Экспериментальные алгоритмы поиска/классификации и сравнение с «basic line»*

© М.С. Агеев[1, 2, 3], Б.В. Добров[1, 3],

Н.В.Лукашевич[1, 3], А.В. Сидоров[3]

[1] Научно-исследовательский вычислительный центр
МГУ им. М.В.Ломоносова
[2] Механико-математический факультет
МГУ им. М.В.Ломоносова
[3] АНО Центр информационных исследований
{ageev, dobroff, louk, alexeys}@mail.cir.ru

Аннотация

В статье описываются подходы, использованные коллективом разработчиков Университетской информационной системы РОССИЯ (УИС РОССИЯ, http://www.cir.ru), для выполнения заданий РОМИП 2004 по поиску в web коллекции, поиску в коллекции правовых документов и классификации правовых документов. Прежде всего мы были заинтересованы в получении для каждой дорожки «отправной точки» (basic line), для чего использовали широко известные методы. Также мы осуществили несколько экспериментов, результаты которых сравниваем с результатами «классических» методов.

1. Введение

В 2003 году коллектив разработчиков информационно-поисковой машины и ряда лингвистических технологий, используемых в Университетской информационной системе РОССИЯ (далее – УИС

* Авторы выражают признательность Российскому фонду фундаментальных исследований за поддержку данной работы (грант № 04-07-90280-в).

РОССИЯ, http://www.cir.ru [11]) принимал участие в дорожке по поиску в Web-коллекции. Основной целью являлось получение «basic line» (отправной точки) для дорожки по поиску, если использовать классическую схему поиска TF*IDF. Кроме того было выполнено еще два экспериментальных прогона.

В 2004 году состав дорожек, в которых мы приняли участие, расширился:

- дорожка по поиску в web-коллекции;
- дорожка по поиску в коллекции правовых документов;
- дорожка по классификации правовых документов.

При этом, как и ранее, прежде всего мы были заинтересованы в получении для каждой дорожки «отправной точки» (basic line), для чего использовали широко известные методы. Наряду с получением оценки «отправной точки» мы осуществили несколько экспериментов, направленных на получение результатов лучше чем у «классических» методов.

2. Дорожка поиска по web-коллекции

Перед участниками поставлена задача классического ad hoc поиска документов по запросу пользователя. Для анализа было предложено более 700 тысяч документов (согласно нашим данным 727995) сайта www.narod.ru.

Оргкомитет представил список из более чем 24 тысяч запросов (24259 штук), для каждого из которых необходимо было выполнить поиск и выдать не более 100 документов. Запросы были получены из лога поисковой системы Яндекс (www.yandex.ru).

Коллекция документов для этой дорожки совпадает с коллекцией документов РОМИП'2003. Набор поисковых запросов 2003 года расширен новыми запросами.

2.1. Описание алгоритмов, использованных для выполнения заданий

Для дорожки поиска документов по web-коллекции мы подготовили два прогона. Первый прогон — классический алгоритм поиска по словам с ранжированием документов по формуле TF*IDF [3, 12]. В РОМИП’2003 один из прогонов мы также выполняли с использованием этого алгоритма и в этом году решили повторить этот прогон.

Второй прогон — экспериментальный. Мы исследовали влияние учёта расстояния между словами на качество поиска. Кроме того, учитывалось попадание слов запроса в заголовок документа.

По сравнению с прошлым годом были учтены некоторые ошибки, возникшие при загрузке документов. Был разработан специальный модуль, который автоматически определял кодировку документа и перекодировал все документы в windows-1251. В результате были обнаружены и перекодированы 6131 документ (0.8%) в кодировке КОИ-8 и 195 документов в кодировке dos.

2.2. Прогон 1: «Отправная точка»

При разборе строки запроса производилась фильтрация специальных символов, наличие которых приводило к неправильной интерпретации запроса (из-за различий в формате языка запросов в Яндекс и УИС РОССИЯ).

Каждый запрос представлялся в виде последовательность словоформ, соединяемых условием «AND», для каждой словоформы вырабатывались леммы, соединяемые условием «OR».

Для получения базовой оценки использовалась модель TF*IDF в формулировке INQUERY [3].

Точнее - вес каждой леммы документа оценивался следующим образом:

$$\mathrm{TFIDF}_D(l) = \beta + (1-\beta)\cdot tf_D(l)\cdot idf_D(l) \quad ,$$

где “term frequency” – учет частотности леммы в документе:

$$tf_D(l) = \frac{\mathrm{freq}_D(l)}{\mathrm{freq}_D(l) + 0.5 + 1.5\cdot\dfrac{dl_D}{\mathrm{avg\_dl}}} \quad ,$$

$freq_D(l)$ - частотность леммы l в документе, dl_D – мера длины документа, avg_dl – средняя длина документа, $\beta = 0.4$.

“Inverse term frequency” - фактически форма штрафования часто используемых в коллекции слов:

$$idf(l) = \frac{\log\left(\dfrac{|c|+0.5}{df(l)}\right)}{\log(|c|+1)} \quad ,$$

где $|c|$ - количество документов в коллекции, $df(l)$ - количество документов, где встретилось лемма l.

Известно [8], что можно применять различные модификации данной формулы - все дают примерно одинаковый результат.

Каждый запрос $Q = w_1 \;\; w_2 \;\; w_3 \ldots \; w_m$

представлялся в виде формулы

$$L(Q)= L(w_1) \,\&\, L(w_2) \,\&\, L(w_3) \,\&\, \ldots \,\&\, L(w_m)\,,$$

где $L(w) = l_1(w)\ OR\ l_2(w)\ OR\ \ldots OR\ l_q(w)$, $l_k(*)$ - леммы морфологического разбора слова.

