Воспроизведение лучших результатов ad hoc поиска семинара РОМИП

Красильников П.В.
Механико-математический факультет
МГУ им. Ломоносова
P.Krasilnikov@gmail.com

Аннотация

Данная работа направлена на более детальное изучение алгоритмов поиска документов, получивших лучшие результаты на семинаре РОМИП в 2004-2006 гг. для задач ad hoc поиска по нормативных документов и по коллекции narod.ru. Цель исследования состояла в том, чтобы путем варьирования параметров небольшого числа основных факторов постараться воспроизвести результаты лучших (но недостаточно задокументированных алгоритмов). Основные факторы, которые были исследованы в настоящей работе, включают, помимо классического TF*IDF веса, вес по парам слов и по кворуму (мере частичного соответствия запроса документу).

1. Введение

Современные поисковые системы учитывают множество (десятки и сотни) факторов для определения документов релевантных заданному запросу. Причем каждый из факторов, чаще всего входит в итоговый алгоритм в нетривиальной форме.

В течение последних трех лет, некоторым участникам [1, 2, 3] семинара РОМИП удалось значительно улучшить качество поиска (точность, полнота) путем учета одновременно нескольких факторов. Однако, было даны недостаточно точные описание алгоритмов, допускающие различные трактовки.

Исследование ставит своей целью попытаться воспроизвести результаты, полученные [1, 2, 3, 4], а также попытаться разобраться с тем, каким образом необходимо учитывать те, или иные факторы.

Данная работа выполнена по гранту компании «Яндекс» «Воспроизведение лучших результатов ad hoc поиска семинара РОМИП. Публикация деталей алгоритмов и результатов исследования влияния различных параметров на качество поиска. Публикация исходных кодов». При подаче заявки на данное исследование, было предусмотрен публичный доступ к исходным кодам и результатам, полученным в процессе работы. Исходные коды и инструкции доступны по адресу http://romip-base.narod.ru. Тестовые коллекции можно получить после подтверждения согласия программного комитете РОМИП, прислав заявку на email указанный в статье. Более подробно о работе системы можно прочесть в разделе **7**.

2. Постановка задачи

2.1 Выбор факторов

Рассматривается классическая задача поиска по коллекции документов. Для некоторой коллекции документов D и запроса Q необходимо выделить документы релевантные запросу и вернуть их в порядке убывания оценки релевантности.

Для решения поставленной задачи обычно применяются алгоритмы, так или иначе учитывающие следующие факторы (наиболее распространенные):

- учет относительной частоты встречаемости слов запроса в найденном документе;
- учет относительной частоты встречаемости слов запроса в документах коллекции;
- взаимного расположения слов;
- близости слов запроса в документе;
- использование морфологии при анализе текста;
- выделение ключевых областей структурированных документов;
- поиск пассажей запроса входящих целиком в документ или в одно предложение;
- вхождение всех слов запроса в документ;
- использование псевдо-обратной связи по релевантности (pseudo-relevance feedback).

Важно иметь ввиду, что при учете каждого из этих факторов существует множество подходов и итоговых математических формул, а так же параметров, значения которых влияют на качество алгоритма и могут быть разными для разных коллекций документов и типов запросов.

В качестве параметров для воспроизведения, исследования и оптимизации, были выбраны следующие:

- вариации формулы TF*IDF – учета относительной частоты встречаемости слов запроса в найденном документе и учета относительной частоты встречаемости слов запроса в документах коллекции;
- координированная «близость» слов в документе и «близость» слов в запросе (учет пар слов);
- вхождение в документ пассажей, содержащих слова запроса
- максимальное число слов запроса входящих

целиком в документ - учет «кворумов»

3. Методы исследования

Основной составляющей поискового алгоритма является ранжирующая функция (параметрами которой являются документ и запрос), зависящая от факторов описанных выше.

Кратко опишем, из каких работы были выбраны первоначальные формулы, а также для каких факторов были придуманы свои вариации:

- для веса основанного на частотных характеристиках отдельных слов запроса в документе, существует множество подходов, однако в данном исследовании использовалась модификация TF*IDF BM25 INQUERY [4], реализованная в системе УИС РОССИЯ. А так же модификация TF*IDF BM25 предложенная в [1, 5];
- при учете «пар слов» запроса в документе была опробована формула, описанная в [1] и несколько собственных интерпретаций и модификаций;
- при учете кворумов и пассажей (которые были объединены в одну группу факторов), были испробованы формулы из [1, 4].

Чтобы учесть все множество существующих подходов, исследование проводится согласно определенной схеме, в несколько этапов.

