УДК 681.3

А. М. Андреев, Д. В. Березкин,
В. В. Сюзев, В. И. Шабанов

МОДЕЛИ И МЕТОДЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВЫХ ДОКУМЕНТОВ

Рассмотрена задача выделения терминов из текста и определения их значимости для программ обработки текстовой информации (поиска, классификации, квазиреферирования, кластеризации). Проанализированы возможные варианты решения задачи, для каждого из вариантов разработаны алгоритмы и соответствующее программное обеспечение. На примере программы автоматической классификации проведено экспериментальное сравнение методов. Наилучшие результаты получены методом приближенного выделения словосочетаний, основанным на статистической информации.

В последние годы все бóльшее внимание уделяется машинной обработке текстов на естественном языке. Это связано, во-первых, с переходом от решения чисто вычислительных и деловых задач к решению проблем коммуникации, поиска и переработки текстовой информации, а во-вторых, с доступностью компьютерных технологий обычным пользователям, чаще всего имеющим дело именно с текстами и документами. Все более значительное число пользователей использует компьютеры для создания, обработки, передачи или публикации текстов различных видов.

С каждым годом увеличивается объем доступной пользователю текстовой информации, причем обрабатывать такой объем данных вручную уже невозможно. Поэтому становится все более актуальной задача автоматического поиска и обработки информации.

Для всех видов программ, выполняющих смысловую обработку текстов, возникает задача выделения терминов, причем от качества ее решения напрямую зависит общая эффективность программы.

Одним из распространенных видов программ, упрощающих анализ и обработку текстовых документов, являются программы классификации (классификаторы). Они позволяют выбрать в массиве текстовой информации группу документов заданного типа (рекламное сообщение, научная статья и т.д.) или заданной тематики (математика, физика и т.д.). Далее документы группы можно просмотреть, найти в них фрагменты, содержащие заданные пользователем слова или выражения.

Классификация вручную. Классификатор обычно представляет собой множество рубрик, объединенных в иерархию (рубрикатор) (рис. 1). Каждой рубрике приписываются соответствующие ее тематике доку-

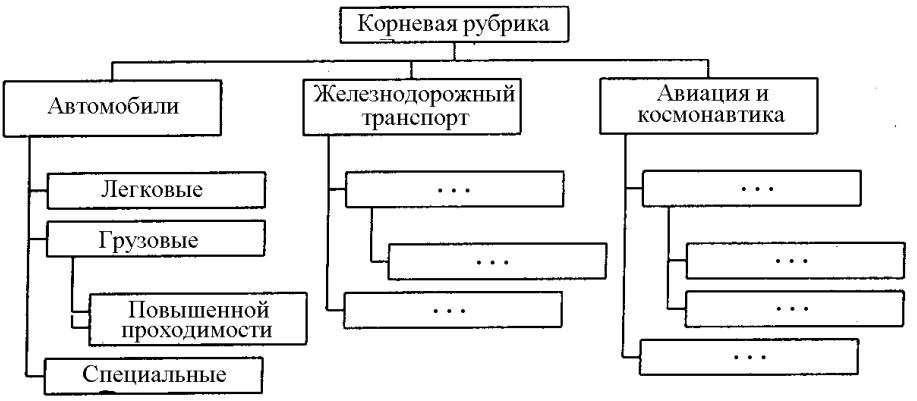


Рис. 1. Пример рубрикатора

менты. В типовом случае иерархия рубрик является деревом (т.е. не содержит циклов), однако возможны ситуации, когда некоторые рубрики являются дочерними сразу для нескольких родительских рубрик (например, новости математики можно поместить в рубрики "Математика" и "Новости науки").

У классификационного поиска имеется один существенный недостаток — документы, как правило, приходится классифицировать вручную. Другими словами, при добавлении в массив нового документа сначала необходимо его проанализировать и определить, к каким рубрикам классификатора он относится ("Микропроцессорные системы", "Сотрудничество компьютерных фирм", "Изобразительное искусство средневековья" и т.д.). И только после этого документ становится доступным для поиска по классификатору.

Понятно, что при небольшом штате технических специалистов или большом потоке входных документов применение классификации вручную становится нереальным. Более того, обеспечить полноту классификации вручную большого объема документов оказывается очень сложно, даже при помощи большого количества специалистов. При классификации вручную часто возникают ошибки, состоящие в том, что документ, соответствующий сразу нескольким рубрикам, приписывается только части из них. Количество ошибок обычно пропорционально размерности рубрикатора.

Перечень рубрик при классификации вручную очень трудно изменять, так как любое изменение требует просмотра и анализа всех документов (например, выделение в рубрике "История России" подрубрик "История СССР" и "История Древней Руси" требует анализа всех документов рубрики "История России").

Также следует отметить, что ошибки классификации вручную непрерывно накапливаются и со временем усиливается потребность в полном пересмотре распределения документов по рубрикам.

Автоматическая классификация. Для решения указанных проблем используют программы классификации, которые автоматически выполняют отнесение документов к рубрикам. Для каждой рубрики такие программы хранят множества признаков, используя которые, можно принять решение о том, соответствует ли очередной документ рубрике или нет.

Множества признаков рубрики в тематическом рубрикаторе часто называют *семантическими образами*. Семантический образ рубрики (тематики) обычно представляет собой список терминов (слов или словосочетаний), каждому из которых поставлен в соответствие вес — некоторое число, характеризующее значимость термина для данной рубрики [1, 2]. Например, семантический образ рубрики "Футбол" может содержать словосочетание "угловой удар" с весом 0,7 и слово "пенальти" с весом 0,9. Это означает, что документ, содержащий слово "пенальти" относится к данной рубрике с бóльшей вероятностью, чем документ, содержащий только словосочетание "угловой удар". Конечно, если в документе встретились оба указанных термина, то вероятность оказывается еще бóльшей.

Подробное обсуждение того, что такое термин с точки зрения программы автоматической обработки текстов, приведено далее.

Чаще всего семантические образы рубрик составляет пользователь-эксперт. Однако наиболее совершенные программы могут решать задачу автоматического обучения (распознавания образов), при которой эксперт приписывает к каждой рубрике множество эталонных документов, а программа самостоятельно выполняет их анализ и строит семантические образы.

Для того чтобы обеспечивать необходимое качество работы, в таких программах необходимо использовать сложные математические и лингвистические алгоритмы. Некоторые элементы таких алгоритмов рассмотрены в настоящей работе.

Использование программных средств автоматической классификации позволяет получить совершенно новые качества систем обработки документов — динамичность и масштабируемость. Действительно, если программа способна обработать десятки или сотни мегабайт текстовой информации за несколько часов, появляется возможность быстро вносить изменения в иерархию рубрик, а также строить системы, обрабатывающие большие потоки текстов в режиме реального времени.

Кроме того, использование автоматических классификаторов позволяет повысить количество рубрик до сотен и даже тысяч и допустить отнесение документа сразу к нескольким рубрикам, что практически невозможно в случае обработки вручную.

Использование развитых программных систем классификации позволяет не только качественно структурировать уже накопленную информацию, но и получать новые знания. Например, с помощью компьютерного анализа статей центральных газет можно сделать очень интересные выводы о наличии скрытых связей в политических и общественных кругах и т.п.

Сценарий использования программы. На рис. 2 показана последовательность операций, которые необходимо выполнить для того, чтобы классифицировать массив документов.

Сначала эксперты составляют тематическое дерево рубрик и заносят его в программу. Затем из массива документов выбирается некоторая часть, которая классифицируется вручную, в результате чего к рубрикам приписываются эталонные документы. Дерево рубрик вместе с приписанными к ним эталонными документами называется *обучающей выборкой*. Затем запускается процедура обучения классификатора, которая формирует семантические образы каждой из рубрик.

