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:
Дмитрий Владимирович Царев

Аннотация

Целью данной работы является исследование и разработка экспериментального прототипа системы сбора и многотемной классификации веб-страниц, с которыми работали пользователи, находящиеся внутри одной сети. Отличие многотемной классификации от традиционной задачи классификации заключается в том, что классы могут пересекаться или даже быть вложенными.

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 был реализован экспериментальный прототип системы сбора и многотемной классификации, удовлетворяющий поставленным выше требованиям, проведено тестирование, показавшее масштабируемость и защищенность разработанного средства.

Содержание

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 предопределённого 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.

Отличие многотемной классификации от многоклассовой проиллюстрировано на рисунке 1.

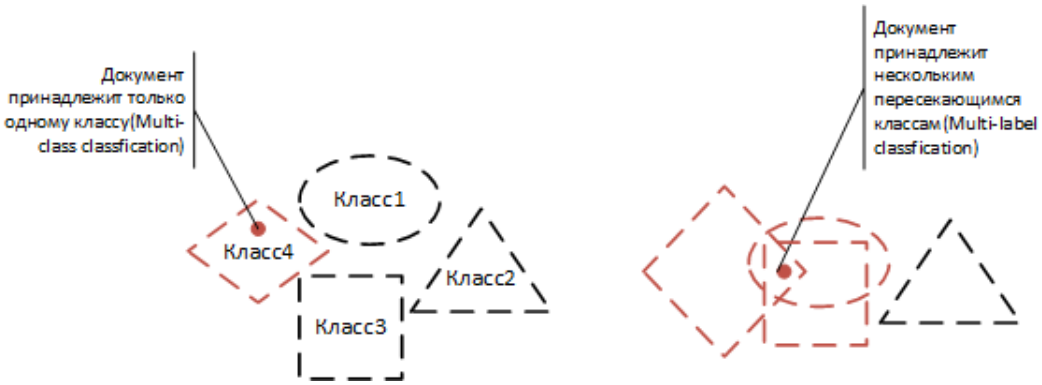

Рисунок 1 – Многотемная и многоклассовая классификация

1.2 Области применения классификации веб-данных

На сегодняшний день существует ряд актуальных 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 осуществляется применение политик безопасности.
- **Фильтрация контента для обнаружения информации, относящейся к определенному судебному 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.
- **Определение 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]. В связи с этим существует объективная 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 данных, связанных с определенным судебным делом (англ. eDiscovery). Средства eDiscovery обеспечивают процесс, с помощью которого организации находят, получают, сохраняют и анализируют документы, связанные с судебным 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.

1.3 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, 11].

Существуют три основных подхода к классификации данных в ECM системах:

- **Машинное обучение.** (IBM Content Classification [2], Symantec eDiscovery [4]). 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 неизвестных документов. При обнаружении конфиденциального документа производятся действия, предписанные политиками безопасности.
- **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 [12]).

Основные достоинства и недостатки методов представлены в таблице 1. Основное

Таблица 1 – Подходы к классификации

	Достоинства	Недостатки
Цифровые отпечатки - обнаружение в тексте цитат из документов-образцов.	Высокая точность детектирования статичных документов.	Чувствительность к текстовым изменениям.
Анализ шаблонов - анализ текстов на основе словарей и регулярных выражений.	Эффективность детектирования формализованных данных.	Не применим для детектирования неформализованных данных.
Машинное обучение – построение на основе обучающего набора статистической модели.	Работают напрямую с содержимым документов. Обучаемость. Возможность обобщения.	Требуется формирование обучающего набора. Возможность ложноотрицательных и ложноположительных срабатываний.

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.

1.4 Специфика задачи классификации текстовых веб-данных

Веб-страницы имеют следующие особенности:

- Содержимое веб-страниц постоянно меняется.
- Содержимое имеет многотемную природу, то есть каждая из страниц может одновременно принадлежать к нескольким тематикам.

На рисунке 2 иллюстрируется многотемность документа.

Рисунок 2 – Многотемная природа документа

1.5 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 меняются.

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, то есть документ может принадлежать только к одному классу из предопределенного набора. Большая часть веб-данных имеет многотемную природу, что подтверждает актуальность разработки прототипа системы сбора и многотемной классификации текстовых веб-данных пользователей. С учетом большого объема анализируемой информации к модулю классификации предъявляется требование масштабируемости.

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

2 Постановка задачи

Необходимо 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).
  - 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.
  - Возможность добавления и удаления тематик.

3 Обзор

Для реализации прототипа системы многотемной классификации веб-страниц, с которыми работал пользователь, необходимо:

- **Осуществить сбор** просмотренных пользователем веб-страниц. Модуль сбора не должен влиять на работу пользователя, поэтому следует разделить обязанности таким образом, чтобы вычислительная нагрузка на компоненту, которая взаимодействует с браузером, была минимальна. В связи с этим компонента модуля сбора будет состоять из:
  - Расширения для браузера, сохраняющего html-код просматриваемой пользователем веб-страницы в локальную базу данных, расположенную на пользовательском компьютере.
  - Модуля отправки данных.

  Для обеспечения защищенности разрабатываемого прототипа компонента сбора должна иметь возможность сохранять данные в локальную файловую систему компьютера.
- **Разработать модуль классификации**, который будет определять принадлежность документа предопределенным классам.

3.1 Расширение для браузера

Для решения задачи необходимо разработать расширение для браузера, имеющее возможность сохранить html код просматриваемой пользователем веб-страницы в локальную базу данных без подтверждения пользователем.

Средства написания расширения для браузера:

- **BHO** (Browser Helper Object) - DLL-модуль, разработанный как плагин для Internet Explorer для обеспечения дополнительной функциональности. В BHO API существует возможность получения доступа к DOM (Document Object Model[15]) 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 при этом BHO имеет возможность записи и чтения данных из файловой системы пользователя без его участия.
- **Kynext** позволяет перехватывать содержимое веб-страницы. Поддерживает браузеры IE, Firefox, Safari, 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. Не предоставляет возможность записи в локальную файловую систему без подтверждения пользователем.

