



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В. ЛОМОНОСОВА
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Треско Константин Игоревич

Исследование и разработка средств многотемной классификации веб-страниц

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научные руководители:
к.ф.-м.н. М.И. Петровский
Д.В. Царев

Аннотация

Целью данной работы является исследование и разработка системы сбора и многотемной классификации веб-страниц, с которыми работали пользователи, находящиеся внутри одной сети. Разрабатываемая система сбора должна удовлетворять требованиям масштабируемости (линейного роста расхода ресурсов при увеличении числа подключений) и защищенности (пользователь не должен иметь возможности фальсифицировать собранные данные). Модуль классификации данных должен обеспечивать многотемную классификацию на основе машинного обучения с возможностью добавления и удаления тематик.

В ходе работы был проведен обзор существующих средств многотемной классификации и средств реализации системы сбора. Также была реализована система сбора и многотемной классификации, удовлетворяющая поставленным выше требованиям, проведено тестирование, показавшее приемлемую масштабируемость.

1 Введение

1.1 Задача классификации

Настоящая работа посвящена исследованию и разработке программных средств сбора и многотемной классификации текстовых данных веб-страниц. Задача классификации многотемных документов (multi-label classification), заключается в определении принадлежности документа к одному или нескольким классам (из predetermined набора классов) на основании анализа совокупности признаков, характеризующих данный документ. В отличие от традиционной задачи классификации, классы могут пересекаться или быть вложенными, то есть документ может принадлежать нескольким классам. Классы, к которым принадлежит документ называются релевантными.

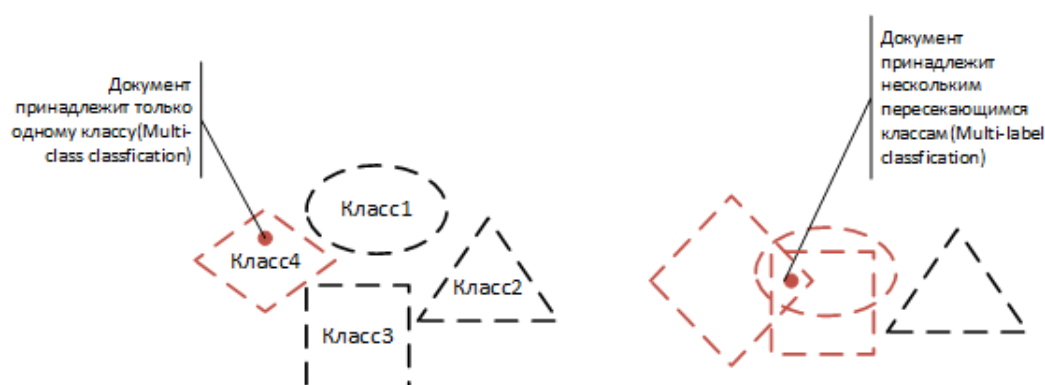


Рисунок 1 – Многотемная и многоклассовая классификация

1.2 Области применения классификации веб-данных

На сегодняшний день существует ряд задач, для решения которых требуются системы сбора и классификации контента, частным случаем которого является веб-контент:

- **Анализ работы сотрудников организации.** Определение тематик документов, с которыми работает пользователь. На основе полученной информации применение политик безопасности

- **Фильтрации контента, для обнаружения информации, относящейся к определенному делу (eDiscovery[3]).** Документы и электронная почта могут фильтроваться в зависимости от присвоенных им классов, чтобы гарантировать, что только материалы с запрошенной бизнес-информацией в рамках расследуемого дела были найдены и сохранены.
- **Определения конфиденциальности документа, для предотвращения утечек информации** По различным экспертным оценкам в настоящее время наибольшие риски для информационной безопасности представляют не внешние, а внутренние угрозы [1] Поэтому существует необходимость в средствах определения конфиденциальности документа

Далее будут рассмотрены классы индустриальных систем, функционал которых включает в себя классификацию текстовых данных (в том числе веб-данных), с которыми работают пользователи.

- **Системы управления корпоративным контентом (англ. Enterprise Content Management, ECM)** - программные решения для управления информационными ресурсами предприятия предоставляют программные средства сбора, анализа, управления, накопления, хранения и доставки документов в масштабах организации. В настоящее время большинство ECM систем включают в себя системы обнаружения данным, связанных с определенным судебным делом (англ. eDiscovery). Средства eDiscovery обеспечивают процесс, с помощью которого организации находят, получают, сохраняют и анализируют документы, связанные с делом.
- **Системы предотвращения утечки данных (англ. Data Loss Prevention, DLP)** - программные решения для предотвращения утечек конфиденциальной информации и минимизации других рисков, связанных с внутренними угрозами.