1. Последовательно, для каждого из выбранных факторов (см. выше) проводится серия экспериментов с различными его интерпретациями. Для каждой интерпретации делается несколько экспериментов с различными значениями параметров формулы (если таковые предполагаются).
2. Выбирается лучшая (в смысле некоторой средней метрики) интерпретация, характеризующаяся собственно формулой, а так же набором значений числовых параметров, при которых этот наилучший результат был получен.
3. Далее, различные факторы (показавшие лучшие результаты в своем классе) последовательно вовлекаются в единую формулу в виде «линейной комбинации» соответствующих весов. То есть с начала в этой линейной комбинации один фактор – TF*IDF, затем два – TF*IDF + «пары слов» и на последнем шаге в линейную комбинацию вовлечены все три фактора – TF*IDF + «пары слов» + пассажи и кворумы.
4. При добавлении очередного «веса» в линейную комбинацию подбирается оптимальный коэффициент при нем, учитывая, что коэффициенты при всех предыдущих уже подобраны. Такой алгоритм поиска наилучшей формулы, с математической точки зрения, вообще говоря, не является обоснованным. Поэтому проводятся дополнительные эксперименты, при которых оптимальный коэффициент при очередном элементе линейной суммы подбирается для некоторой «окрестности» уже подобранных коэффициентов и параметров.
5. Полученные результаты и формулы проверяются на другой коллекции документов и других наборах запросов.

Базовыми заданиями, на которых производился выбор оптимальных параметров являлись коллекции web2004 и legal2004.

4. Метрики

Для оценки качества построенной поисковой системы рассматриваются метрики, используемые на конференции РОМИП в дорожке поиска:

- полнота (recall);
- точность (precision);
- средняя точность (average precision);
- точность на уровне 5 документов (precision(5));
- точность на уровне 10 документов (precision(10));
- 11-точечный график полноты/точности, измеренный по методике TREC (11-point matrix (TREC)).

Точные определения метрик можно найти в [7], в разделе «Официальные метрики».

Основной метрикой, по которой проводилась оптимизация, описанная в предыдущем разделе, была выбрана average precision.

5. Данные

Для проверки эффективности алгоритмов поиска описанных выше, используются web- и legal-коллекции документов российского семинара РОМИП.

Для проверки качества используются результаты оценки по запросам 2004-2006 годов, с учетом размера пула 50 и слабыми требованиями к релевантности (используются матрицы релевантности –or для romip_web и –single для romip_legal).

Исходные документы и запросы были разобраны «на леммы», при помощи инструмента морфологического анализа, используемого в УИС РОССИЯ [4].

6. Процесс исследования

6.1 TF*IDF

Первый фактор, который был воспроизведен в данном исследовании, является TF*IDF [5]. Были опробованы две его интерпретации, встречавшиеся в рассматриваемых публикациях семинара РОМИП.

Первая – TF*IDF BM25 INQUERY, описанная в

[4], с учетом некоторых модификаций, направленных на игнорирование пересчета весов при добавлении документов в коллекции на миллион документов, формула выглядит следующим образом:

$$w_{tfidf-inQery}(d,Q) = \frac{}{\sum_{t\in Q}(0.4+0.6\cdot tf(d,t)\cdot idf(t))}, \quad (1)$$

где

$$tf(d,t) = \frac{freq(d,t)}{freq(d,t)+0.5+1.5\cdot\frac{docLen(d)}{380}}, \quad (2)$$
$$idf(t) = 1-0.16\cdot\log_{10}(df(t))$$

freq(d,t) – число вхождений леммы *t* в документ *d*, *docLen(d)* – длина документа *d* в различных леммах, *df(t)* – число документов коллекции в которые входит лемма *t*.

Вторым вариантом для подсчета TF*IDF веса, была немного упрощенная (т.к. в описании, возможно, была допущена опечатка, а также из-за упрощения исходных данных) а так же исправленная, в силу найденной ошибки формула из [1]:

$$w_{tfidf-y}(d,Q) = \sum_{t\in Q}(-\log(p(t))\cdot tf(d,t)), \quad (3)$$

где

$$tf(d,t) = \frac{freq(d,t)}{freq(d,t)+1+\frac{dl(d)}{350}}, \quad (4)$$
$$p(t) = 1-e^{-1.5\cdot\frac{cf(t)}{|D|}}$$

freq(d,t) – число вхождений леммы *t* в документ *d*, *dl(d)* – длина документа *d* в словах, *cf(t)* – число вхождений лемма *t* в коллекцию, |*D*| - число документов в коллекции.