– **WebMynd** поддерживает браузеры IE, Firefox, Safari, и Chrome. На данный момент доступна бета-версия и неизвестно, будет ли осуществляться дальнейшее развитие и поддержка продукта. Поддерживает JavaScript API. Не является бесплатным программным обеспечением. Не предоставляет возможность записи в локальную файловую систему без подтверждения пользователем.
– **Crossrider** позволяет быстро создавать кросс-браузерные расширения. Использует один API и поддерживает JavaScript и jQuery, что позволяет разработчику с базовыми знаниями JavaScript создавать и поддерживать свой код. Поддерживается возможность кодировать под IE, Firefox, Safari, Chrome. Бесплатен в использовании. Существует документация и демо-видео для некоторых задач. Доступ к файловой системе пользователя осуществляется только с его разрешения.

Обобщенные данные об обзоре средств расширения представлены в таблице 2.

Таблица 2 – Средства расширения для браузера

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

3.2 Методы многотемной классификации

В данном разделе будет проведен обзор методов многотемной классификации, основанных на традиционных подходах, а также метод, разработанный в лаборатории Технологий Программирования, модифицирующий традиционные подходы с учетом их недостатков. Можно выделить три основных подхода к многотемной классификации [30]:

- «Оптимизационный» подход [21, 22, 31].
- Подход на основе декомпозиции в набор независимых бинарных проблем.
- Подход на основе ранжирования [32, 33].

3.2.1 Методы, основанные на «оптимизационном» подходе

К числу алгоритмов, использующих данный подход, относятся алгоритмы, которые решают оптимизационную задачу. Рассмотрим некоторые из данных алгоритмов:

- Основанные на AdaBoost алгоритме (AdaBoost.MH [21],ADTBoost.MH [22]) - минимизируется функция Hamming Loss, где HammingLoss - критерий, который оценивает, сколько раз пара «пример-метка» классифицируется неправильно.
- Multi-Label-kNN - [23] максимизируются апостериорные вероятности принадлежности классам.

Методы, основанные на алгоритме AdaBoost. К числу данных методов можно отнести AdaBoost.MH [21] и ADTBoost.MH [22]. AdaBoost вызывает слабые классификаторы в цикле $t = 1, \ldots, T$. После каждого вызова обновляется распределение весов D_t, которые отвечают важности каждого из объектов обучающего множества для классификации. На каждой итерации веса каждого неверно классифицированного объекта возрастают, таким образом, новый комитет классификаторов «фокусирует своё внимание» на этих объектах.

Алгоритм работы AdaBoost [10]:

Дано: $(x_1, y_1), \ldots, (x_m, y_m) x_i \in X$, $y_i \in Y = \{-1, +1\}$

Изначально всем классам присваивается одинаковый вес $D_t(i) = \frac{1}{m}, i = 1, \ldots, m$.

$for\ t = 1, \ldots, T$

- Находится классификатор, минимизирующий взвешенную ошибку классификации $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j$, где $\epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$.
- Если величина $\epsilon_t \geqslant 0.5$, то алгоритм останавливается.

– Выбирается $\alpha_t \in \mathbf{R}$, обычно $\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$, где ϵ_t взвешенная ошибка классификатора h_t.
– $D_{t+1}(i) = \frac{D_t(i)\, e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$, где Z_t является нормализующим параметром (выбранным так, чтобы D_{t+1} являлось распределением вероятностей, то есть $\sum_{i=1}^{m} D_{t+1}(i) = 1$).

Итоговый классификатор имеет вид: $H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Таким образом, на каждом шаге выбирается оптимальный классификатор, и веса тех объектов x_i, которые он предсказал правильно, уменьшаются, а тех, которые он предсказал неверно - увеличиваются, чтобы на следующем шаге был выбран классификатор, который лучше распознает объекты, неверно распознанные предыдущим.

С точки зрения сформулированных требований метод AdaBoost имеет следующие недостатки:

– Алгоритм не имеет возможности пошагового дообучения, поскольку принцип работы метода рассчитан на то, что обучающий набор задан заранее целиком.
– Для достижения качества классификации, удовлетворяющего потребностям актуальных прикладных задач, этот метод требует большого числа шагов на этапе обучения, а следовательно, обучение выполняется достаточно долго.

В случае многотемного AdaBoost заключительная гипотеза для ранжирования классов получается как взвешенное голосование слабых гипотез (гипотез, полученных при решении задачи многоклассовой классификации), причём вес слабой гипотезы вычисляется с учётом качества классификации этой гипотезой примеров из обучающего набора. Заключительная гипотеза multi-label классификации получается на основе гипотезы ранжирования путём отсечения по нулевому порогу релевантности.

Multi-Label-kNN. Данный метод представляет собой подход многотемного обучения, основанный на методе k ближайших соседей. Основным принципом метода ближайших соседей является то, что объект присваивается тому классу, который является наиболее распространённым среди соседей данного элемента. В многотемном варианте kNN расстояние выражается через косинус угла между векторами признаков элементов: $\rho = 1 - \cos(\vec{a_1}, \vec{a_2})$.

Пусть H_1^l - событие, что тестовый пример $\vec{a}$ имеет метку l, H_0^l - иначе, $E_k^l (j \in \{0, \ldots, k\})$ - событие, что j соседей имеют метку l, $\vec{C_{\vec{a}}}$ - вектор подсчета членства. Вектор категорий $\vec{y_{\vec{a}}}$ определяется, используя принцип максимизации апостериорных вероятностей:

$$\vec{y_{\vec{a}}} = \arg\max_{b \in 0,1} P(H_b^l | E^l_{\vec{C_{\vec{a}}}}), l \in Y$$

где Y - набор классов.

Данный метод имеет следующие недостатки:

– Большие вычислительные затраты.

– Большой объём памяти, необходимый для хранения и работы сразу со всем набором обучающих данных.