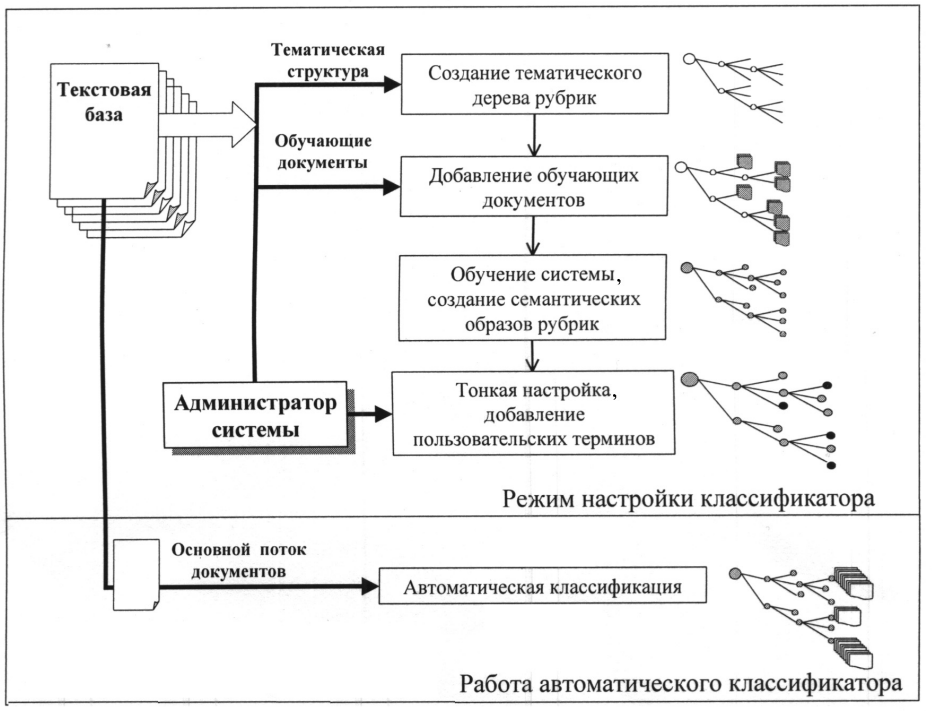


Рис. 2. Схема работы классификатора

После этого программа классификации готова к работе, однако иногда могут потребоваться коррекция и тонкая настройка семантических образов рубрик.

Автоматическая классификация. В задаче классификации текстовых документов объектами являются текстовые документы, а классами — рубрики (тематики), к которым эти документы относятся. Рубрики организованы иерархически, т.е. рубрика "Автомобили" может содержать рубрики с документами более узкой тематики, например "Легковые автомобили", "Грузовые автомобили". Глубина вложенности иерархии не ограничивается, в последней из перечисленных рубрик может иметься подрубрика "Грузовые автомобили повышенной проходимости" (см. рис. 1).

Особенности задачи классификации текстов. Задача классификации текстовых документов имеет две особенности, которые отличают ее от классических задач распознавания, в которых классы объектов, как правило, не пересекаются.

Во-первых, документы, принадлежащие рубрике, косвенно принадлежат также всей цепочке родительских рубрик иерархии. Например, документ, относящийся к рубрике "Легковые автомобили", неявно относится к более общей рубрике "Автомобили".

Во-вторых, объекты (текстовые документы) могут быть одновременно отнесены к нескольким классам, расположенным в разных местах иерархии. Например, документ, описывающий особенности технологии сварки автомобильных кузовов, может быть одновременно отнесен к рубрикам "Легковые автомобили" и "Технологии машиностроения — сварка".

К программам, выполняющим классификацию, предъявляют следующие требования:

— результат классификации не должен зависеть от порядка обработки документов;

— классификация должна быть устойчивой, незначительные изменения данных должны вызывать лишь незначительные изменения результатов классификации.

Параллельные и последовательные методы классификации. Существуют два общих метода классификации: *параллельный и последовательный* [1, 2]. Предположим, что можно описать объект при помощи некоторого вектора. При параллельной классификации производится ряд тестов для всех компонент вектора, а затем делается предположение о соответствии объекта классу на основе объединенного результата этих тестов.

При последовательной классификации сначала проверяется некоторое подмножество компонент вектора описания, а затем в зависимости от результатов этих тестов либо производится классификация, либо выбирается новая совокупность тестов и новое подмножество компонент вектора описания, после чего указанный процесс повторяется.

Если задано некоторое множество тестов, то для выполнения последовательных процедур решения, вообще говоря, потребуется меньше тестов, чем для эквивалентной параллельной процедуры, а значит, будет израсходовано меньше вычислительных ресурсов. Существенный недостаток последовательной процедуры решения состоит в том, что она подвержена ошибкам в случае ненадежности отдельных тестов. Обычно последовательными схемами пользуются при наличии некоторых признаков, которые позволяют на ранних этапах работы процедуры существенно сократить множество возможных вариантов классификации.

Классификаторы текстовой информации обычно используют параллельные процедуры, так как семантические образы всех рубрик имеют примерно одинаковый приоритет. Наличие или отсутствие в документе терминов, содержащихся в семантическом образе одной рубрики, не позволяет сделать предположение о наличии или отсутствии в этом документе терминов, соответствующих другим рубрикам. Следовательно, окончательный вывод о рубриках, которым соответствует документ, можно сделать только после его сопоставления с семантическим образом каждой рубрики.

Для того чтобы ускорить процедуру классификации, можно учесть свойство иерархичности рубрикаторов. Действительно, если документ не соответствует, например, рубрике “Автомобили”, то он, скорее всего, не будет соответствовать подрубрике “Легковые автомобили”. При учете иерархичности алгоритм является последовательно-параллельным: он последовательно спускается вниз по дереву рубрик, параллельно сопоставляя документ со всеми рубриками текущего уровня иерархии.

Алгоритм классификации имеет следующий вид:

1) формируется семантический образ анализируемого документа: выделяются все термины, из них отбираются информативные и определяется их значимость для данного документа;

2) выполняется для корневой рубрики шаг 3);

3) для каждой рубрики текущего уровня иерархии семантический образ документа сопоставляется с семантическим образом данной рубрики и вычисляется мера подобия (числовая величина в диапазоне от 0 до 1, причем 0 означает, что семантические образы абсолютно не пе-

ресекаются, а 1 — что они совпадают и, следовательно, документ однозначно соответствует рубрике);

4) для каждой из рубрик, для которой результат сопоставления с ней документа выше некоторого порогового значения, рассматривается более низкий уровень иерархии и документ сопоставляется с дочерними рубриками. Если это сопоставление не приводит к обнаружению дополнительных рубрик, которым соответствует документ (или если рубрика не имеет дочерних), то документ полагается соответствующим данной рубрике. Если же дополнительные рубрики найдены, то документ полагается соответствующим только этим дополнительным рубрикам.

Рассмотрим работу алгоритма на примере классификации документа, описывающего особенности проектирования грузовых автомобилей повышенной проходимости (в примере используется дерево рубрик, представленное на рис. 1).

В ходе работы алгоритма выполняются следующие операции:

1. Документ сопоставляется с рубриками первого уровня (“Автомобили”, “Железнодорожный транспорт”, “Авиация и космонавтика”). В результате сопоставления принимается решение о том, что документ соответствует автомобильной тематике.

2. Документ сопоставляется с тремя рубриками, дочерними по отношению к рубрике “Автомобили” (“Легковые”, “Грузовые”, “Специальные”). В результате сопоставления принимается решение о том, что документ соответствует рубрике “Грузовые автомобили”.

3. На последнем этапе происходит окончательное сопоставление документа с соответствующей рубрикой (“Грузовые автомобили повышенной проходимости”).

Как было указано выше, в процедуре классификации учитывается то, что документ может быть отнесен к нескольким рубрикам одновременно, причем результат сопоставления документа с рубрикой зависит только от его содержимого и семантического образа этой рубрики. Это означает, что изменение семантического образа одной рубрики (или добавление новой рубрики) не влияет на результат классификации по другим рубрикам того же уровня иерархии.

Границы применимости рассмотренного алгоритма классификации. Алгоритм классификации основан на том, что большинство тематик имеют множества присущих им терминов, наличие которых в документе позволяет предположить его соответствие одной или нескольким из этих тематик. Например, для текстов юридической тематики характерны термины “иск”, “конфискация”, “правообладатель”, а вот термины “время”, “сторона”, “название” почти не отражают тематику документа.

Для того чтобы система классификации эффективно решала поставленную перед ней задачу, множество текстовых документов и множество тематик должны удовлетворять следующим требованиям:

— тематика рубрик и основное содержание документов должно быть представимо в виде набора ключевых слов, словосочетаний и фраз;

— к документам и рубрикам должен быть применим метод оценки смысловой близости между ними, основанный на сопоставлении содержащихся в них слов и словосочетаний.

Очевидно, что для большинства объектов в задачах классификации текстов (новости, документы, полученные из сети Internet, электронные письма) эти требования выполняются [3, 4].

Напротив, для текстовых полей баз данных, содержащих формальные текстовые признаки, а также для "сборных" документов (прейскурантов, списков фамилий депутатов и т.п.) указанные требования не выполняются. Автоматическая классификация таких документов потребует дополнительной настройки программы или даже переделки всего алгоритма.

