Индексная база данных вики-ресурсов: модель, эксперимент, результаты[1]

Крижановский А. А.
Санкт-Петербургский институт информатики и автоматизации Российской академии наук

Новый тип документов в вики-разметке всё более завоёвывает просторы Интернет. Это выражается не только в количестве таких интернет-страниц, но также и в популярности вики-проектов (в частности, Википедии), поэтому всё более актуальной становится задача поиска в вики-текстах. Предложен и реализован способ индексации текстов Википедии на трёх языках: русский, английский, немецкий. Рассмотрена архитектура системы индексирования, включающая программные модули GATE и Lemmatizer. Описаны правила преобразования вики-текстов в тексты на ЕЯ. Построены индексные базы Русской Википедии и Википедии на английском упрощённом языке, выполнено сравнение основных показателей баз данных (число слов, лексем), подтверждающих, что размер Русской Википедии на порядок больше. При этом обнаружен более быстрый рост английской, а именно: за пять месяцев (сент. 2007 — февр. 2008) скорость роста числа статей была больше на 14% и на 7% быстрее чем в русской пополнялся лексикон Википедии на английском упрощённом языке. Выполнена проверка выполнения закона Ципфа для текстов Русской Википедии и Википедии на английском упрощённом языке. В качестве возможных приложений индексной БД рассмотрены методология фильтрации текстовой информации и метод визуализации результатов поиска. Весь исходный код системы индексирования и построенные индексные БД доступны по открытой лицензии GNU GPL.

ВВЕДЕНИЕ

Проведённый в США опрос [Rainie07] говорит о том, что более трети (36%) взрослых пользователей Интернет обращаются за советом к текстам онлайн энциклопедии Википедия. Популярность энциклопедии объясняется огромным количеством, всеохватностью и свежестью материала.

Другая причина народной любви к Википедии кроется в её «авторитетности» в поисковых системах. Например, по данным Hitwise свыше 70% посещений ВП (сб., 17 марта 2007 г.) обеспечены переходами с поисковиков [Rainie07]. Более того, в поисковом интерфейсе Koru[2] анализ ссылок Википедии позволяет определить тематику и терминологию в наборе документов и, в итоге, автоматически уточнить запрос (automatic query expansion) [MilneWitten07].

Данные Википедии можно грубо разделить на текст и ссылки (внутренние, внешние, интервики, категории). Соответственно этому можно назвать три *типа поисковых алгоритмов*[3], которые можно применить к данным Википедии:[4]

1. поиск на основе анализа ссылок, в свою очередь, можно разделить случаи, когда:
   - ссылки заданы явно гиперссылками (HITS [Kleinberg1999], PageRank [Brin1998], [Fortunato2005], ArcRank [Berry2003], Green [Ollivier2007], WLVM [Milne07]);
   - ссылки нужно построить[5] (Similarity Flooding [Melnik2002], алгоритм извлечения синонимов из толкового словаря [Blondel2002], [Blondel2004], [Berry2003]);
2. анализ текста с помощью статистических алгоритмов (ESA [Gabrilovich2007], сходство коротких текстов [Sahami06], извлечение контекстно связанных слов на основе частотности словосочетаний [Pantel2000], LSA [6]);
3. анализ и ссылок, и текста [Bharat1998], [Maguitman2005].

---

1 See English version of this paper: http://arxiv.org/abs/0808.1753. Части данной работы представлены в статьях [Smirnov08] (глава «Архитектура»), [Smirnov08b] (глава с описанием построенных индексных баз данных, проверкой выполнения закона Ципфа), [Krizhanovsky08b] (глава о построенных индексных базах данных).

2 См. http://www.nzdl.org/koru.

3 Нас интересуют алгоритмы поиска документов по ключевым словам, либо поиск «похожих» документов по документу-образцу.

4 См. также обзор и классификацию методов и приложений вычисления сходства коротких текстов в [Pedersen08].

5 Для определения силы связи между словами по совместной встречаемости в документах, либо в общем контексте — могут использоваться специальные алгоритмы [Lande07].

Разработанный ранее адаптированный HITS алгоритм (AHITS) [Krizhanovsky08avia], [Krizhanovsky2006a] выполняет поиск на основе анализа внутренних ссылок Википедии. Многие алгоритмы поиска семантически близких слов[7] в Википедии обходятся без полнотекстового поиска [Krizhanovsky07full] (табл. 3 на стр. 8). Однако экспериментальное сравнение алгоритмов в работах [Gabrilovich2007], [Krizhanovsky07full] показывает, что наилучшие результаты поиска семантически близких слов даёт алгоритм Explicit Semantic Analysis (ESA) [Gabrilovich2007], использующий именно полнотекстовый поиск.

Это подвигнуло нас к созданию общедоступной индексной базы данных Википедии (далее WikIDF[8]) и программных средств для генерации БД, что в целом обеспечит полнотекстовый поиск в энциклопедии и в вики-текстах вообще. Вики-текст — это упрощённый язык разметки HTML.[9] Для индексирования требуется преобразовать его в текст на естественном языке (ЕЯ). Поиск по ключевым словам не будет использовать символы и теги HTML- и вики-разметки, поэтому они будут удалены в ходе преобразования вики-текста в текст на естественном языке (ЕЯ) на предварительном этапе индексирования.

Разработанные программные ресурсы (база данных и система индексирования) позволят учёным проанализировать полученные индексные БД википедий, а разработчикам поисковых систем воспользоваться уже существующей программой и обеспечить поиск по вики-ресурсам за счёт подключения к построенным индексным базам или генерации новых.

Для построения индексной БД и оценки работоспособности выбранного метода индексации необходимо:

- разработать архитектуру системы индексации вики-текстов;
- спроектировать структуру таблиц индексной базы данных;
- определить правила преобразования вики-текста в текст на ЕЯ;
- реализовать программный комплекс для индексации;
- провести эксперименты (здесь проверка выполнения закона Ципфа для вики-текстов).

Структура статьи соответствует поставленным задачам. Заключает работу обсуждение способов улучшения базы, а также проектов и подходов, включающих индексную базу данных как элемент решения. Теперь разберём слаженное взаимодействие подпрограмм, участвующих в построении индексной базы данных.

АРХИТЕКТУРА СИСТЕМЫ ПОСТРОЕНИЯ ИНДЕКСНОЙ БД ВИКИ-ТЕКСТОВ

В данной работе была спроектирована архитектура программной системы индексирования вики-текстов.[10] На рис. 1 показана организация взаимодействия программ GATE [Cunningham2005], Lemmatizer [Sokirko2004] и Synarcher [Krizhanovsky08avia], [Krizhanovsky2006a]. Полезным результатом работы системы будет индексная БД на уровне записей (англ. «*record level inverted index*»), содержащая список ссылок на документы для каждого слова (точнее — для каждой леммы).[11]

Программной системе требуется задать три группы входных параметров. Во-первых, язык текстов Википедии (один из 254 на 16/01/2008) и один из трёх языков (русский, английский, немецкий) для лемматизации, что определяется наличием трёх баз данных, доступных лемматизатору [Sokirko2004]. Указание языка Википедии необходимо для правильного

6 См. в [Lemaire06] о взаимосвязи совместной встречаемости слов и смыслового сходства слов, а особенно разбор того случая, когда частота совместной встречаемости намного превышает сходство слов.

7 Под семантически близкими словами (СБС) подразумеваются слова близкие по значению, встречающиеся в одном контексте. Это могут быть синонимы («чертог», «дворец»), антонимы («запутать», «распутать») и др.

8 WikIDF – аббревиатура, отражающая применение подхода TF-IDF [Robertson2004], [Segalovich2002] к индексированию вики-текстов.

9 См. http://en.wikipedia.org/wiki/Wikitext.

10 Фантазии по поводу применения данной архитектуры для (1) тематической вики-индексации и (2) фильтрации потоков текстов см. в работе [Smirnov08].

11 Об эффективном индексировании см. обзор в работе [Ferragina2008]. Об инвертированном файле см. в [Segalovich2002], а также http://en.wikipedia.org/wiki/Inverted_index.

преобразования текстов в вики-формате в тексты на ЕЯ (рис. 1, функция «Преобразование вики в текст» модуля «Обработчик Википедии»)[12]. Во-вторых, *адрес вики и индексной баз данных*, а именно обычные параметры для подключения к удалённой БД: IP-адрес, имя БД, имя и пароль пользователя. В-третьих, *параметры индексирования*, связанные с ограничениями, накладываемыми пользователем на размер индексной БД, предназначенной для последующего поиска по TF-IDF схеме.[13]

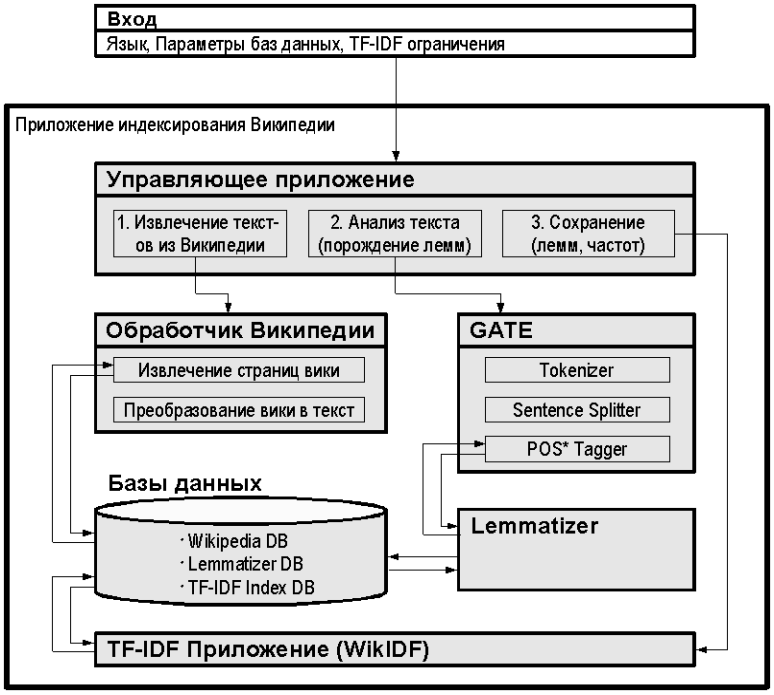

Рис. 1. Архитектура системы индексирования вики-текстов (POS — Part Of Speech)

«Управляющее приложение» выполняет последовательно три шага для каждой статьи (вики-документ), извлекаемой из БД Википедии и преобразуемой в текст на ЕЯ, что и составляет первый шаг. На втором шаге привлекается такой мощный инструмент как GATE [Cunningham2005] вкупе с отечественной разработкой Lemmatizer [Sokirko2004] и объединяющей их программной «прослойкой» RussianPOSTagger[14], конечная цель которых – получение списка лемм и их частоты встречаемости в данной статье.[15] На третьем шаге полученные данные сохраняются в индексную БД[16]: (1) полученные леммы, (2) частота их встречаемости в тексте, (3) указывается, что данные леммы принадлежат данному вики-тексту, (4) увеличивается значение частоты данных лемм в корпусе.

Следует отметить, что две функции модуля «Обработчик Википедии», указанные на рис. 1, а также API доступа к индексной БД («TF-IDF Index DB» из модуля «TF-IDF Приложение») реализованы в программе Synarcher[17]. Указание входных параметров и запуск индексации осуществляются с помощью модуля Synarcher: WikIDF[18].

---

12 Более подробно о преобразование текста см. на стр. 5.

13 Например, ограничение числа связок *слово-страница*. В экспериментах ограничение равно 1000, см. табл. 3.

14 Более подробно о программах *Lemmatizer* и *Russian POS Tagger* см.: http://rupostagger.sourceforge.net.

15 Точнее — суммарной частоты всех словоформ данной лексемы в заданной статье (и во всём корпусе) для каждой леммы. Лемма строится по словоформе с помощью программы Lemmatizer. Определение словоформы, леммы и лексемы см. в Русском Викисловаре.