1.2.1 Основные методы сбора и классификации данных в корпоративных системах

Поиск и анализ информации в ECM системах происходит следующим образом:

- **Сбор данных.** Системы сбора содержат компоненту, установленную на пользовательском компьютере и отвечающую за мониторинг деятельности пользователя. В терминологии IBM данные компоненты называются Искателями [6]
- **Применение методов анализа документов.** В терминологии IBM данный этап называется аналитическим конвейером [7].
- **Индексация данных.** Компоненты индексации добавляют в индекс информацию о новых и измененных документах [8, 9, 10]

Существуют три основных подхода к классификации данных в ЕСМ системах

- **Классификация на основе обучающей выборки** (IBM Content Classification [2], Symantec eDiscovery [4])
- **Классификация на основе заданных правил.** Принадлежность документа к тому или иному классу определяется на основе эвристических правил (сигнатур). Правила могут формироваться на основе контекста – имя отправителя, директория создания и т.п., а также на основе контента – шаблоны текста, ключевые слова и т.п.
- **Определение категорий в неизвестных документах.** В задаче кластеризации нет предопределенного набора классов. Исходное множество документов разбивается на подмножество таким образом, чтобы документы в различных подмножествах существенно отличались.

В DLP системах выделяют следующие технологии классификации данных:

- **Цифровые отпечатки (англ. digital fingerprint)** - технология предназначена для защиты больших по объему документов, содержание которых не изменяется или меняется незначительно. Детектор цифровых отпечатков позволяет автоматически обнаруживать в анализируемом тексте цитаты из документов-образцов, содержащих конфиденциальную информацию.
- **Анализ шаблонов** - технология предназначена для детектирования алфавитно-цифровых объектов по шаблону данных (маске) и позволяет наиболее эффективно выявлять факты пересылки персональных данных или финансовой информации. Кроме того, данная технология может использоваться как вспомогательный метод для обнаружения фактов несанкционированной пересылки внутренних документов, содержащих формализованные данные, образованные по определенному шаблону (например, договоров или счетов в случае детектирования банковских реквизитов, кодов классификаторов и т.д.).
- **Машинное обучение** (InfoWatch [5], Symantec VML [11]). Примерная схема работы классификатора такова: на вход подается обучающий набор документов, состоящий как из конфиденциальных так и не конфиденциальных документов. По этим наборам производится обучение и строится статистическая модель. Далее полученная модель используется для классификации неизвестным документов. При обнаружении конфиденциального документа производятся действия, предписанные политиками безопасности.

Основное преимущество машинного обучения заключается в том, что в отличие от других описанных технологий, оно предназначено для работы не со статическими, а с постоянно меняющимися документами.

1.3 Специфика задачи классификации текстовых веб-данных

Веб – страницы имеют следующие особенности

- Содержимое веб-страниц не подлежит шаблонному разбору
- Содержимое имеет многотемную природу, то есть каждая из страниц может одновременно принадлежать к нескольким тематикам



Рисунок 2 – Многотемная природа документа

1.4 Выводы

На сегодняшний день существует ряд прикладных задач, требующих классификации текстовых данных. Обзор индустриальных систем показал, что наиболее актуальными технологиями классификации текстовых данных являются:

- Метод шаблонов
- Метод цифровых отпечатков
- Метод машинного обучения

Первые две технологии применимы только к статическим данным, в то время как метод машинного обучения позволяет адаптироваться к тому, что содержимое и состав анализируемых данных постоянно меняется.

Для задачи классификации текстовых веб-данных из рассмотренных технологий подходит только машинное обучение с возможностью дообучения, так как веб – данные и набор классов постоянно меняются.

Рассмотренные системы решают задачу многоклассовой классификации, так как документ может принадлежать только к одному из predefined набора

классу. Так как большая часть веб – данных имеет многотемную природу, то существует актуальность разработки системы сбора и многотемной классификации текстовых веб - данных пользователей. С учетом большого объема анализируемой информации к модулю классификации предъявляется требование масштабируемости.

Так как в рассмотренных системах присутствуют компоненты сбора информации, расположенные на пользовательских машинах, то при разработке системы сбора нужно учитывать то, что деятельность компонент никак не должна сказываться на работе пользователя, то есть к системе предъявляется требование производительности. Также в рассмотренных системах пользователь не имеет доступа к собираемой информации, поэтому система сбора должна удовлетворять требованию защищенности.

2 Постановка задачи

Разработать архитектуру и реализовать прототип системы сбора и многотемной классификации текстовых веб-данных пользователя в соответствии с требованиями:

- Модуль сбора должен обеспечивать
 - Масштабируемость (линейный рост расхода ресурсов при росте числа подключений)
 - Производительность (компоненты сбора, установленные на пользовательских машинах, не должны влиять на работу пользователя)
 - Защищенность (пользователь не должен иметь возможности фальсифицировать данные)
 - Функционирование под ОС Windows и браузером IE
- Модуль классификации должен обеспечивать
 - Многотемную классификацию на основе машинного обучения с возможностью дообучения

3 Обзор

Проведенный анализ систем ЕСМ и DLP в контексте задач классификации показывает, что каждый из пользователей является источником анализируемой информации, собираемая информация должна храниться в единой базе данных. Для управления потоком данных из различных источников используется мульти-агентная система [16]. Под агентом подразумевается автономный процесс, способный реагировать на среду исполнения и вызывать изменения в среде, возможно, в кооперации с пользователями или с другими агентами.