Тогда оценка релевантности документа D для запроса Q вычисляется по формуле:

$$\mathrm{V}_D(Q) \;=\; \frac{\sum_{i=1}^{N}\sum_{k}\left(\theta_{ik}\cdot \mathrm{TFIDF}_D(l_{ik}(\mathrm{w}_i))\right)}{\sum_{i=1}^{N}\sum_{k}\left|\theta_{ik}\right|} \qquad (1.1)$$

где $\theta_{ik} = \theta_i = 1.0$ — "вес" леммы в запросе — равен весу, устанавливаемому для соответствующего слова запроса.

Основная работа по исполнению запроса производится СУБД Oracle. Преобразование запроса пользователя в SQL осуществляется средствами программ на языках Java и PL/SQL. Более подробно алгоритм «отправной точки» описан в [12].

2.3. Прогон 2: Влияние расстояния между словами

Второй прогон был предназначен для исследования влияния учета расстояния между словами запроса в документе на качество поиска. Кроме того, учитывалось попадание слов запроса в заголовок документа.

Во втором прогоне исполнялся запрос по словам с учетом морфологического словоизменения как в прогоне 1, а затем вычислялся ранг соответствия документа запросу с учетом расстояния между словами и содержания заголовков документов.

Ранг соответствия документа запросу вычислялся как среднее между TF*IDF-рангом и рангом по расстоянию:

$$\mathrm{Rank}_D\left(Q\right) = \frac{\mathrm{V}_D\left(Q\right) + \mathrm{Near}_D\left(Q\right)}{2} \qquad (1.2)$$

где $\mathrm{Rank}_D\left(Q\right)$ — ранг соответствия документа запросу; $\mathrm{V}_D\left(Q\right)$ — ранг, вычисленный по формуле TF*IDF (1.1). Ранг по близости $\mathrm{Near}_D\left(Q\right)$ вычисляется следующим образом:

1. Если запрос встречается в заголовке документа (внутри тегов title и h1) в виде подстроки, то $\mathrm{Near}_D\left(Q\right) = 2$.
2. Иначе, если запрос встречается внутри документа в виде подстроки, то $\mathrm{Near}_D\left(Q\right) = 1$.

3. Иначе, производится поиск минимального «куска» документа, в котором содержатся все слова запроса. Пусть длина этого куска документа в словах равна $\lambda_D(Q)$. Тогда

$$\mathrm{Near}_D(Q) = \frac{1}{\ln\left(\lambda_D(Q) - |Q| + 4\right)} \qquad (1.3)$$

где $|Q|$ — количество слов в запросе.

2.4. Описание результатов

Оценка результатов прогонов производилась на основе сравнения результатов систем с мнением экспертов РОМИП с использованием общепринятых метрик. При этом было получено множество таблиц релевантности для различных наборов запросов, различных способов сведения в единую таблицу мнений разных экспертов и различных условий оценки. Опишем вкратце, как формировались таблицы оценки, которые были разосланы участникам. В скобках указана часть имени файла результатов, соответствующая данному способу оценки.

1) Использовались три подмножества запросов для оценки:
   a) 48 запросов, выбранных случайно после отправки участниками результатов прогонов (нет суффикса);
   b) 19 запросов, которые использовались при оценке в 2003 году (2003);
   c) объединение пп. a и b — 67 запросов (_all).
2) Результаты оценивались для двух «срезов» выдачи:
   a) учитывались все (не более 100) документы, выданные системой по запросу (summary.txt);
   b) учитывались только документы, попавшие в пул оценки — первые 50 документов, выданных системой (summary_pd50.txt).
3) Использовались два разных способа «слияния» оценок экспертов:
   a) документ считается релевантным, если хотя бы один эксперт оценил его как relevant-minus или выше (output_or_relevant-minus);
   b) документ считается релевантным, если все эксперты оценили его как relevant-minus или выше (output_and_relevant-minus).
4) Эксперты оценивали документы в различных условиях:

a) эксперты использовали расширенные описания запросов, составленные централизованно (with_desc);
b) эксперты не использовали расширенные описания запросов. Каждый эксперт самостоятельно принимал решение об интерпретации запроса (no_desc).

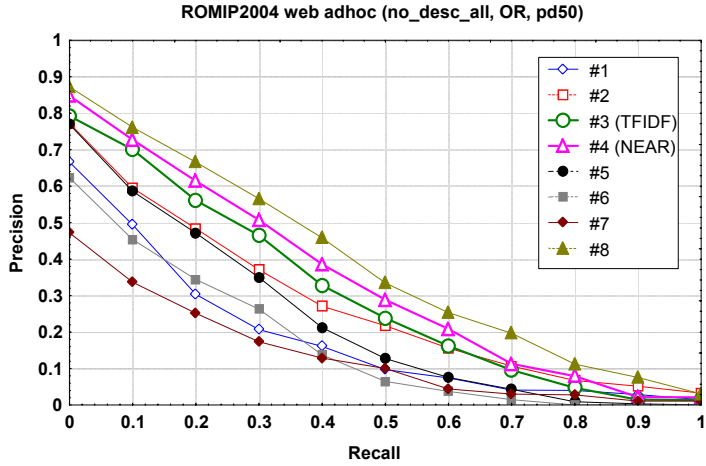

Рис. 1. 11-точечный график полноты/точности для различных прогонов дорожки web adhoc (без описания).

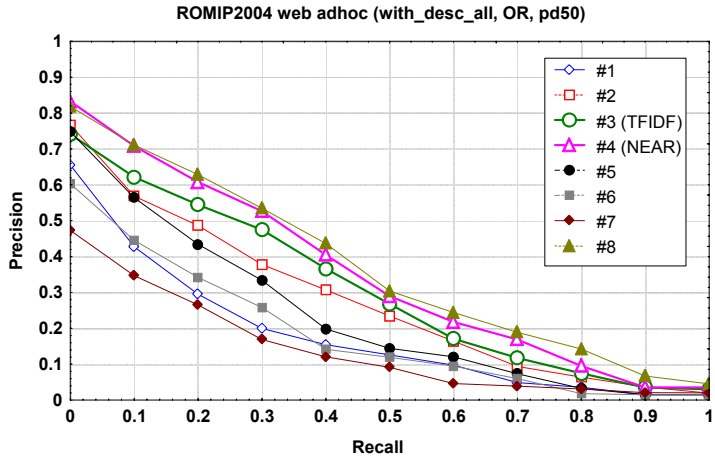

Рис. 2. Рис. 1. 11-точечный график полноты/точности для различных прогонов дорожки web adhoc (с описанием).

