УДК 004.85, 004.89

ТЕХНОЛОГИЯ РАЗРАБОТКИ СИСТЕМ ФИЛЬТРАЦИИ ИНТЕРНЕТ ТРАФИКА НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

В.В. Глазкова, В.А Масляков, И.В. Машечкин, М.И. Петровский

Московский государственный университет им. М.В. Ломоносова,
119071, Россия, Москва, Ленинские горы,
Тел.: 939 1789, e-mail: mash@cs.msu.su

Рассмотрен способ построения систем фильтрации Интернет трафика локальных сетей на основе методов машинного обучения. Огромное количество Интернет ресурсов, основная масса которых на сегодняшний день является динамическими, делают малопригодными традиционные сигнатурные подходы к анализу и фильтрации Интернет информации. Анализ мета информации о ресурсе, такой как URL и тип содержимого, а также анализ содержимого на основе ключевых слов не обладают достаточной точностью, обеспечивающей эффективное решение задачи фильтрации трафика. Авторами предложена оригинальная архитектура, использующая методы машинного обучения для решения задачи многотемной классификации Интернет ресурсов. В работе описаны основные модули системы, их алгоритмы работы и способ организации базы знаний. Разработанная архитектура экспериментально протестирована на эталонных тестовых наборах данных, результаты экспериментов показали достаточно высокую точность и скорость работы.

This report gives an overview of a method of constructing an Internet traffic filtering system based on machine learning approach. Large number of Internet resources, most of which today are dynamic, make little use of traditional signature approaches to the analysis and filtering of Internet information. Analysis of Internet resource meta-information, such as its URL and content type, or analysis based on keywords does not have sufficient accuracy to perform effective traffic filtering. The authors propose an original architecture, which uses machine-learning techniques to perform online multi-class multi-label classification of Internet resource content. This paper describes main modules, algorithms and knowledge base structure of such Internet traffic filtering system. Proposed architecture and algorithms were successfully implemented and tested on standard test data sets. Experiment results have shown fairly high accuracy and speed.

Введение

Проблема контроля доступа к Интернет-ресурсам актуальна и имеет важное прикладное значение по следующим основным причинам: блокирование доступа к нелегальной (экстремистской, антисоциальной и т.п.) информации, предотвращение доступа к Интернет-ресурсам в личных целях в учебное или рабочее время, предотвращение утечки конфиденциальной информации через Интернет.

На сегодняшний день существует множество как коммерческих, так и некоммерческих решений. К наиболее распространённым коммерческим продуктам можно отнести: open-source систему Poesia [1], коммерческие системы CyberPatrol [2], SurfControl [3], NetNanny [4] и др.

Три основных признака систем фильтрации трафика — это их масштаб, способ и время анализа трафика. По масштабу системы можно разделить на:

– комплексные и внедряемые в масштабах целой страны;

– средней сложности, рассчитанные на использование большим количеством пользователей и предоставляемые, как правило, в качестве отдельной услуги Интернет-провайдерами;

– независимые системы, устанавливаемые и настраиваемые в рамках отдельных локальных сетей или организаций.

По способу анализа все системы можно разбить на два больших класса:

– анализирующие лишь общую (мета-) информацию о ресурсе;

– анализирующие в том числе и содержимое (контент) ресурса.

По времени анализа все системы можно также разбить на два класса:

– анализирующие информацию в реальном времени (онлайн), т.е. во время запроса пользователем Интернет-ресурса;

– анализирующие информацию в отложенном режиме (оффлайн), т.е. после того, как пользователь получил доступ к ресурсу.

В данной работе рассматриваются системы масштаба локальных сетей, анализирующие как мета информацию, так и содержимое Интернет ресурсов в режиме реального времени.

Основные количественные показатели при оценке работы систем фильтрации Интернет-трафика следующие:

– точность анализа – процент верно отфильтрованных Интернет-ресурсов;

– излишнее блокирование или ложно-положительные ошибки – процент «хороших» ресурсов, ошибочно запрещенных системой фильтрации;

– недостаточное блокирование или ложно-отрицательные ошибки – процент «плохих» ресурсов, ошибочно разрешенных системой фильтрации;

– скорость анализа – максимальный объем данных, который система может проанализировать в единицу времени.

На сегодняшний день качество систем фильтрации трафика по-прежнему остается достаточно низким: при максимально достижимой точности анализа 90 % системы имеют либо очень большой процент ложно-положительных ошибок (2–5 %), либо низкую скорость анализа, вызывающую существенные задержки у конечных пользователей.