3.2.2 Методы, основанные на декомпозиции в набор независимых бинарных проблем

Работу алгоритмов можно описать следующей последовательностью действий:

– Декомпозиция исходной проблемы в набор независимых бинарных проблем («каждый-против-остальных»). Для каждого из N классов создается одна бинарная проблема.
– В бинарной проблеме для класса l все обучающие примеры, помеченные этим классом, считаются положительными, а все остальные обучающие примеры считаются отрицательными.
– Далее к каждой из N полученных бинарных проблем применяется бинарный алгоритм обучения (например, двуклассовая SVM [13]).
– В результате получаются N бинарных классификаторов (гипотез), каждый из которых независимо от остальных оценивает релевантность каждого из классов.
– Для каждого из N классов будет построена решающая функция бинарной классификации.
– Знак каждой решающей функции дает предсказание релевантности темы l для данного документа.

Достоинства и недостатки данного подхода:

– Так как решение о принадлежности документа принимается независимо от остальных классов, то имеется возможность добавления и удаления классов без необходимости обучения «с нуля».
– С использованием метода опорных векторов достигается высокая точность классификации, отсутствует возможность пошагового обучения.
– При применении алгоритма персептрона для бинарной классификации обеспечивается возможность дообучения, но в результате метод, как правило, имеет низкую точность.
– Из-за того, что количество бинарных подзадач равно числу классов, подход имеет высокую вычислительную сложность.
– Строятся независимые классификаторы, которые не учитывают корреляции между классами.

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 (MMP[26, 27], k-NN [28]).
- Второй этап заключается в построении функции многотемной классификации, отделяющей релевантные классы от нерелевантных.

Метод Multiclass-Multilabel Perceptron. Алгоритм ранжирования MMP поддерживает набор из n прототипов тем - $\vec{w_1}, \ldots, \vec{w_n}$. При получении документа алгоритм модифицирует предложенную гипотезу, то есть изменяет набор прототипов (весов) $\vec{w_1}, \ldots, \vec{w_n}$. Это осуществляется для всех документов из обучающей выборки. Окончательной гипотезой является набор прототипов после одного прохода.

Пусть $\vec{x}$ - документ, $\vec{w_1}, \ldots, \vec{w_n}$ - набор прототипов. Ранжирование осуществляется на основе скалярного произведения, т.е. тема r ранжируется выше, чем тема s, если $(w_r, x) > (w_s, x)$.

Пусть $y \in Y$ - набор релевантных тем, где Y - множество тем. Совершенным называется такое ранжирование, при котором для $\forall s \in y$ и для $\forall k (k \in Y \setminus y)$ s ранжируется выше, чем k, т.е. $(s, y) > (k, y)$.

Качество совершенного ранжирования определяется размером промежутка между самой низкой оценкой среди релевантных тем и самой высокой оценкой среди нерелевантных тем: $\min \vec{w_y} - \max \vec{w_k}$.

Алгоритм ранжирования MMP получает и анализирует обучающие данные последовательно, пример за примером. Данный алгоритм удобен при работе с очень большими наборами обучающих данных, так как он эффективен по памяти и имеет возможность дообучения. Однако данный алгоритм не поддерживает возможность добавления тем без необходимости заново обучать модель ранжирования.

3.2.4 Метод классификации многотемных документов на основе подхода попарных сравнений

В лаборатории Технологий Программирования было предложенно новое решение, которое включает:

– Новый алгоритм ранжирования, основанный на модифицированном для случая существенно пересекающихся классов методе попарных сравнений с помощью набора бинарных классификаторов и вычислении степеней принадлежности классам с использованием обобщенной модели Брэдли-Терри с «ничьей» [29].
– Новый алгоритм построения пороговой функции отсечения нерелевантных классов, строящий пороговую функцию не в исходном пространстве характеристик (как это делает большинство традиционных алгоритмов), а в пространстве релевантностей классов, что позволяет упростить вид пороговой функции, значительно сократить вычисления и в большинстве случаев увеличить точность.

3.2.5 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. Обобщенные данные о традиционных подходах к многоклассовой классификации представлены в таблице 3.

3.3 Выводы из обзора

К компоненте сбора предъявлялись следующие требования:

– **Производительность.** Работа компоненты не должна сказываться на деятельности пользователя.
– **Защищенность.** Пользователь не должен иметь доступа к собранным данным.

Из рассмотренных средств написания расширения для браузера обоим требованиям удовлетворяет BHO. Первое требование удовлетворяется возможностью записи в локальную файловую систему без подтверждения и разделением компоненты сбора на два модуля. Второе требования удовлетворяется установкой прав доступа на папку, в которую сохраняются собранные данные.

Обзор существующих методов классификации многотемных документов показал, что обозначенным в постановке задачи требованиям удовлетворяют только

Таблица 3 – Традиционные подходы многоклассовой классификации

Подход	Недостатки
Оптимизационный подход	Нет возможности дообучения. Большие затраты памяти. Нет возможности удаления и добавления классов.
Подход на основе декомпозиции в набор независимых бинарных проблем.	Нет учета корреляции между классами. Высокая вычислительная сложность.
Подход на основе ранжирования	Высокая вычислительная сложность. Нет возможности дообучения.
Модифицированный подход, разработанный в лаборатории Технологий Программирования	Разработанный метод модифицирован с учетом недостатков существующих подходов, поэтому, с точки зрения сформулированных требований, недостатков нет

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 поставленный требованиям.

4 Исследование и построение решения

Проведенный анализ систем ECM и DLP в контексте задач классификации показывает, что каждый из пользователей является источником анализируемой информации.

Рассматриваемая задача не может быть решена с помощью так называемой монолитной системы [19], то есть системы, в которой функционально различные аспекты связаны между собой, а не являются архитектурно независимыми компонентами.

Для задач, которые сложно или невозможно решить с помошью монолитной системы, используются мультиагентные системы [25, 20].

В мультиагентной системе агенты имеют несколько важных характеристик:

- **Автономность.** Агенты, хотя бы частично, независимы.
- **Ограниченность представления.** Ни у одного из агентов нет представления о всей системе, или система слишком сложна, чтобы знание о ней имело практическое применение для агента.
- **Децентрализация.** Нет агентов, управляющих всей системой.

