Московский Государственный Университет им. М.В. Ломоносова

На правах рукописи

Агеев Михаил Сергеевич

Методы автоматической рубрикации текстов, основанные на машинном обучении и знаниях экспертов

05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата физико-математических наук

Научные руководители: д.ф.-м.н., акад. Бахвалов Н.С.,

д.т.н, проф. Макаров-Землянский Н.В.

Москва, 2004

ОГЛАВЛЕНИЕ

1 Введение

Классификация/рубрикация информации (отнесение порции информации к одной или нескольким категориям из ограниченного множества) является традиционной задачей организации знаний и обмена информацией. В больших информационных коллекциях имеет смысл говорить только об автоматической рубрикации.

Предложено много методов для решения данной задачи посредством автоматических процедур. Существующие методы можно разделить на два принципиально различных класса: методы машинного обучения и методы, основанные на знаниях (также иногда именуемые "инженерный подход").

При применении методов машинного обучения для построения классификатора используется коллекция документов, предварительно отрубрицированная человеком. Алгоритм машинного обучения строит процедуру классификации документов на основе автоматического анализа заданного множества отрубрицированных текстов.

При использовании методов, основанных на знаниях, правила отнесения документа к той или иной рубрике задаются экспертами на основе анализа рубрикатора и, возможно, части текстов, подлежащих рубрицированию.

Отметим некоторую условность названия "методы, основанные на знаниях". Любые методы автоматической классификации текстов в той или иной форме используют знания о свойствах текста на естественном языке и знания об особенностях текстов, принадлежащих той или иной рубрике. Принципиальная разница между двумя группами методов состоит в том, что методы машинного обучения используют математические методы для извлечения знаний из обучающей коллекции текстов, в то время как "инженерный подход" использует знания эксперта о свойствах текстов, принадлежащих рубрикам. Знания эксперта основываются, в первую

очередь, на предыдущем опыте, в частности, на большой коллекции прочитанных ранее текстов, и во вторую очередь, на части текстов, подлежащих рубрицированию.

В настоящее время можно наблюдать существенный разрыв в исследованиях и в практических методах между двумя указанными подходами к автоматической классификации текстов — методами машинного обучения и методами, основанными на знаниях.

В исследованиях, посвященных применению методов машинного обучения для классификации текстов, применяются универсальные алгоритмы, которые применимы для широкого круга задач анализа и обработки информации. Например, метод SVM (Support Vector Machines, [78, 55]) успешно используется для задач распознавания образов и оценки плотности сред. Для задачи классификации текстов эти методы работают с абстрактной векторной моделью документа и не учитывают особенностей задачи тематической классификации текстов и структуры рубрикатора. Тем не менее, во многих случаях методы машинного обучения дают весьма высокие результаты. Качество рубрикации для систем, основанных на машинном обучении, является довольно высоким для небольших рубрикаторов, и сильно падает с увеличением количества рубрик и усложнением структуры рубрикатора.

Во многих случаях, даже при наличии заранее отрубрицированной коллекции документов, методы машинного обучения неприменимы и используется значительно более трудоемкий инженерный подход [2, 8]. Необходимость применения методов, основанных на знаниях, для больших рубрикаторов — 500 и более рубрик — отмечалась, в частности, нескольких докладах на семинаре по практической классификации текстов в рамках конференции SIGIR-2001 и SIGIR-2002 [71, 59]. Инженерный подход обычно обеспечивает высокое качество рубрицирования и "прозрачность" алгоритма

— результаты обработки легко интерпретировать (почему такой-то документ был отнесен к рубрике). К сожалению, при использовании инженерного подхода зачастую совсем не используется ресурс, состоящий в наличии коллекции отрубрицированных текстов. Основной проблемой инженерного подхода является высокая трудоёмкость создания системы автоматической классификации (от 1 до 8 человеко-часов на одну рубрику [82, 30]).

В связи с вышеизложенным, задача повышения эффективности методов автоматической классификации текстов на основе интеграции двух подходов представляется актуальной.

Наше исследование посвящено сравнению различных методов классификации текстов, выделению положительных сторон и проблем каждого из методов, разработке более эффективных методов, использующих преимущества машинного обучения и экспертного подхода. Целью данных исследований является:

- Создание методов автоматической классификации текстов, сочетающих в себе преимущества методов машинного обучения и методов, основанных на знаниях. Разработка эффективных методов машинного обучения, учитывающих особенности задачи классификации текстов.
- Улучшение существующих процедур классификации текстов, использующих инженерный подход — в первую очередь, уменьшение трудоёмкости. Создание различных помощников для автоматической проверки и коррекции описания рубрик и результатов рубрицирования.

Содержание диссертации организовано в соответствии с указанными целями:

- В разделе 2 даётся обзор методов, применяемых для автоматической классификации текстов. Описываются базовые технологии,

применяемые для обработки текстов и общепринятые методы оценки результатов классификации. Наиболее эффективные методы классификации текстов используются в дальнейшем исследовании в качестве отправной точки для сравнения и для разработки более эффективных методов.

- В разделе 3 приводится описание и исследование разработанного автором метода машинного обучения для автоматической классификации текстов, основанного на моделировании логики рубрикатора. Описываемый алгоритм строит правила отнесения документов к рубрике в виде, аналогичном используемому экспертами при инженерном подходе.

  Теоретическое рассмотрение позволяет доказать, что при определённых предположениях о содержании рубрики алгоритм строит описание рубрики, близкое к оптимальному.

  Экспериментальное исследование на различных коллекциях реальных текстов позволяет утверждать что

  1. создаваемые алгоритмом правила описания рубрики соответствуют содержанию рубрики;
  2. алгоритм показывает высокое качество классификации текстов (в одном из сравнительных тестов — лучший результат по сравнению с 8 другими алгоритмами).
- В разделе 4 описываются разработанные автором методы и технологии повышения эффективности методов классификации текстов, основанных на знаниях. Описываемые технологии основаны на статистическом анализе распределения понятий и метаданных в коллекции документов и реализованы в виде интерактивных инструментов в полнотекстовой информационной системе. Разработана методика применения указанных средств для повышения эффективности работы экспертов, создающих описания рубрики.

Данные средства внедрены в технологический процесс построения систем классификации текстов проекта Университетская Информационная Система РОССИЯ, разрабатываемого в НИВЦ МГУ (Научно-Исследовательском Вычислительном Центре МГУ им. М.В. Ломоносова).

2 Обзор методов автоматической рубрикации текстов

В данном разделе даётся обзор основных подходов, применяемых для автоматической классификации текстов. Мы опишем базовые технологии, применяемые для обработки текстов и общепринятые методы оценки результатов классификации.

Стоит отметить, что в рамках данного обзора мы не можем покрыть весь спектр методов и технологий, применяемых для автоматической классификации текстов. Поэтому мы выбрали, с одной стороны, «классические» методы, которые часто цитируются в литературе. С другой стороны, в данном обзоре обосновывается выбор методов, которые мы выбрали в качестве отправной точки для дальнейших исследований по разработке более эффективных методов.

Структура обзора следующая:

- В разделе 2.1 мы опишем основные подходы к представлению текстов для компьютерной обработки. Описываемые подходы являются в некотором смысле «классическими» и используются как алгоритмами классификации текстов (машинного обучения и основанными на знаниях), так и алгоритмами поиска информации (например, в поисковых системах).
- В разделах 2.2 и 2.3 описываются общепринятые метрики качества рубрицирования и способы вычисления метрик на коллекции документов.
- В разделе 2.4 мы дадим обзор публикаций, посвященных практическому сравнению различных методов классификации текстов, основанных на машинном обучении. Основным выводом из нескольких независимых публикаций является преимущество одного из методов — SVM (Support Vector Machines, описание в разделе 2.5.7) — над другими методами машинного обучения. Это позволяет нам выбрать SVM в качестве

отправной точки для сравнения разрабатываемых нами методов с другими методами машинного обучения. Основным недостатком метода SVM является сложность в интерпретации правил отнесения документов к рубрике, которые используются SVM. Это означает, что для достижения целей диссертации — взаимной интеграции методов машинного обучения и методов, основанных на знаниях — SVM мало пригоден и требуются иные подходы.

- В разделе 2.5 мы дадим обзор методов машинного обучения, применяемых для автоматической классификации текстов. Мы выбрали широко известные методы (в частности, упоминаемые в публикациях по сравнению методов). Более подробно описывается метод SVM и методы, строящие описание рубрики в виде, пригодном для анализа человеком (кандидаты для использования в наших целях).
- В разделе 2.6 мы опишем методы автоматической классификации тестов, основанные на знаниях.
- В последнем разделе 2.7 мы опишем выводы из данного раздела.

2.1 Основные подходы к представлению текстов для компьютерной обработки

Первым этапом решения задачи автоматической классификации текстов является преобразование документов, имеющих вид последовательности символов, к виду, пригодному для алгоритмов машинного обучения в соответствии с задачей классификации. Обычно алгоритмы машинного обучения имеют дело с векторами в пространстве $\mathbb{R}^n$ (называемом также пространством признаков). Отображение документов в пространство признаков также используется и методами, основанными на знаниях.

Вторым этапом является построение классифицирующей функции при помощи обучения на примерах.

Качество рубрицирования зависит и от того, как документы будут преобразованы в векторное представление, и от алгоритма, который будет применен на втором этапе. При этом важно отметить, что методы преобразования текста в вектор специфичны для задачи классификации текстов и могут зависеть от коллекции документов, типа текста (простой, структурированный) и языка документа. Методы машинного обучения, применяемые на втором этапе, не являются специфичными для задачи классификации текстов и применяются также в других областях, например, для задач распознавания образов.

Рассмотрим классический подход для отображения текста в вектор, используемый многими системами автоматической классификации текстов. Этот метод основывается на предположении о том, что категория, к которой относится данный документ, зависит от относительной частоты слов, входящих в текст. Это предположение, конечно, является упрощением. Существуют примеры систем, которые учитывают более сложные факторы: порядок слов в тексте [69], структура текста, содержащего разметку [43, 37].

Базовый метод отображения текста в вектор заключается в том, что каждому слову, которое встречается в каком-либо документе, соответствует определенная координата в пространстве признаков. Для слова, встречающегося в документе, значение соответствующей координаты положительно и пропорционально частоте слова в документе. Для слова, которое не встречается в документе, значение соответствующей координаты равно нулю.

Есть несколько причин, по которым следует стремиться уменьшить размер пространства признаков. Во-первых, учет всех встреченных в документах слов приводит к слишком большой размерности пространства,

хотя многие слова слабо влияют на результаты рубрицирования (либо вообще не влияют). Высокая размерность пространства признаков может приводить к высокой вычислительной погрешности и низкой скорости работы алгоритмов обучения. Во-вторых, отображение нескольких близких по значению слов в одну координату может улучшить результаты рубрицирования. Например, различные морфологические формы слова следует считать эквивалентными.

Опишем основные приемы, применяемые для преобразования текстов в векторы пространства признаков.

2.1.1 Использование морфологии

Для того чтобы объединять различные морфологические формы слова в одну координату пространства признаков, каждое слово исходного текста приводится к своей нормализованной форме (лемме). Для английского языка обычно применяется процедура нормализации слов, которая заключается в отсечении окончания слова (stemming). Для русского языка процедура нормализации слов является более сложной, но на данный момент существуют распространённые методы её решения [20]. Отдельной проблемой является тот факт, что в естественном языке одному слову текста может соответствовать несколько различных начальных форм. Например, слову "суда" можно сопоставить две начальные формы: "суд" и "судно". В таких случаях имеет смысл добавлять к тексту обе начальные формы слова. Существуют методы разрешения многозначности слов в тексте [32], которые позволяют определять, какое из значений слова следует использовать в данном случае, однако мы не будем рассматривать эти методы в рамках данной работы.

2.1.2 TF*IDF

Отдельной задачей при преобразовании текста в вектор является вычисление значений координат в пространстве $\mathbb{R}^n$, соответствующих признакам, также называемых *весами* признаков. Выбор весов признаков существенно влияет на качество рубрицирования. В статье [75] приводится подробное исследование различных подходов к выбору весов признаков. Результаты экспериментов, описанных в этой статье, показывают, что одной из лучших формул вычисления весов является

$$w_i = \frac{tf_i \cdot idf_i}{\sqrt{\sum_j \left(tf_j \cdot idf_j\right)^2}} \qquad (2.1)$$

Где w_i - вес i-го слова, tf_i - частота встречаемости i-го слова в данном документе (term frequency), $idf_i = \log\frac{N}{n}$ - логарифм отношения количества всех документов в коллекции к количеству документов, в которых встречается i-е слово (inverse document frequency).

Такой выбор формулы можно обосновать теоретически следующими соображениями:

1) Чем чаще слово встречается в документе, тем оно важней. Этот факт учитывает множитель tf_i.
2) Если слово встречается во многих или во всех документах, то это слово не может являться существенным критерием принадлежности документа рубрике и его вес следует понизить. Наоборот, если слово встречается в малом количестве документов, то его вес следует повысить. Множитель idf_i учитывает это соображение и соответствует весу слова ("контрастности") в данной коллекции документов.
3) Для того чтобы учесть различную длину текстов документов в коллекции, веса слов документов следует нормализовать. В

формуле (1) веса нормализуются так, чтобы сумма квадратов весов каждого документа была равна 1.

Существуют также другие варианты формулы tf*idf, которые дают близкие по качеству результаты. В наших экспериментах мы использовали TF*IDF в формулировке INQUERY [56, 14]:

$$w_i = \beta + (1-\beta)\cdot\frac{tf_i}{tf_i + 0.5 + 1.5\cdot\frac{dl}{\text{avg\_dl}}}\cdot\frac{\log\left(\frac{N+0.5}{n}\right)}{\log(N+1)}$$

где dl — мера длины документа, avg_dl – средняя длина документа, β = 0.4, где N — количество документов в коллекции, n — количество документов, где встретилось i-е слово.

В некоторых случаях для вычисления веса слова в тексте привлекается также дополнительная информация [37]. Например, можно учитывать информацию о структуре текста и словам, встреченным в заголовке, присваивать больший вес [54].

2.1.3 Борьба с высокой размерностью: сокращение числа используемых атрибутов путем выделения наиболее значимых.

Даже после приведения всех слов документа к нормализованной форме, полученное пространство признаков имеет очень большую размерность (десятки тысяч). Эту размерность можно существенно уменьшить без ухудшения качества рубрицирования, если выкинуть слова, слабо влияющие на результаты рубрицирования [81].

Во-первых, обычно из списка признаков удаляют так называемые "стоп-слова" — предлоги, союзы и т.п. Это не сильно сокращает размерность пространства признаков (список стоп-слов составляется вручную и обычно

является небольшим). Но зато удаление стоп-слов обычно улучшает качество рубрицирования за счет удаления информационного шума.

Во-вторых, из списка признаков можно удалить слишком редко встречающиеся слова.

Опишем эксперимент по обработке коллекции русскоязычных текстов "Нормативно-правовые акты РФ" из НТЦ "СИСТЕМА" (Научно-технический центр правовой информации "Система" ФАПСИ РФ). Всего было обработано 10372 документа общим объемом около 65 мегабайт. После приведения слов к нормальной форме получилось 202584 различных слов, из которых около 80% встречались всего только в одном документе. После отсечения всех слов, встречающихся менее чем в 5 документах, получилось 23118 различных слов. При этом результаты рубрицирования при помощи SVM (п. 2.5.7) изменились менее чем на 5 процентов по всем рубрикам, причем в основном в лучшую сторону (видимо, за счет уменьшения вычислительной погрешности).

Кроме удаления редко встречающихся слов часто применяется методы выделения слов с использованием критерия *информационного веса слова в рубрике* (mutual information gain). Информационный вес слова в рубрике определяется по формуле

$$MI(x_i, c) = \sum_{x_i \in \{0,1\}} \sum_{c \in \{0,1\}} P(x_i, c) \log \frac{P(x_i, c)}{P(x_i) P(c)} \qquad (2.2)$$

Здесь

$$P(x_i = 1) = 1 - P(x_i = 0) = \frac{\text{количество документов, содержащих слово } x_i}{\text{количество всех документов}},$$

$$P(c = 1) = 1 - P(c = 0) = \frac{\text{количество документов, принадлежащих рубрике } c}{\text{количество всех документов}},$$

$P(x_i, c)$ - вероятности совместного распределения слов и рубрики. Легко видеть, что если распределения слова x_i и рубрики c статистически

независимы, то $MI(x_i, c) = 0$. Если же между встречаемостью слова x_i и рубрики c имеется строгая логическая зависимость, то $MI(x_i, c)$ - максимально. Метод сокращения размерности на основе выделения наиболее информационно-значимых слов применяется, например, в работе [58].

2.1.4 Использование дополнительных атрибутов документа

В некоторых случаях, кроме слов, в векторное представление текста включаются также дополнительные атрибуты документа. Это позволяет улучшить качество рубрицирования [54]. В разделе 2.6.3 мы опишем подход к построению векторного представления документов, основанный на терминологическом индексе.

2.2 Метрики качества рубрицирования

Основным критерием оценки качества работы методов автоматической классификации текстов является сравнение результатов работы метода с оценками экспертов. При этом "идеальным" алгоритмом считается тот, для которого выводы, сделанные системой, согласуются с мнением экспертов оценщиков [9, 74, 42].

Для проведения таких оценок существуют, с одной стороны, свободно доступные коллекции отрубрицированных документов [70]. С другой стороны, ежегодно проводятся конференции по практической оценке методов классификации: международная TREC (Text REtrieval Conference, http://trec.nist.gov) и российская РОМИП (Российский семинар по Оценке Методов Информационного Поиска, http://romip.narod.ru [51]).

Наиболее широко применяемыми оценками качества рубрицирования являются полнота и точность. Введём определения. Пусть C — множество документов, принадлежащих рубрике и u — множество документов, автоматически приписанных рубрике.

Определение (полнота классификации документов по рубрике): *полнота (recall) классификации документов по рубрике вычисляется как отношение количества документов, правильно приписанных (автоматически) к рубрике к общему количеству документов, относящихся к данной рубрике:*

$$r(u) = \frac{|u \cap C|}{|C|}$$

Определение (точность классификации документов по рубрике): *точность (precision) классификации документов по рубрике вычисляется как отношение количества документов, правильно приписанных (автоматически) к рубрике к общему количеству документов, приписанных к данной рубрике:*

$$p(u) = \frac{|u \cap C|}{|u|}$$

Полнота и точность классификации обычно измеряются в процентах. Для идеального алгоритма и полнота, и точность равны 100%.

Более простая оценка качества классификации - процент правильно классифицированных документов среди всех документов - редко используется для оценки качества автоматической классификации документов, так как эта оценка плохо отражает реальные свойства алгоритма для малочастотных рубрик. Например, если к некоторой рубрике относится всего 1% документов (довольно типичная ситуация), то тривиальный алгоритм, который не приписывает рубрику ни к одному документу, будет правильно классифицировать 99% документов. В то же время полнота для данного алгоритма будет равна нулю.

В некоторых случаях для оценки качества классификации требуется оценка в виде одного числа. Существует несколько известных формул для

такой оценки [79, 77], одной из наиболее часто используемых является так называемая F-measure:

$$\mathrm{F}(\mathrm{u}) = \frac{2}{\frac{1}{\mathrm{p}(\mathrm{u})} + \frac{1}{\mathrm{r}(\mathrm{u})}}$$

Или, в общем случае,

$$\mathrm{F}_{\beta}(\mathrm{u}) = \frac{1+\beta}{\frac{\beta}{\mathrm{p}(\mathrm{u})} + \frac{1}{\mathrm{r}(\mathrm{u})}} \qquad (2.3)$$

Здесь $\beta > 0$ — параметр, устанавливающий отношение важности параметров полноты и точности. Если $\mathrm{p}(\mathrm{u}) = 0$ или $\mathrm{r}(\mathrm{u}) = 0$, то $\mathrm{F}(\mathrm{u}) = 0$ и $\mathrm{F}_{\beta}(\mathrm{u}) = 0$.

В случае если производится классификация документов по нескольким рубрикам, для получения сводных оценок качества рубрицирования применяются различные методы усреднения характеристик по всем рубрикам. Так как различные рубрики имеют разную частотность, выбор метода усреднения представляет собой отдельную задачу [79]. Приведем здесь два наиболее часто применяемых метода усреднения: microaverage и macroaverage.

Пусть для каждой рубрики $\mathrm{C}_1,\ldots,\mathrm{C}_n$ автоматически приписаны документы $\mathrm{u}_1,\ldots,\mathrm{u}_n$. Тогда сводные оценки полноты и точности можно определить как

$$\mathrm{p}_{macroavg} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\mathrm{u}_i \cap \mathrm{C}_i\right|}{\left|\mathrm{u}_i\right|},\ \mathrm{r}_{macroavg} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\mathrm{u}_i \cap \mathrm{C}_i\right|}{\left|\mathrm{C}_i\right|} \qquad \text{(макроусреднение)},$$

$$\mathrm{p}_{microavg} = \frac{\sum_{i=1}^{n}\left|\mathrm{u}_i \cap \mathrm{C}_i\right|}{\sum_{i=1}^{n}\left|\mathrm{u}_i\right|},\ \mathrm{r}_{microavg} = \frac{\sum_{i=1}^{n}\left|\mathrm{u}_i \cap \mathrm{C}_i\right|}{\sum_{i=1}^{n}\left|\mathrm{C}_i\right|} \qquad \text{(микроусреднение)}.$$

Аналогично можно определить макро- и микроусредненные оценки *F*. Макроусреднение применяется чаще, так как отражает поведение метода в среднем по рубрикам.

2.3 Оценки метода машинного обучения на коллекции документов

При решении задачи классификации текстов методами машинного обучения типичной является ситуация, когда имеется готовая коллекция отрубрицированных текстов, на которой нужно произвести обучение алгоритма. При этом необходимо получить некоторые оценки качества рубрикации, которые можно будет использовать для сравнения различных методов и оптимизации параметров метода.

Важно отметить, что эти оценки качества нельзя получить, проверяя метод на коллекции документов, которая была использована для обучения. Иначе можно получить слишком завышенные оценки качества классификации. Кроме того, можно создать простейший алгоритм, который при оценке на коллекции документов для обучения будет давать 100% полноты и точности, и не будет работать на новых документах, на которых он не обучался. Такой алгоритм просто "запоминает" все полученные в процессе обучения документы вместе с соответствующими рубриками и сравнивает документы для рубрицирования с запомненными.

Обычно для оценки качества коллекцию отрубрицированных документов разбивают на две части: *обучающее* (*тренировочное*) множество и *тестовое* (*проверочное*) множество. Алгоритм обучают на тренировочном множестве. Обученный алгоритм применяют к тестовому множеству и вычисляют на тестовом множестве метрики качества рубрицирования (полноту, точность и т.п., раздел 2.2).

Естественно, что качество рубрицирования зависит от того, как было разбито множество отрубрицированных документов на обучающее и тестовое множество. Здесь важно отметить два момента:

- Чем больше обучающее множество, тем лучше можно обучить алгоритм. В то же время, на малом тестовом множестве оценки качества могут быть слишком грубыми.
- Специально подобранное разбиение отрубрицированных документов может сильно повлиять на результаты и привести к повышению или, наоборот, понижению оценок качества.

Обычно для опытов по сравнению различных алгоритмов машинного обучения разбиение выполняют случайно либо по некоторому признаку, не зависящему от содержания документа (например, дате). Для опытов по сравнению различных методов обучения разбиение фиксируют. Для коллекции документов Reuters-21578 [70], например, существуют фиксированные разбиения на обучающее и тестовое множество, которые рекомендуют использовать разработчикам методов машинного обучения для тестирования своих методов и опубликования экспериментов по сравнению с другими методами. Эти разбиения описаны в документе, сопровождающем коллекцию Reuters [70].

Кроме тестирования методов на фиксированном разбиении часто используется метод усреднения метрик качества по различным разбиениям. Такой метод называется *кросс-валидацией*. Опишем алгоритм кросс-валидации:

1. Множество отрубрицированных документов (пусть их N) разбивается на k частей (k - параметр кросс-валидации).
2. Для i от 1 до k
    2.1. Составляется тестовое множество из i-й части (в нем N/k документов).

2.2. Составляется обучающее множество из всех остальных документов (в нем N-N/k документов).

2.3. Алгоритм обучается на обучающем множестве, и вычисляются оценки качества работы на тестовом множестве.

3. Вычисляются усредненные оценки качества по всем k тестам.

Алгоритм кросс-валидации позволяет получить весьма точные оценки качества работы алгоритма. Чем больше k, тем больше получается множество для обучения и выше качество работы. С другой стороны, чем больше k, тем ниже скорость работы.

В конференциях по оценке методов классификации текстов, таких как TREC и РОМИП, применяется фиксированное разбиение множества документов. Участники получают отрубрицированную коллекцию документов для обучения плюс коллекцию документов, которые необходимо отрубрицировать (без указания классификации). После того, как участники присылают результаты классификации, оргкомитет вычисляет оценки качества рубрицирования и публикует результаты.

2.4 Обзор публикаций, посвященных практическому сравнению методов машинного обучения

Задача сравнения различных методов классификации текстов очень важна с практической точки зрения. Существует множество проблем, которые приходится решать для получения достоверных результатов сравнения. Одной из таких проблем является выбор коллекции документов, на которой должно производиться сравнение. Эта коллекция документов должна обладать следующими свойствами:

1. Достаточно большой объем отрубрицированных документов.
2. Высокое качество рубрикации (малое количество ошибок).
3. Доступность коллекции.

Некоторым стандартом сейчас считается коллекция документов Reuters [70]. Опишем некоторые результаты сравнения методов машинного обучения на задаче классификации текстов.

В статье [63] сравниваются следующие методы машинного обучения на задаче классификации текстов коллекции Reuters:

- метод Байеса
- метод Роше
- деревья решений C4.5
- метод k-ближайших соседей
- SVM с различными функциями ядра

Результаты показали, что метод SVM имеет преимущество над другими методами машинного обучения.

В статье [58] исследуются различные методы усечения пространства признаков, и производится сравнение методов машинного обучения на задаче классификации текстов коллекции Reuters. Сравниваются следующие методы:

- FindSimilar (аналог k-ближайших соседей)
- метод Байеса
- Байесовы сети
- деревья решений
- SVM

Результаты показали, что метод SVM имеет преимущество над другими исследуемыми методами машинного обучения.

В статье [80] производится сравнение нескольких методов машинного обучения на той же коллекции документов Reuters:

- метод Байеса
- метод k-ближайших соседей
- LLSF (линейная регрессия)
- нейронные сети
- SVM

В этой статье автор отмечает, что его результаты рубрицирования отличаются от опубликованных в [63]. А именно: результаты рубрицирования SVM несколько хуже, а результаты рубрицирования при помощи метода Байеса и метода k-ближайших соседей лучше, чем опубликованные в [63]. Тем не менее, выводы делаются те же: SVM имеет (хоть и небольшое) преимущество перед другими методами машинного обучения.