1. Существующие подходы

Традиционно в существующих системах анализа и фильтрации Интернет информации применяется так называемый **сигнатурный подход**, основанный на использовании экспертной базы знаний адресов Интернет-ресурсов. Такая база знаний содержит адреса ресурсов, с каждым из которых связан набор тем (категорий), к которым, по мнению экспертов, относится данный Интернет-ресурс.

Типичный сценарий работы системы Интернет фильтрации трафика, основанной на сигнатурном подходе:

– работа системы начинается с приведения базы данных сигнатур в актуальное состояние. Эта работа обычно осуществляется с помощью экспертов, обновляющих базу данных сигнатур;

– в базе данных сигнатур можно отметить некоторые ресурсы, как "положительные", или "легальные", потому, как в момент обновления их содержимое может быть абсолютно безвредным;

– после обновления базы данных начинается обработка запросов пользователей в режиме реального времени;

– если пользователь запрашивает Интернет-ресурс, помеченный ранее как легальный, то система предоставляет доступ к данному ресурсу;

– однако после обновления содержание ресурса могло измениться на нежелательное. В результате чего пользователь получает доступ к нежелательному содержанию.

К достоинствам таких систем можно отнести высокую скорость работы и централизованную базу данных сигнатур. Однако, системы, основанные на экспертных базах знаний адресов, обладают рядом существенных недостатков:

– невозможность анализировать трафик в реальном времени (онлайн). Анализ в реальном времени необходим, когда содержимое (контент) одного и того же ресурса может динамически изменяется во времени, а на сегодняшний день это свойственно подавляющему большинству Интернет-ресурсов;

– при анализе Интернет-ресурсов никак не учитывается их содержимое, что приводит к существенному снижению точности таких систем;

– невозможность анализа исходящего Интернет трафика (для предотвращения утечки конфиденциальной информации) ;

– необходимость использования внешних баз знаний о ресурсах, что может быть недопустимо по соображениям безопасности;

– качество функционирования таких систем существенно зависит от качества и оперативности компаний, поддерживающих постоянное обновление баз знаний. Как правило, для поддержания баз знаний в актуальном состоянии требуется большое количество экспертов. В связи со стремительными темпами роста Интернета осуществлять обновление баз знаний становится всё сложней и сложней как с технической, так и с экономической точки зрения.

Таким образом, применение сигнатурного подхода для анализа трафика имеет ряд существенных недостатков, связанных с неспособностью этого подхода адаптироваться к постоянной динамике изменения Интернет-ресурсов.

2. Предлагаемый подход

2.1. Предлагаемая архитектура системы и основные модули. Авторами предлагается система фильтрации трафика, основанная не на сигнатурных подходах, а на **методах машинного обучения**, а именно методах многотемной классификации Интернет ресурсов.

Интеллектуальные методы на основе обнаружения и применения знаний обладают такими достоинствами:

– самообучаемость и адаптируемость – способность автоматически оперативно подстраиваться к динамически изменяющемуся содержимому Интернет-ресурсов;

– автономность – независимость от внешних баз знаний и экспертов.

К основным недостаткам методов интеллектуального анализа данных можно отнести:

– необходимость наличия обучающего набора;

– повышенный риск ложно-положительных ошибок.

Основными модулями системы являются:

кэш-прокси-сервер – модуль, ответственный за перехват запросов из локальной сети и их переадресацию системе фильтрации трафика;

ядро – центральный модуль системы фильтрации трафика, через который выполняются все операции в рамках системы;

модуль принятия решений – модуль ответственный за принятия решения о разрешение или блокировке доступа к ресурсам;

модуль разбора и классификации – модуль, ответственный за лексический разбор содержимого ресурса и его классификацию;

робот – модуль, ответственный за скачивание содержимого ссылок из Интернета.

На рисунке 1 представлена предлагаемая архитектура интеллектуальной системы анализа и фильтрации Интернет информации.