Разрабатываемая система должна состоять из:

- Агентов мониторинга, расположенных на каждом из пользовательских компьютеров, собирающих текстовые данные просматриваемых пользователем веб – страниц и отправляющих их агенту консолидации
- Агента консолидации, принимающего данные от всех агентов мониторинга и сохраняющего их в единую базу данных
- Модуля классификации

Агент мониторинга состоит из двух компонент

- Расширение для браузера, сохраняющее html код просматриваемой пользователем веб – страницы в локальную базу данных, расположенную на пользовательском компьютере
- Модуль передачи информации из базы данных агенту консолидации

3.1 Расширение для браузера

Для решения задачи необходимо расширение для браузера, имеющее возможность сохранить html код просматриваемой пользователем веб – страницы в локальную базу данных. Так как система сбора должна удовлетворять требованиям производительности, то сохранение и обработка данных должны происходить без участия пользователя.

- **ВНО** (Browser Helper Object) - DLL-модуль, разработанный как плагин для Internet Explorer для обеспечения дополнительной функциональности. В ВНО API существует возможность получения доступа к DOM текущей страницы. Также существует возможность обработки событий и управления навигацией. Для задачи перехвата контента данное решение подходит. Минусом является то, что данное решение подходит только для браузера IE. Но ВНО имеет возможность записи и чтения данных из файловой системы пользователя, без его участия

Таблица 1 – Средства расширения для браузера

	ВНО	Crossrider	Kynext	WebMynd
Кроссплатформеность	-(только IE)	+	+	+
Бесплатность	+	+	+	-
Поддерживаемые языки	C++,C#	JavaScript	Kynext Rules Language	JavaScript
Возможность записи в локальную ФС	+	-	-	-

- **Kynext** позволяет перехватывать содержимое веб – страницы. Поддерживает IE, Firefox, Safari, and Chrome, но для написания необходимо использовать проприетарный язык Kynext Rules Language. Еще одним минусом является то, что работа написанного расширения зависит от работоспособности самого расширения Kynext.
- **WebMynd** поддерживает IE, Firefox, Safari, and Chrome. Пока доступна бета версия, и не известно, будет ли дальнейшая поддержка продукта. Поддерживает JavaScript API. Не является бесплатным программным обеспечением.
- **Crossrider** позволяет быстро создавать кросс-браузерные расширения. Использует один API и поддерживает JavaScript и jQuery, так что разработчик с базовыми знаниями JavaScript может писать и поддерживать свой код. Имеется возможность писать под IE, Firefox, Safari, and Chrome. Бесплатен в использовании. Существует документация и демо – видео для некоторых задач. Доступ к файловой системе пользователя только с его разрешения.

3.1.1 Выводы из обзора средств реализации расширения браузера

Компоненты системы сбора, установленные на пользовательских машинах не должны влиять на работу пользователя. Только ВНО при сохранении файла не требует подтверждения пользователя.

Таким образом для решения задачи сбора подходить только ВНО, так как это средство может записывать в любое место на диске пользователя без его подтверждения.

3.2 Методы многотемной классификации

Можно выделить три основных подхода к многотемной классификации [20]

- «Оптимизационный» подход
- Подход на основе декомпозиции в набор независимых бинарных проблем
- Подход на основе ранжирования

3.2.1 Методы, основанные на "оптимизационном" подходе

К оптимизационным относятся методы классификации, в которых в явном виде задан показатель качества, который необходимо обратить в экстремум (максимум или минимум) по множеству допустимых разбиений.

К данным алгоритмам можно отнести:

- Основанные на AdaBoost алгоритме (AdaBoost.MH [17], ADTBoost.MH [18])
- минимизируется функция Hamming Loss)
- Multi-Label-kNN - [19] максимизируются апостериорные вероятности принадлежности классам

Основной недостаток алгоритмов, основанных на оптимизационном подходе, заключается в том, что для решения задачи оптимизации необходимо хранение всего тренировочного набора, таким образом, нет возможности добавления и удаления тематик или дообучения

3.2.2 Методы, основанные на декомпозиции в набор независимых бинарных проблем

Кратко суть методов, основанных на декомпозиции в набор независимых бинарных проблем, можно описать так:

- Пусть дано N классов. Для каждого из N классов строится бинарный классификатор
- Далее, основываясь на решающей функции для каждого из классификаторов, можно определить релевантность класса

Основным поводом для критики методов этой группы является то, что строятся независимые классификаторы, которые не учитывают корреляцию между классами, что существенно для многотемной классификации. Еще одним недостатком является высокая вычислительная сложность, так как количество бинарных подзадач равно числу классов, а каждая подзадача обучается на всём тренировочном наборе.