Обычно в мультиагентных системах исследуются программные агенты [24].

В рассматриваемых корпоративных системах агентами являлись компоненты системы сбора [6]. С учетом обзора ECM и DLP и методов построения систем, содержащих архитектурно независимые компоненты, разрабатываемый прототип будет иметь в своей основе мультиагентный подход [20].

Разрабатываемый прототип состоит из:

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

4.1 Агент мониторинга

Агент мониторинга состоит из нескольких компонент:

- Расширение для браузера.
- Модуль передачи данных агенту консолидации.

Разрабатываемый агент должен выполнять следующие функции:

- **Сохранение просмотренных пользователем веб-страниц.** Каждая из посещенных страниц должна подаваться на вход модулю классификации, поэтому на браузер устанавливается расширение, которое позволяет сохранять html-код просматриваемой веб-страницы. Обзор существующих технологий написания расширения для браузера показал, что для доступа расширения к файловой системе необходимо подтверждение пользователя. С таким ограничением пользователю придется каждый раз при загрузке страницы подтверждать сохранение, что может сильно сказаться на производительности пользователя. Это ограничение позволяет обойти только расширение для IE, написанное с помощью Browser Helper Object. Так как предложенное решение позволяет сохранять данные в любое место на диске, то требование безопасности можно обеспечить установкой прав доступа на папку, в которой сохраняются просмотренные веб-страницы.
- **Обеспечение целостности данных при сбое.** При функционировании системы может возникнуть ситуация(сбой сети), при которой агент консолидации будет недоступен. Для предотвращения утери информации перед отправкой все страницы загружаются в локальную файловую систему пользователя и удаляются только в случае успешной отправки их агенту консолидации.
- **Контроль нагрузки на сеть.** Для уменьшения нагрузки на сеть данные перед отправкой сжимаются. Также для обеспечения распределения нагрузки на сеть предложено организовывать передачу собранных агентом данных в пакетном режиме по одной или нескольким из следующих стратегий:
  - **Фиксированные объемы данных.** Агент накапливает определенный объем информации и затем передает их на сервер консолидации.
  - **Через равные промежутки времени.** Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема.
- **Отправка данных агенту консолидации.** Отправка данных осуществляется вышеописанными стратегиями, при получении ответа от агента консолидации все успешно отправленные данные удаляются для предотвращения непрерывного увеличения объема локальной базы данных. Передаваемые данные содержат не только html-код просмотренных пользователем веб-страниц, но и дополнительную информацию:
  - Логин пользователя, посещавшего веб-страницу.
  - Имя компьютера, на котором находится пользователь.
  - Дата посещения.
  - URL страницы.

Схема работы агента мониторинга представлена на рисунке 3.

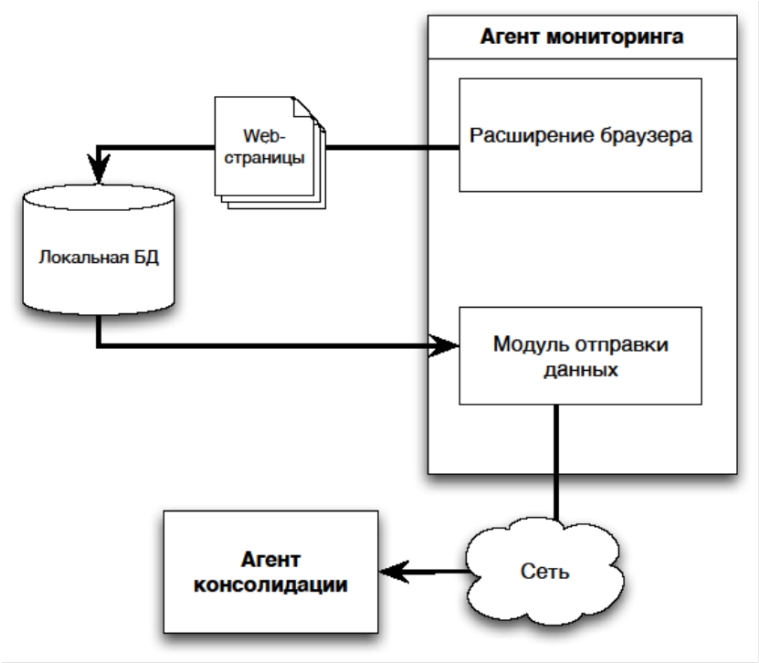

Рисунок 3 – Архитектура агента мониторинга

4.2 Агент консолидации

Агент мониторинга представляет следующие сценарии функционирования:

- **Получение данных от агентов мониторинга.** Каждый агент мониторинга либо по истечении определенного времени, либо при достижении определенного объема накопленных данных посылает их агенту консолидации. Так как количество клиентов достаточно велико, то запросы необходимо обрабатывать асинхронно и для каждого агента выделять отдельный поток.
- **Сохранение полученных данных в единую базу.** Для удобства классификации данные должны храниться в единой базе. Каждый документ должен иметь уникальный идентификатор, поэтому при записи в базу каждому документу присваивается автоинкрементируемый ID и итоговая таблица в базе данных имеет следующие поля:
  - Логин пользователя, посещавшего веб-страницу.
  - Имя компьютера, на котором находится пользователь.
  - Дата посещения.
  - URL страницы.
  - Html-код веб-страницы.
  - ID веб-страницы.

Архитектура агента консолидации представлена на рисунке 4.