Также стоит отметить статью [68]. Автор этой статьи участвовал в конференции TREC-2001 и получил высокие результаты в конкурсе batch filtering (классификация текстов). В предисловии автор статьи пишет следующее: "Моя цель в TREC-2001 была проста: запустить задания по некоторым конкурсам (чтобы поучаствовать в конференции), потратить минимум времени (так как я был занят в этом году большим проектом) и получить достойный результат (маркетинг!)". Льюис использовал программу SVM_light с небольшими модификациями и получил то, чего добивался. В трех номинациях результаты Льюиса были лучшими на большинстве рубрик.

2.5 Обзор методов машинного обучения

Важным этапом при решении задачи классификации текстов является выбор метода машинного обучения, который будет применяться к векторному представлению документов.

Методы классификации объектов, основанные на обучении, впервые введены в рассмотрение в 1960-е годы [48, 52, 19, 35]. В настоящее время

разработано множество методов машинного обучения, которые применяются при решении широкого круга задач [78, 27, 36, 23, 26, 53]. Многие из этих методов применялись для решения задач классификации текстов. Рассмотрим основные методы машинного обучения, для которых опубликованы результаты применения для задач классификации текстов.

2.5.1 Метод Байеса

Метод Байеса основан на анализе совместных распределений признаков документа и категорий [80]. Документу $D = \langle d_1, d_2, \ldots, d_n \rangle$ сопоставляется наиболее вероятная апостериори категория по формуле

$$c^* = \arg\max_{c \in C} P(c \mid x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n) \qquad (2.4)$$

В задаче классификации текстов метод Байеса применяется отдельно для каждой категории и принимается решение, принадлежит документ категории или нет.

Апостериорная вероятность принадлежности документа рубрике вычисляется по формуле Байеса, связывающей априорную вероятность с апостериорной:

$$P(c \mid x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n) = \frac{P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n)} \qquad (2.5)$$

Подставляя (2.5) в (2.4), получаем:

$$c^* = \arg\max_{c \in C} \frac{P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n)} \qquad (2.6)$$

Так как знаменатель не зависит от категории, его можно сократить:

$$c^* = \arg\max_{c \in C} P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n \mid c) \cdot P(c) \qquad (2.7)$$

Условные вероятности $P(x_1 = d_1, x_2 = d_2, \ldots, x_n = d_n \mid c)$ можно вычислить в предположении условной независимости переменных $x_1, x_2, \ldots, x_n$. В этом случае, формула для определения наиболее вероятной категории будет выглядеть следующим образом:

$$c^* = \arg\max_{c \in C} P(c) \cdot \prod_{i=1..n} P(x_i = d_i \mid c) \qquad (2.8)$$

Для коллекции обучающих документов вероятности $P(x_i = d_i \mid c)$ вычисляются по формуле

$$P(x_i = d_i \mid c) = \frac{|\{D \in Ex \mid c \in Rub(D) \wedge D_i = d_i\}| + 1}{|\{D \in Ex \mid c \in Rub(D)\}|} \qquad (2.9)$$

Добавление единицы в числителе нужно для того, чтобы документы, содержащие не встречающиеся больше нигде признаки (например, уникальные слова), имели отличную от нуля вероятность для всех рубрик. В статье [64] рассматриваются различные формулы для аппроксимации $P(x_i = d_i \mid c)$ и их влияние на качество рубрикации.

Конечно, предположение о независимости переменных $x_1, x_2, \ldots, x_n$ является слишком сильным (поэтому метод Байеса иногда называют «наивным» — naive bayes classifier). На самом деле это предположение практически никогда не выполняется. Тем не менее, метод Байеса дает на удивление высокие результаты в задаче классификации текстов. [80, 79, 63].

Метод Байеса обладает высокой скоростью работы и простотой математической модели. Этот метод часто используется в качестве базового метода при сравнении различных методов машинного обучения.

2.5.2 Метод k-ближайших соседей

Метод k-ближайших соседей (k-nearest neighbours, k-NN), в отличие от других, не требует фазы обучения. Для того чтобы найти рубрики, релевантные документу *d*, этот документ сравнивается со всеми документами

из обучающей выборки. Для каждого документа *e* из обучающей выборки, находится расстояние - косинус угла между векторами признаков:

$$\rho(d,e) = \cos(d,e)$$

Далее из обучающей выборки выбираются k документов, ближайших к *d* (k - параметр). Для каждой рубрики вычисляется релевантность по формуле

$$s(c_j,d) = \sum_{e\in\{\text{k ближайших соседей}\}\wedge c_j\in Rub(e)} \cos(d,e)$$

Рубрики с релевантностью выше некоторого заданного порога считаются соответствующими документу. Параметр k обычно выбирается в интервале от 1 до 100.

Данный метод показывает довольно высокую эффективность [80], но требует довольно больших вычислительных затрат на этапе рубрикации.

2.5.3 Rocchio classifier

Классификатор Роше (Rocchio) — один из самых простых методов классификации. Для каждой категории вычисляется взвешенный центроид по формуле

$$\vec{g}_c = \frac{1}{|R_c|}\sum_{d\in R_c}\vec{d} - \gamma\frac{1}{|R_{c,k}|}\sum_{d\in R_{c,k}}\vec{d}$$

Здесь R_c - множество документов, принадлежащих категории; $R_{c,k}$ - *k* документов, не принадлежащих категории, наиболее близких к центроиду $\frac{1}{|R_c|}\sum_{d\in R_c}\vec{d}$; γ — параметр, указывающий относительную важность учета отрицательных примеров (обычно используется $\gamma < 1$).

После вычисления взвешенных центроидов для каждой категории, классификатор Роше определяет принадлежность документа рубрике при помощи вычисления расстояния между вектором обрабатываемого документа и центроидом каждой рубрики. Полученное расстояние

сравнивается с заданным порогом. В качестве функции расстояния часто используется косинус между векторами.

Данный метод обладает полезной особенностью: взвешенные центроиды можно быстро пересчитать при добавлении новых отрубрицированных примеров. Эта особенность полезна, например, в задаче адаптивной фильтрации, когда пользователь постепенно указывает системе, какие документы выбраны правильно, а какие нет. В ответ система может уточнить результаты, учитывая новые отрубрицированные документы.

Существует множество различных модификаций данного метода. В связи со своей простотой, данный метод часто используется в качестве базового метода для сравнения с другими методами.

2.5.4 Нейронные сети.

Искусственные нейронные сети (ИНС) - это большой класс систем, архитектура которых имеет аналогию с построением нервной ткани из нейронов [62, 33]. ИНС состоит из набора «нейронов», соединенных между собой. Каждый нейрон представляет собой элементарный преобразователь входных сигналов в выходные. Выходные сигналы вычисляются как функция от входных сигналов. Как правило, передаточные функции всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Некоторые входы нейронов помечены как внешние входы сети, а некоторые выходы - как внешние выходы сети. Подавая любые числа на входы сети, мы получаем какой-то набор чисел на выходах сети. Таким образом, работа нейросети состоит в преобразовании входного вектора в выходной вектор, причем это преобразование задается весами сети.

Для того чтобы сеть решала заданную функцию, ее надо "натренировать" на данных, для которых известны и значения входных параметров, и правильные ответы на них. Тренировка состоит в подборе

весов межнейронных связей, обеспечивающих наибольшую близость ответов сети к известным правильным ответам.

Нейронные сети имеют очень широкий спектр применения. Архитектура ИНС позволяет эффективно распараллеливать процесс обучения и применения НС. Имеется ряд экспериментов по использованию нейронных сетей для классификации текстов. В статье [80] отмечается очень долгое время обучения ИНС. Это связано с тем, что для задач высокой размерности требуется ИНС с большим количеством узлов.

2.5.5 Деревья решений.

Деревья решений (decision trees) разбивают данные на группы на основе значений переменных пространства признаков, в результате чего возникает иерархия операторов "ЕСЛИ-ТО", которые классифицируют данные [33, 25]. Для принятия решения, к какой категории отнести данный документ, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня. Вопросы имеют вид "значение переменной x_i больше порога b_i?". Если ответ положительный, осуществляется переход к правому узлу этого дерева, если отрицательный - к левому узлу. Затем следует вопрос, связанный с соответствующим узлом.

Для автоматического построения деревьев решений при помощи обучения на примерах разработан ряд алгоритмов [73, 33]. Рассмотрим один из таких алгоритмов - CLS [24]. Этот алгоритм циклически разбивает обучающие примеры на классы в соответствии с переменной, имеющей наибольшую классифицирующую силу. Каждое подмножество примеров, выделяемое такой переменной, вновь разбивается на классы с использованием переменной с наибольшей классифицирующей способностью и т.д. Разбиение заканчивается тогда, когда в подмножестве оказываются лишь элементы из одного класса. В ходе процесса образуется дерево решений.

Для определения переменной с наибольшей классифицирующей силой используется критерий информационного веса слова в рубрике (раздел 2.1.3 формула (2.2)).

Обычно после построения «точного» дерева решений к полученному дереву применяются различные процедуры усечения и преобразования дерева для того, чтобы обеспечить баланс между сложностью дерева (количеством узлов) и качеством обучения. Классическим подходом для преобразования деревьев решений является алгоритм C4.5 [73].

Как уже было отмечено, сложность интерпретации результатов рубрицирования является одним из характерных недостатков для методов машинного обучения. Деревья решений являются приятным исключением: построенное дерево легко поддается анализу, результат работы алгоритма можно интерпретировать в наглядных терминах. Существуют программы наглядного графического отображения деревьев решений.

В то же время, некоторые недостатки деревьев решений мешают эффективному применению алгоритмов, основанных на деревьях решений, для автоматической классификации текстов.

Один из известных недостатков деревьев решений именуется проблемой повторения (replication problem) [72]. Пусть рубрика описывается простой формулой вида

$$C = (A \cap Б) \cup (B \cap Г) \qquad (2.10)$$

где А, Б, В и Г — некоторые множества документов, соответствующие одному из признаков. В этом случае дерево решений для рубрики С (описывающее рубрику без ошибок) будет обязательно повторять некоторые из элементов А, Б, В, Г (см. рис 1). Всего в дереве решений для формулы (2.10) имеется 6 узлов. Для более сложных дизъюнктивных формул,

аналогичных (2.10), проблема повторения узлов усугубляется: размер дерева решений может экспоненциально расти при увеличении длины формулы.

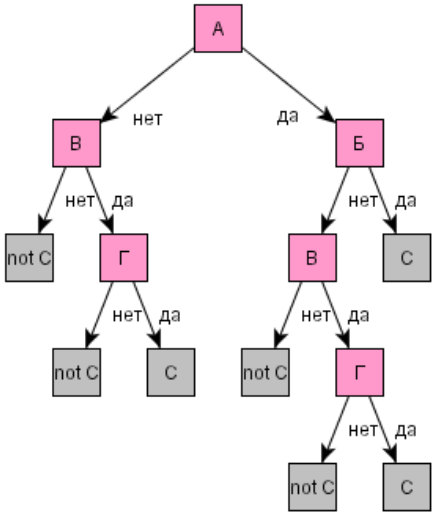

Рис. 1. Дерево решений для формулы (2.10). Листья В и Г повторяются по два раза.

Ещё одним недостатком деревьев решений для задач классификации текстов является тот факт, что алгоритм построения деревьев решений даёт одинаковый вес «положительным» и «отрицательным» ветвлениям в узлах. Большое количество «отрицательных» веток в описании рубрики может приводить к трудно интерпретируемым правилам и «переобучению» алгоритма классификации [72].

2.5.6 Построение булевых функций

Этот класс алгоритмов машинного обучения строит правила классификации в виде «если выполняется формула, то рубрика А». К основным методам построения правил классификации можно отнести

- построение правил вывода на основе деревьев решений;
- методы ограниченного перебора для правил заданного вида.

Первым способом является построение правил на основе деревьев решений. Действительно, каждому пути по дереву решений от корня до листа соответствует правило вида

$$\bigcap_i f_i^{\nu_i} \to \tilde{\mathrm{C}} \qquad (2.11)$$

где f_i — некоторый признак (значение булевской переменной), $\nu_i \in \{0,1\}$ (то есть f_i может входить в формулу с отрицанием или без), $\tilde{\mathrm{C}}$ — некоторая рубрика или множество документов, к которым не относится ни одной рубрики. Таким образом, для одной рубрики можно преобразовать дерево решений в следующую формулу:

$$\mathrm{C} = \bigcup_j \bigcap_i f_{i,j}^{\nu_{i,j}} \qquad (2.12)$$

где в правой части объединены все пути к листьям, для которых дерево решений выводит данную рубрику. Такой подход, однако, обладает теми же недостатками, что и деревья решений: существует проблема повторения правил и переобучения за счёт большого веса отрицательных ветвей (см. раздел 2.5.5).

Другим способом построения правил является перебор всевозможных правил в виде формул заданного вида. В нашей стране впервые методы такого типа были разработаны М.М. Бонгардом в 1960-е годы [23, 26, 45]. В алгоритме «Кора» выполняется полный перебор формул определённого вида и отбираются формулы, подходящие для описания заданного класса объектов.

В статье [72] описывается алгоритм построения формул вида $\mathrm{C} = \bigcup_j \bigcap_i f_{i,j}^{\nu_{i,j}}$ для задач медицинской диагностики. В описываемой задаче пространство признаков состоит из различных симптомов и медицинских показателей и имеет размерность порядка одного-трёх десятков. Приводится алгоритм перебора различных формул и оценивается время его работы.

Количество шагов алгоритма, требуемое для полного перебора, экспоненциально зависит от размерности пространства признаков.

В книге [33] также отмечается, что алгоритмы подбора логических правил заданного вида требуют экспоненциального перебора вариантов.

Нам представляется, что построение логических правил описания рубрики является эффективным подходом к проблеме автоматической классификации текстов, но существующие алгоритмы не подходят для этого в силу специфики задачи. Далее (в главе 3) мы опишем разработанный нами алгоритм построения формул описания рубрики, использующий частичный перебор вариантов.

2.5.7 Support Vector Machines.

Метод опорных векторов (Support Vector Machines, SVM) разработан В. Вапником на основе принципа структурной минимизации риска — одновременного контроля количества ошибок классификации на множестве для обучения и «степени обобщения» обнаруженных зависимостей [78, 55, 27].

В наиболее простом случае метод SVM заключается в нахождении гиперплоскости в пространстве признаков, разделяющей $\mathbf{R}^n$ на две части: в одной находятся все положительные примеры (документы, принадлежащие рубрике), а в другой — все отрицательные примеры (документы, не принадлежащие рубрике). При этом среди всех таких гиперплоскостей находится та, для которой минимальное расстояние (зазор) до ближайших примеров максимально.

Нахождение оптимальной плоскости методом SVM сводится к решению оптимизационной задачи с линейными ограничениями типа равенств и неравенств [78]:

$$L_D(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \to \max$$
$$0 \le \alpha_i \le C \qquad (2.13)$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0$$

Здесь $K(x_i, x_j)$ - функция ядра SVM, которая в простейшем случае равна евклидову скалярному произведению векторов x_i и x_j. Для задачи (2.13) предложены эффективные методы решения [65, 66].

Существуют также обобщения метода SVM для случая, когда разделяющей гиперплоскости не существует. В этом случае SVM ищет разделяющую гиперплоскость, одновременно минимизируя количество ошибок и максимизируя зазор между разделяющей гиперплоскостью и ближайшими примерами. SVM также может искать отделяющее рубрику правило в классе нелинейных нелинейных разделяющих поверхностей. Для этого используются нелинейные функции ядра [78, 55].

Метод SVM работает с абстрактной векторной моделью предметной области. Это позволяет применять SVM для решения различных задач машинного обучения. SVM используется для задач распознавания образов, распознавания речи, классификации текстов.

2.5.7.1 Программные реализации SVM

Существуют готовые реализации алгоритма SVM. Достаточно подробный список можно найти в интернете на http://www.kernel-machines.org/software.html. Один из свободно распространяемых пакетов, SVM_light [65], специально приспособлен для задач высокой размерности, которые возникают в задаче автоматической классификации текстов. Этот

пакет использовался несколькими исследователями для обработки текстовых массивов [63, 68].

Для наших исследований мы использовали SVM_light v. 3.50 [65].

2.5.7.2 Оптимизация параметров SVM

В 2002 году нами были проведены исследования по улучшению результатов работы SVM для задач автоматической классификации текстов, которые показали, что для практического применения необходимо оптимизировать стандартную методику применения SVM. В статьях [6, 11] описывается метод оптимизации параметров SVM, разработанный нами для улучшения скорости и качества классификации документов методом SVM.

Эксперименты показали, что лишь параметр «относительный вес ошибок 1-го и 2-го рода» существенно влияет на качество классификации, а другие параметры не существенно влияют на качество классификации текстов.

Был разработан метод оптимизации параметра «относительный вес ошибок 1-го и 2-го рода» (будем обозначать его *j*), основанный на переборе значений данного параметра в некотором интервале. Были получены экспериментальные оценки зависимости границ оптимального интервала перебора от количества релевантных (pos_ex) и нерелевантных (neg_ex) рубрике документов в коллекции для обучения:

$$\tilde{j} \in \left[1, \max\left(1.5\frac{\text{neg\_ex}}{\text{pos\_ex}}, 1\right)\right] \qquad (2.14)$$

В результате получено заметное (~10%) улучшение качества классификации по сравнению с другими методами [68] оптимизации параметров SVM.

В наших экспериментах мы использовали указанный метод оптимизации параметра j следующим образом:

1. Коллекция отрубрицированных документов разбивалась на две подколлекции — для обучения (70% документов) и для оценки качества результатов (30% документов).
2. Осуществлялся перебор из 10 значений параметра j в интервале (2.14). Для каждого j SVM обучалась на подколлекции обучения. Обученная SVM применялась к подколлекции тестирования и вычислялись значения полноты, точности и F-меры рубрицирования на подколекции тестирования. Определялось оптимальное значение $\tilde{j}$, для которого достигался максимум F-меры.
3. SVM с параметром $j = \tilde{j}$ обучалась на всей коллекции отрубрицированных документов и применялась к документам, которые необходимо отрубрицировать.

SVM применялась только для тех рубрик, для которых было не менее четырёх документов в коллекции для обучения.

Для того чтобы проверить корректность полученных результатов, были также воспроизведены эксперименты [63, 58] по применению SVM для рубрицирования документов из коллекции Reuters21578 [70]. Результаты рубрицирования согласуются с опубликованными данными.

2.6 Обзор методов, основанных на знаниях

В основе идеологии инженерного подхода лежит убежденность в том, что рубрикатор создается осмысленно. То есть за каждой рубрикой лежит обычно некий раздел области деятельности, который может быть представлен небольшим вербальным описанием. Задача эксперта состоит в том, чтобы составить описание рубрики на некотором формальном языке. Полученное описание затем используется для автоматического отнесения документов к той или иной рубрике.

В этом разделе мы представим обзор применяемых на практике методов классификации, основанных на знаниях. К сожалению, для методов этого класса существует значительно меньше публикаций в научной литературе, а имеющейся информации не всегда достаточно для подробного анализа применяемых методов [41]. Поэтому для некоторых систем [82, 60] представляемый обзор будет очень кратким.

Недостаток публикаций связан в первую очередь с тем, что применение методов, основанных на знаниях, требует значительного вклада труда экспертов на описание каждой рубрики, и поэтому применяется в основном лишь в коммерческих системах классификации. Кроме того, в отличие от методов машинного обучения, классификация с использованием знаний экспертов специфична для каждой предметной области и методы классификации текстов, основанные на знаниях, не могут применяться, например, для классификации графических изображений. В то же время, существуют обширные исследования методов построения экспертных систем для других областей знания [28, 38, 49].

Мы рассмотрим три системы классификации, основанные на знаниях:

1. технологию классификации компании LexisNexis;
2. технология классификации, разработанную для агентства новостей Reuters;
3. технологию классификации, разработанную в рамках проекта УИС РОССИЯ.

Методы классификации в УИС РОССИЯ мы рассмотрим подробно так как, во-первых, для неё имеются достаточно подробные описания, и, во-вторых, далее наше исследование будет базироваться на этих методах.

2.6.1 Технология классификации LexisNexis

В статье [82] описывается технология автоматического рубрицирования документов, применяемая в большой электронной

библиотеке LexisNexis (http://www.lexisnexis.com). В описываемой технологии в основном используется ручное описание рубрик. Гибкий механизм описания правил приписывания рубрик поддерживает следующие поисковые возможности:

- Поиск слов и словосочетаний, без учета морфологического словоизменения (для английского языка)
- Учет частотности слов с ручным заданием пороговых значений
- Учет местоположения слов в документе: заголовок, аннотация, текст в начале документа, основной текст
- Поддерживается задание круга источников получения информации для данной рубрики

Описание одной рубрики требует 4-8 часов ручной работы без учета времени тестирования. Для большинства рубрик качество классификации находится на уровне более 90% полноты и точности (к сожалению, в указанной статье методика получения данных оценок не приводится).

2.6.2 Технология классификации Reuters

В статье [60] описывается система классификации документов CONSTRUE/TIS (Categorization of News Stories, Rapidly, Uniformly and Extensibly/Topic Identification System). Система CONSTRUE/TIS использовалась агентством Reuters для классификации потоков новостных сообщений. Для отнесения документов к той или иной рубрике эксперт описывает правила в виде булевских формул. В качестве элементов выступают слова. Вот пример правила, описывающего рубрику wheat:

```
if      (wheat & farm) or
        (wheat & commodity) or
        (bushels & export) or
        (wheat & tonnes) or
        (wheat & whinter and (¬ soft))
then
        WHEAT
else
         (¬WHEAT)
```

2.6.3 Технология классификации документов на основе тезауруса УИС РОССИЯ

В статье [30] описывается технология классификации документов УИС РОССИЯ ([16, 34], http://www.cir.ru). Опишем (отчасти цитируя [30]) данный алгоритм классификации.

Для решения задачи рубрицирования по большим классификаторам в УИС РОССИЯ применяется комплекс из нескольких компонентов:

- большой лингвистический ресурс – тезаурус по общественно-политической тематике, специально предназначенный для автоматической обработки и автоматических выводов о содержании текста;
- специальное программное обеспечение АЛОТ (Автоматической Лингвистической Обработки Текста), позволяющее строить модель тематического содержания текста;
- специальная технология описания смысла рубрики посредством понятий тезауруса.

Алгоритм классификации УИС РОССИЯ показал высокую эффективность при создании систем рубрикации для различных текстовых

коллекций. Кроме того, описание рубрикатора посредством опорных понятий имеет следующие дополнительные преимущества:

- является прообразом свободного от субъективизма комментария к рубрикатору, который может пополняться и уточняться;
- позволяет в помощь эксперту для каждого термина в тексте определить список соответствующих рубрик;
- при выводе рубрики всегда можно показать/объяснить, почему была выведена та или иная рубрика, что позволяет быстро уточнять описание рубрик, анализируя замеченные ошибки рубрикации.

2.6.3.1 База знаний

Общественно-политический тезаурус РуТез (далее — Тезаурус РуТез) является основой тематического анализа в рамках Автоматизированной лингвистической обработки текстов (АЛОТ), используемого в УИС РОССИЯ. Тезаурус РуТез разработан [32] АНО Центр Информационных Исследований.

Общественно-политический тезаурус как ресурс для автоматической обработки текстов обладает следующими основными особенностями.

Во-первых, Тезаурус — это иерархическая сеть понятий, которая включает значительно больше понятий, отношений, синонимов, чем традиционные тезаурусы для ручного индексирования (75 тысяч терминов, 29 тысяч понятий, связанных более чем 100 тысячами непосредственных отношений в Общественно-политическом тезаурусе против 9.8 тысяч терминов, 6.8 тысяч понятий, связанных 15 тысячами отношениями в близком по тематике тезаурусе LIV — тезаурусе исследовательской службы Библиотеки Конгресса США [67]). Во-вторых, важнейшей особенностью является интеграция Тезауруса в процесс автоматической обработки текстов, что позволяет организовать обратную связь, анализируя результаты обработки.

Понятийная сеть Тезауруса включает до 10 уровней иерархии. Ценным свойством Тезауруса является возможность использовать транзитивность иерархических отношений (с учетом иерархии – 850 тысяч отношений, то есть в среднем каждое понятие связано с 28 другими).

2.6.3.2 Тематическое представление содержания документа

Значимость термина для содержания текста определяется в результате построения так называемого *тематического представления текста*, слабо зависящего от величины и типа текстов. Основные этапы построения тематического представления текста таковы (подробнее см. [30, 39]):

- Сопоставление текста с Тезаурусом создает для текста «понятийный индекс», в котором указывается, какие *понятия* Тезауруса и в каком месте текста обнаружены;
- Для каждого понятия текста нахождение по тезаурусным связям тематически близких понятий и отражение этой информации в так называемой тезаурусной проекции текста;
- Использование связей понятий в тезаурусной проекции для разрешения многозначности терминов;
- Построение «текстовых связей» для каждого понятия текста, то есть фиксация для каждого вхождения каждого понятия трех соседних понятий вправо и трех влево. Выбор таких цифр величина экспериментальная, однако, согласуется и с экспериментами в области исследования кратковременной памяти;
- Построение *тематических узлов* — групп близких по смыслу понятий. Тематические узлы строятся для тех понятий, которые отличаются своей частотностью или местоположением в заголовках, начале текста;
- Выбор среди построенных тематических узлов основных тематических узлов, то есть тех, которые моделируют элементы основной темы

текста. Выбор производится на основе анализа суммированных текстовых связей тематических узлов.

В зависимости от того, элементом какой структуры тематического представления оказывается понятие *d* тезауруса, формируется оценка значимости *ω(d;D)*. Типичные значения — 0.9 для центра основного тематического узла, 0.7 — для элемента основного тематического узла, 0.75 — для центра *«локального тематического узла»* и т.д. Окончательно вес понятия для текста определяется добавлением стабилизирующего фактора, учитывающего частотность понятия в документе:

$$\theta(d) = \alpha \cdot \omega(d;D) + (1-\alpha) \cdot \frac{freq(d;D)}{\max\limits_{c \in D} freq(c;D)}, \qquad (2.15)$$

$$\alpha = 0.7$$

2.6.3.3 Описание смысла рубрики понятиями тезауруса

Каждая рубрика *C* описывается дизъюнкцией альтернатив, каждый дизъюнкт D_i представляет собой конъюнкцию:

$$R = \bigcup_i D_i = \bigcup_i \left[\bigcap_j K_{ij} \right] = \bigcup_i \left[\bigcap_j \left(\bigcup_k d_{ijk} \right) \right] \qquad (2.16)$$

Конъюнкты $K_{i,j}$ в свою очередь описываются экспертами с помощью так называемых «опорных» понятий тезауруса $d_{i,j,k}$. Для каждого опорного понятия задается правило его расширения *f(·)*, определяющее каким образом вместе с опорным понятием учитывать подчиненные ему по иерархии понятия. Выделяются три случая — без расширения, полное расширение по дереву иерархии тезауруса и расширение только по родо-видовым связям.