3.2.3 Методы, основанные на подходе ранжирования с последующим отсечением нерелевантных классов

Работу алгоритмов можно разделить на два этапа

- Первый этап состоит в обучении алгоритма ранжирования, который упорядочивает все классы по степени их релевантности для заданного классифицируемого объекта
- Второй этап заключается в построении функции многотемной классификации, отделяющей релевантные классы от нерелевантных

3.3 Выводы из обзора методов многотемной классификации

Обзор существующих методов классификации многотемных документов показал, что обозначенным в постановке задачи требованиям удовлетворяют только методы, основанные на подходе «каждый-против-остальных», но качество классификации не удовлетворяет потребностям прикладных задач.

Поэтому был выбран алгоритм, разработанный в лаборатории Технологий Программирования, основанный на подходе попарных сравнений и удовлетворяющий всем поставленным требованиям.

4 Исследование и построение решения

4.1 Разбиение на подзадачи

Для решения задачи сбора и многотемной классификации текстовых данных веб – страниц пользователя был выбран мультиагентный подход [?], поэтому система состоит из:

- **Агента мониторинга**, собирающего информацию и отправляющего ее агенту консолидации
- **Агента консолидации**, сохраняющего информации со всех источников в единую базу данных
- **Модуля классификации**, предоставляющего возможность определения принадлежности документа к одному из предопределенного набора классов

4.1.1 Агент мониторинга

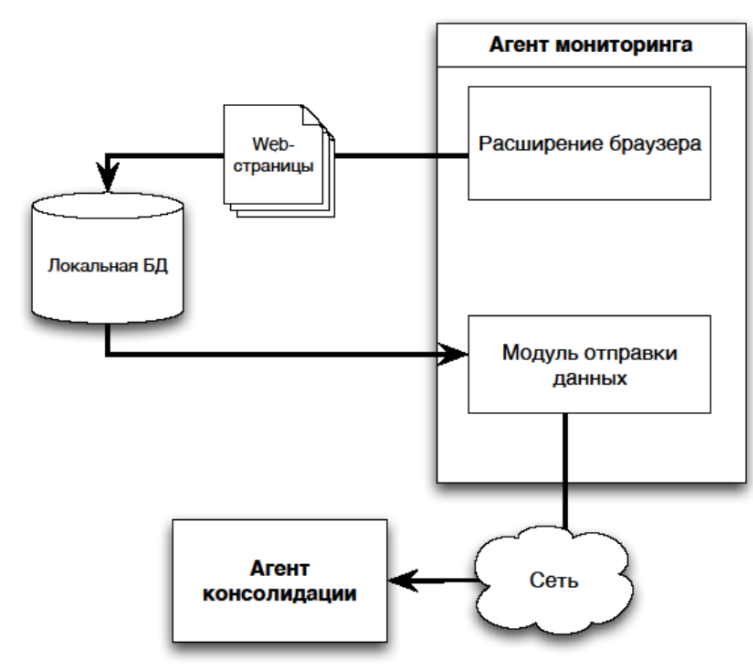


Рисунок 3 – Архитектура агента мониторинга

Агент мониторинга состоит из нескольких компонент:

- Расширение для браузера
- Модуль передачи данных агенту консолидации

Расширение для браузера, написанное с помощью ВНО, считывает html код просматриваемой пользователем веб-страницы и сохраняет ее в локальную базу данных. Сохранение в локальную базу данных осуществляется для контроля нагрузки на агент консолидации и возможности отправления данных по расписанию, а также на случай потери связи с агентом консолидации.

Модуль передачи данных агенту подключается к локальной базе данных и отправляет хранящуюся в ней информацию агенту консолидации, при получении ответа от агента консолидации отправленные данные удаляются из локальной базы данных, чтобы размер локальной базы данных не увеличивался постоянно.