Итого, было составлено и разослано участникам 3*2*2*2=24 таблицы релевантности, для каждой из которых были вычислены метрики качества поиска (6 метрик + 11 точек TREC).

Чтобы провести анализ результатов мы выбрали две таблицы релевантности для детального исследования, а затем проанализировали степень влияния используемой таблицы релевантности на результаты. Мы выбрали таблицы релевантности, соответствующую пунктам 1c-2b-3a-4b и 1c-2b-3a-4a (см. список выше).

Результаты различных прогонов показаны на Рис.1-4. Наши прогоны обозначены «TFIDF» и «NEAR» (прогон 1 и прогон 2 соответственно).

Из приведённых графиков можно сделать следующие выводы:

1. Прогон 1, использующий «классический» метод поиска и ранжирования TF*IDF показывает весьма высокие результаты.
2. Учёт расстояния между словами и наличия слов запроса в заголовках повышает качество поиска (однако прогон «near» проигрывает одному из алгоритмов других участников). К сожалению, из нашего эксперимента трудно сделать вывод, насколько сильно на результаты влияет фактор учёта заголовков.

2.5. Устойчивость результатов

На рис. 5-6 показаны графики результатов систем для различных таблиц релевантности. По оси абсцисс находятся номера таблиц релевантности в следующем порядке (указаны имена файлов с результатами):

OR		
1	no_desc	summary
2	no_desc	summary_pd50
3	no_desc_all	summary
4	no_desc_all	summary_pd50
5	no_desc2003	summary
6	no_desc2003	summary_pd50
7	with_desc	summary
8	with_desc	summary_pd50
9	with_desc_all	summary
10	with_desc_all	summary_pd50
11	with_desc2003	summary
12	with_desc2003	summary_pd50

AND		
13	no_desc	summary
14	no_desc	summary_pd50
15	no_desc_all	summary
16	no_desc_all	summary_pd50
17	no_desc2003	summary
18	no_desc2003	summary_pd50
19	with_desc	summary
20	with_desc	summary_pd50
21	with_desc_all	summary
22	with_desc_all	summary_pd50
23	with_desc2003	summary
24	with_desc2003	summary_pd50

Исследуем устойчивость полученных результатов относительно различных таблиц релевантности.

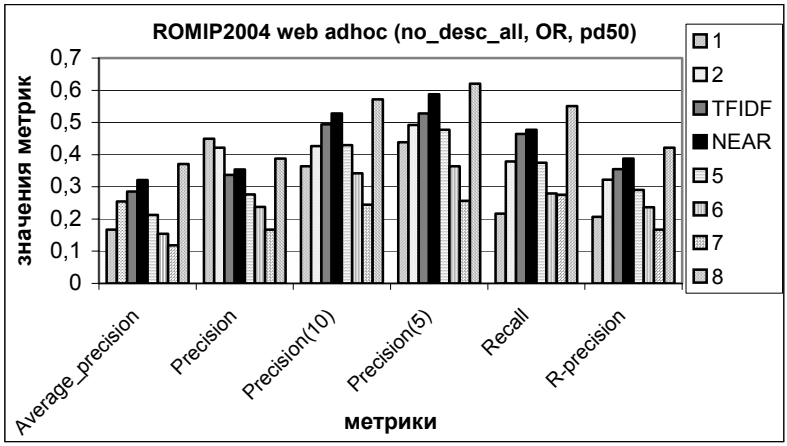

Рис. 3 Метрики качества поиска для различных прогонов (без описания).

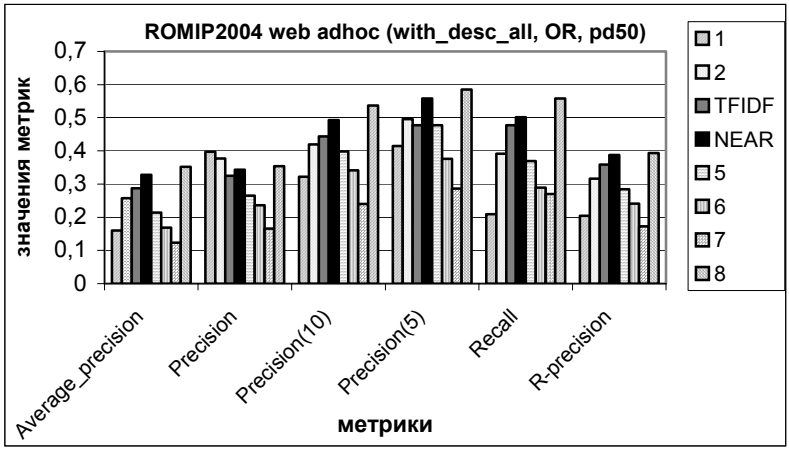

Рис. 4 Метрики качества поиска для различных прогонов (с описанием).

По оси ординат указано значение метрики Average Precision для рис. 5 и Precision(10) для рис 6.

