень известных классификатору рубрик и коэффициент, показывающий ранг рубрики (степень соответствия классифицированного текста тематике каждой из известных рубрик). На первом месте – рубрика с наибольшим весом. В правом углу интерфейса есть флажок «детализация» – если на него нажать, будет отображаться более подробная информация о том, какой вклад дают термы и биграммы в формирование результата. Можно попробовать найти в интернете примеры текстов про космос, театр, культуру, медицину и посмотреть, насколько точно они классифицируются.

Приведенный на Рис. 7 пример текста наиболее соответствует медицинской тематике:

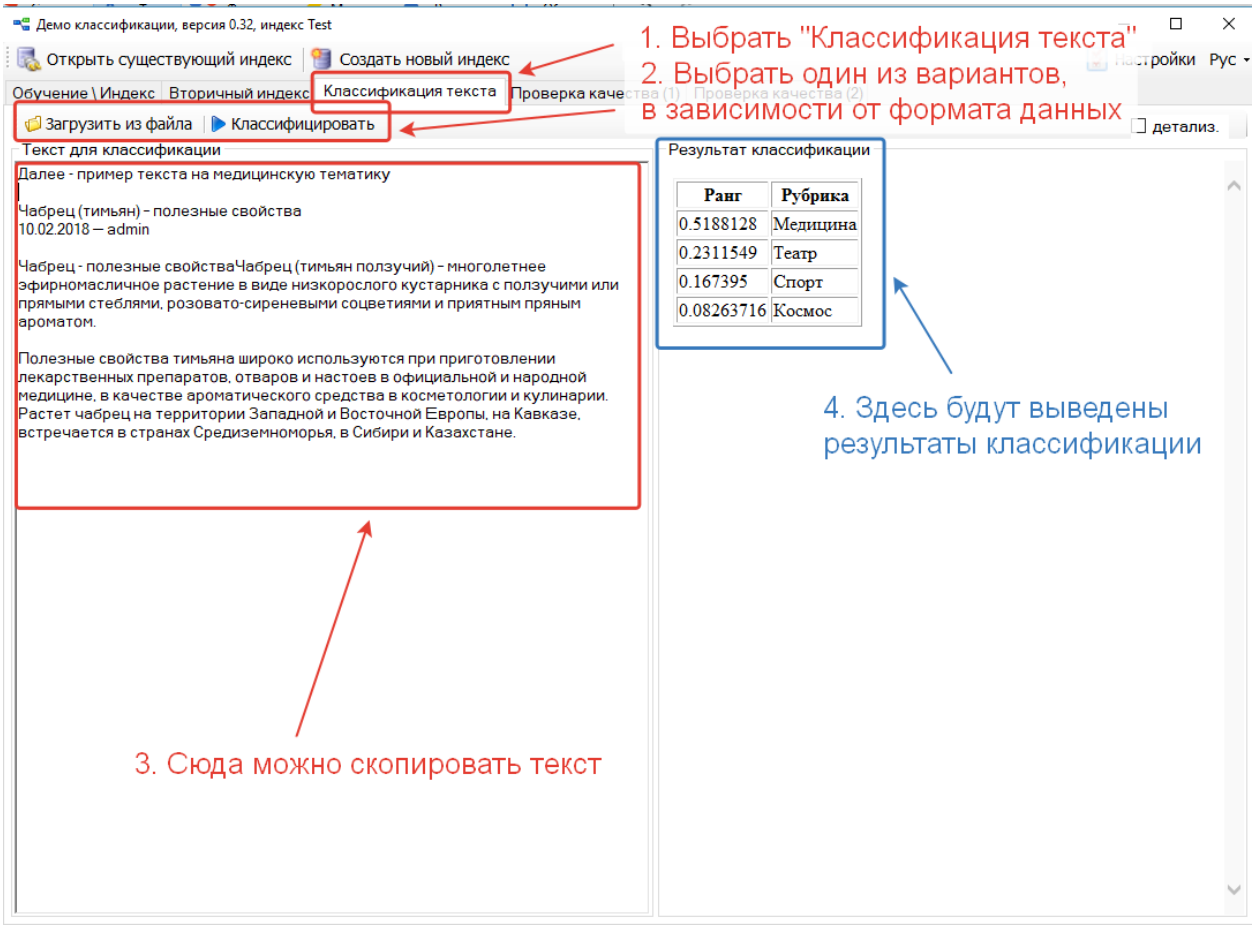


Рис. 7 - Оценка соответствия текста

В дальнейшем можно продолжать обучать классификатор, добавляя новые рубрики или пополняя примеры текстов в уже имеющихся рубриках, можно проверять качество классификации, проверяя тестовую выборку, можно проводить тонкую настройку параметров и алгоритмов классификации и так далее.

2.3.8. Настройки системы

Обращаем Ваше внимание, что работа с настройками системы предполагает углубленное понимание технологии процесса классификации. Изменение настроек не вызовет необратимых последствий, но может резко изменить результативность работы системы, причем в большинстве случаев связь между изменением настроек и изменением качества классификации не будет очевидна. Поэтому скорее всего, у Вас не получится просто «поиграть» настройками для получения наилучшего результата классификации. С вопросами тонкой настройки системы Вы можете обратиться по реквизитам, указанным в разделе 2.1 или на сайте ООО “Преферентум” (http://preferentum.ru).

Настройки системы вызываются кнопкой «Настройки» в верхнем правом углу главного экрана системы см. Рис. 8.

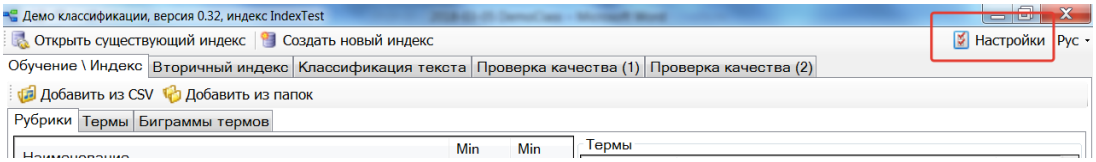


Рис. 8 - Вызов настроек системы

Меню настроек представлено на Рис. 9.