На **Рис 1.** приведены 11-точечные графики полноты/точности согласно этим двум формулам, а также график системы показавшей наилучший результат (коллекция romip_web, запросы 2004 года). Согласно плану исследования, описанному в разделе **3**, для дальнейших исследований была выбрана формула $w_{tfidf-inQery}$.

Исследование, не ставило своей целью выяснить причины такой существенной разницы, однако, автору показалось странным такое значительное отставание одного из рассматриваемых вариантов TF*IDF от другого. Поэтому были проведены те же вычисления для запросов другого года. На **Рис 2**. видно, что для запросов 2006 года, эти формулы имеют сравнимое качество поиска.

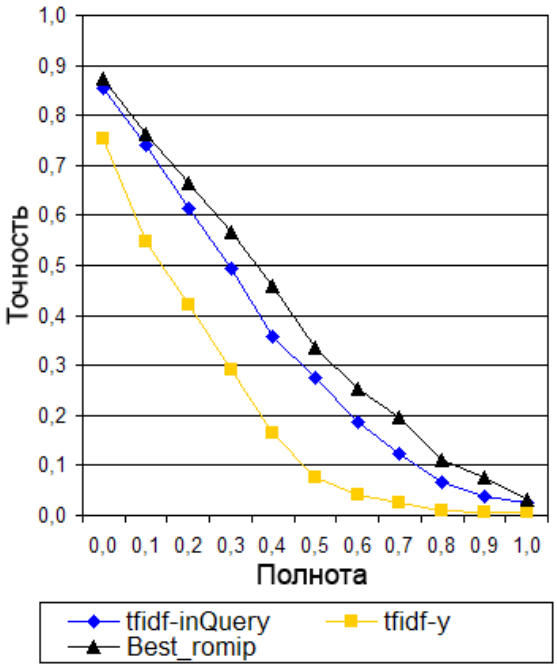

Рис. 1. Сравнение результатов на коллекции web-2004 для разных формул TF*IDF

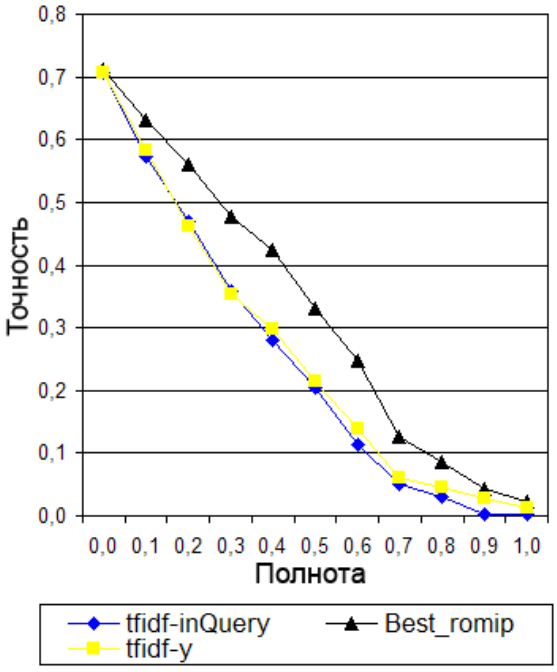

Рис. 2. Сравнение результатов на коллекции web-2006 для разных формул TF*IDF

6.2 Пары слов

Следующий фактором, который был выбран для исследования, является учет «пар слов», упомянутый в [1, 2]. Была опробована формула, описанная в [1]:

$$w_{pair-y} = \sum_{t,s \in Q} (\log(p(t)) + \log(p(s))) \cdot \frac{TF(t,s)}{1 + TF(t,s)} \quad (5)$$

где *p(t)* вычисляется согласно формуле указанной ранее, а *TF* есть число вхождений пары лемм *(t,s)* в документ, с учетом весов: пара учитывается, когда слова запроса встречаются в тексте подряд *(TF + 1)*, через слово *(TF + 0.5)* или в обратном порядке *(TF + 0.5)*. Плюс еще специальный случай, когда слова, идущие в запросе через одно, в тексте встречаются подряд *(TF + 0.1)*.

Однако, оказалось, что такая формула вносит ненулевой хотя бы в один документ для сравнительно небольшого числа запросов (проверенные запросы за 2004 год). Поэтому была придумана следующая интерпретация, в которой имеются два параметра:

$$w_{pair}(d,Q) = \frac{\sum_{t,s \in Q, |t-s|_Q \le a} p(d,t,s)}{\sum_{t,s \in Q, |t-s|_Q \le a} (idf(t) + idf(s))}, \quad (6)$$

где $p(d,t,s)$ равняется $idf(t) + idf(s)$, если леммы *t* и *s* входят в документ *d* на расстоянии не большем чем *b* (параметр алгоритма) и равняется нулю иначе; $|t-s|_Q$ – расстояние между леммами *t* и *s* в запросе, *idf(t)* вычисляется согласно формуле указанной ранее. Стоит заметить, если слова стоят в обратном порядке, то они получат тот же «вес», если бы они стояли в правильном порядке (то есть так, как в запросе).