Автоматическое обучение классификатора. Как было указано выше, составление семантических образов вручную для большого дерева рубрик крайне затруднительно. Поэтому такая процедура должна быть автоматизирована.

Процедуры, которые на основе предъявленных им образцов объектов составляют множества признаков каждого из классов объектов, называют процедурами распознавания образов. Различают процедуры, основанные на фиксированной выборке, и процедуры, основанные на последовательной выборке [5].

В случае фиксированной выборки несколько объектов для каждого из известных классов предъявляются системе распознавания до начала классификации. На основе этой выборки процедура распознавания вырабатывает правило классификации, применяемое затем к объектам, которые предопределяются указанной выборкой, но в ней не содержатся. Правило классификации далее не изменяется, даже когда происходят ошибки классификации.

При распознавании образов, использующем последовательную выборку, информация, получаемая из первоначальной выборки, является лишь предварительной, и она учитывается при построении соответствующего первоначального правила классификации. После выработки правила рассматривается следующая выборка, к которой применяется имеющееся правило классификации (часто результат новой классификации состоит лишь из одного объекта). Оценивается результат клас-

сификации и, если окажется необходимым, находится новое правило. Эту процедуру можно повторять, пока не будет удовлетворен некоторый критерий работы правила.

Итерационные процедуры обучения можно строить на основе нейронных сетей.

Итерационные модели удобнее для использования, тогда как статические проще в реализации. В задачах анализа текстовой информации, для которых разнообразие объектов, поступающих на вход системы, очень велико, построение итерационного алгоритма распознавания образов, не обращающегося к обучающим выборкам предыдущих итераций и пользующегося только текущим состоянием правила классификации и текущей обучающей выборкой, крайне затруднительно. Следует также учитывать возможность наличия (и накопления) ошибок в обучающей выборке, особенно если дерево рубрикатора большое, а экспертов, формирующих выборку, много.

Авторами настоящей работы предложен алгоритм распознавания образов, использующий фиксированную выборку. В случае необходимости дополнительного обучения классификатора достаточно добавить в исходную обучающую выборку дополнительные документы и повторить процедуру распознавания образов. Достоинством выбранного варианта является то, что специалисты, которые разрабатывают рубрикатор и подбирают для него эталонные документы, всегда могут проверить, имеются ли противоречия в составляемой ими обучающей выборке.

Алгоритм автоматического обучения классификатора имеет следующий вид.

1. Из документов выделяются термины.

2. Из всех терминов каждого из документов отбираются информативные и определяется их значимость (важность) для данного документа. Из документа выбираются N наиболее значимых терминов (N — значение некоторой функции от объема документа).

3. Строится индекс по терминам. Индекс содержит текст термина (нормальную форму и форму, пригодную для чтения), а также информацию о том, в каких документах встречается данный термин и сколько раз.

4. Определяется значимость каждого термина относительно всей базы. Эта величина вычисляется из значимости термина для каждого из документов, в которых он встречается, статистики его распределения по документам, морфологической информации, а также информации о контексте, в котором встречается термин. Подробнее вычисление значимости термина будет рассмотрено далее.

5. Выбирается некоторое количество наиболее значимых терминов.

6. Выбранные термины распределяются по рубрикам таким образом, чтобы каждой рубрике был приписан термин, часто встречающийся в документах этой рубрики (и ее подрубрик) и редко — в документах других рубрик. В результате каждой рубрике присваиваются списки терминов с весами, которые являются семантическими образами рубрик.

В настоящей работе рассмотрены шаги 1 и 2 алгоритма. Подробное обсуждение остальных шагов выходит за рамки настоящей работы. Описание алгоритмов обучения см. в работах [1, 3].

Методы выделения терминов из текста. Будем называть термином минимальную единицу документа, используемую программами обработки текстов. Как было отмечено, в качестве терминов используются слова или словосочетания. Разумеется, ни слова, ни словосочетания не являются терминами в прямом смысле этого слова. Однако за ними удобно оставить это название, поскольку термины, как правило, проблемно ориентированы. Иначе говоря, термины образуют понятия, относящиеся к той или иной предметной области, тематической рубрике, разделу знаний и т.д. Поэтому терминам может быть придана вполне определенная тематическая семантика [3].

Слова. Обычно программы начинают свою работу с разбиения документа на составные части: разделы, фрагменты, предложения и, наконец, слова. Дальше слова обычно процесс разбиения не идет, так как слоги и буквы уже не могут отражать смысл документа. Таким образом, для программ смысловой обработки текста слово является атомарным элементом.

Значимость. Не все слова текста равноправны. Некоторые более значимы (т.е. лучше отражают семантику документа), другие менее. Например, слово “классификация” значимо для настоящей работы (поскольку она посвящена проблемам классификации), а слово “программа” менее значимо, так как в настоящей работе рассматривается лишь один частный вид программ. Как видно из примера, значимость трудно определить формально, а еще труднее — сформулировать простой способ сравнения значимости разных слов (какое слово, например, более значимо для данной статьи: “формула” или “таблица”?).

Тем не менее, в программах смысловой обработки текстов используется понятие значимостей слов, предложений и более крупных элементов. При этом программы дают значимости численную оценку (обычно в диапазоне от 0 до 1), более значимому слову обычно ставится в соответствие бóльшее число.

Словосочетания. Более крупной единицей текста по сравнению со словами являются словосочетания. Очевидно, что они точнее отража-

ют смысл текста. Например, словосочетание "классификация текстов" точнее соответствует смыслу настоящей работы, чем слово "классификация".

Разбить текст на слова очень просто: границей слова является символ пробела или знак препинания. Разбиение же текста на словосочетания представляет собой сложную задачу — далеко не все идущие подряд слова текста составляют осмысленное связное словосочетание, которое можно использовать для анализа. Более того, такие словосочетания часто формируются из слов, которые в тексте не идут подряд (например, "осмысленное словосочетание" в случае предыдущего предложения).

Для выделения в тексте словосочетаний используют формальные методы, основанные на синтаксическом анализе [5–7], или приближенные, статистические методы. Один из таких методов будет рассмотрен далее.

Эквивалентность терминов. Для программы некоторые слова удобно считать эквивалентными. Например, поисковая программа будет по запросу "конь" находить документ, в котором присутствует только слово "коня". В зависимости от задачи эквивалентными могут считаться:

— слова, записанные буквами разного регистра (СЛОВО, слово, Слово, слоВО);

— формы одного и того же слова (конь, коня, коню, коням и т.д.);

— слова, находящиеся в одном и том же словообразовательном ряду (слово, словарь, словарный);

— слова, имеющие один и тот же корень (море, заморский);

— элементы одного синонимического ряда (холод, стужа, мороз).

При сопоставлении словосочетаний количество вариантов увеличивается: можно учитывать или не учитывать порядок слов, синтаксическую структуру и т.д. [3].

Нормальная форма термина. Процедуры проверки терминов на эквивалентность очень сложны. Для того чтобы их упростить, термины приводят к специальному виду — нормальной форме таким образом, чтобы все эквивалентные термины имели одну и ту же нормальную форму. После такого преобразования проверка эквивалентности сводится к двоичному сравнению. При сравнении слов без учета регистра достаточно понизить регистр букв слова, для других критериев эквивалентности приведение к нормальной форме более сложно.

Для сравнения с точностью до формы слова существительные приводят к именительному падежу единственного числа, глаголы — к инфинитиву и т.д. Для этого необходимо использовать морфологические словари [8] или специальные таблицы для приближенного преобразования [7].

Омонимия. При нормализации часто оказывается, что для слова может быть построено несколько нормальных форм. Например: слово "устав" имеет две нормальные формы: "устав" (существительное) и "устать" (глагол). Определение, какую именно нормальную форму использовать (разрешение омонимии), представляет собой исключительно сложную задачу, требующую глубокого синтаксического или даже семантического анализа текста. Все известные на сегодняшний день алгоритмы разрешения омонимии или требуют использования словарей, специально составляемых для каждой тематики [7], или не требуют специальной настройки, но дают ощутимый процент ошибок в реальных текстах [9]. Если омонимию разрешить не удалось, сравнивать приходится каждую возможную пару нормальных форм сопоставляемых слов.

Иногда при морфологическом анализе вместо нормальной формы слова используют некоторый числовой идентификатор (так называемый *идентификатор лексемы*). В качестве идентификатора может использоваться порядковый номер элемента морфологической таблицы, который описывает данное слово и все варианты его изменения [9], а может использоваться специальная хэш-функция.