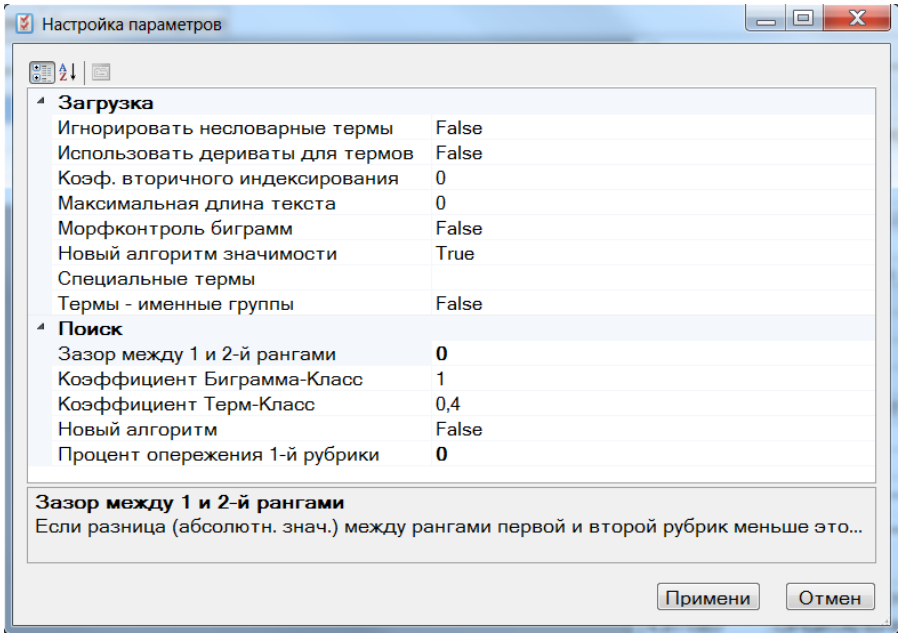


Рис. 9 - Меню настроек системы

- Меню «Загрузка»:
  - «Игнорировать несловарные термы»: По умолчанию параметр установлен в «false». Если параметр установлен в «true», при анализе текста система будет игнорировать все определения, не входящие в словарь русского языка: специализированные термины, жаргонизмы, сокращения и т.д.
  - «Игнорировать собственные имена»: По умолчанию параметр установлен в «false». Если параметр установлен в «true», система будет игнорировать имена, отчества, географические наименования и т.д.
  - «Использовать дериваты для термов»: По умолчанию параметр установлен в «false». Если параметр установлен в «true», однокоренные слова и словоформы система будет приводить к единому корню. В некоторых случаях данный параметр позволяет поднять качество классификации.

- «Коэф. вторичного индексирования»: функционал, за который отвечает данный параметр, находится в стадии модификации.
- «Максимальная длина текста»: позволяет ограничить размер анализируемого текста[6].
- «Морфоконтроль биграмм»: По умолчанию параметр установлен в «false». Если параметр установлен в «true», при анализе биграмм система будет учитывать их морфологическое согласование, связность слов (правильность падежей, согласование лиц, множественного и единственного числа и т.д.). Несогласованные биграммы анализироваться не будут.
- «Новый алгоритм значимости»: По умолчанию параметр установлен в «true». Увеличивает количество термов в индексе.
- «Специальные термы»: позволяет ввести строку обозначений, сокращений, терминов, значимых с точки зрения анализируемого текста. Данный параметр может быть полезен при анализе специализированных или профессиональных текстов. Список термов вводится через пробел.
- «Термы – именные группы»: По умолчанию параметр установлен в «false». Если установить в «true», в качестве термов система будет использовать также именные группы.

- Меню «Поиск»:
  - «Зазор между 1 и 2-й рангами»: Устанавливается в процентах, по умолчанию равен нулю. Данный параметр позволяет установить разницу между первыми двумя рубриками, при которой система будет считать классификацию уверенной. Установка порога уверенной классификации необходима для отсечения из массива классифицируемой информации текстов, которые система не может достаточно уверенно отнести к единственной рубрике.

Рассмотрим пример, в котором данный параметр установили равным 30%. Тогда, если при классификации система отнесет текст к двум рубрикам с вероятностью[7] 0.6 и 0.2 (разница между вероятностями 0.4 или 40%), классификация будет считаться уверенной. А

[6] Параметр имеет смысл устанавливать в случае, когда системе приходится анализировать длинные тексты, в которых информационно значимой является только первая часть. Например, в текстах электронной переписки классификацию можно проводить только по последнему в цепочке письму. Предыдущую переписку можно опустить, так как там могут обсуждаться смежные вопросы, которые неизбежно ухудшат качество классификации. В этом случае, параметр «Максимальная длина текста» необходимо установить примерно равным средней длине письма в файлах переписки.

[7] Напомним, что вероятность отнесения текста к рубрике в терминах системы называется «ранг рубрики».

при отнесении текста к рубрикам с рангами 0.6 и 0.4 классификация текста будет считаться неуверенной.

Понятие уверенной классификации также рассматривается в разделе «Как понять, что классификация уверенная?».

- «Коэффициент «Биграмма-Класс»: Позволяет регулировать значимость словосочетаний при классификации текста. При увеличении данного параметра относительно параметра «Терм-Класс» словосочетания будут более значимы при проведении классификации, чем отдельные слова.
- «Коэффициент «Терм-Класс»: При проведении классификации позволяет регулировать значимость выделяемых из текста термов. При увеличении данного параметра относительно параметра «Биграмма-Класс» слова («термы») будут более значимы при проведении классификации, чем словосочетания.
- «Новый алгоритм значимости»: увеличивает количество термов в индексе.
- «Процент опережения 1-й рубрики»: параметр аналогичен «Зазору между 1 и 2-й рангами», но устанавливается в относительной величине между рубриками.

Например, если данный параметр установить соответствующим 80%, тогда при отнесении текста к рубрикам с рангами 0.8 и 0.7 классификация будет считаться неуверенной, а в случае с рангами 0.8 и 0.6 – уверенной.

2.4.Как улучшить качество классификации

2.4.1. Как измерить точность классификации?

Существует несколько подходов к оценке точности. Прежде всего, нужно определиться с решаемой задачей. Варианты решаемых задач:

- Классификатор должен выдать единственный результат: категорию, к которой относится обрабатываемый документ.
- Необходимо определить множественный набор категорий, к тематике которых имеет отношение проверяемый документ. В этом случае классификатор выдает ранжированный список требуемых категорий.

2.4.2. Как понять, что классификация уверенная?

Такой вопрос возникает, когда классификатор должен выдавать только одну правильную категорию. Нужно иметь какой-то критерий, показывающий, насколько можно доверять выбранной категории. Результат работы классификатора – это список категорий, упорядоченный

по убыванию вероятности отнесения классифицируемого объекта к данной категории. Поэтому в качестве критерия уверенности лучше всего использовать соотношение между вероятностью первой и вероятностями последующих категорий. Первая, вторая и дальнейшие категории в данном контексте означают номера категорий в списке, отсортированном по убыванию вероятности. Т.е. первая – это категория с максимальной вероятностью, вторая – следующая за первой и т.п. Если отношение вероятности второй категории к вероятности первой превышает некоторое пороговое значение, такая классификация будет считаться неуверенной и передаваться на рассмотрение оператору. Введение порога уверенной классификации позволяет осуществить разделение всего классифицируемого потока на две части: полностью автоматически классифицируемый поток (уверенная классификация) и поток, предварительно обрабатываемый автоматически, но верифицируемый в дальнейшем оператором (неуверенная классификация). Отметим, что «неуверенная» не означает «неправильная» – в большинстве случаев выбранная оператором категория будет совпадать с тем, что предложила система.

На Рис. 10 приведен пример уверенной классификации: вероятность отнесения объекта к первой категории сама по себе весьма высокая (более 60%) и в несколько раз превышает вероятность второй и последующих категорий.