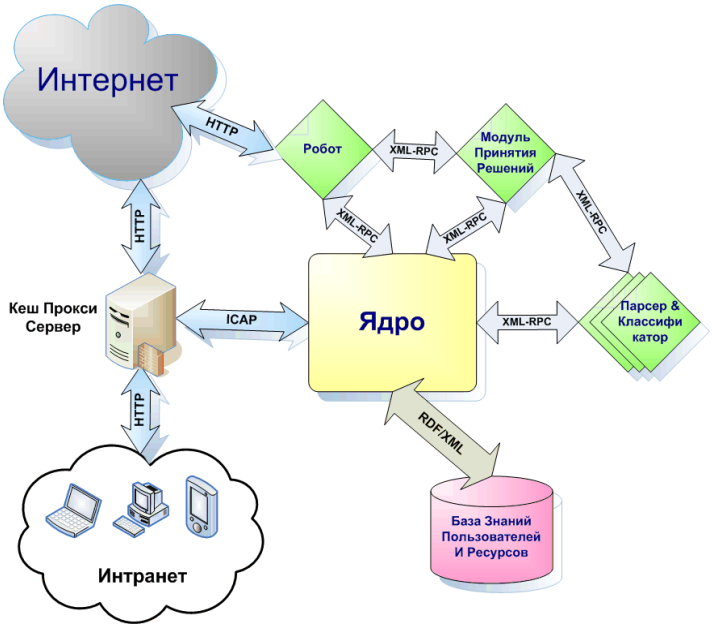

Рис. 1. Архитектура системы фильтрации трафика

Кэш-прокси-сервер используется в более чем 90 % систем масштаба локальных сетей. Сегодня существует множество различных реализаций кэш-прокси-серверов, такие как: Squid [5], Jigsaw (W3C Web Server), Shweby, Microsoft ISA Server и т.п.

Основными функциями кэш-прокси-сервера является анализ HTTP-трафика локальной сети с целью кэширования (сохранение) для оптимизации эффективности доступа пользователей локальной сети к Интернету за счёт сокращение среднего времени доступа к Интернет-ресурсам. Однако многие реализации позволяют использовать прокси-сервера не только для кэширования, но и для таких задач как фильтрация трафика и анализ содержимого на предмет наличия вирусов, троянов и прочего. Примерами таких кэш прокси-серверов могут быть Squid [5] или Shweby.

Для взаимодействия прокси-сервера и системы фильтрации трафика можно использовать один из следующих протоколов: XML-RPC [6], SOAP [7] или ICAP [8]. Для фильтрации трафика лучше всего подходит ICAP в силу того, что он является логическим расширением HTTP протокола и добавляет минимальное количество избыточной информации к анализируемым http запросам и ответам. В отличии от XML-RPC и SOAP он также является стандартизованным IETF протоколом и имеет множество эффективных реализаций. Более подробно о преимуществах и недостатках отдельных протоколов взаимодействия с кэш прокси-сервером можно ознакомится в [9]. Основным недостатком ICAP является то, что используемый прокси-сервер должен поддерживать данный протокол. Однако, на сегодняшний день большинство из популярных кэш прокси-серверов, например таких как Squid, его поддерживают.

Основная идея взаимодействия кэш прокси-сервера и системы фильтрации трафика с использованием протокола ICAP заключается в следующем: ICAP кэш прокси-сервер содержит встроенный ICAP-клиент, перенаправляющий новые HTTP запросы и ответы пользователей на ICAP Server, как правило встроенный в ядро системы фильтрации трафика.

Протокол ICAP как уже отмечалось ранее очень похож на HTTP и поддерживает три основные команды:

- OPTIONS — используется для получения настроек кэш прокси-сервера;
- REQMOD — используется для фильтрации входящих запросов пользователей;
- RESPMOD — используется для фильтрации ответов Интернет.

Кэш прокси-сервер не делает различий между типами анализируемого HTTP-трафика и перехватывает как входящий, так и исходящий трафик. Исходящий трафик перехватывается на этапе фильтрации запроса пользователя. В этом случае система осуществляет фильтрацию на основе IP-адреса или домена машины, к которой адресован запрос, либо на основе содержимого запроса, используя методы классификации или выделения ключевых слов.

Ядро – центральный элемент системы. В него встроен ICAP-сервер [8], получающий и фильтрующий запросы от кэш прокси- сервера. Основными функциями ядра являются:

контроль процесса фильтрации входящего и исходящего трафика, т.е. идентификация того, кто запрашивает информацию, хранение каждого запроса в базе знаний, передача запросов модулю принятия решений нет, сохранение результатов классификации и модуля принятия решений в базе знаний;

предоставление API для других модулей, например API для сохранения ссылок, полученных с помощью анализа классификатором содержимого ресурса, API для модуля принятия решений, который может запросить дополнительную информацию о ресурсах, пользователях или статистике;

организация работы с базой знаний и предоставление интерфейса базы знаний, которая позволит пользователям и администраторам системы смотреть статистику и настраивать систему;

идентификация того, кто запрашивает информацию. В настоящее время предлагается использовать идентификацию по IP-адресу, но технически можно добавить идентификацию с помощью LDAP и других протоколов;

хранение белых списков разрешенных доменов и IP-адресов, черных списки запрещенных доменов IP-адресов, пользователей системы и их прав для различных категорий ресурсов.