Каждый из пользователей, является источником собираемой информации. При этом задача требует того, чтобы пользователь не имел доступа к собираемым данным. В случае реализации агента мониторинга с помощью ВНО защищенность может осуществляться с помощью установки прав доступа на папку, в которой хранятся собираемые данные

4.1.2 Агент консолидации

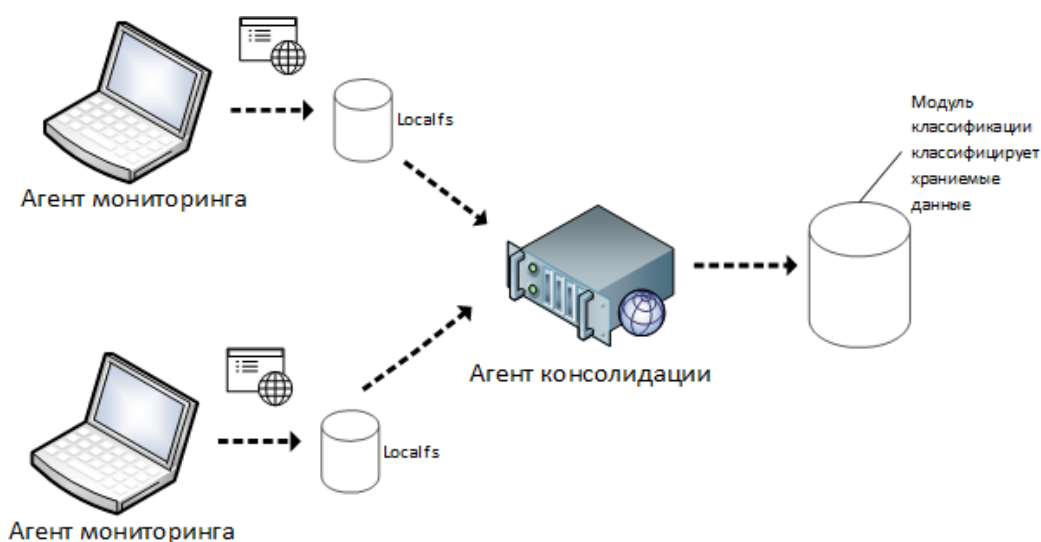


Рисунок 4 – Архитектура агента сбора

Агент консолидации сохраняет данные, полученные от всех агентов мониторинга, в единую базу данных. Также в базу данных заносится дополнительная информация о страницах:

- Логин пользователя, посещавшего веб-страницу
- Имя компьютера, на котором находится пользователь

- Дата посещения
- ID веб-страницы

При реализации необходимо учитывать то, что количество пользователей может быть достаточно велико, поэтому необходимо обеспечить приемлемую масштабируемость агента сбора

4.1.3 Модуль классификации

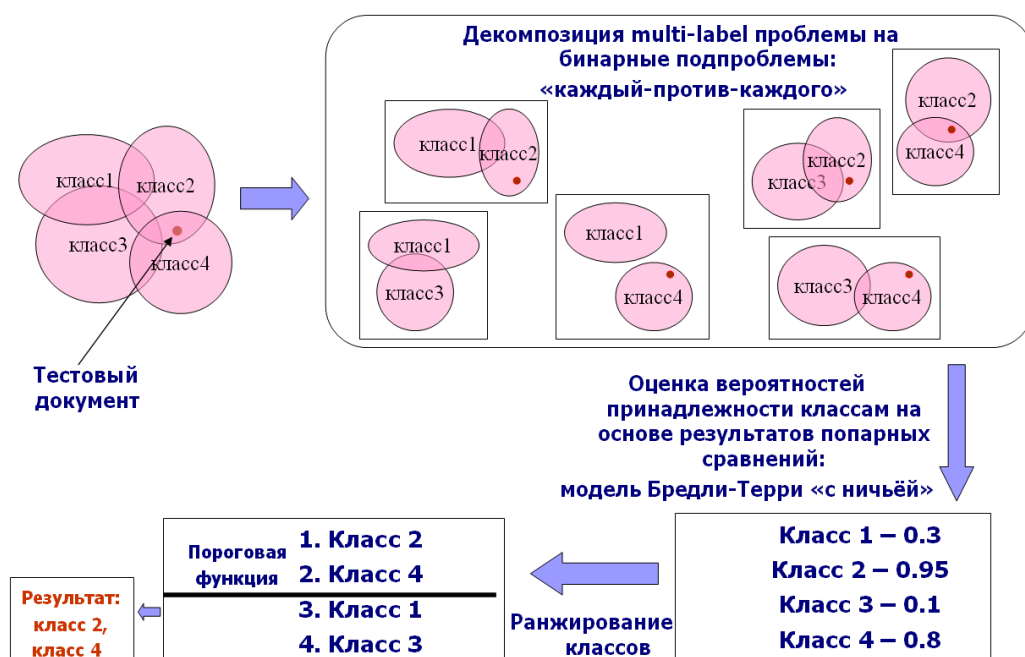


Рисунок 5 – Работа модуля классификации

В ходе обзора было принято решение использовать модуль классификации, реализованный в лаборатории Технологий Программирования, основанный на подходе попарных сравнений (декомпозиция типа «каждый-против-каждого»).

Предложенное решение включает новый алгоритм ранжирования, основанный на модифицированном для случая существенно пересекающихся классов, и новый алгоритм построения пороговой функции отсеечения нерелевантных классов.