Также в качестве $p(d,t,s)$ были опробованы:

1. $p(d,t,s) = 1$, если леммы *t* и *s* входят в документ *d* на расстоянии не большем чем *b*, ноль – иначе.
2. $p(d,t,s) = idf(t) \cdot idf(s)$, если леммы *t* и *s* входят в документ *d* на расстоянии не большем чем *b*, ноль – иначе.
3. $p(d,t,s) = \log(p(t)) + \log(p(s))$, если леммы *t* и *s* входят в документ *d* на расстоянии не большем чем *b*, ноль – иначе. *p(t)* определяется, также как и в формуле 4.

Перечисленные подходы показали результаты в среднем ниже, чем первоначальный.

Параметры *a* и *b* были введены для того, чтобы иметь возможность управлять следующими ограничениями:

1. Какие слова из запроса рассматривать в качестве «пар». Часто, особенно это характерно для веб-запросов, пользователь вводит не запрос целиком, а набор ключевых слов. Поэтому совершенно необязательно, что «важные пары» состоят только из слов расположенных близко в запросе.
2. Как близко должны быть расположены слова в документе, чтобы можно было предполагать, что между ними есть смысловая связь.

Были подобраны оптимальные значения параметров *a* и *b*, ***a=5, b=3***. При подборе была выбрана оптимальная пара, по метрике average precision по коллекциям romip_web, romip_legal. Пояснение: average precision оптимизировался чисто для формулы $w_{pair}(d,Q)$, без какого-либо учета TF*IDF и прочих факторов.

Теперь в качестве результирующего «веса» была использована линейная комбинация $w_{tfidf-inQery}$ и w_{pair}:

$$w_2(d,Q) = (1-\lambda) \cdot w_{tfidf-inQuery}(d,Q) + \lambda \cdot w_{pair}(d,Q) \quad (7)$$

После подбора оптимального коэффициента λ (для линейной комбинации) была получена формула:

$$w_2(d,Q) = 0.9 \cdot w_{tfidf-inQuery}(d,Q) + 0.1 \cdot w_{pair}(d,Q) \quad (8).$$

На **рис.3** $w_{tfidf-inQuery}$, w_2 и лучший результат на РОМИП.

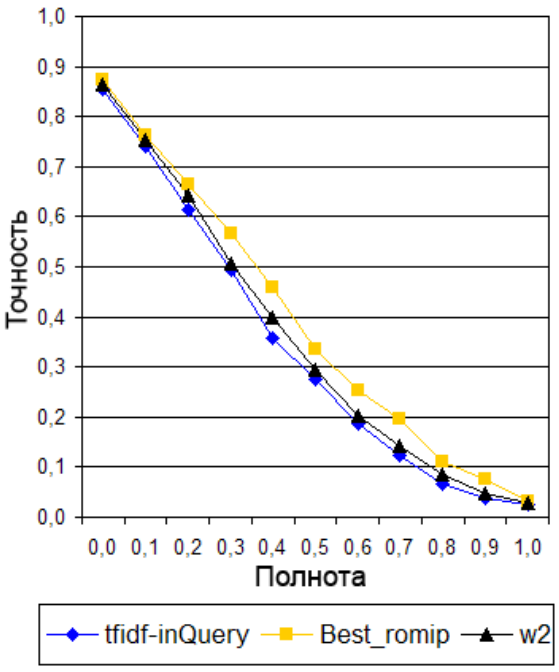

Рис. 3 Сравнение результатов на коллекции web-2004 для TF*IDF и TF*IDF+пары слов

Видно, что удалось немного улучшить результат по сравнению с «чистым» TF*IDF. Стоит заметить, что увеличение average precision при этом произошло не для всех запросов, для некоторых запросов значение этой метрики уменьшилось.

6.3 Учет всех слов запроса

Последний шаг – учет всех слов запроса, или «кворумов». Было опробовано несколько вариантов, опубликованных в [1, 4], в итоге был выбран вариант, описанный в [4], учитывающий размер минимального окна, содержащего все слова запроса:

$$w_{\min-window}(d,Q) = \\ = \frac{1}{\ln(mv(d,Q) - |Q| + 4)}, \quad (9)$$

где *mv(d, Q)* – размер минимального «окна», содержащего все слова запроса *Q*, |*Q*| - длина запроса.

