Применение методов машинного обучения для автоматизации тематической разметки интернет-доменов

А. Т. Тлеубаев[1,2]***, С. А. Ступников***[1,2,3]
a.tleubayev@corp.mail.ru, sstupnikov@ipiran.ru
[1]Mail.ru, Москва, Ленинградский проспект, д.39;
[2]МГУ имени М. В. Ломоносова, Москва, Ленинские горы, д. 1;
[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Работа посвящена применению методов машинного обучения для задачи автоматизации тематической разметки интернет-доменов. Конкретная задача состоит в автоматическом отнесении интернет-домена к некоторой категории из предопределенного иерархического дерева категорий. Применялись различные классификаторы, хорошо зарекомендовавшие себя в работе с сильно разреженными признаковыми пространствами большой размерности. Признаковые пространства формировались на основании текстов с главных страниц доменов с применением подходов TF-IDF и N-грамм. Разработаны два подхода к применению методов классификации для решения задачи: прямой и многоуровневый. При прямом подходе применяется единственный классификатор, для каждого домена предсказывается его категория, которая может быть любого уровня в дереве категорий. При многоуровневом подходе применяется множество классификаторов: каждому множеству категорий с одним родителем соответствует отдельный классификатор. Классификаторы применяются иерархически — от корневых категорий к листовым. Используется также комбинация предложенных подходов. Одним из практических применений работы является профилирование пользователя на основании посещенных им сайтов и дальнейшее предложение персонализированной рекламы.

Ключевые слова: *тематическая разметка интернет-доменов; машинное обучение; классификация текстов; классификация веб-страниц*

DOI: 10.21469/22233792.4.3.05

1 Введение

В современном мире огромное количество людей имеет доступ к сети Интернет. Практически каждый из нас ежедневно посещает множество разнообразных сайтов. Сайты, в свою очередь, оснащены счетчиками, которые регистрируют посетителей. Информацию с данных счетчиков можно использовать для анализов в различных целях. Одним из важных применений является анализ поведения пользователей, показывающий, какие сайты посещает конкретный пользователь. Большую часть сайтов можно разделять по тематике на категории и подкатегории, образуя, тем самым определенную иерархию. На основании категорий посещенных сайтов, для пользователя можно сформировать его профиль. В дальнейшем опираясь на профиль, пользователю может быть предложена персонализированная реклама.

В компании Mail.ru для подобной задачи используется специализированная система поддержки тематической разметки интернет-доменов (раздел 3), заполненная размеченными по категориям доменами.

Система поддержки тематической разметки доменов содержит дерево, содержащее обширное количество категорий по различным тематикам. Постоянное пополнение и расширение дерева является важной задачей. Однако, ручная разметка доменов — это очень трудозатратная задача. Разметка доменов могла бы продвигаться гораздо быстрее и была бы более экономически эффективной при снижении временных и материальных затрат за счет частичной автоматизации, а также оптимизации с помощью использования методов машинного обучения.

Данная работа посвящена применению методов машинного обучения, а именно методов классификации, для автоматизации тематической разметки доменов, с признаками, основанными на текстах с главных страниц сайтов. В работе описываются этапы предварительной обработки данных, полученных из системы поддержки тематической разметки доменов, выбор подходов, признаковых пространств и классификаторов для решения задачи, их тестирование и анализ результатов работы. Проведен анализ ошибок, их классификация и выяснение причин их возникновения.

В разделе 2 рассмотрены родственные работы, на основе которой производился анализ уже имеющихся методов решения задачи классификации интернет-доменов и текстов. В разделе 3 рассматриваются исходные данные; описываются этапы предварительной обработки, способы разбиения на наборы данных. В разделе 4 приводится описание подходов к применению методов машинного обучения для автоматизации тематической разметки; производится разбор каждого из методов. В разделе 5 дается описание признаковых пространств и классификаторов, работа которых проверяется в экспериментах. В разделе 6 дается описание программной реализации разработанного метода тематической разметки доменов. В разделе 7 представлены результаты работы разработанного метода; приводится анализ результатов, выявление ошибок. В разделе 8 представлены выводы по проделанной работе и планы по дальнейшим исследованиям.

2 Родственные работы

Задача тематической разметки веб-страниц значительно отличается от классической задачи классификации текстов из-за наличия большого количества дополнительной информации, представленной структурой HTML, изображениями, гиперссылками, шумовым контентом в виде рекламных баннеров, панелей навигации и т. д. [1–3]. Применение методов классификации текста к веб-страницам приводит к смещению алгоритма классификации, потере ориентации на основные темы и контент.

В работе [2] предлагается использовать специальные методы резюмирования текста, ориентированные на структуру веб-страниц. Была показана жизнеспособность и эффективность методов резюмирования веб-страниц для извлечения основной тематики страницы. Резюмирование текста позволило в среднем улучшить точность на величину около 9% по сравнению с классификацией на основе чистого текста. Специфичность методов резюмирования заключается в использовании слов из *title* и *meta*-тегов, отдельно производится учет слов с ссылками, отмеченных курсивом, подчеркиванием и т. д. Примеры использования в виде признаков как обычного текста, так и контекста, гиперссылок и разметки HTML, для обучения классификатора SVM приведены в работе [4].

Рассматривалось также применение извлеченной информации из URL-адресов, проверялось доменное сходство страниц. В качестве признаков использовались кодировки стран в URL-адресах, проводились работы по нахождению соответствия между словами в *title* и URL-адресе при помощи посимвольного подхода [5].

К сожалению, при решении поставленной задачи использование резюмирования веб-страниц на таком уровне, на котором это предложено в работе [2], сопряжено со значительными трудностями. Во-первых, на некоторых веб-страницах поля *title* и *meta*-тегов могут быть пустыми. Во-вторых, *title* и *meta*-теги могут быть заданы бессмысленно или заполнены по умолчанию. Также разработчики могут злоупотреблять возможностями, добавляя неправильные поля, для того, чтобы поисковые системы иначе анализировали веб-страницу. Помимо этого, одной из проблем является то, что в упомянутых выше работах использовались готовые наборы данных, вручную размеченные людьми. Наборы содержали более миллиона веб-страниц, большая часть из которых была хорошо структурирована, содержала конкретную информацию в *title* и *meta*-тегах.

На основании анализа родственных работ было принято решение использовать в данном исследовании классические методы классификации текстов, при этом извлечение признаков производится обычно при помощи методов TF-IDF [6] и N-грамм [7], а для классификации используются линейные или метрические классификаторы [8–10]. В работе [8] приведены примеры работы классических методов классификации текстов, их сравнение со сверточными нейронными сетями. Эксперименты показали, что сверточные нейронные сети справляются с задачей классификации текстов в общем случае не хуже конкурирующих методов. На больших наборах данных (более 1 миллиона текстов) лучше работают сверточные нейронные сети. Также одним из преимуществ сетей оказалось возможность обрабатывать тексты с неизвестным синтаксисом, так как в сетях применяется посимвольный подход [11].

В работах [9, 12] используются немного усовершенствованные классические методы классификации, производятся надстройки к классификаторам SVM и kNN, что позволяет добиться чуть лучших результатов.

3 Данные о тематической разметке доменов и их предварительная обработка

Система поддержки тематической разметки доменов включает дерево категорий, в котором представлена 41 независимая *корневая* категория; всего же в дереве 899 категорий (по состоянию на июнь 2017 г.). Каждая корневая категория посвящена некоторой тематике, а каждая ее категория-потомок посвящена более узкой тематике корневой категории — это категории второго уровня дерева категорий. Категории второго уровня также имеют категории-потомки, посвященные еще более узкой тематике категории-родителя. Например, корневой является категория «Компьютеры», примером ее категории-потомка второго уровня являются «Комплектующие», примером категории третьего уровня — «Видеокарты». Максимальный уровень категории в дереве — пятый. *Листовой* категорией называется категория, не имеющая категорий-потомков.

Данные по каждой категории включают множество пар из *шаблонов* и количества посещений страниц, удовлетворяющих шаблону за сутки; список ключевых фраз, характеризующих категорию; а также общую статистику посещений доменов. Шаблон может представлять собой домен, регулярное выражение на страницах домена или пару домен — набор ключевых фраз. Всего в категориях представлено около 120 тыс. шаблонов.

Ниже приведены примеры различныых видов шаблонов:

- домен — `microsoft.com`;
- домен и регулярное выражение — `avito.ru @ ^https?://(www\.)?(\w+\.)?avito\.ru/[^/]+/tovary_dlya_kompyutera.*$`;
- домен и фраза — `aliexpress.com @ #phrases`.

