УДК 004.912

К ВОПРОСУ ОБ ИНДЕКСИРОВАНИИ ВИКИ-ТЕКСТОВ

А. А. Крижановский[1] А. В. Смирнов[1,*]

[1] *Учреждение Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН*

199178, Санкт-Петербург, 14 линия д.39

Новый тип документов в вики-разметке завоёвывает Интернет. Это выражается не только в увеличении количества интернет-страниц в этой разметке, 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 и немецкий. Рассмотрена архитектура системы индексирования, включающая программные модули GATE и систему лемматизации Lemmatizer. Описаны правила преобразования вики-текстов в тексты на ЕЯ. Построены индексные базы Русской Википедии и Simple English Wikipedia. Проверено выполнение закона Ципфа для текстов Русской Википедии и Simple English Wikipedia.

1. ВВЕДЕНИЕ

Проведённый в США опрос [1] показал, 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 популярности Википедии кроется в её «авторитетности» в поисковых системах. Например, по данным Hitwise свыше 70 % посещений Википедии обеспечены переходами с поисковиков [1].

Работа выполнена при финансовой поддержке РФФИ (проект № 08-07-00264) и Программы фундаментальных исследований Президиума РАН (проект № 213 "Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация").

* Electronic address: {aka,smir}@iias.spb.su

Данные Википедии можно разделить на текст и ссылки (внутренние, внешние, интервики, категории). Внутренние ссылки связывают страницы внутри одного сайта. Интервики указывают на статью, описывающую данный энциклопедический термин, но на другом языке. Категории тематически классифицируют статьи. Всё это позволяет выделить следующие *три типа поисковых алгоритмов*:

- поиск на основе *анализа ссылок*, в котором можно разграничить случаи, когда:
  - ссылки заданы явно гиперссылками (HITS [2], PageRank [3, 4], ArcRank [5], Green [6], WLVM [7]);
  - ссылки нужно построить (Similarity Flooding [8], алгоритм извлечения синонимов из толкового словаря [5], [9], [10]);
- *анализ текста* с помощью статистических алгоритмов (ESA [11], сходство коротких текстов [12], извлечение контекстно связанных слов на основе частотности словосочетаний [13], LSA [33]);
- *анализ и ссылок, и текста* [14], [15].

Разработанный ранее адаптированный HITS алгоритм (AHITS) [16, 17] выполняет поиск семантически близких слов на основе анализа внутренних ссылок Википедии. Под *семантически близкими словами* (СБС) подразумеваются слова близкие по значению, встречающиеся в одном контексте. Это могут быть синонимы («чертог», «дворец»), антонимы («запутать», «распутать»), гиперонимы и гипонимы («самолёт» – «планер», «идти» – «ковылять»), холонимы и меронимы («граф» – «вершина», «глаз» – «хрусталик»). Многие алгоритмы поиска СБС в Википедии обходятся без полнотекстового поиска [18]. Однако экспериментальное сравнение алгоритмов [11, 18] показывает, что наилучшие результаты поиска семантически близких слов даёт алгоритм Explicit Semantic Analysis (ESA) [11], использующий именно полнотекстовый поиск.

Поэтому предложено создать общедоступную индексную базу данных Википедии (далее WikIDF) и программные средства для её генерации, что обеспечит полнотекстовый поиск в энциклопедии и в корпусах вики-текстов. Вики-текст – это упрощённый язык разметки HTML. Для индексирования требуется преобразовать его в текст на естественном языке (ЕЯ), чтобы поиск по ключевым словам не учитывал символы и

теги HTML- и вики-разметки. Для индексирования вики-текстов в виду относительной простоты реализации был выбран подход TF-IDF [19, 20].

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

Для разработки системы индексирования и построения индексной БД необходимо:

- разработать архитектуру системы построения индексной БД вики-текстов;
- спроектировать структуру таблиц индексной базы данных;
- определить правила преобразования вики-текста в текст на ЕЯ;
- реализовать программный комплекс для индексации (программный интерфейс доступа к индексной БД);
- провести эксперименты.

Структура статьи соответствует поставленным задачам. Заключает работу обсуждение способов улучшения индексной БД, а также проектов и подходов, включающих индексную БД как элемент решения других задач (информационный поиск и т. д.).