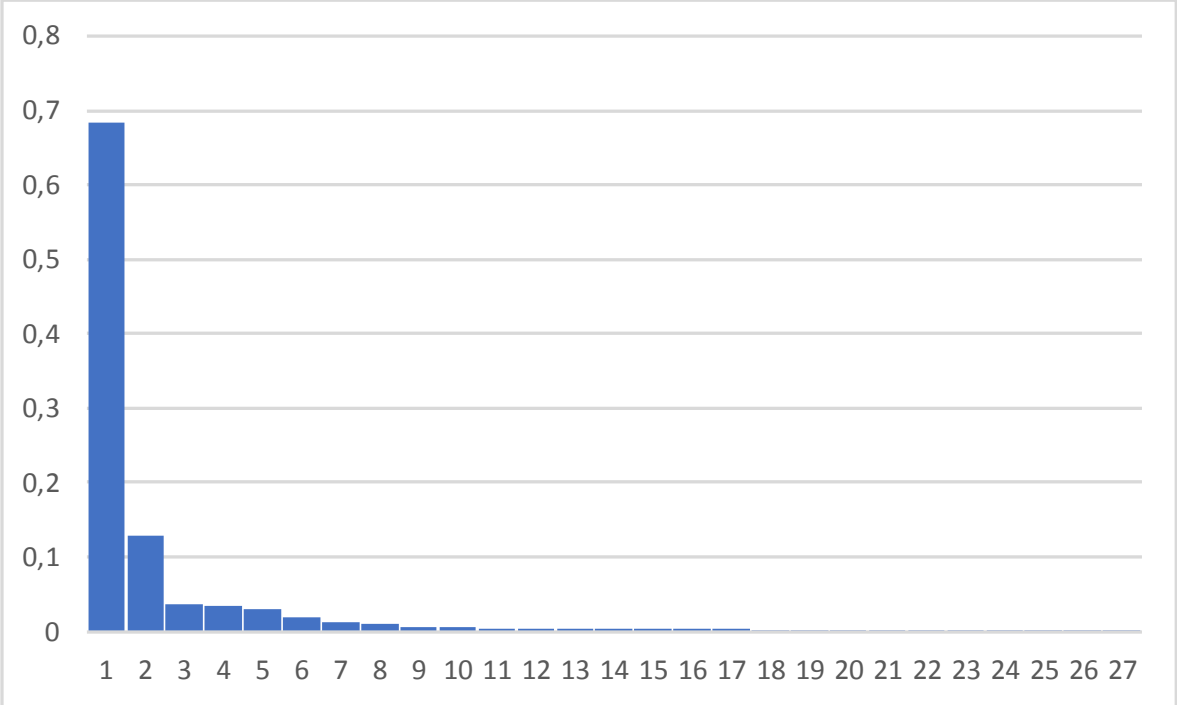


Рис. 10 - Пример уверенной классификации

На Рис. 11 пример классификации другого объекта – видно, что вероятности первой, второй и последующих категорий близки между собой и относительно невелики. Вероятность второй

категории составляет более 80% от первой. Такая классификация при соответствующей настройке порога может быть отнесена к неуверенной.

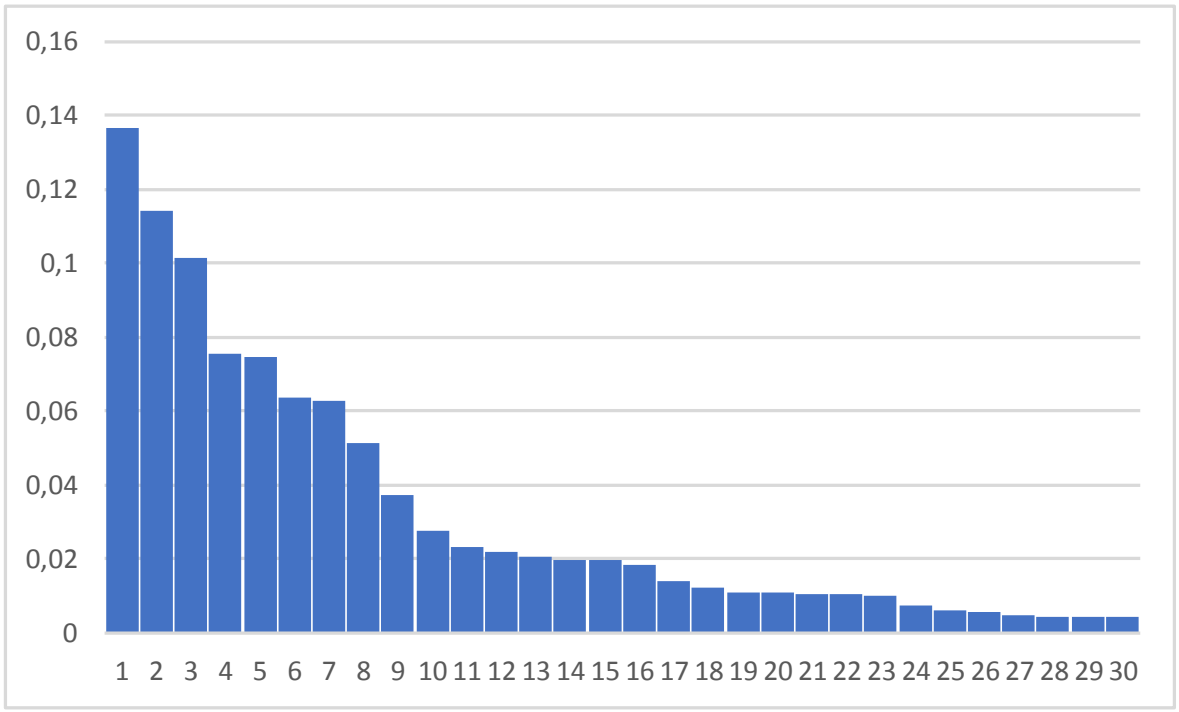


Рис. 11 - Пример неуверенной классификации

Могут наблюдаться и более сложные ситуации – в документе с близкими вероятностями детектируются сразу несколько тематических категорий. Это происходит, например, если в документе обсуждаются сразу две темы см..