Функция нормализации, считающая эквивалентными слова из одного словообразовательного ряда, может использовать в своей работе словообразовательные словари [9] или работать приближенным методом. Во втором случае у слова отсекается окончание и один или несколько суффиксов. Оставшаяся часть слова называется псевдоосновой и используется в качестве нормальной формы.

Для учета синонимических рядов необходимо использовать разного рода тезаурусы — общей лексики или тематические. Если в процессе нормализации слова в таких словарях найти не удалось, то выполняется нормализация с точностью до словообразовательного ряда или приведение слова к нормальной форме. Так же, как и в предыдущих случаях, в ходе нормализации слову может быть приписано несколько нормальных форм из-за омонимии, на этот раз — семантической. Разрешать такую омонимию еще сложнее, чем в предыдущем случае.

При нормализации словосочетаний можно учитывать их синтаксическую структуру, а можно ограничиться грубым методом, согласно которому каждое из слов словосочетания нормализуется отдельно, а затем нормальные формы слов упорядочиваются по алфавиту. При этом из словосочетания желательно сначала удалить предлоги, местоимения и другие малоинформативные слова. Например, словосочетания "просторы морей", "в море просторно" и "морские просторы" имеют одну и ту же нормальную форму "мор простор".

Стоп-словари. Для повышения эффективности и уменьшения размерности массивов терминов, обрабатываемых процедурой распознавания образов, выполняется фильтрация неинформативных терминов по специальному списку — стоп-словарю. Примерами стоп-слов являются слова "в", "для", "который". Примером стоп-выражений является словосочетание в "то же время". На первом этапе анализа текстов документов помечаются слова и выражения, присутствующие в указанном стоп-словаре. После подсчета значимостей предложений и их статистического анализа помеченные слова из текста удаляются и на дальнейших этапах процедуры выделения и анализа терминов не используются.

Выбор типа терминов и функции нормализации для классификатора текстов. Правильность выбора типа используемых в программе терминов является одним из наиболее существенных факторов, влияющих на общую эффективность работы классификатора. Для каждого из рассмотренных выше видов терминов и вариантов функции нормализации характерны ошибки, приводящие, в конечном итоге, к неправильным результатам классификации.

Ошибки выделения и сопоставления терминов обычно условно разделяют на две категории.

Термины, являющиеся семантически эквивалентными с точки зрения человека, имеют различающиеся машинные представления (нормальные формы). Данный вид ошибок влияет на полноту классификации. Пусть, например, в семантическом образе рубрики "генетика" присутствует термин "хромосома", а в семантическом образе анализируемого документа — термин "хромосомы". Если указанные термины имеют различающиеся нормальные формы, система решит, что в документе термин "хромосома" отсутствует, а значит, сопоставление приводит к ложному выводу о том, что документ не имеет ничего общего с рубрикой.

Термины, семантически различные с точки зрения человека, имеют совпадающие нормальные формы. Данный вид ошибок влияет на точность классификации. Примером такого рода ошибок может служить вхождение в векторы-индексы слова "замок", причем в семантическом образе рубрики "архитектура" оно использовалось для обозначения сооружения, а в документе — устройства для запирания дверей. Сопоставление векторов может привести к неправильному выводу о смысловой близости документа и рубрики.

Необходимо отметить, что оба типа ошибок успешно компенсируются размерами обучающих выборок классификатора. Действительно, большой массив документов обучающей выборки для рубрики "генети-

ка" скорее всего содержит слово "хромосома" во всех формах, а накопление в семантических образах рубрик большого количества терминов существенно уменьшает вероятность ложных срабатываний программы из-за ошибок второго рода.

В ходе тестирования разработанного авторами настоящей работы программного комплекса была проведена оценка общей эффективности классификации, основанной на использовании пяти способов выделения и нормализации терминов. Для этого было последовательно проведено обучение классификатора тремя выборками, причем после каждого сеанса обучения оценивались показатели полноты $w(D)$ и точности $u(D)$ работы классификатора. Обучающие множества состояли из 125, 295 и 450 документов соответственно, причем вторая выборка включала первую, а третья — вторую. Во всех трех случаях для классификации использовалась одна и та же база объемом 350 документов. Результаты приведены в табл. 1.

Рассмотрим указанные в таблице типы терминов.

Таблица 1

Результаты экспериментальной оценки эффективности использования различных видов терминов

Тип терминов	Выборка 1 (125 документов)		Выборка 2 (295 документов)		Выборка 3 (450 документов)	
	$w(D)$	$u(D)$	$w(D)$	$u(D)$	$w(D)$	$u(D)$
Словоформы	0,05	0,23	0,12	0,35	0,32	0,40
Лексемы	0,20	0,36	0,39	0,69	0,45	0,74
Псевдоосновы	0,22	0,59	0,60	0,62	0,79	0,65
Словосочетания (синтаксический анализ)	0,18	0,75	0,56	0,82	0,71	0,91
Словосочетания (приближенный метод)	0,23	0,70	0,63	0,75	0,78	0,83

Словоформы. В качестве терминов использовались отдельные слова, извлеченные из текста, без дополнительной обработки. Данный вариант приводит к крайне низкой полноте классификации при невысоких показателях точности. Малая точность объясняется тем, что процедура распознавания образов считает непохожими документы, содержащие разные формы одного и того же слова.

Лексемы. В качестве терминов использовались отдельные слова, причем каждое из них приводилось к нормальной форме (единственное число, именительный падеж и т.д.). Например, для слова "людей"

нормальная форма “человек”. Полнота выше, чем в предыдущем случае. Точность также выше за счет учета словоформ при определении меры подобия документов. Омонимия мало повлияла на результаты классификации.

Псевдоосновы. В процессе анализа слов выполняется приближенное определение окончания и суффиксов, после чего они удаляются. Отсечение суффиксов позволяет уменьшить количество ошибок первого рода, так как модель учитывает словообразование (слова “море” и “морской” имеют одинаковую основу “мор”), однако количество ошибок второго рода сильно увеличивается из-за высокой степени нечеткости при определения основы (слово “мор” также имеет основу “мор”).

Словосочетания (синтаксический анализ). В процессе анализа текстов выполняется поверхностный синтаксический анализ различных комбинаций слов. Распознанные именные и предложные группы приводятся к нормальной форме (именительный падеж, единственное число). Использование словосочетаний существенно уменьшает количество ошибок второго рода, так как словосочетания более точно идентифицируют предметную область, нежели отдельные слова.

Словосочетания (приближенный метод). В процессе анализа текстов выполняется приближенный (статистический) анализ различных комбинаций слов. Не все выбранные таким образом группы слов действительно являются словосочетаниями, некоторые из них образуют синтаксически некорректные словосочетания, однако эксперимент показал наивысшую эффективность данного метода. Нормализация выполняется следующим образом: из словосочетания удаляются предлоги, союзы, местоимения и другие малоинформативные слова, затем у слов отсекаются окончания и суффиксы, а получившиеся в результате псевдоосновы упорядочиваются по алфавиту.

Итак, эксперименты подтверждают практическую пользу использования словосочетаний для классификации текстовой информации. В качестве функции нормализации желательно использовать нечеткую функцию, использующую псевдоосновы.

Алгоритм приближенного выделения словосочетаний. Рассмотрим алгоритм приближенного выделения словосочетаний из текста, основанный на статистической обработке слов и цепочек соседних слов, встречающихся в тексте. От языка текстов зависят модули разбиения текста на предложения, предложений — на слова, модули, определяющие информативность слов в предложении, а также модуль, вычисляющий основы слов. Время, необходимое на разработку таких модулей для некоторого языка (имеются в виду, в первую очередь, европейские языки), на порядок меньше времени, необходимого для разработки мо-

дуля выделения словосочетаний, основанного на синтаксическом анализе текста.

Алгоритм выявляет часто повторяющиеся в документе группы слов, причем сопоставление выполняется с помощью функции нормализации, отсекающей окончание и суффиксы слов. Для каждой формируемой нормальной формы запоминается также соответствующий исходный текст словосочетания, причем если таких текстов несколько (документ содержит несколько эквивалентных форм словосочетания), выбирается тот вариант, который встречается в документе чаще всего.

Алгоритм приближенного выделения словосочетаний имеет следующий вид.