2. АРХИТЕКТУРА СИСТЕМЫ ПОСТРОЕНИЯ ИНДЕКСНОЙ БД ВИКИ-ТЕКСТОВ

Для построения индексной базы данных необходимо, во-первых, автоматическое разбиение текста на слова и, во-вторых, лемматизация слов. При этом был выбран подход, когда для решения каждой подзадачи не разрабатывается с нуля новая программа, а используются существующие компьютерные программы с открытым исходным кодом. Для решения первой задачи использовалась система GATE [21] (Java инструментарий, позволяющий выполнять обработку текста на многих языках), для второй – программа лемматизации Lemmatizer [22]. Для работы с данными Википедии (здесь для извлечения текстов из базы данных Википедии) использовалась программа Synarcher [16, 17].

Архитектура программной системы индексирования вики-текстов представлена на рис. 1. На рисунке показано взаимодействие программных модулей GATE, Lemmatizer и Synarcher. В результате работы всей системы генерируется индексная БД на уровне записей (англ. *"record level inverted index"* [19]), содержащая список ссылок на документы для каждого слова (леммы).

Программной системе требуется задать три группы входных параметров. Во-первых, *язык* текстов Википедии (один из 265 на 03/01/2009) и один из трёх языков (русский, английский, немецкий) для лемматизации, что определяется наличием трёх баз данных, доступных лемматизатору [22]. Указание языка Википедии необходимо для правильного преобразования текстов в вики-формате в тексты на ЕЯ (рис. 1, функция «Преобразование вики в текст» модуля «Обработчик Википедии»). Во-вторых, *адрес вики и индексной баз данных*, а именно 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. Например, ограничение числа связок слово-страница (в экспериментах из практических соображений ограничение было задано равным 1000).

«Управляющее приложение» выполняет последовательно три шага для каждой статьи (вики-текста), извлекаемой из БД Википедии и преобразуемой в текст на ЕЯ (что и составляет первый шаг).

На втором шаге с помощью программ GATE, Lemmatizer и объединяющего их программного интерфейса RussianPOSTagger строится список лемм и их частоты встречаемости в данной статье, точнее – вычисляется суммарная частота всех словоформ данной лексемы в заданной статье (и во всём корпусе) для каждой леммы.

На третьем шаге полученные данные сохраняются в индексную БД[1]: (1) полученные леммы, (2) частота их встречаемости в данном тексте, (3) принадлежность леммы данному вики-тексту, (4) частота встречаемости лемм в корпусе (увеличивается значение частоты лемм, полученных по данному тексту).

Отметим, что обе функции модуля «Обработчик Википедии», указанные на рис. 1,

[1] Построенные индексные БД Русской Википедии и Simple Wikipedia доступны по адресу: `http://rupostagger.sourceforge.net`, см. соответственно, пакеты `idfruwiki` и `idfsimplewiki`.

а также API доступа к индексной БД («TF-IDF Index DB» из модуля «TF-IDF Приложение») реализованы в программе Synarcher (`http://synarcher.sourceforge.net`). Указание входных параметров и запуск индексации осуществляются с помощью её модуля WikIDF, представляющего собой консольное приложение на языке Java.

3. ТАБЛИЦЫ И ОТНОШЕНИЯ В ИНДЕКСНОЙ БД

Для хранения информации в индексной БД используется реляционная модель данных. При этом:

- данные наполняются один раз и далее используются *только для чтения* (поэтому не рассматриваются такие вопросы, как: обновление индекса, добавление записи, поддержка целостности);
- данные хранятся в несжатом виде, т. е. не архивируются.

В ходе индексации из текста удаляется вики- и HTML- разметка; выполняется лемматизация, леммы слов сохраняются в индексную базу данных. Эта база данных не содержит данных, указывающих позицию слов в тексте.

Набор 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. далее)», а именно (рис. 2):

1. *term* – таблица содержит леммы слов (поле *lemma*); число документов, содержащих словоформы данной лексемы (*doc_freq*); суммарную частоту 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.

Окончание «*_id*» в названии полей таблиц обозначает уникальный идентификатор (рис. 2). В нижней части каждой таблицы перечислены поля, проиндексированные для

ускорения поиска. Между полями таблиц задано отношение *один ко многим* – между таблицами *term* и *term_page* (поле term_id), а также таблицами *page* и *term_page* (поле page_id).