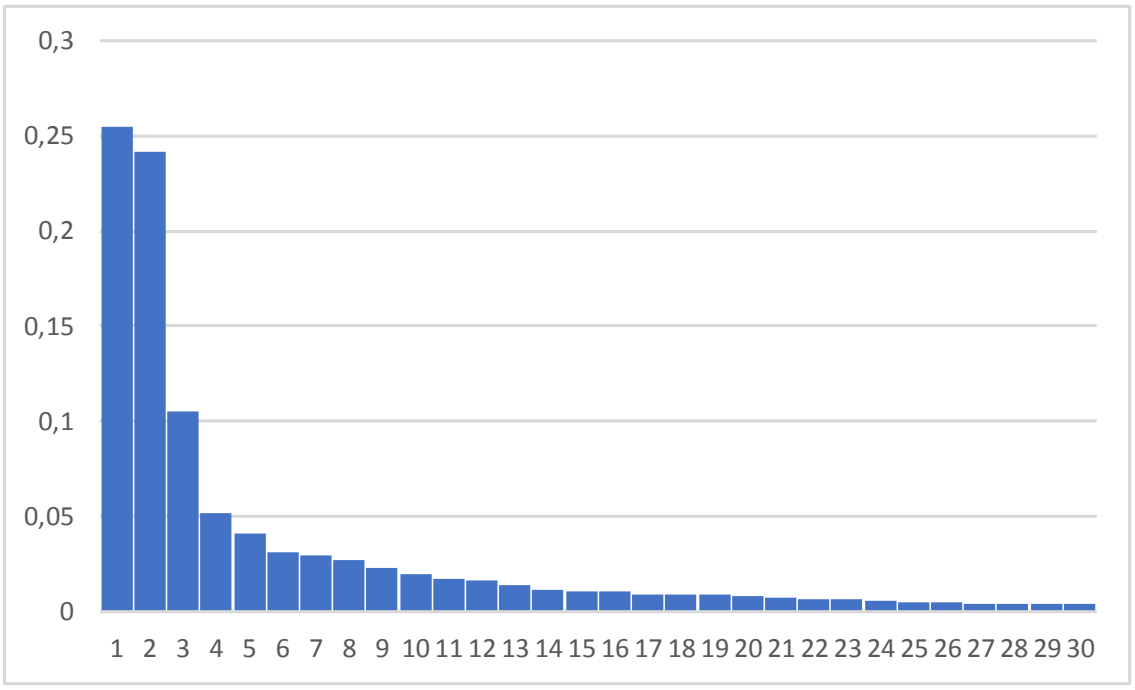


Рис. 12 - Информация уверенно относится к 2-м категориям

2.4.3. Как оценить точность классификации, если тестовая выборка содержит ошибки?

Рассмотрим данный вопрос на конкретном примере. В одном из наших проектов требовалось классифицировать обращения, поступающие по электронной почте на первую линию технической поддержки. Массив данных был разделен в соотношении 3:1 на обучающую и тестовую выборки. После предварительного обучения классификатора и прогона тестовой выборки выяснилось, что зависимость точности классификации от доли автоматически классифицируемых сообщений имеет следующий вид (синяя линия на Рис. 13)

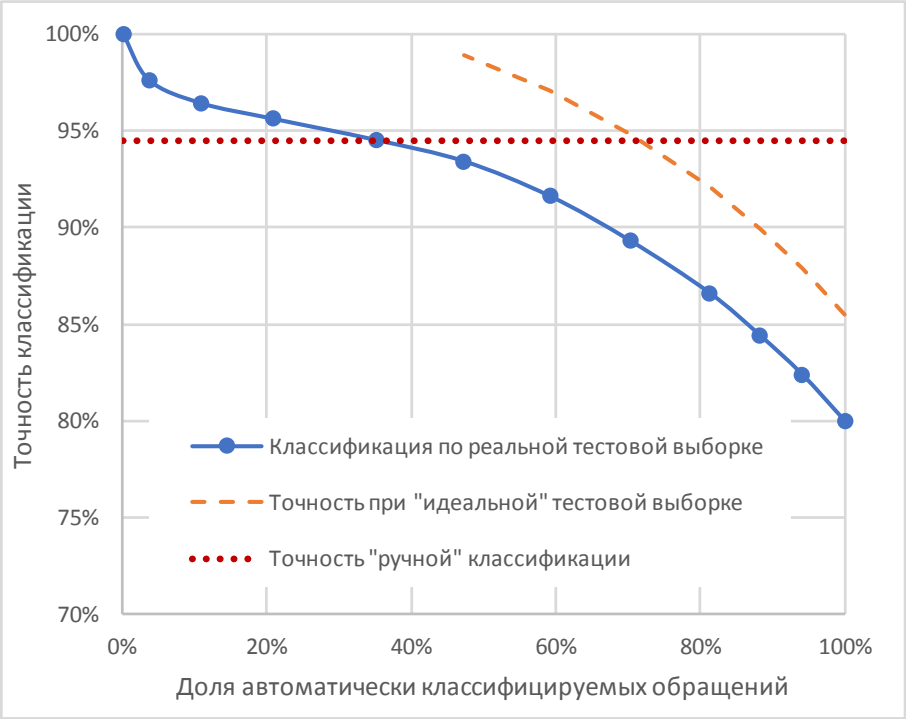


Рис. 13 - Зависимость точности классификации от доли автоматически классифицируемых сообщений (пример)

Видно, что на полном потоке сообщений (автоматически классифицируются все 100% сообщений) демонстрируемая на тестовой выборке точность составляла 80% (правильно классифицировались 4 из 5 сообщений). Повышение доли сообщений, передаваемых на ручную классификацию, или, что то же самое, снижение доли автоматически классифицируемых сообщений проводилось путем настройки порога уверенной классификации[8]. Повышение порога уверенной классификации приводило к росту точности. Так, 80% всего потока

[8] Порог уверенной классификации определяется параметрами о «Зазор между 1 и 2-й рангами» и «Процент опережения 1-й рубрики», см. описание меню «Настройки» системы.

классифицировались на реальной тестовой выборке с точностью 87%. Однако в дальнейшем рост точности несколько замедляется и на 30-40% потока составляет около 94-95%. Еще большее увеличение порога оставляет в автоматически классифицируемом потоке только самые «однозначные» сообщения и точность приближается к 100%. «Перегиб» графика на уровне точности 94-95% (красный пунктир) как раз и связан с наличием человеческих ошибок в тестовой выборке. Особенно наглядно такие характерные «перегибы» можно увидеть на другом графике (см. Рис. 14), на котором показано влияние длины классифицируемого сообщения на точность классификации по различным категориям. Интересно, что уровень и характер человеческих ошибок меняется от категории к категории. В целом можно отметить, что средний уровень точности в 94-95% является вполне типичным для проводимой оператором классификации обращений в ServiceDesk и наблюдается на массивах данных и во многих других организациях. Но, раз уж мы знаем, что уровень человеческих ошибок в тестовой выборке 5-6%, то, следовательно, реальная точность работы классификатора может быть выше на те же самые 5-6% и тогда будет иметь примерно такой вид (оранжевая линия на рисунке):