Опорный концепт может быть как «положительным», который добавляет нижерасположенные понятия в описание конъюнкта, так и «отрицательным», который вырезает свои подчиненные понятия. Последовательность учета положительных и отрицательных опорных

понятий регулируется заданием специального атрибута. Результатом применения расширения опорных понятий является совокупность понятий тезауруса, полностью описывающая конъюнкт.

Следует подчеркнуть, что в данной методологии достаточно хранить только опорные понятия, полное же описание рубрики может быть каждый раз пересчитано заново при изменении тезауруса.

Типичные цифры о параметрах описания: на одну рубрику рубрикатора в среднем приходится 1-2 дизъюнкта, 2-3 конъюнкта, 10-20 опорных понятия («положительных» и «отрицательных»), 200-400 понятий полного описания, то есть 400-800 текстовых входов.

2.6.3.4 Автоматическое рубрицирование на тематическом представлении

Оценка релевантности содержания текста рубрике (вес рубрики) рассчитывается путём соотнесения документа с булевской формулой описания рубрики (2.16), с учётом информации о весах понятий в тексте, входящих в описание рубрики.

В УИС РОССИЯ вес конъюнкта рассчитывается по формуле:

$$\theta(K_{i,j}) = \min\left\{ 1.0; \max\left(\theta\left(d_{i,j,k}\right), \chi \cdot \theta\left(p_{i,j,m}\right)\right)\right\} \qquad (2.17)$$

где $d_{i,j,k}$ — понятия, не требующие подтверждения; $p_{i,j,m}$ – понятия, требующие подтверждения; χ — множитель, равный единице, если имеются понятия, не требующие подтверждения, и нулю иначе.

Вес дизъюнкта предназначен учитывать не только сумму весов составляющих его конъюнктов, но и меру близости конъюнктов в тексте:

$$\theta\left(D_i\right) = \frac{\sum_{j=1}^{m}\theta\left(K_{i,j}\right) + \sum_{j<k} S\left(K_{i,j}, K_{i,k}\right)}{m + C_m^2} \qquad (2.18)$$

здесь

$$S\left(K_{i,j},K_{i,k}\right)=\min\left\{1.0;\frac{\sum s\left(c_{i,j,q}\in K_{i,j},d_{i,k,w}\in K_{i,j}\right)}{\max s\left(c\in D,d\in D\right)}\right\}$$

— сумма всех текстовых связей между понятиями одного конъюнкта и понятиями другого, деленная на значение максимальной текстовой связи между любыми двумя понятиями текста. Значение текстовых связей $s\left(c_1,c_2\right)$ между понятиями c_1 и c_2 зависит от того, насколько часто эти понятия встречаются в тексте рядом (в пределах «окна» заданного размера). Значение члена $S\left(K_{i,j},K_{i,k}\right)$ равно обычно единице для сильно связанных конъюнктов и принимает малое значение, если понятия различных конъюнктов обсуждались в разных местах текста.

Вес рубрики представляется максимумом весов входящих в описание рубрики альтернатив. В случае имеющихся иерархических связей между рубриками оценка релевантности нижестоящих рубрик переносится на вышестоящие рубрики. Так что при запросе по вышестоящей рубрике будут выходить и документы, к которым были приписаны нижестоящие рубрики.

Алгоритм рубрицирования работает следующим образом. Для всех понятий тезауруса, найденных в тексте, определяется множество рубрик, которые могут быть определены в тексте. Для каждой рубрики происходит расчет ее веса по формулам (2.15), (2.17) и (2.18). В результирующем множестве остаются рубрики, вес которых превосходит задаваемый заранее для коллекции порог.

Средняя скорость описания рубрик экспертами УИС РОССИЯ сравнительно невысока – 100-200 рубрик на одного эксперта в месяц. За счёт использования сокращенного описания рубрики с использованием опорных концептов эта скорость в 10 раз выше, чем скорость описания рубрик LexisNexis ([82], раздел 2.6.1), но все равно для больших рубрикаторов

требует значительных затрат времени экспертов. Требуется разработка менее трудоемких методов классификации текстов.

Данная технология была использована для построения систем автоматической рубрикации по большим рубрикаторам — до 3000 рубрик, в том числе:

- рубрикации правового законодательства РФ (3000 рубрик),
- рубрикации нормативных документов по президентскому классификатору нормативно-правовых актов [44] (1100 рубрик),
- рубрикации документов из сети интернет по тематике компьютерной безопасности,
- системы классификации текстов для ГАС «Выборы» (ЦИК РФ).

2.7 Выводы

Мы рассмотрели основные подходы, применяемые для автоматической классификации текстов. Детальное рассмотрение исследованных публикаций подтверждает наш тезис о том, что существует значительный разрыв в исследованиях и практических методах между методами классификации, основанными на машинном обучении и методами, основанными на знаниях. А именно:

- Методы, основанные на знаниях
  - практически не используют коллекции отрубрицированных текстов, если такая имеется;
  - требуют значительных ресурсов (времени экспертов) для построения правил описания рубрики.
- Наиболее эффективные методы машинного обучения
  - работают с абстрактной моделью предметной области и не учитывают особенностей задачи классификации текстов;
  - строят описание рубрики, не пригодное для анализа содержания рубрики и дальнейшего уточнения.

- Методы построения логических правил, основанные на машинном обучении
    - непригодны для задачи классификации текстов, так как используют перебор вариантов, возможный лишь для задач малой размерности.

Проведённый анализ различных методов позволяет нам выбрать для дальнейших исследований

- наиболее эффективный метод машинного обучения SVM в качестве отправной точки для сравнения качества работы других методов машинного обучения;
- метод классификации, используемый в УИС РОССИЯ, в качестве эффективного и доступного для исследования метода, основанного на знаниях.

3 Метод машинного обучения, основанный на моделировании логики рубрикатора

Методы построения классификаторов, используемые экспертами при инженерном подходе, подразумевают описание рубрики в виде правил относительно простого вида (см. раздел 2.6). Например, в УИС РОССИЯ [29] применяются булевские формулы фиксированной структуры, в качестве элементов запроса используются понятия Тезауруса РуТез [32]. Получаемые правила классификации имеют простой смысл и легко поддаются интерпретации.

В то же время, широко используемые алгоритмы машинного обучения получают представления рубрики, которые трудно или вообще невозможно понять и интерпретировать.

В основе идеологии инженерного подхода лежит убежденность в том, что рубрикатор создается осмысленно. То есть за каждой рубрикой лежит обычно некий раздел области деятельности, который может быть представлен небольшим вербальным описанием. Мотивацией для данной работы была необходимость создать алгоритм машинного обучения, который бы моделировал смысл рубрики, составленной человеком, по результатам рубрицирования. Необходимым требованием для данного алгоритма было построение правил описания рубрики, которые можно легко интерпретировать.

Данная постановка задачи отличается от классической задачи построения автоматической процедуры классификации текстов, максимизирующей метрики качества рубрицирования — полноту и точность. В нашем случае важной метрикой качества алгоритма является также экспертная оценка соответствия полученных правил классификации смыслу рубрики.

Построение алгоритма, который строит легко интерпретируемые правила описания рубрики важно с теоретической и практической точки зрения. Важным концептуальным результатом является создание алгоритма для извлечения знаний (моделирования логики экспертов) о структуре рубрикатора из коллекции отрубрицированных документов. В отличие от классической постановки задачи извлечения знаний[1] из массива структурированной информации рассматриваемый алгоритм работает с коллекцией неструктурированной информации — текстами на естественном языке.

Описываемый алгоритм имеет множество практических применений. Наглядное описание содержания рубрики, построенное по коллекции документов, можно рассматривать в качестве краткой аннотации коллекции документов, что позволяет анализировать структуру рубрикатора и выявлять особенности множества отрубрицированных документов. Можно анализировать логику работы экспертов, которые рубрицировали документы вручную и сравнивать логику работы разных экспертов.

Алгоритм позволяет использовать наглядные описания коллекции документов в сочетании с другими методами машинного обучения. Например, можно построить описание набора документов, ошибочно классифицированных некоторым методом машинного обучения. Это позволяет анализировать причины неустойчивой работы методов машинного обучения.

Для больших рубрикаторов сложной, иерархической структуры актуальной проблемой является документирование принципов отнесения

---

[1] Извлечение знаний из массива данных (Data Mining) — современное, бурно развивающееся направление в области обработки информации [33, 36]. Обычно под термином Data Mining рассматриваются задачи извлечения ранее неизвестных закономерностей из массива структурированной информации, например, из реляционной базы данных.

документов к той или иной рубрике. Автоматически построенные наглядные описания рубрик можно использовать в полуавтоматической процедуре составления «комментария» к рубрикатору, свободного от субъективности отдельных экспертов.

Автоматически построенное описание рубрики может также использоваться для помощи экспертам, которые составляют описания рубрик при инженерном подходе. Использование таких автоматических помощников позволяет повысить скорость и качество работы экспертов.

Структура этого раздела следующая: сначала мы конкретизируем постановку задачи и приведём описание алгоритма (раздел 3.1), затем опишем исследование алгоритма. Исследование алгоритма проводилось двумя способами:

- аналитическое исследование упрощенной (базовой) версии алгоритма, в предположении существования «идеальной» рубрики (раздел 3.2);
- экспериментальное исследование алгоритма на реальных задачах классификации текстов (раздел 3.3).

Экспериментальное исследование алгоритма проводилось на основе сравнения результатов работы нашего алгоритма с результатами работы других алгоритмов классификации текстов. Эксперименты проводились на различных коллекциях. Были опробованы различные модификации алгоритма.

3.1 Описание алгоритма ПФА (алгоритма построения формул)

В качестве основы для моделирования мы используем подход к описанию рубрики, используемый в УИС РОССИЯ (раздел 2.6.3, [29, 32]). Согласно этому подходу, описание рубрики экспертом представляется в виде

булевской формулы над понятиями тезауруса (2.16), которую можно интерпретировать как запрос к полнотекстовой информационной системе вида

$$\mathrm{U} = \bigcup_i \bigcap_j \left(\bigcup_k t_{i,j,k} \setminus \bigcup_l t'_{i,j,l} \right) \qquad (3.1)$$

где U — множество документов, принадлежащих рубрике, а $t_{i,j,k}$ и $t'_{i,j,l}$ — множество документов, содержащих некоторое понятие тезауруса. Выбор структуры формулы и понятий, включаемых в формулу, производится экспертом на основе знаний предметной области и, возможно, частичного анализа коллекции документов.

Задача моделирования логики рубрикатора при помощи машинного обучения, в нашем случае, состоит в построении формул вида (3.1) на основе анализа множества отрубрицированных документов. Основными требованиями для алгоритма являются:

1. высокое качество рубрицирования;
2. экспертная оценка качества полученных формул;
3. приемлемая скорость работы алгоритма.

Построение кратчайшей формулы вида (3.1), описывающей рубрику, представляется задачей, требующей перебора большого числа вариантов. Количество вариантов для полного перебора можно оценить как s^H, где s — размерность пространства признаков, а H — количество элементов в формуле. Для задачи классификации текстов типичная размерность пространства признаков — несколько тысяч, а формула может состоять из нескольких десятков элементов, поэтому такой перебор не представляется возможным[2].

[2] В разделе 2.5.6 приводится обзор статьи [72], где рассматривается задача построения формул подобного вида для медицинской диагностики. Пространство признаков в таком случае имеет размерность порядка 10 и перебор вариантов становится возможным.

Для решения поставленной задачи нами разработан алгоритм машинного обучения, который строит формулы описания рубрики в упрощенном виде, несколько отличающемся от (3.1), но также соответствующем логике построения рубрикатора. Различные модификации алгоритма строят формулы вида:

$$\text{a. } U = \bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i} t_{i,j} \text{ (алгоритм ПФА)} \qquad (3.2)$$

$$\text{b. } U = \bigcup_{i=1}^{k} \left(\left(\bigcap_{j=1}^{J_i} t_{i,j} \right) \setminus \bigcup_{m=1}^{M_i} t'_{i,m} \right) \qquad (3.3)$$

$$\text{c. } U = \left(\bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i} t_{i,j} \right) \setminus \left(\bigcup_{i=1}^{k} \bigcap_{j=1}^{J'_i} t'_{i,j} \right) \qquad (3.4)$$

где $t_{i,j}$, $t'_{i,j}$ — множества документов, содержащих некоторое понятие тезауруса (или, в общем случае, некоторый элемент векторного представления документов). Конъюнкции, составляющие формулу, имеют длину J_i от 1 до 3.

Отметим, что формулы вида (3.2) и (3.4) можно преобразовать к виду (3.1), а для формул вида (3.3) также возможно и обратное преобразование.

Основная версия алгоритма (формулы вида (3.2), будем называть этот алгоритм ***ПФА*** — от «алгоритм построения формул») соответствуют подбору нескольких альтернатив, в сумме описывающих смысл рубрики, каждая альтернатива может быть задана как пересечение нескольких независимых факторов. Ограничения на длину конъюнкции (от одного до трёх элементов) соответствует типичному размеру конъюнкций, используемых экспертами. Модификация алгоритма для построения формул вида (3.3) соответствует "уточнению" полученных конъюнктов при помощи отрицания. Модификация (3.4) является в некотором смысле альтернативой «набиранию/перебору» множества мелких частных случаев проявления

рубрики и позволяет учесть иерархию рубрик — правила вида «то, что относится к нижерасположенной рубрике, не входит в корневую рубрику».

Полученная алгоритмом формула может использоваться в качестве запроса к полнотекстовой информационной системе. При этом каждому элементу формулы — понятию тезауруса — сопоставляется множество документов, содержащих данное понятие.

Далее мы опишем алгоритм построения формулы описания рубрики, который строит описание последовательно, без использования значительного по объему перебора вариантов. Алгоритм состоит из 4 шагов, выполняемых последовательно. Первые два шага алгоритма допускают несколько вариантов реализации, и от этого зависит вид получаемых формул. Каждый следующий шаг не зависит от реализации предыдущего. Третий шаг является основным и наиболее трудоёмким. Сначала мы приведём описание основной версии алгоритма, которая строит формулы вида (3.2), а затем — модификаций алгоритма.

Описываемый алгоритм имеет ряд параметров, которые могут влиять на скорость вычисления и качество (полноту, точность, размер) получаемых формул.

3.1.1 Шаг 1: вычисление векторного представления

Первым шагом алгоритма является вычисление векторного представления текстов документов. В наших экспериментах мы использовали два различных векторных представления: словарное представление и тезаурусное представление. Опишем их по отдельности.

1. В словарном представлении все слова, встречающиеся в документах, были приведены к нормальной форме. Элементами формул являются множества документов, содержащих то или иное нормализованное слово.

2. В тезаурусном представлении для каждого документа строится терминологический индекс при помощи процедуры Автоматической Лингвистической Обработки Текстов [29, 30], разработанной в рамках проекта УИС РОССИЯ (см. раздел 2.6.3.2). Элементами формул являются множества документов, содержащих то или иное понятие тезауруса.

Алгоритм построения формул может также использовать другие векторные представления текстов. Однако стоит отметить, что малая длина формул описания рубрики достигается в методе классификации УИС РОССИЯ в значительной степени за счёт использования тезауруса. Поэтому для других векторных представлений эффективность алгоритма может быть ниже, так как предположение о существовании короткой формулы описания рубрики может не выполняться.

При описании алгоритма мы будем называть элементы формул *термами*, имея в виду элементы формул, используемые в векторном представления документов.

Термы, встречающиеся менее, чем в 5 документах, были усечены. Для каждого терма вычисляются полнота, точность и F-мера описания рубрики запросом, состоящим из одного только этого терма (описание метрик см. раздел 2.2).

3.1.2 Шаг 2: построение конъюнктов

Следующим шагом работы алгоритма является вычисление конъюнкций, состоящих из двух или трех различных термов. Целью данного шага является вычисление конъюнкций с высокими показателями полноты и точности. Алгоритм перебирает различные пары и тройки термов, и для каждого конъюнкта-кандидата вычисляется полнота и точность описания рубрики этим *конъюнктом*.

Для того чтобы сократить перебор, вычисляются только конъюнкты, состоящие из термов с высокими показателями полноты и точности. В зависимости от параметров алгоритма применяются различные стратегии усечения перебора:

1. перебираются все конъюнкции, состоящие из термов, полнота и точность которых выше некоторого порога;
2. перебираются все конъюнкции, состоящие из термов из списка первых N термов с наибольшим показателем F-меры.

Из списка полученных конъюнктов выкидываются те, для которых

a. полнота ниже некоторого порога;
b. точность ниже некоторого порога.

Например, в экспериментах на коллекции Reuters-21578 мы перебирали конъюнкции, состоящие из 100 термов с наибольшей F-мерой. Из списка полученных конъюнктов откидывали те, полнота которых ниже порога 20% или точность ниже порога 3%.

В результате этого шага алгоритма мы получаем расширенный набор свойств документов. Расширенный набор свойств состоит из *конъюнктов* — конъюнкций из одного, двух или трёх термов. Для каждого конъюнкта вычислены полнота, точность и F-мера описания рубрики запросом, состоящим из одного только этого конъюнкта.

Опишем некоторые возможные модификации шага 2, которые позволяют строить другие формулы описания рубрики.

3.1.2.1 Построение конъюнктов с учетом отрицаний

Одно из возможных расширений алгоритма ПФА строит формулы с отрицаниями вида (3.3). Для построения формул такого вида вводится новый шаг алгоритма, который строит конъюнкты вида

$$\mathrm{v}' = \left(\bigcap_{j=1}^{J_i} t_{i,j} \right) \setminus \bigcup_{m=1}^{M_i} t'_{i,m} \qquad (3.5)$$

где в скобках находится один из конъюнктов, построенных на шаге 2.

Целью этого шага является поиск конъюнктов, качество которых (задаваемое функцией F_β (2.3)) можно существенно улучшить при помощи добавления отрицаний. При помощи добавления отрицаний можно повысить точность конъюнкта за счёт уменьшения полноты.

Построение новых конъюнктов производится сразу после выполнения шага 2, и на последующих шагах алгоритма конъюнкты с отрицаниями участвуют в построении формулы наравне с обычными конъюнктами.

Опишем этапы построения конъюнктов с отрицаниями. Для некоторых параметров алгоритма укажем конкретные значения, которые мы использовали в экспериментах.

1. Выбор конъюнктов-кандидатов на улучшение. Такими конъюнктами считаются те, у которых высокая полнота (более 90%) и низкая точность (менее 90%). Чтобы сократить перебор, попытки улучшения производятся только для нескольких (например, 10) наилучших по F-мере кандидатов.
2. Выбор термов для исключения. Такими термами считаются все термы, частота которых выше порога 10 документов, и количество документов, не принадлежащих рубрике более 6:

$$K_{\text{minus\_elements}} = \left\{ t : \left| t \right| \geq 10 \;\&\; \left| t \setminus \mathrm{C} \right| \geq 6 \right\}$$

3. Для каждого конъюнкта-кандидата на улучшение (обозначим его v) производится поиск 10 наилучших термов для выкидывания:

$$\mathrm{v}' = \arg\max_{10,\ \mathrm{v}\setminus t} \left(\mathrm{F}(\mathrm{v}\setminus t) :\ t \in K_{\text{minus\_elements}} \ \&\ \left| t \cap (\mathrm{v}\setminus \mathrm{C}) \right| \geq 2 \ \&\ \mathrm{F}(\mathrm{v}\setminus t) > \mathrm{F}(\mathrm{v}) \right)$$

Если такие термы найдены, то v' добавляются к списку конъюнктов.

4. Шаги 1-3 повторяются несколько (не более 10) раз для того, чтобы построить конъюнкты, из которых удаляются несколько термов.

3.1.3 Шаг 3: построение дизъюнкции

Следующим шагом алгоритма является построение формулы в виде дизъюнкции элементарных конъюнкций. В качестве элементарных конъюнкций выступают термы и конъюнкты из списка вычисленных на шаге 2.

Сначала выбирается первый элементарный конъюнкт - "начало" формулы. Первый элементарный конъюнкт u_1 выбирается с максимальным значением функции

$$F_{1stDisj}(\mathrm{v}_j) = \frac{1}{\frac{\mathrm{w}_{\mathrm{p1}}}{\mathrm{p}(\mathrm{v}_j)} + \frac{1}{\mathrm{r}(\mathrm{v}_j)}} \qquad (3.6)$$

$$\mathrm{u}_1 = \arg\max_j \left(F_{1stDisj}(\mathrm{v}_j) \right)$$

где $\mathrm{p}(\mathrm{v}_j)$ и $\mathrm{r}(\mathrm{v}_j)$ — точность и полнота конъюнкта v_j соответственно. Данная формула — частный случай функции F-мера с весами (см. раздел 2.2). Для первого дизъюнктивного члена формулы точность более важна, чем полнота, поэтому вес полноты первого дизъюнкта w_{p1} нужно выбирать больше 1. В экспериментах на коллекции Reuters-21578 мы использовали значение $\mathrm{w}_{\mathrm{p1}} = 5$.

Далее формула наращивается постепенно новыми конъюнктами по шагам. На каждом шаге имеется текущая вычисленная формула и

рассматривается возможность улучшить эту формулу при помощи добавления нового члена в дизъюнкцию.

Пусть U_{i-1} — формула, вычисленная до шага $i \geq 2$, $U_1 = u_1$. Пусть C — множество документов, принадлежащих рубрике. Для каждого конъюнкта-кандидата v_j вычисляется:

1. *дополняющая точность*, то есть точность v_j на непокрытой части рубрики

$$p(v_j \mid U_{k-1}) = \frac{\left|(v_j \setminus U_{k-1}) \cap C\right|}{\left|v_j \setminus U_{k-1}\right|}$$

2. *дополняющая полнота*, то есть полнота v_j на непокрытой части рубрики:

$$r(v_j \mid U_{k-1}) = \frac{\left|(v_j \setminus U_{k-1}) \cap C\right|}{\left|C \setminus U_{k-1}\right|}$$

3. *функция качества конъюнкта*

$$F_{qual}(v_j \mid U_{i-1}) = \frac{w_{ap} + w_{ar} + w_r}{\frac{w_{ap}}{p(v_j \mid U_{k-1})} + \frac{w_{ar}}{r(v_j \mid U_{k-1})} + \frac{w_r}{r(v_j)}} \qquad (3.7)$$

где w_{ap}, w_{ar}, w_r — некоторые весовые коэффициенты, $r(v_j)$ — полнота конъюнкта v_j. Например, для коллекции Reuters-21578 мы использовали $w_{ap} = 10,\ w_{ar} = 5,\ w_r = 2$.

Наилучшим считается конъюнкт, функция качества которого максимальна.

Формула (3.7) задает критерий выбора оптимального конъюнкта для добавления в формулу. В формуле содержатся три коэффициента: *вес полноты* w_r (weight of recall), *вес дополняющей точности* w_{ap} (weight of

additional precision) и *вес дополняющей полноты* w_{ar} (weight of additional recall).

Вес полноты влияет на "качество" добавляемых элементов, то есть их соответствие общей тематике рубрики. Если положить этот вес равным нулю, то в момент, когда формулой описано уже более 90% документов рубрики, в качестве кандидатов на добавление будут попадать термы, слишком специфичные для оставшихся 10% документов. Эти термы могут не иметь отношения к тематике рубрики.

Вес дополняющей полноты влияет на скорость "сходимости" алгоритма, то есть на количество конъюнктов в результирующей формуле. Чем выше этот вес, тем короче получается формула. Если положить этот вес равным нулю, то будут добавляться новые конъюнкты, которые добавляют мало (1-10) новых документов к описанию рубрики.

Вес дополняющей точности влияет на качество классификации коллекции обучения при помощи полученной формулы. Чем больше этот параметр, тем большую точность можно будет получить при фиксированном значении полноты. С другой стороны, слишком высокое значение веса дополняющей точности может привести к переобучению, то есть к излишней подгонке формулы под заданную обучающую коллекцию.

Получаемые формулы зависят от отношения величин данных параметров, но не от абсолютных значений параметров. В разделе 3.2 мы получим оценки качества получаемой формулы в зависимости от соотношения данных параметров. В разделе 3.3 мы приведем примеры и покажем, как описанные параметры влияют на результаты работы алгоритма.

К текущей формуле добавляется конъюнкт, функция качества которого максимальна:

$$u_i = \arg\max_j \left(F_{qual}\left(v_j \mid U_{i-1} \right) \right)$$
$$U_i = U_{i-1} \bigcup u_i$$

Процесс повторяется до тех пор, пока не будет выполнено хотя бы одно из следующих условий остановки:

1. Величина $r\left(u_i \mid U_{i-1}\right) = 0$ для наилучшего конъюнкта равна нулю (нет улучшения полноты);
2. Количество дизъюнктов больше 20 (формула слишком сложна);
3. prec<10% и recl>90% (слишком маленькая точность);
4. recl>99% (достигнут хороший результат).

Пороги, указанные в пунктах 2-4, являются параметрами алгоритма. Их можно менять в зависимости от задачи.

3.1.4 Шаг 4: усечение формулы

Размер получающейся формулы, полнота и точность получающегося описания, и степень "подгонки" формулы под конкретную выборку документов зависят от соотношения параметров алгоритма. Наиболее важными являются параметры, встречающиеся в формуле (3.7).

При наращивании формулы дополнительными конъюнктами полнота растет, а точность - в целом убывает. Алгоритм останавливается тогда, когда либо рубрика покрыта практически полностью (99%), либо когда невозможно найти подходящий конъюнкт. Для получения формулы, реализующей оптимальное соотношение полноты и точности, мы усекаем полученную формулу до того места, где достигается максимум F-меры.

3.1.5 Построение формулы с отрицаниями

Одно из возможных расширений алгоритма ПФА строит формулы с отрицаниями вида (3.4):

$$\mathrm{U} = \left(\bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i} t_{i,j} \right) \setminus \left(\bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i'} t'_{i,j} \right)$$

то есть описание рубрики состоит из «положительной» части и описания «ошибок» рубрицирования. Описание рубрики U можно представить в виде разности двух формул вида (3.2), которые строит алгоритм ПФА:

$$\mathrm{U} = \mathrm{U}_1 \setminus \mathrm{U}_2$$

Для построения такой формулы сначала применяется алгоритм ПФА к документам рубрики C. Полученную формулу обозначим U_1. Затем составляется новая «псевдорубрика», состоящая из документов

$$\mathrm{C}' = \mathrm{U}_1 \setminus \mathrm{C}$$

К псевдорубрике C' применяется алгоритм ПФА (возможно, с другими параметрами алгоритма) и получается описание псевдорубрики U_2.