Данная схема БД позволяет получить, во-первых, список лемм слов заданного документа. Точнее – это может быть меньшее число, чем все леммы слов данного текста. Поскольку для слов, встречающихся больше чем в N документах, $N+1$ запись «слово-документ» не будет записана в таблицу *term_page*. Во-вторых, можно получить список документов, содержащих словоформы лексемы, заданной своей леммой.

Напомним читателю формулу TF-IDF, которая используется для вычисления весов ключевых слов. И покажем, что данных в разработанной схеме БД (рис. 2) достаточно, чтобы воспользоваться этой формулой.

«idf (обратная частота термина в документах, обратная документная частота) – показатель поисковой ценности слова (его различительной силы)» [19]. В 1972 г. Karen Sparck Jones предложил 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 [20] представлено её теоретическое обоснование.

Всего в корпусе D документов, термин (лексема) t_i встречается в DF_i документах (чему соответствует значение поля БД *term.doc_freq*), где *term.doc_freq* – это сокращённая запись, указывающая на поле *doc_freq* таблицы *term* индексной базы данных. Для заданного термина t_i вес документа $w(t_i)$ определяется как [20]:

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

где TF_i – число вхождений термина t_i в документ (поле *term_page.term_freq*), idf 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 индексной БД позволяют вычислить обратную частоту термина t_i в корпусе.

4. ПРЕОБРАЗОВАНИЕ ВИКИ-ТЕКСТА В ТЕКСТ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ПОМОЩЬЮ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Тексты Википедии содержат вики-разметку. 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. В ходе работы возникали вопросы типа: «как и какие элементы разметки обрабатывать?». Представим, аналогично работе [25], вопросы и принятые решения в табл. 1. Для некоторых преобразований в таблице приведены регулярные выражения [26].

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.

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

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

(ii) Выполняются следующие преобразования вики-тегов:

1. текст примечаний (`<ref>`) извлекается и добавляется в конец всего текста;
2. удаляются двойные фигурные скобки и текст внутри них (`{{шаблон}}`); (данная подфункция вызывается дважды, чтобы удалить `{{шаблон в {{шаблоне}}}}`, более глубокие вложения в данной версии не учитываются);
3. удаляются таблицы и текст (`{| таблица 1 \n {| таблица в таблице 1 \n |}|}`);
4. удаляется знак ударения в текстах на русском языке (например, *Кòтор*);
5. удаляется тройной апостроф, окружающий текст и обозначающий `'''жирное выделение'''`; текст остаётся;

6. удаляется двойной апостроф, обозначающий *''наклонное начертание''*; текст остаётся;
7. из тега изображения извлекается его название, прочие элементы удаляются;
8. обрабатываются двойные квадратные скобки (раскрываются внутренние ссылки, удаляются интервики и категории);
9. обрабатываются одинарные квадратные скобки, обрамляющие гиперссылки: ссылка удаляется, текст остаётся;
10. удаляются символы (заменяются на пробел), противопоказанные XML парсеру (протокола XML-RPC программы RuPOSTagger): *<, >, &, "*; удаляются также их «XML-безопасные» аналоги: *<, >, &, "*; а также: *', , –, —* символы *
,
,
* заменяются символом перевода каретки.

Данный преобразователь вики-текста реализован в виде одного из Java-пакетов программы Synarcher [17]. В табл. 2 приведён фрагмент статьи Русской Википедии «Через тернии к звёздам (фильм)» и показан результат комплексного преобразования текста по всем вышеуказанным правилам.

5. API ИНДЕКСНОЙ БАЗЫ ДАННЫХ ВИКИ

В настоящее время существуют следующие программные интерфейсы (API) для работы с данными Википедии:

- FUTEF API для поиска в английской Википедии с учётом категорий Википедии (`http://api.futef.com/apidocs.html`). Поисковик реализован как веб-сервис на основе Yahoo!, результат возвращается в виде Javascript объекта JSON;
- интерфейс для вычисления семантического сходства слов в Википедии [27]. Здесь запрос идёт из Java через XML-RPC к Perl-процедуре, затем посредством MediaWiki выполняется обращение к БД;
- интерфейс к Википедии и Викисловарю [28];

- набор интерфейсов для работы с данными Википедии, хранимыми в XML базе данных Sedna (`http://wikixmldb.dyndns.org`).