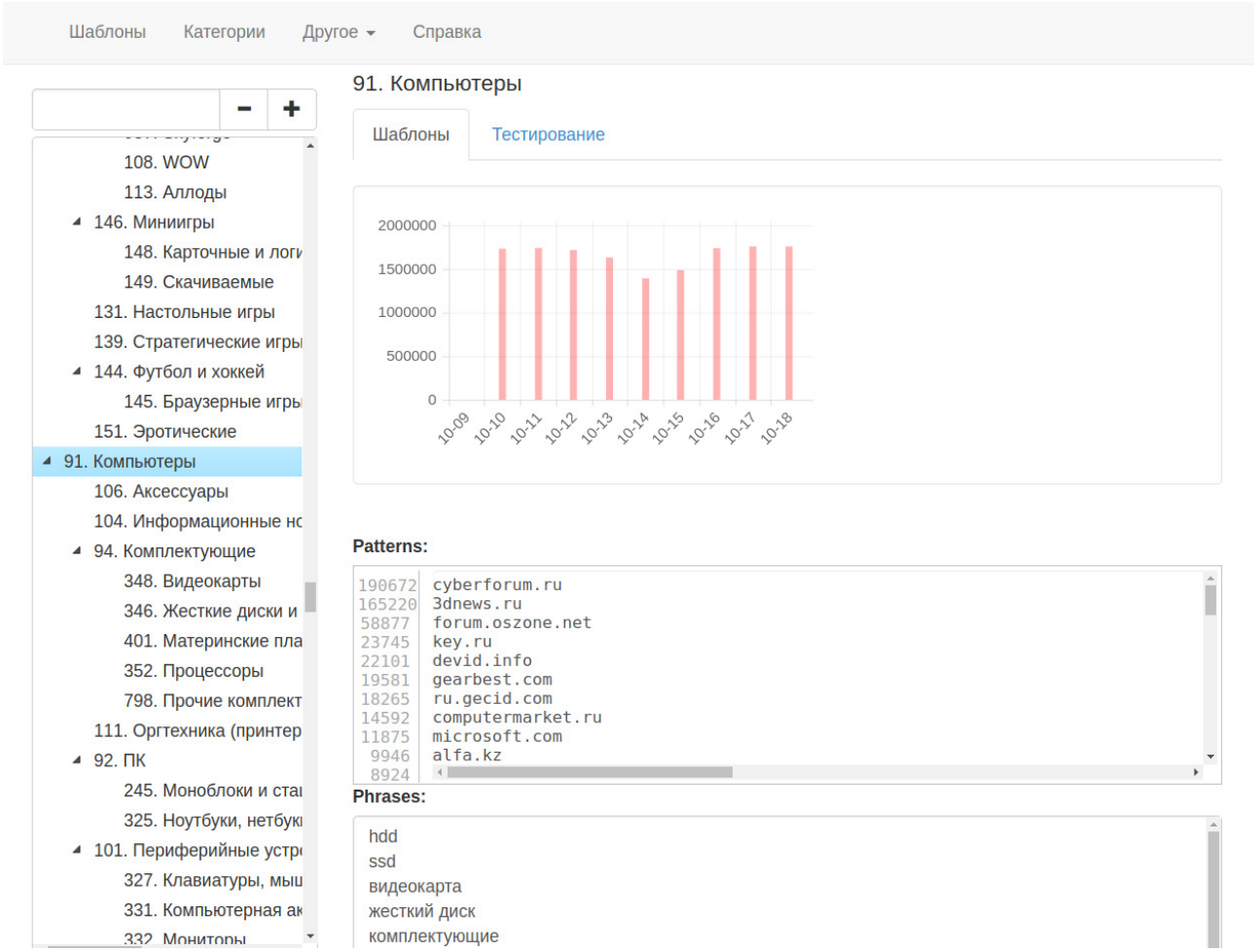

Рис. 1 Интерфейс системы поддержки тематической разметки интернет-доменов

Домен не всегда полностью соответствует тематике одной категории. С использованием регулярных выражений (диалект регулярных выражений из языка Python) можно производить разделения страниц домена по различным категориям для более точного подсчета статистики посещаемости. Фразы используются в том случае, если разделение страниц домена при помощи регулярных выражений невозможно. Тогда производится поиск и подсчет посещаемости страниц домена, содержащих в своем названии словосочетания из списка фраз.

К сожалению, сбор и обработка данных со всех страниц домена являются достаточно трудозатратными процессами, поэтому в работе производился сбор данных только с главных страниц доменов. Опыт показывает, что, как правило, на главных страницах достаточно информации о тематике домена. Поэтому регулярные выражения и списки фраз в рамках данной работы не учитываются — их использование является предметом дальнейших исследований.

Таким образом, входные данные из системы разметки доменов имеют следующий формат — это множество пар $\langle$*домен, категория*$\rangle$.

3.1 Предварительная обработка данных

Для каждого домена из дерева категорий был скачан текст с главной страницы. Домены, для которых скачать текст не удалось, были удалены из набора данных. Также были

удалены все домены, при скачивании которых произошли какие-либо ошибки (403, 404, 503 и т. д.).

Для подготовки текстов были проведены следующие операции:

- удален контекст, относящийся к кодировкам html-страниц;
- применен морфологический анализатор (из библиотеки pymorphy2 [13] для языка программирования Python), при помощи которого был проведен разбор каждого слова, которое было приведено к нормальной форме (например, форма единственного числа, именительного падежа для существительных);
- тексты были объединены в единый корпус;
- произведено формирование списка слов, встречающихся в текстах не более 5 раз; произведено удаление из текстов слов из полученного списка;
- произведено удаление предлогов, местоимений и часть наиболее часто встречающихся слов.

Рассмотрим пример результатов предварительной обработки следующего текста:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3c.
org/TR/1999/REC-html401-19991224/loose.dtd"> <html xml:lang="en" xmlns="http://
www.w3.org/1999/xhtml"><head>    <title> Продажа компьютеров по низким ценам (о
т 9000р): компьютерный магазин предлагает за 9000р - 155000р купить компьютер в
офис. Сборка на заказ в нашем интернет магазине компьютеров.</title> <meta cont
ent="Компьютерный интернет-магазин. Продажа компьютеров. Купить компьютеры в оф
ис. Сборка на заказ в нашем интернет магазине компьютеров. обзор характеристики
магазин
```

Текст после удаления html-разметки выглядит следующим образом:

```
Продажа компьютеров по низким ценам (от 9000р): компьютерный магазин предлагает
за 9000р - 155000р купить компьютер в офис. Сборка на заказ в нашем интернет маг
азине компьютеров.\nГлавная |\nО компании |\nКонтакты |\nГарантия |\nДоставка |\
nСпособы оплаты\nНе дозвонились?\nx\nОставьте Ваш телефон, мы перезвоним: \nВаше
имя:\nВаш
```

Текст после нормализации выглядит следующим образом:

```
продажа компьютер по низкий цена от 9000р компьютерный магазин предлагать за 900
0р 155000р купить компьютер офис сборка на заказ наш интернет магазин компьютер
главный компания контакт гарантия доставка способ оплата не дозвониться оставить
ваш
```

Текст после удаления низкочастотных слов выглядит следующим образом:

```
продажа компьютер низкий цена 9000р компьютерный магазин предлагать 9000р купить
компьютер офис сборка заказ интернет магазин компьютер главный компания контакт
гарантия доставка способ оплата дозвониться оставить телефон
```

При объединении текстов в единый корпус и формировании списка слов для удаления, было решено выбрать слова, которые повторяются не более 5 раз. Для каждого слова было подсчитано число его повторений. Количество слов с повторами более 6 раз значительно меньше, что делает их более уникальными для контекста (рис. 2).

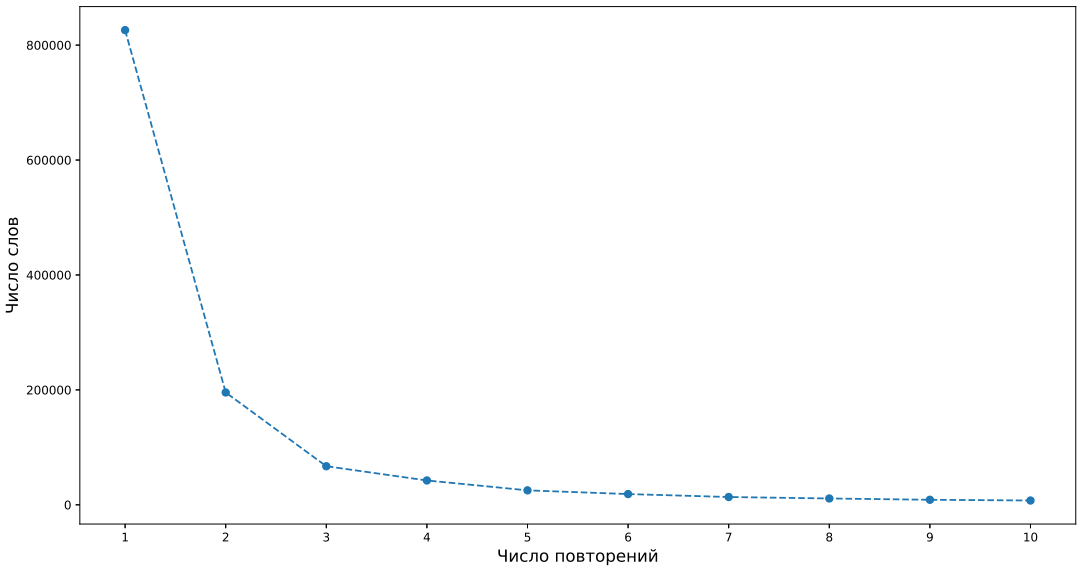

Рис. 2 График отношения количества слов и их повторений

Данная фильтрация текстов позволила заметно уменьшить число слов в корпусе (с более чем 1 миллиона до 30 тысяч), что уменьшает признаковое пространство, и увеличивает скорость работы алгоритмов.

После скачивания и обработки текстов с главных страниц данные обогатились обработанными текстами и представляют собой множество триплетов: $\langle$*домен, категория, обработанный текст*$\rangle$.

3.2 Наборы данных

Было решено создать две выборки данных. В первую выборку (*общий случай*) вошли данные из 17 корневых категорий и все их категории-потомки. Во вторую выборку (*частный случай*) вошли данные из корневой категории «Досуг» и все ее категории-потомки. Данная категория была выбрана в качестве *частного случая*, так как пополнение и переработка категории проводилась весной 2017 г., непосредственно перед началом проведения исследований описанных в данной работе.

Разделение данных на две выборки необходимо для проведения экспериментов. Частный случай содержит только одну корневую категорию и всего 47 категорий-потомков. Эксперименты, проведенные для выборки данных из частного случая, позволяют оценить качество работы методов: точность предсказания категории и время работы. На основании полученных результатов производится анализ работы методов, и отбор методов для тестирования в общем случае. Выборка в общем случае содержит более 500 категорий-потомков.

Обе выборки данных были разбиты на обучающую и тестовую, в отношении 4 к 1. Обучающая выборка используется для обучения алгоритмов классификации, а на тестовой производится проверка точности обучения. При разбиении данных случайным образом, может возникнуть проблема, когда домены из определенной категории не присутствуют в обучающей выборке. По рис. 3 можно судить о количестве категорий с малым числом доменов. Поэтому при разбиении данных на обучающую и тестовую выборки применялся

специальный метод, позволяющий гарантировать присутствие доменов из каждой категории в обучающей и тестовой выборках. При этом пришлось пожертвовать категориями, в которых присутствовало только по одному домену.

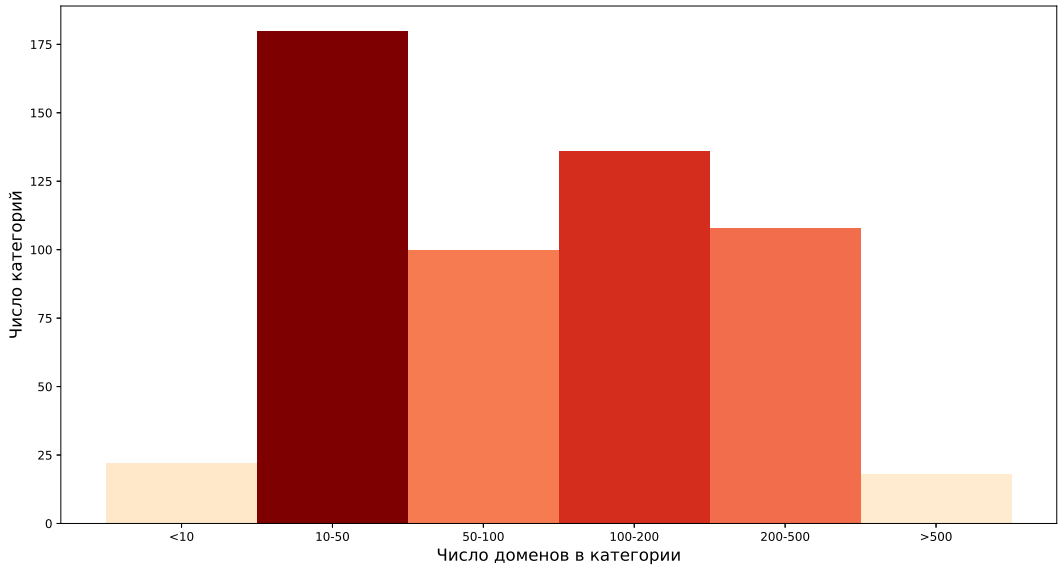

Рис. 3 Гистограмма числа категорий и количества доменов

4 Подходы к применению методов машинного обучения для автоматической тематической разметки

Для решения задачи предложены два подхода. Первый подход называется *прямым* — применяется единственный классификатор, для каждого домена предсказывается его категория, категория может быть любого уровня. Второй подход называется *многоуровневым* — применяется множество классификаторов, где каждому множеству категорий с одним родителем соответствует один классификатор. Классификаторы применяются иерархически, сначала для корневых категорий, далее для категорий второго уровня и т.д. Также используется комбинация подходов (подраздел 4.3). Для каждого из подходов производится собственная подготовка входных данных для классификации.

4.1 Прямой подход

При прямом подходе применяется единственный классификатор, для каждого домена предсказывается его категория, категория может быть любого уровня. Для прямого подхода был разработан алгоритм, целью которого является преобразование предобработанных данных (раздел 3) в выборку для дальнейшего обучения классификатора.

Алгоритм 1. *Формирование выборки — прямой подход.*
Вход: M — множество триплетов вида $\langle$*домен, категория, обработанный текст*$\rangle$,
ancestor — функция, выдающая по заданной категории v ее категорию-предок *ancestor*(v) из дерева категорий,
descendant — функция, выдающая по заданной категории v множество

descendant(v) категорий-потомков v из дерева категорий,
Root — множество корневых категорий
Выход: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$
Локальные переменные: *Urls* — множество доменов, тексты которых попали в множество O
$Urls \leftarrow \varnothing$
$O \leftarrow \varnothing$
Цикл для всех $C \in Root$
$O \leftarrow O \cup$ Алгоритм 2(C, *Urls*)
Конец цикла
Вернуть O

Алгоритм 1 предназначен для формирования выборки в прямом подходе, на вход он получает множество триплетов M вида $\langle$*домен, категория, обработанный текст*$\rangle$. Также алгоритму передается вся информация о дереве категорий: множество корневых категорий *Root*, функции *ancestor* и *descendant*, выдающие по заданной категории v категорию-предка *ancestor(v)* и множество *descendant(v)* категорий-потомков v из дерева категорий соответственно. На выход алгоритм выдает множество пар вида $\langle$*категория, обработанный текст*$\rangle$. В дальнейшем это множество используется для обучения классификатора в прямом подходе.

В *Алгоритме 1* формируется два множества, первое из них — выходное множество O, второе — множество доменов *Urls*, действительно попавших в выборку. Для формирования выходного множества *Алгоритм 1* использует *Алгоритм 2*, вызывая его для каждой корневой категории. В *Алгоритме 2* производится обход дерева категорий в глубину, начиная с полученной на вход категории.

В конечном итоге в *Алгоритме 1* объединяются все множества, полученные при помощи *Алгоритма 2* и выдаются в качестве ответа. Полученное множество пар используется для обучения классификатора, который относит входной текст с главной страницы домена к одной из категорий, вошедших в обучающее множество.

Алгоритм 2. *Формирование выборки для категории — прямой подход.*
Вход: C — категория,
Urls — множество доменов
Выход: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$
$O \leftarrow \varnothing$
Цикл для всех $\langle D, C, T\rangle \in \{\langle d, C, t\rangle | \langle d, C, t\rangle \in M\}$
Если $D \notin Urls$ то
$O \leftarrow O \cup \{\langle C, T\rangle\}$
$Urls \leftarrow Urls \cup \{D\}$
Конец если
Конец цикла
Цикл для всех $V \in$ *descendant*(C)
$O \leftarrow O \cup$ Алгоритм 2(V, *Urls*)
Конец цикла
Если $|O| \leqslant |\{\langle d, C, t\rangle | \langle d, C, t\rangle \in M\}|/5$ то
$O \leftarrow \{\langle ancestor(C), T\rangle | \langle C', T\rangle \in O\}$

Вернуть O

Алгоритм 2 предназначен для формирования обучающей выборки для категории в прямом подходе, на вход он получает категорию C и множество *Urls*. Алгоритм обрабатывает категорию C, производя обход всех триплетов которые относятся к данной категории. Если домен отсутствует в множестве *Urls*, то он добавляется в него, а в множество O добавляются текст и категория домена. Повторений доменов не допускается, для того чтобы при обучении не происходило разногласий по причине повторов одного домена несколько раз с разными категориями в обучающей выборке. Далее *Алгоритм 2* вызывается для каждой категории-потомка категории C, выход которого объединяется с множеством O. После этого проверяется отношение мощности множества O к количеству изначальных триплетов для данной категории. Если мощность множества O составляет менее 20% от числа триплетов, то для всех пар в данном множестве категория изменяется на категорию-родителя категории C. В дальнейшем категория C и все ее категории-потомки не участвуют в обучении, доменам из тестовой выборки не будут присваиваться данные категории. Это сделано для того, чтобы не порождать большое количество категорий с очень малым числом доменов, а за счет них пополнять размерность категорий лежащих на уровне выше.

Набор данных для *общего случая* содержит более 500 категорий. После преобразования данного набора данных для прямого подхода 120 категорий-потомков сливаются со своими категориями-родителями и более не участвуют в обучении.

Примерами категорий-потомков, которые сливаются со своими категориями-родителями являются категории, связанные с комплектующими и периферией компьютера «Клавиатуры, мыши, комплекты», «Процессоры», «Материнские платы»; категории, связанные с марками автомобилей «Porsche», «Lexus», «Lifan» (сливаются с категориями автомобили по странам); категории, связанные с одеждой «Штаны, джинсы и тд (для женщин)», «Верхняя одежда (для мужчин)», «Спортивная одежда и обувь». Как правило, со своими категориями-родителями сливаются те категории, которые содержат преимущественно шаблоны — регулярные выражения над доменами из категорий-родителей.