Он предоставляет следующие сценарии работы:

- **Обучение.** Построение модели классификации на основе совокупности заранее рубрицированных гипертекстовых документов
- **Классификация.** Применение построенной модели к новому классифицируемому документу
- **Дообучение.** Модификация модели классификации на основе дообучения на новых документах с релевантными для них тематиками

- **Удаление темы.** Удаление тематики классификации из модели без необходимости последующего обучения "с нуля"

5 Описание практической части

В данном разделе будет рассмотрена архитектура разработанного программного средства, представлен основной сценарий работы с ним, а также пояснены детали реализованных механизмов. Также будут приведены некоторые характеристики функционирования разработанного средства.

5.1 Общая архитектура разработанного средства

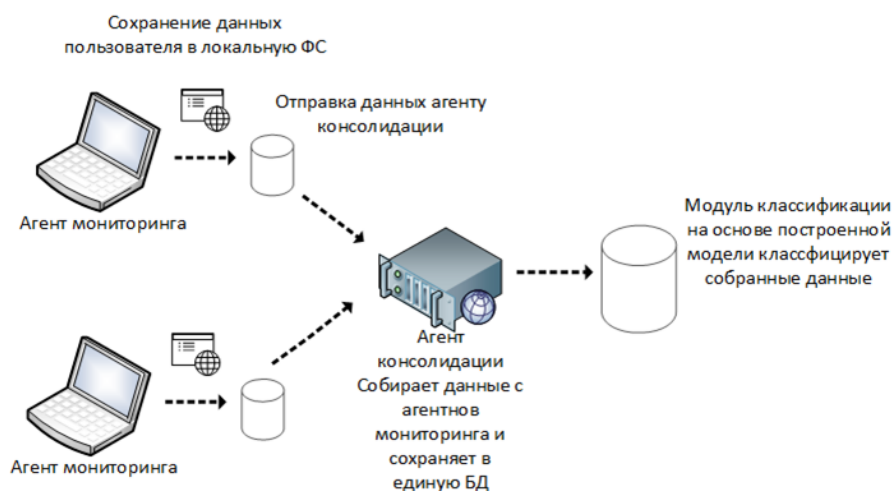


Рисунок 6 – Архитектура прототипа

5.1.1 Мониторинг пользователей

Задача сбора просмотренных пользователем веб-страниц осуществляется с помощью расширения для браузера IE.

При открытии окна браузера расширение создает подключение к существующей базе данных, реализованных с помощью SQLite. Если нет базы данных нет, то создается новая, имеющая следующие поля:

- Имя пользователя
- Имя компьютера
- URL просмотренной страницы

- Дата
- HTML код страницы

Каждый раз, когда пользователь загружает новую страницу, происходит событие, по которому html код просмотренной веб-страницы сохраняется в локальную базу данных.

Расширение написано с помощью ВНО (Browser Helper Object) на языке программирования C#, объем кода - 350 строк

5.1.2 Сбор данных

Каждый агент, расположенный на пользовательском компьютере подключается к базе данных, в которую были записаны данные расширением для браузера, и отправляет данные агенту сбора.

При настройке системы можно указать временной интервал, по которому будут отправляться данные. При отправке сохраняется время, когда была отправлена последняя просмотренная пользователем веб-страница.

При получении ответа от агента сбора, все записи, просмотренные до сохраненной временной метки удаляются, так как агент сбора успешно их сохранил, если же ответ не получен, то посылаются все записи, которые хранятся в базе.

Соединение агента мониторинга и агента сбора осуществляется с помощью TCP/IP сокетов. Каждый новый агент сбора обрабатывается в агенте консолидации асинхронно в отдельном потоке, что обеспечивает высокую скорость взаимодействия

Детали реализации модуля передачи данных агенту консолидации:

- Количество строк кода - 250
- Язык реализации - C#
- Доступ к локальной файловой системе без подтверждения пользователя осуществляется с помощью утилиты `icacls` [13], которая позволяет менять Integrity Levels [13] файлов. Так как Internet Explorer имеет право записи только в папки с Low Integrity уровнем, то для записи в нужное место необходимо создать папку и указать Integrity Level.

Агент сбора при получении посылает ответ агенту мониторинга, закрывает соединение и записывает полученные данные в единую базу данных, при этом к каждой записи добавляется ID.

Детали реализации агента консолидации:

- Агент сбора был написан на примере асинхронного сокет сервера [12]
- Язык написания - C#
- Количество строк кода - 600

5.1.3 Модуль многотемной классификации

С учетом требований, сформулированных в постановке задачи, для многотемной классификации был выбран модуль, разработанный в лаборатории Технологий Программирования.