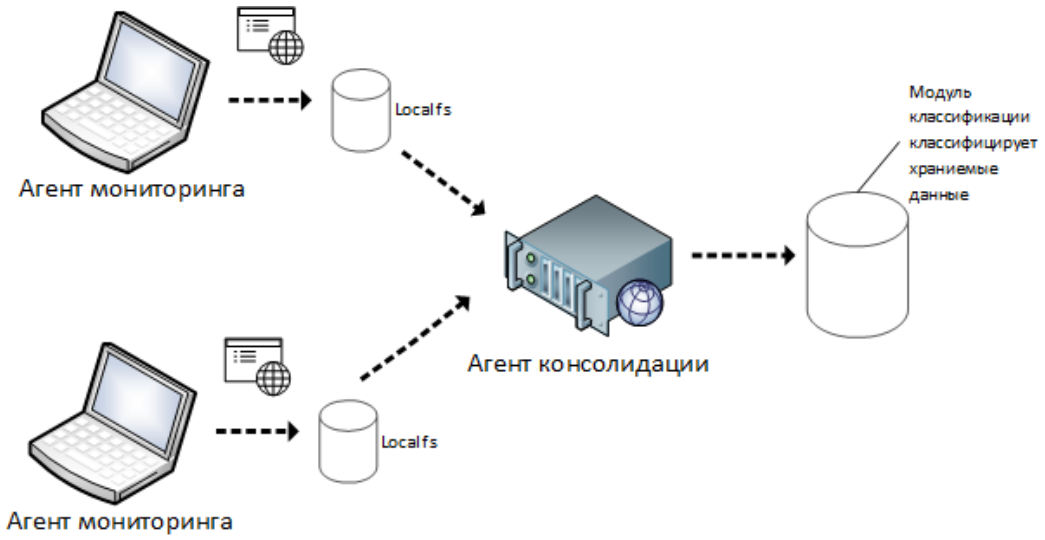

Рисунок 4 – Архитектура агента консолидации

4.3 Модуль 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»).

новый алгоритм ранжирования, основанный на модифицированном для случая существенно пересекающихся классов методе попарных сравнений с помощью набора бинарных классификаторов и новый алгоритм построения пороговой функции отсечения нерелевантных классов. Модуль многотемной классификации состоит из:

- компоненты лексического анализа (парсер) – осуществляет разбор, выделение признаков и преобразование веб-страниц во внутреннее представление;
- компоненты вычисления меры сходства – определяет значения близости между документами (значения функции ядра) на основе выданного парсером представления и осуществляет кэширование этих значений;
- классификатора, который строит дообучаемую модель классификации и на её основе осуществляет классификацию многотемных веб-страниц.

Работа модуля классификации представлена на рисунке 5.
Он предоставляет следующие сценарии работы:

- **Обучение.** При обучении на вход классификатору подается набор документов, к каждому из которых приписывается набор релевантных для него тем. Результат работы классификатора - модель классификации, с помощью которой можно определить набор релевантных тем для произвольного документа со схожей тематикой. Задача многотемной классификации решается с помощью метода, разработанного в лаборатории Технологий Про-

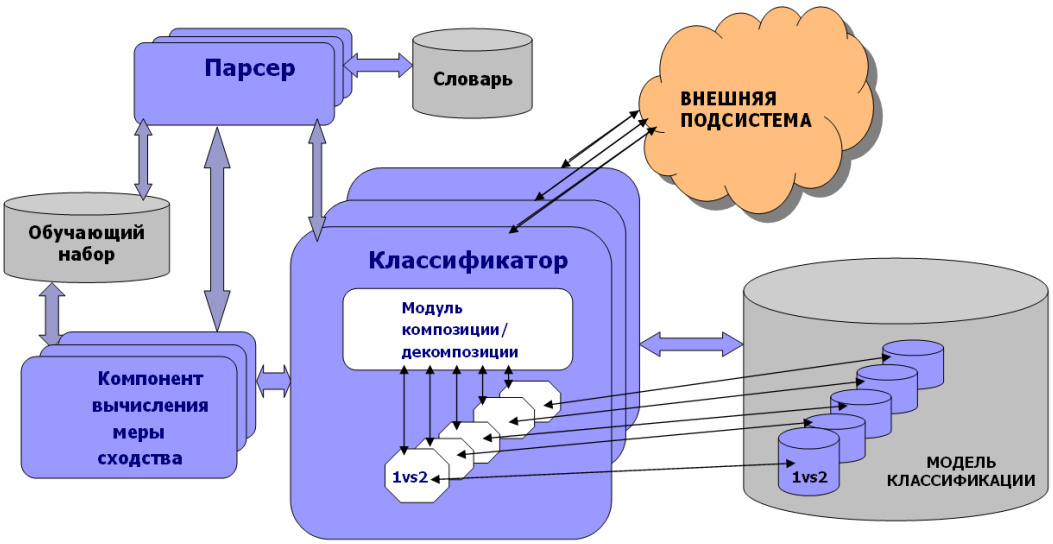

Рисунок 5 – Архитектура модуля классификации

граммирования на основе подхода попарных сравнений. Результаты обучения сохраняются в модели классификации, которая является частью контекста классификации. Обучение бинарных классификаторов осуществляется методом опорных векторов. В режиме обучения также определяются коэффициенты пороговой функции, которые сохраняются в модели.

- **Классификация.** Классификацию можно разделить на следующие этапы:
  - **Применение модели к новому классифицируемому документу.** Каждая из бинарных моделей, полученная на этапе обучения применяется к классифицируемому документу.
  - **Объединение результатов бинарных классификаторов.** Результаты, предсказанные всеми бинарными классификаторами, объединяются вместе, чтобы оценить результирующие степени принадлежности классам. В результате этого получаем значения релевантности для всех классов.
  - **Нахождение порогового значения.** Пороговое значение определяется на этапе обучения. С помощью него отсекаются нерелевантные для документа классы.
- **Дообучение.** При дообучении происходит модификация существующей модели классификации на основе новой обучающей выборки. Так же, как и при обучении, осуществляется разделение задачи на бинарные подзадачи. Для дообучения бинарных классификаторов используется алгоритм персептрона, инициализированный коэффициентами, вычисленными на основе обученных бинарных моделей классификации. При дообучении используются только новые данные, и во время дообучения документы классифицируются, основываясь на старой модели, при завершении дообучения старая модель заменяется новой.