Каждый пользователь может принадлежать к одной или нескольким группам. Каждому пользователю или группе назначается белый и черный список разрешенных и запрещенных доменов и IP-адресов, а также список разрешенных и запрещенных категорий ресурсов. Для идентификации ресурсов используется его URL. Поэтому каждый запрос однозначно идентифицируется временем запроса, пользователем, который его запросил, URL ресурса. Для совместимости с другими компонентами было решено использовать XML-RPC–протокол [6] из-за своей простоты, большого числа библиотек, поддержке различных языков, стабильности, масштабируемости и эффективности. Использование XML-RPC позволяет писать компоненты на разных языках и размещать их на разных машинах.

Одной из основных частей системы является модуль принятия решений. Основной задачей этого модуля является анализ данных, поступающих в ядро, и принятие решений: разрешить ли или блокировать тем или иным пользователям доступ к запрашиваемому Интернет-ресурсу.

Модуль принятия решений работает в два этапа:

анализ и фильтрация запросов поступающих от пользователей. На данном этапе ядро передает следующие параметры модулю принятий решений: пользователь, запрашивающий информацию, как упоминалось выше, для этого предлагается использовать IP-адрес машины, с которой запрашивается ресурс, URL ресурса и метаинформация о ресурсе, т.е. все заголовки, полученные из HTTP-запроса. Используя эту информацию, модуль пытается принять решение, которое может быть принято, если, например, домен запрашиваемого ресурса попал в белый или черный список для текущего пользователя, или если категории ресурса были получены ранее;

если этой информации недостаточно для модуля принятия решений, он запрашивает содержимое ресурса. Ядро перенаправляет запрос кэш-прокси-серверу, тот загружает содержимое из Интернета и возвращает его ядру. Ядро вызывает метод модуля принятий решений, отвечающий за фильтрацию содержимого. Вместе с содержимым передаётся информация о пользователе, сайте ресурсов, дополнительные метаданные, такие как тип содержимого ресурса, дата последней модификации и другие метаданные, полученные из HTTP-ответа. Для получения информации о категориях ресурса модуль принятия решений может обратиться к классификатору.

2.2. Модуль классификации на основе методов машинного обучения. Одним из центральных модулей системы является модуль классификации. Модуль решает задачу определения релевантных тем HTML-документов. Задача определения релевантных тем документов состоит в предсказании для HTML-документов набора релевантных тем (из предопределенного набора анализируемых тем). Для решения этой задачи модуль осуществляет лексический разбор (парсинг) HTML-документов, преобразуя их в некоторое внутреннее представление. Далее модуль решает задачу многотемной (multi-label) классификации, используя это выбранное представление в качестве формального представления HTML-документов.

Модуль работает в двух режимах: режиме обучения и режиме классификации новых HTML-документов.

В режиме обучения на основе обучающей совокупности, состоящей из заранее рубрицированных HTML-документов, строится математическая модель классификации, которая позволит определять релевантные категории для произвольных ресурсов схожего содержимого. Впоследствии эта математическая модель может уточняться за счёт пошагового дообучения на новых ресурсах, для которых известны релевантные категории. В предлагаемом подходе учитывается, что Интернет-ресурсы, как правило, являются многотемными (multi-label), то есть каждый Интернет-ресурс может быть отнесен более чем к одной релевантной теме или категории. Для решения задачи многотемной классификации реализован подход на основе декомпозиции multi-label-проблемы

в набор задач бинарной классификации на основе подходов "каждый-против-остальных" и "каждый-против-каждого". Для начального обучения бинарных классификаторов используется SVM, а для дообучения модели – Kernel Perceptron.

В режиме классификации новых документов классификации осуществляется применение построенной модели к новому классифицируемому документу. В результате этого получаем значения релевантности для всех тем (из предопределённого на этапе обучения набора тем), находим такой новый порог (пороговое значение) и далее на основе его уже выделить наиболее релевантные темы. Пороговое значение, как и релевантности тем, также определяется на основе модели и характеристик нового классифицируемого документа.

Для представления HTML-документов реализованы подходы на основе ключевых слов (стемминг) и *n*-грамм. Стемминг подразумевает разделение документа на слова и выделение корней данных слов, в случае *n*-грамм документ разбивается на участки длинной *n*, каждый из которых трактуется как отдельное слово.

В качестве меры сходства используется частотная мера сходства (TF-IDF), а также модифицированная мера сходства на основе *k*-spectrum kernel. Кроме того, для повышения точности классификации при преобразовании во внутреннее представление модуль учитывает ссылочную структуру HTML-документов, производя замену ссылок в данном документе на идентификаторы релевантных тем, к которым они относятся.