Этот вес был добавлен в линейную комбинацию, которая теперь учитывала три фактора и выглядит следующим образом:

$$w_3(d,Q) = \alpha \cdot w_{tfidf-inQuery}(d,Q) \\ + \beta \cdot w_{pair}(d,Q) \\ + \gamma \cdot w_{\min-window}(d,Q) \quad (10)$$

При подборе оптимальных коэффициентов α, β, γ использовались полученные ранее, $\alpha = 0.9$, $\beta = 0.1$ и было найдено оптимальное значение $\gamma = 0.3$. Подбор значения γ, при небольших изменениях α, β особых улучшений в качестве поиска (рассматривалась метрика average precision) не дал.

Итоговая формула:

$$w_3(d,Q) = 0.9 \cdot w_{tfidf-inQuery}(d,Q) \\ + 0.1 \cdot w_{pair}(d,Q) + \\ + 0.3 \cdot w_{\min-window}(d,Q) \quad (11)$$

На **рис. 4** – 11-точечные графики с учетом двух [w_2] и трех [w_3] факторов, а также лучший результат соответствующего года семинара РОМИП.

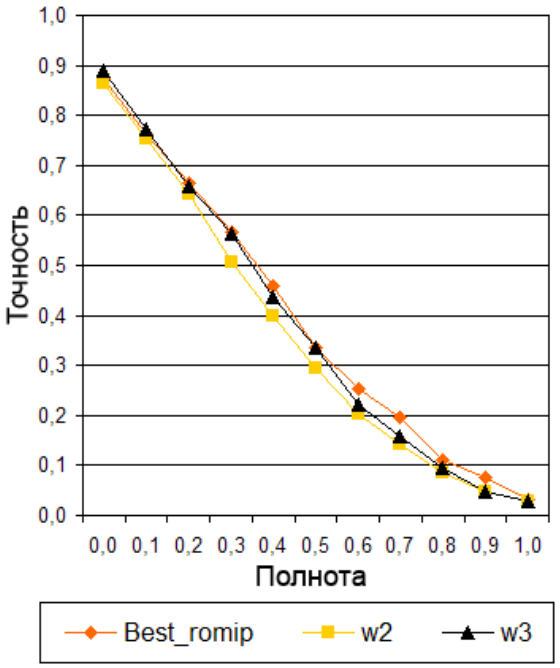

Рис. 4. Сравнение результатов на коллекции web-2004 для TF*IDF + пары слов и TF*IDF + пары слов + минимальное окно

6.4 Тестирование полученных параметров на других задачах семинара РОМИП

Для проверки полученной формулы, она была опробована на другой коллекции документов – romip_legal, а так же на запросах других годов для коллекции romip_web. При этом использовался тот же набором подобранных ранее коэффициентов и значений параметров.

На **рисунках 5-9** 11-точечные графики построенные для трех ключевых результатов:

- лучший результат соответствующего года и дорожки семинара РОМИП
- рассчитанный по формуле $w_{tfidf-inQery}$
- результат, полученный с учетом трех факторов [w_3], по формуле 11.

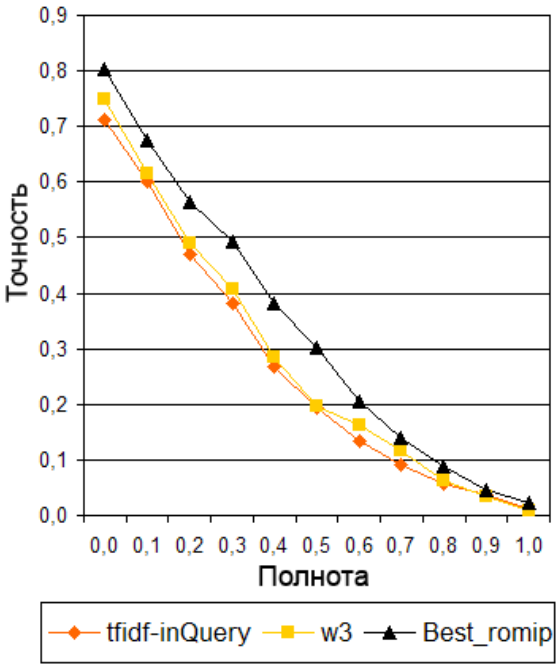

Рис. 5. Сравнение результатов на коллекции web-2005 для TF*IDF и TF*IDF + пары слов + минимальное окно