3.2 Аналитическое исследование алгоритма

В этом разделе мы исследуем работу алгоритма ПФА при некоторых предположениях относительно задачи и реализации алгоритма [3]. Мы рассмотрим некоторую «идеальную» ситуацию, когда рассматриваемая рубрика описывается некоторой формулой $\mathrm{U}^* = \bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i} t_{i,j}$ с полнотой и точностью, равной единице. Такая ситуация соответствует случаю, когда краткое вербальное описание рубрики может точно моделироваться булевской формулой. Кроме того, мы рассмотрим несколько упрощенную версию алгоритма, уменьшив в алгоритме ПФА количество параметров (порогов и весовых коэффициентов). Мы рассмотрим поведение алгоритма ПФА на шаге 3 — построении дизъюнкции, считая векторное представление

документа и набор конъюнктов фиксированными. Упрощенный алгоритм будем называть ***ПФБА*** — «построение формул, базовый алгоритм». Все результаты, верные для упрощенного алгоритма ПФБА, верны и для полной версии алгоритма при определенном задании параметров и предположении, что список конъюнктов, вычисленных на шаге 2 алгоритма ПФА, содержит все конъюнкты формулы U^*.

Основным результатом этого раздела является математически строгое доказательство того, что алгоритм ПФБА при условии существования точной формулы и достаточно «жестких» параметрах алгоритма получит хорошую формулу. Основная теорема этого раздела устанавливает связь между параметрами алгоритма и качеством i-го конъюнкта. Следствия из этой теоремы позволяют оценить скорость сходимости алгоритма и вычислить значения параметров алгоритма, для которых алгоритм за N шагов получит формулу, полнота и точность которой не менее $1-\eta$ (для любого наперед заданного параметра $\eta > 0$). Для получения оценок используются методы, аналогичные [18].

Структура данного раздела следующая:

1. в разделе 3.2.1 мы опишем алгоритм ПФБА и его отличие от «полной версии» алгоритма построения формул;
2. затем, в разделе 3.2.2 опишем некоторые свойства метрик, которые нужны для доказательства теоремы;
3. в разделе 3.2.3 приводятся основные результаты по аналитическому исследованию алгоритма построения формул.

3.2.1 Описание алгоритма ПФБА

Рассмотрим третий шаг алгоритма ПФА — построение дизъюнкции. Пусть задана рубрика C — непустое множество документов и множество конъюнктов

$$\mathrm{V} = \left\{ \mathrm{v}_j \right\},\ j = 1..s$$

Каждый конъюнкт представляет собой непустое множество документов. Алгоритм ПФА собирает формулу

$$\mathrm{U} = \bigcup_{k=1}^{N} \mathrm{u}_\mathrm{k},\quad \mathrm{u}_\mathrm{k} \in \mathrm{V}$$

последовательно выбирая конъюнкты по шагам.

Полная версия алгоритма предусматривает задание различных весовых коэффициентов для выбора первого и последующих конъюнктов. Это вызвано эвристическими соображениями: точность первого конъюнкта сильнее влияет на выбор всей формулы, чем точность последующих конъюнктов. Для аналитического исследования мы упростим алгоритм, и будем считать весовые параметры для выбора первого и последующих конъюнктов одинаковыми. Кроме того, вместо трёх весовых коэффициентов в формуле (3.7), рассмотрим их отношения:

$$\mathrm{F}_{\beta,\gamma}\left(\mathrm{v}_j \mid \mathrm{U}_{k-1}\right) = \frac{1+\beta+\gamma}{\dfrac{\beta}{\mathrm{p}\left(\mathrm{v}_j \mid \mathrm{U}_{k-1}\right)} + \dfrac{1}{\mathrm{r}\left(\mathrm{v}_j \mid \mathrm{U}_{k-1}\right)} + \dfrac{\gamma}{\mathrm{r}\left(\mathrm{v}_j\right)}},\ \beta = \frac{\mathrm{w}_{\mathrm{ap}}}{\mathrm{w}_{\mathrm{ar}}}, \gamma = \frac{\mathrm{w}_{\mathrm{r}}}{\mathrm{w}_{\mathrm{ar}}} \quad (3.8)$$

Обозначим текущую формулу, выбранную на шаге k как U_k. На нулевом шаге $\mathrm{U}_0 = \varnothing$. На шаге k выбирается элементарный конъюнкт

$$\mathrm{u}_\mathrm{k} = \arg\max_{\mathrm{v_j} \in \mathrm{V}} \left(\mathrm{F}_{\beta,\gamma}\left(\mathrm{v}_\mathrm{j} \mid \mathrm{U}_{\mathrm{k}-1}\right)\right) \quad (3.9)$$

и добавляется к формуле $\mathrm{U}_{\mathrm{k}-1}$: $\mathrm{U}_\mathrm{k} = \mathrm{U}_{\mathrm{k}-1} \bigcup \mathrm{u}_\mathrm{k}$.

Алгоритм заканчивает работу, когда достигнут 100% уровень полноты либо когда для всех конъюнктов-кандидатов $F_{\beta,\gamma}\left(v_j \mid U_{k-1}\right)=0$.

3.2.2 Свойства метрик полнота, точность, F-мера

Для исследования свойств алгоритма ПФБА необходимо доказать несколько вспомогательных утверждений о свойствах функций, участвующих в построении алгоритма.

Напомним определения функций полноты, точности и F-меры (раздел 2.2). Пусть C — множество документов, принадлежащих рубрике и u — множество документов, автоматически приписанных рубрике. Тогда

- полнота $r(u)=\frac{|u\cap C|}{|C|}$,
- точность $p(u)=\frac{|u\cap C|}{|u|}$,
- F-мера $F_\beta(u)=\frac{1+\beta}{\frac{\beta}{p(u)}+\frac{1}{r(u)}}$. Параметр $\beta>0$. Если $p(u)=0$ или $r(u)=0$, то $F_\beta(u)=0$.

Алгоритм ПФБА также использует следующие функции. Пусть v и w – некоторые множества документов. Тогда

- $r(u \mid v)=\frac{|(u\setminus v)\cap C|}{|C\setminus v|}$ — *дополняющая полнота*, то есть полнота v на непокрытой u части рубрики
- $p(u \mid v)=\frac{|(u\setminus v)\cap C|}{|C\setminus v|}$ — *дополняющая точность*, то есть точность v на непокрытой u части рубрики

- Взвешивающая функция $F_{\beta,\gamma}$ с параметрами $\beta > 0$ и $\gamma \geq 0$ определяется следующим образом:

$$F_{\beta,\gamma}(u \mid v) = \frac{1+\beta+\gamma}{\frac{\beta}{p(u \mid v)} + \frac{1}{r(u \mid v)} + \frac{\gamma}{r(u)}}$$

Если $p(u \mid v) = 0$, или $r(u \mid v) = 0$, или $r(u) = 0$, то $F_{\beta,\gamma}(u \mid v) = 0$.

Утверждение 1 (свойства полноты):

1. $0 \leq r(u) \leq 1$
2. $0 \leq r(u \mid v) \leq 1$
3. $r(u \mid \varnothing) = r(u)$
4. если $u \subseteq v$, то $r(u \mid v) = 0$
5. $r(u \cup v) = r(u) + (1 - r(u))r(v \mid u)$
6. $r(u \cup v) \geq r(u)$
7. $r(u \cup v) \leq r(u) + r(v)$
8. $r(u \cup v \mid w) \leq r(u \mid w) + r(v \mid w)$
9. если $r(u) = 1$, то $r(u \mid v) = 1$
10. $r(u \cap v) \leq \max(r(u), r(v))$

Доказательство: свойства 1-4 и 10 тривиально следуют из определений полноты и дополняющей полноты. Свойство 6 следует из свойств 5, 1 и 2. Докажем свойство 5:

$$r(u \cup v) = \frac{|(u \cup v) \cap C|}{|C|} = \frac{|(u \cap C) \cup ((v \setminus u) \cap C)|}{|C|} = \frac{|u \cap C|}{|C|} + \frac{|(v \setminus u) \cap C|}{|C|} =$$

$$= r(u) + \frac{|C \setminus u|}{|C|} r(v \mid u) = r(u) + \frac{|C| - |u \cap C|}{|C|} r(v \mid u) = r(u) + (1 - r(u))r(v \mid u)$$

Свойство 5 доказано.

Докажем свойство 7.

$$r(u \cup v) = \frac{|(u \cup v) \cap C|}{|C|} = \frac{|(u \cap C) \cup (v \cap C)|}{|C|} \le \frac{|u \cap C|}{|C|} + \frac{|v \cap C|}{|C|} = r(u) + r(v)$$

Свойство 7 доказано. Аналогично доказывается свойство 8.

Докажем свойство 9. Если $r(u) = 1$ то $C \subseteq u$. Следовательно,

$$r(u \mid v) = \frac{|(u \setminus v) \cap C|}{|C \setminus v|} = \frac{|(u \cap C) \setminus (v \cap C)|}{|C \setminus v|} = \frac{|C \setminus v|}{|C \setminus v|} = 1$$

Свойство 9 доказано.

Утверждение доказано.

<u>Утверждение 2 (свойства точности):</u>

1. $0 \le p(u) \le 1$
2. $0 \le p(u \mid v) \le 1$
3. $p(u \mid \varnothing) = p(u)$
4. $\min(p(u), p(v \mid u)) \le p(u \cup v) \le \max(p(u), p(v \mid u))$

<u>Доказательство:</u> свойства 1-3 тривиально следуют из определения точности и дополняющей точности. Докажем свойство 4.

Обозначим

$$a_1 = |u \cap C|,\ b_1 = |C \setminus u|,\ a_2 = |(v \setminus u) \cap C|,\ b_2 = |C \setminus (v \setminus u)|$$

Можно выразить точность через эти величины следующим образом:

$$p(u) = \frac{a_1}{b_1},\ p(v \mid u) = \frac{a_2}{b_2},\ p(u \cup v) = \frac{a_1 + a_2}{a_1 + a_2 + b_1 + b_2}$$

Так как свойство 4 симметрично относительно замены $\mathrm{p}(\mathrm{u}) \leftrightarrow \mathrm{p}(\mathrm{v}\,|\,\mathrm{u})$, то без ограничения общности можно считать, что $\frac{a_1}{b_1} \geq \frac{a_2}{b_2}$. Тогда

$$\mathrm{p}(\mathrm{u} \cup \mathrm{v}) = \frac{a_1 + a_2}{a_1 + a_2 + b_1 + b_2} = \frac{a_1}{a_1 + b_1} \cdot \frac{a_1 \cdot \frac{a_1 + b_1}{a_1} + a_2 \cdot \frac{a_1 + b_1}{a_1}}{a_1 + a_2 + b_1 + b_2} \leq$$

$$\leq \frac{a_1}{a_1 + b_1} \cdot \frac{a_1 \cdot \frac{a_1 + b_1}{a_1} + a_2 \cdot \frac{a_2 + b_2}{a_2}}{a_1 + a_2 + b_1 + b_2} = \frac{a_1}{a_1 + b_1} \cdot \frac{a_1 + a_2 + b_1 + b_2}{a_1 + a_2 + b_1 + b_2} = \mathrm{p}(\mathrm{u})$$

Так как, по предположению, $\mathrm{p}(\mathrm{u}) \geq \mathrm{p}(\mathrm{v}\,|\,\mathrm{u})$, то из этого следует, что $\mathrm{p}(\mathrm{u} \cup \mathrm{v}) \leq \max\left(\mathrm{p}(\mathrm{u}), \mathrm{p}(\mathrm{v}\,|\,\mathrm{u})\right)$.

Теперь, также предполагая $\frac{a_1}{b_1} \geq \frac{a_2}{b_2}$, докажем, что $\mathrm{p}(\mathrm{u} \cup \mathrm{v}) \geq \min\left(\mathrm{p}(\mathrm{u}), \mathrm{p}(\mathrm{v}\,|\,\mathrm{u})\right)$

$$\mathrm{p}(\mathrm{u} \cup \mathrm{v}) = \frac{a_1 + a_2}{a_1 + a_2 + b_1 + b_2} = \frac{a_2}{a_2 + b_2} \cdot \frac{a_1 \cdot \frac{a_2 + b_2}{a_2} + a_2 \cdot \frac{a_2 + b_2}{a_2}}{a_1 + a_2 + b_1 + b_2} \geq$$

$$\geq \frac{a_2}{a_2 + b_2} \cdot \frac{a_1 \cdot \frac{a_1 + b_1}{a_1} + a_2 \cdot \frac{a_2 + b_2}{a_2}}{a_1 + a_2 + b_1 + b_2} = \frac{a_2}{a_2 + b_2} \cdot \frac{a_1 + a_2 + b_1 + b_2}{a_1 + a_2 + b_1 + b_2} = \mathrm{p}(\mathrm{v}\,|\,\mathrm{u})$$

Утверждение доказано.

<u>Утверждение 3 (свойства F-меры):</u>

1. $0 \leq \mathrm{F}_\beta(\mathrm{u}) \leq 1$
2. $\min\left(\mathrm{p}(\mathrm{u}), \mathrm{r}(\mathrm{u})\right) \leq \mathrm{F}_\beta(\mathrm{u}) \leq \max\left(\mathrm{p}(\mathrm{u}), \mathrm{r}(\mathrm{u})\right)$
3. если $\mathrm{p}(\mathrm{u}) = \mathrm{r}(\mathrm{u})$, то $\mathrm{F}_\beta(\mathrm{u}) = \mathrm{p}(\mathrm{u}) = \mathrm{r}(\mathrm{u})$ (следствие свойства 2)
4. $\mathrm{F}_\beta(\mathrm{u}) \leq \frac{\beta + 1}{4}\left(\frac{\mathrm{p}(\mathrm{u})}{\beta} + \mathrm{r}(\mathrm{u})\right)$

5. $F_1(u) \le \frac{p(u)+r(u)}{2}$ (следствие свойства 5)

Доказательство: свойства 1 и 3 следуют свойства 2. Свойство 5 следует из свойства 4. Докажем свойство 2.

В случае если $p(u)=0$ либо $r(u)=0$ утверждение тривиально. Иначе, рассмотрим два случая: $0 < p(u) \le r(u)$ и $p(u) > r(u) > 0$

1. Пусть $p(u) \le r(u)$. Тогда

$$F_\beta(u) = \frac{1+\beta}{\frac{\beta}{p(u)}+\frac{1}{r(u)}} = p(u)\frac{1+\beta}{\beta+\frac{p(u)}{r(u)}} \le p(u)\frac{1+\beta}{\beta+1} = p(u)$$

$$F_\beta(u) = \frac{1+\beta}{\frac{\beta}{p(u)}+\frac{1}{r(u)}} = r(u)\frac{1+\beta}{\beta\frac{r(u)}{p(u)}+1} \ge r(u)\frac{1+\beta}{\beta\cdot 1+1} = r(u)$$

2. Пусть $p(u) > r(u)$. Тогда

$$F_\beta(u) = \frac{1+\beta}{\frac{\beta}{p(u)}+\frac{1}{r(u)}} = r(u)\frac{1+\beta}{\beta\frac{r(u)}{p(u)}+1} < r(u)\frac{1+\beta}{\beta\cdot 1+1} = r(u)$$

$$F_\beta(u) = \frac{1+\beta}{\frac{\beta}{p(u)}+\frac{1}{r(u)}} = p(u)\frac{1+\beta}{\beta+\frac{p(u)}{r(u)}} > p(u)\frac{1+\beta}{\beta+1} = p(u)$$

Свойство 2 доказано.

Докажем теперь свойство 4. Для этого докажем, что разность левой и правой части неравенства свойства 4 неположительна.

$$\frac{1+\beta}{\frac{\beta}{\mathrm{p}(\mathrm{u})}+\frac{1}{\mathrm{r}(\mathrm{u})}}-\frac{\beta+1}{4}\left(\frac{\mathrm{p}(\mathrm{u})}{\beta}+\mathrm{r}(\mathrm{u})\right)=\frac{\beta+1}{4}\frac{4-\left(\frac{\mathrm{p}(\mathrm{u})}{\beta}+\mathrm{r}(\mathrm{u})\right)\left(\frac{\beta}{\mathrm{p}(\mathrm{u})}+\frac{1}{\mathrm{r}(\mathrm{u})}\right)}{\frac{\beta}{\mathrm{p}(\mathrm{u})}+\frac{1}{\mathrm{r}(\mathrm{u})}}=$$

$$=\frac{\beta+1}{4}\frac{2-\frac{\mathrm{p}(\mathrm{u})}{\beta\cdot\mathrm{r}(\mathrm{u})}-\frac{\beta\cdot\mathrm{r}(\mathrm{u})}{\mathrm{p}(\mathrm{u})}}{\frac{\beta}{\mathrm{p}(\mathrm{u})}+\frac{1}{\mathrm{r}(\mathrm{u})}}=-\frac{\beta+1}{4}\frac{\left(\sqrt{\frac{\mathrm{p}(\mathrm{u})}{\beta\cdot\mathrm{r}(\mathrm{u})}}-\sqrt{\frac{\beta\cdot\mathrm{r}(\mathrm{u})}{\mathrm{p}(\mathrm{u})}}\right)^2}{\frac{\beta}{\mathrm{p}(\mathrm{u})}+\frac{1}{\mathrm{r}(\mathrm{u})}}\le 0$$

Свойство 4 доказано.

Утверждение 3 доказано.

3.2.3 Исследование сходимости алгоритма ПФБА для «идеальной» рубрики

Предположим, что существует точная формула U^* длины n, описывающая рубрику с полнотой и точностью, равной единице. Оценим метрики качества *i*-го конъюнкта в формуле, которую построит алгоритм ПФБА с параметрами $\beta>0$ и $\gamma\ge 0$. Эти оценки сформулированы в виде следующей теоремы:

Теорема 1: *Пусть существует формула* $\mathrm{U}^*=\mathrm{u}_1^*\cup\mathrm{u}_2^*\cup\ldots\cup\mathrm{u}_n^*,\ \mathrm{u}_\mathrm{j}^*\in\mathrm{V}$ *длины n, описывающая рубрику* C *с полнотой и точностью, равной единице. Обозначим* $\rho=\min\limits_{j=1..n}\left(\mathrm{r}\left(\mathrm{u}_j^*\right)\right),\ \rho>0.$

Пусть на шаге $i\ge 1$ *алгоритмом ПФБА с параметрами* $\beta>0$ *и* $\gamma\ge 0$ *построена формула* $\mathrm{U}_\mathrm{i}=\mathrm{u}_1\cup\mathrm{u}_2\cup\ldots\cup\mathrm{u}_i$. *Тогда выполняются следующие неравенства:*

1. $\mathrm{p}\left(\mathrm{u}_\mathrm{i}\,|\,\mathrm{U}_{i-1}\right)\ge\dfrac{\beta}{\beta+(n-1)+\gamma\left(\dfrac{1}{\rho}-1\right)}$

2. $\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \geq \dfrac{1}{n+\gamma\left(\dfrac{1}{\rho}-1\right)}$

3. $\mathrm{r}(\mathrm{u}_i) \geq \rho \dfrac{1}{1+\dfrac{n-1}{\gamma \cdot \rho}}$, если $\gamma > 0$

Для доказательства Теоремы 1 докажем следующую лемму:

Лемма *1*: *На каждом шаге* $i \geq 1$ *алгоритма ПФБА, если* $\mathrm{r}(\mathrm{U}_{i-1}) < 1$, *то для некоторого* j $\mathrm{r}(\mathrm{u}_j^* \mid \mathrm{U}_{i-1}) \geq \dfrac{1}{n}$.

Доказательство*:* Из условия теоремы 1 $\mathrm{r}(\mathrm{U}^*) = 1$. Из свойств полноты (Утверждение 1 раздела 3.2.2) следует, что

$$\mathrm{r}(\mathrm{U}^* \mid \mathrm{U}_{i-1}) = 1 \qquad (3.10)$$

Из свойств полноты также следует, что

$$\mathrm{r}\left(\bigcup_{j=1}^{n} \mathrm{u}_j^* \,\middle|\, \mathrm{U}_{i-1}\right) \leq \sum_{j=1}^{n} \mathrm{r}(\mathrm{u}_j^* \mid \mathrm{U}_{i-1}) \qquad (3.11)$$

Из (3.10) и (3.11) следует, что

$$\frac{1}{n}\sum_{j=1}^{n} \mathrm{r}(\mathrm{u}_j^* \mid \mathrm{U}_{i-1}) \geq \frac{1}{n}$$

Следовательно, для некоторого j $\mathrm{r}(\mathrm{u}_j^* \mid \mathrm{U}_{i-1}) \geq \dfrac{1}{n}$, что и требовалось доказать.

Доказательство Теоремы 1*:* Пусть на i-м шаге имеется построенная формула U_{i-1} и $\mathrm{u}_i = \arg\max_{\mathrm{u} \in \mathrm{V}} \left(\mathrm{F}_{\beta,\gamma}(\mathrm{u} \mid \mathrm{U}_{i-1})\right)$.

Из Леммы 1 следует, что для некоторого конъюнкта $u_j^* \in V$ дополняющая полнота $r(u_j^* \mid U_{i-1}) \geq \frac{1}{n}$. Так как u_j^* — часть точной формулы U^*, то $p(u_j^* \mid U_{i-1}) = 1$ и $r(u_j^*) \geq \rho$. Следовательно,

$$F_{\beta,\gamma}(u_j^* \mid U_{i-1}) \geq \frac{1+\beta+\gamma}{\frac{\beta}{1} + \frac{1}{\left(\frac{1}{n}\right)} + \frac{\gamma}{\rho}}$$

Так как $F_{\beta,\gamma}(u_i \mid U_{i-1}) \geq F_{\beta,\gamma}(u_j^* \mid U_{i-1})$, то, следовательно,

$$\frac{1+\beta+\gamma}{\frac{\beta}{p(u_i \mid U_{i-1})} + \frac{1}{r(u_i \mid U_{i-1})} + \frac{\gamma}{r(u_i)}} \geq \frac{1+\beta+\gamma}{\beta+n+\frac{\gamma}{\rho}}$$

Учитывая, что числители и знаменатели правой и левой части положительны, получим

$$\frac{\beta}{p(u_i \mid U_{i-1})} + \frac{1}{r(u_i \mid U_{i-1})} + \frac{\gamma}{r(u_i)} \leq \beta + n + \frac{\gamma}{\rho} \qquad (1.12)$$

1. Для доказательства нижней оценки для дополняющей точности выделим $p(u_i \mid U_{i-1})$ из неравенства (1.12):

$$p(u_i \mid U_{i-1}) \geq \frac{\beta}{\beta+n+\frac{\gamma}{\rho} - \frac{1}{r(u_i \mid U_{i-1})} - \frac{\gamma}{r(u_i)}}$$

и воспользуемся неравенствами $r(u_i \mid U_{i-1}) \leq 1$ и $r(u_i) \leq 1$. Получим:

$$p(u_i \mid U_{i-1}) \geq \frac{\beta}{\beta + (n-1) + \gamma\left(\frac{1}{\rho} - 1\right)}$$

Первая часть Теоремы 1 доказана.

2. Для доказательства нижней оценки для дополняющей полноты выделим $\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1})$ из неравенства (1.12):

$$\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \geq \frac{1}{\beta + n + \frac{\gamma}{\rho} - \frac{\beta}{\mathrm{p}(\mathrm{u}_i \mid \mathrm{U}_{i-1})} - \frac{\gamma}{\mathrm{r}(\mathrm{u}_i)}}$$

и воспользуемся неравенствами $\mathrm{p}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \leq 1$ и $\mathrm{r}(\mathrm{u}_i) \leq 1$. Получим:

$$\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \geq \frac{1}{n + \gamma\left(\frac{1}{\rho} - 1\right)}$$

Вторая часть Теоремы 1 доказана.

3. Для доказательства нижней оценки для полноты выделим $\mathrm{r}(\mathrm{u}_i)$ из неравенства (1.12) (это можно сделать при условии $\mathrm{wr} > 0$ в условии теоремы):

$$\mathrm{r}(\mathrm{u}_i) \geq \frac{\gamma}{\beta + n + \frac{\gamma}{\rho} - \frac{\beta}{\mathrm{p}(\mathrm{u}_i \mid \mathrm{U}_{i-1})} - \frac{1}{\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_i)}}$$

и воспользуемся неравенствами $\mathrm{p}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \leq 1$ и $\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \leq 1$. Получим:

$$\mathrm{r}(\mathrm{u}_i) \geq \frac{\gamma}{(n-1) + \frac{\gamma}{\rho}}$$

Или, преобразовав, получим:

$$\mathrm{r}(\mathrm{u}_i) \geq \rho \frac{1}{1 + \frac{n-1}{\gamma \cdot \rho}}$$

Третья часть Теоремы 1 доказана.

Теорема 1 доказана.

Воспользуемся теоремой 1 для того, чтобы оценить оптимальный выбор параметров алгоритма ПФБА для получения формулы заданного качества. Пусть алгоритм ПФБА строит по шагам формулу $\mathrm{U} = \bigcup_{i=1}^{N} \mathrm{u}_i$

Следствие 1*: Пусть выполняются условия теоремы 1 и задан параметр* $\eta \in (0,1]$. *Тогда, если* $\gamma \geq \frac{1-\eta}{\eta}\frac{n-1}{\rho}$, *то для любого* $i = 1..N$ $\mathrm{r}(\mathrm{u}_i) \geq \rho(1-\eta)$.

Доказательство: Если $\eta = 1$, то утверждение теоремы тривиально. Иначе, пусть

$$\gamma \geq \frac{1-\eta}{\eta}\frac{n-1}{\rho} \qquad (1.13)$$

Так как правая часть последнего неравенства положительна, то из теоремы 1 пункта 3 следует, что

$$\mathrm{r}(\mathrm{u}_i) \geq \rho \frac{1}{1+\frac{n-1}{\gamma \cdot \rho}} \qquad (1.14)$$

Подставляя (1.13) в (1.14), получаем

$$\mathrm{r}(\mathrm{u}_i) \geq \rho(1-\eta)$$

Что и требовалось доказать.

Следствие 2*: Пусть выполняются условия теоремы 1 и задан параметр* $\eta \in (0,1)$. *Тогда, если*

$$\beta \geq \frac{1-\eta}{\eta}\left((n-1)+\gamma\left(\frac{1}{\rho}-1\right)\right) \qquad (1.15)$$

то для любого $i = 1..N$

1. $\mathrm{p}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \geq 1-\eta$
2. $\mathrm{p}(\mathrm{U}_i) \geq 1-\eta$.