1. Из текста выделяется очередное предложение. Условия выделения конца предложения (для русских текстов) следующие:

— в конце слова стоит один из знаков конца предложения: «.», «?», «!»;

— слово, за которым знак конца предложения, состоит не менее, чем из двух букв;

— после знака конца предложения располагается один или несколько пробелов;

— следующее слово начинается с большой буквы.

2. Определяются информативные слова предложения. На вход соответствующего модуля подается предложение, на выходе имеем информацию о том, какие слова предложения информативны.

3. Запоминаются информативные слова предложения. Из информативного слова выделяется псевдооснова и запоминается в области информативных слов. Вместе с псевдоосновой запоминается слово. Область информативных слов представляет собой последовательность пар псевдооснова—слово.

Если слово, из которого выделена псевдооснова, находится от начала файла на расстоянии, не превышающем некоторое значение nWordsInHeader, то для этой псевдоосновы указывается специальный признак вхождения в начало файла. В типовом случае чем ближе появление слова к началу документа, тем выше его значимость. Например, в предложении имеется информативное слово “компьютерными”, для которого выделена псевдооснова “компьютер”. Эта псевдооснова, а также слово “компьютерными” запоминаются в области информативных слов.

4. Из предложения выделяются словосочетания. Под словосочетанием понимаем группу из двух или более информативных слов. Максимально допустимое количество слов словосочетания задается кон-

стантой nMaxWords. При выделении словосочетания из информативных слов предложения действуют следующие ограничения:

— между словами словосочетания не должно быть знаков препинания;

— между соседними словами словосочетания могут располагаться другие слова (как неинформативные, так и информативные) в количестве не большем, чем заданное специальной константой nMaxWordsDistance. Если она равна нулю, то между словами словосочетания не должно быть других слов.

Например, пусть nMaxWords $= 3$, а nMaxWordsDistance $= 1$. Рассмотрим следующее предложение.

Таким образом, для каждого термина вычисляем столько различительных весов, сколько рубрик на уровне иерархии.

Из данного предложения выделяем словосочетания:

термина вычисляем, термина вычисляем различительных, вычисляем различительных, вычисляем различительных весов, различительных весов, рубрик уровне, рубрик уровне иерархии, уровне иерархии.

5. Запоминаются словосочетания. Для слов словосочетания выделяются псевдоосновы. Для каждого словосочетания в области словосочетаний запоминается:

— совокупность псевдооснов слов словосочетания;

— собственно словосочетание.

Например, для словосочетания “термина вычисляем различительных” запоминаем совокупность псевдооснов “термин вычисл различ”, собственно словосочетание “термина вычисляем различительных”.

Если словосочетание находится от начала файла на расстоянии, не превышающем величины nWordsInHeader, то для этого словосочетания указывается специальный признак вхождения в начало файла.

6. Формируется частотный список псевдооснов, вычисляются веса.

6.1. Формируется частотный список псевдооснов. После обработки всех предложений текста в области информативных слов располагаются псевдоосновы информативных слов текста вместе со словами, из которых эти псевдоосновы выделены. Если слово несколько раз встретилось в тексте, то столько же раз псевдооснова этого слова записывается в область информативных слов. Поэтому для определения частоты встречаемости выполняется сортировка области информативных слов по коду содержащихся в ней псевдооснов. При этом одинаковые псевдоосновы оказываются рядом. В результате вместо области информативных слов получаем частотный список псевдооснов, в котором для

каждой псевдоосновы указана ее частота встречаемости в тексте.

Если для некоторой псевдоосновы указан признак вхождения в начало файла, то считаем, что это не одно появление, а несколько — количество задается константой nDocHeaderMultiply. Для каждой псевдоосновы помимо частоты запоминается самая частотная ее словоформа. При подсчете частотности словоформы признак вхождения в начало файла не учитывается.

Например, после сортировки имеем следующий фрагмент области информативных слов:

частот частоты*
частот частота
частот частота
частот частотой
частот частоты
частот частота
частот частотный
частот частотного

Здесь звездочкой обозначен признак вхождения псевдоосновы в начальную часть файла.

Предположим, что nHeaderWordMultiply $= 3$. Тогда приведенный выше фрагмент преобразуется в следующий:

частот 10 частота

Для ускорения дальнейшего поиска следует сделать индекс к частотному списку – например, для каждой пары первых букв псевдооснов указать смещение относительно начала частотного списка зоны с такими псевдоосновами.

6.2. Вычисляются веса псевдооснов по формуле

$$W = N^{C_1};$$

здесь N — частота псевдоосновы; C_1 — показатель степени (число с дробной частью).

7. Формируется частотный список словосочетаний. Выполняем действия, аналогичные описанным для шага 6. Сортировка выполняется по совокупности псевдооснов словосочетаний. В результате рядом оказываются словосочетания с одинаковым набором псевдооснов. Помимо определения частоты словосочетания отбирается самая частотная форма словосочетания.

Если для некоторого словосочетания указан признак вхождения в начало файла, то считаем, что это не одно появление, а несколько — количество задается константой nHeaderPhraseMultiply.

При определении частоты словосочетания учитывается признак вхождения словосочетания в начало файла. Однако при определении самой частотной формы словосочетания этот признак не учитывается.

В результате получаем частотный список словосочетаний, в котором каждый элемент включает следующие составляющие:

— совокупность псевдооснов словосочетания;

— частота словосочетания;

— самая частотная форма словосочетания.

Например, после сортировки имеем следующий фрагмент области словосочетаний:

частот словосоч частоты словосочетания*
частот словосоч частота словосочетания
частот словосоч частота словосочетания
частот словосоч частотой словосочетания
частот словосоч частоты словосочетания
частот словосоч частота словосочетания

Здесь звездочкой обозначен признак вхождения словосочетания в начальную часть файла.

Предположим, что nHeaderPhraseMultiply $= 3$. Тогда приведенный выше фрагмент преобразуется в следующий:

частот словосоч 8 частота словосочетания

Из полученного списка удаляем информацию о словосочетаниях, частота которых меньше некоторой минимально допустимой частоты.

8. Определяем веса словосочетаний.

8.1. Определяются частоты псевдооснов словосочетания. Для каждой псевдоосновы словосочетания из частотного списка псевдооснов определяется ее частота появления в тексте.

8.2. Определяется первая составляющая веса словосочетания по формуле

$$W_1 = \frac{N}{N_m};$$

здесь N — частота словосочетания; N_m — минимальная из частот псевдооснов словосочетания.

8.3. Определяется вес словосочетания по формуле

$$W = W_1^{C_2} N^{C_3} M^{C_4};$$

здесь M — число слов словосочетания; C_2, C_3, C_4 — некоторые константы (числа в диапазоне $0 \ldots 1$).

9. Сортируются псевдоосновы и словосочетания по убыванию веса. Для каждого словосочетания приводятся следующие данные:

— псевдоосновы слов словосочетания;

— самая частотная форма словосочетания;

—частота словосочетания и входящих в него псевдооснов; вес словосочетания.

Для каждой псевдоосновы приводятся следующие данные:

— псевдооснова;

— самая частотная словоформа псевдоосновы;

— частота псевдоосновы;

— вес псевдоосновы.

10. Выводится результирующий список, содержащий термины документа.

Методы определения значимости терминов. Качество решения задачи определения значимости терминов влияет на общую эффективность классифицирующей системы. Обзор методов определения весов приведен в работе [6]. Веса терминов можно определять на основе различных характеристик: позиционных, учитывающих расположение термина в документе (например, в заглавии, резюме, и т.д.), семантических, являющихся функцией отношений терминов к некоторым другим словам, или прагматических — например, в такой системе, где собственным именам придается очень большое значение. Кроме того, можно еще использовать веса, выводимые из частот терминов, или веса, зависящие от лексических свойств терминов.

Обычно при построении семантических образов рубрик используются не все термины эталонных документов, а только те, смысловая значимость (вес) которых выше некоторого порогового значения, что позволяет сократить объем вычислений.

В работе [1] характеристические множества терминов рубрик формируются чисто статистическими методами, без всякого учета контекста и явлений словоизменения, синонимии и полисемии естественного языка. Действительно, частотный анализ слов и анализ их совместной встречаемости позволяют скомпенсировать отсутствие этого учета, однако из экспериментов следует, что показатели эффективности систем, основанных только на статистических методах, оказываются низкими, что приводит к необходимости значительного увеличения объемов используемых обучающих выборок. Данный вывод подтверждается в работе [3].