Таким образом, предлагается осуществлять категоризацию ресурсов и содержимого Интернет-трафика на основе не только статических правил, заданных экспертом, но и на основе построения и применения Data Mining-моделей классификации гипертекстовой информации, что позволяет сделать систему адаптивной и обучаемой. Исследование существующих методов классификации многотемных объектов применительно к задаче фильтрации Интернет-информации показало, что эти методы не имеют возможности дообучения, которая очень важна для рассматриваемой задачи.

Единственным существенным недостатком является необходимость наличия обучающих данных. Для этой цели может использоваться один из стандартных наборов данных, таких как Reuters-2000 [10], который может быть дополнен организацией, в которой будет использоваться система.

При использовании методов машинного обучения, сценарий работы системы фильтрации Интернет-трафика выглядит следующим образом:

работа системы начинается с полностью автоматизированного процесса обучения. В ходе этого процесса система обучается на некотором обучающем наборе, например Reuters-2000;

если далее в какой-то момент времени пользователь запрашивает некоторый Интернет-ресурс с нежелательным содержанием, запрос перенаправляется системе Интернет-фильтрации;

в отличие от сигнатурного подхода система выполняет полный анализ содержимого в реальном времени и присваивает анализируемому ресурсу, заданные на этапе обучения категории (процесс классификации);

на основе результатов классификации и текущих прав пользователя, система принимает решение о разрешении или запрете доступа к нежелательному содержанию.

2.3. База знаний на основе онтологического представления. В качестве базы знаний предлагается использовать базу данных на основе онтологического представления ресурсов. Онтологии в общем случае описывают: классы/типы/множества объектов; объекты; аттрибуты/свойства/характеристики объектов; отношения между объектами; события – изменение аттрибутов или отношений.

В своё время было разработано несколько распространённых языков описания онтологий:

1. OWL [11], стандарт World Wide Web Consortium, стандарт, развившийся из RDF [12] и RDFS.
2. DAML+OIL, стандарт ISO, предшественник OWL.
3. CycL, язык, разработанный Cycorp Inc. для собственной онтологии Cyc.

На сегодняшний день OWL является самым распространённым языком для создания и описания онтологий. OWL является более мощным средством описания, нежели XML, RDF или RDFS. OWL как и RDF основан на выражениях-тройках <субъект, предикат, объект>. Самым частым представлением OWL, как и в случае с RDF, является XML-нотация.

Для работы с онтологиями можно использовать такие средства разработки онтологий и баз знаний как Jena [13] и Protege [14]. В качестве низкоуровневого хранения можно использовать как реляционные БД MySQL, PostgreSql, Oracle, MSSQL, так и родные RDF базы данных, например AllegroGraph [15].

Схема базы знаний представляет с собой онтологию, описаную с помощью OWL, мощности OWL-Lite с использованием концепций из таких онтологий как Dublin Core и FOAF.

Для описания ресурсов предлагается использовать классы InteractiveResource из онтологии Dublin Core [16] и его свойства, а для описания пользователей отлично подходят такие концепции онтологии FOAF [17], как User и Group и их свойства: membership, first name, last name, e-mail и другие.

Данные две онтологии являются на сегодняшний день двумя самыми популярными способами описания ресурсов и пользователей соответственно. Получившаяся онтология системы фильтрации трафика будет понятна как другим системам, использующим онтологии Dublin Core и FOAF, так и в свою очередь может получать данные из этих же систем. Для работы с базой знаний системы фильтрации трафика, другим системам

достаточно будет скачать онтологию базы знаний. Таким образом выбранное представление базы знаний обеспечивает удобный способ хранения и доступа к данным системы, а также удобный экспорт/импорт и взаимодействие с существующими системами. Общая схема базы знаний и её связь с основными функциями системы показана на рис. 2.