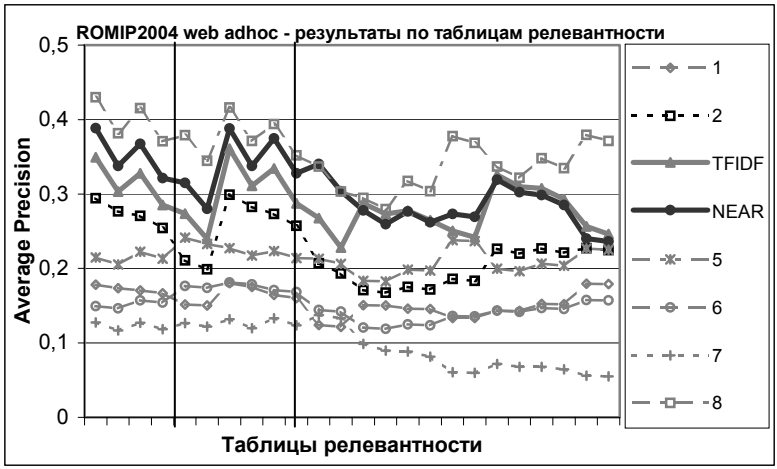

Рис. 5 График зависимости Average Precision от таблицы релевантности. Вертикальными линиями отмечены таблицы релевантности, выбранные нами для анализа.

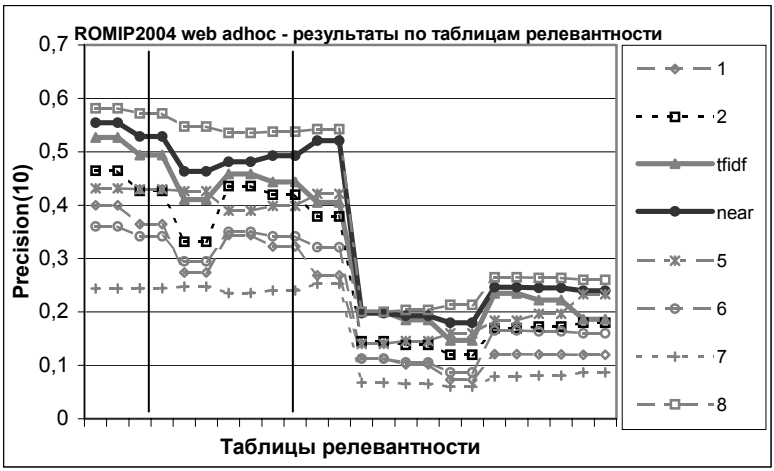

Рис. 6 График зависимости Precision(10) от таблицы релевантности. Вертикальными линиями отмечены таблицы релевантности, выбранные нами для анализа.

Основной вывод, который мы делаем из анализа рис. 5-6 состоит в том, что

1. «Места» различных прогонов меняются в основном в пределах одной позиции в зависимости от таблицы релевантности. Поэтому результаты анализа, проведенного нами на двух выделенных таблицах релевантности, можно обобщить на все таблицы.
2. Точность системы на первых 10 документах существенно зависит от способа объединения оценок разных экспертов. Для матриц «or» (левая половина рис. 6) Precision(10) существенно выше, чем для матриц «and» (правая таблица релевантности.

3. Дорожка поиска по коллекции нормативных документов

Перед участниками поставлена задача классического ad hoc поиска документов по запросу пользователя. Для анализа было предложено около 60 тысяч документов (согласно нашим данным 60015) из коллекции нормативных документов РФ компании «Кодекс».

Оргкомитет представил список из 12925 запросов, для каждого из которых необходимо было выполнить поиск и выдать не более 100 документов.

Поскольку задание и способ его оценки для этой дорожки аналогичны дорожке поиска по web-коллекции, мы использовали аналогичные методы выполнения заданий (за исключением прогона 2) и анализа результатов.

3.1. Описание алгоритмов, использованных для выполнения заданий

Для дорожки поиска документов по коллекции нормативных документов мы подготовили два прогона. Алгоритм первого прогона для этой дорожки полностью совпадает с алгоритмом первого прогона «отправная точка» для дорожки поиска по web-коллекции (см. раздел 2.2). Во втором прогоне мы попробовали повысить качество поиска путём повышения веса слов запроса, встретившихся в заголовках документов. Опишем подробнее алгоритм второго прогона.

3.2. Прогон 2: Повышения веса слов запроса, встретившихся в заголовках

Во втором прогоне исполнялся запрос по словам с учетом морфологического словоизменения как в прогоне 1, а затем вычислялся ранг соответствия документа запросу с учетом содержания заголовков документов.

Ранг соответствия документа запросу вычислялся как среднее между TF*IDF-рангом и рангом по заголовкам:

$$\mathrm{Rank}_D(Q) = \frac{\mathrm{V}_D(Q) + \mathrm{HdrFreq}_D(Q)}{2} \tag{1.4}$$

где $\mathrm{Rank}_D(Q)$ — ранг соответствия документа запросу; $\mathrm{V}_D(Q)$ — ранг, вычисленный по формуле TF*IDF (1.1). Ранг по заголовкам $\mathrm{HdrFreq}_D(Q)$ равен отношению количества слов запроса, встретившихся в заголовке, к длине запроса:

$$\mathrm{HdrFreq}_D(Q) = \frac{\left|\mathrm{HdrWords} \cap Q\right|}{\left|Q\right|} \tag{1.5}$$

где HdrWords — множество слов в заголовке документа, $\left|Q\right|$ — количество слов в запросе.

3.3. Описание результатов

Анализ результатов для дорожки поиска по нормативным документам проведём аналогично анализу результатов дорожки web-поиска.

Участникам были разосланы результаты, вычисленные для четырёх таблиц релевантности. Опишем как формировались эти таблицы релевантности. В скобках указана часть краткого названия таблицы релевантности, которое мы будем использовать при построении графиков.

1) Результаты оценивались для двух «срезов» выдачи:
   a. учитывались все (не более 100) документы, выданные системой по запросу (pd100);
   b. учитывались только документы, попавшие в пул оценки — первые 50 документов, выданных системой (pd50).
2) Использовались два разных способа «слияния» оценок экспертов:
   a. документ считается релевантным, если хотя бы один эксперт оценил его как relevant-minus или выше (or);

b. документ считается релевантным, если все эксперты оценили его как relevant-minus или выше (and).