– **Удаление темы.** За счёт применения подхода на основе декомпозиции в набор бинарных проблем удаление темы не требует обучения «с нуля» модели классификации. При удалении тематики классификации из модели классификации удаляется набор бинарных моделей, в которых участвовала эта тема. Остальные бинарные модели при этом остаются неизменными. Модель пороговой функции модифицируется с учётом обучения пороговой функции без удаляемой тематики. После обновления в текущем контексте модели классификации новые документы классифицируются уже без учёта удалённой тематики.

Общая архитектура модуля представлена на рисунке 6.

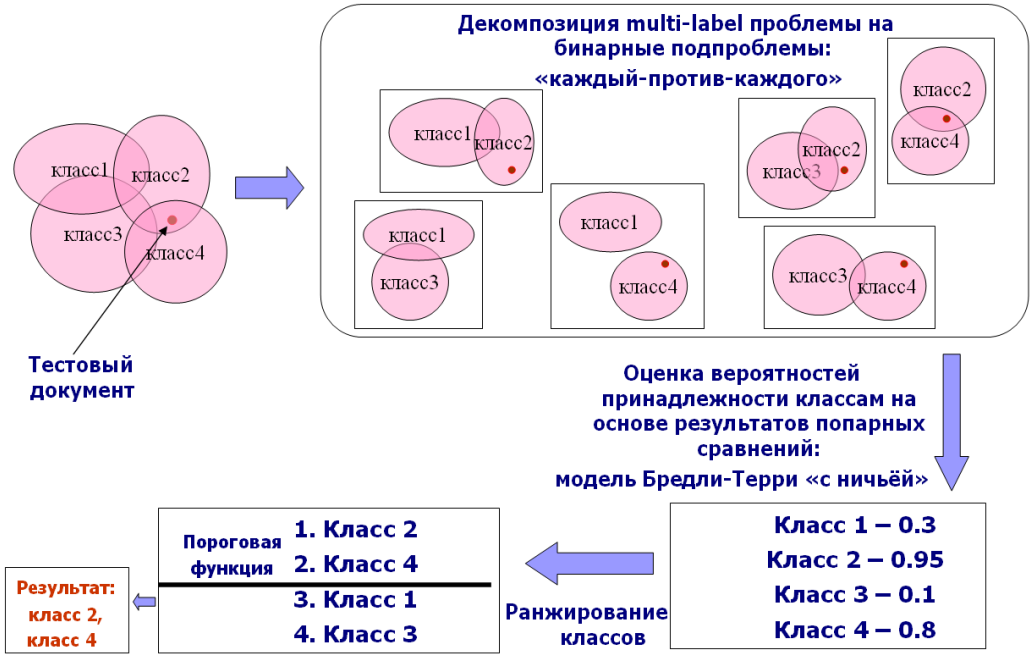

Рисунок 6 – Работа модуля классификации

4.4 Выводы

Предложенное решение состоит из:

– Агента мониторинга.
– Агента консолидации.
– Модуля классификации.

В ходе исследования было принято решение о декомпозиции компоненты мониторинга на два модуля:

– Расширение для браузера.

– Модуль отправки данных.

Такое решение позволило обеспечить производительность компоненты сбора. Требование защищенности осуществляется с помощью установки прав на директорию, в которую сохраняются данные. Для предотвращения утери информации при сбое соединения данные изначально сохраняются на локальную файловую систему пользователя. Нагрузка на сеть контролируется с помощью формирования расписания отправки данных агенту консолидации и сжатия данных перед отправкой.

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

При разработке прототипа системы использовался мультиагентный подход, прототип состоит из:

- Агента мониторинга.
- Агента консолидации.
- Модуля многотемной классификации.

Архитектура реализованного прототипа представлена на рисунке 7.

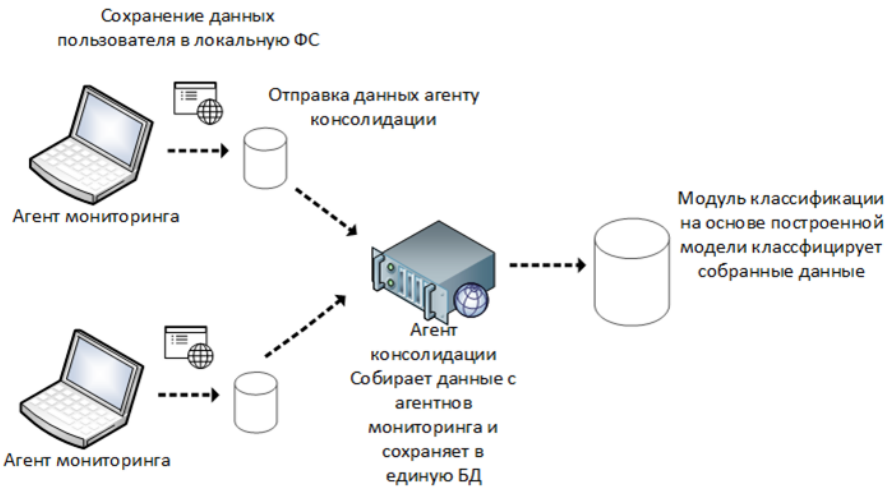

Рисунок 7 – Архитектура прототипа

5.1.1 Агент мониторинга

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

При открытии окна браузера расширение создает подключение к существующей базе данных, реализованной с помощью SQLite. Если базы данных нет, то создается новая, имеющая таблицу со следующими полями:

- 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 BHO (Browser Helper Object) на языке программирования C#, объем кода - 350 строк.

5.1.2 Агент сбора

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 [16], которая позволяет менять Integrity Levels [16] файлов. Так как Internet Exploler 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 [14].
- Язык написания - C#.
- Количество строк кода - 600.