16 Построенные таким способом индексные БД Русской Википедии и Simple Wikipedia доступны по адресу: http://rupostagger.sourceforge.net, см., соответственно, пакеты *idfruwiki* и *idfsimplewiki*.

17 Эта функциональность доступна начиная с версии *Synarcher 0.12.5*, см. http://synarcher.sourceforge.net

18 WikIDF — это консольное приложение на языке Java. Оно зависит от библиотек программы Synarcher и поставляется в комплекте с последней.

ТАБЛИЦЫ И ОТНОШЕНИЯ В ИНДЕКСНОЙ БД

Для хранения информации, полученной при индексировании, используется реляционная база данных [Codd90]. Принципы построения индексной БД таковы:[19]

- данные наполняются единожды и далее используются *только для чтения* (поэтому не рассматриваются такие вопросы, как: обновление индекса, добавление записи, поддержка целостности);
- из викитекста удаляется вики- и HTML- разметка; выполняется лемматизация, леммы слов сохраняются в базу;
- данные хранятся в несжатом виде.

Число таблиц в индексной базе данных, их наполнение и связи между ними были определены исходя из решаемой задачи: поиск текстов по заданному слову с помощью TF*IDF формулы (см. ниже). Для этого достаточно трёх[20] таблиц[21] и ряда полей (рис. 2):

1. *term* — таблица содержит леммы слов (поле *lemma*); число документов, содержащих словоформы данной лексемы (*doc_freq*); суммарная частота словоформ данной лексемы по всему корпусу (*corpus_freq*);
2. *page* — список названий проиндексированных документов (поле *page_title* в точности соответствует полю одноимённой таблицы в БД MediaWiki); число слов в документе (*word_count*);
3. *term_page* — таблица, связывающая леммы словоформ, найденных в документах с этими документами.

Постфикс «*_id*» в названии полей таблиц обозначает уникальный идентификатор (рис. 2). Ниже горизонтальной полосы в рамке каждой таблицы перечислены поля, проиндексированные для ускорения поиска. Между полями таблиц задано отношение *один ко многим* — между таблицами *term* и *term_page* (term_id), а также таблицами *page* и *term_page* (page_id).

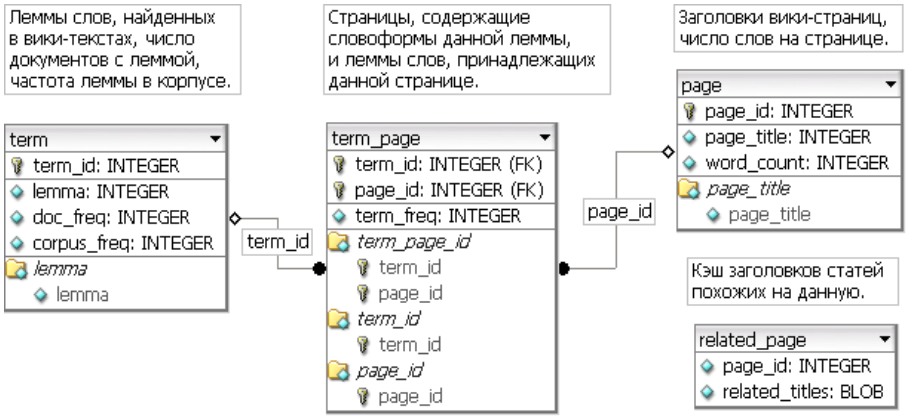

Рис. 2. Таблицы и отношения в индексной базе данных WikIDF[22]

---

19 См. принципы проектирования индексов поисковых систем: http://en.wikipedia.org/wiki/Index_(search_engine).

20 Четвёртая таблица *related_page* нужна для вспомогательной функции — кэширования похожих страниц, найденных с помощью AHITS алгоритма [Krizhanovsky2006a]. Она использовалась при экспериментальной оценке работы AHITS, поскольку слова в тестовом наборе многократно повторялись [Krizhanovsky07full].

21 Для сравнения см. таблицы индексной БД на стр. 10 в работе [Papadakos08], где дополнительная таблица указывает смещение слов (позиция в тексте). В работе [Papadakos08] описана архитектура греческой поисковой системы Mitos.

22 Для проектирования и визуализации таблиц БД использовалась система визуального проектирования баз данных DBDesigner 4, см. http://www.fabforce.net/dbdesigner4.

Данная схема БД позволяет получить:

- список лемм слов заданного документа;[23]
- список документов, содержащих словоформы лексемы, заданной своей леммой.[24]

Напомним читателю формулу TF-IDF (1), с прицелом на которую и была спроектирована вышеуказанная схема БД. Всего в корпусе D документов, термин (лексема) t_i встречается в DF_i документах (чему соответствует значение поля БД *term.doc_freq*[25]). Для заданного термина t_i вес документа $w(t_i)$ определяется как [Robertson2004]:

$$w(t_i) = TF_i \cdot idf(t_i) \quad ; \quad idf(t_i) = \log \frac{D}{DF_i} \tag{1}$$

где TF_i — число вхождений термина t_i в документ (поле *term_page.term_freq*), *idf*[26] служит для уменьшения веса высокочастотных слов. Можно нормализовать TF_i, учтя длину документа, то есть разделив на число слов в документе (поле *page.word_count*). Таким образом, значения полей БД позволяют вычислить обратную частоту термин t_i в корпусе.

Отметим, что в результате построения индексной БД оказалось, что: размер индекса составляет 26-38% от размера файла с текстами индексируемого вики-проекта.[27,28]

ПРЕОБРАЗОВАНИЕ ВИКИ-ТЕКСТА С ПОМОЩЬЮ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Тексты Википедии написаны по правилам вики-разметки. Существует насущная необходимость в преобразовании вики-текста, а именно в удалении, либо «раскрытии» тегов вики (то есть извлечении текстовой части). Если опустить данный шаг, то в сотню наиболее частых слов индексной БД попадают специальные теги, например «ref», «nbsp», «br» и др.[29] В ходе работы возникали вопросы: «как и какие элементы разметки обрабатывать?». Представим, аналогично работе [Vahitova06] (стр. 20), возникшие вопросы и принятые решения в табл. 1. Для наиболее интересных, но могущих быть записанными в одну строку преобразований в таблице приведены регулярные выражения [Friedl2001].

---

23 Точнее: возможно меньше число, чем все леммы. Поскольку для слов, встречающихся больше чем в N документах (здесь тысяча), N+1 связка «слово-документ» не будет записана в таблицу *term_page*.

24 То же ограничение, до 1000 документов здесь.

25 term.doc_freq — это сокращённая запись, указывающая на поле *doc_freq* таблицы *term* индексной базы данных.

26 «idf (обратная частота термина в документах, обратная документная частота) – показатель поисковой ценности слова (его различительной силы)» [Segalovich2002]. В 1972 г. Karen Sparck Jones предложил такую эвристику: «термин запроса, встречающийся в большом количестве документов, обладает слабой различительной силой (широко употребимое слово), ему должен быть присвоен меньший вес, по сравнению с термином, редко встречающимся в документах коллекции (редкое слово)». Данная эвристика показала свою пользу на практике и в работе [Robertson2004] представлено её теоретическое обоснование.

27 Цифры 26-38% получены как отношение значения поля «Исходный дамп, размер» к «Размер сжатого файла дампа индексной БД» в табл. 3 на стр. 9.

28 В работе [Papadakos08] всё значительно лучше, а именно: худшее (максимальное) значение составляет 13%, при этом «до 40% результирующего индекса в БД индекса Mitos — это информация о позиции слов» (то есть то, что в индексе WikIDF отсутствует вовсе).

29 Собственно, анализ самых частых терминов, полученных в индексной БД, позволял находить элементы вики-разметки, требующие обработки. Код переписывался и база генерировалась вновь.

Таблица 1

Решения по парсингу вики-текста

N	**Вопросы** **Исходный текст**	**Ответы** **Преобразованный текст**

1	Заголовки (подписи) рисунков
	`[[Изображение:Через-тернии-к-звёздам.jpg|thumb|«Через тернии к звёздам»]]`
	`[[Image:Asimov.jpg|thumb|180px|right|[[Isaac Asimov]] with his [[typewriter]].]]`
		Оставить (извлечь)
		`«Через тернии к звёздам»`
		
		`[[Isaac Asimov]] with his [[typewriter]].`

2	Интервики	Оставить или удалить определяется пользователем (параметр `b_remove_not_expand_iwiki`).
3	Название категорий	Удалить.
	Регулярное выражение (RE): **\[\[Категория:.+?\]\]**	
4	Шаблоны; цитаты[30]; таблицы	Удалить.
5	Курсивный шрифт и «жирное» написание. `'''italic''' ''bold''`	Знаки выделения (апострофы) удаляются. `italic bold`

6	Внутренняя ссылка
	
	`[[w:Wikipedia:Interwikimedia_links|text to expand]]`
	`[[run]]`
	`[[Russian language|Russian]]`
	`в [[космос|космическом пространстве]].`
		Оставить текст, видимый пользователю, удалить скрытый текст.
		`text to expand`
		
		`run`
		`Russian`
		`в космическом пространстве.`

	RE: внутренняя ссылка без вертикальной черты: **\[\[([^:|]+?)\]\]**	

7	Внешняя ссылка
	
	`[http://example.com Russian]`
	`[http://www.hedpe.ru сайт hedpe.ru – русский фан-сайт]`
		Оставить текст, видимый пользователю, удалить сами гиперссылки.
		`Russian`
		`сайт – русский фан-сайт`

	RE: Имя сайта (без пробелов), содержащее точку '.' хотя бы раз, кроме последнего символа: **(\A|\s)\S+?[.]\S+?[^.]([\s,!?]|\z)**	
8	Примечание. `word1<ref>Ref text.</ref> – word2.`	Раскрывать, переносить в конец текста. `word1 word2.\n\nRef text.`

Для преобразования текстов в вики-формате в тексты на ЕЯ последовательно выполняются следующие шаги, которые можно разбить на две группы: (i) удаление и (ii) преобразование текста.[31]

(i) Удаляются такие теги (вместе с текстом внутри них):

1. HTML комментарии (`<!-- ... -->`);
2. теги выключения форматирования (`<pre>...</pre>`)[32];
3. теги исходных кодов `<source>` и `<code>`.

(ii) Выполняются преобразования вики-тегов:

4. Извлекается текст примечаний `<ref>`, добавляется в конец текста;
5. Удаляется двойные фигурные скобки и текст внутри них (`{{шаблон}}`);[33]
6. Удаляются таблицы и текст (`{| table 1 \n {| A table in the table 1 \n|}|}`).
7. Удаляется знак ударения в текстах на русском языке (например, *Ко́мор*);

30 Имеются в виду короткие цитаты: *{{цитата|текст}}*, см. http://ru.wikipedia.org/wiki/Википедия:Шаблоны/Форматирование.

31 См. код функции *wikipedia.text.WikiParser.convertWikiToText* в программе Synarcher.

32 Поскольку теги <pre> обычно «оборачивают» исходный код программ, не содержащий текстов на ЕЯ, см. http://en.wikipedia.org/wiki/Wikipedia:How_to_edit_a_page#Character_formatting.

33 Данная подфункция вызывается дважды, чтобы удалить `{{шаблон в {{шаблоне}}}}`. Более глубокие вложения в данной версии не учитываются.

8. Удаляется тройной апостроф, окружающий текст и обозначающий '''жирное выделение'''; текст остаётся;
9. Удаляется двойной апостроф, обозначающий ''наклонное начертание''; текст остаётся;
10. Из тега изображения извлекается его название, прочие элементы удаляются;
11. Обрабатываются двойные квадратные скобки (раскрываются внутренние ссылки, удаляются интервики и категории);
12. Обрабатываются одинарные квадратные скобки, обрамляющие гиперссылки: ссылка удаляется, текст остаётся;
13. Удаляются символы (заменяются на пробел), противопоказанные XML парсеру:[34] <, >, &, "; удаляются также их «XML-безопасные» аналоги: <, >, &, "; а также: ', , –, — символы
,
,
 заменяются символом перевода каретки.