Для анализа мы выбрали таблицу релевантности “or_pd100”, соответствующую пунктам 1a-2a .

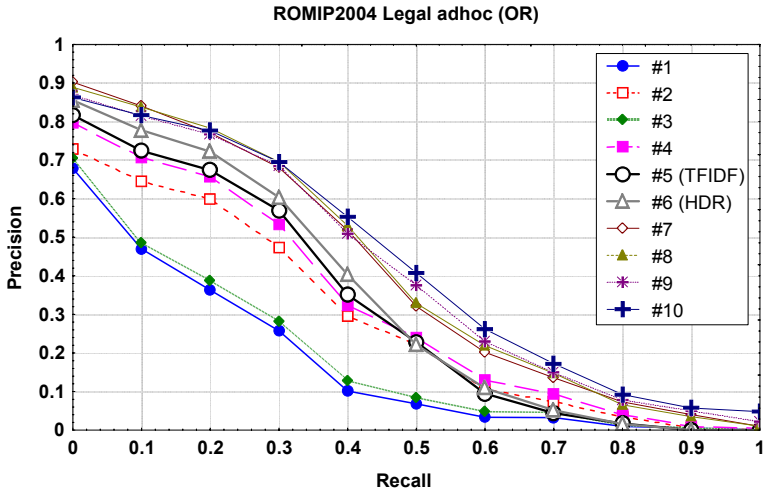

Рис. 7. 11-точечный график полноты/точности для различных прогонов. Наши прогоны — «TFIDF» и «HDR».

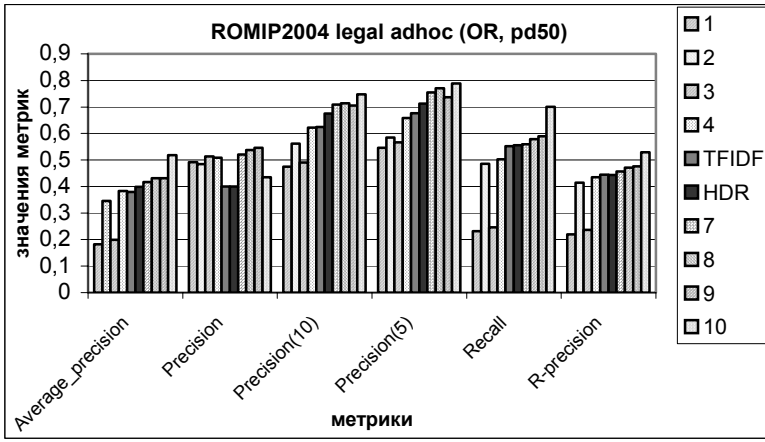

Рис. 8 Метрики качества поиска для различных прогонов. Наши прогоны — «TFIDF» и «HDR».

Результаты различных прогонов показаны на рис. 7 и 8. Наши прогоны обозначены «TFIDF» и «HDR» (прогон 1 и прогон 2 соответственно). Из приведённых графиков можно сделать следующие выводы:

1) Прогон 1, использующий «классический» метод поиска и ранжирования TF*IDF показывает «средние» результаты — обгоняет несколько алгоритмов и проигрывает нескольким.
2) Учёт заголовков повышает качество поиска, но не намного.

4. Дорожка по тематической классификации коллекции нормативных документов

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.

4.1. Описание алгоритмов, использованных для выполнения заданий

Для дорожки тематической классификации документов мы подготовили три прогона. Два прогона основаны на алгоритме машинного обучения SVM (Support Vector Machines) — широко известном алгоритме, который хорошо себя зарекомендовал в задачах классификации текстов [4, 6, 9, 10]. Третий прогон основан на алгоритме машинного обучения нашей разработки, строящем описания рубрик в виде булевской формулы.

Целью первого прогона было получение «отправной точки» (basic line) 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 ЦИИ (подробности ниже).

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

4.2. Прогон 1: SVM по леммам

Мы выбрали SVM среди других методов машинного обучения в связи с тем, что в нескольких независимых исследованиях SVM показал преимущество перед другими методами машинного обучения на задаче классификации текстов [4, 6, 10]. Отметим, однако, что эти исследования проводились на одной и той же коллекции документов — это коллекция финансовых сообщений информационного агентства Рейтер [7].

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 [2, 9].

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.

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.

В качестве программной реализации SVM использовалась свободно доступная SVM_light v. 3.50 [5].

4.2.1. Оптимизация параметров SVM

В 2002 году нами были проведены исследования по улучшению результатов работы SVM, которые показали, что для практического применения необходимо оптимизировать стандартную методику применения SVM. В статье [1] 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:

$$\tilde{j} \in \left[1, \max\left(1.5\frac{\text{neg\_ex}}{\text{pos\_ex}}, 1\right)\right] \qquad (1.6)$$

В результате получено заметное улучшение качества классификации по сравнению с другими методами оптимизации параметров SVM.

Для дорожки тематической классификации РОМИП'2004 мы использовали указанный метод оптимизации параметра j следующим образом:

Коллекция документов для обучения (4496 документов) разбивалась на две подколлекции — для обучения (70% документов) и для оценки качества результатов (30% документов).

Осуществлялся перебор из 10 значений параметра j в интервале (1.6). 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.

SVM с параметром $j = \tilde{j}$ обучалась на всей коллекции обучения (4496 документов) и применялась к документам, которые необходимо отрубрицировать (55519 документов).

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

4.2.2. Векторное представление документов

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 [3, 12]. 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.

4.3. Прогон 2: SVM по леммам+терминам

Для прогона 2 применялся описанный выше 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 терминах Тезауруса ЦИИ. Терминологический индекс документа строится на этапе предварительной обработки программой Автоматической Лингвистической Обработки Текстов (АЛОТ) [14].

Для каждого понятия в документе на этапе предварительной обработки документа в УИС РОССИЯ 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 [14].