Доказательство: Пусть выполняется условие (1.15)

Из теоремы 1 пункта 1 следует, что

$$\mathrm{p}\left(\mathrm{u_i} \mid \mathrm{U}_{i-1}\right) \geq \frac{\beta}{\beta + (n-1) + \gamma\left(\frac{1}{\rho} - 1\right)} \qquad (1.16)$$

Так как функция $f(\beta) = \frac{\beta}{\beta + x}$ возрастает при фиксированном параметре $x > 0$ и $\beta > 0$, то, подставляя (1.15) в (1.16), получаем искомое неравенство:

$$\mathrm{p}\left(\mathrm{u}_i \mid \mathrm{U}_{i-1}\right) \geq 1 - \eta \qquad (1.17)$$

Теперь докажем по индукции, что $\mathrm{p}\left(\mathrm{U}_i\right) \geq 1 - \eta$.

1. Так как $\mathrm{p}\left(\mathrm{U}_1\right) = \mathrm{p}\left(\mathrm{u}_1\right) = \mathrm{p}\left(\mathrm{u}_1 \mid \varnothing\right) = \mathrm{p}\left(\mathrm{u}_1 \mid \mathrm{U}_0\right)$, то $\mathrm{p}\left(\mathrm{U}_1\right) \geq 1 - \eta$
2. Из свойства точности (Утверждение 2 раздела 3.2.2) следует, что $\mathrm{p}\left(\mathrm{U}_{i-1} \cup \mathrm{u}_i\right) \geq \min\left(\mathrm{p}\left(\mathrm{U}_{i-1}\right), \mathrm{p}\left(\mathrm{u}_i \mid \mathrm{U}_{i-1}\right)\right)$. По предположению индукции и из (1.17) следует, что $\mathrm{p}\left(\mathrm{U}_{i-1} \cup \mathrm{u}_i\right) \geq 1 - \eta$. По определению $\mathrm{p}\left(\mathrm{U}_i\right) = \mathrm{p}\left(\mathrm{U}_{i-1} \cup \mathrm{u}_i\right)$.

Следовательно, для любого $i \geq 1$

$$\mathrm{p}\left(\mathrm{U}_i\right) \geq 1 - \eta$$

Следствие 2 доказано.

***Следствие 3**: Пусть выполняются условия теоремы 1 и задан параметр $\eta \in (0,1)$. Тогда для любого $i = 1..N$*

1. $\mathrm{r}(\mathrm{U}_i) \geq 1 - \left(1 - \dfrac{1}{n + \gamma\left(\dfrac{1}{\rho} - 1\right)}\right)^i$
2. если $i \geq \dfrac{\ln \eta}{\ln\left(1 - \dfrac{1}{n + \gamma\left(\dfrac{1}{\rho} - 1\right)}\right)}$, то $\mathrm{r}(\mathrm{U}_i) \geq 1 - \eta$

Доказательство: Обозначим

$$\zeta = \frac{1}{n + \gamma\left(\dfrac{1}{\rho} - 1\right)} \tag{1.18}$$

Из теоремы 1 пункта 2 следует, что $\mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1}) \geq \zeta$. Из определения дополняющей полноты следует, что $\mathrm{r}(\mathrm{U}_i) = \mathrm{r}(\mathrm{U}_{i-1}) + (1 - \mathrm{r}(\mathrm{U}_{i-1})) \cdot \mathrm{r}(\mathrm{u}_i \mid \mathrm{U}_{i-1})$.

Запишем рекуррентные соотношения:

$$\begin{cases} \mathrm{r}(\mathrm{U}_0) = 0 \\ \mathrm{r}(\mathrm{U}_i) \geq \mathrm{r}(\mathrm{U}_{i-1}) + (1 - \mathrm{r}(\mathrm{U}_{i-1})) \cdot \zeta, \; i \geq 1 \end{cases}$$

Так как $0 \leq \mathrm{r}(\mathrm{U}_i) \leq 1$, то эта система эквивалента

$$\begin{cases} \mathrm{r}(\mathrm{U}_0) = 0 \\ (1 - \mathrm{r}(\mathrm{U}_i)) \geq (1 - \mathrm{r}(\mathrm{U}_{i-1}))(1 - \zeta), \; i \geq 1 \end{cases}$$

Решая рекуррентное соотношение, получаем

$$(1 - \mathrm{r}(\mathrm{U}_i)) \leq (1 - \zeta)^i \tag{1.19}$$

Преобразовывая, получаем:

$$\mathrm{r}\left(\mathrm{U}_i\right) \geq 1-\left(1-\zeta\right)^i$$

Первое утверждение следствия доказано.

Теперь докажем второе утверждение следствия. Выражение (1.19) эквивалентно

$$\ln\left(1-\mathrm{r}\left(\mathrm{U}_i\right)\right) \leq i \cdot \ln\left(1-\zeta\right)$$

Пусть $i \geq \dfrac{\ln \eta}{\ln\left(1-\zeta\right)}$. Подставляя выражение для i в правую часть, получим

$$\ln\left(1-\mathrm{r}\left(\mathrm{U}_i\right)\right) \leq \ln \eta$$

Это эквивалентно

$$\mathrm{r}\left(\mathrm{U}_i\right) \geq 1-\eta$$

Вторая часть утверждения доказана.

Следствие 3 доказано.

Следствия 1 и 2 позволяют утверждать, что, если существует формула, которая точно описывает заданное множество документов (рубрику), то алгоритм ПФБА, при достаточно «жестких» параметрах (т.е. больших значениях β и γ) построит формулу, описывающую рубрику с заданным уровнем качества. А именно: для любого, сколь угодно малого параметра $\eta > 0$ можно выбрать параметры алгоритма так, что

1. точность полученной формулы будет не менее $1-\eta$
2. каждый конъюнкт будет иметь покрытие (полноту) почти как у точной формулы: $\mathrm{r}\left(\mathrm{u}_i\right) \geq \rho\left(1-\eta\right)$

Следствие 3 позволяет вычислить длину формулы, которая описывает рубрику с заданным уровнем полноты.

Есть несколько моментов, связанных с доказанными утверждениями об алгоритме, которые необходимо отметить:

1. Алгоритм ПФБА может не найти точной формулы, а только формулу, полнота и точность которой близка к 100%.
2. Согласно следствию 3, ограничение на длину полученной формулы не зависит от параметра β — веса важности точности относительно полноты. Алгоритм всегда может получить достаточно короткую формулу, вне зависимости от β, если короткая точная формула существует.
3. Согласно следствию 3, длина найденной формулы может быть существенно больше длины оптимальной формулы.

Эти свойства алгоритма являются побочным эффектом, связанным с тем, что основная задача алгоритма и цель его разработки — построение «хорошей» формулы для *реальных* задач. Обычно для реальных задач точной формулы, описывающей рубрику, не существует, либо точная формула имеет неприемлемо большую длину. Предположения, при которых верна теорема 1 — в некотором смысле предельный случай реальной задачи. Аналитически исследовать реальную задачу не представляется возможным. Поэтому исследование алгоритма на реальных задачах проводилось экспериментально.

Для случая, когда точной формулы не существует (или она очень длинная), алгоритм ищет приближенную формулу описания рубрики. При этом параметр β влияет на соотношение полноты и точности конъюнктов, добавляемых на шаге алгоритма, а, следовательно, и на соотношение полноты и точности при фиксированной длине формулы. Большие значения β приводят к более точной и более длинной формуле.

Если бы задача состояла в том, чтобы строить только точную формулу (в предположении существования таковой), то можно было бы обойтись более простым алгоритмом. А именно: на каждом шаге выбирать точный конъюнкт с максимальной дополняющей полнотой

$$\mathrm{u}_i = \arg \max_{j:\mathrm{p}(\mathrm{u}_j)=1} \left(\mathrm{r}\left(\mathrm{u}_j \mid \mathrm{U}_{i-1} \right) \right)$$

Отметим, что такой алгоритм является предельным случаем алгоритма ПФБА при $\beta \to \infty$.

Покажем на примерах конкретные значения параметров алгоритма ПФБА, которые обеспечивают получение формулы с заданной точностью. Пусть существует точная формула U^* длины n и требуется построить формулу, описывающую рубрику с полнотой и точностью не ниже $1-\eta$. Параметр γ можно установить равным нулю (не требуется высокая полнота каждого отдельного конъюнкта). В таблице 1 для различных значений n и $1-\eta$ приведены значения параметра β алгоритма ПФБА и количество шагов N (длина построенной формулы), которые обеспечивают заданные полноту и точность. Строки таблицы соответствуют различным значениям длины точной формулы n, а столбцы — уровням полноты/точности $1-\eta$.

n	уровень полноты/точности $1-\eta$			
	70%	80%	90%	95%
2	$\beta = 2.4, N = 2$	$\beta = 4, N = 3$	$\beta = 9, N = 4$	$\beta = 19, N = 5$
5	$\beta = 9.4, N = 6$	$\beta = 16, N = 8$	$\beta = 36, N = 11$	$\beta = 76, N = 14$
10	$\beta = 21, N = 12$	$\beta = 36, N = 16$	$\beta = 81, N = 22$	$\beta = 171, N = 29$
20	$\beta = 45, N = 24$	$\beta = 76, N = 32$	$\beta = 171, N = 45$	$\beta = 361, N = 59$

Таблица 1. Значения параметра β алгоритма ПФБА и количество шагов N (длина построенной формулы) в зависимости от длины точной формулы n и заданного уровня полноты/точности $1-\eta$.

3.3 Экспериментальное исследование алгоритма построения формул ПФА

В этом разделе мы исследуем алгоритм построения формул на реальных задачах классификации текстов. Исследование основано на общедоступных коллекциях текстов, размеченных экспертами по заданному рубрикатору. Результаты работы алгоритма ПФА сравнивались с результатами работы других методов классификации текстов. В результате сравнения можно утверждать, что алгоритм ПФА показывает качество классификации, сравнимое с результатами других алгоритмов. Кроме того, мы продемонстрируем формулы, полученные алгоритмом, и покажем, что построенные формулы действительно отражают смысл рубрики (а это как раз и являлось основной целью создания алгоритма).

Для одной из рубрик коллекции Reuters-21578 мы исследуем влияние различных параметров и модификаций алгоритма на вид получаемых формул и качество обучения. Мы покажем, что, в зависимости от требований к

получаемому результату, алгоритм может создать описание рубрики различной степени подробности.

Исследование проводилось на следующих коллекциях документов:

1. Reuters-21578 — коллекция финансовых новостей агентства Reuters (на английском языке). 21578 документов, 135 рубрик.
2. РОМИП-2004 Legal — коллекция нормативных актов РФ из БД "Кодекс" (на русском языке). 60015 документов, 170 рубрик.

Подробное описание данных коллекций и экспериментов на них приводится в разделах 3.3.2 и 3.3.3 соответственно.

3.3.1 Описание программной реализации алгоритма

Опишем вкратце технологии, на основе которых реализована программа построения формул на основе машинного обучения. Алгоритм реализован в виде программы на языке Java с использованием реляционной СУБД Oracle 9i и средств информационной системы УИС РОССИЯ [13, 12, 16].

Для проведения эксперимента коллекция документов загружается в УИС РОССИЯ стандартными средствами информационной системы. При этом в СУБД Oracle создаётся схема данных, содержащая информацию о каждом документе. Каждому документу присваивается уникальный идентификатор, и загружаются словарный и тематический индексы документов. Приведение слов к нормальной форме и выделение понятий осуществляется программой Автоматической Лингвистической Обработки Текстов [29, 30]. Словарный и тематический индекс, а также формальные атрибуты документов (в том числе приписанные рубрики) загружаются при помощи АРМ загрузки данных. Схема данных, используемая программой построения формул, представлена на рис. 2.

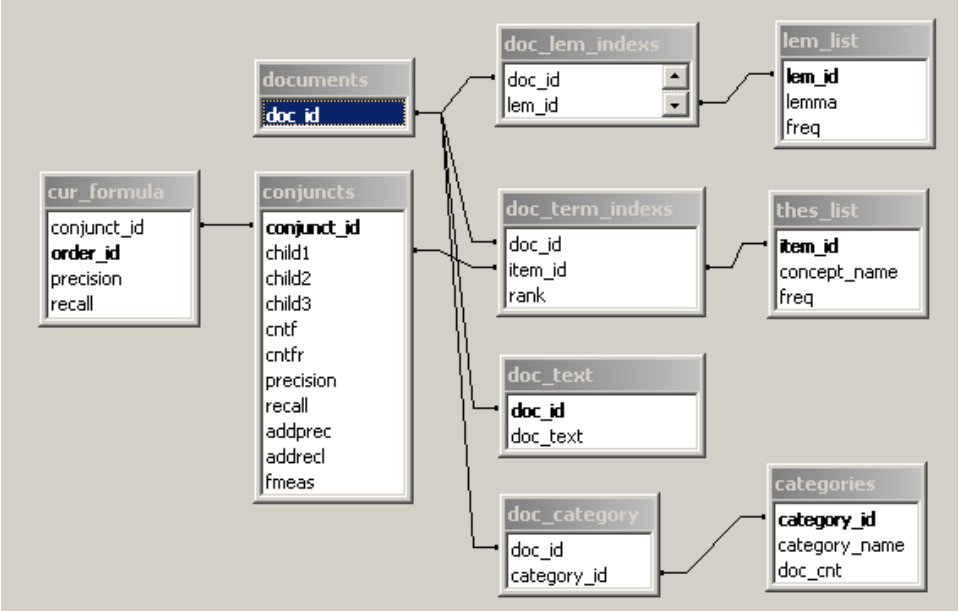

Рис. 2. Схема данных в СУБД Oracle9i, используемая программой построения формул.

Программа, реализующая алгоритм построения формул получает на вход

1. параметры подключения к СУБД Oracle;
2. номер рубрики, для которой нужно построить формулу;
3. параметры алгоритма построения формул.

Основные операции по обработке данных производятся средствами SQL в СУБД Oracle. Для построения формул используются таблицы `conjuncts` и `cur_formula` (см. рис. 2). Таблица `conjuncts` хранит список вычисленных конъюнктов и метрики качества конъюнктов. В таблице `cur_formula` хранятся идентификаторы конъюнктов, составляющих формулу, в порядке их добавления в формулу. В процессе построения формулы значения метрик для этих таблиц обновляются при помощи операторов SQL и PL/SQL-процедур.

Для проводимых экспериментов время работы алгоритма составляло 3-10 минут на одну рубрику.

3.3.2 Эксперименты на коллекции Reuters-21578

Коллекция Reuters-21578 создана новостным агентством Рейтерс и предоставлена в свободный доступ для проведения исследований в области автоматической обработки текстов [70]. Коллекция состоит из 21578 документов — финансовых новостей агентства Рейтерс, выпущенных в 1987 году. Все документы отрубрицированы экспертами агентства Рейтерс и компании Carnegie Group по рубрикатору, состоящему из 135 рубрик.

Коллекция была адаптирована для тестирования методов машинного обучения Д. Льюисом [70] — документы переведены в удобный для автоматической обработки формат, а классификация документов — уточнена. Некоторым документам был присвоен статус "классификация не точна". Кроме того, множество документов было разбито на коллекцию, рекомендуемую для обучения (9603 документа, 74%) и коллекцию, рекомендуемую для тестирования метода машинного обучения (3299 документов, 26%), всего — 12902 документа. Такое разбиение называется ModApte split. Именно это разбиение рекомендуется использовать исследователям для тестирования своих методов и публикации результатов. Использование такого разбиения способствует воспроизводимости публикуемых результатов.

Коллекция получила очень широкое распространение в научном сообществе. Для многих методов машинного обучения опубликованы результаты тестирования на коллекции Reuters-21578 ModApte split. Коллекция доступна для скачивания по адресу http://www.daviddlewis.com/resources/testcollections/reuters21578/.

Мы провели тестирование нашего метода построения формул на коллекции Reuters-21578 ModApte split [1]. Использовались следующие параметры алгоритма построения формул:

1. Шаг 1. В качестве векторного представления использовалось словарное представление. Все слова, встречающиеся в документах, были приведены к нормальной форме. Слова, встречающиеся менее чем в 5 документах, были усечены.
2. Шаг 2. Для набора конъюнктов перебирались все конъюнкции, состоящие из 100 слов с наибольшей F-мерой. Из списка полученных конъюнктов откидывались те, полнота которых ниже порога 20% или точность ниже порога 3%.
3. Шаг 3. Для основного теста использовались параметры формул (3.6) и (3.7): $w_{p1} = 5, w_{ap} = 10, w_{ar} = 5, w_r = 2$. Условия остановки: количество дизъюнктов больше 20; prec<10% и recl>90%; recl>99%. Кроме того, для исследования поведения алгоритма при вариации параметров, использовались различные значения параметров w_{p1}, w_{ap}, w_{ar} и w_r.

В таблице 2 приводятся сравнение результатов нашего метода с SVM (Support Vector Machines) [63, 78]. Результаты приведены для рубрик с количеством документов более 100 (столбец doc_cnt). В столбцах "Joachims P/R b.p." и "Dumais et.al. P/R b.p." приводятся результаты, опубликованные в [63] и [58] соответственно.

К сожалению, в указанных работах опубликованы результаты только для наиболее частотных 10 рубрик [57]. В столбце "Our SVM" мы приводим результаты нашей реализации тестов SVM (см. раздел 2.5.7.2, [6, 11]). В столбце "disj formulae" приводятся результаты работы нашего алгоритма построения формул. Использовались формулы вида (3.2).

NAME	DOC_CNT	Joachims P/R b.p.	Dumais et.al. P/R b.p.	Our SVM	disj formulae
earn	3964	98,20	98,00	97,79	90,70
acq	2369	92,60	93,60	95,69	82,01
...	2108			83,72	56,06
money-fx	717	66,90	74,50	72,83	58,54
grain	582	91,30	94,60	89,00	88,89
crude	578	86,00	88,90	82,82	69,31
trade	486	69,20	75,90	77,45	64,52
interest	478	69,80	77,70	75,57	56,59
ship	286	82,00	85,60	74,55	69,60
wheat	283	83,10	91,80	89,59	89,74
corn	237	86,00	90,30	86,31	90,32
dlr	175			69,81	51,79
money-sup	174			74,01	48,54
oilseed	171			65,96	78,57
sugar	162			88,54	85,37
coffee	139			92,72	91,80
gnp	136			83,57	75,56
veg-oil	124			77,56	70,97
gold	124			64,48	61,54
soybean	111			61,56	74,70
nat-gas	105			61,03	44,44
bop	105			69,13	53,52

Таблица 2. Результаты сравнения метода построения формул с SVM на рубриках Reuters-21578.

Результаты позволяют утверждать, что алгоритм построения формул показывает результаты классификации, сравнимые с SVM, хотя в среднем – несколько хуже.

3.3.2.1 Анализ полученных формул

Для многих рубрик получились короткие и понятные формулы, соответствующие названию рубрики.

Например:

Рубрика	Формула	Полнота	Точность
Coffee	/Лемма="COFFEE"	100	84,85
Soybean	/Лемма="SOYBEAN"	93,94	62
Wheat	/Лемма="WHEAT"	98,59	82,35
Corn	/Лемма="CORN"	100	82,35
Alum	(/Лемма="ALUMINIUM"	97,14	59,65

	AND /Лемма="TONNE") OR /Лемма="ALUMINIUM" OR /Лемма="ALUMINUM" OR (/Лемма="ALUMINA" AND /Лемма="TONNE") OR /Лемма="ALCOA"		

Для некоторых рубрик формулы получились весьма длинными. Например:

Рубрика	Формула	Полнота	Точность
Interest	/Лемма="02-333" OR (/Лемма="BANK" AND /Лемма="CUT" AND /Лемма="RATE") OR (/Лемма="PCT" AND /Лемма="MARKET" AND /Лемма="MONEY") OR (/Лемма="REPURCHASE" AND /Лемма="CUSTOMER") OR (/Лемма="DISCOUNT" AND /Лемма="RATE")	61,38	50,84

Результаты показывают, что в целом имеет место следующая зависимость: чем длиннее формула, тем хуже результаты. Это явление можно объяснить тем, что в случае, когда для рубрики существует простое вербальное описание, то, скорее всего, будет существовать и короткая формула, описывающая рубрику. Рубрицирование при помощи такой формулы будет давать результаты, близкие к ручному рубрицированию.

В то же время рубрики, которые сложно описать одним-двумя словами имеют менее четкие, менее формальные границы определения и описать принадлежность рубрике в виде булевской формулы сложно.

3.3.2.2 Анализ влияния различных параметров

На примере рубрики "gold" рассмотрим, как некоторые параметры алгоритма влияют на качество описания рубрики. Мы применяли алгоритм построения формулы на рубрике "gold" с различными параметрами и вычисляли метрики качества рубрицирования — полноту и точность — на коллекции обучения и на коллекции тестирования.

В таблицах результатов для каждого запуска алгоритма мы приводим полученную формулу и вычисленную полноту и точность на разбиении.

Варьируя параметр "вес дополняющей полноты" в формуле (3.7) можно получать формулы различной длины:

Короткую формулу можно получить, задав следующие параметры формулы (3.7): ($w_{ap} = 10, w_{ar} = 15, w_r = 2$). Вес дополняющей полноты w_{ar} большой, поэтому получается короткая формула:

Формула	**Полнота TRAIN**	**Точность TRAIN**	**Полнота TEST**	**Точность TEST**
/Лемма="OUNCE"	67,02	82,89	53,33	72,73

Попытки повышения полноты путем простого добавления конъюнктов ведут обычно к существенному уменьшению точности:

Формула	**Полнота TRAIN**	**Точность TRAIN**	**Полнота TEST**	**Точность TEST**
/Лемма="OUNCE" OR /Лемма="GOLD"	100	50,27	100	51,72

В данном случае лемма "GOLD" дает сильное улучшение полноты (до 100%) предыдущей формулы, но при этом точность резко падает.

Можно добиться хороших результатов «набирая» дизъюнкциями частные случаи. Слово "gold" встречается в конъюнкции с другими словами. Такую формулу можно получить, установив параметр "вес дополняющей полноты" малым ($w_{ap} = 10, w_{ar} = 0.1, w_r = 2$), и установив более жесткие условия на точность первого конъюнкта в формуле (3.6):

Формула

```
/Лемма="MINEWORKER"
OR (/Лемма="OUNCE" AND /Лемма="GOLD")
OR (/Лемма="TON"   AND /Лемма="GOLD")
OR (/Лемма="GOLD"  AND /Лемма="CONTAIN")
OR (/Лемма="GOLD" AND /Лемма="DEPOSIT")
OR (/Лемма="GOLD" AND /Лемма="UNDERGROUND")
OR (/Лемма="GRADE" AND /Лемма="GOLD")
OR (/Лемма="SILVER" AND /Лемма="SHORT")
OR (/Лемма="COIN" AND /Лемма="GOLD")
OR (/Лемма="BULLION" AND /Лемма="GOLD")
```

Полнота TRAIN	**Точность TRAIN**	**Полнота TEST**	**Точность TEST**
97,87	85,98	63,33	76,00

Можно сгруппировать элементы формулы так, чтобы формула имела «трёхуровневый» вид, аналогичный используемому экспертами:

Формула

```
/Лемма="MINEWORKER"
OR (/Лемма="GOLD"
    AND (
      /Лемма="OUNCE"
      OR /Лемма="TON"
      OR /Лемма="CONTAIN"
      OR /Лемма="DEPOSIT"
      OR /Лемма="UNDERGROUND"
      OR /Лемма="GRADE"
      OR /Лемма="COIN"
      OR /Лемма="BULLION"
     )
   )
OR (/Лемма="SILVER" AND /Лемма="SHORT")
```

Полнота TRAIN	**Точность TRAIN**	**Полнота TEST**	**Точность TEST**
97,87	85,98	63,33	76,00

Здесь важно заметить, что для полученной формулы полнота и точность на тестовой коллекции документов сильно ниже полноты и точности на обучающей коллекции. То есть алгоритм слишком сильно "подгоняет" формулу под обучающую выборку.

В качестве альтернативы набору формулы из малочастотных конъюнктов мы испытали возможность построения формул с отрицанием вида (3.3) и (3.4).

Построение формулы с отрицанием вида (3.3) позволяет получить более короткую (по количеству слов) формулу с высокими результатами:

Формула

```
(
/Лемма="GOLD"
 AND NOT /Лемма="NET"
 AND NOT /Лемма="AGREEMENT"
 AND NOT /Лемма="COMMON"
 AND NOT /Лемма="ACCOUNT"
 AND NOT /Лемма="FRANCE"
 AND NOT /Лемма="BLOCK"
 AND NOT /Лемма="UNCHANGED"
 AND NOT /Лемма="BOARD"
 AND NOT /Лемма="DE"
 AND NOT /Лемма="CUT"
)
OR (
        /Лемма="OUNCE"
   AND /Лемма="ORE"
   AND /Лемма="RESOURCE" )
```

Полнота TRAIN	**Точность TRAIN**	**Полнота TEST**	**Точность TEST**
100,00	87,04	76,67	69,70

При этом имеет место большое различие результатов на коллекции обучения и коллекции тестирования.

Модификация алгоритма (3.4) дает формулу

Формула

```
(    /Лемма="OUNCE"
  OR /Лемма="GOLD"
)
AND NOT (
  ( /Лемма="CURRENCY" AND /Лемма="RESERVES")
  OR ( /Лемма="CONVERT" )
  OR ( /Лемма="ROSE" AND /Лемма="RESERVES" )
  OR ( /Лемма="STRENGTH" )
  OR ( /Лемма="RESULTED" )
  OR ( /Лемма="SPECIAL"
       AND /Лемма="RESERVE"
     )
  OR ( /Лемма="95" AND /Лемма="ROSE" )
  OR ( /Лемма="REPAYMENT" )
  OR ( /Лемма="WEEKLY" )
  OR ( /Лемма="END-FEBRUARY"
       AND /Лемма="RESERVE"
     )
)
```

Полнота TRAIN	Точность TRAIN	Полнота TEST	Точность TEST
100,00	61,84	96,67	60,42

Важно отметить, что в данном случае параметры качества рубрицирования не сильно различаются на коллекции обучения и коллекции тестирования.

Полученная формула отражает известный факт [61] что эксперты, рубрицирующие документы Reuters не относят тематику "золотые резервы" к рубрике gold. Соответственно, некоторые конъюнкты, содержащие слово "reserves" вычитаются из описания рубрики.