Данный преобразователь вики-текста воплощён в виде одного из Java-пакетов программной системы Synarcher [Krizhanovsky2006a]. Для замены элементов текста широко использовались регулярные выражения [Friedl2001] языка Java. В табл. 2 приведён фрагмент статьи Русской Википедии «Через тернии к звёздам (фильм)». Показан результат комплексного преобразования текста по всем вышеуказанным правилам.

Таблица 2

Пример преобразования вики-текста[35]

Исходный текст в вики-разметке	**Преобразованный текст**
{{Фильм | РусНаз = Через тернии к звёздам }} [[Изображение:Через-тернии-к-звёздам 2.jpg|thumb|«Через тернии к звёздам»]] '''«Чéрез тéрнии к звёздам»''' – [[научная фантастика|научно-фантастический]] двухсерийный фильм [[режиссёр]]а [[Викторов, Ричард Николаевич|Ричарда Викторова]] по сценарию [[Кир Булычёв|Кира Булычёва]]. == Сюжет == {{сюжет}} [[XXIII]] век. [[Звездолёт]] дальней разведки обнаруживает в [[космос]]е погибший корабль неизвестного происхождения, на нём – гуманоидных существ, искусственно выведенных путём клонирования. Одна девушка оказывается жива, её доставляют на [[Земля (планета)|Землю]], где [[учёный]] Сергей Лебедев поселяет её в своём доме. == В ролях == * [[Елена Метёлкина]] – ''Нийя'' == Ссылки == {{викицитатник}} * [http://ternii.film.ru/ Официальный сайт фильма] [[Категория:Киностудия им. М. Горького]] [[en:Per Aspera Ad Astra (film)]]	«Через тернии к звёздам» «Через тернии к звёздам» научно-фантастический двухсерийный фильм режиссёра Ричарда Викторова по сценарию Кира Булычёва. == Сюжет == XXIII век. Звездолёт дальней разведки обнаруживает в космосе погибший корабль неизвестного происхождения, на нём гуманоидных существ, искусственно выведенных путём клонирования. Одна девушка оказывается жива, её доставляют на Землю, где учёный Сергей Лебедев поселяет её в своём доме. == В ролях == * Елена Метёлкина Нийя == Ссылки == * Официальный сайт фильма

34 Имеется в виду парсер протокола XML-RPC системы RuPOSTagger.
35 См. http://ru.wikipedia.org/wiki/Через_тернии_к_звёздам_(фильм).

API ИНДЕКСНОЙ БАЗЫ ДАННЫХ ВИКИ

Укажем существующие программные интерфейсы (API) для работы с данными ВП:

- FUTEF API для поиска в английской Википедии с учётом категорий ВП.[36] Поисковик реализован как веб-сервис на основе Yahoo!, результат возвращается в виде Javascript объекта JSON;[37]
- интерфейс для вычисления семантического сходства слов в ВП [Ponzetto07]. Здесь запрос идёт из Java через XML-RPC к Perl-процедуре, затем посредством MediaWiki выполняется обращение к БД;
- интерфейс к Википедии и Викисловарю [Zesch08]. Проведены эксперименты по извлечению данных из Английского и Немецкого Викисловаря. Главный недостаток программы — лицензия — «только для исследовательских целей».
- набор интерфейсов для работы с данными ВП, хранимыми в XML базе данных Sedna.[38]

Поскольку структура индексной БД отличается от схемы БД MediaWiki (для работы с БД MediaWiki уже написано достаточное количество необходимых функций в программе Synarcher), постольку возникла необходимость в разработке «сопряжения» для программного управления индексом. Итак, разработан программный интерфейс для работы с базой данных WikIDF.

I.) Интерфейс верхнего уровня позволяет:[39]

1. получить список терминов для данной вики-страницы, упорядоченный по значению TF-IDF;[40]
2. получить список документов, содержащих словоформы лексемы по заданной лемме; документы упорядочены по значению частоты термина (TF).[41]

II.) Функции низкого уровня для работы с отдельными таблицами индексной БД (рис. 2) реализованы в пакете `wikipedia.sql_idf` программы Synarcher.

ЭКСПЕРИМЕНТЫ ПО ПОСТРОЕНИЮ ИНДЕКСНЫХ БАЗ ДАННЫХ

Разработанная программная система индексирования вики-текстов позволила построить индексные БД для Simple English[42] (далее SEW) и Русской[43] (далее RW) википедий и провести эксперименты. Статистическая информация об исходных БД, о парсинге и о размерах полученных БД представлены в табл. 3.

В двух столбцах («RW / SEW 07» и «RW / SEW 08») указано во сколько раз параметры русского корпуса превосходят английский по дампам от 2007 года (SEW от 9 и RW от 20 сентября) и от 2008 года (SEW от 14 и RW от 20 февраля). Данными, характеризующими корпус текстов Русской Википедии, можно назвать большое количество лексем (1.43 млн) и общего числа слов (32.93 млн). Размер Русской Википедии примерно на порядок больше Английской (столбец «RW/SEW 08»): статей больше в 9.5 раз, лексем — в 9.6, всего слов — в 14.4 раза.

Значения следующих двух столбцов («SEW 08/07 %» и «RW 08/07 %») указывают, насколько выросли (по сравнению с собой же) английский и русский корпуса за пять месяцев с сентября 2007 до февраля 2008 гг.

---

36 См. http://api.futef.com/apidocs.html.

37 См. http://json.org/json-ru.html.

38 См. http://modis.ispras.ru/sedna/ и http://wikixmldb.dyndns.org/help/use-cases/ — разработка Института системного программирования РАН.

39 См. пример работы с данными функциями в файле: synarcher/wikidf/src/wikidf/ExampleAPI.java.

40 См. табл. 4 с результатами работы данной функции в приложении на стр. 22.

41 См. табл. 5 с результатами работы данной функции в приложении на стр. 24.

42 1000 наиболее частотных слов, полученных по текстам Википедии на упрощённом английском языке, см. http://simple.wiktionary.org/wiki/User:AKA_MBG/English_Simple_Wikipedia_20080214_freq_wordlist. (14.02.2008)

43 1000 наиболее частотных слов, полученных по текстам Русской Википедии (20 февраля 2008), см. http://ru.wiktionary.org/wiki/Конкорданс:Русскоязычная_Википедия/20080220.

В последнем столбце (SEW↑ — RW↑) указано насколько быстрее шёл рост английского корпуса по сравнению с русским (разница предыдущих двух столбцов), а именно: на 14% быстрее появлялись статьи и на 7% быстрее пополнялся лексикон Википедии на английском упрощённом языке.

Таблица 3

Статистика по Русской Википедии и Simple Wikipedia, парсингу и сгенерированным индексным базам данных.

БД Википедии	**Simple English (SEW 07)**	**Simple English (SEW 08)**	**Russian (RW 07)**	**Russian (RW 08)**	**RW/SEW 07**	**RW/SEW 08**	**SEW 08/07 %**	**RW 08/07 %**	**SEW↑ –RW↑ %**
База данных Википедии							**SEW↑**	**RW↑**	
Исходный дамп, дата	9 сент. 2007	14 фев. 2008	20 сент. 2007	20 фев. 2008	–	–	–	–	–
Исходный дамп, размер, МБ[44]	15.13	21.11	240.38	303.59	15.9	14.4	140	126	13
Статей, тыс.	19.235	25.22	204.78	239.29	10.7	**9.5**	131	117	**14**
Парсинг									
Парсинг всего, ч	3.474	3.628	52.37	69.26	15.1	19.1	104	132	-28
Парсинг одной страницы, сек	0.65	0.518	0.9206	1.042	1.42	2.01	79.7	113.2	-33.5
Индексная база данных Википедии									
Лексем в корпусе, млн	0.1213	0.1487	1.2406	**1.434**	10.2	**9.6**	123	116	**7**
Лексема-страница (<=1000 для слова)[45], млн	1.3365	1.6524	13.49	15.71	10.1	9.5	123.6	116.5	7.2
Слов в корпусе, млн	1.765	2.2839	26.7	**32.93**	15.1	**14.4**	129	123	6
Размер сжатого файла дампа индексной БД, МБ	5.74	7.15	66.1	77.5	11.5	10.8	125	117	7

Чтобы время парсинга, приведённое в табл. 3, имело смысл, укажем параметры рабочего компьютера и версии двух основных программ: ОС Debian 4.0 etch, ядро Linux 2.6.22.4, процессор AMD 2.6 ГГц, 1 ГБ RAM, Java SE 1.6.0_03, MySQL 5.0.51a-3.

Теперь обратимся к такому интересному вопросу корпусной лингвистики как распределение частот слов и заключим экспериментальный раздел проверкой наличия распределения Ципфа для текстов Википедии.

ПРОВЕРКА ВЫПОЛНЕНИЯ ЗАКОНА ЦИПФА ДЛЯ ВИКИ-ТЕКСТОВ[46]

Эмпирический закон Ципфа говорит о том, что частота употребления слова в корпусе обратно пропорциональна его рангу в списке упорядоченных по частоте слов этого корпуса [Manning99] (стр. 23), то есть второе по частоте слово будет употребляться в текстах в два раза реже чем первое, третье — в три раза и так далее.

Другая формулировка закона Ципфа гласит: если построить список слов, отранжировав слова по уменьшению их частоты встречаемости в некотором *достаточно большом* тексте, и

44 Размер файла «...-pages-articles.xml.bz2», содержащего тексты статей.

45 Число связок «слово-статья» в корпусе, учёт не более 1000 для одного слова. Число *1000* здесь — это один из входных параметров программного комплекса построения индексной БД, см. «TF-IDF ограничения» на рис. 1.

46 Следует признать, что к данному вопросу авторов привлёк рисунок с распределением частот слов в Английской Википедии за 2006 г., см. http://en.wikipedia.org/wiki/Zipf%27s_law#Related_laws.

нарисовать график логарифма частот слов в зависимости от логарифма порядкового номера в списке, то получится прямая [Robertson2004]. Такой график мы и нарисуем.[47]

На рис. 3 слова упорядочены (по убыванию частоты) вдоль оси абсцисс, вдоль оси ординат отложена частота слов. Кривая, составленная из знаков «+», построена по данным корпуса текстов Русской Википедии «RW 08»[48]. С помощью метода наименьших квадратов пакета Scilab [Campbell06] были построены и нарисованы *аппроксимирующие кривые* y_{100}^{RW} по первым ста наиболее частотным словам корпуса (см. рис. 3, кривая голубого цвета, пунктир с точкой) и $y_{10\text{K}}^{RW}$ по первым 10 тыс. слов (розовая линия, длинный пунктир):

$$y_{100}^{RW}(x)=\frac{\mathrm{e}^{14.51}}{x^{0.819}} \quad ; \quad y_{10\text{K}}^{RW}(x)=\frac{\mathrm{e}^{16.13}}{x^{1.048}} \tag{2}$$

Знакам «X» на рис. 3 соответствуют данные Википедии «SEW 08». Аналогично нарисованы аппроксимирующие кривые: y_{100}^{SEW} (зелёного цвета, точечный пунктир) и $y_{10\text{K}}^{SEW}$ (красного цвета, пунктир с двумя точками).

$$y_{100}^{SEW}(x)=\frac{\mathrm{e}^{12.83}}{x^{0.974}} \quad ; \quad y_{10\text{K}}^{SEW}(x)=\frac{\mathrm{e}^{14.29}}{x^{1.174}} \tag{3}$$

Отметим, что аппроксимирующая линия $y_{10\text{K}}^{RW}$ является более пологой (с угловым коэффициентом равным −1.048) чем более крутая линия $y_{10\text{K}}^{SEW}$ (коэффициент −1.174), соответствующая более резкому падению частот английских слов. Возможные объяснения таковы. Во-первых, размер Русской Википедии на порядок больше (табл. 3), и, по-видимому, задействован более широкий словарь для описания большего количества понятий. Во-вторых, авторы Википедии на английском упрощённом языке (Simple Wikipedia) осознанно стараются пользоваться более простыми словами и, таким образом, обходятся меньшим словарным запасом.