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

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

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

В прогоне 3 использовался разработанный нами метод машинного обучения, который строит описание рубрики в виде булевской формулы — запроса к полнотекстовой информационной системе. Элементами формул являются понятия Тезауруса ЦИИ. Алгоритм строит формулы вида

$$U = \bigcup_{i=1}^{k} \bigcap_{j=1}^{J_i} t_{i,j} \qquad (1.7)$$

где $t_{i,j}$ — множества документов, содержащих некоторый термин тезауруса. Конъюнкции, составляющие формулу, имеют длину J_i от 1 до 3. Подробное описание этого алгоритма опубликовано в [13].

Мотивацией для разработки такого алгоритма 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.

В качестве основы для моделирования мы используем подход к описанию рубрики, используемый в УИС РОССИЯ [15, 16]. Согласно этому подходу, описание рубрики экспертом представляется в виде булевской формулы вида

$$\mathrm{U} = \bigcup_i \bigcap_j \left(\bigcup_k t_{i,j,k} \setminus \bigcup_l t'_{i,j,l} \right) \tag{1.8}$$

где U — множество документов, принадлежащих рубрике, а $t_{i,j,k}$ и $t'_{i,j,l}$ — множество документов, содержащих некоторый термин тезауруса. Выбор структуры формулы и понятий, включаемых в формулу, производится экспертом на основе знаний предметной области и, возможно, частичного анализа коллекции документов. Типичные цифры о параметрах описания: на одну рубрику рубрикатора в среднем приходится 1-2 дизъюнкта, 2-3 конъюнкта, 10-20 опорных понятий («положительных» и «отрицательных»).

Задача моделирования логики рубрикатора при помощи машинного обучения, в нашем случае, состоит в построении формул вида (1.8) на основе анализа множества отрубрицированных документов. Разработанный нами алгоритм машинного обучения строит формулы описания рубрики в виде (1.7), несколько отличающемся от (1.8), но так же соответствующем логике построения рубрикатора.

4.5. Описание результатов

4.5.1. Методика оценки

Оценка результатов участников в дорожке тематической классификации нормативных документов производилась с использованием большого разнообразия таблиц релевантности (4 варианта) и метрик качества рубрицирования (8 метрик). Результаты работы систем вычислялись на двух различных подмножествах

рубрик, назовём их А (12 рубрик) и Б (40 рубрик). Использовались следующие оценки соответствия рубрики документу (таблицы релевантности):

1) “ideal” — оценки, проставленные экспертами ИС «Кодекс» для рубрик из А;
2) “ideal40” — оценки, проставленные экспертами ИС «Кодекс» для рубрик из Б;
3) “and_relevant-minus” — оценки, проставленные экспертами РОМИП для рубрик из А; документ считается соответствующем рубрике, если хотя бы один эксперт оценил его как relevant-minus или выше;
4) “or_relevant-minus” — оценки, проставленные экспертами РОМИП для рубрик из А; документ считается соответствующем рубрике, если все эксперты оценили его как relevant-minus или выше.

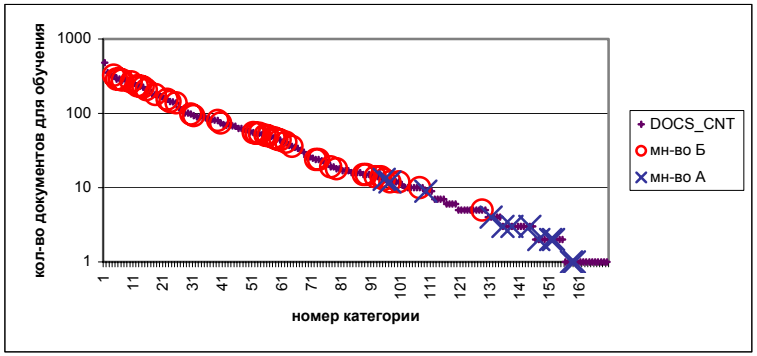

Рис. 9 Зависимость количества документов для обучения от номера рубрики в логарифмической шкале (рубрики упорядочены по убыванию частотности).

К сожалению необходимо отметить, что набор рубрик, попавших во множество А, отличается низкой частотностью документов в коллекции для обучения. Среднее значение частотности документов в коллекции для обучения для рубрик из А равно 4,6, в то время как для всех рубрик это значение равно 56,7.

На рис. 9 представлена зависимость количества документов для обучения от номера рубрики в логарифмической шкале (рубрики упорядочены по убыванию частотности). Рубрики, попавшие в выборки А и Б выделены знаками: А — треугольниками, Б — кругами.

Из рис. 1 видно, что множество А состоит из малочастотных рубрик, а множество Б равномерно распределено среди рубрик с частотностью более 10.

Для методов машинного обучения количество примеров обучения играет очень важную роль. В связи с этим, множество рубрик А нельзя считать представительным для всего множества рубрик, а результаты по таблицам релевантности “and_relevant-minus” и “or_relevant-minus” не отражают поведения систем, основанных на машинном обучении, в среднем по всему рубрикатору.

На наш взгляд, наиболее представительной является выборка, состоящая из объединения множеств А и Б, состоящая из 50 рубрик.

Таблицу релевантности, состоящую из оценок, проставленных экспертами ИС «Кодекс» для рубрик из $А \bigcup Б$ будем обозначать “ideal50”.

4.5.2. Таблицы результатов

На рис. 10 представлены результаты прогонов участников для таблицы релевантности “ideal50”. Наши прогоны обозначены “svm_lem”(п.4.2), “svm_thes”(п.4.3) и “formul”(п.4.4) 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.