4.2 Многоуровневый подход

При многоуровневом подходе применяется множество классификаторов, где каждому множеству категорий с одним родителем соответствует один классификатор. В данном подходе формирование данных для обучения производится иерархически, и для каждого классификатора формируется своя обучающая выборка. Для многоуровневого подхода был разработан алгоритм, целью которого является преобразование предобработанных данных (раздел 3) в выборку для дальнейшего обучения классификаторов.

Алгоритм 3. *Формирование выборки — многоуровневый подход.*
Вход: M — множество триплетов вида $\langle$*домен, категория, обработанный текст*$\rangle$,
ancestor — функция, выдающая по заданной категории v ее категорию-предок $ancestor(v)$ из дерева категорий,
descendant — функция, выдающая по заданной категории v множество $descendant(v)$ категорий потомков v из дерева категорий,
Root — множество корневых категорий
Выход: S — функция, которая для категории v выдает множество $S(v)$ вида $\langle$*категория, обработанный текст*$\rangle$

Локальные переменные: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$,
$Urls$ — множество доменов, тексты которых попали в множество O
$Urls \leftarrow \varnothing$
$O \leftarrow \varnothing$
Цикл для всех $C \in Root$
$O \leftarrow O \cup$ Алгоритм 4$(C,\ Urls)$
Конец цикла
$S \leftarrow \{ROOT \mapsto O\}$
Цикл для всех $C \in Root$
Алгоритм 5$(C,\ S)$
Конец цикла
Вернуть S

Алгоритм 3 предназначен для формирования выборки в многоуровневом подходе, аналогично *Алгоритму 1* он получает на вход множество триплетов M вида $\langle$*домен, категория, обработанный текст*$\rangle$, информацию о дереве категорий: множество корневых категорий *Root*, функции *ancestor* и *descendant* выдающие по заданной категории v категорию-предка и множество категорий-потомков v из дерева категорий соответственно. Но на выход алгоритм выдает не множество, а функцию, которая для категории v выдает множество пар вида $\langle$*категория, обработанный текст*$\rangle$.

В *Алгоритме 3* формируется два множества: O - множество пар вида $\langle$*категория, обработанный текст*$\rangle$ и множество доменов *Urls*. Также в алгоритме формируется функция S.

Для произвольной категории C и сформированного для нее множества пар O, функция S обновляется (в *Алгоритме 3* и *Алгоритме 5*) путем добавления к ней пары $C \mapsto O$. Это означает что функция S для категории C выдает множество *S(C)* равное множеству O.

В *Алгоритме 3* производится формирование выборки для классификатора корневых категорий (классификатора первого уровня), так как у данных категорий отсутствует категория-родитель. Для формирования выборки каждой корневой категории используется *Алгоритм 4*, после чего все сформированные множества объединяются и добавляются в функцию S по идентификатору категории $ROOT$, который обозначает фиктивную категорию — родителя всех корневых категорий. *S(ROOT)* выдает множество пар, которые используются для обучения классификатора 1 уровня в многоуровневом подходе.

Далее для каждой корневой категории вызывается *Алгоритм 5*, чтобы сформировать выборки для обучения классификаторов следующих уровней.

Алгоритм 4. *Формирование выборки для категории — многоуровневый подход.*
Вход: C — категория,
$Urls$ — множество доменов
Выход: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$
$O \leftarrow \varnothing$
Цикл для всех $\langle D, C, T\rangle \in \{\langle d, C, t\rangle | \langle d, C, t\rangle \in M\}$
Если $D \notin Urls$ то
$O \leftarrow O \cup \{\langle C, T\rangle\}$
$Urls \leftarrow Urls \cup \{D\}$
Конец если
Конец цикла

Цикл для всех $V \in descendant(C)$
    $O \leftarrow O \cup$ Алгоритм 4$(V, Urls)$
Конец цикла
$O \leftarrow \{\langle C, T\rangle | \langle C', T\rangle \in O\}$
Вернуть O

Формирование выборки для категории в многоуровневом подходе производится в *Алгоритме 4*. *Алгоритм 4* практически полностью идентичен *Алгоритму 2*, за исключением того, что в конце алгоритма не проверяется отношение мощностей множеств, а всем парам из выходного множества присваивается обрабатываемая категория.

Данный алгоритм необходим для формирования выборки для поддерева категории, когда домены категории C объединяются со всеми доменами категорий-потомков данной категории. Всем собранным доменам в качестве категории выставляется категория C. Таким образом, полученная выборка включает все домены, входящие в поддерево, где корневой вершиной является категория C.

Данный алгоритм является вспомогательным для *Алгоритма 5*, в котором производится формирование обучающей выборки для классификатора произвольного уровня (за исключением первого).

Алгоритм 5. *Формирование выборки для категории и ее потомков — многоуровневый подход.*

Вход: C — категория,
    S — функция которая для категории v выдает множество $S(v)$ вида
    $\langle$*категория, обработанный текст*$\rangle$
Локальные переменные: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$,
    $Urls$ — множество доменов, тексты которых попали в множество O
$Urls \leftarrow \varnothing$
$O \leftarrow \varnothing$
Цикл для всех $\langle D, C, T\rangle \in \{\langle d, C, t\rangle | \langle d, C, t\rangle \in M\}$
    Если $D \notin Urls$ то
        $O \leftarrow O \cup \{\langle C, T\rangle\}$
        $Urls \leftarrow Urls \cup \{D\}$
    Конец если
Конец цикла
Цикл для всех $V \in descendant(C)$
    $O \leftarrow O \cup$ Алгоритм 4$(V, Urls)$
Конец цикла
$S \leftarrow S \cup \{C \mapsto O\}$
Цикл для всех $V \in descendant(C)$
    Алгоритм 5(V, S)
Конец цикла

Формирование выборки для категории и ее потомков в многоуровневом подходе производится в *Алгоритме 5*. В данном алгоритме производится формирование отдельных множеств пар вида $\langle$*категория, обработанный текст*$\rangle$ для категорий с одним родителем (множество O) и множества доменов $Urls$. На вход алгоритм получает категорию C и функцию S, куда будут добавляться новые значения.

Алгоритм производит обход всех триплетов, которые относятся к категории C. Если домен отсутствует в множестве *Urls*, то он добавляется в него, а в множество O добавляются текст и категория домена. Далее производится обработка всех потомков категории C, для каждого из них вызывается *Алгоритм 4*, ответ которого объединяется с множеством O. Сформированное множество пар O для категории C и всех ее потомков, добавляется в функцию S.

Таким образом, в *Алгоритме 5* производится формирование выборки для обучения классификатора, который производит классификацию среди категории C и всех ее категорий-потомков.

После этого для каждого потомка категории C вызывается *Алгоритм 5*, чтобы таким же образом сформировать обучающие выборки для классификаторов следующих уровней.

Функция S является результатом формирования обучающих выборок для множества классификаторов в многоуровневом подходе.

4.3 Комбинация многоуровневого и прямого подходов

В работе применялась также комбинация многоуровневого и прямого подходов. При комбинированном подходе используется множество классификаторов, включающее в себя классификатор корневых категорий, соответствующий классификатору первого уровня из многоуровневого подхода; и отдельные классификаторы для каждой корневой категории и ее подкатегорий (выборки для которых формируются на основании прямого подхода).

Для данного подхода формирование данных производится при помощи уже разработанных алгоритмов. Некоторое преобразования необходимо только для *Алгоритма 3*.

Алгоритм 3*. *Формирование выборки — комбинированный подход.*
Вход: M — множество триплетов вида $\langle$*домен, категория, обработанный текст*$\rangle$,
$ancestor$ — функция, выдающая по заданной категории v ее категорию-предок $ancestor(v)$ из дерева категорий,
$descendant$ — функция, выдающая по заданной категории v множество $descendant(v)$ категорий потомков v из дерева категорий,
$Root$ — множество корневых категорий
Выход: S — функция ,которая для категории v выдает множество $S(v)$ вида $\langle$*категория, обработанный текст*$\rangle$
Локальные переменные: O — множество пар вида $\langle$*категория, обработанный текст*$\rangle$,
Urls — множество доменов, тексты которых попали в множество O