Рис. 3 показывает, что закон Ципфа в целом выполняется для текстов википедий, то есть кривую на рисунке с логарифмическим масштабом вполне можно аппроксимировать прямой. При этом данные Simple Wikipedia (0.20)[49] соответствуют данному закону немного лучше корпуса русских текстов (0.23). Такое пристрастие выполнения закона к текстам на английском упрощённом языке, по сравнению с русским, можно объяснить либо особенностью упрощённого языка, либо разницей между русским и английским языками. Для окончательного выяснения вопроса нужно решить задачу промышленного масштаба, а именно: построить индексную БД для огромной Английской Википедии (English Wikipedia).

47 Исходный код для построения линейных аппроксимаций в Scilab методом наименьших квадратов приводится в приложении на стр. 26.

48 «RW 08» — расшифровку и пояснения см. в предыдущей главе и в табл. 3.

49 0.20 — разница между угловыми коэффициентами (степенями наклона) аппроксимирующих прямых, построенных по ста (0.974) и 10 тысячам (1.174) английских слов.

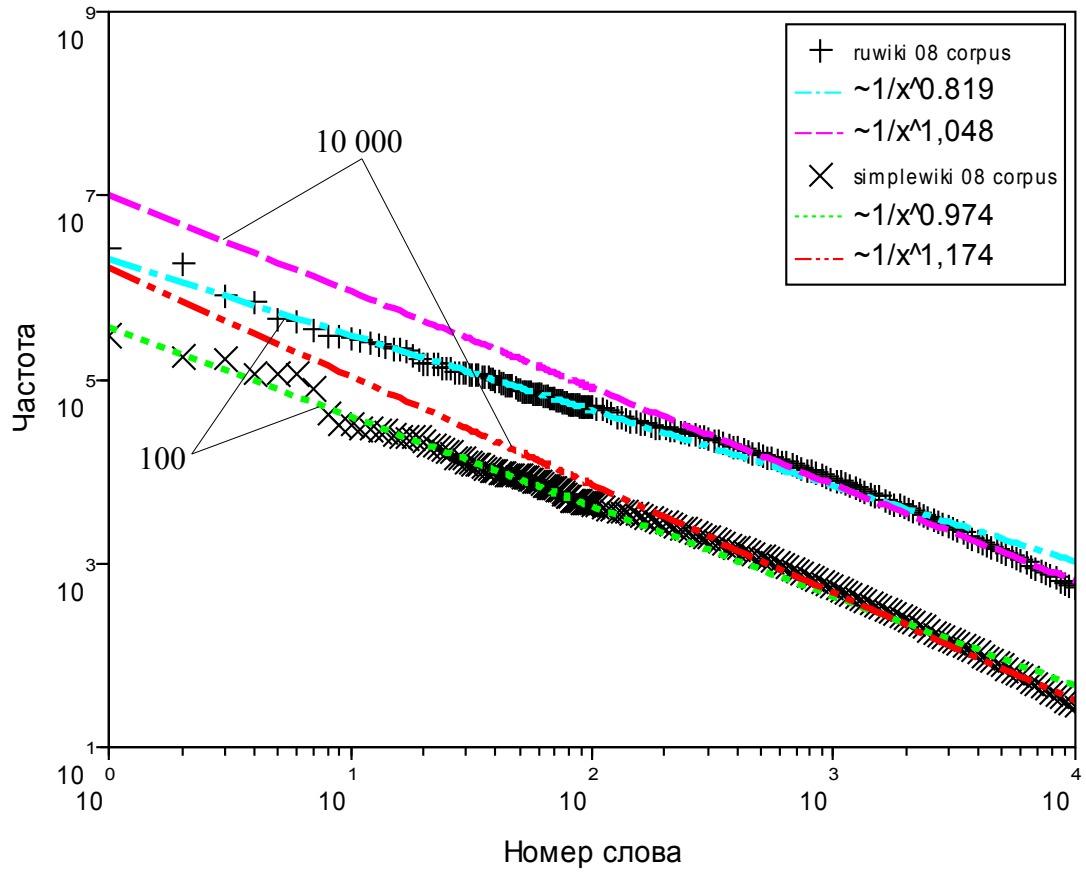

Рис. 3. Линейная зависимость убывания частоты употребления слов в корпусе от порядкового номера (ранга) слова в списке слов, упорядоченных по частоте, в масштабе логарифм-логарифм для Русской Википедии (ruwiki) и Simple Wikipedia (simplewiki) на февраль 2008 г., линейная аппроксимация по 100 и 10 000 наиболее частотных слов

На рис. 4 представлено распределение частоты слов в текстах двух википедий *в два момента времени*, то есть на рисунке приведены данные для тех же четырёх корпусов (индексных БД), что представлены в табл. 3. Перечислим значения пяти кривых (сверху вниз) на рис. 4:

- «*ruwiki 08 corpus*» (линия красного цвета, пунктир) — частота слов в корпусе Русской Википедии на 20.02.2008;
- «*07 corpus*» (фиолетовый цвет, пунктир: две точки, один штрих) — частота слов в корпусе Русской Википедии на 20.09.2007;
- «*ruwiki 07 doc*» (серый цвет, широкая полоса) — число документов, содержащих лексемы тех же слов (на абсциссе «Номер слова»), для которых на графике «*07 corpus*» указаны их частоты в корпусе (Русская Википедия, 20.09.2007);[50]
- «*simplewiki 08 corpus*» (фиолетовый цвет, непрерывная линия) — частота слов в корпусе Simple Wikipedia на 14.02.2008;
- «*simplewiki 07 corpus*» (зелёный цвет, пунктир) — частота слов в корпусе Simple Wikipedia на 09.09.2007.

50 Можно построить аналогичный график, если отсортировать слова не по частоте слов в корпусе, а по числу документов, содержащих слова. В этом случае взаимно поменяется характер кривых «07 corpus» и «ruwiki 07 doc».

График на рис. 4 внизу тот же, что и вверху, за исключением того, что прологарифмирована не только частота слов в корпусе и число документов (с этим же словом), но также и число слов (ранг в списке). Графики показывают, что характер зависимостей (и выполнение закона Ципфа) сохранился за истекшие полгода в обеих википедиях.

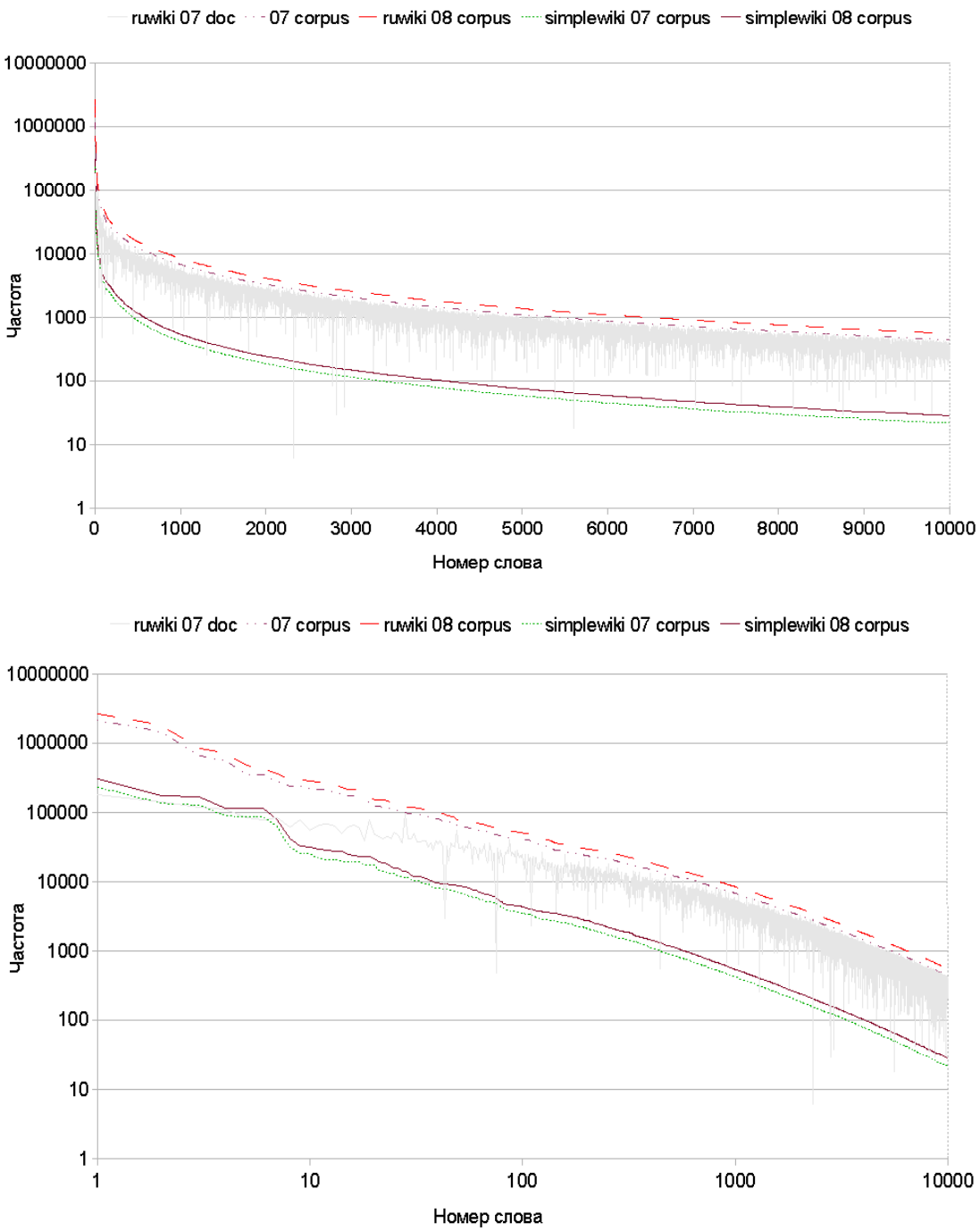

Рис. 4. Распределение частоты десяти тысяч наиболее употребимых слов в вики-корпусах на русском (ruwiki) и английском упрощённом (simplewiki) языках за 2007 и 2008 гг. (вверху); проверка выполнения закона Ципфа для данных корпусов (внизу)

Предложим читателю ещё ряд экспериментов, которые можно выполнить, пользуясь данными индексной БД:

а) взять первую 1000 слов по частоте в корпусе, упорядочить по числу документов, построить график;

б) найти число слов с частотой 1, 2, ... 10 ... 1000 слов в корпусе (привести в таблице и

построить гистограмму);

в) найти число слов длиной 1 буква, 2, 3.. 30 (таблица и гистограмма);

г) сравнить изменение ранга слов (по популярности, то есть частоте употребления) в корпусе со временем (слово, ранг, повышение/понижение — на сколько), (подсказка: нужно построить две индексных БД для разных по времени дампов википедий); указать часто употребимые слова, ранг которых изменился максимально;

д) найти число различных (уникальных) лексем в документе (вики-странице); среднее и максимальное число по всем документам; то же, но нормированное на число слов в документе; вычислить список первых десяти самых «пёстрых» документов, то есть богатых своим лексиконом, а именно: содержащих максимальное значение отношения числа уникальных лексем к числу слов в документе; обратная задача: построить упорядоченный список самых «нудных», т.е. длинных, но бедных лексиконом документов.

ОБСУЖДЕНИЕ

Недостатки реализации.

- индекс создаётся единожды, при обновлении корпуса текстов его нужно перестраивать полностью;
- при значении переменной doc_freq_max[51] равной 100 (а не 1000, например), короткие статьи имеют мало вхождений в таблицу term;
- одна словоформа может иметь несколько лексем в базе системы Lemmatizer (лексемы имеют разные ID), чтобы сохранить эту информацию в БД WikIDF, можно добавить ещё одну таблицу[52].