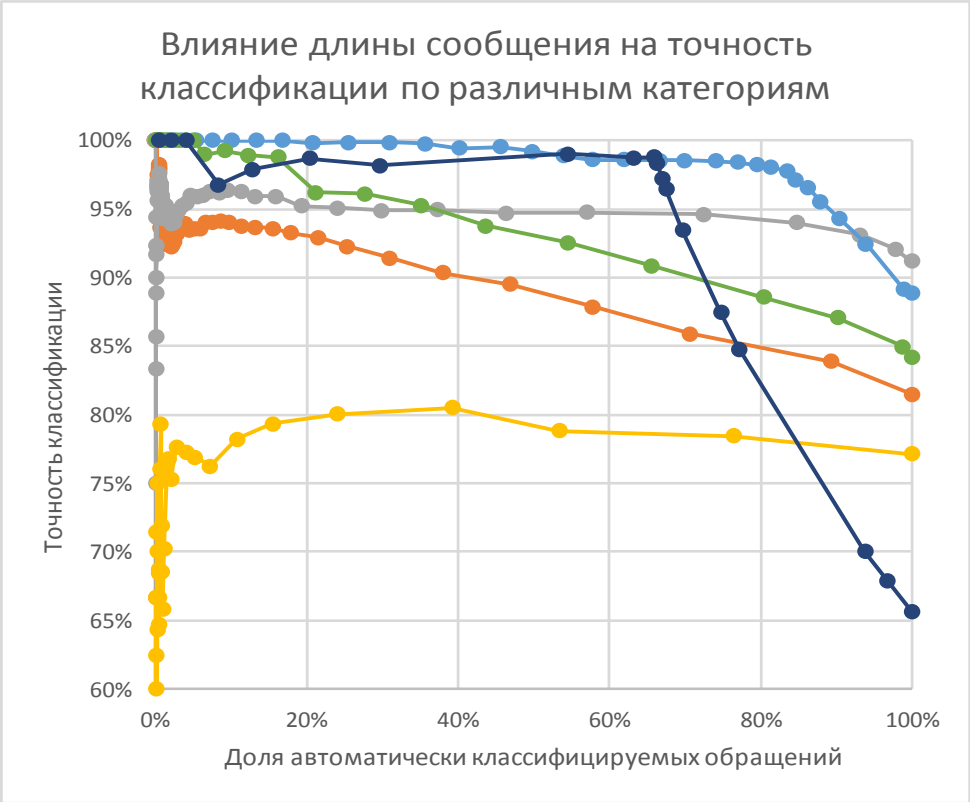


Рис. 14 - Влияние длины сообщения на точность классификации по различным категориям

Отметим, что теоретическое качество работы обученного классификатора позволяет оценить пересечение красной и оранжевой линии на Рисунке 13: классификатор способен

обрабатывать примерно 70% всего потока сообщений с точностью 94-95% (по крайней мере, не хуже, чем это делает человек). Тогда, если на ручном разборе сообщений первой линии техподдержки работает 10 человек, 7 из 10 могут быть заменены автоматической классификацией.

2.4.4. Какие факторы снижают качество классификации?

2.4.4.1. Недостаточное количество информации для обучения

Наверное, это одна из самых частых причин недостаточно высокого качества классификации. Статистические методы машинного обучения начинают работать, когда удается проиндексировать существенную часть семантики[9] классифицируемого информационного потока. Если в каждом новом сообщении хотя бы 80% лексики (слов, словосочетаний и т.д.) уже встречались ранее, используемые для классификации алгоритмы могут выделить важные классификационные признаки и более или менее точно подобрать для сообщения подходящую категорию. В отличие от человека искусственный интеллект пока не способен обучаться на ограниченном количестве примеров. Хотя снижение требований к объему обучающей выборки является одним из активно развиваемых сейчас направлений научных исследований в области машинного обучения, пока для обучения классификатора требуется предъявить ему хотя бы 50-100 Кб[10] текста на каждую категорию. И желательно, чтобы этот текст был разделен не менее чем на несколько десятков документов. Чем больше категорий в классификаторе, тем больший объем обучающей выборки потребуется для достижения приемлемой (не менее 85-90%) точности автоматической классификации.

На Рис. 15 показана зависимость между объемом обучающей выборки и качеством проводимой классификации, на примере данных ОАО «РЖД» за 2016 год:

[9] То есть словарного запаса.

[10] В грубом приближении, в зависимости от кодировки текста, 50-100 страниц.

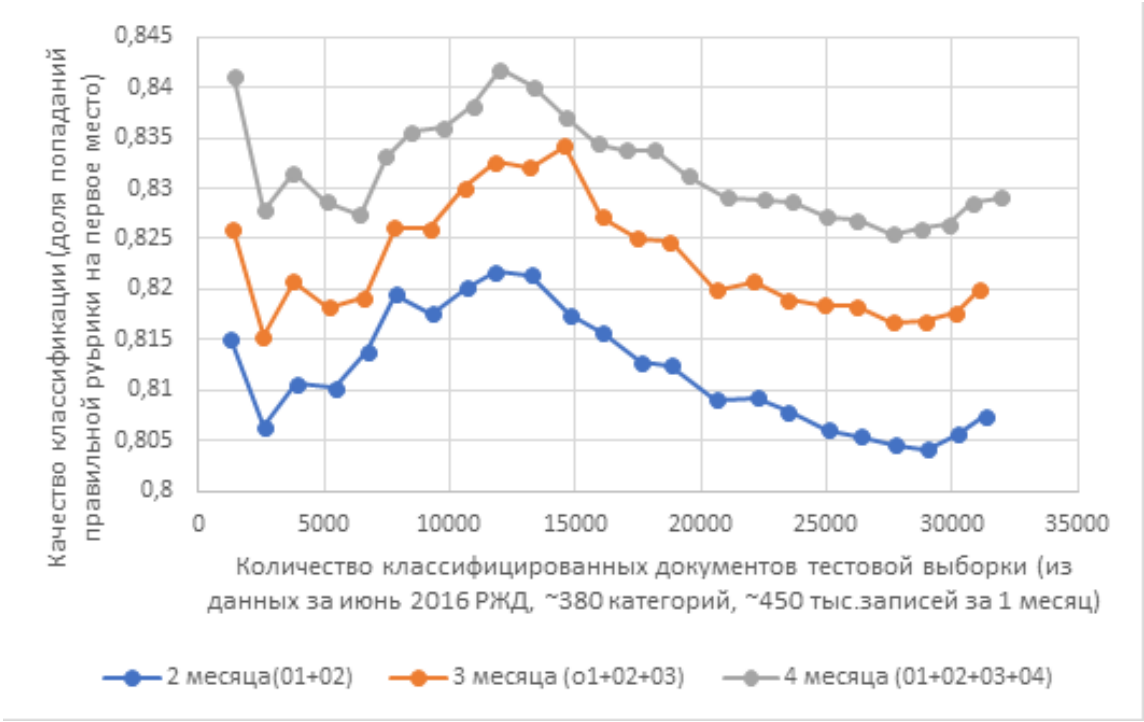


Рис. 15 - Влияние объема обучающей выборки на качество классификации

На Рис. 16 на примере отдельной категории показана зависимость между объемом обучающей значимой информации в категории и качеством проводимой классификации:

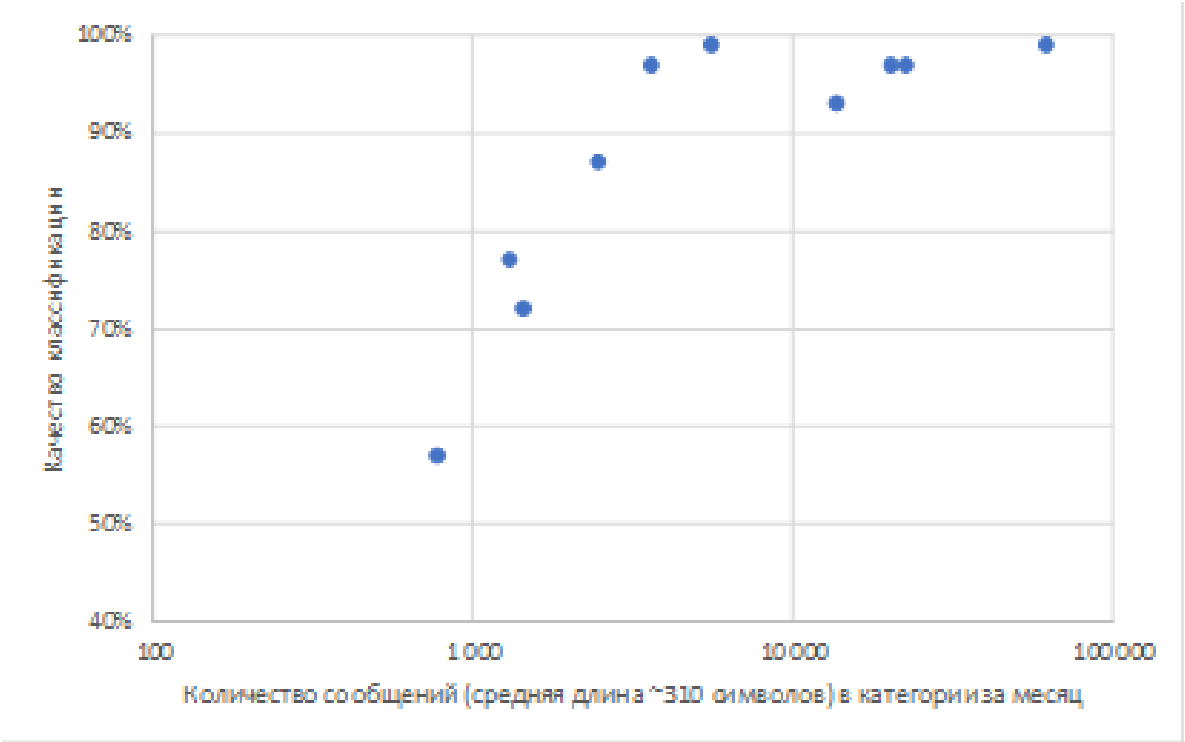


Рис. 16 - Влияние объема доступной информации в категории на качество классификации

2.4.4.2. Слишком большое количество категорий

Это вариант предыдущей ситуации, когда объем выборки на первый взгляд большой, но количество категорий еще больше, так что на все категории информации не хватает. Ситуация становится еще хуже, если учесть неоднородность и неравномерность распределения информации по категориям. Закон Парето (правило 20/80) проявляется к том, что 80% всего потока информации приходятся на 20% всех категорий. Может встречаться и более ярко выраженная нелинейность – например, 90/10. В результате для небольшого количества категорий информации достаточно и даже с большим избытком, а на все остальные – не хватает. Такая ситуация характерна для больших организаций, в которых, например, классификаторы ИТ-услуг эволюционно развивались в течение многих лет, отражая развитие ИТ-инфраструктуры компании. При этом из классификатора не удалялись старые категории, а новые регулярно добавлялись.

Отметим, что работа с огромными классификаторами является очень непростой задачей и для человека. Оператору горячей линии тяжело удержать в голове многие сотни категорий и не ошибаться при классификации входящих сообщений. Высокая текучесть кадров в этой профессии также не способствует накоплению знаний и правильной классификации. Не менее сложна работа с огромными классификаторами и для пользователя информационных систем, которому проходится тратить множество ментальных усилий для заполнения форм на порталах при вводе запросов на обслуживание.

2.4.4.3. Некачественная обучающая выборка

Обучающая выборка – результат классификации информации человеком, а человеку, как известно, свойственно ошибаться. Если человек ошибочно отнес документ к неправильной категории, при обучении классификатора такой документ будет негативно влиять на способность системы распознать правильную категорию. Классификатор как бы «замыливается», теряет резкость. Большинство обучающих выборок, с которыми нам приходилось иметь дело, содержали ошибки такого рода. При формировании тестовых выборок ошибки, естественно, попадают и в них и иногда выявляются при детальном просмотре и анализе результатов машинной классификации.

Самый простой вариант загрязнения обучающей выборки – очевидная техническая ошибка: работая в информационной системе, человек промахнулся с выбором нужной категории из справочника или перенес документ не в ту папку при сортировке файлов на диске. Ошибки такого рода будут очень заметны при автоматической классификации. Их несложно выявить с помощью простого методологического приема: после обучения классификатора необходимо провести

контрольную классификацию обучающей выборки, на которой проводилось обучение. При результативности контрольной классификации, соответствующей 99-100%, можно считать, что обучающая выборка не содержит документов, ошибочно отнесенных в неправильные категории.

Но встречаются и более сложные случаи, когда у разных экспертов имеются разные мнения относительно выбора правильной категории для конкретного документа. Если взять, например, 1000 документов и попросить нескольких человек расклассифицировать их по 50 категориям, почти наверняка результаты классификации у разных экспертов будут отличаться друг от друга. Насколько сильно? Это зависит от сложности поставленной задачи, характера документов, предметной области, жизненного опыта экспертов, их настроения, погоды и от множества других факторов. Как показали наши эксперименты, расхождения такого рода могут превышать 10% от общего количества документов.

При обучении системы ошибки такого рода должны быть согласованы с экспертами. После выработки единого мнения, отнесения спорных текстов в нужную категорию и повторного обучения системы, при проведении дальнейшей классификации результаты работы системы будут являться своего рода централизованным «экспертным» мнением[11].

2.4.4.4. Слишком короткие документы

Сложной для автоматической классификации является ситуация, когда классифицируемые тексты очень короткие – одно, два, три слова – или даже небольшое предложение. Примерами такими коротких текстов могут быть заголовки писем электронной почты или первая фраза, произнесенная абонентом при звонке на горячую линию или краткое содержание входящего документа в СЭД. В короткой текстовой строке значимым обычно оказываются одно-два слова или биграммы и классификация становиться не очень надежной. Хуже того, таких значимых слов может и не обнаружиться вовсе, а все имеющиеся будут близки по значимости к стоп-словам. Если слова «Добрый день» и «Прошу помочь» с равной вероятностью встречаются во всех категориях, то по одной только фразе «Добрый день! Прошу помочь» или «О рассмотрении обращения» даже самый квалифицированный эксперт не догадается, о чём идет речь.

2.4.4.5. Слишком длинные документы

В целом при росте объема классифицируемой информации качество классификации возрастает, однако в этом случае появляется другой риск. Чем больше текста содержит каждый отдельный документ, тем выше вероятность того, что в этом документе затрагиваются сразу

[11] Что является одним из преимуществ технологии автоматизированной классификации.

несколько тем. Например, обращения на горячую линию обычно короткие – до нескольких десятков слов. Как правило, они посвящены только одной проблеме и их классификация будет однозначной. Напротив, обращения граждан в органы государственной власти иногда могут быть очень объемными, занимать несколько страниц текста и затрагивать сразу несколько волнующих человека проблем. При классификации такого рода документов будут выявляться признаки наличия множества тематик. И не обязательно правильная маршрутизация будет определяться доминирующей в документе тематикой. Представим себе письмо, в котором 90% текста посвящены описанию проблем со здоровьем, а в 5% говорится о том, что эти проблемы возникли в результате травмы, полученной из-за ямы на дороге. Такое письмо скорее всего должно быть отправлено в службу, ответственную за жилищно-коммунальное хозяйство для устранения ямы, а не в департамент здравоохранения.