Для интеграции системы сбора с модулем многотемной классификации был написан модуль, который позволяет:

- Обучать классификатор, подавая ему на вход тренировочный набор, содержащий веб-страницу и тематику, к которой данный документ принадлежит
- Выгружать веб-страницы из заданной базы данных и подавать их на вход классификатору. На выходе создается .csv файл следующего вида:

Таблица 2 – OutputeFile.csv

WebPageId	$topic_1$	$topic_2$...	$topic_N$
ID_1	$weight_{11}$	$weight_{12}$...	$weight_{1N}$
ID_2	$weight_{21}$	$weight_{22}$...	$weight_{2N}$
...
ID_N	$weight_{N1}$	$weight_{N2}$...	$weight_{NN}$

5.2 Экспериментальные исследования

В ходе обзора существующих решений было выдвинуто требование масштабируемости агента консолидации.

Так как разработка велась на языке программирования C# в среде разработки Visual Studio, то для проведения нагрузочного тестирования было решено воспользоваться встроенными средствами Visual Studio [15].

Ход тестирования

- Для нагрузочного тестирования были созданы Unit тесты [14], содержащие код компоненты агента мониторинга, который взаимодействует с агентом консолидации
- Visual Studio позволяет выбирать такие параметры сценария тестирования как: начальное количество клиентов, максимально количество клиентов, и скорость увеличения количества клиентов

- Далее, задавая параметры сценариев, было проведено исследование зависимости нагрузки ЦП и оперативной памяти от количества подключенных клиентов

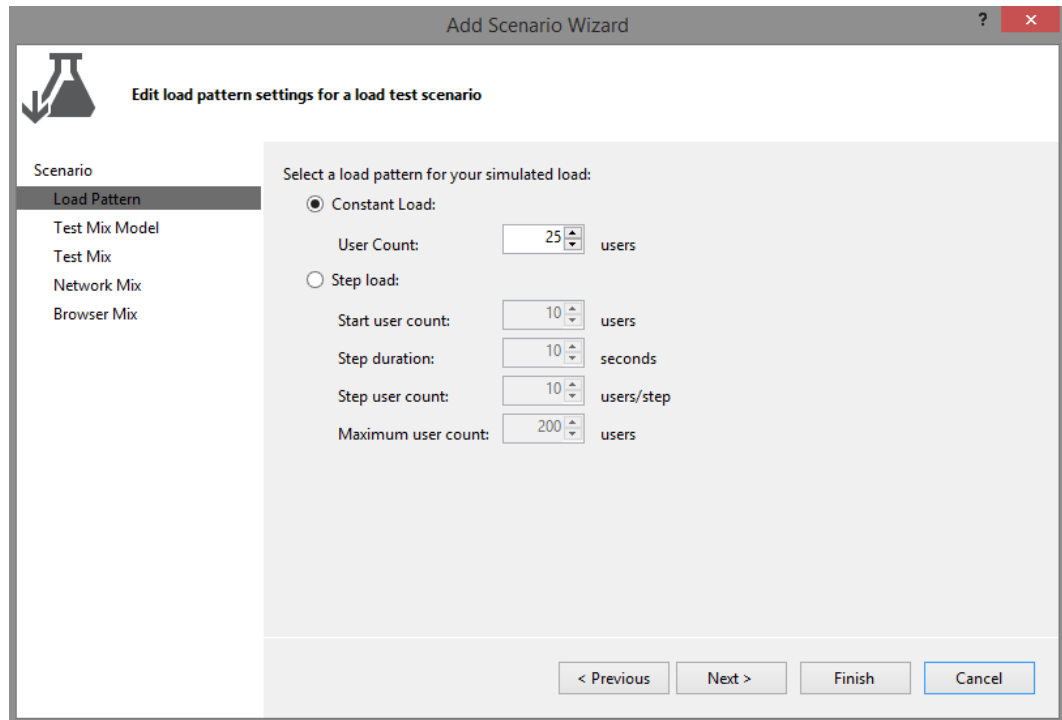


Рисунок 7 – Параметры сценария тестирования

5.2.1 Исследование зависимости нагрузки ЦП от количества подключенных клиентов

Сценарий тестирования:

- Начальное количество клиентов - 1. Каждый подключенный клиент с заданной частотой посылает агенту консолидации просмотренные пользователем веб-страниц
- С установленной скоростью увеличивается количество подключенных агентов мониторинга
- С помощью средств Visual Studio отрисовывается график, отражающий зависимость нагрузки ЦП от количества подключенных клиентов