Выводы и предложения:

- поскольку *tf-idf* вес указывает на значимость слова во всём корпусе текстов, постольку вес слова, например, «байт» будет, скорее всего, мал в корпусе текстов о программировании и значителен – в корпусе текстов о биологии. Нужно продумать использование категорий для уточнения значения веса[53]. Таким образом, вес будет зависеть от проблемной области вики-текстов и у одного и того же слова будет разным в текстах разной тематики;
- полезным дополнительным ресурсом для индексирования будет размеченный корпус Английской Википедии [Atserias08];[54]
- добавить в таблицу *term* (рис. 2) поле «коэффициент вариации D», то есть оценку специфичности слова для отдельной предметной области ([LyashevskayaSharoff08], стр. 347).

Существуют альтернативные способы построения индексной БД, к которым можно было бы обратиться в дальнейшей работе:

1) Модуль ANNIC системы GATE[55], основанный на поисковом движке Lucene [Aswani05];
2) Индексирующая система MG4J [56] [BoldiVigna07];
3) Инструментарий Lemur с возможностями индексирования (английский, китайский, арабский языки) и поисковиком INDRI.[57]

51 doc_freq_max variable is the limit for the table *term_page*. If the number of documents which contain a lemma (e.g. lemma «website») > doc_freq_max, then the table *term_page* (fields *term_id* and *page_id*) contains only ID of first doc_freq_max documents which contain the term.

52 Например, добавить таблицу *meaning* (связанную с таблицей *term*), хранящую номер значения слова (возможно из Викисловаря) и/или ID леммы из базы Lemmatizer.

53 Возможно, с помощью некоторого коэффициента, уменьшающего вес тех слов, которые часто употребляются в данной проблемной области.

54 См. http://www.yr-bcn.es/semanticWikipedia.

55 См. также в [Witte07] описание варианта работы с вики из системы GATE.

56 См. http://mg4j.dsi.unimi.it.

57 См. http://www.lemurproject.org.

ПЛАНЫ НА БУДУЩЕЕ

Перспективные направления исследований, связанные с индексной БД таковы:

1) Динамическая индексация;
2) *Определение значения многозначных слов* (WSD). В работе [Magnini02] предположили, что метки предметных областей (например, *мед.*, *архит.*, *спорт.*) позволят находить семантические отношения между значениями слов. Результаты экспериментов показали, что значения многозначных слов действительно определяются с высокой степенью точности для большого количества слов благодаря данным меткам [Magnini02]. Для экспериментов использовали такой ресурс, как WordNet Domain[58] (расширенная версия WordNet), где каждый синсет содержал метки предметных областей. По-видимому, формирование индексных БД проблемно-ориентированных корпусов на основе категорий вики, предложенное в данной работе, является конкурентным вспомогательным подходом к решению задачи WSD, поскольку высокое значение частоты термина в такой БД указывает на его принадлежность к этой предметной области;
3) Фильтрация потока текстов (описание подхода см. ниже или в работе [Smirnov08]);
4) Визуализация результатов поиска;
5) Поиск ключевых слов с учётом семантических отношений (синонимы, гипонимы и др.) например, в [Shamsfard06] (на основе WordNet или CYC) или Викисловаря; [59]
6) Оценка точности поиска методом TF-IDF, определение оптимальных поисковых параметров за счёт (i) отсечения высокочастотных слов [Meyer07], (ii) сравнения мер сходства в векторном пространстве слов [Meyer07]; (iii) учёта аддитивной модели расчёта релевантности документа запросу [Gulin06].

Визуализация

Алгоритм визуализации результатов поиска с помощью индексной БД вики-текстов.

1) Выбрать список термов из индексной БД, которые имеют максимальный вес TF (по схеме TF-IDF) для данного вики-текста – это аналог ключевых слов;
2) выполнить шаг (1) для каждого вики-текста данной проблемной области, получить множество ключевых слов для каждого текста;
3) применить полученные ключевые слова (данной проблемной области) для фильтрация текстов, то есть выделения из множества текстов тех, которые интересны.

Предлагается выполнить визуализацию результатов, полученных на шаге (2). Нарисовать граф, связывающий документы через их общие ключевые слова.[60] Воплощение этой идеи в виде интерактивного графического приложения позволит показать:

- текст статьи, для выбранной вершины статьи;
- статьи, содержащие данное ключевое слово;
- ключевые слова, принадлежащие данной статье.

Для реализации данной функциональности предлагается использовать исходный код приложения Synarcher [Krizhanovsky2006a]. Необходимы дополнительные исследования для того, чтобы детально разработать алгоритм шага (3).

Методология фильтрации текстов с помощью данных индексной базы Википедии

Подход фильтрации потока текстов (information filtering) на основе индексной базы данных Википедии (WP) состоит из четырёх этапов (рис. 5). Первый этап — это индексирование

58 См. http://wndomains.itc.it

59 См. также в [Muller07] описание расширенной булевой модели на основе Lucene, в которой расширение запроса выполняется с помощью синонимов и гипонимов GermaNet.

60 В данном случае можно было бы нарисовать вершины двух типов – статьи (отображаются их названия) и ключевые слова. Значение дуг, связывающих слова и документы – очевидно.

вики-текстов («*WP Indexing*»). Поток текстов может представлять собой почтовые сообщения, тогда фильтрация заключается в обнаружении спама [Cormack07]. Если это новостной поток, то тексты новостей могут классифицироваться по нескольким темам (topics), заданным пользователем.

Создание профиля (Profile Generation). Итак, с одной стороны, построена индексная БД ВП («*1. Indexing*»), с другой стороны, пользователь предоставил корпус документов, отражающих неявно одну или несколько тем, его интересующих. Необходимо:

1. Построить *список лемм* (*термов*) для каждого документа пользователя (с помощью GATE [Cunningham2005] и Lemmatizer [Sokirko2004]);
2. Найти множество *статей ВП* близких этим термам, например с помощью алгоритма ESA [Gabrilovich2007].
3. Получить *список категорий* для каждой статьи ВП с помощью Synarcher API [Krizhanovsky2006a].

Теперь необходимо выполнить кластеризацию («*4. Clustering*») этого (возможно огромного) списка категорий с тем, чтобы получить несколько наборов категорий, при этом каждый набор соответствует одной из тем, интересующей пользователя. Категории в одном наборе не должны включать друг друга, то есть являться родителями друг друга (вероятно, в одном наборе будут категории одного уровня). На этом этапе пользователь может уточнить эти множества категорий, которые и составляют его профиль.

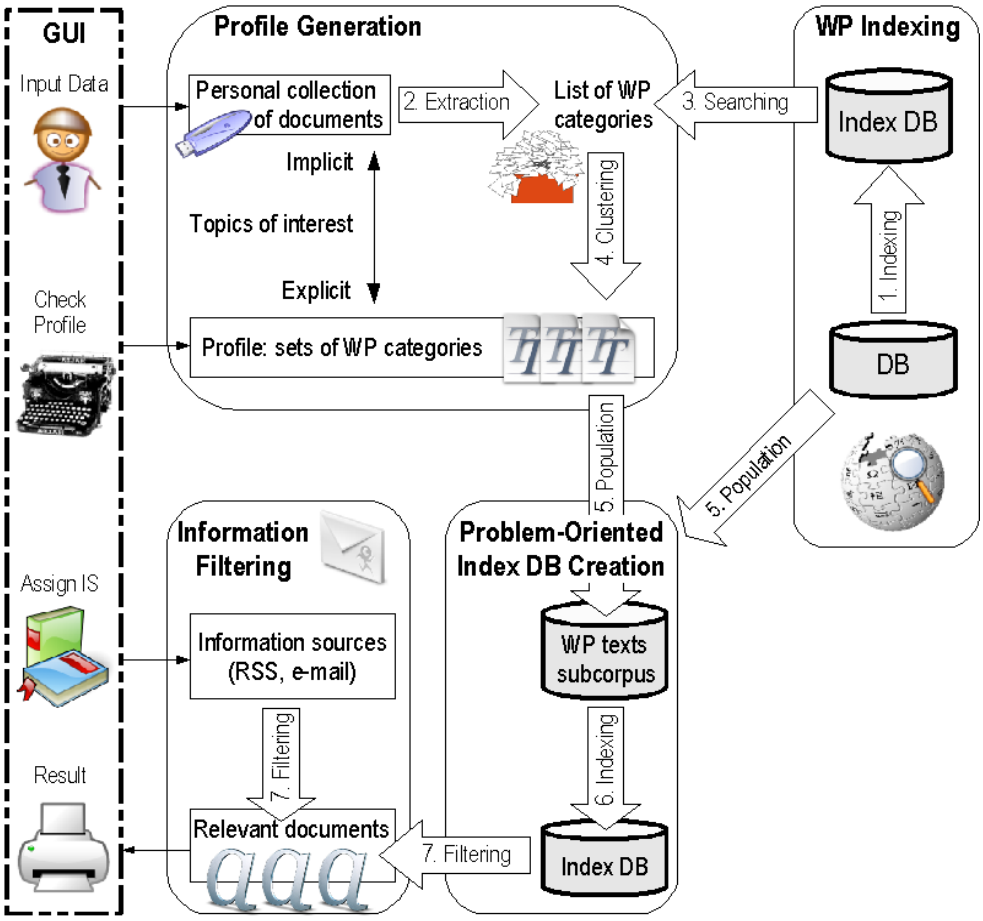

Рис. 5. Подход к фильтрации информации на основе индексной базы данных (DB) Википедии (WP)

Создание проблемно-ориентированной индексной БД (Problem-oriented Index DB Creation). Для каждого такого множества категорий извлекаются тексты БД ВП тех статей, которые принадлежат данным категориям или их подкатегориям («*5. Population*»). Далее полученный проблемно-ориентированный подкорпус текстов ВП индексируется, то есть строится

проблемно-ориентированная индексная БД («*6. Indexing*»).[61]

Фильтрация информации (Information filtering). Пользователь выбирает список источников текстовой информации и указывает для каждого источника — к какой проблемной области он относится. При появлении в них новых текстов необходимо:

1. Построить *список терминов (лемм*) для данного текста.
2. Выполнить поиск этих терминов в соответствующей проблемно-ориентированной индексной БД ВП, получить из БД вес этих терминов, подсчитать общий вес документа.
3. Представить пользователю список текстов, упорядоченный по весу.

Таким образом, в данном подходе индексная БД вики-текстов строится несколько раз:

- для всей Википедии («*1. Indexing»*);
- для проблемно-ориентированного подкорпуса ВП («*6. Indexing»*);

С помощью системы индексирования вики-текстов строятся списки лемм:

- для личной коллекции документов пользователя («*2. Extraction»*);
- для новостного потока («*7. Filtering»*).

О РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ И ФИЛЬТРАЦИИ ТЕКСТОВ

Автоматическая фильтрация текстов предназначена облегчить работу пользователя, сделать более результативным постоянный поиск информации. Программный модуль фильтрации текстов — важная составляющая «рекомендательных» систем (recommender system) и систем фильтрации информации (information filtering system).

Рекомендующая система пытается предоставить интересные пользователю данные, то есть информацию, соответствующую его кругу интересов и предпочтений. Рекомендующие системы (системы фильтрации информации) делят на системы, анализирующие содержание искомых документов (content-based) (данная работа) и системы, использующие явные или неявные предпочтения многих пользователей (collaborative filtering system) [Herrera-Viedma04].