Если тексты в обучающей выборке – это сообщения электронной почты, в них могут присутствовать длинные цепочки предварительных обсуждений, посвященных разным сопутствующим проблемам, могут прилагаться в качестве вложений дополнительные документы, не имеющие прямого отношения к сути проблемы. Наконец, во многих организациях принято добавлять в конец сообщений электронной почты объемные подписи, разного рода «дисклаймеры», упоминания требований к конфиденциальности, которые снижают качество обучения.

2.4.4.6. Изменчивость или неоднородность обучающей выборки

Распределение информационного потока по категориям может меняться с течением времени. И для этого даже не обязательно ожидать появления новых категорий – могут заметно меняться смысловая наполненность одних и тех же категорий и соотношение между различными категориями в информационном потоке.

Например, заявки на горячую линию могут отражать особенности различных этапов эксплуатации корпоративных информационных систем. Вначале внедрения в потоке обращений встречается большое количество вопросов, обусловленных новизной системы для пользователей, происходящими изменениями в бизнес-процессах, особенностями интеграции новой системы в ландшафт корпоративных информационных систем. В ходе эксплуатации уже внедренной информационной системы уменьшается количество вопросов и заметно меняется их содержание: пользователи привыкли к новой системе и на горячую линию поступают другие вопросы про выдачу прав доступа новым сотрудникам, архивирование данных, устранение сбоев и т.п.

Для контекста обращений граждан в органы государственной власти характерна зависимость тематики некоторых категорий от времени года. Кроме того, у отдельных групп граждан наблюдаются сезонные обострения, это явление хорошо знакомо сотрудникам, регистрирующим обращения граждан в системах электронного документооборота.

В управленческом документообороте маршрутизация входящих официальных документов по резолюциям руководства может меняться в результате перераспределений обязанностей и полномочий между структурными подразделениями организации. Такая изменчивость информационного потока приводит к тому, что обученная система через некоторое время начинает «отставать от жизни», что будет проявляться как снижение точности классификации.

2.4.4.7. Наличие в тексте ошибок распознавания текста (ошибок OCR)

Данная проблема возникает, когда требуется классифицировать поток документов, в котором встречаются распознанные рукописные документы или типовые формы документов с заполненными от руки полями. Безусловно, снижение качества распознавания текста пропорционально снизит и качество проводимой классификации.

2.5. API системы Preferentum.Class

Система Preferentum.Class написана на C# .NET (управляемый код) и является самодостаточной системой, которая предоставляет API для своей интеграции в пользовательские информационные системы. Возможны следующие способы интеграции:

- непосредственное использование классов .NET сборок DLL – для приложений на .NET;
- через REST-протокол взаимодействия с модулем CLASS.Server.exe, который является кроссплатформенным и работает как на Windows, так и на Unix, Mac и Android (под управлением Mono) без каких-либо предварительных установок;
- через аналогичный REST-протокол с Web-сервером, устанавливаемом на IIS (Internet Information Server) под Windows;

2.5.1. Базовый сценарий использования

Каждый классификатор располагается в своей директории локального (по отношению к серверу) компьютера. Изначально директория пуста. Обучение состоит в вызове внешней системой функции **ADD(rubric, text)** нужное число раз, подавая на вход указание рубрики (если такой ещё нет, то она будет добавлена) и неструктурированного текста.

Для того чтобы добавляемая информация стала доступной для классификации, необходимо выполнить операцию COMMIT(), во время которой происходит перерасчёт всех статистик. ВНИМАНИЕ! Поскольку данная операция может оказаться продолжительной (до десятка минут для большого индекса), то выполнять её следует после добавления всех текстов, а не после каждого ADD.

Классификация производится функцией **CLASSIFY(text)**, которая для входного текста возвращает ранжированный список пар [rank, rubric]. Отметим, что классификация может производиться параллельно с дообучением, просто до выполнения COMMIT рубрики будут определяться на основе информации, полученной в результате последнего коммита.

Система Preferentum.Class предоставляет функцию **TOTEXT(file)**, которая выделяет тексты из файлов практически любых форматов – DOC, DOCX, PDF, RTF, TXT, HTML, ODT и др. Для архивов будет возвращаться суммарный текст, который удалось выделить из внутренних файлов. Но тексты из картинок не выделяются, так как здесь не используется OCR, а только непосредственно текстовая информация, если таковая присутствует.

2.5.2. Интеграция с приложением .NET

Основные функции находятся в сборке CLASS.Core.dll, которая должна быть подключена к проекту. Помимо этого, должны быть также подключены все dll, имеющие префикс ITS и EP. После чего можно напрямую использовать классы:

```
/// <summary>
/// Служба классификации
/// </summary>
public static class EngineService
{
    /// <summary>
    /// Текущая версия
    /// </summary>
    public static Version Version = new Version(0, 12);
    /// <summary>
    /// Вызывать в самом начале
    /// </summary>
    public static void Initialize() { }
    /// <summary>
    /// Вызывать в самом конце
    /// </summary>
    public static void Deinitialize() { }
```

```
    /// <summary>
    /// Создать экземпляр движка работы с классификатором
    /// </summary>
    /// <returns></returns>
    public static IClassificator CreateClassificator() { }

    /// <summary>
    /// Извлечение текста из файла практически любого формата
   (DOC, DOCX, PDF, RTF, TXT, HTML ...)
    /// Из архива будут взяты все файлы и вернёт объединённый текст этих файлов)
    /// </summary>
    /// <param name="fileName">имя файла (может быть null)</param>
    /// <param name="content">содержимое файла (если null, то
   fileName должен быть полный путь к локальному файлу)</param>
    /// <returns>текст или null при невозможности выделения</returns>
    public static string ToText(string fileName, byte[] content = null);
}

/// <summary>
/// Классификатор
/// </summary>
public interface IClassificator : IDisposable
{
    /// <summary>
    /// Инициализировать движок
    /// </summary>
    /// <param name="indexPath">директория с индексом</param>
    void Open(string indexPath);

    /// <summary>
    /// Конфигурация (доступна после открытия индекса)
    /// </summary>
    ClassConfig Config { get; }

    /// <summary>
    /// Заново начать работать (после Close)
    /// </summary>
    void Start();
    /// <summary>
    /// Вызывать в конце работы с движком
```

```
        /// </summary>
        void Close();
        /// <summary>
        /// Очистить весь индекс
        /// </summary>
        void ClearAll();

        /// <summary>
        /// Добавить рубрику (если рубрика с таким именем есть, то не будет добавляться).
        /// </summary>
        /// <param name="name"></param>
        void AddRubric(string name);
        /// <summary>
        /// Добавить текст рубрики (обучение)
        /// </summary>
        /// <param name="rubricName">если рубрика отсутствует, то будет добавлена</param>
        /// <param name="text"></param>
        void AddText(string rubricName, string text);
        /// <summary>
        /// Пересчитать статистики после процесса обучения
        /// </summary>
        void Commit();

        /// <summary>
        /// Получить список рубрик
        /// </summary>
        /// <returns></returns>
        List<RubricInfo> GetRubrics();
        /// <summary>
        /// Классифицировать текст
        /// </summary>
        /// <param name="text">классифицируемый текст</param>
        /// <param name="detailInfo">при true будет формировать
              детализацию Items у RubricInfo</param>
        /// <returns>упорядоченный по рангам список рубрик</returns>
        List<RubricInfo> Classify(string text, bool detailInfo);
    }
    /// <summary>
    /// Элемент результата классификации (привязка к рубрике)
    /// </summary>
    public class RubricInfo : IComparable<RubricInfo>
```

```
{
    /// <summary>
    /// Ранг рубрики
    /// </summary>
    public float Rank;
    /// <summary>
    /// Имя рубрики
    /// </summary>
    public string Name;
    /// <summary>
    /// Количество термов у рубрики
    /// </summary>
    public int Terms;
    /// <summary>
    /// Количестиво биграмм термов у рубрики
    /// </summary>
    public int Bigrams;
    /// <summary>
    /// Список элементов (термов и биграмм) с коэффициентами связи с рубрикой
    /// </summary>
    public Dictionary<string, float> Items = null;
}
```