```
Urls ← ∅
O ← ∅
Цикл для всех C ∈ Root
    O ← O ∪ Алгоритм 4(C, Urls)
Конец цикла
    S ← {ROOT ↦ O}
Цикл для всех C ∈ Root
    Urls ← ∅
    S ← S ∪ {C ↦Алгоритм 2(C, Urls)}
Конец цикла
Вернуть S
```

*Алгоритм 3** предназначен для формирования выборки в комбинированном подходе. Аналогично *Алгоритму 3* он получает на вход множество триплетов M вида $\langle$*домен, категория, обработанный текст*$\rangle$, информацию о дереве категорий: множество корневых категорий *Root*, функции *ancestor* и *descendant* выдающие по заданной категории v категорию-предка и множество категорий-потомков v из дерева категорий соответственно. На выход алгоритм выдает функцию, которая для категории v выдает множество пар вида $\langle$*категория, обработанный текст*$\rangle$.

По сравнению с *Алгоритмом 3*, *Алгоритм 3** содержит следующие изменения: была добавлена реинициализация множества *Urls* пустым множеством, так как для формирования каждой отдельной выборки требуется создавать свое собственное множество *Urls*; вызов *Алгоритма 5* был заменен на вызов *Алгоритма 2*; а результат работы *Алгоритма 2* добавляется в функцию S.

Как и в *Алгоритме 3*, в *Алгоритме 3** производится формирование выборки для корневых категорий, но вместо выборок для каждого уровня формируются выборки для поддеревьев корневых вершин.

5 Признаки и классификаторы

5.1 Признаки

Большинство современных алгоритмов машинного обучения работают с признаковыми описаниями объектов, поэтому обычно все документы переводятся в многомерное вещественное признаковое пространство. Каждое слово или группа слов при этом приобретают некоторое численное значение.

Одной из моделей перевода текста в векторное пространство является *мешок слов (Bag of Words)* [14]. В данной модели порядок слов в тексте не важен, и все документы представляются матрицей $T = (T)_{(d,w)}$, где строка соответствует отдельному документу или тексту, а столбец определенному слову. Элемент $t_{(d,w)}$ матрицы соответствует количеству вхождений слова w в документ d.

Метод мешка слов может быть скомбинирован со статистической мерой TF-IDF [6], используемой для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. При этом элементу матрицы $t_{(d,w)}$ соответствует значение функции $TF\_IDF(w, d, D)$ [6] (вес слова), где D — количество документов в корпусе. Вес слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции.

В текстах часто информацию несут не только отдельные слова, но и некоторые последовательности слов (контексты). Поэтому для того чтобы учесть особенности языка предлагается учитывать кроме отдельных слов, последовательности слов — N-граммы [7]. В модели перевода текста в векторное пространство *мешок N-грамм*, скомбинированной с мерой TF-IDF, все документы также представляются в виде матрицы, где для каждого документа вектор признаков состоит из значений функции TF_IDF для слов и всех последовательностей из N слов.

В данной работе применялись модели мешок слов и мешок N-грамм в сочетании с мерой TF-IDF. Рассматривались два случая N-грамм: N-граммы только длины 2 и N-граммы длин 1 и 2.

Применялся также метод перевода текста в векторное пространство *Word2vec*. Метод использовался как способ снижения признакового пространства: для каждого слова из корпуса находится его векторное представление в соответствии с *Word2vec*, после чего к векторам слов применяется метод кластеризации K-Means. Слова разделяются на

некоторое количество кластеров. Далее каждое слово заменяется соответствующим ему кластером. В новом виде тексты состоят уже не из слов, а из номеров кластеров слов. И к новым текстам применялся выше описанный метод *мешка слов*. Число кластеров напрямую зависит от числа категорий и различности их тематического наполнения.

5.2 Классификаторы

В качестве классификаторов в задаче анализа текстов обычно используют линейные классификаторы, применяются также классификаторы, основанные на вычисление расстояний между объектами. Признаковое пространство в рассматриваемой задаче имеет большую размерность за счет большого количества слов во всей выборке, и является сильно разреженным. Для *общего случая* величина признакового пространства составляет около 160 тыс., с максимальным количеством ненулевых компонент в векторах выборки около 7 тыс. и средним количеством ненулевых компонент по всем доменам около 350. Для *частного случая* всего признаковое пространство имеет 82 тыс. компонент, с максимальным количеством ненулевых компонент около 4 тысяч и средним числом ненулевых компонент около 320. На основании анализа решаемой задачи и родственных работ для экспериментов были выбраны следующие методы классификации:

- логистическая регрессия [8, 10] — метод линейной классификации, позволяющий оценить апостериорные вероятности принадлежности объектов классам; при этом апостериорное распределение вероятностей основано на логистической (сигмоидной) функции;
- гребневая регрессия (классификация) [10] — метод линейной классификации, основанный на построении разделяющей поверхности с регуляризацией квадрата нормы решения;
- ближайший центроид [3] — алгоритм классификации, где каждый класс из обучающей выборки представлен своим центром, а объект классифицируется по классу ближайшего центра;
- метод опорных векторов (support vector machine — SVM) [9] — метод линейной классификации, основная идея которого заключается в поиске разделяющей гиперплоскости с максимальным зазором.

Все выбранные классификаторы способны работать с сильно разреженными признаковыми пространствами большой размерности.

Также для сравнительного анализа использовалась библиотека классификации текстов *fastText*, разработанная компанией Facebook. В данной библиотеке используется подход *Word2vec*, при помощи которого для слов формируются характеризующие их низкоразмерные вектора. Текст представляется в виде вектора низкой размерности, который получается путем суммирования векторов, соответствующих словам, входящих в текст. Для классификации текстов используется иерархический классификатор *Softmax* [15].

6 Программная реализация

Для реализации предложенных подходов был реализован программный комплекс, архитектура которого изображена на рис. 4.

Программный комплекс был реализован на языке программирования Python.

Данные, полученные из системы поддержки тематической разметки интернет-доменов, передаются в модуль *предварительной обработки данных*. В модуле из полученных данных выделяется множество доменов, главные страницы которых скачиваются при помощи библиотеки *Requests* (библиотека для выполнения HTTP-запросов). Далее данные

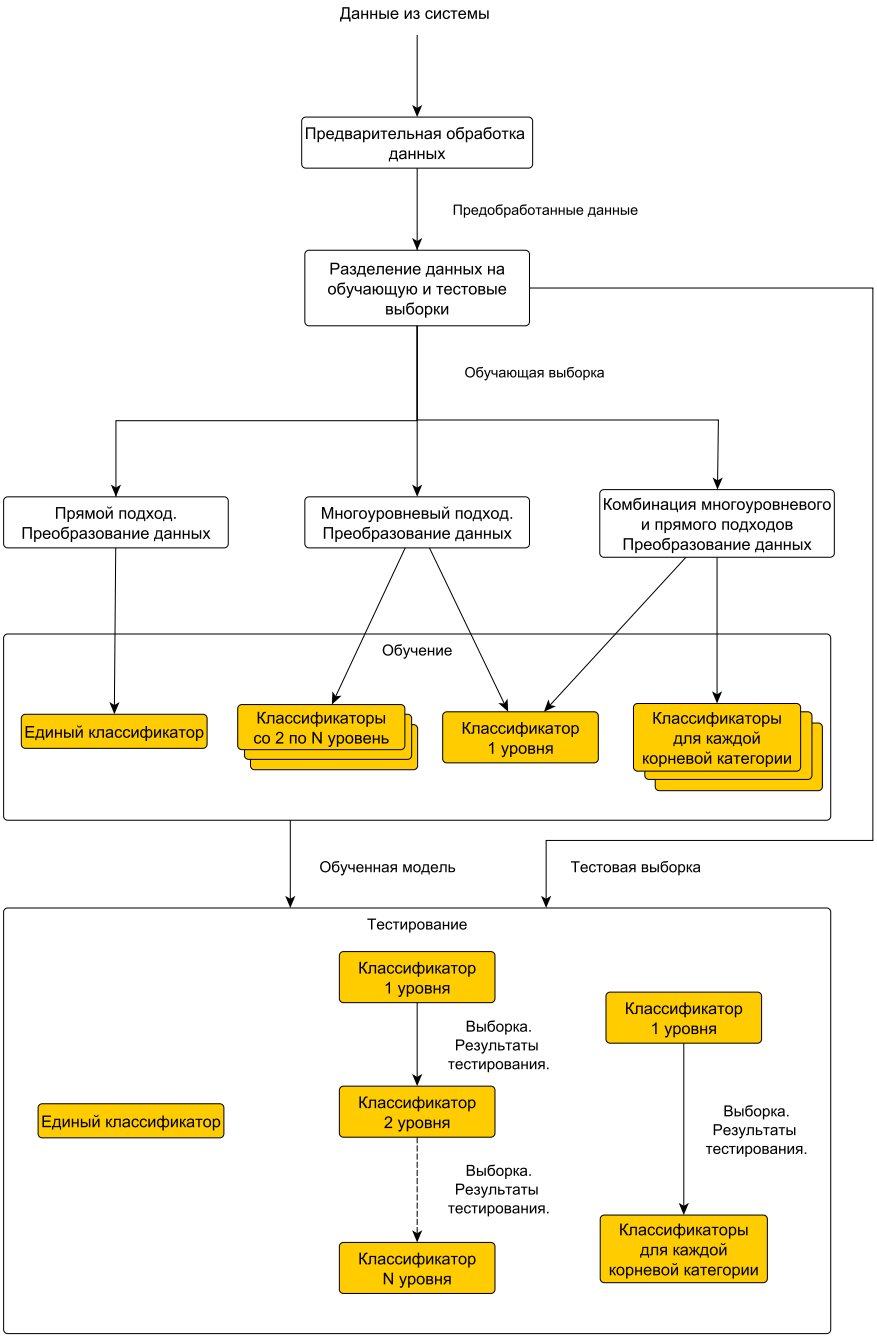

Рис. 4 Архитектура программного комплекса

проходят процедуру предобработки, очищаются от мусора и ненужной информации (раздел 3). Для этого используются библиотеки *BeautifulSoup* (библиотека, содержащая парсер для синтаксического разбора файлов HTML/XML) и библиотека *pymorphy2* (библиотека, представляющая собой морфологический анализатор). Для передачи предобработанных данных, все данные упаковываются в матрицы библиотеки *numpy* (библиотека, обеспечивающая поддержку больших многомерных массивов и матриц). В отдельном модуле данные разделяются на обучающую и тестовую выборки, для этого используется функция разделения данных из библиотеки *scikit-learn* (библиотека, содержащая наборы методов