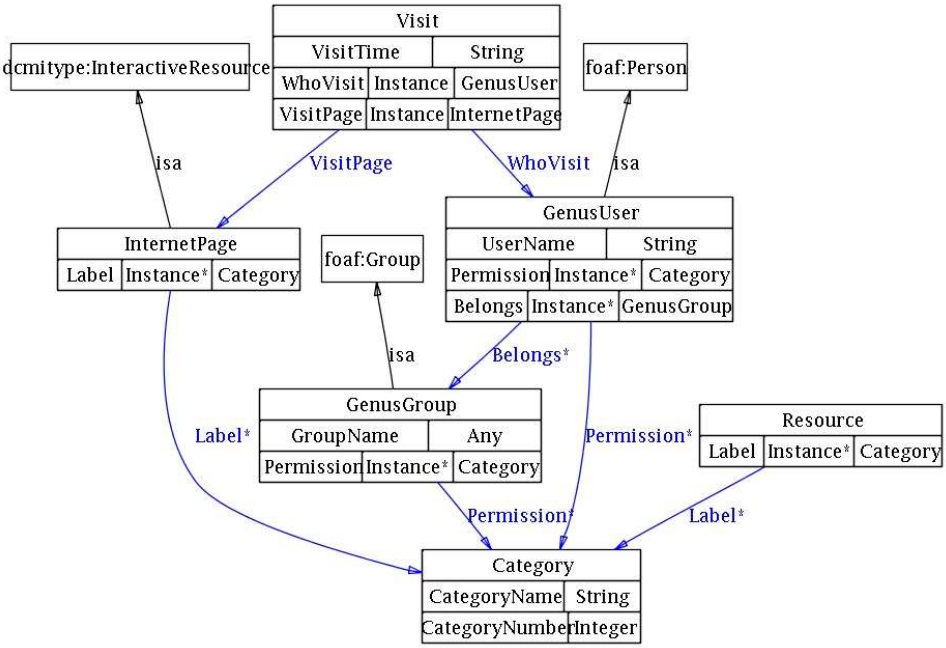

Рис. 2. Онтология базы знаний

3. Эксперименты

3.1. Тестовый набор. При тестировании прототипа системы на производительность использовался обучающий набор BankResearch [18]. Обучение происходило на 1100 документах из заданного набора: по 100 документов на каждую из 11 тем.

В качестве тестовых страниц использовались набор релевантных страниц по 11 соответствующим тематикам, взятых из каталога http://directory.google.com и отобранных вручную.

Набор тестовых страниц задаётся списком:

категория 1, URL документа 1

категория 1, URL документа 2

.....

категория *j*, URL документа *i*

…

категория 11, URL документа *n*.

В силу того, что обучающий набор состоял из англоязычных ресурсов, тестовый набор также содержал лишь англоязычные ресурсы.

3.2. Критерии оценки скорости и точности работы. Основные характеристики оцениваемые при тестировании прототипа системы и её модулей: скорость классификации, скорость принятия решений системой, скорость работы базы знаний, точность работы системы.

Скорость классификации отражает насколько быстро модуль классификации определяет категорию анализируемого документа.

Скорость принятия решения отражает, сколько потребуется в среднем времени, чтобы принять решение о том, разрешить или запретить доступ к ресурсу. В данной цепочке участвуют следующие компоненты системы: ядро, модуль принятия решений, парсер, классификатор, база знаний. Учитывается время, потраченное на разбор документа парсером и классификацию, время принятия решений и время сохранения в базу знаний. Общее время принятия решений складывается из этих времён и накладных расходов, связанных с межмодульными вызовами функций.

Отдельно замерялись время скачивания Интернет-ресурса из Интернета и время сохранения в базу знаний.

На основе полученных данных рассчитывались такие показатели как:

отношение скорости скачивания ресурса к общей скорости принятия решений, отражающее насколько меньше по сравнению с временем загрузки время фильтрации системой;

отношение скорости сохранения в базу знаний к общей скорости принятия решений, отражающее насколько эффективно работает предлагаемая база знаний и выбранное представление ресурсов, пользователей и статистики.

Под *истинной категорией* понимается та категория, которая была вручную выбрана на этапе составления тестового набора.

Точность работы системы оценивалась результатами классификации, а именно следующими параметрами:

промежуточная точность классификации (0 <= Pi <= 1), отражает какова вероятность по мнению классификатора, что ресурс принадлежит своей истинной категории;

окончательная точность классификации, отражает принадлежит ли с учётом пороговой функции ресурс к своей истинной категории или нет. Возможные значения: 1 – если принадлежит, 0 – если не принадлежит.

Помимо обозначенных параметров, также замерялись размер Интернет ресурса в килобайтах и рассчитывались средние показатели для каждого из полученных параметров.

При тестировании использовалась следующая аппаратно-программная платформа: компьютер с процессором Amd Athlon 64 3200+, 2048 мегабайтами ОЗУ, жёстким диском Western Digital 2500JS, установленной операционной системой Ubuntu Fiesty 7.04 (linux kernel 2.6.20-16), веб сервером apache tomcat 5.0 и базами данных Postgresql 8.2.4 и Berkeley DB 4.6.