В программе “Классификатор” [10, 11], разработанной одним из авторов настоящей работы, учитываются некоторые из перечисленных выше характеристик. В ней ранги W_i терминов t_i, можно представить в виде комбинации следующих составляющих:

$$W_i = CW_{\text{lex}i}^a W_{\text{cont}i}^b W_{\text{stat}i}^c, \qquad (1)$$

где $W_{\text{lex}i}$ — собственная (лексическая) значимость термина t_i, которая зависит только от самого термина; $W_{\text{cont}i}$ — контекстная значимость термина t_i, зависящая от информативности фрагментов текста, в которых он употребляется; $W_{\text{stat}i}$ — статистическая значимость термина t_i, зависящая от характеристик его распределения в документах; a, b, c — некоторые константы, характеризующие значимость каждого из факторов и определяемые экспериментально; C — нормирующий коэффициент.

Далее рассмотрим методы подсчета перечисленных составляющих значимостей.

Определение собственной значимости терминов. Собственную значимость термина t_i определим как произведение двух составляющих, первая из которых определяется вероятностью ошибок первого рода (потери полноты), а вторая — вероятностью ошибок второго рода (потери точности):

$$W_{\text{lex}} = (1 - P_1)^d (1 - P_2)^e, \tag{2}$$

где d, e — некоторые константы, определяемые экспериментально.

Вероятность P_1 зависит от количества терминов естественного языка, синонимичных данному, а вероятность P_2 — от количества его собственных смысловых значений. Для оценки обеих составляющих можно использовать поиск термина t_i практически в любых словарях соответствующего естественного языка (толковом, словообразовательном, словаре перевода на другой язык и т.д.). Вторую составляющую определяют на основе количества словарных статей, где термин присутствует в заглавии, а первую — на основе количества статей, где термин присутствует в остальном тексте статьи (толковании, переводе и т.п.).

Однако использование данного метода связано с большими затратами ресурсов. Вместо этого метода в программном комплексе “Классификатор” используется метод, в котором оценка обеих составляющих W_{lex} производится на основе морфологических характеристик слов терминов при помощи библиотеки морфологического анализа слов русского и английского языков MLMA. Вероятность ошибки первого рода (вероятность синонимии) может определяться как функция от части речи (признака отглагольности) [4]. Действительно, существительное “стол” почти не имеет синонимов, в отличие от глагола “ездить”. В работе [4] средние значения вероятности наличия синонимии для отдельных слов экспериментально оценены для различных частей речи и представлены в виде табл. 2.

Таблица 2

Средние вероятности синонимии для разных частей речи русского языка

Часть речи	Вероятности синонимии
Глаголы	0,45
Неотглагольные существительные	0,05
Отглагольные существительные	0,21
Прилагательные	0,06

При замене синонимом любого из слов вероятность наличия синонимии в термине, состоящем из нескольких слов, определим следующим образом:

$$P_1 = z_1^{n-1}\left(1 - \prod_{i=1}^{n}(1 - p_i)\right), \qquad (3)$$

где n — количество слов в термине; p_i — вероятность наличия синонимии для i-го слова термина; z_1 — поправочный коэффициент, учитывающий возможность сочетания синонимов слов термина.

Вклад величины z_1 в значение выражения (3) проиллюстрируем следующим примером: рассмотрим термин “грустный праздник”. Слово “грустный” имеет синонимы “мрачный”, “тоскливый”. Слово “праздник” имеет синонимы “выходной”, “красный день календаря”. Видно, что словосочетания “мрачный выходной” и “тоскливый выходной” являются возможными альтернативами исходному термину, а словосочетания “грустный красный день календаря” и “тоскливый красный день календаря”, скорее всего, в документах выделены не будут. В данном случае $z_1 = 0{,}66$. Экспериментально установлено, что формула (3) позволяет получить удовлетворительные результаты при $z_1 = 0{,}7$.

Вероятность ошибок второго рода для терминов, состоящих из одного слова, можно грубо оценить при использовании морфологического словаря, подсчитав количество омонимов термина. Например, слово “стол” имеет один омоним, а слово “устав” — два. Вероятность ошибок второго рода для терминов, состоящих из нескольких слов, практически равна нулю. Формула для вычисления вероятности P_2 для однословного термина имеет вид

$$P_2 = 1 - c^{h-1}, \qquad (4)$$

где h — количество омонимов термина; c — настроечный коэффициент, принятый равным 0,7.

Итак, на основании представленных рассуждений выбран следующий алгоритм определения показателей собственной значимости терминов.

1. Для каждого слова s_i термина t_i выполняются шаги 2—4.

2. Выполняется морфологический анализ слова s_i. Если оно известно библиотеке машинной морфологии MLMA, выполняется шаг 3, иначе — шаг 4.

3. Для каждого из омонимов слова s_i определяется принадлежность к классам слов из таблицы 2. Определяется среднее значение вероятности наличия синонимии для всех омонимов. Выполняется шаг 2.

4. Выполняется приближенная оценка вероятностей принадлежности к категориям из таблицы 2 методом, описанным в работах [4, 7]. Выбирается класс, вероятность принадлежности слова s_i к которому максимальна, и из таблицы извлекается соответствующее значение вероятности наличия синонимии. Выполняется шаг 2.

5. Вычисляется величина P_1 по формуле (3).

6. Если термин t_i состоит из одного слова и морфологизован на шаге 2, то определяется значение величины P_2 по формуле (4), иначе принимается $P_2 = 0$.

7. Вычисляется значимость W_{lex} термина t_i по формуле (2).

Определение контекстной значимости терминов. Контекстную значимость W_{conti} определим как функцию от значимостей каждого вхождения τ_{ik} термина t_i в текст документа. Часто используется следующий метод суммирования оценок значимости:

$$\begin{aligned}
&W_{12} = \max\bigl(w_{i1}, w_{i2}, 1 - (1 - w_{i1})^{\gamma}(1 - w_{i2})^{\gamma}\bigr),\\
&W_{123} = \max\bigl(w_{i2}, w_{i3}, 1 - (1 - w_{12})^{\gamma}(1 - w_{13})^{\gamma}\bigr),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&W_{\text{conti}} = \max\bigl(w_{i1\ldots n-1}, w_{in}, 1 - (1 - w_{12\ldots n-1})^{\gamma}(1 - w_{in})^{\gamma}\bigr),
\end{aligned} \tag{5}$$

где w_{ik} — значимость k-го появления термина t_i, $0 \le w_{ik} \le 1$; n — количество вхождений термина в документ; γ — настроечный коэффициент в диапазоне 0,5 . . . 1, определяемый на этапе наладки программы (начальное значение $\gamma = 0{,}7$). Малые значения параметра γ приводят к тому, что на величину W_{conti} влияют только наиболее значимые составляющие, тогда как большие значения параметра γ — к тому, что рассматривается большое количество вхождений.

Величины w_{ik} определяются на основе значимостей предложений текстов документов методами квазиреферирования текстов.

В работе [9] описан метод автоматического квазиреферирования, обладающий следующими основными особенностями:

— предложения текста делятся на автосемантичные (не связанные с другими предложениями) и синсемантичные (связанные с другими предложениями);

— автосемантичные предложения более информативны, чем синсемантичные, поэтому в первом приближении квазиреферат может быть составлен из всех автосемантичных предложений текста;

текст разбивается на группы предложений, первое из которых автосемантичное, а остальные синсемантичные, причем первое синсемантичное предложение связано с автосемантичным предложением, второе синсемантичное предложение связано с первым синсемантичным предложением и т.д.;

— формальным признаком синсемантичности предложения является наличие в нем коннектора.

Коннекторы делятся на безоценочные и оценочные, или логико-смысловые [9]. Коннекторы бывают истинные и ложные. Рассмотрим предложение: “В частности, недостаточно подробно освещен вопрос…”. В данном случае “в частности” является индикатором наличия связи между предложениями и поэтому является истинным коннектором. С другой стороны, словосочетание “в частности” для предложения “Не следует вдаваться в частности” является ложным коннектором. Для различения истинных и ложных коннекторов составляются специальные правила, учитывающие контекстное окружение предполагаемых коннекторов. Коннектор может быть словом (however, однако), словосочетанием (on the other hand, с другой стороны). Коннекторы разбиваются на группы, выражающие различные отношения между связанными предложениями: итог, противопоставление, следствие и т.д. Синсемантичные предложения, содержащие коннекторы разных групп, при прочих равных условиях имеют разные веса.