Таблица 1 Результаты работы классификаторов (частный случай)

	Мешок слов	Мешок N-грамм	
		N = (1, 2)	N =2
Логистическая регрессия	0.7114 0.7721	0.6894 0.7474	0.4836 0.5123
Гребневая регрессия	0.7478 0.8209	0.7474 0.8137	0.6846 0.7450
Ближайший центроид	0.6478 0.7809	0.6882 0.8353	0.5783 0.7370
SVM, gamma = 0.5	0.7098 0.7913	0.6734 0.7498	0.4248 0.4516
fastText	0.7042 0.7913		

и функций машинного обучения). Обучающая выборка передается в три модуля, соответствующие предлагаемым подходам (подраздел 3.2). В каждом модуле обучающая выборка преобразуется соответствующим образом (раздел 4). Перевод текстов в многомерное вещественное признаковое пространство осуществляется при помощи библиотеки *scikit-learn*, которая содержит необходимые функции преобразования признаковых пространств (подраздел 5.1) Преобразованная выборка передается в *модуль обучения*.

Для прямого подхода обучается один единственный классификатор (подраздел 4.1). Для многоуровневого и комбинированного подходов обучается классификатор первого уровня дерева категорий. Отдельно для многоуровневого подхода обучаются классификаторы для каждого из уровней дерева категорий; для комбинированного подхода обучаются классификаторы для каждой корневой категории (подраздел 4.2).

Реализованные алгоритмы машинного обучения также используются из библиотеки *scikit-learn*. В модуль *тестирование* передаются обученные модели и тестовая выборка. Тестовая выборка передается подается на вход обученной модели (или иерархически — нескольким моделям, в зависимости от подхода). В качестве ответа модель выдает категорию-ответ для каждого домена из тестовой выборки.

7 Эксперименты

7.1 Проверка качества

Были выбраны два подхода проверки качества. В первом подходе домен считается классифицированным верно, если предсказанная категория совпадает с категорией из тестовых данных. Во втором подходе домен считается классифицированным верно, если предсказанная категория совпадает с категорией из тестовых данных, либо с ее категорией-родителем или с одной из категорий-потомков (семантически наиболее близкими категориями).

7.2 Результаты экспериментов

Для набора данных — частного случая (категории «Досуг») проводилось тестирование только прямого подхода, так как многоуровневый комбинированный подход рассчитан на данные с большим числом категорий и доменов.

В таблицах 1 и 2 приведены результаты тестирования классификаторов и признаков, а также время их работы для частного случая. В таблице 1 в каждой ячейке приведено по

Таблица 2 Время работы классификаторов (частный случай), сек.

	Мешок слов	Мешок N-грамм	
		N = (1, 2)	N =2
Логистическая регрессия	16.11	141.06	104.11
Гребневая регрессия	28.66	149.50	125.45
Ближайший центроид	0.13	1.21	01.01
SVM, gamma = 0.5	138.58	214.25	233.83
fastText	6.9		

Таблица 3 Результаты использования *Word2vec* (частный случай)

Число кластеров	500	1000	2000	3000	4000
Гребневая регрессия	0.7178	0.7338	0.7478	0.7434	0.7402
	0.7929	0.8049	0.8209	0.8193	0.8129

два числа: первое результат проверки качества первым подходом, второе результат проверки качества вторым подходом (подраздел 7.1). Анализ результатов позволяет сделать несколько выводов:

– вне зависимости от признаков, лучший результат показала гребневая регрессия;
– настройка параметров позволяет значительно улучшить результат;
– использование N-грамм практически не улучшает результаты, полученные с применением метода мешка слов, но при этом классификация занимает больше времени.

Преимущество метода мешка слов над методом N-грамм состоит в том, что в первом случае получаются признаковые пространства меньшей размерности, благодаря чему обучение классификаторов производится значительно быстрее. Использование N-грамм практически не улучшило результаты, из-за чего можно предположить, что контекст не сильно влияет на классификацию, а больший вес имеют ключевые слова. Также проверялось влияние использования N-грамм длины 3 на результаты работы методов, при этом никаких улучшений качества методов практически не произошло, так как почти все триграммы были уникальными. Ввиду полученных результатов для частного случая, метод N-грамм не применялся для набора данных — общего случая.

Результаты, полученные при помощи метода *fastText* немного уступают лучшим результатам, полученным при помощи гребневой регрессии и мешка слов.

Для частного случая применялся метод *Word2vec* для снижения признакового пространства. Применение данных признаков показало улучшение результатов классификации от 0.5% до 3%. Для гребневой регрессии полученные результаты приведены в таблице 3. Проводилось исследование метода при разделении слов на различное число кластеров, для данной выборки данных наилучшим показало себя разделение на 2000 кластеров. По сравнению с другими методами классификации обучение метода *Word2vec* и дальнейшая кластеризация полученных слов занимает слишком много времени.

В таблицах 4 и 5 приведены результаты тестирования классификаторов и время их работы набора данных — общего случая. В таблице 4 в каждой ячейке приведено по два числа: первое результат проверки качества первым подходом, второе результат проверки качества вторым подходом (подраздел 7.1). Анализ результатов позволил сделать следующие выводы:

– как и в частном случае, лучший результат показала гребневая регрессия;

Таблица 4 Результаты работы классификаторов (общий случай)

	Мешок слов		
	Прямой подход	Многоуровневый подход	Комбинация подходов
Логистическая регрессия	0.6077 0.7007	0.5216 0.6180	0.5868 0.6851
Гребневая регрессия	0.6364 0.7225	0.5400 0.6630	0.6293 0.7230
Ближайший центроид	0.5517 0.6610	0.4511 0.5732	0.5268 0.6315
SVM, gamma = 0.5	0.5923 0.6873	0.5126 0.6322	0.5804 0.6838
fastText	0.5932 0.7013		

Таблица 5 Время работы классификаторов (общий случай), сек.

	Мешок слов		
	Прямой подход	Многоуровневый подход	Комбинация подходов
Логистическая регрессия	1312	91.61	133.15
Гребневая регрессия	1608	133.84	188.6
Ближайший центроид	1.45	2.15	1.82
SVM, gamma = 0.5	4889	2038	2080
fastText	55.19		

– прямой подход показывает результаты лучше, чем многоуровневый;
– комбинированный подход показывает результаты, практически идентичные прямому подходу;
– время работы при комбинированном подходе намного меньше, чем при прямом подходе.

Комбинированный подход обладает рядом преимуществ перед прямым подходом: меньшее время работы; меньшие затраты по памяти; гибкость: для каждой корневой категории обучается отдельный классификатор, и настройку каждого классификатора можно производить отдельно.

Сравнение результатов работы методов при полностью случайном разбиении данных на обучающую и тестовую выборки и при помощи специального метода (раздел 3.2) показывает, что полностью случайное разбиение для первого подхода проверки качества выдает результаты в среднем на 1-2% хуже для частного случая и 0.5-1.5% для общего случая. Разница в случае второго подхода проверки качества незначительна.