Структура предложенной индексной БД (рис. 2) отличается от схемы БД MediaWiki (при этом для работы с БД MediaWiki уже написано достаточное количество необходимых функций в программе Synarcher), поэтому возникла необходимость в разработке «сопряжения» для программного управления индексом. С этой целью был разработан программный интерфейс для работы с базой данных WikIDF.

Функции верхнего уровня (интерфейса WikIDF) позволяют, во-первых, получить список терминов для заданной вики-страницы, упорядоченный по значению TF-IDF. Во-вторых, получить список документов, содержащих словоформы лексемы по заданной лемме; документы упорядочены по значению частоты термина (TF).

Функции низкого уровня предназначены для работы с отдельными таблицами индексной БД (рис. 2) и позволяют прочитать, сохранить или удалить запись в таблице.

6. ПРОВЕРКА ВЫПОЛНЕНИЯ ЗАКОНА ЦИПФА ДЛЯ ВИКИ-ТЕКСТОВ

Эмпирический закон Ципфа говорит о том, что частота употребления слова в корпусе обратно пропорциональна его рангу в списке упорядоченных по частоте слов этого корпуса [29], 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 нарисовать график логарифма частот слов в зависимости от логарифма порядкового номера в списке, то получится прямая [20]. Такой график представлен на рис. 3.

Кривая, составленная из знаков "+", построена по данным корпуса текстов Русской Википедии от 20 февраля 2008 (RW). С помощью метода наименьших квадратов пакета Scilab [30] были построены *аппроксимирующие кривые* y_{100}^{RW} по первым ста наиболее частотным словам корпуса (см. рис. 3, пунктир с точкой) и y_{10K}^{RW} по первым 10 тыс. слов (длинный пунктир):

$$\mathrm{y}_{100}^{RW}(\mathrm{x}) = \frac{e^{14.51}}{x^{0.819}}; \quad \mathrm{y}_{10K}^{RW}(\mathrm{x}) = \frac{e^{16.13}}{x^{1.048}}$$

Знакам “X” на рис. 3 соответствуют данные Simple English Wikipedia от 14 февраля 2008 (SEW).

Аналогично нарисованы аппроксимирующие кривые: y_{100}^{SEW} (точечный пунктир) и y_{10K}^{SEW} (пунктир с двумя точками).

$$\mathrm{y}_{100}^{SEW}(\mathrm{x}) = \frac{e^{12.83}}{x^{0.974}}; \quad \mathrm{y}_{10K}^{SEW}(\mathrm{x}) = \frac{e^{14.29}}{x^{1.174}}$$

Отметим, что аппроксимирующая линия y_{10K}^{RW} является более пологой (с угловым коэффициентом равным -1.048), чем более крутая линия y_{10K}^{SEW} (коэффициент -1.174), соответствующая более резкому падению частот английских слов. Возможные объяснения таковы. Во-первых, размер Русской Википедии на порядок больше, и, по-видимому, задействован более широкий словарь для описания большего количества понятий. Во-вторых, авторы Википедии на английском упрощённом языке (Simple English 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) соответствуют данному закону немного лучше корпуса русских текстов (0.23). Значение 0.20 – это разница между угловыми коэффициентами (степенями наклона) аппроксимирующих прямых, построенных по ста (0.974) и 10 тысячам (1.174) английских слов.

Таким образом, закон Ципфа выполняется несколько лучше для текстов на английском упрощённом языке, что можно объяснить либо особенностью упрощённого языка, либо разницей между русским и английским языками. Для окончательного выяснения вопроса нужно построить индексную БД не для Simple English Wikipedia, а для Английской Википедии (English Wikipedia).

7. ОБСУЖДЕНИЕ

Недостатки созданной системы индексирования вики-текстов:

- индекс создаётся единожды, при обновлении корпуса текстов его нужно перестраивать полностью; необходимо инкрементальное непрерывное индексирование;

- при значении переменной `doc_freq_max` (ограничивающей размер индексной БД) равной 100 (а не 1000, например), короткие статьи имеют мало вхождений в таблицу *term*, т.е. для малого числа слов данной статьи будут указаны связки «лексема-страница» в таблице *term_page*;

- одна словоформа может иметь несколько лексем в базе системы Lemmatizer (лексемы имеют разные ID), чтобы сохранить эту информацию в БД WikIDF, нужно добавить ещё одну таблицу.