3.3.3 Эксперименты на коллекции РОМИП-2004

Российский семинар по Оценке Методов Информационного Поиска (РОМИП, http://romip.narod.ru) — открытая инициатива группы российских

ученых для проведения независимой оценки методов информационного поиска, ориентированных на работу с русскоязычной информацией [51]. Инициатива имеет некоммерческий характер, однако в РОМИП принимают участие исследователи как из научных организаций, так и исследовательские подразделения ведущих российских компаний в области построения информационно-поисковых систем. Методология оценки методов информационного поиска и формат семинара базируется на опыте аналогичных международных конференциях, таких как TREC (Text REtrieval Conference, http://trec.nist.gov), SUMMAC (TIPSTER Text Summarization Evaluation Conference, http://www.itl.nist.gov/iaui/894.02/related_projects/tipster_summac/) и CLEF (Cross-Language Evaluation Forum, http://clef.iei.pi.cnr.it:2002/).

Первый семинар РОМИП проводился в 2003 году. Мы приняли участие в дорожке поиска по web-коллекции РОМИП'2003. Исследование результатов показало [14], что классические методы поиска TF*IDF показывают весьма высокие (один из лучших) результаты.

В 2004 году набор дорожек РОМИП был расширен. Мы приняли участие в трёх дорожках: дорожке поиска по web-коллекции, поиска по коллекции нормативных документов и дорожке тематической классификации нормативных документов.

В этом разделе мы опишем результаты, полученные на дорожке классификации текстов.

Задание состояло в построении процедуры автоматической классификации текстов для коллекции нормативных документов законодательства Российской Федерации из БД СПС «Кодекс». Рубрикатор состоит из 183 рубрик, являющихся подмножеством большого иерархического рубрикатора нормативных документов. Для обучения процедуры классификации предлагается коллекция из 4496 документов,

отрубрицированных по данному классификатору экспертами компании «Кодекс». Для тестирования предоставлены 55519 документов, для которых необходимо автоматически определить рубрики, к которым эти документы относятся. Для некоторых рубрик нет документов в коллекции обучения, всего рубрик с ненулевым количеством документов для обучения — 170.

Всего для дорожки классификации было прислано 9 прогонов от 5 участников [47, 50, 46, 22, 7] (т.е. сравнивалось 9 различных алгоритмов классификации текстов), в том числе наши 3 прогона.

Мы подготовили три прогона. Два прогона основаны на алгоритме машинного обучения SVM (см. раздел 2.5.7). Третий прогон основан на алгоритме построения формул ПФА (раздел 3.1).

Целью первого прогона было получение «отправной точки» для дорожки классификации тестов. Первый прогон основан на широко распространённых технологиях — свободно распространяемой версии SVM, нормализации слов и формуле TF*IDF, с минимальными дополнениями.

Во втором прогоне мы попытались улучшить результаты «отправной точки» при помощи расширенного векторного представления документов. Второй прогон использует тот же алгоритм машинного обучения, что и первый прогон, но к словарному представлению документов добавляются понятия Тезауруса РуТез (подробности ниже).

Целью третьего прогона было испытание разработанного нами алгоритма машинного обучения, основанного на моделировании логики рубрикатора.

3.3.3.1 Прогон 1: SVM по леммам

Для прогона 1 мы использовали векторную модель документа, основанную на нормализованных словах. Все слова, встречающиеся в документе, были приведены к нормальной форме (лемме). Документ представляется множеством лемм, которые в него входят. Вес леммы

вычисляется по формуле TF*IDF [14, 56]. Леммы, встречающиеся менее чем в четырёх документах, были усечены.

В результате получилось 21746 различных лемм и 1203087 пар лемма-документ для обучающей выборки из 4496 документов.

Для оптимизации параметров SVM применялся метод, описанный в разделе 2.5.7.2 и в [6, 11].

SVM применялась только для тех рубрик, для которых было не менее четырёх документов в коллекции для обучения.

3.3.3.2 Прогон 2: SVM по леммам+понятиям

Для прогона 2 применялся метод SVM. Разница между прогоном 1 и прогоном 2 состоит лишь в использовании другого векторного представления документов.

Для прогона 2 мы расширили лемматическое векторное представление документов, использованное в прогоне 1. Каждый документ описывается набором лемм, которые в него входят (с TF*IDF-весами), плюс терминологическим индексом, основанным на понятиях Тезауруса РуТез. Терминологический индекс документа строится на этапе предварительной обработки программой Автоматической Лингвистической Обработки Текстов (АЛОТ) [29, 30].

Понятия, встречающиеся менее чем в четырёх документах, были усечены.

В расширенном индексе обучающей выборки документов получилось 29918 различных лемм/понятий и 1569958 пар «лемма/понятие»-документ.

3.3.3.3 Прогон 3: Метод машинного обучения, основанный на моделировании логики рубрикатора

В прогоне 3 использовался метод построения формул ПФА со следующими параметрами (раздел 3.1):

1. Шаг 1. В качестве векторного представления использовалось терминологическое представление на основе понятий Тезауруса. Термы, встречающиеся менее чем в четырёх документах, были усечены.
2. Шаг 2. Для набора конъюнктов перебирались все конъюнкции, состоящие из 50 термов с наибольшей F-мерой. Из списка полученных конъюнктов откидывались те, частотность которых меньше 2 или точность ниже порога 20%.
3. Шаг 3. Для основного теста использовались параметры формул (3.6) и (3.7): $w_{p1} = 5, w_{ap} = 4, w_{ar} = 1, w_r = 2$. Условия остановки: количество дизъюнктов больше 50; prec<10% и recl>90%; recl>99%.

3.3.3.4 Методика оценки результатов

Оценка результатов производилась по 50 случайно отобранным рубрикам из 170 рубрик, для которых было выполнено задание классификации [7]. Для вычисления оценок результаты автоматической классификации сравнивались с оценками, проставленными экспертами ИС «Кодекс».

В качестве основной метрики качества рубрицирования используется средняя F-мера по 50 выбранным рубрикам (см. раздел 2.2 — макроусреднение).

Для того, чтобы оценить корректность выборки рубрик, мы проанализировали распределение частотности документов для выбранных рубрик по сравнению с другими рубриками. На рисунке 3 показан график зависимости количества документов для обучения от номера рубрики. Рубрики, выбранные для оценки результатов (50 рубрик), выделены кругами. График показывает, что выбранные рубрики распределены в целом равномерно на множестве всех рубрик.

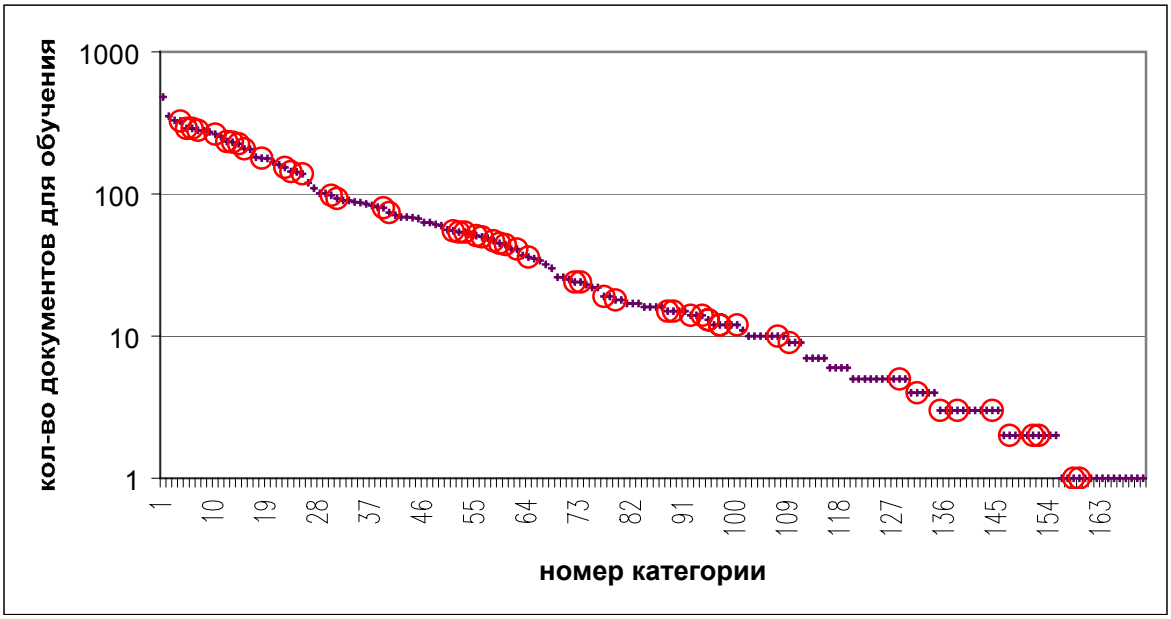

Рис. 3 Зависимость количества документов для обучения от номера рубрики в логарифмической шкале (рубрики упорядочены по убыванию частотности). Кругами выделены рубрики, выбранные для оценки.

3.3.3.5 Таблицы результатов

На рис. 4 представлены результаты прогонов участников. Наши прогоны обозначены “svm_lem”, “svm_thes” и “formul” соответственно. Из рисунка видно, что прогон 3 — алгоритм построения формул — показывает лучшие результаты по F-мере и по полноте, хотя и проигрывает по точности SVM и ещё двум алгоритмам. SVM показывает лучшие результаты по точности, но сильно проигрывает по полноте. В результате — более низкие результаты по F-мере, чем у прогона 3 и ещё одного алгоритма. Можно также отметить, что использование расширенного терминологического представления документов повышает результаты SVM, но ненамного.

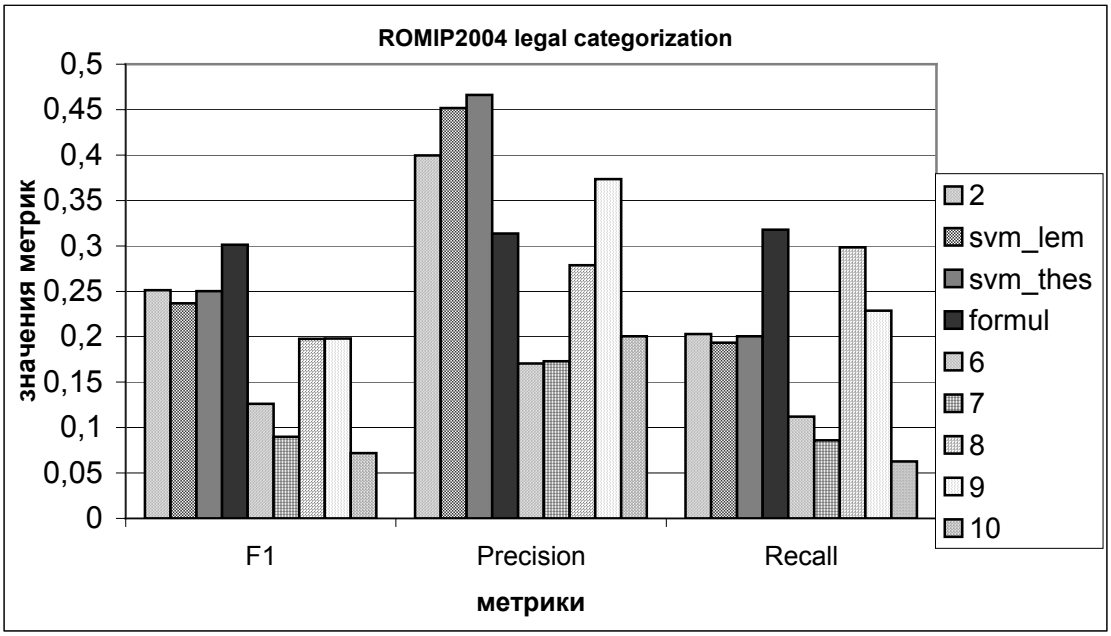

Рис. 4 Результаты прогонов участников по 50 выбранным рубрикам.

Мы проанализировали зависимость качества классификации (выражаемого F-мерой) от количества документов для обучения. Для этого множество рубрик было разбито на 4 части в зависимости от количества документов для обучения. Результаты показаны на рис. 5.

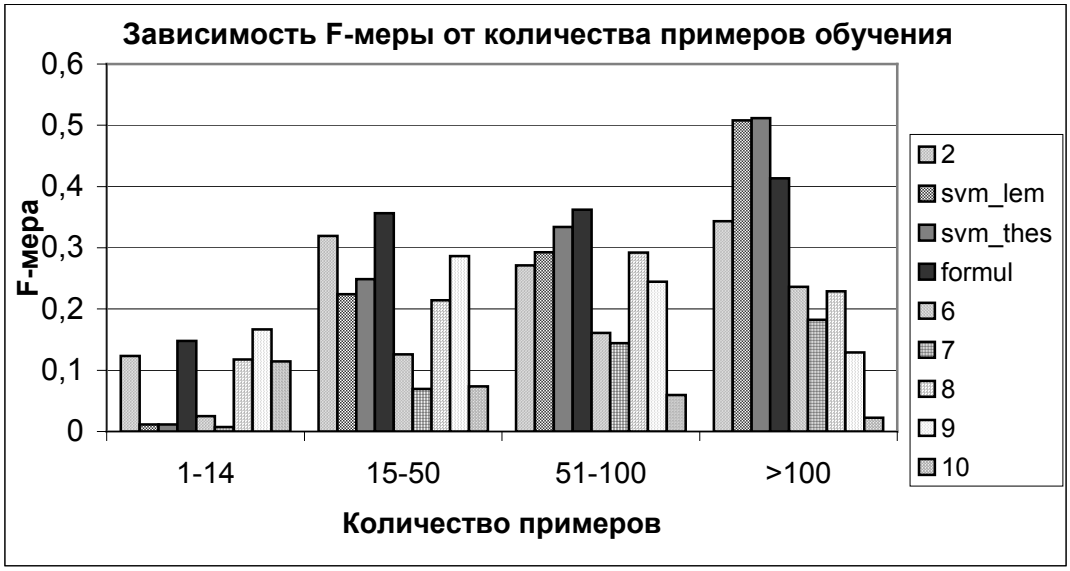

Рис. 5 Зависимость F-меры от количества примеров для обучения
(в среднем для рубрик, частотность которых
попадает в указанный интервал)

Можно отметить, что для метода SVM наблюдается резкая зависимость качества классификации от количества примеров для обучения — чем больше примеров, тем выше качество классификации. Для рубрик с частотностью выше 100 SVM показывает лучшие результаты. Для малочастотных рубрик качество классификации SVM падает до нуля (для рубрик с частотностью менее 4 SVM мы просто не запускали). Стоит отметить, что для некоторых алгоритмов наблюдается обратная зависимость. По-видимому, для малочастотных рубрик имеет смысл использовать другие алгоритмы.

Для алгоритма построения формул зависимость результатов от количества примеров выражена неярко. Однако для малочастотных рубрик (1-14 примеров) качество очень низкое. Возможно, это отчасти связано с тем, что на рубриках с частотностью менее 3 мы алгоритм построения формул не запускали (таких рубрик 5 из 17 в первом интервале).

3.3.3.6 Описания рубрик, полученные алгоритмом построения формул

Основной целью создания алгоритма построения формул было создание метода машинного обучения, который бы строил правила описания рубрики, которые можно легко интерпретировать. Покажем на примере нескольких рубрик, какие правила описания рубрики были построены алгоритмом.

Мы выбрали 6 рубрик с различными значениями количества документов для обучения. В таблице 3 представлены названия рубрик и правила, построенные алгоритмом (запросы к поисковой системе). Для каждой рубрики указано количество документов для обучения и значения F-меры для следующих алгоритмов:

- алгоритм построения формул на множестве обучения (обозначается train (t));
- алгоритм построения формул на множестве тестирования — то есть полученный результат на дорожке классификации (formul (f));
- SVM с расширенным векторным представлением — наш прогон 2 (svm_thes (s));
- наилучший результат, показанный участниками на этой рубрике (best (b)).

Рубрика (номер, имя, кол-во док-в)	**F-мера (train (t) formul (f) svm_thes(s) best(b))**
901800651 Основы государственного управления (327 документов)	t 37% f 30% s 38% b 38%
/Термин="ГОСУДАРСТВЕННЫЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ" OR (/Термин="ТЕРРИТОРИАЛЬНОЕ УПРАВЛЕНИЕ" AND /Термин="УТВЕРДИТЬ (ОКОНЧАТЕЛЬНО УСТАНОВИТЬ, ПРИНЯТЬ)") OR /Термин="ЛИЦЕНЗИРОВАНИЕ" OR /Термин="ФЕДЕРАЛЬНЫЙ ОРГАН ИСПОЛНИТЕЛЬНОЙ ВЛАСТИ" OR (/Термин="СТАТИСТИКА" AND /Термин="ИНФОРМАЦИЯ") OR (/Термин="ЗАМЕСТИТЕЛЬ МИНИСТРА" AND /Термин="КОНТРОЛЬ" AND /Термин="КОДЕКС") OR (/Термин="АННУЛИРОВАТЬ (ОБЪЯВИТЬ НЕДЕЙСТВ., ОТМЕНИТЬ)" AND /Термин="ПОЛОЖЕНИЕ (СВОД ПРАВИЛ, ЗАКОНОВ, КАСАЮЩ.ЧЕГО-Н.)" AND /Термин="ПРАВИТЕЛЬСТВО РОССИИ")	

Рубрика (номер, имя, кол-во док-в)	**F-мера (train (t) formul (f) svm_thes(s) best(b))**
9001375 Здравоохранение (208 документов)	t 68% f 61% s 71% b 71%
/Термин="МЕДИЦИНСКОЕ ОБОРУДОВАНИЕ" OR /Термин="МИНИСТЕРСТВО ЗДРАВООХРАНЕНИЯ"	
9001435 Учет и статистика (74 документа)	t 54% f 48% s 58% b 62%
/Термин="ГОСУДАРСТВЕННЫЙ КОМИТЕТ ПО СТАТИСТИКЕ"	
901716530 Право международных договоров (44 документа)	t 64% f 62% s 69% b 70%
/Термин="РАТИФИКАЦИЯ" OR (/Термин="ПОСТАНОВИТЬ" AND /Термин="СССР" AND /Термин="КРЕМЛЬ") OR /Термин="КОНСУЛЬСКАЯ КОНВЕНЦИЯ"	
9001711 Органы юстиции (15 документов)	t 41% f 37% s 18% b 37%
/Термин="ЮРИДИЧЕСКАЯ ЭКСПЕРТИЗА" OR /Термин="МИНИСТР ЮСТИЦИИ" OR /Термин="ГЛАВНОЕ УПРАВЛЕНИЕ ИСПОЛНЕНИЯ НАКАЗАНИЙ" OR (/Термин="КЛАССНЫЙ ЧИН" AND /Термин="НОТАРИАТ") OR /Термин="ЭЛЕКТРОННО-ЦИФРОВАЯ ПОДПИСЬ"	
3800301 Семейное право (9 документов)	t 58% f 26% s 5% b 42%

Рубрика (номер, имя, кол-во док-в)	**F-мера (train (t) formul (f) svm_thes(s) best(b))**
/Термин="ПРИЕМНАЯ СЕМЬЯ" OR /Термин="ПРИЕМНЫЙ РОДИТЕЛЬ" OR (/Термин="РОДИТЕЛЬСКОЕ ПОПЕЧЕНИЕ" AND /Термин="СВИДЕТЕЛЬСТВО О РОЖДЕНИИ") OR (/Термин="ПОСОБИЕ ПО БЕРЕМЕННОСТИ И РОДАМ" AND /Термин="УСЫНОВЛЕНИЕ" AND /Термин="РОЖДЕНИЕ РЕБЕНКА")	

Таблица 3. Описания рубрик, полученные алгоритмом построения формул. Для каждой рубрики указаны результаты различных алгоритмов (описание см. выше)

На основе анализа таблицы 3 можно сделать следующие выводы:

1. Для большинства рубрик (строки 2, 3, 4, 5, то есть 4 из 6) алгоритм создаёт формулы, соответствующие названию рубрики.
2. Для этих рубрик алгоритм показывает результаты, близкие к наилучшим.
3. Качество результатов на множестве обучения и множестве тестирования примерно одинаковое, то есть нет эффекта «переобучения».
4. Для двух рубрик алгоритм построил «плохие» формулы:
5. На первой рубрике «Основы государственного управления» алгоритм даёт длинную и довольно бессмысленную формулу. Однако на этой рубрике ВСЕ примененные алгоритмы показали невысокий результат — максимум 38%.
6. На последней рубрике алгоритм показал плохие результаты (отставание от лидера 26% против 42%, переобучение 58% train / 26% test, формула не соответствует рубрике). Возможно, это

связано с недостаточным количеством документов для обучения — 9 документов.

7. Примечательно, что формальное «качество» формулы (низкое значение F-меры) кореллирует с плохой оценкой построенной формулы человеком.

3.4 Выводы

Мы построили новый алгоритм машинного обучения для автоматической классификации текстов. Данный алгоритм строит описания рубрики в виде булевской формулы фиксированной структуры. В отличие от распространённых алгоритмов классификации текстов, используемое представление рубрики пригодно для анализа и модификации человеком и аналогично используемому экспертами при инженерном подходе.

На практике алгоритм может применяться для:

1. автоматической классификации текстов, аналогично другим алгоритмам машинного обучения;
2. описания рубрики в инженерном подходе, с последующим уточнением;
3. анализа логики работы экспертов, которые рубрицировали документы вручную, анализа расхождений в логике работы различных экспертов/методов классификации;
4. создания наглядных описаний (аннотации) коллекции документов, документирования принципов отнесения документов к рубрикам (автоматизированного создания комментариев к рубрикаторам).

Проведено аналитическое и экспериментальное исследование алгоритма.

Доказано, что при некоторых предположениях относительно рубрики и параметрах алгоритма будет построено описание рубрики, близкое к оптимальному. А именно, использовались следующие предположения:

1. рубрика описывается короткой формулой заданной структуры;
2. документы отрубрицированы без ошибок;
3. заданы достаточно «жёсткие» параметры алгоритма.

При условии выполнения этих предположений получены оценки параметров алгоритма, при которых достигается заданный уровень полноты/точности и длины формулы.

Экспериментальное исследование алгоритма проводилось на двух коллекциях документов: широко известной англоязычной коллекции Reuters-21578 и русскоязычной коллекции нормативных документов РОМИП’2004. Эксперименты показали высокую эффективность алгоритма и соответствие получаемых формул содержанию рубрики. В экспериментах на коллекции РОМИП’2004 алгоритм построения формул показал лучший результат, обогнав по качеству классификации SVM и алгоритмы других участников РОМИП’2004 — ещё 6 алгоритмов.

4 Тематический анализ коллекции документов

Одним из развиваемых нами подходов является улучшение существующих процедур классификации текстов, использующих инженерный подход. В этом разделе описаны результаты, полученные в этом направлении.

При создании правил описания рубрик эксперты используют свои знания о свойствах текстов, принадлежащих рубрике. Знания эксперта основываются, в первую очередь, на предыдущем опыте, в частности, на большой коллекции прочитанных ранее текстов, и во вторую очередь, на части текстов, подлежащих рубрицированию. Так как объем документов коллекции может быть очень большим, знания эксперта могут не отражать всё разнообразие тематики текстов, подлежащих рубрицированию. Основная идея предлагаемого подхода к повышению качества рубрицирования состоит в создании программ-помощников, которые предоставляют эксперту информацию о тематике текстов, подлежащих рубрицированию, основываясь на анализе полных текстов коллекции и рубрик, присвоенных документам (при наличии размеченной коллекции текстов).

Алгоритмы работы программы-помощников основаны на статистическом анализе коллекции документов с привлечением методов машинного обучения. Одним из таких алгоритмов является метод машинного обучения, описанный в разделе 3.

Программы-помощники позволяют ускорить описание рубрики, повысить точность описания, выявить некоторые ошибки классификации. Кроме того, некоторые средства оказались очень эффективны для использования в поисковой системе для расширения и уточнения запросов.

Мы опишем различные алгоритмы программ-помощников и примеры применения для реальных задач поиска документов и классификации текстов.

Описанные алгоритмы реализованы и встроены в полнотекстовую информационную систему УИС РОССИЯ.

4.1 Тематический анализ коллекции документов on-line

Вспомогательные средства поиска, основанные на статистическом анализе запроса и содержания документов, постепенно получают широкое распространение и внедряются в различных поисковых системах. Стоит отметить развитые средства анализа коллекции документов и агрегирования данных, развиваемые в течение многих лет компаниями TextWise (http://www.textwise.com) и Inxight (http://www.inxight.com). В поисковых машинах Teoma (http://www.teoma.com) [76] и Vivisimo (http://www.vivisimo.com) используется кластеризация найденных документов по классификатору тем, описываемых словосочетаниями. Из российских систем можно отметить Галактика-Зум (http://zoom.galaktika.ru) [21], основанную на выделении наиболее значимых слов и словосочетаний типа прилагательное/местоимение+существительное; различные подходы к визуализации результатов компании "Гарант-Парк-Интернет" (http://research.metric.ru).

4.1.1 Анализ по тезаурусу

Особенностью нашего подхода [4, 10] является использование качественного терминологического ресурса — Тезауруса РуТез.

Тематический анализ результатов запроса производится при помощи выделения понятий Тезауруса, наиболее характерных (контрастных) для документов, полученных в результате исполнения запроса. Список дескрипторов понятий упорядочивается по убыванию значимости и показывается рядом с результатами запроса. Степень важности понятия обозначается цветом — более значимые понятия имеют более теплые цвета.

Интерфейс пользователя [17, 15] позволяет уточнить запрос, добавив или удалив заинтересовавшее понятие в/из строки запроса (для этого достаточно одного нажатия клавиши "мыши"). Можно также войти в тезаурус, воспользоваться навигацией по иерархии понятий тезауруса для расширения/сужения запроса.

Опишем алгоритм вычисления коэффициента значимости (веса) понятия в результатах запроса. Для каждого понятия в документе на этапе предварительной обработки документа вычисляется коэффициент значимости (ранг) понятия в данном документе — число от 1 до 100. Ранг понятия в документе зависит от частоты встречаемости в документе и от тематической структуры документа (места в иерархии, так называемого, "тематического представления" содержания документа), вычисляемой на основе связей тезауруса [31]. Для всех понятий, встречающихся в результатах запроса, вычисляется вес (коэффициент значимости в результатах запроса) по формуле:

$$\mathrm{Weight}(t,q)=\mathrm{AvgRank}(t,q)^2 \cdot \mathrm{Recl}(t,q) \cdot \log\frac{1}{\mathrm{Cnt}(t)} \qquad (4.1)$$

где

- Weight(t,q) - вес понятия t;
- AvgRank(t,q) - средний ранг понятия t среди документов, встретившихся в результатах запроса q и содержащих данное понятие;
- Recl(t,q) - "полнота покрытия" понятия t по результатам запроса q, то есть отношение количества найденных документов, которые содержат термины данного понятия, к общему количеству найденных документов;
- Cnt(t) - частотность встречаемости понятия t по всей коллекции, то есть количество документов коллекции, которые содержат термины данного понятия.