При тестировании использовались следующие параметры классификации:

– разбор документов с использованием стемминга, т.е. выделением корня каждого слова, встретившегося в html документе;

– разбор документов на основе N-грамм (N = 3), т.е. разбиением документов на лексемы, фиксированной длины N = 3.

Результаты тестов сгруппированы по тематикам, т.е. вычислены средние значения показателей производительности и точности для 10 документов каждой из тематик (табл. 1 и табл. 2).

Таблица 1. Таблица результатов с использованием стемминга

Название тематики	**Размер документа (кб)**	**Время Скачивания (сек)**	**Время Классификации**	**Время сохранения в БЗ**	**Время принятия решений**	**Время скачивания / Время принятия решений**	**Время сохранения в БЗ / Время принятия решений**	**Промежуточная точность классификации**	**Окончательная точность классификации**
Банки	23.0	2.8432	0.3753	0.0189	0.6221	4.4356	0.0295	0.6847	1.0
Общество	19.0	2.4136	0.3881	0.0137	0.5303	4.4368	0.0252	0.6868	0.9
Страхование	25.0	2.7283	0.4157	0.0192	0.7387	3.5998	0.0253	0.6674	1.0
Java	29.0	2.968	0.498	0.0143	0.7576	3.8451	0.0185	0.7332	0.7
C/C++	34.0	3.6262	0.4867	0.0188	1.0319	3.4512	0.0179	0.9987	1.0
Visual Basic	33.0	1.8976	0.3772	0.0147	1.3864	1.3544	0.0105	0.7912	0.8
Астрономия	16.0	0.9588	0.2932	0.0169	0.638	1.464	0.0258	0.5628	0.8
Биология	40.0	2.7806	0.403	0.02	1.2325	2.22	0.016	0.8146	1.0
Футбол	31.0	0.7367	0.349	0.0146	1.1736	0.62	0.0123	0.9366	1.0
Мотоциклы	39.0	1.5798	0.3278	0.0141	1.6981	0.9227	0.0082	0.8968	1.0
Спорт	41.0	2.2696	0.5429	0.0141	1.3980	1.6073	0.0010	0.5420	0.9
Итого	**33.0**	**2.0484**	**0.3558**	**0.015**	**0.8917**	**2.2591**	**0.0166**	**0.7559**	**0.918**

Таблица 2. Таблица результатов с использованием N-грамм

Название тематики	**Размер документа (кб)**	**Время Скачивания (сек)м**	**Время Классификации**	**Время сохранения в БЗ**	**Время принятия решений**	**Время скачивания / Время принятия решений**	**Время сохранения в БЗ / Время принятия решений**	**Промежуточная точность классификации**	**Окончательная точность классификации**
Банки	23.0	2.0524	0.6419	0.0068	0.9194	2.2159	0.0073	0.7699	1.0
Общество	19.0	1.0848	0.5202	0.0105	0.6431	1.6597	0.0161	0.7383	1.0
Страхование	25.0	2.5475	0.7005	0.0133	0.8769	2.8617	0.0149	0.7937	1.0
Java	29.0	2.4737	0.7968	0.0128	0.8965	2.7204	0.0141	0.7546	0.9
C/C++	34.0	3.3574	0.8047	0.0118	1.1456	2.9008	0.0102	0.9971	1.0
Visual Basic	33.0	2.7263	0.7079	0.0123	1.5719	1.7209	0.0078	0.7294	0.8
Астрономия	16.0	1.7221	0.5346	0.0133	0.8849	1.9173	0.0148	0.8635	0.9
Биология	40.0	3.856	0.6302	0.0137	1.3895	2.748	0.0098	0.8621	1.0
Футбол	31.0	0.933	0.5972	0.0139	1.2802	0.721	0.0107	0.9548	1.0

Мотто-циклы	39.0	2.3968	0.5735	0.013	1.8206	1.3072	0.0071	0.8748	1.0
Спорт	41.0	3.8290	0.8478	0.0194	1.514	2.4971	0.0127	0.6188	0.7
Итого	**33.0**	**2.1045**	**0.5916**	**0.011**	**1.039**	**2.0043**	**0.0105**	**0.8143**	**0.9363**

Заключение

Предложенная архитектура системы фильтрации трафика на основе методов машинного обучения позволил получить: необходимую скорость анализа (десятичные доли секунды); необходимую точность анализа (более чем 90 % с 1 % ложных положительных ошибок); адаптацию и самообучение, позволяющие подстраиваться к потребностям конкретной организации; масштабируемость системы, позволяющую устанавливать систему в организациях различного масштаба; независимость от внешних баз знаний и экспертов.