Другие подходы к автоматическому квазиреферированию предполагают использование дополнительных маркерно-индикаторных механизмов [6, 12]. Под маркерами понимаются слова и выражения, указывающие на различные структурные части реферируемого текста. Например, в случае научной статьи сочетание слов “является актуальным” указывает на принадлежность предложения к структурной части “постановка проблемы”. В соответствии с принятым решением о принадлежности предложения к некоторой структурной части ему присваивается некоторый дополнительный вес, который для различных структурных частей принимает разное значение. Под индикаторами понимаются слова и выражения, которыми в тексте выделяются предложения,

имеющие повышенную информативность. Примерами таких выражений являются "итак", "в качестве вывода следует отметить", "другими словами". При наличии в предложении индикатора ему присваивается дополнительный вес, который для различных индикаторов может иметь разное значение. Экспериментальная оценка представленного в работе [9] метода показала, что для 70 % текстов автоматически сформированные квазирефераты были вполне удовлетворительного качества.

В работе [7] рассмотрена возможность установления синтаксических связей между предложениями на основе лексического повтора. Экспериментальным путем установлено, что при повторении существительного связь действительно существует в 92 % случаев. Для прилагательных и глаголов данный параметр равен соответственно 65 % и 34 %. Метод установления связей на основе лексического повтора может быть использован в дополнение к методу коннекторов.

В программном комплексе "Классификатор" используются оба указанных метода квазиреферирования. Вес терминов определяется на основе следующих характеристик:

— числа появлений термина в тексте (чем больше появлений, тем выше вес);

— распределения по автосемантичным и различным типам синсемантичных предложений (например, при прочих равных условиях термин, входящий в пять автосемантичных предложений, имеет более высокий вес по сравнению с термином, входящим в пять синсемантичных предложений);

— количества связей между предложениями, образованных за счет лексического повтора термина (чем больше связей, тем выше вес); при этом следует учесть приведенные в работе [7] вероятностные характеристики образования связи при лексическом повторе в случае принадлежности термина различным частям речи; в частности, при повторе словосочетания вероятность образования связи выше аналогичной вероятности при повторе слов этого словосочетания;

— размеров связных фрагментов текста, содержащих термин (чем больше средний размер, тем выше вес); под связным фрагментом понимается несколько последовательных предложений текста, первое из которых автосемантичное, а остальные синсемантичные, причем связи между предложениями установлены вследствие наличия не только коннекторов, но и лексического повтора;

— наличия в предложении индикативного выражения, указывающего на его важность (если, например, в предложении содержится выражение "в качестве вывода следует отметить", то все термины этого

предложения должны получить приращение веса); в общем случае для каждого индикативного выражения приращение веса имеет разные значения;

— принадлежности термина к предложениям, входящим в различные структурные части текста. Так, появление термина в структурной части "выводы" более важно, чем в структурной части "введение". Для вычисления составляющей веса необходимо разбить текст на структурные составляющие. Каждая структурная составляющая в общем случае содержит несколько групп связанных предложений (первое из них автосемантичное).

Учет всех указанных параметров выполняется согласно следующему алгоритму.

Переменные: P_{s_i} — вес предложения s_i; L_i — вероятность смысловой связи предложения s_i с предложением s_{i-1}.

1. Для всех предложений полагается $P_{s_i} = 1$, $L_i = 0$.

2. Для каждого предложения s_i выполняются шаги 3–5, затем шаг 6.

3. Сопоставляется текст предложения s_i с элементами словарей коннекторов и индикаторов, величина P_{s_i} модифицируется согласно следующей формуле:

$$P'_{s_i} = P_{s_i} \frac{P_{\text{con}}}{P_{\text{ind}}}, \tag{6}$$

где P_{con} — среднее значение весов (важностей) всех коннекторов, сопоставленных с текстов предложения (данная величина равна единице, если ни один из коннекторов сопоставить не удалось); P_{ind} — то же для индикаторов (данная величина равна единице, если ни один из индикаторов сопоставить не удалось).

Важности коннекторов и индикаторов представляют собой числа в диапазоне $0{,}5 \ldots 1$, определяемые на этапе создания соответствующих словарей.

4. Если количество конннекторов, распознанных в предложении s_i, не равно нулю, модифицируется L_i по формуле

$$L'_i = 1 - (1 - L_i)c_L, \tag{7}$$

где c_L — настроечный коэффициент, равный 0,4.

5. Определяется количество и вид информативных терминов, одновременно присутствующих в соседних предложениях s_i и s_{i-1}. В случае, когда количество больше или равно значению настроечного параметра C_{nl}, величины L_i модифицируются в соответствии со следующей формулой:

$$L''_i = 1 - (1 - L'_i) \prod_{i=1}^{n} p_i, \tag{8}$$

где n — количество повторяющихся слов; p_i — коэффициент, зависящий от части речи повторяемого слова (вероятность отсутствия смысловой связи при лексическом повторе данного типа слов).

6. Все предложения s_i с вероятностью $L_i > C_{l\min}$ ($C_{l\min} = 0{,}5$) полагаются синсемантичными, остальные — автосемантичными.

7. Для каждого вхождения t_{ij} всех терминов t_i вычисляется w_{ij} по формуле

$$w_{ij} = P_{s_k}(1 - L_k)^{1/3} c_1 \frac{c_2 N_j + 1}{c_2 N_j}, \qquad (9)$$

где N_j — число предложений в связном фрагменте текста, содержащем данное вхождение термина t_{ij} (под связным фрагментом понимается несколько последовательных предложений текста, первое из которых автосемантичное, а остальные синсемантичные); k — номер предложения, в котором содержится t_{ij}; c_1, c_2 — настроечные коэффициенты, $c_1 = 0{,}43$, $c_2 = 3$.

8. Для всех терминов t_i вычисляется $W_{\text{cont}i}$ по формуле (5).

Определение статистической значимости терминов. Распределение частот появлений терминов по документам и характеристики их совместной встречаемости позволяют сделать выводы об их информативности. Например, термины, которые присутствуют почти во всех документах с большой частотой, скорее всего не могут служить признаками тематики. Эксперимент показывает, что такими характеристиками обладают термины общей лексики: “почти”, “больший”, “меньший” и т.д.

В работе [6] описаны несколько базовых методов определения статистической значимости.

При использовании метода частотных мер анализируют частоты f_{nk} появлений термина t_n в документе d_k, суммарные частоты появлений этого термина в наборе F_k и т.д. Часто предполагают, что термины, имеющие высокую частоту появления, не являются специфическими, но все же они могут дать большое число возможных совпадений при сравнении терминов рубрики и документа, обеспечивая, таким образом, классификацию многих релевантных документов (т.е. увеличивая полноту). Термины, имеющие низкую частоту появления, дают очень небольшое число совпадений, но если такие термины попали в список признаков рубрики и были найдены в обрабатываемых документах, то это почти наверняка говорит о релевантности соответствующего документа рубрике.

Существует интуитивное предположение, что наилучшими индексационными терминами, т.е. терминами, наиболее ценными для представления содержания документа, являются термины, не слишком редкие и не слишком частые. Поскольку, однако, удаление терминов, имеющих высокую частоту появления, может повлиять на полноту, выдви-

гались предложения считать значимость для нечастых терминов более высокой.

Вторая группа методов базируется на определении соотношения сигнал/шум по аналогии с теорией передачи информации Шеннона. Для набора из n документов шум N_k термина t_k выражается следующим образом:

$$N_k = \sum_{i=1}^{n} \frac{f_{ki}}{F_k} \log_2 \frac{F_k}{f_{ki}},$$

а сигнал S_k определяется формулой

$$S_k = \log_2(F_k) - N_k.$$

Шум находится в обратной зависимости от частоты употребления термина в наборе документов. Для равномерных распределений, когда термин встречается одинаковое число раз в каждом документе набора, шум принимает максимальное значение. Например, если термин t_k встречается один раз в каждом документе ($f_{ki} = 1$ для всех i), то $N_k = \log_2 n$, $S_k = 0$. Использование в качестве веса термина отношения S_k/N_k позволяет достигать неплохих результатов при определении значимости.

Третий класс методов основан на определении величины σ_k^2 распределения частоты термина. Если f_k^* — средняя частота термина t_k в n документах, то несмещенная выборочная оценка среднеквадратичного уклонения определяется следующим образом:

$$\sigma_k^2 = \frac{\sum_{i=1}^{n} (f_{ki} - f_k^*)^2}{n-1}.$$

Тогда параметром, с помощью которого можно оценивать пригодность некоторого термина, служит отношение $F_k\sigma_k^2/f_k^*$. Если термин имеет близкое к равномерному распределение, т.е. если все f_{ki} имеют одинаковый порядок, то σ_k^2 мало, и это показывает, что термин не очень полезен. С другой стороны, если термин t_k редок и встречается только в нескольких документах, большая часть частот f_{nk} равна нулю и σ_k^2 мало. Наибольшие значения уклонения имеют термины с асимметричным распределением и средним значением частоты появления в документах.