Полученный список понятий упорядочивается по убыванию веса и выдаётся на странице результатов запроса.

Отметим, что формула (4.1) аналогична TF*IDF, но адаптирована для коллекции документов.

Тематический анализ результатов запроса реализован средствами СУБД Oracle9i, где хранятся все данные УИС РОССИЯ. Для быстрого тематического анализа документов применяется приближенная оценка по сокращенному списку найденных документов. В результате вычисление краткого тематического анализа результатов запроса занимает 1-2 секунды для любого запроса к информационной системе.

4.1.2 Анализ по метаданным

В УИС РОССИЯ поддерживаются различные типы коллекций из разных источников информации. Для каждого источника информации определено множество атрибутов (метаданных), в том числе:

1. дата публикации;
2. имена авторов;
3. название источника информации;
4. название организации, где проводилось исследование (для научных отчетов);
5. рубрики;
6. ключевые слова;
7. и т.п.

Алгоритм анализа результатов запроса по метаданным аналогичен алгоритму тематического анализа по понятиям Тезауруса, за исключением функции ранжирования. В качестве функции ранжирования используется количество документов, содержащих тот или иной атрибут.

4.1.3 Анализ с использованием алгоритма построения формул

Одним из алгоритмов интерактивного анализа результатов запроса является алгоритм построения формул над понятиями тезауруса, описанный в разделе 3.1. В данном случае в качестве обучающией выборки на вход алгоритма ПФА подаётся всё множество документов коллекции, по которой производится поиск. При этом множество документов, попавших в результаты запроса, образуют множество положительных примеров, а остальные документы коллекции — множество отрицательных документов. В качестве результатов анализа пользователю показывается формула, построенная алгоритмом ПФА и описывающая множество документов, найденных по запросу пользователя.

Всвязи с тем, что алгоритм ПФА требует значительно больших временных ресурсов, чем алгоритм тематического анализа, описанный в разделе 4.1.1, интерактивный анализ с использованием алгоритма ПФА используется только для относительно небольших (до 50000) коллекций документов.

4.1.4 Применение тематического анализа в ИС

Тематический анализ и анализ по метаданным являются эффективным инструментом, повышающим функциональность информационной системы. В этом разделе мы опишем применения тематического анализа для поиска документов. Методы повышения эффективности рубрицирования, основанные на тематическом анализе, мы опишем более подробно в следующем разделе.

4.1.4.1 Оценка тематики документов, найденных пользователем по запросу к поисковой системе

Выдаваемые поисковой системой УИС РОССИЯ списки представляют собой набор ключевых понятий для данной выборки документов. Это

позволяет оценить тематику полученного набора документов, не просматривая каждый документ. Также можно оценить количество документов, относящихся к той или иной теме.

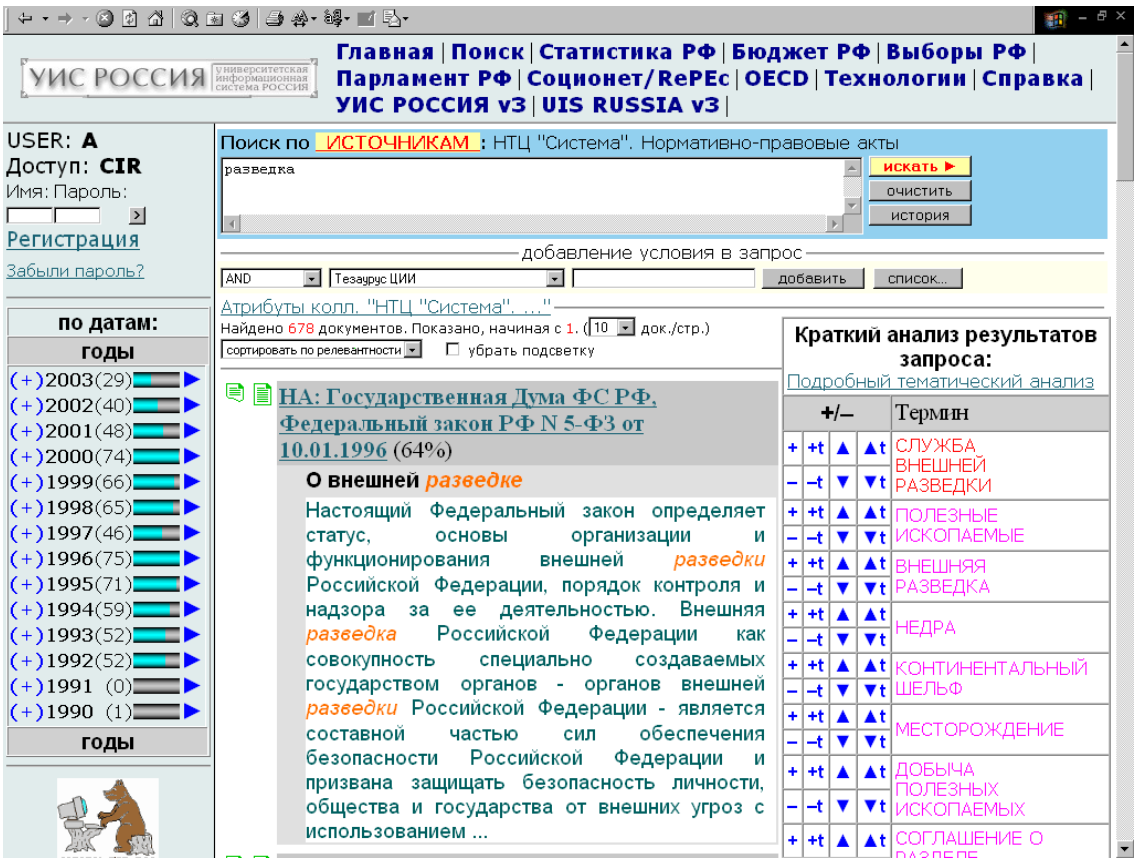

Рис. 6 Страница результатов запроса для запроса «разведка». В правой колонке — результаты тематического анализа результатов запроса. В левой колонке — анализ по метаданным (поле «дата документа»).

Например, по запросу "разведка" по коллекции нормативно-правовых актов РФ (НТЦ "Система") получаем список понятий (см. рис. 6):

+(277) +t(278) СЛУЖБА ВНЕШНЕЙ РАЗВЕДКИ;

+(141) +t(414) ПОЛЕЗНЫЕ ИСКОПАЕМЫЕ;

+(144) +t(144) ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПРАВИТЕЛЬСТВЕННОЙ СВЯЗИ И ИНФОРМАЦИИ;

+(207) +t(350) ДОБЫЧА ПОЛЕЗНЫХ ИСКОПАЕМЫХ;

+(140) +t(410) НЕДРА;

+(126) +t(372) МЕСТОРОЖДЕНИЕ;

+(202) +t(202) ФЕДЕРАЛЬНАЯ СЛУЖБА БЕЗОПАСНОСТИ;

+(253) +t(276) МИНИСТЕРСТВО ОБОРОНЫ

...

В этом списке для каждого понятия около знака "+" указано количество документов, содержащих данное понятие и около знака "+t" количество документов, содержащих термины понятия при расширении по иерархии тезауруса (“дереву тезауруса”).

4.1.4.2 Интерактивное уточнение (сужение) запроса

С помощью тематического анализа легко обрабатывать многозначные запросы, в частности приведённый в предыдущем примере. Здесь множество документов, найденных по слову "разведка" распадается на две темы: "геологическая разведка" и "разведывательная деятельность". Можно уточнить запрос, "кликнув" один раз "мышкой" на ссылку "+" или "+t" рядом с нужным понятием, что приведет к появлению в текстовом поле запроса дополнительного условия, например:

/Термин_расш="ГЕОЛОГИЧЕСКАЯ РАЗВЕДКА".

Исполняя модифицированный запрос, можно получить документы, относящиеся только к указанной теме, включая документы, содержащие также такие понятия как "ГЕОЛОГИЧЕСКИЕ РАБОТЫ", "ГЕОЛОГИЧЕСКИЙ ПОИСК", "ГЕОЛОГИЧЕСКОЕ ИЗУЧЕНИЕ"; "ГЕОЛОГИЧЕСКОЕ ИССЛЕДОВАНИЕ" и т.д.

4.1.4.3 Двуязычный поиск документов

Так как тезаурус двуязычен, то можно получить тематический анализ коллекции англоязычных документов на русском языке и наоборот (см. рис. 7 ниже).

4.1.4.4 Поиск документов, похожих на данный

Данную функцию сейчас реализуют многие системы, хотя результат, зачастую, оставляет желать лучшего. При помощи средств анализа результатов запроса пользователь может сформировать запрос, используя понятия, содержащиеся в найденном документе. Данный подход отличается большей гибкостью по сравнению с жестким заданием функции похожести документов, так как пользователь может самостоятельно задать "в каком смысле" нужно искать похожие документы.

4.1.4.5 Выявление скрытых зависимостей между темами, объектами, событиями на основе анализа коллекции документов

Список поняий, полученный в результате анализа результатов запроса, иногда содержит понятия, связь которых с тематикой запроса неочевидна. Интерактивные средства уточнения запроса позволяют выявить связь между тематикой запроса и найденными в результате анализа понятиями.

4.1.4.6 Отслеживание временных закономерностей обсуждения данной темы

Анализ результатов запроса по дате публикации документа предоставляет пользователю информацию о количестве документов, относящихся к теме запроса и опубликованных в данный период времени (день/месяц/год). Пользователь может исследовать временные закономерности обсуждения заданной теме по количеству публикаций, релевантных данной теме.

Рассмотрим пример. Пользователь вводит запрос (/Термин_расш=«ИРАК» AND /Дата=«2003») и выполняет поиск по коллекции средств массовой информации. Результатом запроса является 3208 статей СМИ, выпущенных в 2003 году (см. рис. 7). Каждый документ

содержит слово «ИРАК», либо его синонимы (ИРАКСКИЙ, ИРАКСКАЯ РЕСПУБЛИКА), либо понятия, расположенные ниже по иерархии тезауруса (БАГДАД, БАСРА, КИРКУК, ИРАКСКИЙ ДИНАР и т.д.).

В левой панели результатов запроса показана диаграмма распределения документов по датам. Для каждого месяца 2003 года указано количество документов, выпущенных в этом месяце. Из диаграммы явно видно, что большинство документов были опубликованы в марте и апреле (это время начала войны в Ираке).

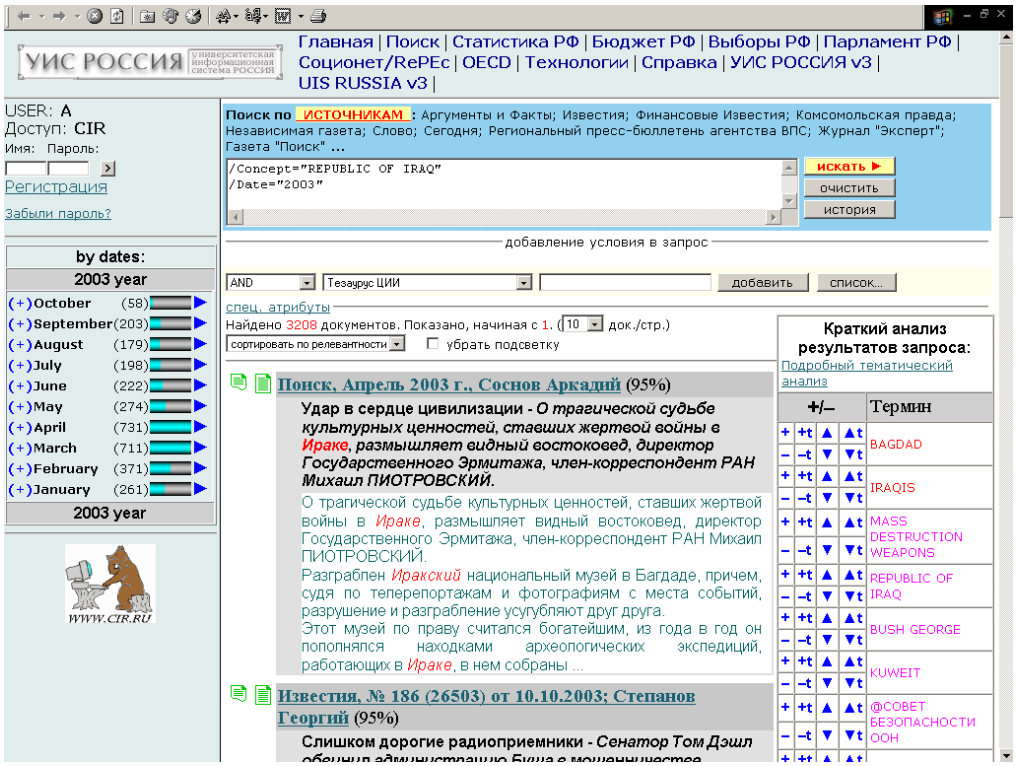

Рис. 7 результаты запроса (/Термин_расш=«ИРАК» AND /Дата=«2003»). В левой панели — диаграмма анализа по датам и частотность документов по месяцам.

4.1.4.7 Анализ связей между авторами статей, научными организациями и изучаемыми темами

Вот ряд задач, которые можно решать при помощи средств анализа результатов запроса по метаданным:

- Найти авторов, занимающихся данной проблемой

- Найти организации, которые исследуют данные вопросы
- Найти журналы, где публикуются данные авторы
- Найти соавторов данных авторов
- Выявить смежные темы, которыми занимаются те же авторы/организации

Рассмотрим пример. Пользователь вводит запрос «migration» и выбирает поиск по коллекции научных материалов Соционет/RePec. В результате запроса найдено 1419 документов (см. рис. 8).

Пользователь может выбрать анализ результатов запроса по любому из полей метаданных для коллекции Соционет (автор статьи, организация, где работает автор, название дисциплины, ключевые слова, рубрики JEL). Если выбрать «анализ по авторам», то в правой панели появится список авторов, опубликовавших статьи по теме миграции. Далее, можно найти все документы одного из авторов, и проанализировать тематику публикаций (по рубрикам JEL, ключевым словам или понятиям Тезауруса). Таким образом, с помощью интерактивных средств анализа по тезаурусу и метаданным можно анализировать взаимное распределение различных атрибутов для любого запроса к информационной системе.

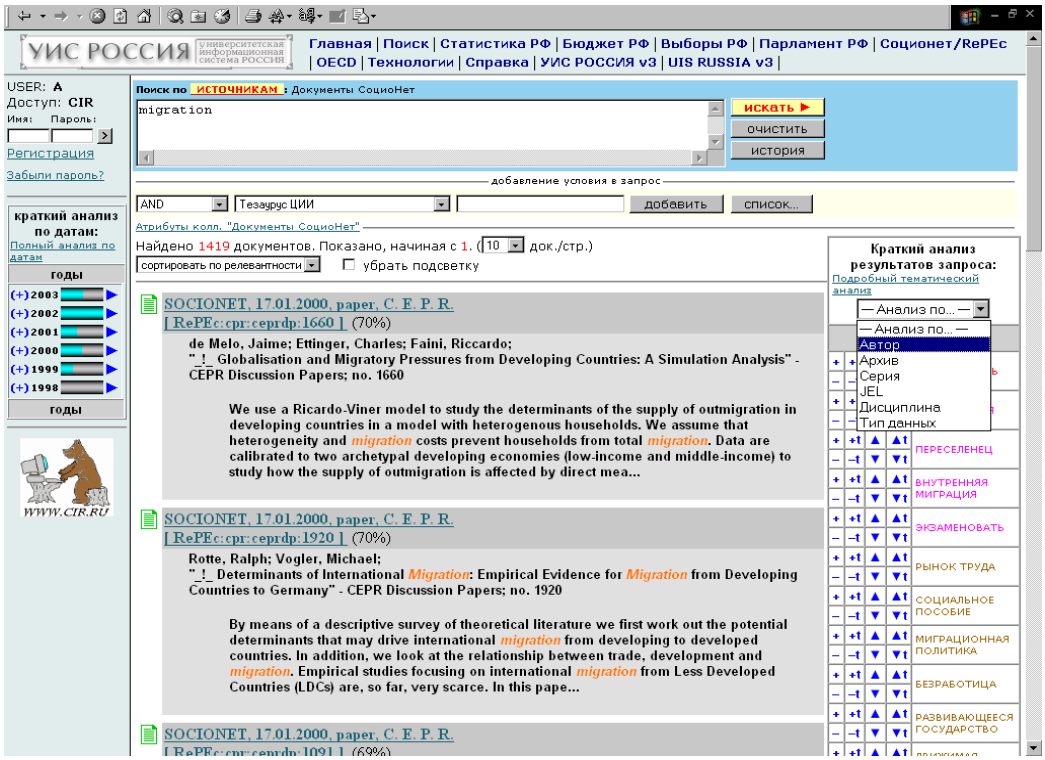

Рис. 8. Анализ по атрибутам документов, найденных в результате запроса. В правой панели можно выбрать тип анализа.

4.2 Повышение эффективности рубрицирования, основанное на тематическом анализе

В сложных задачах рубрикации единственным способом решения задачи является итерационное уточнение правил рубрицирования. Уточнение рубрикации производится на основе сравнения результатов автоматической рубрикации и результатов ручного рубрицирования. Выявленные несоответствия могут происходить как от некорректной классификации документов вручную, так и по причине некорректного описания рубрик для автоматической классификации. В данном разделе мы проведём классификацию различных проблем ручного и автоматического рубрицирования и опишем способы ускорения процедур уточнения рубрикации, базирующиеся на интерактивном использовании тематического анализа коллекции документов.

Данный раздел разработан автором при взаимодействии с экспертами проекта УИС РОССИЯ, составляющими описания рубрик. Экспертами были выявлены проблемы, возникающие при описании рубрикатора в «традиционной» для УИС РОССИЯ технологии построения системы рубрицирования. Автором были предложены методы решения возникающих проблем на основе использования тематического анализа коллекции документов и разработаны соответствующие инструменты, встроенные в УИС РОССИЯ.

4.2.1 Общие проблемы ручной классификации для больших рубрикаторов

В процессе анализа результатов ручного рубрицирования по большим рубрикаторам, даже проводимого высококвалифицированными экспертами, было выявлено три типа проблем ручного рубрицирования [5].

4.2.1.1 Проблема определения и соблюдения ограничивающих правил рубрицирования

Суть проблемы заключается в том, что ограничивающие правила рубрицирования, не связанные непосредственно с формулировкой конкретной рубрики, являются серьезной базой для субъективизма:

- об этих правилах забывает часть экспертов,
- для разных рубрик эти правила соблюдаются с разной степенью последовательности,
- эти правила неизвестны пользователю, в большой степени он опирается на буквальную формулировку рубрики.

Выбор такого рода правил напрямую зависит от четкого определения ролей рубрикатора в информационно-правовой системе, взаимодействия этих ролей с ролями других типов информации (например, указателей – действующий или не действующий документ), моделью пользователя

системы, сценариями работы различных типов пользователей с рубрикатором [40].

4.2.1.2 Проблема документов, отнесенных экспертами к рубрике ошибочно

Процент таких документов в общем количестве документов обычно невелик. Важность нахождения такого рода документов состоит в том, что с большой вероятностью ошибочная рубрика проставлена вместо правильной рубрики, и данный документ не будет найден по правильной рубрике.

4.2.1.3 Проблема пропущенных экспертами документов

Нахождение пропущенных экспертами документов является непростой задачей, и может усугубляться проблемой существования ограничивающих правил, по которым не приняты окончательные решения, и большого количества «промежуточных документов», для которых неясно, должны ли они принадлежать рубрике или нет.

Важным шагом является определение набора документов-кандидатов в рубрику для дополнительного просмотра. Наиболее оптимальным здесь, видимо, является опора на результаты анализа результатов автоматического и ручного рубрицирования. Представляются полезными следующие шаги:

- первичный анализ результатов автоматического и ручного рубрицирования, выявление рубрик, в которых явление пропуска релевантных документов носит массовый характер;
- консультация с экспертами, что документы, которые показались релевантными, действительно такими являются;
- по результатам классификации исправление результатов автоматического рубрицирования, чтобы достичь максимального значения полноты без снижения содержательной точности (т.е. без

дополнительного появления в результатах автоматического рубрицирования явно нерелевантных документов);

- представление экспертам новых результатов автоматического рубрицирования;
- на множестве документов, помещенных в рубрику автоматической системой, но не взятых в рубрику экспертами, эксперты должны просмотреть практически все документы в этом множестве один за другим и решить, каким документам добавить анализируемую рубрику;
- возможно, необходимо использовать систему решений не из двух значений (принадлежит рубрике или не принадлежит), а из трех: добавить еще – условно принадлежит – в случаях расхождения между экспертами или неясности решения.

4.2.2 Использование информеров при решении задач классификации

При формировании или модифицировании логической формулы, описывающей рубрику, необходимо производить различные оценки полноты и точности рубрикации. Информеры УИС РОССИЯ (см. рис. 6-8) позволяют экспертам производить данные оценки интерактивно, что повышает эффективность труда — работа ускоряется, и результаты имеют лучшие показатели по критерию полноты и точности.

Опишем алгоритм работы специалиста по рубрикации для решения различных задач поддержки рубрицирования по сложному рубрикатору.

4.2.2.1 Создание терминологического описания для рубрики

По содержанию рубрика обычно разделяется на несколько элементарных единиц, упоминания которых прямо или косвенно должны быть найдены в тексте документов рубрики.

Для того чтобы составить для рубрики терминологическое описание, необходимо выявить элементарные смыслы рубрики, найти, какими терминами эти смыслы могут выражаться. Далее необходимо записать булевское выражение, в котором понятия, выражающие разные составляющие смыслы рубрики, будут соединяться конъюнкцией, а понятия, выражающие один и тот же смысл дизъюнкцией.

Одним из способов начального набора понятий в рубрику является использование алгоритма построения формул, описанного в разделе 3. Другим способом является набор формулы вручную с использованием инструментов тематического анализа коллекции документов.

Рассмотрим «модельную» рубрику «ИСЧИСЛЕНИЕ АКЦИЗОВ ПРИ ИМПОРТЕ».

Каждый текст, относящийся к этой рубрике, должен содержать термины, относящиеся к сфере импорта, и термины, относящиеся к сфере акцизов.

Выполняем поиск по рубрике – получаем набор документов, отнесенных к рубрике экспертами.

Выбираем из информера понятия, относящиеся к акцизам: *ПОДАКЦИЗНЫЙ ТОВАР*, *АКЦИЗ*, *МАРКА АКЦИЗНОГО СБОРА*. Удаляем из выдачи документы, содержащие эти понятия, чтобы определить, какие еще понятия могут относиться к сфере акцизов.

Собираем теперь понятия, относящиеся к *импорту*. Возвращаемся к запросу по рубрике. Изучаем информер — имеется понятие *ИМПОРТ*. Удаляем документы, включающие это понятие, из выдачи.

Информер больше понятий не дает. Начинаем изучать оставшиеся тексты. В текстах содержатся слова *ввоз, ввезти, ввозить, ввозной*. Убираем эти документы.

В информере появились понятия *ТАМОЖЕННАЯ ПОШЛИНА, ТАМОЖЕННОЕ ОФОРМЛЕНИЕ ТОВАРОВ, ГОСУДАРСТВЕННЫЙ*

ТАМОЖЕННЫЙ КОМИТЕТ. В сочетании с вопросами акцизами эти понятия должны указывать на импорт.

Таким образом, мы получаем формулу:

```
(ПОДАКЦИЗНЫЙ ТОВАР
 или АКЦИЗ
 или МАРКА АКЦИЗНОГО СБОРА)
и
(ИМПОРТ
 или ВВОЗ
 или ТАМОЖЕННАЯ ПОШЛИНА
 или ТАМОЖЕННОЕ ОФОРМЛЕНИЕ ТОВАРОВ
 или ТАМОЖЕННЫЙ КОМИТЕТ)
```

На каждом шаге происходит контроль оставшегося количества документов. Процесс уточнения формулы прекращается, если достигнут требуемый уровень ошибки.

Если название рубрики выглядит как состоящее из одного термина, то это часто не означает, что достаточно упоминания этого термина в тексте, чтобы присвоить тексту рубрику. Часто такой текст должен обсуждать какие-то значимые для данного понятия части, свойства и ситуации.

Так, тексты в рубрике «ОБЩЕСТВА С ОГРАНИЧЕННОЙ И С ДОПОЛНИТЕЛЬНОЙ ОТВЕТСТВЕННОСТЬЮ» должны содержать не только термины «*общество с ограниченной ответственностью»* или «*общество с дополнительной ответственностью»*, но и обсуждать такие важнейшие аспекты для этих организаций, как создание, регистрация, учредители, уставный капитал, собственность и т.п.

Таким образом, реально рубрика также разлагается на два элементарных смысла — тот, что назван в формулировке, и что-то вроде «общие вопросы», и описывать рубрику нужно в виде конъюнкции двух

частей. Понятия, которые нужно включить во вторую часть конъюнкции, т.е. те которые важны для функционирования первой части, могут быть набраны из правой панели экранного интерфейса. Для упомянутой рубрики на правой панели мы увидим: *УСТАВНЫЙ КАПИТАЛ*, *УЧРЕДИТЕЛЬ*, *РЕГИСТРАЦИЯ ЮРИДИЧЕСКИХ ЛИЦ*, *СОВЕТ ДИРЕКТОРОВ*.

4.2.2.2 Использование программы автоматической рубрикации для нахождения ошибок ручного рубрицирования

Для нахождения ошибочных документов в совокупности документов, приписанных рубрике экспертами, необходимо убедиться, что каждый из документов рубрики, упоминает явно или косвенно каждый их элементарных смыслов.

Выполним запрос на поиск документов, приписанных экспертами рубрике «СТРАХОВЫЕ ВЗНОСЫ В ПЕНСИОННЫЙ ФОНД».

Выберем и мысленно зафиксируем один из элементарных смыслов. Например, выберем понятие *ПЕНСИОННЫЙ ФОНД*. Нам нужно проследить его наличие в каждом документе рубрики. Для этого будем выбирать на правой панели понятия, которые могут выражать в тексте этот смысл, например, *ГОСУДАРСТВЕННЫЙ ПЕНСИОННЫЙ ФОНД*, *ПЕНСИОННОЕ СТРАХОВАНИЕ*, *ПЕНСИОННЫЙ ФОНД*.