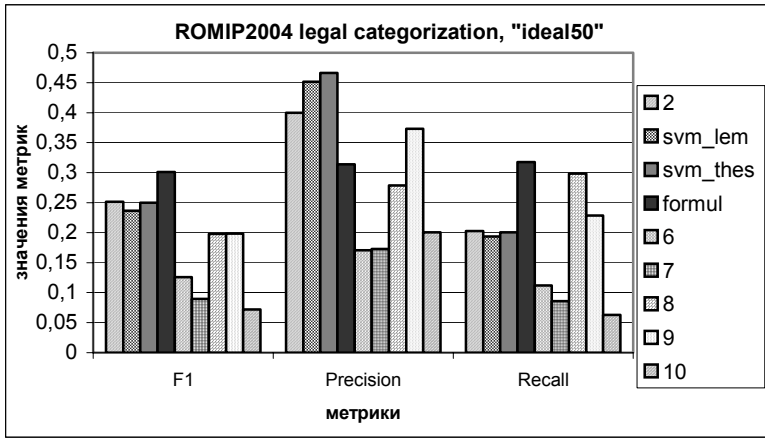

Рис. 10 Результаты прогонов участников для таблицы релевантности “ideal50”.

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. В качестве таблицы релевантности использовалась “ideal50”. Результаты показаны на рис. 11.

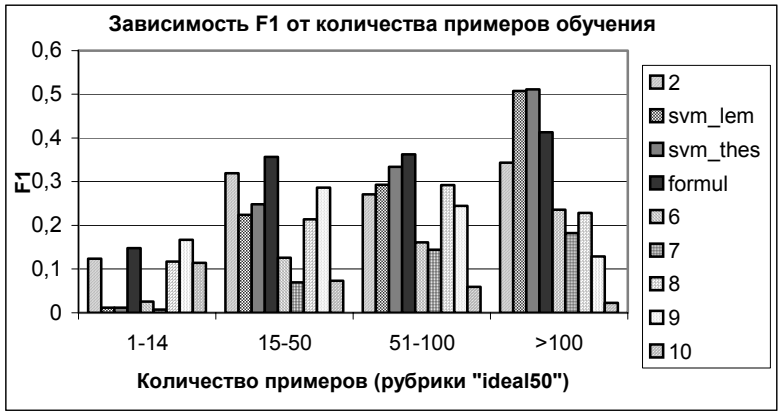

Рис. 11 Зависимость F-меры от количества примеров для обучения (в среднем для рубрик, частотность которых попадает в указанный интервал)

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).

На рисунках 12-15 отображены результаты для таблиц релевантности “ideal”, “ideal40”, “or_relevant-minus” и “and_relevant-minus”.

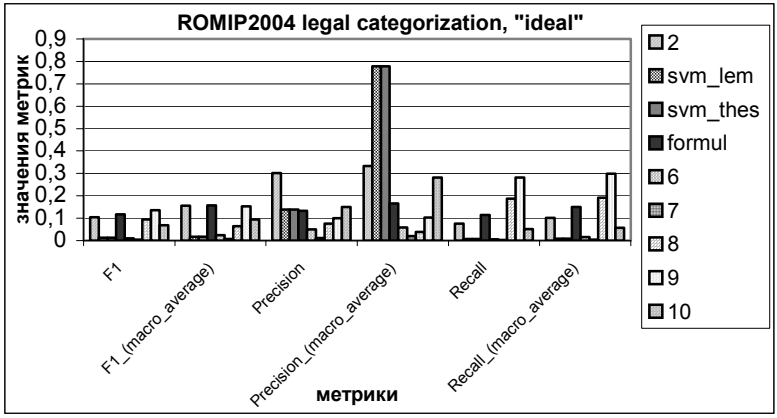

Рис. 12 Результаты прогонов участников для таблицы релевантности “ideal”

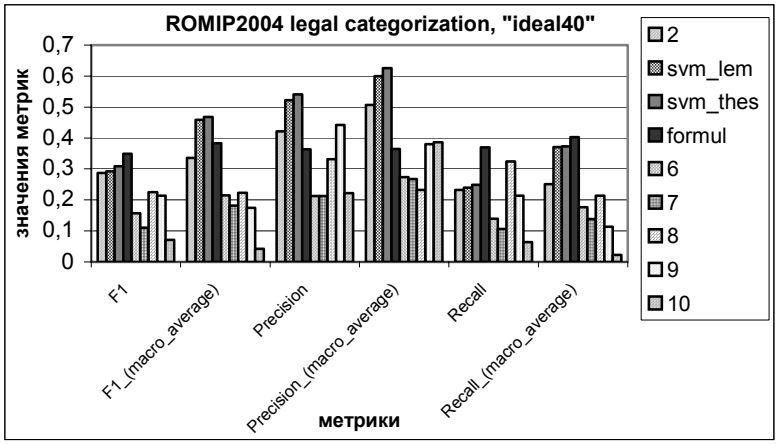

Рис. 13 Результаты прогонов участников для таблицы релевантности “ideal40”

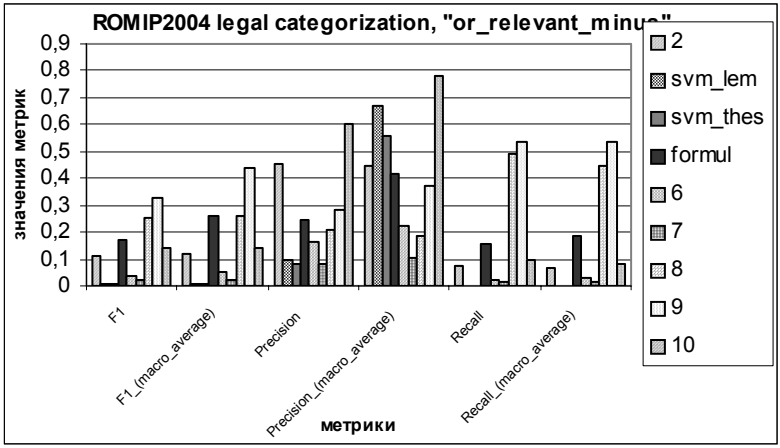

Рис. 14 Результаты прогонов участников для таблицы релевантности “or_relevant-minus”