В работе рекомендующих систем используют как тезаурусы (например, WordNet [Magnini01]), так и онтологии [Middleton02]. Сравнение результатов поиска семантически близких слов на основе данных WordNet [Banerjee02], [Fellbaum1998], [Lin98], [Resnik95], [Resnik99], [WuPalmer94], GermaNet [Muller2007] и Английской Википедии (ВП) [Strube2006] в работах [Krizhanovsky07full], [Calderan2006], [Strube2006] показывает преимущество последней. Википедия превосходит WordNet в том, что:

- больший размер (WordNet 3.0 содержит 155 тыс слов и 118 тыс. синсетов, Английская же Википедия содержит 946 млн слов[62] и 176 тыс (на сент. 2006) категорий[63],). В тоже время большой размер является трудностью для реализации поиска. Поэтому популярной для экспериментов является ВП на упрощённом английском поскольку она меньше по размеру на два порядка (см., например, эксперименты в [RuizCasado2005]).
- глоссы WordNet (несколько предложений, описывающих синсет) дают меньше информации о концепте, чем статья в несколько страниц в ВП. Это важно при полнотекстовом анализе, например, результаты поиска в ВП [Strube2006] превзошли результаты адаптированного алгоритма Lesk на основе анализа глосс [Banerjee02]. В работе [Calderan2006] предложен способ искусственного увеличения размера глосс в WordNet (метод «extended gloss overlaps»), чтобы справиться с проблемой краткости глосс.
- Википедии представлены на многих языках, не только на английском.

---

61 См. работу [GaoyingCui08] о построении проблемно-ориентированного подкорпуса Википедии на основе ссылок.

62 См. http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons (от 27 янв. 2008)

63 См. http://stats.wikimedia.org/EN/TablesWikipediaEN.htm (от 30 окт. 2006)

Категории Википедии обладают и достоинствами и недостатками. Ценность представляют не сами категории, а их взаимосвязь, образованная ими иерархическая структура, а также привязка статей к категориям. Дерево категорий ВП — это пример системы тегов, выработанной совместными усилиями (folksonomy). В работе [Milne2006] отмечают, что одним из основных источников информации в программных проектах, связанных с Википедией, являются категории. Но даже этот источник данных не является однозначно определённым, поскольку категории могут представлять не только родовидовые отношения (гиперонимия, *is-a*), но также отношения часть-целое (меронимия), наличие свойств (*has-property*) [Strube2006]. Поэтому не случайно есть работы, например [RuizCasado2005], по интеграции данных Википедии и тезауруса WordNet, чётко указывающим тип отношения между синсетами.[64]

ЗАКЛЮЧЕНИЕ

Растёт не только Интернет, подрастает и Википедия, причём по последним данным [Geser07] энциклопедия увеличивается и совершенствуется в шестимерном пространстве, а именно растёт:

1. число языков, на которых излагается материал ВП;
2. число активных участников;[65]
3. число тематических направлений (у каждой новой группы участников, у каждой языковой группы — свои интересы);
4. число статей в целом; а в «больших» Википедиях глубина проработки (формально — это размер статьи, число правок);
5. связность страниц (то есть число внутренних ссылок, интервики, категории);
6. «погружение» Википедии в паутину Веб за счёт увеличения числа внешних ссылок.

Поисковые системы и вики-ресурсы всё более тесно кооперируются.

С одной стороны, благодаря насыщенности вики-текстов гиперссылками и в виду особенностей алгоритмов, основанных на анализе ссылок (например, PageRank [Brin1998]), поисковики награждают вики-тексты высоким рейтингом, то есть первыми позициями в результатах поиска [Rainie07]. С другой стороны, поиск внутри вики-сайтов осуществляется как с помощью встроенного в MediaWiki поиска по БД, так и с помощью специализированных систем поиска в Википедии[66]: Wikia Search[67], Lucene-search[68], FUTEF[69], а также в Русской Википедии: Qwika[70]. Отметим, что будущее поисковых систем, возможно, зиждется на распределённом поиске с помощью P2P приложений [Wu07]. Важной задачей поисковиков является индексирование.

В статье рассмотрена архитектура и реализация программной системы индексирования вики-текстов WikIDF. При индексировании вычисляются список лемм и их частота встречаемости в вики-статье с помощью системы GATE, морфологического анализатора Lemmatizer и объединяющего их модуля RussianPOSTagger. Данная работа не является первой, применяющей модуль Lemmatizer к текстам Википедии.[71] С помощью системы WikIDF построены индексные базы данных для Русской Википедии и Википедии на английском упрощённом языке.

---

64 В работе [RuizCasado2005] предложен подход автоматического установления соответствия статей этой энциклопедии и синсетов WordNet.

65 Со временем число участников растёт, но относительное число высоко активных участников (больше 100 правок в месяц) снижается [Geser07] (стр. 18).

66 См. большой список поисковиков для поиска в Википедии: http://en.wikipedia.org/wiki/Wikipedia:Searching.

67 Поисковик Wikia Search. См. http://search.wikia.com/. См. также описание доступа к индексной базе поисковика http://search.wikia.com/wiki/Tech/Open_Index.

68 См. http://ls2.wikimedia.org и http://clucened.com.

69 См. http://futef.com.

70 См. http://www.qwika.com/find-ru/.

71 Методика построения индексной базы. 23.10.2006. http://ru.wikipedia.org/wiki/Википедия:Частотный_словник

Планируемые варианты развития системы WikIDF таковы. Во-первых, включение WikIDF (как одного из компонент) в программный комплекс Synarcher[72] для выполнения полнотекстового (а не только по заголовкам вики-страниц) поиска семантически близких слов. Таким образом, с помощью полнотекстового поиска в вики-текстах будет возможна генерация корневого набора страниц в адаптированном HITS алгоритме (AHITS) [Krizhanovsky2006a], что, по-видимому, улучшит результат поиска семантически близких слов. Вторая (не связанная с первой) задача, заключается в преобразовании в машинную форму Викисловаря, в первую очередь семантических отношений Викисловаря, что также можно использовать для улучшения поиска в вики-текстах с помощью WikIDF.

В статье представлены параметры исходных баз данных двух википедий: Русская Википедия и Simple English Wikipedia (на английском упрощённом языке). Приведены временные характеристики индексирования данных баз, а также описаны количественные свойства построенных индексных баз. Обнаружен более быстрый рост англоязычной Википедии, а именно: за пять месяцев (сент. 2007 — февр. 2008) скорость роста числа статей была больше на 12% и на 6% быстрее чем в русской пополнялся лексикон Википедии на английском упрощённом языке.

Эксперименты показали, что закон Ципфа в целом выполняется для текстов обеих википедий. Сохраняется линейный характер зависимости убывания частоты слов от ранга в частотном списке слов (в масштабе логарифм-логарифм) за истекшие полгода в обеих википедиях.

Наличие доступного приложения для построения индексной базы вики-текстов имеет большое практическое значение. Данное приложение является необходимой частью для создания такой полнотекстовой поисковой системы, которая выполняет поиск в том числе и по вики-текстам.

БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке РФФИ (проект № 08-07-00264), Президиума РАН (проект № 14.2.35) и ОИТВС РАН (проект № 1.9).

Список источников литературы

[Vahitova06]. Вахитова Д. Создание корпуса текстов по корпусной лингвистике. 2006. http://matling.spb.ru/files/kurs/Vahitova_Corpus.doc

[Gulin06]. Гулин А., Маслов М., Сегалович И. Алгоритм текстового ранжирования Яндекса на РОМИП-2006 // Труды РОМИП'2006, октябрь, 2006 . http://download.yandex.ru/company/03_yandex.pdf

[Krizhanovsky07full]. Крижановский А.А. Оценка результатов поиска семантически близких слов в Википедии: Information Content и адаптированный HITS алгоритм // Вики-конференция 2007. Тезисы докладов. Санкт-Петербург, 27-28 октября 2007 . http://arxiv.org/abs/0710.0169

[Krizhanovsky08avia]. Крижановский А.А. Автоматизированный поиск семантически близких слов на примере авиационной терминологии // *Автоматизация в промышленности*. – 2008. – Т.**4** (64). – С. 16-20. http://whinger.narod.ru/paper/avia_terms_search_by_AHITS_08.pdf

[Lande07]. Ландэ Д.В., Григорьев А.Н., Дармохвал А.Т. Взаимосвязь понятий в документах - совместное появление или контекстная близость? // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог 2007». Бекасово, 2007 . http://www.dialog-21.ru/dialog2007/materials/pdf/LandeD1.pdf

[LyashevskayaSharoff08]. Ляшевская О.Н., Шаров С.А. Частотный словарь национального корпуса русского языка: концепция и технология создания // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог 2008». Бекасово, 2008. – С. 345-351. http://www.dialog-21.ru/dialog2008/materials/pdf/53.pdf

[Segalovich2002]. Сегалович И.В. Как работают поисковые системы. http://www.dialog-21.ru/trends/?id=15539&f=1

72 Пакет WikIDF входит в программу Synarcher, однако поиск семантически близких слов в данной версии Synarcher выполняется без обращения к пакету WikIDF либо индексным базам данных.

[Smirnov08b]. Смирнов А.В., Крижановский А.А. Экспериментальная оценка построенных индексных баз данных Википедии (to appear) // Международная конференция AIS/CAD '08. Дивноморское, 3 - 10 сентября 2008 .
[Sokirko2004]. Сокирко А.В. Морфологические модули на сайте www.aot.ru // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции «Диалог 2004». «Верхневолжский», 2004. – С. 559-564. http://www.aot.ru/docs/sokirko/Dialog2004.htm
[Friedl2001]. Фридл Дж. Регулярные выражения. Библиотека программиста. СПб.: Питер, 2001. – 352 с. – ISBN 5-272-00331-4.
[Aswani05]. Aswani N., Tablan V., Bontcheva K., Cunningham H. Indexing and querying linguistic metadata and document content. In *Proceedings of Fifth International Conference on Recent Advances in Natural Language Processing (RANLP2005).* Bulgaria, Borovets, 2005. http://gate.ac.uk/sale/ranlp05/ranlp05-annic.pdf
[Atserias08]. Atserias J., Zaragoza H., Ciaramita M., Attardi G. Semantically annotated snapshot of the English Wikipedia. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).* Morocco, Marrakech, May 26 – June 1, 2008. http://www.lrec-conf.org/proceedings/lrec2008/pdf/581_paper.pdf
[Banerjee02]. Banerjee S., Pedersen T. An Adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02).* Mexico City, February, 2002. http://www.d.umn.edu/~tpederse/Pubs/cicling2002-b.pdf
[Bharat1998]. Bharat K., Henzinger M. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 98)*, 1998. – pp. 104-111 ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf
[Blondel2004]. Blondel V., Gajardo A., Heymans M., Senellart P., Dooren P. A measure of similarity between graph vertices: applications to synonym extraction and web searching. – SIAM Review, 2004. – Vol. 46, No. 4, pp. 647-666. http://www.inma.ucl.ac.be/~blondel/publications/areas.html
[Blondel2002]. Blondel V., Senellart P. Automatic extraction of synonyms in a dictionary. In *Proceedings of the SIAM Workshop on Text Mining.* Arlington (Texas, USA), 2002. http://www.inma.ucl.ac.be/~blondel/publications/areas.html
[BoldiVigna07]. Boldi P., Vigna S. Efficient optimally lazy algorithms for minimal-interval semantics. 2007. http://vigna.dsi.unimi.it/papers.php
[Brin1998]. Brin S., Page L. The anatomy of a large-scale hypertextual Web search engine. 1998. http://www-db.stanford.edu/~backrub/google.html
[Calderan2006]. Calderan M. Semantic Similarity Library, Technical Report #DIT-06-036, University of Trento, 2006. http://www.dit.unitn.it/~accord/Publications/Semantic%20Similarity%20Library%20Thesis%20Proposal.pdf
[Campbell06]. Campbell S., Chancelier J.-P., Nikoukhah R. Modeling and Simulation in Scilab/Scicos. – Springer, 2006. – 313 pp. – ISBN 978-0-387-27802-5.
[Muller07]. Christof M., Gurevych I., Muhlhauser M. Integrating semantic knowledge into text similarity and information retrieval. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC). (to appear).* Irvine, CA, 2007. http://proffs.tk.informatik.tu-darmstadt.de/TK/abstracts.php3?lang=en&paperID=575
[Codd90]. Codd E. F. The relational model for database management: version 2. – Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990. – 567 pp. – ISBN 0-201-14192-2 . http://portal.acm.org/toc.cfm?id=SERIES11430&type=series&coll=ACM&dl=ACM
[Cormack07]. Cormack G. V., Hidalgo J. G., Sanz E. P. Spam filtering for short messages. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007).* Lisbon, Portugal, November, 2007. http://plg.uwaterloo.ca/~gvcormac/cikm07.pdf
[Cunningham2005]. Cunningham H., Maynard D., Bontcheva K., Tablan V., Ursu C., Dimitrov M., Dowman M., Aswani N., Roberts I. Developing language processing components with GATE (user's guide), Technical report, University of Sheffield, U.K., 2005. http://www.gate.ac.uk
[Fellbaum1998]. Fellbaum C. WordNet: an electronic lexical database. – MIT Press, Cambridge, Massachusetts, 1998. – 423 pp. – ISBN 0-262-06197-X.
[Ferragina2008]. Ferragina P. String algorithms and data structures. http://arxiv.org/abs/0801.2378
[Fortunato2005]. Fortunato S., Boguna M., Flammini A., Menczer F. How to make the top ten: Approximating PageRank from in-degree. 2005. http://arxiv.org/abs/cs/0511016
[Gabrilovich2007]. Gabrilovich E., Markovitch S. Computing semantic relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI).* Hyderabad, India, January, 2007. http://www.cs.technion.ac.il/~gabr/papers/ijcai-2007-sim.pdf
[GaoyingCui08]. Gaoying Cui, Qin Lu, Wenjie Li, Yirong Chen. Corpus exploitation from Wikipedia for ontology construction. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).* Morocco, Marrakech, May 26 – June 1, 2008. http://www.lrec-conf.org/proceedings/lrec2008/pdf/541_paper.pdf
[Geser07]. Geser H. From printed to "wikified" encyclopedias. Sociological Aspects of an incipient cultural revolution. In: Sociology in Switzerland: Towards Cybersociety and Virtual Social Relations. Zuerich, June 2007. http://socio.ch/intcom/t_hgeser16.pdf