Выводы и предложения по улучшению системы индексирования вики-текстов:

- вес *tf-idf* указывает на значимость слова во всём корпусе текстов, поэтому вес слова, например, «байт» будет, скорее всего, мал в корпусе текстов о программировании и значителен – в корпусе текстов о биологии. Целесообразно использование категорий для уточнения значения веса. Таким образом, вес будет зависеть от проблемной области вики-текстов и у одного и того же слова будет разным в текстах разной тематики;

- для этой же цели (уточнения веса слова в зависимости от тематики текста) целесообразно добавить в таблицу term (рис. 2) поле «коэффициент вариации D», то есть оценку специфичности слова для отдельной предметной области ([32], стр. 347).

- полезным дополнительным ресурсом для индексирования будет размеченный корпус Английской Википедии [31], что позволит выполнять поиск, учитывая семантическую разметку, например, используя 45 категорий верхнего уровня иерархии наборов синонимов WordNet, присвоенных словам Википедии;

Планируемые варианты развития системы WikIDF ориентированы, во-первых, на включение WikIDF (как одного из компонент) в программу Synarcher для выполнения полнотекстового (а не только по заголовкам вики-страниц) поиска семантически близких слов. 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. Таким образом, с помощью полнотекстового поиска в вики-текстах будет возможна генерация корневого набора страниц в адаптированном

HITS алгоритме (AHITS) [17], что, по-видимому, улучшит результат поиска семантически близких слов. Во-вторых, преобразование в машинную форму Викисловаря, в первую очередь семантических отношений Викисловаря, также позволит сделать поиск в вики-текстах более полным и точным.

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

- модуль ANNIC системы GATE, основанный на поисковом движке Lucene [34]; см. также в [35] описание варианта работы с вики из системы GATE.
- индексирующая система MG4J [36];
- инструментарий Lemur с возможностями индексирования (английский, китайский, арабский языки) и поисковиком INDRI (см. `http://www.lemurproject.org`).

Перспективные направления исследований, связанные с индексной БД, таковы:

- определение значения многозначных слов. В работе [37] сделали предположение, 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 [37]. Для экспериментов использовали такой ресурс, как WordNet Domain (расширенная версия WordNet, см. `http://wndomains.itc.it`), где каждый синсет содержал метки предметных областей;
- фильтрация потока текстов [38];
- поиск ключевых слов с учётом семантических отношений (синонимы, гипонимы и др.), например, в [39] (на основе WordNet или CYC) или Викисловаря;
- оценка точности поиска методом TF-IDF, определение оптимальных поисковых параметров за счёт *a*) отсечения высокочастотных слов [40]; *b*) сравнения мер сходства в векторном пространстве слов [40]; и *c*) учёта аддитивной модели расчёта релевантности документа запросу [41].

8. ЗАКЛЮЧЕНИЕ

Растёт не только Интернет, подрастает и Википедия, причём по последним данным [42] энциклопедия увеличивается и совершенствуется в шестимерном пространстве, а именно растут:

- число языков, на которых излагается материал Википедии;
- число активных участников (со временем число участников растёт, но относительное число высоко активных участников, т.е. больше 100 правок в месяц, снижается [42]);
- число тематических направлений (у каждой новой группы участников, у каждой языковой группы – свои интересы);
- число статей в целом; а в «больших» Википедиях глубина проработки (формально – это размер статьи, число правок);
- связность страниц (то есть число внутренних ссылок, интервики, категорий);
- «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 [3]), поисковики присваивают вики-текстам высокий рейтинг, то есть ставят их на первые позиции в результатах поиска [1]. 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: Wikia Search, Lucene-search, FUTEF, а также в Русской Википедии: Qwika. Отметим, что будущее поисковых систем, возможно, зиждется на распределённом поиске с помощью P2P приложений [43]. Важной задачей поисковиков было и остаётся индексирование текстов.

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. С помощью системы

WikIDF построены индексные базы данных для Русской Википедии и Википедии на английском упрощённом языке.

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 14 % и на 7 % быстрее чем в русской пополнялся лексикон Википедии на английском упрощённом языке.