Используя кнопку “-“, удаляем из выдачи документы, содержащие эти понятия. То есть на множестве документов рубрики выполняем запрос:

```
/CLASS=    "СТРАХОВЫЕ ВЗНОСЫ В ПЕНСИОННЫЙ ФОНД"
AND NOT   /Термин="ГОСУДАРСТВЕННЫЙ ПЕНСИОННЫЙ ФОНД"
AND NOT   /Термин="ПЕНСИОННОЕ СТРАХОВАНИЕ"
AND NOT   /Термин="ПЕНСИОННЫЙ ФОНД"
```

Смотрим еще раз на правую колонку, и если находим еще понятия, соответствующие выбранному элементарному смыслу, то удаляем содержащие их документы и т.д. Так продолжаем, пока правая колонка уже не содержит такого рода понятий.

Если документы уже закончились, то это означает, что выбранный смысл найден в каждом из документов, и можно переходить к следующему смыслу.

В противном случае, необходимо вызывать на экран оставшиеся документы и, читая их, понять, какие слова или термины в них указывают на искомый элементарный смысл.

В нашем случае выяснилось, что многие из оставшихся текстов содержат аббревиатуру ПФР. Удаляем из выдачи документы, содержащие найденное слово или термин.

Повторяем процедуру, ища понятия, соответствующие элементарному смыслу, на правой панели, или слова внутри текста.

Находим, что многие оставшиеся тексты содержат формулу «страховые взносы во внебюджетные фонды» и понятие ВНЕБЮДЖЕТНЫЙ ФОНД в правой колонке, удаляем документы с эти понятием.

В результате повторения процедуры остаются документы, отнесение которых к рубрике регулируется не содержимым, но внешними параметрами («Внесение изменений», «Досье на проект») и ошибочные документы.

Для нахождения пропущенных экспертами релевантных документов необходимо сначала сформировать множество документов, в которых весьма вероятно могут находиться такие документы. В качестве такого множества могут служить документы из выдачи процедуры автоматической рубрикации для данной рубрики и (или) документы, выданные по запросу – булевскому выражению из слов и (или) понятий, сформированному на основе

формулировки рубрики, например, (СТРАХОВОЙ ВЗНОС and ПЕНСИОННЫЙ ФОНД).

Из полученной таким образом выдачи документов необходимо удалить документы, приписанные рубрике экспертами.

Результирующее множество документов необходимо изучить. Здесь выполняем следующую процедуру.

По содержанию документы в результирующем множестве могут подразделяться на несколько классов:

- документ явно нерелевантен;
- документ явно релевантен – пропущенный документ найден и должен быть добавлен к множеству документов рубрики,
- документ касается темы рубрики, но акцент документа несколько смещен – таких документов в рассматриваемом множестве может быть достаточно много.

Для рассмотрения последнего типа документов необходимо выполнить следующие шаги:

1) необходимо выяснить, сколько документов, похожих на найденный, включено экспертами в рубрику. Для этого на множестве документов, полученных в результате ручной рубрикации, выполняется булевский запрос из слов и понятий, наиболее полно отражающий суть документа;

2) по всему этому множеству документов должно быть принято решение о включении (не включении) в рубрику;

2а) или все эти документы должны быть включены в рубрику, и тогда к рубрике нужно приписать соответствующее правило о включении и добавить найденный документ;

2б) если принято решение не включать такой тип документов, тогда правило не включения также должно быть зафиксировано, а подобные

документы, прежде включенные в рубрику, должны быть удалены из нее как ошибочные.

После анализа документа необходимо по возможности как можно точнее описать его основное содержание в виде булевского запроса и удалить всю совокупность аналогичных документов из рассматриваемого множества, после чего начинать рассмотрение следующего документа.

4.2.2.3 Итерационное повышение полноты автоматического рубрицирования

Для повышения полноты автоматического рубрицирования необходимо найти понятия, которые выражают элементарные смыслы рубрики, но не были учтены в текущем терминологическом описании рубрики.

Для этого из множества документов, приписанных рубрике экспертами, вычитается множество документов, помещенное в ту же рубрику при автоматическом рубрицировании, т.е. формируется набор документов рубрики, на котором программа автоматического рубрицирования проработала неудачно.

Пропущенные элементарные смыслы пытаемся найти на правой панели экрана. Удаляем из набора документы, содержащие эти понятия (используем кнопку «-» на правой панели).

Продолжаем поиск дополнительных понятий для включения в терминологическое описание на правой панели.

В некоторый момент мы не можем найти добавления в терминологическое описание ни на правой панели, ни в текстах документов.

Если документы остались, то обычно это документы трех видов:

- документы, отнесенные к рубрике экспертами ошибочно,
- документы вида «внесение изменений, не содержащие в явном виде смысловых элементов рубрики,

- документы, в которых присутствуют все элементарные смыслы рубрики, но рубрика получает при автоматическом рубрицировании слишком небольшой вес (например, потому, что текст большой, а релевантная фраза одна).

4.2.2.4 Итерационное повышение точности автоматического рубрицирования

Для определения способов повышения точности автоматического рубрицирования необходимо получить набор документов, которые были включены в рубрику в процессе автоматического рубрицирования, но не были включены в рубрику экспертами. Для этого в оболочке УИС РОССИЯ необходимо выполнить запрос по рубрике для документов, отнесенных к этой рубрике в процессе автоматического рубрицирования, а затем удалить из выдачи, те документы, которые были включены в рубрику экспертами.

Полученные документы и необходимо изучить, просматривая их один за другим.

Могут встретиться следующие случаи:

1) очередной документ релевантен – это означает, что программа отработала правильно, а эксперты пропустили документ и не включили его в рубрику

2) для очередного документа непонятно, должен ли он включаться в рубрику – необходимо задать дополнительные вопросы по поводу правил экспертного рубрицирования

3) очередной документ явно нерелевантен.

Для выяснения причин нерелевантности документа, нужно сравнить содержание документа с терминологическим описанием рубрики и выяснить, какие именно термины или совокупности терминов привели к проставлению этой рубрики программой.

Причинами появления нерелевантной рубрики у документа могут быть следующие:

3.1) В терминологическом описании содержится понятие без дополнительных условий, и именно по нему текст был отнесен к рубрике. Если появление таких нерелевантных текстов по данному понятию – массовое явление, то в терминологическое описание рубрики необходимо добавить к этому понятию дополнительные условия, в виде тех понятий, которые также должны встретиться в тексте;

3.2) Текст приписан к рубрике на основе двух различных понятий, встретившихся в этом тексте – в терминологическом представлении рубрики была записана конъюнкция этих двух понятий. Совместная встречаемость этих понятий в тексте иногда дает анализируемую рубрику, но достаточно часто приводит к ложной рубрикации. Например, если при описании терминологической формулы для рубрики «Страховые взносы в Пенсионный Фонд» в формулу была бы включена (или случайно образовалась) конъюнкция *ПЛАТЕЖ* и *ПЕНСИЯ*, то часто эта пара понятий давала бы тексты о выплате пенсий, а не о платежах в Пенсионный фонд.

Для исправления возникшей ситуации могут быть сделаны следующие шаги:

1) Возможно, можно обойтись без данной пары понятий в конъюнкции терминологического описания. Конъюнкции для каждого понятия из пары нужно сделать уже, не включая неудачную пару.

2) Возможно данную пару понятий нужно уточнить дополнительными условиями, т.е. превратить конъюнкцию из пары в тройку

3) Возможно, из этих двух понятий нужно образовать более длинный термин. Так, мы пытались сделать терминологическое описание для рубрики «НАЛОГ НА ПРИОБРЕТЕНИЕ АВТОТРАНСПОРТНЫХ СРЕДСТВ», как конъюнкцию *налог* + *приобретение* + *автотранспортное средство*, но затем пришли к выводу, что наилучший результат автоматическое рубрицирование

даст, если ввести в тезаурус такой длинный термин и построить терминологическое описание данной рубрики на базе этого термина.

3.4) Ложную рубрику дает неправильно разрешенная многозначность термина, как это было с термином *журнал* для рубрики «ГАЗЕТЫ, ЖУРНАЛЫ» или термином *единый налог* для рубрики «УЧЕТ И ОТЧЕТНОСТЬ ПО ЕСН». Если явление массовое, то может помочь внесение в тезаурус дополнительных однозначных терминов, содержащих обнаруженный многозначный термин, в качестве составной части, например, *журнал учета, кассовый журнал* и т.п.

3.5) Возможно, что текст нерелевантен, потому что существует правило, о том, что такого рода тексты должны относиться к другой рубрике. Данное правило может быть записано в списке правил нормативно-правового рубрицирования.

3.6) Несмотря на все предпринятые усилия, может сохраняться явление так называемой ложной корреляции, когда одна и та же пара терминов в тексте иногда дает правильную рубрику, а иногда нет.

Так, например, при анализе результатов автоматического рубрицирования для рубрики «СОЗДАНИЕ, РЕОРГАНИЗАЦИЯ И ЛИКВИДАЦИЯ ТАМОЖЕН И ТАМОЖЕННЫХ ПОСТОВ» была выявлена группа явно нерелевантных документов, полученных при автоматическом рубрицировании, когда создаются или ликвидируются склады, комиссии, зоны при таможнях.

С этой проблемой очень трудно бороться, однако в наших экспериментах она встречается только примерно в 3% рубрик.

4.3 Выводы

В сложных задачах рубрикации, когда методы машинного обучения не применимы либо не дают требоемого уровня качества классификации, единственным способом решения задачи является итерационное уточнение

правил рубрицирования. Уточнение рубрикации производится на основе сравнения результатов автоматической рубрикации и результатов ручного рубрицирования.

В данной главе описаны средства тематического анализа коллекции документов, расширяющие возможности полнотекстовой информационной системы. Разработана методика применения указанных средств для итерационного уточнения правил классификации, разрабатываемых экспертами при инженерном подходе. Дана классификация различных проблем, возникающих при описании рубрик, и предложены методы решения с использованием средств анализа коллекции документов.

Предложенные средства повышают скорость работы экспертов, которые строят описания рубрик, и позволяют устранить ряд ошибок, возникающих из-за различного толкования смысла рубрик.

Кроме того, разработанные средства применяются в качестве эффективного средства поиска и анализа информации в полнотекстовой информационной системе УИС РОССИЯ.

5 Заключение

К основным результатам, полученным автором и описанным в данной диссертации (главы 3 и 4), относятся:

1. Разработан новый метод машинного обучения для автоматической классификации текстов, основанный на моделировании логики работы экспертов. Разработанный метод создаёт булевские формулы описания рубрики, пригодные для анализа и доработки экспертами, создающими методы классификации текстов, основанные на знаниях.
2. Доказано, что при некоторых предположениях относительно рубрики и параметрах разработанного алгоритма, будет построено описание рубрики, близкое к оптимальному. Получены оценки параметров алгоритма, при которых достигается заданный уровень полноты/точности и длины формулы.
3. Проведено экспериментальное исследование разработанного алгоритма. Экспериментально доказана высокая эффективность алгоритма и соответствие получаемых формул содержанию рубрики. В экспериментах на коллекции РОМИП'2004 (дорожка тематической классификации Российского семинара по Оценке Методов Информационного Поиска 2004 года) алгоритм построения формул показал лучший результат по сравнению с 8 другими алгоритмами классификации текстов.
4. Разработаны средства интерактивного тематического анализа коллекции документов и анализа по метаданным, основанные на статистическом анализе распределения атрибутов документов и методе машинного обучения, основанном на моделировании логики рубрикатора. Разработанные средства расширяют возможности полнотекстовой информационной системы.

5. Разработана методика применения средств тематического анализа для итерационного уточнения правил классификации, разрабатываемых экспертами при инженерном подходе. Предложены методы решения ряда проблем, возникающих при описании рубрик в «инженерном» подходе, с использованием средств анализа коллекции документов. Предложенные средства повышают скорость работы экспертов, которые строят описания рубрик, и позволяют устранить ряд ошибок, возникающих из-за различного толкования смысла рубрик.

Данная работа объединяет два различных подхода к построению систем автоматической классификации текстов: методы машинного обучения и методы, основанные на знаниях. Разработанные методы позволяют эффективно решать задачу классификации текстов за счёт использования преимуществ обоих подходов.

По теме диссертационной работы опубликовано 18 печатных работ. Основное содержание диссертации отражено в публикациях [1-8, 10, 11, 18].

Описанные алгоритмы и технологии реализованы и внедрены в технологический процесс построения систем классификации текстов проекта УИС РОССИЯ, разрабатываемого в НИВЦ МГУ.

6 Список литературы

Публикации автора по теме диссертации

[1] Агеев М.С., Добров Б.В., Макаров-Землянский Н.В. Метод машинного обучения, основанный на моделировании логики рубрикатора. // RCDL'2003 Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Пятая всероссийская науч. конф. — Санкт-Петербург, 2003.

[2] Ageev M., Dobrov B., Loukachevitch N. Text Categorization Tasks for Large Hierarchial Systems of Categories // SIGIR 2002 Workshop on Operational Text Classification Systems / Eds. F.Sebastiani, S.Dumas, D.D.Lewis, T.Montgomery, I.Moulinier — Univ. of Tampere, 2002 — p.49-52.

[3] Агеев М.С. Метод машинного обучения для автоматической классификации текстов. // Труды XXVI Конференции молодых ученых механико-математического факультета МГУ. Москва, Мехмат, МГУ, 2004. (в печати).

[4] Ageev M., Dobrov B., Makarov-Zemlyanskii N. On-line Thematic and Metadata Analysis of Document Collection // New Trends in Intelligent Information Processing and Web Mining'2004: Proceedings of the International Conference / Springer, Advanced in Soft Computing — Zakopane, Poland, May 2004 — pp 279-286

[5] Агеев М.С., Добров Б.В., Лукашевич Н.В. Поддержка системы автоматического рубрицирования для сложных задач классификации текстов. // RCDL'2004 Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Шестая всероссийская науч. конф. — Пущино, 2004.

[6] Ageev M.S., Dobrov B.V. Support Vector Machine Parameter Optimization for Text Categorization Problems. // Вестник Национального Технического Университета «ХПИ» — Харьков, Украина, 2004. — №1 — стр. 3-14

[7] Агеев М.С., Добров Б.В., Лукашевич Н.В., Сидоров А.В. Экспериментальные алгоритмы поиска/классификации и сравнение с «basic line». // Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004. — стр. 62-89

[8] Агеев М.С., Добров Б.В., Лукашевич Н.В., Сложные задачи автоматической рубрикации текстов. // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. — Новороссийск, сентябрь 2002.

[9] Агеев М.С., Кураленок И.Е. Официальные метрики РОМИП'2004. // Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004.

[10] Агеев М.С., Добров Б.В., Тематический анализ коллекции документов on-line. // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. — Новороссийск, сентябрь 2003. — стр 249-252.

[11] Ageev M., Dobrov B. Support Vector Machine Parameter Optimization for Text Categorization Problems. // Information Systems Technology and its Applications (ISTA'2003): Proceedings of International Conference / LNI GI, 2003. — Vol 30 — pp. 165-176.

[12] Агеев М.С., Добров Б.В., Журавлев С.В., Лукашевич Н.В., Сидоров А.В., Юдина Т.Н., Технологические аспекты организации доступа к разнородным информационным ресурсам в университетсткой информационной системе РОССИЯ. // Электронные библиотеки, 2002 — Том.5 — Выпуск 2

[13] Агеев М.С., Добров Б.В., Журавлев С.В., Лукашевич Н.В., Макаров-Землянский Н.В., Сидоров А.В., Интеграция разнородных информационных ресурсов в Университетской информационной системе РОССИЯ. // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. — Новороссийск, сентябрь 2002

[14] Агеев М.С., Добров Б.В., Лукашевич Н.В., Сидоров А.В., Штернов С.В. "Отправная точка" для дорожки по поиску в РОМИП (предварительный анализ). // Труды РОМИП'2003 (Российский семинар по Оценке Методов Информационного Поиска) — НИИ Химии СПбГУ / Под ред. И.С.Некрестьянова — Санкт-Петербург, 2003 — стр. 87-110.

[15] Агеев М.С., Журавлев С.В., Ламбурт В.Г. Подготовка Web-версий традиционных изданий. // Открытые Системы, 2000. — №12

[16] Агеев М.С., Журавлев С.В., Захаров В.А. Опыт построения полнотекстовой информационной системы на базе автоматизированной лингвистической обработки текстов с использованием Интернет-технологий Oracle // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. — Новороссийск, сентябрь 1999.

[17] Агеев М.С., Журавлев С.В., Карасев О.И., Ламбурт В.Г. Некоторые вопросы автоматизации подготовки публикаций в Интернет // Научный сервис в сети ИНТЕРНЕТ: Труды Всероссийской науч. конф. — Новороссийск, сентябрь 2000

[18] M. Ageev. Martin's game: a lower bound for the number of sets. // Theoretical Computer Science, 2002. — V. 289/1 — pp.871-876.

Активная библиография

[19] Айзерман М.А., Браверман Э.М., Розоноер Л.И. Метод потенциальных функций в теории обучения машин. — М.: Наука,1970.

[20] Аношкина Ж.Г. Морфологический процессор русского языка. // Бюллетень машинного фонда русского языка / отв. редактор В.М. Андрющенко — М., 1996. — Вып.3, с.53-57.

[21] Антонов А.В., Пример задачи поиска "жизненных историй" — НТИ, Серия 1. — 2003. — № 7 — С.12-17.

[22] Антонов А.В., Козачук М.В., Мешков В.С. Галактика-Зум: Отчет об участии в семинаре РОМИП 2004. // Российский семинар по Оценке

Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004. — стр. 133-141

[23] Бонгард М.М. Проблема узнавания. — М.: Наука, 1967. — 320 с.

[24] Брукинг А. и др. Экспертные системы. Принципы работы и примеры. Пер. с англ.; Под ред. Р.Форсайта. — М.: Радио и связь, 1987.

[25] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах — М: Физматлит, 2004 — 704 стр.

[26] Вайнцвайг М.Н. Алгоритм обучения распознаванию образов "Кора" // Алгоритмы обучения распознаванию образов / Под ред. В.Н. Вапника. — М.: Сов. радио, 1973. — стр. 110-116.

[27] Вапник В.Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

[28] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: Учебник для вузов. — СПб.: Питер, 2000. — 384 с.

[29] Добров Б.В., Лукашевич Н.В., Автоматическая интеллектуальная обработка текстов на основе тезаурусно организованных знаний // Труды шестой национальной конференции по ИИ (КИИ-98). — 1998. — т. II. — с.486-491.

[30] Добров Б.В., Лукашевич Н.В., Автоматическая рубрикация полнотекстовых документов по классификаторам сложной структуры // Восьмая национальная конференция по искусственному интеллекту — Коломна, 2002.

[31] Добров Б.В., Лукашевич Н.В., Использование тематического представления содержания текста для автоматической обработки документов // V Нац. конф. по искусственному интеллекту. — Казань, 1996.

[32] Добров Б.В., Лукашевич Н.В., Тезаурус и автоматическое концептуальное индексирование в университетской информационной

системе РОССИЯ. // Третья Всероссийская конференция по Электронным Библиотекам "Электронные библиотеки: перспективные методы и технологии, электронные коллекции" — Петрозаводск, 2001 — С.78-82.

[33] Дюк В., Самойленко А. Data Mining: учебный курс. — изд-во Питер, 2001.

[34] Журавлев С.В., Юдина Т.Н., Информационная система РОССИЯ // НТИ. Сер.2. — 1995. — № 3. — С.18-20.

[35] Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики, 1978, — вып. 33, — с. 5-68.

[36] Загоруйко Н.Г. Прикладные методы анализа данных и знаний — Новосибирск: Изд-во Ин-та математики, 1999. — 270 с.

[37] Загорулько Ю.А., Кононенко И.С., Костов Ю.В., Сидорова Е.В. Классификация деловых писам в системе документооборота // Международная конференция ИСТ'2003 "Информационные системы и технологии" — Новосибирск, 2003,

[38] Искусственный интеллект. Справочник в трех томах. / под ред. Захарова В.Н., Попова Э.В., Поспелова Д.А., Хорошевского В.Ф. — М.: Радио и связь, 1990. — Т.2

[39] Лукашевич Н.В., Автоматическое рубрицирование потоков текстов по общественно-политической тематике // НТИ. Сер.2., 1996. — № 10. — С.22_30.

[40] Маковский А.Л., Новиков Д.Б., Силкина А.В., Симбирцев А.Н., Принципы построения системы классификации правовых актов // Правовой классификатор и правовой тезаурус с законотворчестве и юридической практике / Сост. В.Б.Исаков и др. — М., ГД РФ: Изд-во Гуманитарного университета, 1998. — с.5-28.

[41] Мегапьютер Интеллидженс: Реферирование и классификация текстов (информация на web-сайте компании) http://www.megaputer.ru/doc.php?detail/040923_detail.html

[42] И. Кураленок, И. Некрестьянов. Оценка систем текстового поиска. / Программирование. — 28(4), 2002 — стр. 226-242

[43] Некрестьянов И.С. Тематико-ориентированные методы информационного поиска: Дис. канд. физ-мат. наук: 05.13.11 / С-Пб. гос. унив. — Санкт-Петербург, 2000.

[44] О классификаторе правовых актов: Указ Президента РФ №511 от 15 марта 2000г.

[45] Объедков С. А. Алгоритмические аспекты ДСМ-метода автоматического порождения гипотез. / НТИ, Серия 2. — Выпуск 1-2, 1999 — стр. 64-74.

[46] Осипова Н. Анализ результатов тестирования алгоритма София при решении задачи классификации коллекции правовых документов. // Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004. — стр. 110-118

[47] Плешко В.В., Ермаков А.Е., Голенков В.П. RCO на РОМИП 2004. // Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004. — стр. 43-61

[48] Поспелов Д.А. Становление информатики в России. / В кн. "Очерки истории информатики в России". — Редакторы-составители Д. А. Поспелов и Я. И. Фет. — Новосибирск: Научно-издательский центр ИГГМ СО РАН, 1998

[49] Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах — М.: Наука, 1989. — 189 с.

[50] Рыбинкин В.В. Система рубрикации данных "Синдбад". // Российский семинар по Оценке Методов Информационного Поиска (РОМИП 2004) — Пущино, 2004. — стр. 90-99

[51] Труды РОМИП'2003 — НИИ Химии СПбГУ / Под ред. И.С.Некрестьянова — Санкт-Петербург, 2003 — 132 с.

[52] Хант Э. Искусственный интеллект. — М.: Мир. 1978. — Часть 2. Распознавание образов.

[53] Чесноков С.В. Детерминационный анализ социально-экономических данных. — М:, "Наука", 1982.

[54] Beuster G. MIC — A System for Classification of Structured and Unstructured Texts. Diploma Thesis. — University Koblenz, 2001.

[55] Burges C.J.C. A tutorial on support vector machines for pattern recognition. // Data Mining and Knowledge Discovery, — 2(2):955-974, 1998.

[56] Callan J.P., Croft W.B. and Harding S.M. The INQUERY Retrieval System // Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications. / A.M. Tjoa and I. Ramos (eds.), Database and Expert System Applications. — Springer Verlag, New York, 1992. — pp.78-93.

[57] Debole F., Sebastiani F., An Analysis of the Relative Hardness of Reuters-21578 Subsets // Journal of the American Society for Information Science and Technology, 2004

[58] Dumais S., Platt J,, Heckerman D., Sahami M. Inductive learning algorithms and representations for text categorization. // In Proc. Int. Conf. on Inform. and Knowledge Manage., 1998.

[59] Dumais S., Lewis D., Sebastiani F. Report on the Workshop on Operational Text Classification Systems (OTC-02) // SIGIR-2002 — Tampere, Finland, 2002

[60] Hayes P.J., Weinstein S.P. Construe: A System for Content-Based Indexing of a Database of News Stories // Proceedings of the Second Annual Conference on Innovative Applications of Intelligence, 1990.

[61] Hayes P. Intelligent High-Volume Text Processing Using Shallow, Domain-Specific Techniques. / In P. Jacobs (Ed.) Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval. — Lawrence Erlbaum, Hillsdale, NJ, 1992. — pp 227--241.

[62] Haykin, S. Neural Networks: A Comprehensive Foundation. — New York: Macmillan College Publishing, 1994

[63] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. // Proceedings of ECML-98, 10th European Conference on Machine Learning — 1998.

[64] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. // Proceedings of ICML-97, 14th International Conference on Machine Learning. — 1996.

[65] Joachims T. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods / Support Vector Learning, Schölkopf B., Burges C., Smola A. (ed.), — MIT-Press, 1999.

[66] Joachims T. Estimating the Generalization Performance of a SVM Efficiently. // Proceedings of the International Conference on Machine Learning, — Morgan Kaufman, 2000.

[67] Legislative Indexing Vocabulary — Congressional Research Service. The Library of Congress. Twenty-first Edition, 1994. — 546 p.

[68] Lewis D. Applying Support Vector Machines to the TREC-2001 Batch Filtering and Routing Tasks. Proceedings of TREC-2001 conference.

[69] Lewis D. Feature Selection and Feature Extraction for Text Categorization. // Proceedings of the DARPA Workshop on Speech and Natural Language. — Harriman, New York, 1992. — pp. 212-217

[70] Lewis D. Reuters-21578 text categorization test collection. Distribution 1.0 http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt

[71] Lewis D., Sebastiani F. Report on the Workshop on Operational Text Classification Systems (OTC-01) // SIGIR-2001 — New Orleans, 2001

[72] Marshall R.J. Generation of Boolean classification rules. // Proceedings of Computational Statistics 2000 — Utrecht, The Netherlands, / eds Bethlehem and PGM van der Heijden, — Springer-Verlag, Heidelberg, 2000 — pp. 355-360.

[73] Quinlan J.R. C4.5 Programs for machine learning. — Morgan Kaufmann, — San Mateo, Californie, 1993.

[74] van Rijsbergen C.J. Information Retrieval. — Butterworth's and Co. — London, 1979 — 2nd edition.

[75] Salton G, Buckley C. Term-Weighting Approaches in Automatic Text Retrieval. / Information Processing and Management, —1988 — pp. 513-523.

[76] Teoma: Adding a New Dimension to Search: The Teoma Difference is Authority http://sp.teoma.com/docs/teoma/about/searchwithauthority.html

[77] The Twelfth Text Retrieval Conference (TREC 2003). Appendix 1. Common Evaluation Measures. http://trec.nist.gov/pubs/trec12/appendices/measures.ps

[78] Vapnik V. The Nature of Statistical Learning Theory. — Springer-Verlag — New York, 1995.

[79] Yang Y. An Evaluation of Statistical Approaches to Text Categorization. / Journal of Information Retrieval, 1999 — V.1 — pp. 67--88.

[80] Yang Y., Liu X. A re-examination of text categorization methods. // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999 — pp. 42-49.

[81] Yang Y., Pedersen J. A comparative study on feature selection in text categorization. // In: Proc. of ICML-97, 14th International Conf. On machine Learning — Nashville, USA, 1997. — pp. 412-420.

[82] Wasson M. Classification Technology at LexisNexis. // SIGIR 2001 Workshop on Operational Text Classification.