При тестировании компонентов системы фильтрации трафика на производительность средний размер страницы оказался равным 33 килобайтам, а среднее время скачивания порядка 2–2.3 секунд. С другой стороны, среднее время классификации не превышало 1 сек., время сохранения в базу знаний оказалось на порядок меньше времени классификации, а суммарное время принятия решений оказалось в среднем в 2 раза меньше времени скачивания.

Таким образом, применение интеллектуальных методов фильтрации на основе методов машинного обучения не значительно увеличивает время ожидания выполнения HTTP-запросов пользователей, а значит, предложенный подход может эффективно использоваться в реальных системах фильтрации.

Работа поддерживавается грантами РФФИ № 06-01-00691, грантом поддержки научных школ № 02.445.11.7427 и грантом президента РФ МК-4264.2007.9.

1. *Present* and Future of Open-source Content-based Web Filtering [электронный ресурс] : настоящее и будущее систем контентной фильтрации веб-трафика с открытыми исходными кодами // ILC.- Режим доступа: http://www.ilc.cnr.it/poesia_prg/POESIA_FinalWorkshop_Program.htm.
2. *CyberPatrol* Internet Security Software [электронный ресурс] : коммерческая система фильтрация трафика CyberPatrol // SurfConrol plc.- Режим доступа: http://www.cyberpatrol.com/.
3. *SurfControl* [электронный ресурс] : коммерческая система фильтрации трафика масштаба локальных сетей на основе URL и ключевых слов / SurfControl plc.- Режим доступа: http://www.surfcontrol.com/.
4. *NetNanny* Parental Control [электронный ресурс] : коммерческая система родительского контроля детского доступа в Интернет.- Режим доступа: http://www.netnanny.com/.
5. *Squid*: Optimising Web Delivery[электронный ресурс] : Open-source кэщ прокси-сервер. – Режим доступа: http://www.squid-cache.org/
6. Xml-RPC Home Page [электронный ресурс] : протокол межмодульного взаимодействия XML-RPC // Dave Winer.- изд. 15.06.1999. – Режим доступа http://www.xmlrpc.com/.
7. *SOAP* [электронный ресурс] : Протокол доступа к объектам // World Wide Web Consortium.- Режим доступа: http://www.w3.org/TR/soap/.
8. *Internet* Content Adaptation Protocol (ICAP) [электронный ресурс] : протокол модификации Интернет запросов // J. Elson, A. Cerpa. – Режим доступа: http://www.ietf.org/rfc/rfc3507.txt.
9. *ICAP* vs. SOAP: Which One is Better for Edge Services [электронный ресурс]: ICAP или SOAP. Что лучше для граничных сервисов / Vikrant Mastoli, Valmik Desai and Weisong Shi.- Режим доступа: http://www.cs.wayne.edu/~weisong/papers/mastoli03-see-techreport.pdf
10. *Reuters* Corpora [электронный ресурс]: Описание набора данных Reuters-2000.- Режим доступа: http://trec.nist.gov/data/reuters/reuters.html.
11. *Ontology* Web Language [электронный ресурс] : язык для описания веб-онтологий // World Wide Web Consortium.- Режим доступа: http://www.w3.org/TR/owl-features/.
12. *Resource* Description Framework [электронный ресурс] : язык описания ресурсов // World Wide Web Consortium.- Режим доступа: http://www.w3.org/RDF/.
13. *Jena* Semantic Web Framework [электронный ресурс] : библиотека для работы с онтологиями с открытыми исходными кодами // Hewlett-Packard Development Company, L.P..- Режим доступа: http://jena.sourceforge.net/.
14. *The* Protege Ontology Editor and Knowledge Acquisition System [электронный ресурс]: система для создания, редактирования и работы с базами знаний / Stanford Medical Informatics.- Режим доступа: http://protege.stanford.edu/.
15. *AllegroGraph* 64bit RDF-Store [электронный ресурс] : специализированная база данных для хранения и доступа к RDF триплетам // Franz Inc. 2000-2007.- Режим доступа: http://franz.com/products/allegrograph/.
16. *Dublin* Core Metadata Initiative [электронный ресурс]: язык описания ресурсов на основе OWL и RDF // DMCI. – Режим доступа: http://dublincore.org/.
17. *Friend* of a Friend (FOAF) project [электронный ресурс] : онтология для описания социальных сетей на основе OWL // Robert Benchley.- Режим доступа: http://www.foaf-project.org/.
18. *Bank* Research Dataset [электронный ресурс]: Набор данных BankResearch. – Режим доступа: http://lib.stat.cmu.edu/datasets/bank-research.zip.