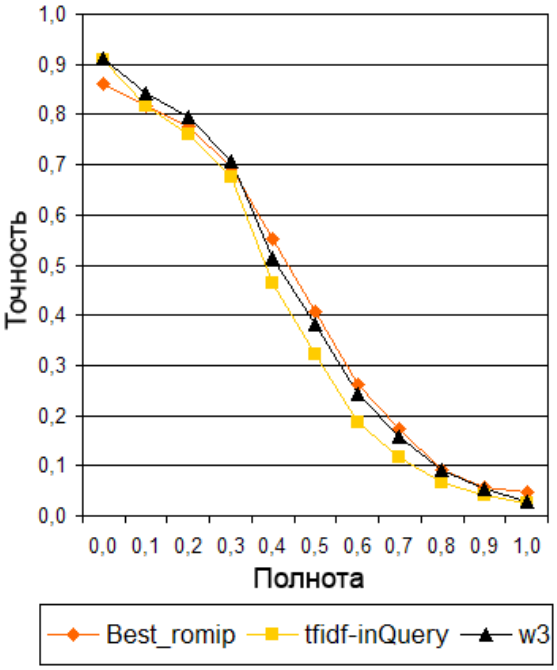

Рис. 7. Сравнение результатов на коллекции legal-2004 для TF*IDF и TF*IDF + пары слов + минимальное окно

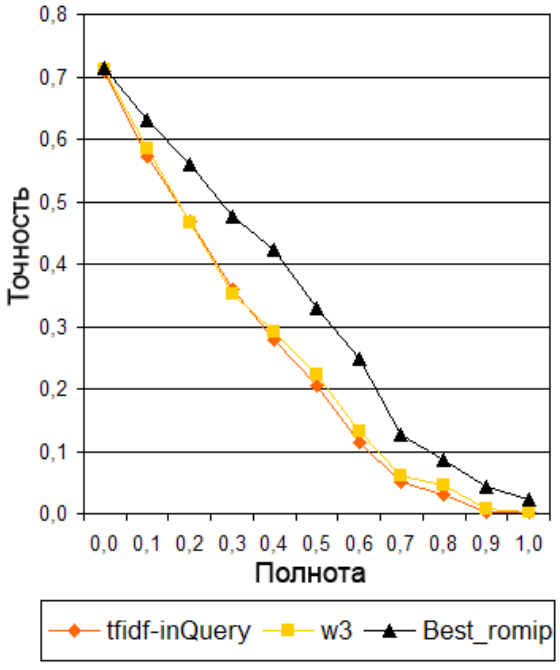

Рис. 6. Сравнение результатов на коллекции web-2006 для TF*IDF и TF*IDF + пары слов + минимальное окно

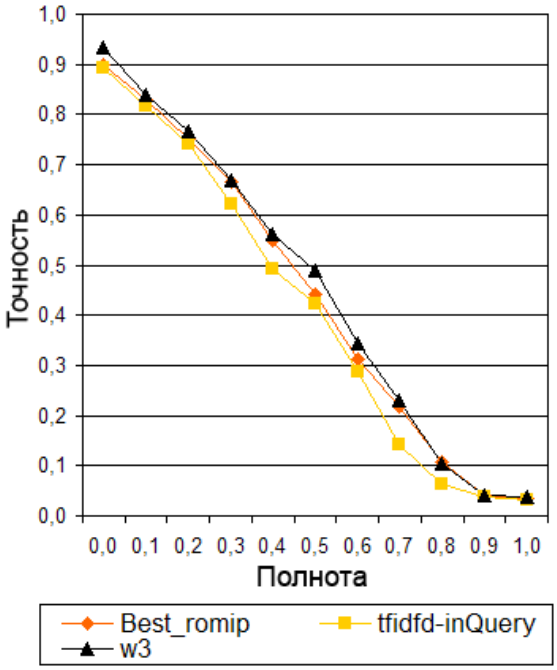

Рис. 8. Сравнение результатов на коллекции legal-2005 для TF*IDF и TF*IDF + пары слов + минимальное окно