[Herrera-Viedma04]. Herrera-Viedma E., Porcel C., Lopez A.G., Olvera M.D., Anaya K. A fuzzy linguistic multi-agent model for information gathering on the web based on collaborative filtering techniques. In *Atlantic Web Intelligence Conference, AWIC'04. Lect. Notes in Artificial Intelligence.* Cancun, Mexico, 2004. – pp. 3-12 http://decsai.ugr.es/%7Eviedma/papers/Awic2004.zip

[Kleinberg1999]. Kleinberg J. Authoritative sources in a hyperlinked environment. – Journal of the ACM, 1999. – Vol. 5, No. 46, pp. 604-632. http://www.cs.cornell.edu/home/kleinber

[Krizhanovsky2006a]. Krizhanovsky A. Synonym search in Wikipedia: Synarcher. In *11-th International Conference "Speech and Computer" SPECOM'2006.* Russia, St. Petersburg, June 25-29, 2006. – pp. 474-477 http://arxiv.org/abs/cs/0606097

[Krizhanovsky08b]. Krizhanovsky A. Experiments on corpus index generated from Wikipedia (to appear). In *Corpus Linguistics – 2008.* Russia, St. Petersburg, October 6–10, 2008.

[Lemaire06]. Lemaire B., Denhiere G. Effects of high-order co-occurrences on word semantic similarities. – Current Psychology Letters - Behaviour, Brain and Cognition, 2006. – Vol. 18, No. 1. http://arxiv.org/abs/0804.0143

[Lin98]. Lin D. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning,* Madison, Wisconsin, July, 1998. http://www.cs.ualberta.ca/~lindek/papers.htm

[Magnini01]. Magnini B., Strapparava C. Using WordNet to improve user modelling in a web document recommender system. In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources.* Pittsburgh, June, 2001. http://multiwordnet.itc.it/paper/WordnetWumNAACL.pdf

[Magnini02]. Magnini B., Strapparava C., Pezzulo G., Gliozzo A. The role of domain information in word sense disambiguation. – Journal of Natural Language Engineering, 2002. – Vol. 8, No. 4, pp. 359-373. http://www.istc.cnr.it/doc/1a_16p_Magnini-NLE-2002.pdf

[Maguitman2005]. Maguitman A. G., Menczer F., Roinestad H., Vespignani A. Algorithmic Detection of Semantic Similarity. 2005. http://www2005.org/cdrom/contents.htm

[Manning99]. Manning C. D., Schutze H. Foundations of Statistical Natural Language Processing. – The MIT Press, 1999. – 620 pp. – ISBN 0262133601. http://nlp.stanford.edu/fsnlp/

[Melnik2002]. Melnik S., Garcia-Molina H., Rahm E. Similarity flooding: a Versatile graph matching algorithm and its application to schema matching. In *18th ICDE.* San Jose CA, 2002. http://research.microsoft.com/~melnik/publications.html

[Meyer07]. Meyer M., Rensing C., Steinmetz R. Categorizing learning objects based on Wikipedia as substitute corpus. In *First International Workshop on Learning Object Discovery & Exchange (LODE'07).* Greece, Crete, September 18, 2007. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-311/paper09.pdf

[Middleton02]. Middleton S. E., Alani H., Shadbolt N. R., Roure D. C. D. Exploiting synergy between ontologies and recommender systems. In *Semantic Web Workshop 2002 At the Eleventh International World Wide Web Conference.* Hawaii, USA, 2002. http://eprints.ecs.soton.ac.uk/6487/1/www-paper.pdf

[Milne07]. Milne D. Computing Semantic Relatedness using Wikipedia Link Structure. // New Zealand Computer Science Research Student Conference (NZCSRSC'2007). Hamilton, New Zealand. . http://www.cs.waikato.ac.nz/~dnk2/publications/nzcsrsc07.pdf

[Milne2006]. Milne D., Medelyan O., Witten I. H. Mining domain-specific thesauri from Wikipedia: a case study. In *Proc. of the International Conference on Web Intelligence (IEEE/WIC/ACM WI'2006).* Hong Kong, 2006. http://www.cs.waikato.ac.nz/~olena/publications/milne_wikipedia_final.pdf

[MilneWitten07]. Milne D., Witten I.H., Nichols D.M. A knowledge-based search engine powered by Wikipedia. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM'2007).* Portugal, Lisbon, 2007. http://www.cs.waikato.ac.nz/~dnk2/publications/cikm07.pdf

[Muller2007]. Muller C., Gurevych I., Muhlhauser M. Integrating semantic knowledge into text similarity and information retrieval. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC)*, 2007. – pp. (to appear) http://proffs.tk.informatik.tu-darmstadt.de/TK/abstracts.php3?lang=en&paperID=575

[Ollivier2007]. Ollivier Y., Senellart P. Finding related pages using Green measures: an illustration with Wikipedia. In *Association for the Advancement of Artificial Intelligence.*Vancouver, Canada, 2007. http://pierre.senellart.com/publications/ollivier2006finding.pdf

[Pantel2000]. Pantel P., Lin D. Word-for-word glossing with contextually similar words. In *Proceedings of ANLP-NAACL 2000.* Seattle, Washington, May, 2000. – pp. 75-85 http://www.cs.ualberta.ca/~lindek/papers.htm

[Papadakos08]. Papadakos P., Vasiliadis G., Theoharis Y., Armenatzoglou N., Kopidaki S., Marketakis Y., Daskalakis M., Karamaroudis K., Linardakis G., Makrydakis G., Papathanasiou V., Sardis L., Tsialiamanis P., Troullinou G., Vandikas K., Velegrakis D., Tzitzikas Y. The anatomy of Mitos web search engine. http://arxiv.org/abs/0803.2220

[Pedersen08]. Pedersen T. Computational approaches to measuring the similarity of short contexts: a review of applications and methods. – South Asian Language Review (to appear), 2008. – Vol. 1, No. 1. http://arxiv.org/abs/0806.3787

[Ponzetto07]. Ponzetto S. P., Strube M. An API for measuring the relatedness of words in Wikipedia. In *Companion Volume to the Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics.* Prague, Czech Republic, 23-30 June, 2007. http://www.eml-research.de/english/homes/ponzetto/pubs/acl07.pdf

[Rainie07]. Rainie L., Tancer B. Wikipedia users // Reports: Online Activities & Pursuits. 2007. http://www.pewinternet.org/pdfs/PIP_Wikipedia07.pdf
[Resnik95]. Resnik P. Disambiguating noun groupings with respect to WordNet senses. In *Proceedings of the 3rd Workshop on Very Large Corpora.* MIT, June, 1995. http://xxx.lanl.gov/abs/cmp-lg/9511006
[Resnik99]. Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. – Journal of Artificial Intelligence Research (JAIR), 1999. – Vol. 11, No. , pp. 95-130. http://www.cs.washington.edu/research/jair/abstracts/resnik99a.html
[Robertson2004]. Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF. – Journal of Documentation, 2004. – Vol. 60, No. , pp. 503-520. http://www.soi.city.ac.uk/~ser/idfpapers/Robertson_idf_JDoc.pdf
[RuizCasado2005]. Ruiz-Casado M., Alfonseca E., Castells P. Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets. 2005. http://arantxa.ii.uam.es/~castells/publications/awic05.pdf
[Sahami06]. Sahami M., Heilman T. D. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, 2006. http://robotics.stanford.edu/users/sahami/papers-dir/www2006.pdf
[Shamsfard06]. Shamsfard M., Nematzadeh A., Motiee S. ORank: an ontology based system for ranking documents. – International Journal of Computer Science, 2006. – Vol. 1, No. 3, pp. 225-231. http://www.waset.org/ijcs/v1/v1-3-30.pdf
[Smirnov08]. Smirnov A., Krizhanovsky A. Information filtering based on wiki index database. In *Proceedings of the 8th International FLINS Conference on Computational Intelligence in Decision and Control.* Spain, Madrid, September 21 – 24, 2008. http://arxiv.org/abs/0804.2354
[Strube2006]. Strube M., Ponzetto S. WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 06)*. Boston, Mass., July 16-20, 2006. http://www.eml-research.de/english/research/nlp/publications.php
[Berry2003]. Survey of text mining: clustering, classification, and retrieval, M. Berry (Ed.). – Springer-Verlag, New York, 2003. – 244 pp. – ISBN 0-387-955631.
[Witte07]. Witte R., Gitzinger T. Connecting wikis and natural language processing systems. In *WikiSym'07.* Canada, Quebec, October 21–23, 2007. http://www.wikisym.org/ws2007/_publish/Witte_WikiSym2007_NaturalLanguageProcessing.pdf
[Wu07]. Wu L.-S., Akavipat R., Menczer F. 6S: P2P Web index collecting and sharing application. In *Proc. RIAO*, 2007. http://sixearch.org/paper/6S_P2P_Web-1.pdf
[WuPalmer94]. Wu Z., Palmer M. Verb semantics and lexical selection. In *Proc. of ACL-94*, 1994. – pp. 133-138 http://acl.ldc.upenn.edu/P/P94/P94-1019.pdf
[Zesch08]. Zesch T., Mueller C., Gurevych I. Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC).* Morocco, Marrakech, May 26 – June 1, 2008. http://elara.tk.informatik.tu-darmstadt.de/publications/2008/lrec08_camera_ready.pdf

ПРИЛОЖЕНИЕ 1: ПРИМЕР РАБОТЫ API ИНДЕКСНОЙ БД WIKIDF

В таблице 4 приведены данные, полученные с помощью функций работы с индексной базой данных[73]:

- *lemma* — леммы слов вики-страницы «Через_тернии_к_звёздам_(фильм)»; причём леммы тех словоформ, которые были распознаны системой Lemmatizer [Sokirko2004], записаны в индексную БД заглавными буквами, прочие словоформы – в том виде, как они встретились в тексте;
- *TF-IDF* — значение лемм TF-IDF, вычисленное по формуле (1) на стр. 5; *N* в формуле (число страниц в корпусе) равно 204 475;
- *TF* — частота лексемы на данной странице; отметим, что «Нийя» и «Нийи» не распознаны как словоформы одной лексемы, а «ИСФАРА», «Исфары» — распознаны;
- *DF* — число вики-страниц с данной лексемой.