Применение специального метода разбиения данных позволяет уменьшать ошибку предсказания категории родителя и категории потомка, но никак не влияет на другие ошибки.

Также стоит отметить что после применения классификатора первого уровня в многоуровневом и комбинации подходов его точность не превышает 85%, что говорит о то что окончательная точность не будет превышать данного показателя.

Таблица 6 Результаты использования *Word2vec* (общий случай)

	TF-IDF	w2v500	w2v 1000	w2v 2000	w2v 3000	w2v 4000
Компьютеры	0.6190 0.7143	0.6243 0.7196	0.6349 0.7196	0.6138 0.7090	0.6032 0.6984	0.6138 0.7037
Красота и уход за собой	0.7152 0.7848	0.7152 0.7848	0.7053 0.7815	0.7086 0.7848	0.7185 0.7848	0.7219 0.7914
Ремонт	0.6920 0.8033	0.6838 0.7939	0.6909 0.8056	0.6838 0.8033	0.6932 0.8009	0.6803 0.7951
Мобильные устройства	0.6301 0.6795	0.6192 0.6658	0.6274 0.6712	0.6274 0.6767	0.6274 0.6795	0.6219 0.6740
Игровые интересы	0.7119 0.7246	0.6419 0.6568	0.6822 0.6949	0.6949 0.7076	0.7076 0.7182	0.7055 0.7161
Недвижимость	0.4968 0.7238	0.4968 0.7516	0.4989 0.7109	0.4818 0.7238	0.4925 0.7281	0.4925 0.7409
Финансовые услуги и страхование	0.5492 0.7220	0.5322 0.7254	0.5559 0.7424	0.5593 0.7322	0.5559 0.7153	0.5525 0.7220
Туризм по странам	0.5634 0.6690	0.5423 0.6667	0.5587 0.6667	0.5798 0.6808	0.5610 0.6714	0.5704 0.6667
Одежда	0.5762 0.6952	0.5613 0.6543	0.5539 0.6543	0.5725 0.6766	0.5762 0.6766	0.5762 0.6840

Как и частном случае, применение метода *fastText* показало немного уступающие по точности результаты по сравнению с лучшими результатами, полученными с помощью гребневой регрессии и прямого подхода. Но метод *fastText* значительно выигрывает по памяти и времени, потратив на обучение и тестирование не более минуты секунд.

Для общего случая также применялся метод *Word2vec* для снижения признакового пространства, но только для комбинации подходов. Для каждой корневой категории слова как и в частном случае разделялись на 2000 кластеров. В среднем это дало 1% прироста к результатам. Было решено провести исследование зависимости точность классификации от числа кластеров, для различных классификаторов корневых категорий. В таблице 6 приведены результаты классификации при разделении на различное число кластеров. Для каждой категории приведено число входящих в нее подкатегорий. Из таблицы можно заметить что оптимальное число кластеров различно для разных корневых категорий.

Метод *Word2vec* для снижения признакового пространства работает в отдельных случаях. Для одних категорий, таких как «Компьютеры», «Ремонт» и «Красота и уход за собой», при настройке количества кластеров можно добиться некоторого улучшения качества. Для таких категорий как «Мобильные устройства» и «Игровые интересы» снижение размерности при помощи метода *Word2vec* не дает каких либо улучшений в качестве классификации.

Также по результатам из таблицы 6 можно выделить несколько категорий для которых плохо отрабатывают автоматические методы классификации. К таким категориям относятся «Недвижимость», «Финансовые услуги и страхование», «Одежда» и «Туризм по странам». Поддеревья данных корневых категорий в основном состоят из очень близких по тематике категорий, наполненных по большей части шаблонами одних и тех же доменов. Точность классификации для этих категории примерно равна 50-55%, что явля-

ется довольно низким результатом. Поэтому наиболее правильным решением для данных категорий будет отказаться от автоматической классификации доменов в категории поддеревьев данных корневых категорий.

7.3 Анализ ошибок классификации

Был проведен анализ ошибок классификации и выделены следующие группы ошибок:

- предсказание категории родителя;
- предсказание категории потомка;
- предсказание категории со схожей тематикой;
- неверная разметка данных;
- установленный по умолчанию не русский язык на главной странице домена.

В качестве примеров доменов и категорий, для из которых совершаются ошибки предсказания категории родителя или потомка можно привести следующие: для большей половины доменов из категории «Сценическое, студийное и профессиональное музыкальное оборудование» была предсказана ее категория потомок «Музыкальные инструменты», это такие домены как `spb.music-expert.ru`, `spb.muzzshop.ru`. Для доменов категории «Работа за границей» предсказана родительская категория «Эмиграция»: например, ошибочно были классифицированны домены `westwork.org.ua`, `jobwest.com.ua`, кроме работы за границей также содержащие информацию касательно эмиграции.

В качестве примеров ошибок предсказания схожих по тематике категории можно привести следующие:

- домены из категории «Велоспорт» были классифицированы в категорию «Велосипеды», ошибочно классифицированные домены `velohack.com`, `cycleinfo.net`. Домены `velostok.com`, `bikemotive.com.ua` должны относится к категории «Велосипеды», так как они являются магазинами;
- домены из категории «Android» (игры на Android) были классифицированы в категорию «ПО и игры для мобильных устройств»;
- доменам из категории «Всё для спорта и туризма» и доменам ее потомков часто предсказывается категория «Активный отдых»;
- очень сильное пересечение категорий потомков наблюдается в родительских категориях «Недвижимость», «Красота и уход за собой».

Примеры неверной разметки: домен `lp.videozayac.ru` находившийся в категории «Сценическое, студийное и профессиональное музыкальное оборудование» является сайтом студии видеодизайна, и никак не может относится к данной категории.

В случае, когда у главной страницы домена по умолчанию стоит не русский язык, то для него сформированное признаковое пространство не имеет сходств с доменами из той же тематики.

Хорошая классификация наблюдается в категориях, заведомо содержащих какие-либо ключевые слова, к таким категориям относятся «Авто по маркам», специализированные категории компьютерных игр по их названиям, «Книги», «Астрология», «Знакомства» и т.д.

8 Заключение

В работе была рассмотрена задача автоматизации тематической разметки интернет-доменов при помощи методов машинного обучения. Было проведен анализ родственных работ по тематике автоматизации тематической разметки интернет-доменов и текстов, методов решения данных задач и результатов их работы.

Были предложены новые подходы по решению задачи автоматизации тематической разметки интернет-доменов: прямой и многоуровневый. При *прямом подходе* применяется единственный классификатор, для каждого домена предсказывающий его категорию, категория может быть любого уровня. При *многоуровневом подходе* применяется множество классификаторов, где каждому множеству категорий с одним родителем соответствует один классификатор. Классификаторы применяются иерархически, сначала для корневых категорий, далее для категорий второго уровня и т.д. Также используется комбинированный подход, где применяются классификаторы из двух вышеупомянутых подходов.

Проведены эксперименты для оценки работы предложенных подходов, проверялась работа различных классификаторов для разных признаковых пространств.