5.1.3 Особенности программной реализации модуля классификации

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

Можно выделить следующие особенности реализации [30]:

- Для обеспечения скорости классификации, удовлетворяющей интерактивному режиму работы пользователей, использовались следующие средства. Для программной реализации модуля был выбран язык C++, который является одним из наиболее эффективных языков программирования для реализации решения алгоритмически сложных задач. Для повышения скорости классификации модель классификации для текущего контекста хранится в оперативной памяти.
- Эффективная обработка параллельных запросов на классификацию достигнута при помощи следующих средств. Разработанная архитектура модуля обладает свойством масштабируемости, что позволяет эффективно распределять вычислительную нагрузку по разным вычислительным узлам. Разработанный модуль позволяет создать набор идентичных классификаторов (на разных вычислительных узлах), обрабатывающих поток поступающих запросов, и с помощью средств синхронизации поддерживать модели классификации у всех классификаторов в актуальном состоянии.
- Разработанный модуль имеет «обёртку» на языке Python, которая обеспечивает интерфейс для встраивания в различные приложения и удобные средства тестирования отдельных компонентов модуля. На языке Python реализован XML-RPC интерфейс; система управления контекстами и модули, управляющие последовательностью обработки документов.
- Возможность осуществления классификации новых документов параллельно с дообучением (добавлением темы) достигнута при помощи использования средств синхронизации для оперативной подмены текущей модели классификации на дообученную.

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

- Производить парсинг html-кода страницы.
- Обучать классификатор, подавая ему на вход тренировочный набор, содержащий веб-страницы и соответствующие им тематики.
- Выгружать веб-страницы из заданной базы данных и подавать их на вход классификатору. На выходе создается .csv файл с весами тематик для каждого документа.

Модуль написан на языке программирования Python, объем - 200 строк.

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 [18].

Ход тестирования:

- Для нагрузочного тестирования были созданы Unit тесты [17], содержащие код компоненты агента мониторинга, который взаимодействует с агентом консолидации.
- Visual Studio позволяет выбирать такие параметры сценария тестирования, как: начальное количество клиентов, максимальное количество клиентов, скорость увеличения количества клиентов, пример интерфейса показан на рисунке 8.
- Далее было проведено исследование зависимости нагрузки ЦП и свободной оперативной памяти от количества подключенных клиентов.

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-страницы.

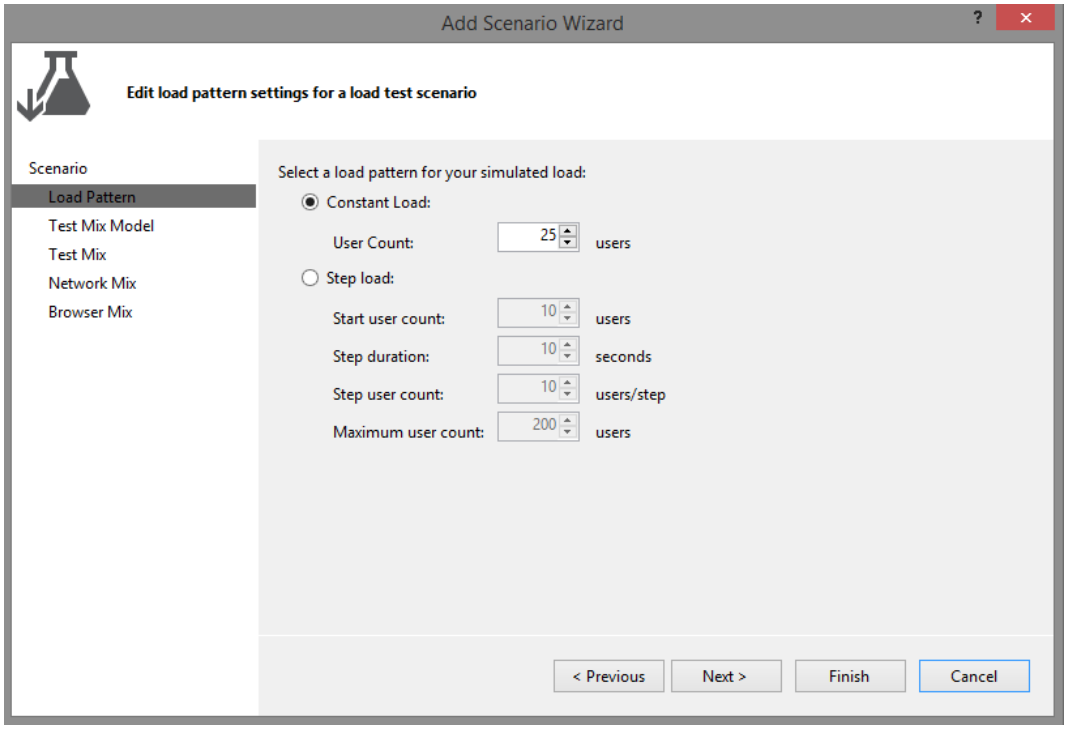

Рисунок 8 – Параметры сценария тестирования

– С установленной скоростью увеличивается количество подключенных агентов мониторинга.
– С помощью средств Visual Studio строится график, отражающий зависимость нагрузки ЦП от количества подключенных клиентов.

Результаты тестирования проиллюстрированы на рисунке 9. Красным цветом отмечен рост клиентов, зеленым - загрузка ЦП.

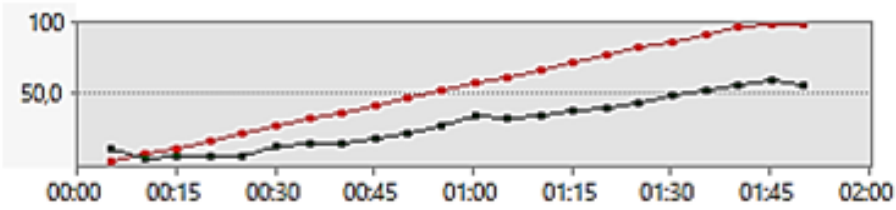

Рисунок 9 – Зависимость нагрузки ЦП от количества клиентов

5.2.2 Исследование зависимости количества свободной оперативной памяти от количества клиентов

Сценарий тестирования аналогичен сценарию из пункта 5.2.1. Результаты тестирования приведены на рисунке 10. Красным цветом отмечен рост клиентов, синим - количество свободной оперативной памяти.