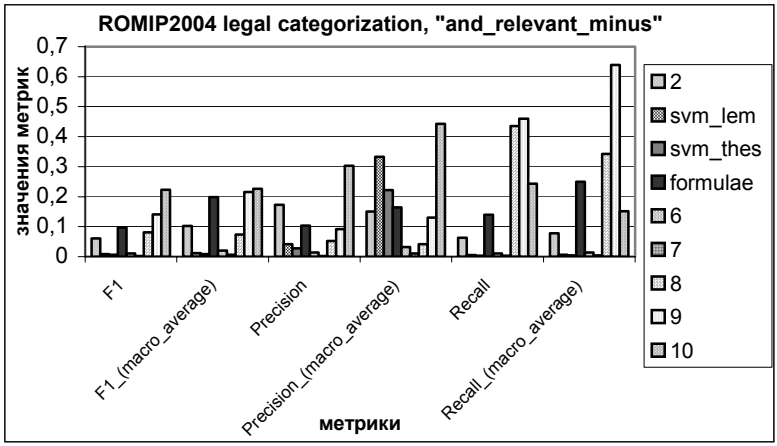

Рис. 15 Результаты прогонов участников для таблицы релевантности “and_relevant-minus”

4.6. Описания рубрик, полученные алгоритмом построения формул

Как мы уже отмечали в разделе 4.4, 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 1 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:

1. алгоритм построения формул на множестве обучения (обозначается train (t));
2. алгоритм построения формул на множестве тестирования — то есть полученный результат на дорожке классификации (formul (f));
3. SVM с расширенным векторным представлением — наш прогон 2 (svm_thes (s));
4. 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")	
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"	

Таблица 1. Описания рубрик, полученные алгоритмом построения формул. Для каждой рубрики указаны результаты различных алгоритмов (описание см. выше)

Рубрика (номер, имя, кол-во док-в)	**F-мера (train (t) formul (f) svm_thes(s) best(b))**
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%
/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 1. (Окончание)

На основе анализа таблицы 1 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) коррелирует с плохой оценкой построенной формулы человеком.

Заключение

Предварительный анализ результатов:

- широко известные, хорошо описанные в литературе методы, как TF*IDF для adhoc поиска и Support Vector Machine для классификации с обучением, показывают достаточно высокие результаты;
- интересной задачей является разработка либо совершенно новых методов либо гибридных методов, превосходящих по результатам «классические» методы.
- РОМИП окреп в организационном плане, что выразилось в организации большего количества дорожек, большем количестве прогонов, выполненном участниками;

По нашему мнению задачи РОМИП на 2005 год:

1) популяризация РОМИП - желательно сделать возможным для всех желающих документы РОМИП (после регистрации и с ограничением на количество, конечно), возможность дополнительной оценки релевантности;
2) привлечение новых участников: снижение «барьера входа», например, посредством предоставления новым участникам данных морфологических индексов, basic line для

«классических» методов; налаживание кооперации с «родственными» конференциями типа CLEF;

3) улучшение организации РОМИП за счет увеличения периода выполнения заданий и сокращения периода подготовки заданий – желательно, чтобы задания рассылались не позже 10 февраля
4) решение задач увеличения финансирования за счет поиска «заказчиков» выполнения тех или иных дорожек, получение грантовой поддержки, в том числе от европейских программ.

Литература

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

[2] Burges C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition // Data Mining and Knowledge Discovery – 1998, V.2, No.2 – pp.121-167.

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

[4] 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.

[5] Joachims T., Making large-scale support vector machine learning practical // Advances in Kernel Methods: Support Vector Machines / B.Schölkopf, C. Burges, A. Smola (eds.) - MIT Press: Cambridge, MA" – 1998. (http://svmlight.joachims.org/)

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

[7] Reuters-21578 text categorization test collection (www.daviddlewis.com/resources/testcollections/reuters21578/)

[8] Salton G, Buckley C., Term-Weighting Approaches // Automatic Text Retrieval. Information Processing and Management. 1988. 24, 5. - pp.513-523.

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

[10] Yang Y., Liu X., A re-examination of text categorization methods // Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval / M.A.Hearst, F.Gey, R.Tong (eds.) - ACM Press: New York, Berkeley, 1999 – pp. 42--49

[11] 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

[12] 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, октябрь 2003, – СПб: НИИ Химии СПбГУ - стр. 87-109.

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

[14] Добров Б.В., Лукашевич Н.В., Построение и использование тематического представления содержания документов. V национальная конференция с международным участием "Искуственный интеллект-96", Казань, 1996, Том I, С.130-134.

[15] Добров Б.В., Лукашевич Н.В., Автоматическая рубрикация полнотекстовых документов по классификаторам сложной структуры // Восьмая национальная конференция по искусственному интеллекту. КИИ-2002. 7-12 октября 2002, Коломна – М.: Физматлит – Т.1 – С.178-186.

[16] Лукашевич Н.В., Салий А.Д., Тезаурус для автоматического рубрицирования и индексирования: разработка, структура, ведение // НТИ. Сер.2. - 1996. - №1. - С.1-6.

Experimental algorithms vs. basic line for web ad hoc, legal ad hoc, and legal categorization in RIRES2004

Mikhail S. Ageev, Boris V. Dobrov,
Natalia V. Loukachevitch, Alexey V. Sidorov

This article describes approaches used by team of UIS RUSSIA (University Information System of Russian inter-University Social Science Information and Analytical consortium, http://www.cir.ru/eng/) search engine for RIRES2004 (Russian Information Retrieval Evaluation Seminar) tracks. We participated in ad hoc track on web collection, ad hoc track on collection of legal documents, and text categorization of legal documents track. Our main goal was to obtain a "basic line" for all tracks using well-known methods. We also made several experimental runs, and compare the performance to "basic line".