Как было обнаружено на практике, полезными характеристиками обладают также параметры, основанные на способности термина различать документы набора. Рассмотрим некоторый набор документов и вычислим среднее значение некоторого коэффициента подобия

$S(d_i, d_j)$ по формуле для всех пар документов набора:

$$S^* = c \sum_{\substack{i=1,\, j=1 \\ i \neq j}} S(d_i, d_j). \qquad (10)$$

Рассмотрим теперь исходный набор терминов, причем пусть из всех описаний документов исключен термин t_k, и пусть S_k^* — среднее значение коэффициента подобия в этом случае. Если термин t_k имеет высокую частоту появления и распределение частот, близкое к равномерному, то исключение этого термина уменьшает средний коэффициент попарного подобия документов, т.е. имеем $S_k^* < S_k$. Напротив, если термин t_k имеет асимметричное распределение (т.е. он приписан только некоторым документам), вероятно, что его исключение увеличит среднее значение коэффициента попарного подобия, т.е. $S_k^* > S_k$.

Определим дискриминационное значение каждого термина t_k как некоторую функцию от $S_k^* - S_k$. Эксперименты показывают [13], что данный метод, несмотря на большую его вычислительную сложность, позволяет получать наилучшие оценки значимости терминов.

В программном комплексе "Классификатор" выбран последний метод статистической оценки значимости, причем вычисления проводятся с помощью алгоритма, подобного описанному в работе [13], по следующей формуле:

$$W_{\text{stat}i} = \frac{1}{\sum_{i=1}^{N} \overline{f_i^2}} \left(\frac{\sum_{i=1}^{N} \overline{f_i^2}\, \overline{f_i}^2}{\sum_{i=1}^{n} \overline{f_i}^2} - \overline{f_i}^2 \right), \qquad (11)$$

где $\overline{f_i}$ — среднее число появлений термина t_i в документе; $\overline{f_i^2}$ — средний квадрат числа появлений термина t_i в документе.

Заключение. Эксперименты подтверждают практическую пользу применения словосочетаний в качестве терминов при анализе текстов. Процедуры выделения словосочетаний, основанные на поверхностном синтаксическом анализе и на статистическом анализе, привели к примерно равным показателям качества классификации.

Необходимо отметить следующее.

Синтаксический анализ проводится значительно медленнее, чем прелагаемые процедуры, хотя использует меньший объем памяти. Дело в том, что статистическая процедура во время анализа формирует в памяти полный список всех возможных терминов документа, а только потом начинает обработку. Напротив, синтаксический анализ выполняется для каждого предложения индивидуально и не требует накопления

промежуточных результатов. Низкая скорость синтаксического анализа определяется большим количеством вариантов, которые необходимо перебрать при выявлении связей между словами.

С помощью синтаксического анализа получаются более точные термины, однако плохо обрабатываются незнакомые морфологическим библиотекам слова. Таким образом, если текст содержит много идентификаторов, аббревиатур или английских слов, синтаксический анализ будет работать хуже статистического. Например, в тестовых документах с помощью синтаксического анализа выделены слова Windows и NT по отдельности, а с помощью статистического — словосочетание Windows NT.

Статистическая процедура значительно проще переносится на другой язык. В настоящее время авторами настоящей работы подготовлен модуль приближенного выделения терминов из английских текстов.

СПИСОК ЛИТЕРАТУРЫ

1. Chekuri Ch., Goldwasser M. H. Web Search Using Automatic Classification (Computer Science Department, Stanford University).
2. Каневский Е. А. Методы классификации текста // Труды Международного семинара "Диалог'98" по компьютерной лингвистике и ее приложениям. Т. 2. – М., 1998. – С. 488–497.
3. Сомин Н. В., Соловьева Н. С., Соловьев С. В. Система рубрикации текстовых сообщений // Труды Международного семинара "Диалог'98" по компьютерной лингвистике и ее приложениям. Т. 2. – М., 1998. – С. 574–581.
4. Шабанов В. И. Автоматическое индексирование запросов в документальной ИПС, основанное на статистической и морфологической информации // КомпьюЛог. – 1997. – № 3. – С. 20–24.
5. Хант Э. Искусственный интеллект. – М.: Мир, 1978. – 560 с.
6. Солтон Дж. Динамические библиотечно-информационные системы. – М.: Мир, 1979. – 550 с.
7. Белоногов Г. Г., Богатырев В. И. Автоматизированные информационные системы. – М.: Сов. радио, 1973. – 325 с.
8. Грамматический словарь русского языка / Под ред. А.А. Зализняка. – М.: Русский язык, 1977.
9. Ашманов И. С. Методы анализа текстов на естественном языке, используемые при проверке его орфографии и пунктуации / Дисс.. . . канд. техн. наук. – М., 1994.
10. Андреев А. М., Березкин Д. В., Шабанов В. И. Методы выделения терминов из текста // Современные информационные технологии: Сб. докл. – М.: МГТУ им. Н.Э.Баумана, 2001. – С. 117–127.
11. Шабанов В. И., Андреев А. М., Сюзев В. В. Построение ассоциативных связей в системе обработки текстов // Современные информационные технологии. Юбилейн. сб. трудов кафедры. – М.: МГТУ им. Н.Э. Баумана, 2002. – С. 191–196.
12. Перевозчикова К. В. Экспериментальное исследование вторичных документов, полученных машинным экстрагированием по маркерно-индикаторному методу // НТИ. Сер. 2. – 1987. – № 6. – С. 23–29.

13. Харин Н. П. Метод ранжирования выдачи, учитывающий автоматически построенные ассоциативные отношения между терминами // НТИ. Сер. 2. – 1990. – № 9. – С. 19–23.

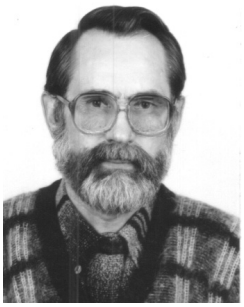
Статья поступила в редакцию 24.04.2003

Арк Михайлович Андреев родился в 1943 г., окончил в 1967 г. МВТУ им. Н.Э. Баумана. Канд. техн. наук, доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана. Автор более 70 научных работ в области вычислительных средств и систем управления.

A.M. Andreev (b. 1943) graduated from the Bauman Moscow Higher Technical School in 1967. Ph. D. (Eng.), ass. professor of "Computer Systems and Networks" department of the Bauman Moscow State Technical University. Author of over 70 publications in the field of computational means and control systems.

Дмитрий Валерьевич Березкин родился в 1966 г., окончил в 1990 г. МГТУ им. Н.Э. Баумана. Канд. техн. наук, директор Научно-производственного центра "Интелтек Плюс". Автор около 30 научных работ в области вычислительных средств.

D.V. Beryozkin (b. 1966) graduated from the Bauman Moscow State Technical University in 1990. Ph. D. (Eng.), director of the Scientific and Industrial Center "INTELTEK PLUS". Author of about 30 publications in the field of computational means.



Владимир Васильевич Сюзев родился в 1946 г., окончил в 1970 г. МВТУ им. Н.Э. Баумана. Д-р техн. наук, профессор, зав. кафедрой "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана. Автор более 100 научных работ в области методов и алгоритмов цифровой обработки информации.

V.V. Syuzev (b. 1946) graduated from the Bauman Moscow Higher Technical School in 1970. D. Sc. (Eng.), professor, head of "Computer Systems and Networks" department of the Bauman Moscow State Technical University. Author of over 100 publications in the field of methods and algorithms of digital processing of data.

Владислав Игоревич Шабанов родился в 1976 г., окончил в 1999 г. МГТУ им. Н.Э. Баумана. Аспирант кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана, сотрудник ОАО "Рамблер", ведущий разработчик поисковой машины Rambler. Автор 9 научных работ в области методов, моделей и алгоритмов интеллектуальной обработки текстовых документов.

V.I. Shabanov (b. 1976) graduated from the the Bauman Moscow State Technical University in 1999. Post-graduate of "Computer Systems and Networks" department of the Bauman Moscow State Technical University. Author of 9 publications in the field of methods, models and algorithms of intellectual processing of text documents.