2.5.3. REST-протокол взаимодействия с EXE-сервером

Сервер Preferentum.Class реализован как консольное приложение CLASS.Server.exe, которое работает под управлением .NET Framework 4.0 и выше. В современных версиях Windows этот фреймворк уже встроен, а для Unix, Mac и Android существуют бесплатные версии под названием Mono, которые должны быть предварительно установлены – в это случае запуска из командной строки производится как Mono ИМЯ_ПРИЛОЖЕНИЯ.

После запуска и инициализации сервер слушает указанный в настойках порт TCP, по которому к нему приходят обращения в виде REST-подобного запроса, и возвращает результат в формате XML или JSON. Корректная остановка сервера производится путём нажатия консольной клавиши ESC или REST-команды STOP. Некорректная остановка может привести к порче открытого индекса, если прерывание случилось в момент перерасчёта статистик.

Параметры серверу задаются в командной строке (лучше сделать bat-файл) (см. Таблица 1).

Таблица 1 – Параметры сервера

Ком.строка	***По умолчанию***	***Описание***

-port	4567	Номер TCP-порта, по которому идёт взаимодействие
-address	*	Если нужно указать конкретный IP-адрес текущего компьютера
-outformat	xml	Формат возвращаемых результатов Xml или Json
-index	нет	Директория индекса основного классификатора (который используется по умолчанию, если в запросе классификатор не указан)
-allindexes	нет	Если несколько классификаторов, то их индексы должны располагаться внутри одной директории, при этом имя подпапки является именем классификатора.

Пример запуска: `mono CLASS.Server.exe –port 7080 –index "C:\Test\Index"`

Для проверки связи с сервером используйте утилиту TestServer.exe с параметрами –address и –port, она только отправляет запрос версии (command=version).

Опишем протокол взаимодействия. В результате любой операции сервер возвращает XML или JSON файл в кодировке UTF-8. В случае какой-либо ошибки там будет узел error с её кратким описанием:

```
<result><error>Описание ошибки</error></result>
```

```
{ "error" : "Описание ошибки" }
```

Параметры REST-запроса оформляются стандартно: в URL они разделяются амперсандом и имеют вид КЛЮЧ=ЗНАЧЕНИЕ. Поступающие на вход параметры дублируются соответствующими узлами результирующих XML\JSON. К основным параметрам относятся:

- out – формат результата (json или xml), если не задан, то берётся из настроек сервера;
- name – имя классификатора, если не задан, то берётся тот, который установлен у сервера по умолчанию через параметр сервера IndexPath, если задан, то это имя поддиректории в директории, заданной параметром сервера AllIndexesPath;
- command – обязательный параметр, определяющий команду. Возможные значения: ADD, COMMIT, CLASSIFY, CLEAR, STOP, VERSION

Команда VERSION: возвращает пустой результат с версией, используется для проверки связи с сервером.

Команда ADD: обучение классификатора, дополнительные параметры:

- rubric – имя рубрики, обязательный параметр (если такая рубрика отсутствует, то будет добавлена)
- text – обучающий текст

Вместо текста можно в виде контента запроса передавать файл, и тогда текст по возможности будет выделен из него (см. TOTEXT раздела «Базовый сценарий»). Формат файла система определяет по содержимому, но для большей точности можно параметром filename передать имя файла. Для контроля выделения в возвращаемый xml\json помещается узел с первыми 300 символами текста.

Команда COMMIT: фиксация обучения классификатора, дополнительных параметров нет.

Команда CLEAR: очистка классификатора, удаление всех рубрик и статистики.

Команда CLASSIFY: классификация текста, дополнительные параметры:

- text – классифицируемый текст, вместо этого параметра можно передать файл через контент запроса (аналогично как для ADD);

Результат классификации в json\xml оформляется узлами rubric. Вот пример результатов для запрос http://localhost:4567/?command=classify&text=%D0%A8%D0%BB%D0%B0%20%D1%81...:

```
<result version="0.12">
 <command>CLASSIFY</command>
 <text length="32">Шла саша по шоссе и сосала сушку</text>
 <rubric rank="0.136">Технологические решения</rubric>
 <rubric rank="0.102">Мероприятия по охране окружающей среды</rubric>
 <rubric rank="0.080">Сети связи</rubric>
</result>
```

Или аналогичный в JSON:

```
{ "version":"0.12",
 "command":"CLASSIFY",
 "text":"Шла саша по шоссе и сосала сушку",
 [   { "rank":"0.136", "rubric":"Технологические решения" },
```

{ "rank":"0.102", "rubric":"Мероприятия по охране окружающей среды" },
{ "rank":"0.080", "rubric":"Сети связи" }]
}

2.5.4. REST-протокол взаимодействия с Web-сервером

Система Preferentum.Class имеет аналогичный по функционалу EXE-серверу Web-сервер, который устанавливается под Windows и работает под управлением IIS (Internet Information Server). Помимо REST-протокола, сайт этого Web-сервера предоставляет некоторый браузерный интерфейс, служащий в основном демонстрационным и тестовым целям – он не предназначается конечному пользователю.

Параметры REST-протокола абсолютно аналогичны описанным выше, только в URL нужно обязательно добавлять Rest.ashx. Например:

http://localhost:4567/**Rest.ashx**?command=classify&text=%D0%A8%D0%BB%D0%B0%20%D1%81...

Установка под IIS стандартна, требуется .NET Framework 4.0 или выше.

Некоторые параметры устанавливаются в Web.config.xml:

- IndexPath – директория основного классификатора, используемого, когда имя классификатора не задано в запросе. По умолчанию, это App_Data\Index в серверной папке;
- AllIndexesPath – директория для поддиректорий с другими классификаторами. По умолчанию, это App_Data.

Протокол работы сервер пишет в App_Data\log.txt