Среди предложенных подходов наиболее перспективным и эффективным был признан комбинированный подход, сочетающий в себе элементы из многоуровневого и прямого подходов. У данного подхода есть ряд плюсов, точность данного подхода превосходит точность работы многоуровневого подхода и не уступает точности прямого подхода. Комбинированный подход менее ресурсозатратный по памяти и по времени. Также комбинированный подход обладает большей гибкостью, так как для каждой корневой категории классификатор обучается отдельно, то и настройку каждого классификатора может производится различная.

В дальнейшем планируется провести ряд исследований улучшающих работу выбранного подхода, за счет увеличения признакового пространства путем использования информации из имеющихся открытых интернет-каталогов, содержащих информацию о доменах и их тематической принадлежности. Также планируется использовать информацию о шаблонах, благодаря которой для одного домена можно будет использовать не только главную страницу, но и другие страницы, содержащие полезную информацию.

Литература

[1] *Saleh A. I., Al Rahmawy M. F., Abulwafa A. E.* A semantic based Web page classification strategy using multi-layered domain ontology World Wide Web, 2017.

[2] *Shen D., Chen Z., Yang Q., Zeng H. J., Zhang B., Lu Y., Ma W. Y.* Web-page classification through summarization ACM Press, In the Proceedings of the 27th annual international ACM SIGIR 04, conference on. Re-search and Development in Information Retrieval, New York, 2004.

[3] *Qi X., Davison B. D.* Web page classification: Features and algorithms ACM Comput. Surv, 2009.

[4] *Sun A., Lim E. P., Ng W. K.* Web classification using support vector machine ACM Press, Proceedings of the 4th International Workshop on Web Information and Data Management, New York, 2004.

[5] *Meshkizadeh S., Rahmani A. M., Dezfuli M. A.* Webpage Classification based on Compound of Using HTML Features & URL Features and Features of Sibling Pages International Journal of Advancements in Computing Technology, 2010.

[6] *Jones K. S.* A statistical interpretation of term specificity and its application in retrieval Journal of Documentation, 1972.

[7] *Damashek M.* Gauging similarity with n-grams: Language-independent categorization of text Science, New Series, 1995.

[8] *Zhang X., Zhao J., LeCun Y.* Character-level convolutional networks for text classification In Advances in Neural Information Processing Systems, 2015.

[9] *Colas F., Brazdil P.* Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks IFIP International Federation for Information Processing, 2006.

[10] *Zhang T., Oles F. J.* Text Categorization Based on Regularized Linear Classification Methods Information Retrieval, 2001.

[11] *Kim Y.* Convolutional neural networks for sentence classification IEMNLP, 2014.

[12] *Han EH., Karypis G., Kumar V.* Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification PAKDD, 2001.

[13] Библиотека pymorphy2 URL: `https://pymorphy2.readthedocs.io`

[14] *Harris Z.* Distributional structure Word, 1954.

[15] *Joulin A., Grave E., Bojanowski P., Mikolov T.* Bag of Tricks for Efficient Text Classification arXiv preprint arXiv:1607.01759, 2016.

Поступила в редакцию 21.12.2018

Application of machine learning methods for subject classification of the internet domains

***A. T. Tleubaev*[1,2], *S. A. Stupnikov*[1,2,3]**

a.tleubayev@corp.mail.ru, sstupnikov@ipiran.ru

[1]Lomonosov Moscow State University, Moscow, Leninskie Gory 1;

[2]Mail.ru, Moscow, Leningradsky prospekt 39;

[3]Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences

The paper is devoted to the application of machine learning methods for the automation Subject classification of the Internet- domains. The specific task is to automatically assign the Internet- domain to a category from a predefined hierarchical category tree. Various classifiers were used in the work, they proved themselves well in the work with strongly discharged feature spaces of large dimension. The characteristic spaces were formed on the basis of texts from the main pages of domains using the TF-IDF and N-gram approaches. Two approaches to the application of classification methods for solving the problem are developed: direct and multilevel. With a direct approach, a single classifier is used, for each domain its category is predicted, the category can be of any level in the category tree. At the multilevel approach the set of classifiers is applied: to each set of categories with one parent there corresponds the separate classifier. Classifiers are applied hierarchically — from root to leaf categories. A combination of the proposed approaches is also used. One of the practical applications of the work is user profiling based on the sites visited by them and further offering personalized advertising.

Keywords: *subject classification of the internet domains, machine learning, text classification, web page classification*

DOI: 10.21469/22233792.4.3.05

References

[1] Saleh A. I., Al Rahmawy M. F., Abulwafa A. E. *A semantic based Web page classification strategy using multi-layered domain ontology World Wide Web*, 2017.

[2] Shen D., Chen Z., Yang Q., Zeng H. J., Zhang B., Lu Y., Ma W. Y. *Web-page classification through summarization ACM Press*, In the Proceedings of the 27th annual international ACM SIGIR 04, conference on. Re-search and Development in Information Retrieval, New York, 2004.

[3] Qi X., Davison B. D. *Web page classification: Features and algorithms ACM Comput. Surv*, 2009.

[4] Sun A., Lim E. P., Ng W. K. *Web classification using support vector machine ACM Press*, Proceedings of the 4th International Workshop on Web Information and Data Management, New York, 2004.

[5] Meshkizadeh S., Rahmani A. M., Dezfuli M. A. *Webpage Classification based on Compound of Using HTML Features & URL Features and Features of Sibling Pages International Journal of Advancements in Computing Technology*, 2010.

[6] Jones K. S. *A statistical interpretation of term specificity and its application in retrieval Journal of Documentation*, 1972.

[7] Damashek M. *Gauging similarity with n-grams: Language-independent categorization of text Science, New Series*, 1995.

[8] Zhang X., Zhao J., LeCun Y. *Character-level convolutional networks for text classification In Advances in Neural Information Processing Systems*, 2015.

[9] Colas F., Brazdil P. *Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks IFIP International Federation for Information Processing*, 2006.

[10] Zhang T., Oles F. J. *Text Categorization Based on Regularized Linear Classification Methods Information Retrieval*, 2001.

[11] Kim Y. *Convolutional neural networks for sentence classification IEMNLP*, 2014.

[12] Han EH., Karypis G., Kumar V. *Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification PAKDD*, 2001.

[13] Library pymorphy2 URL: `https://pymorphy2.readthedocs.io`

[14] Harris Z. *Distributional structure Word*, 1954.

[15] Joulin A., Grave E., Bojanowski P., Mikolov T. *Bag of Tricks for Efficient Text Classification arXiv preprint arXiv:1607.01759*, 2016.

Received December 21, 2018