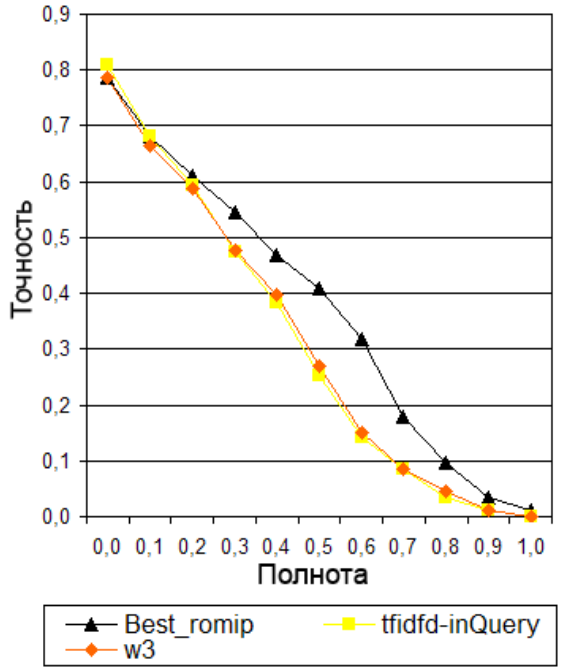

Рис. 9. Сравнение результатов на коллекции legal-2006 для TF*IDF и TF*IDF + пары слов + минимальное окно

7. Исходные коды и тестовые коллекции

При подаче заявки на данное исследование, отдельным пунктом было указано, об обязательной публикации исходных (и бинарных) кодов программного обеспечения которое было использовано для экспериментов.

Отметим, что разработанное программное обеспечение предназначено для оптимизации параметров расчетных формул для достаточно небольшого числа запросов.

Требования, которые учитывались при построении системы:

- возможность легко модифицировать формулы, с небольшими изменениями исходного кода
- легко получать формулы вида «линейная комбинация» весов $w_1, w_{2...} w_n$, которые уже посчитаны.
- быстро получать значения метрик, характеризующих качество поиска для данной коллекции документов и данного набора запросов.

Учитывая эти требования, можно описать процедуру работы с системой:

1. Для данной коллекции и данного набора запросов (при исследовании использовались только группы запросов соответствующего года семинара РОМИП) вычислить «веса» всех документов согласно формулам $w_1, w_{2...} w_n$;
2. Для данной коллекции и данного набора запросов по уже имеющимся значениям весов $w_1, w_{2...} w_n$ для каждой пары документ-запрос вычислить вес $f(w_1, w_{2...} w_n)$. Чаще всего в качестве f выступает линейная комбинация.

Для того, чтобы получить возможность воспроизвести результаты, как это было сделано в этом исследовании, необходимо три составляющих:

- Тестовые коллекции в томе виде, с которым работает разработанное программное обеспечение.
- ПО а также необходимые инструкции по установке и запуску (без внесения изменений в имеющиеся алгоритмы).
- Исходные коды, для реализации собственных идей и алгоритмов.

Не смотря на то, что ПО взаимодействует с реляционной базой данных, все описанные в данной работе формулы (а точнее соответствующие алгоритмы) могут быть реализованы на «инвертированных списках», которые обычно применяются для хранения поисковых индексов.

7.1 Тестовые коллекции

Для работы с тестовыми коллекциями, они должны быть предварительно загружены в базу данных MySQL (использовалась версия 5.0, доступная по адресу http://www.mysql.com).

Каждая тестовая коллекция представляется в виде пяти таблиц:

- docs - таблица с информацией о документах
- lemms - таблица с информацией о леммах
- doc_lemms - таблица с леммо-позициями в документах
- queries_morfo - таблица с разобранными на леммы заданиями (запросами)
- checked_queries - таблица с идентификаторами проверенных запросов для разных годов

Размеры исходных данных (для загрузки), а также загруженных данных (включая индексы) приведены в Таблице 1.

	Размер исходных данных	Размер загруженных данных
romip-web	7.3 Gb	13.0 Gb
romip-legal	1.5 Gb	3.4 Gb

Таблица. 1. Размеры исходных и загруженных коллекций.

Эти исходные данные можно получить,

обратившись к автору или в НИВЦ МГУ, после получения разрешения от программного комитета семинара РОМИП на использование тестовых дорожек. Исходные данные поставляются в виде DVD диска с данными для загрузки, а также содержащего все необходимые инструкции по загрузке в базу данных и использованные автором конфигурационные настройки (что, может быть, поможет сэкономить время на загрузку, а так же необходимое минимальное дисковое пространство).

7.2 Установка и запуск ПО

Программное обеспечение написано на языке Java и доступно для скачивания по адресу http://romip-base.narod.ru. Для запуска необходимо иметь установленным Java Runtime Environment (версии >= 1.6) или Java Development Kit (версии >= 1.6), если необходимо внести изменения в исходные код. Дистрибутивы доступны по адресу http://java.sun.com.

Необходимый для запуска (без изменения исходных кодов) комплект файлов и директорий следующий:

- /romip_base.jar – бинарный файл (для виртуальной машины Java) для запуска,
- /romip2003.jar - бинарный файл, использующийся для вычисления метрик качества поиска,
- /lib/ - директория, содержащая необходимые внешние библиотеки, для запуска приложения,
- /relevant_matrixes/ - директория, содержащая таблицы релевантности (документов запросам) для коллекций romip_legal, romip_web с запросами, использованными в 2004/2005/2006 годах.
- /readme.txt – файл, с необходимыми инструкциями по запуску.