Результаты тестирования проиллюстрированы ниже

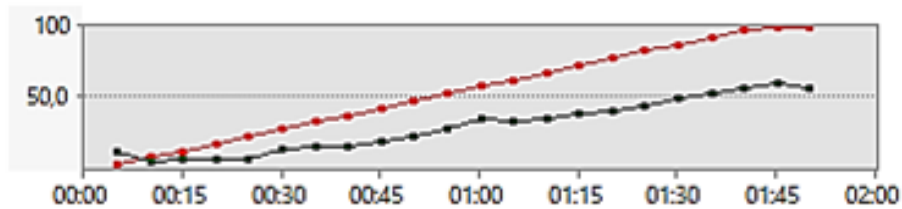


Рисунок 8 – Зависимость нагрузки ЦП от количества клиентов

5.2.2 Исследование использования оперативной памяти от количества клиентов

Сценарий тестирования аналогичен сценарию из предыдущего пункта. Результаты тестирования приведены ниже

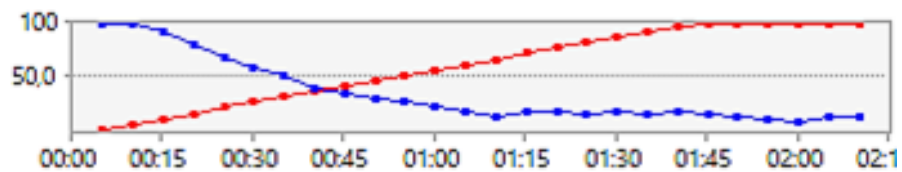


Рисунок 9 – Зависимость использования оперативной памяти от количества клиентов

5.3 Итоги тестирования

Зависимость нагрузки центрального процессора от количества подключенных клиентов линейна, что говорит о масштабируемости разработанного средства.

Увеличение использования оперативной памяти при увеличении количества подключенных клиентов достаточно линейно, что также говорит о масштабируемости разработанного средства

6 Заключение

В настоящей выпускной квалификационной работе были проведены исследования архитектур средств многотемной классификации веб-страниц. Был проведен анализ существующих алгоритмов многотемной классификации и выбран тот, который удовлетворяет требованиям поставленной задачи.

Была осуществлена программная реализация предложенной системы, состоящей из:

- Агента мониторинга
- Агента консолидации
- Модуля многотемной классификации

Были разработаны сценарии и проведено нагрузочное тестирование, которое показало хорошую масштабируемость.

Список литературы

1. Аналитический Центр InfoWatch, Безопасность информации в корпоративных информационных системах. Внутренние угрозы (<http://www.infowatch.ru/analytics/reports/4609>)
2. *Component overview (Content Classification 8.8.0)*
3. *Electronic discovery*. (wikipedia.org/wiki/Electronic_discovery)
4. *Предиктивное обучение* (<http://www.symantec.com/ru/ru/predictive-coding/>)
5. *Infowatch БКФ* (<http://www.infowatch.ru/technologies>)
6. *text122. Component overview (IBM Watson Content Analytics 3.5.0)* (http://www-01.ibm.com/support/knowledgecenter/SS5RWK_3.5.0/com.ibm.discovery.es.nav.doc/i1ysaov)
7. *Официальная документация IBM в Интернет* (http://www-01.ibm.com/support/knowledgecenter/SS5RWK_3.5.0/com.ibm.discovery.es.nav.doc/i1ysaov)
8. *Content Analytics. Официальный сайт OpenText в Интернет* (<http://www.opentext.com/what-we-do/products/discovery/content-analytics>)
9. *EMC Kazeon File Intelligence* (<http://www.emc.com/content-management/emc-kazeon-file-intelligence.htm>)
10. *Using the Taxonomy Proposer to discover new categories* (<http://www-01.ibm.com/support/knowledgecenter/>)
11. *Symantec Machine Learning* (<http://eval.symantec.com/mktginfo/enterprise/>)
12. *Асинхронный сокет сервер* ([https://msdn.microsoft.com/ru-ru/library/fx6588te\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/fx6588te(v=vs.110).aspx))
13. *Утилита icacls* (<https://msdn.microsoft.com/en-us/library/bb625965.aspx>)
14. *Create and run unit tests*. (<https://www.visualstudio.com/en-us/get-started/code/create-and-run-unit-tests-vs>)
15. *Walkthrough: Creating and Running a Load Test Containing Unit Tests* ([https://msdn.microsoft.com/en-us/library/vstudio/ff355993\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/vstudio/ff355993(v=vs.110).aspx))
16. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы
17. Schapire R. E., Singer Y. *BoosTexter* A boosting-based system for text categorization
18. Comite F. D., Gilleron R., Tommasi M. Learning multi-label alternating decision tree from texts and data Machine Learning and Data Mining in Pattern Recognition, MLDM 2003 Proceedings, Lecture Notes in Computer Science 2734. Berlin, 2003. pp. 35–49s
19. Zhang M.-L., Zhou Z.-H. *A k-nearest neighbor based algorithm for multi-label classification* Proceedings of the 1st IEEE International Conference on Granular Computing (GrC'05). Beijing, China, 2005. pp. 718-721
20. Глазкова В.В. Исследование и разработка методов построения программных средств классификации многотемных гипертекстовых документов