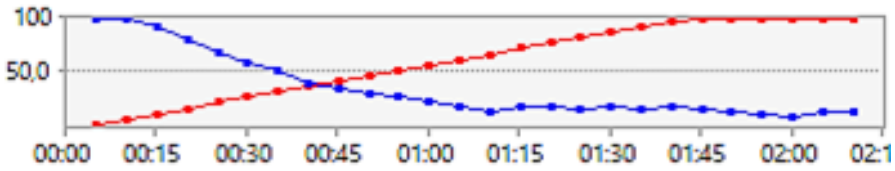

Рисунок 10 – Зависимость использования оперативной памяти от количества клиентов

5.3 Результаты 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 линейно, что также говорит о масштабируемости разработанного средства.

6 Заключение

В настоящей выпускной квалификационной работе были проведены исследования методов многотемной классификации веб-страниц. В ходе работ был проведен обзор современных индустриальных систем, которые решают ряд задач, требующих сбора и классификации контента, с которыми работали пользователи, в частности, веб-страниц. Были рассмотрены различные подходы к классификации текстовых данных: метод шаблонов, метод цифровых отпечатков, методы на основе машинного обучения. Было принято решение, что требованиям поставленной задачи удовлетворяют только методы, основанные на машинном обучении, так как только они могут адаптироваться к тому, что содержание веб-страниц постоянно меняется и имеет многотемную природу.

Был проведен анализ основных подходов к решению задачи многотемной классификации: «оптимизационный» подход, подход на основе декомпозиции в набор независимых бинарных проблем, подход на основе ранжирования, выявлены недостатки, не позволяющие применять их для решения поставленной задачи. Также был рассмотрен и выбран метод многотемной классификации на основе подхода попарных сравнений, учитывающий недостатки традиционных подходов.

Была осуществлена программная реализация прототипа системы сбора и многотемной классификации веб-страниц, состоящей из:

- Агента мониторинга.
- Агента консолидации.
- Модуля многотемной классификации.

С помощью средств модульного и нагрузочного тестирования, предоставленного средой Miscrosoft Visual Studio были реализованы сценарии нагрузочного тестирования и измерены следующие параметры функционирования разработанного прототипа системы:

- Зависимость нагрузки центрального процессора от количества подключенных клиентов.
- Зависимость количества свободной оперативной памяти от количества подключенных клиентов.

Результаты тестирования показали линейный рост нагрузки центрального процесса при увеличении количества подключенных клиентов, что говорит о масштабируемости разработанного прототипа.

Также результаты тестирования показали линейное убывание свободной оперативной памяти при увеличении количества клиентов, что тоже говорит о масштабируемости.

Реализация модуля сбора данных с помощью Browser Helper Object позволила сохранять данные о просмотренных пользователем веб-страницах в локальную

файловую систему в директорию с ограниченным правом доступа, что обеспечивает защищенность (обычный пользователь не имеет доступа к собранным данным) разработанного прототипа.

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. *text*122. Component overview (IBM Watson Content Analytics 3.5.0) (http://www-01.ibm.com/support/knowledgecenter/SS5RWK_3.5.0/com.ibm.discovery.es.na
7. *Официальная документация IBM в Интернет* (http://www-01.ibm.com/support/knowledgecenter/SS5RWK_3.5.0/com.ibm.discovery.es.nav.doc/iiysaov
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. *Работа AdaBoost алгоритма* (https://ru.wikipedia.org/wiki/AdaBoost)
11. *Using the Taxonomy Proposer to discover new categories* (http://www-01.ibm.com/support/knowledgecenter/
12. *Symantec Machine Learning* (http://eval.symantec.com/mktginfo/enterprise/)
13. *Boutell M. R., Luo J., Shen X., Brown C.M. Learning multi-label scene classification* Pattern Recognition. 2004. №37. pp. 1757-1771.
14. *Асинхронный сокет сервер* (https://msdn.microsoft.com/ru-ru/library/fx6588te(v=vs.110).aspx)
15. *Document Object Model* https://ru.wikipedia.org/wiki/Document_Object_Model
16. *Утилита icacls* (https://msdn.microsoft.com/en-us/library/bb625965.aspx)
17. *Create and run unit tests.* (https://www.visualstudio.com/en-us/get-started/code/create-and-run-unit-tests-vs)
18. *Walkthrough: Creating and Running a Load Test Containing Unit Tests* (https://msdn.microsoft.com/en-us/library/vstudio/ff355993(v=vs.110).aspx)
19. *Monolithic system* (https://en.wikipedia.org/wiki/Monolithic_system)
20. *Таненбаум Э., Ван Стеен М.* Распределенные системы. Принципы и парадигмы
21. *Schapire R. E., Singer Y. BoosTexter* A boosting-based system for text categorization
22. 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

23. *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
24. *Software agent* (https://en.wikipedia.org/wiki/Software_agent)
25. *Мультиагентная система* (https://en.wikipedia.org/wiki/Multi-agent_system)
26. *C. Crammer, Y. Singer. A family of additive online algorithms for category ranking* Machine Learning Research. №3. 2003. pp. 1025–1058
27. *Crammer C., Singer Y. A new family of online algorithms for category ranking* Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. Tampere, Finland, 2002. pp. 151 – 158
28. *Minh Duc Cao, Xiaoying Gao. Combining Content and Citation for Scientific Document Classification* AI2005, LNAI 3809, 2005. pp. 143-152
29. P.V. Rao and L.L. Kupper. Ties in paired-comparison experiments A generalization of the Bradley–Terry model, Amer. Statist. Assoc, 62, 1967. pp. 194–204.
30. *Глазкова В.В.* Исследование и разработка методов построения программных средств классификации многотемных гипертекстовых документов
31. *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
32. *C. Crammer, Y. Singer. A family of additive online algorithms for category ranking* Machine Learning Research. №3. 2003. pp. 1025–1058
33. *Minh Duc Cao, Xiaoying Gao. Combining Content and Citation for Scientific Document Classification* AI2005, LNAI 3809, 2005. pp. 143-152.