После загрузки тестовых коллекций в базу данных и установки виртуальной машины Java, можно произвести запуск ПО, выполнив следующую команду (в общем виде):

```
java -Xmx1024m
-jar romip_base.jar
-w romip_base.wf.<название класса
     взвешивающей функции>
-d <имя базы данных>
-y <год запросов>
<параметры для данного класса>
<параметры общего характера>
```

Более подробно о параметрах, а так же необходимые примеры можно прочесть в файле /readme.txt. После того как все необходимые вычисления будут произведены, подсчитанные метрики будут записаны в следующий файл:

```
/output
  /<имя базы данных>
    /<год запросов>
      /<название класса
        взвешивающей
        функции><параметры>.txt
```

7.3 Структура исходных кодов

Исходные коды поставляются в виде проекта для среды разработки NetBeans (доступной по адресу http://www.netbeans.org). Собственно исходные коды находятся в папке **/src** и все классы разбиты на два пакета:

- romip_base – содержит основные классы для запуска, обработки входных параметров а так же алгоритмы для подсчета факторов.
- romip_base.wf - содержит множество простых классов для создания итоговой взвешивающей функции, путем составления формулы из подсчитанных значений тех или иных факторов.

Подробнее о структуре классов и их предназначении можно прочитать в файле /readme.html а также в комментариях соответствующих классов.

8. Выводы

Основным выводом проведенного исследования является то, что результаты лучших систем по поиску на коллекции нормативных документов и коллекции narod.ru 2004-2005 годов могут быть приближены подбором оптимальных параметров небольшого числа факторов. Для запросов 2006 года, заметно отставание от лучших результатов.

Интересный результат был получен при исследовании способов учета пар слов в запросе и сопоставляемом документе. При учете пар слов, оказалось выгоднее использовать слова из запроса находящиеся даже на большом расстоянии (а не только подряд или через одно). При этом, в документе слова должно располагаться достаточно близко (не более чем через два слова).

Учет пар слов, а так же веса на основе «кворума» позволяет улучшить классическую TF*IDF формулу.

9. Дальнейшая работа

Автор планирует продолжить исследования по нескольким направлениям:

- Учет форматирования документов. Для web коллекции это, очевидно, учет html форматирования: стиль и размер шрифта, заголовки, ключевые слова страницы, заголовок (title) страницы и так далее. Для legal коллекции учет заголовков документов;
- Проведение экспериментов по очистке коллекций документов;
- Учет метаданных документов;
- использование псевдо-обратной связи по релевантности (pseudo-relevance feedback);

Применение классификации запросов (длина, средний *idf* по всем словам запроса, максимальный *idf* по всем словам запроса и тому подобные), и согласно этой классификации выбирать формулу.

10.Благодарности

Автор благодарит Агеева М.С. и Доброва Б.В. за помощь в подготовке исходных данных для исследования и полезные обсуждения результатов работы.

Литература

[1] А. Гулин, М. Маслов, И. Сегалович. Алгоритм текстового ранжирования Яндекса на РОМИП'2006.

[2] А. Федоровский, М. Костин, А. Проскурин. Mail.Ru на РОМИП'2005.

[3] Илья Сегалович, Михаил Маслов. Яндекс на РОМИП-2004. Некоторые аспекты полнотекстового поиска и ранжирования в Яндекс.

[4] М.С. Агеев, Б.В. Добров, Н.В.Лукашевич, А.В. Сидоров. Экспериментальные алгоритмы поиска/классификации и сравнение с "basic line". РОМИП'2004.

[5] S. E. Robertson, S. Walker, S. Jones, M. M.Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In TREC-3, 1994.

[6] Callan J.P., Croft W.B. and Harding S.M., The INQUERY Retrieval System.

[7] Российский семинар по Оценке Методов Информационного Поиска. http://www.romip.ru

Reproduction best ad hoc search results of ROMIP seminar.

Pavel V. Krasilnikov

This work is aimed at more detailed studying document retrieval algorithms, that received best results at ROMIP seminar in 2004-2006 for ad hoc task on legal documents collection and narod.ru collection. Goal of the research is to reproduce results of best (but not so well documented) algorithms using certain amount of parameters and factors. Factors explored in this research, besides classical TF*IDF weights was: weight based on word pairs frequency and quorum weight (partial similarity between document and query).