Таблица 4

Список лемм статьи «Через_тернии_к_звёздам_(фильм)»[74], упорядоченных по значению TF-IDF

N	TF-IDF	lemma	DF	TF
1	46.14	**Нийя**	2	4
2	25.62	ТЕРНИЕ	40	3
3	23.07	Десса	2	2
4	23.07	Туранчокс	2	2
5	22.78	ЛЕБЕДЕВ	687	4
6	19.33	Ледогоров	13	2
7	17.62	ПЕРЕРАБОТАТЬ	575	3
8	16.93	ВИКТОРОВ	43	2
9	14.37	ЗВЕЗДОЛЕТ	155	2
10	12.63	ГИБНУТЬ	369	2
11	12.23	людьми-клонами	1	1
12	12.23	Дессу	1	1
13	11.54	**Нийи**	2	1
14	11.54	Ракан	2	1
15	10.84	**Гидрокостюмы**	**4**	1
16	10.84	Колотун	4	1
17	10.62	**лий**[75]	5	1
18	10.44	Лиелдидж	6	1
19	10.15	Семенцова	8	1
20	10.15	МЕТЕЛКИНА	8	1
21	10.03	**ИСФАРА**	9	1
22	9.39	Глан	17	1
23	9.34	Улдис	18	1
24	9.34	Торки	18	1
25	9.28	ПОСЕЛЯТЬ	19	1

73 См. API индексной базы данных вики на стр. 8.

74 Версия от 18 февраля 2008, см. http://ru.wikipedia.org/w/index.php?title=Через_тернии_к_звёздам_(фильм)&oldid=7484929.

75 Неясно, почему имя «Лий» оказалось написанным со строчной буквы.

26	9.09	ДРЕЙЕР	23	1
27	8.67	Рыбникова	35	1
28	8.54	АВОСЬ	40	1
29	8.47	СЕЛЕНА	43	1
30	8.47	Стриженов	43	1
31	8.28	Дворжецкий	52	1
32	8.26	ПУЗЫРЬКОВЫЙ	53	1
33	8.24	двухсерийный	54	1
34	8.17	ГУМАНОИДНЫЙ	58	1
35	8.07	НЕВЕСОМОСТЬ	64	1
36	8.07	НАЖИВА	64	1
37	8.07	СВЕРХЧЕЛОВЕЧЕСКИЙ	64	1
38	8.05	ЮНОНА	65	1
39	8.01	НИЛЬСКИЙ	68	1
40	8.01	Ясулович	68	1
41	7.98	МОНОПОЛИСТ	70	1
42	7.74	РЫБНИК	89	1
43	7.7	НОСИК	93	1
44	7.66	РАСТЯНУТЬ	96	1
45	7.64	КЛОНИРОВАНИЕ	98	1
46	7.61	Dolby	101	1
47	7.6	ПОДЕЛИТЬСЯ	102	1
48	7.5	ДИВЕРСАНТ	113	1
49	7.42	БУЛЫЧЕВ	123	1
50	7.38	ФАДЕЕВ	128	1
51	7.27	ЩЕРБАКОВ	142	1
52	7.14	КЛИМ	162	1
53	7.13	НЕМИНУЕМЫЙ	163	1
54	7.06	XXIII	176	1
55	7.03	ЗЕМЛЯНИН	181	1
56	6.76	НЕФТЕПЕРЕРАБАТЫВАЮЩИЙ	237	1
57	6.65	ВАЦЛАВ	264	1
58	6.65	СПЕЦЭФФЕКТ	264	1
59	6.4	ЛАЗАРЕВ	341	1
60	6.36	ПОГОНЯ	355	1
61	6.35	ВОСТОРГ	356	1
62	6.34	КАРЕЛЬСКИЙ	362	1
63	6.32	СОРВАТЬ	368	1
64	6.29	КИРА	378	1
65	6.29	ВЫБРОСИТЬ	380	1

66	6.21	ТИТР	409	1
67	6.14	ТАДЖИКИСТАН	441	1
68	6.13	Digital	445	1
69	5.96	ВЫРЕЗАТЬ	530	1
70	5.94	ОБНОВИТЬ	539	1
71	5.94	ДИНАМИКА	540	1
72	5.93	БИЗНЕСМЕН	542	1
73	5.87	ДОСТАВЛЯТЬ	580	1
74	5.86	ГЛЕБ	585	1
75	5.82	НАУЧНО-ФАНТАСТИЧЕСКИЙ	606	1
76	5.79	СЧЕСТЬ	625	1
77	5.69	ИДЕОЛОГИЧЕСКИЙ	692	1
78	5.68	СНИЖАТЬ	697	1
79	5.65	ДОРОЖКА	717	1
80	5.63	КАДРЫ	735	1
81	5.54	СООБРАЖЕНИЕ	801	1
82	5.5	ЗАБРОСИТЬ	837	1
83	5.49	ЛЮДМИЛА	844	1
84	5.38	ВСПОМИНАТЬ	941	1
85	5.33	МЕЛОДИЯ	986	1
86	5.33	ПРОНИКАТЬ	993	1
87	5.28	ВАДИМ	1037	1

Также с помощью функций работы с индексной базой данных получен список документов, содержащих словоформы лексем «чародей» и «волшебник», документы упорядочены по значению частоты терминов (*ΣTF*) (табл. 5):

- *ΣTF* — общая частота заданных двух лексем на данной странице;
- *page_title* — заголовок вики-страницы;
- *n_words* — общее число слов, составляющих страницу.

Таблица 5

Список вики-страниц, содержащих одновременно леммы «чародей» и «волшебник», упорядоченных по их общей частоте.[76]

N	ΣTF	page_title	n_words
1	24	Волшебная_страна_(Волкова)	11062
2	14	Heroes_of_Might_and_Magic_II	2788
3	9	Альбус_Дамблдор	1118
4	7	Гарри_Поттер_и_Принц-полукровка	905
5	4	Фэйст,_Раймонд	406
6	4	Понедельник_начинается_в_субботу	3291
7	4	Might_and_Magic_VII:_For_Blood_and_Honor	2834
8	3	Фауст,_Иоганн	2072

76 Для окончательной красоты нужно ещё отсортировать страницы по длине, самые короткие — вперёд.

9	3	Сухинов,_Сергей_Стефанович	205
10	3	Король_Артур	1118
11	3	Дом_Телванни	394
12	3	Гафт,_Валентин_Иосифович	829
13	2	Ырамась	127
14	2	Хеопс	1574
15	2	Светин,_Михаил_Семёнович	304
16	2	Дивайт_Фир	352
17	2	Волхвы	1505
18	2	Брукс,_Терренс_Дин	547
19	2	Анорач	335
20	2	Heroes_of_Might_and_Magic	1765
21	2	Everquest_II	2861

ПРИЛОЖЕНИЕ 2: АППРОКСИМАЦИЯ В ПАКЕТЕ SCILAB

Входными данными служит файл, содержащий упорядоченный список частот слов. Приведём пример первых пяти строчек файла `ruwiki_20080220_tenK.txt`:

```
2630654
1781831
825127
698571
465278
```

Далее приведён исходный код для построения линейных аппроксимаций методом наименьших квадратов и визуализации результатов в пакете Scilab [Campbell06]. Результат работы программы представлены на рис. 3.

```
// read numbers from file to arrays
N_max=10000

// Russian Wikipedia
f=mopen('C:\experiments\ruwiki_20080220_tenK.txt','r');
for i=1:N_max
    corpus_freq=mfscanf(f,'%d');
    RU(i)=corpus_freq;
end
mclose(f);

// Simple Wikipedia
f=mopen('C:\experiments\simplewiki_20080214_only_corpusfreq10K.csv','r');
for i=1:N_max
    corpus_freq=mfscanf(f,'%d');
    SW(i)=corpus_freq;
end
mclose(f);


function z=fun(p)
    // printf(' --- start z=fun: p=%d,%d,%d\n', p(1),p(2),p(3)); // debug
    for v=1:N_max
        z(v)=log(A(v))-p(1)*log(v)-p(2);
        // printf('A(v)=%d, result z(%d)=%d\n', A(v), v, z(v)); // debug
    end
endfunction


// main two lines
//p0=[0 0];
//[ff,p]=leastsq(fun,p0);
// it's commented because:


// parameters found by leastsq by 100 and 10000 data points
p_sw100=[-0.9740246 12.826292]
p_sw10K=[-1.1740404 14.290272]

p_ru100=[-0.8186653 14.507994]
p_ru10K=[-1.0484371 16.127217]


// pars is parameter: p_sw100 or p_sw10K, etc.
function z=fn_approx(pars,v)
        z=exp(pars(1)*log(v)+pars(2));
endfunction
```

```
  xbasc(); // clear the screen

  // The graphic of the function with the fitted parameters
  N2 = 100;
       // all points [1..N2-1] will be drawn,
       // several (5) points (thanks to log) within [N2..N_max] will be drawn
  t=[1:N2-1 logspace(log10(N2),log10(N_max),100)]

  plot2d(t,[SW(t)     fn_approx(p_sw100,t)'     fn_approx(p_sw10K,t)'     RU(t)
fn_approx(p_ru100,t)' fn_approx(p_ru10K,t)'],logflag="ll",axesflag=1)

  legends(['ruwiki    08    corpus';'~1/x^0.819';'~1/x^1,048';'simplewiki    08
corpus';'~1/x^0.974';'~1/x^1,174'], [-1,4,6,-2,3 5],opt="ur")

  a=gca();
  a.x_label.text="Word rank"; // Word rank     // Номер слова
  a.y_label.text="Frequency"; // Frequency     // Частота
  a.font_size=2;
  a.x_label.font_size=3;
  a.y_label.font_size=3;

  e=gce();    // legend


  // a.children
  // 1 - approx_ru_10K
  // 2 - approx_ru100
  // 3 - ruwiki
  // 4 - approx_sw_10K
  // 5 - approx_sw100
  // 6 - simplewiki


  // ruwiki
    ruwiki = a.children.children(3);
    ruwiki.thickness=1;
    ruwiki.mark_style=1;
    ruwiki.mark_mode="on";
    ruwiki.line_mode="off";
  e_ru = e.children(11);
  e_ru.font_size = 1;

  // approx_ru100
    approx_ru100 = a.children.children(2);
    approx_ru100.thickness=3;
    approx_ru100.foreground=4;
  e.children(9).font_size = 3;
  e.children(10).line_style = 4;
    approx_ru100.line_style = 4;

  // approx_ru_10K
    approx_ru_10K = a.children.children(1);
    approx_ru_10K.thickness=3;
  e.children(7).font_size = 3;
  e.children(8).line_style   = 2;   // polyline
    approx_ru_10K.line_style = 2;



  // simplewiki 08 corpus

  // simplewiki
    simplewiki = a.children.children(6);
    simplewiki.thickness=1;
    simplewiki.mark_style=2;
    simplewiki.mark_mode="on";
    simplewiki.line_mode="off";
  e_sw = e.children(5);
  e_sw.font_size = 1;
```

```
// approx_sw100
  approx_sw100 = a.children.children(5);
  approx_sw100.thickness=3;
  approx_sw100.foreground=3;
e.children(3).font_size = 3;    // text = "~1/x^0.819'"
e.children(4). line_style = 3;
  approx_sw100.line_style = 3;

// approx_sw_10K
  approx_sw_10K = a.children.children(4);
  approx_sw_10K.thickness=3;
  approx_sw_10K.foreground=5;
e.children(1).font_size = 3;    // text = "~1/x^1,048"
e.children(2).line_style = 5;   // polyline
  approx_sw_10K.line_style = 5;

drawnow();
```