СПИСОК ЛИТЕРАТУРЫ

1. *Rainie L., Tancer B.* Wikipedia users // Reports: Online Activities & Pursuits. 2007.
   http://www.pewinternet.org/pdfs/PIP_Wikipedia07.pdf
2. *Kleinberg J.*, Journal of the ACM, **5**, 46 (1999).
   http://www.cs.cornell.edu/home/kleinber
3. *Brin S., Page L.* The anatomy of a large-scale hypertextual Web search engine. 1998.
   http://www-db.stanford.edu/~backrub/google.html
4. *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
5. Survey of text mining: clustering, classification, and retrieval, M. Berry (Ed.). Springer-Verlag, New York, 2003.
6. *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
7. *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
8. *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
9. *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
10. *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, **46**, 4, (2004).

11. *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

12. *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

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

14. *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.
ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf

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

16. *Крижановский А. А.* Автоматизированный поиск семантически близких слов на примере авиационной терминологии. Автоматизация в промышленности. **4**, 64, (2008).

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

18. *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

19. *Сегалович И. В.* Как работают поисковые системы.
http://company.yandex.ru/articles/

20. *Robertson S.* Understanding inverse document frequency: on theoretical arguments for IDF. Journal of Documentation, **60**, (2004).

http://www.soi.city.ac.uk/~ser/idfpapers/Robertson_idf_JDoc.pdf

21. *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.
22. *Сокирко А. В.* Морфологические модули на сайте www.aot.ru // Компьютерная лингвистика и интеллектуальные технологии. Труды международной конференции "Диалог 2004". "Верхневолжский", 2004.
http://www.aot.ru/docs/sokirko/Dialog2004.htm
23. *Codd E. F.* // The relational model for database management: version 2. Addison-Wesley, MA, 1990.
24. *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. 2008.
http://arxiv.org/abs/0803.2220
25. *Вахитова Д.* Создание корпуса текстов по корпусной лингвистике, 2006.
http://matling.spb.ru/files/kurs/Vahitova_Corpus.doc
26. *Фридл Дж.* // Регулярные выражения. СПб.: Питер, 2001.
27. *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, June 23-30, 2007.
http://www.eml-research.de/english/homes/ponzetto/pubs/acl07.pdf
28. *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
29. *Manning C. D., Schutze H.* // Foundations of Statistical Natural Language Processing. The MIT Press, 1999.
30. *Campbell S., Chancelier J.-P., Nikoukhah R.* // Modeling and Simulation in Scilab/Scicos.

Springer, 2006.

31. *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. Morocco, Marrakech, May 26 – June 1, 2008.
http://www.lrec-conf.org/proceedings/lrec2008/pdf/581_paper.pdf
32. *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.
http://www.dialog-21.ru/dialog2008/materials/pdf/53.pdf
33. *Куралёнок И., Некрестьянов И.* Автоматическая классификация документов на основе латентно-семантического анализа // "Электронные библиотеки: перспективные методы и технологии, электронные коллекции". Санкт-Петербург, 18-22 октября 1999.
http://www.dl99.nw.ru
34. *Aswani N., Tablan V., Bontcheva K., Cunningham H.* Indexing and querying linguistic metadata and document content. // RANLP2005. Bulgaria, Borovets, 2005.
http://gate.ac.uk/sale/ranlp05/ranlp05-annic.pdf
35. *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
36. *Boldi P., Vigna S.* Efficient optimally lazy algorithms for minimal-interval semantics, 2007.
http://vigna.dsi.unimi.it/papers.php
37. *Magnini B., Strapparava C., Pezzulo G., Gliozzo A.* The role of domain information in word sense disambiguation. Journal of Natural Language Engineering, **8**, 4, (2002).
http://www.istc.cnr.it/doc/1a_16p_Magnini-NLE-2002.pdf
38. *Smirnov A., Krizhanovsky A.* Information filtering based on wiki index database. // In FLINS'08. Spain, Madrid, September 21-24, 2008.
http://arxiv.org/abs/0804.2354
39. *Shamsfard M., Nematzadeh A., Motiee S.* ORank: an ontology based system for ranking documents. International Journal of Computer Science, **1**, 3, (2006).
http://www.waset.org/ijcs/v1/v1-3-30.pdf

40. *Meyer M., Rensing C., Steinmetz R.* Categorizing learning objects based on Wikipedia as substitute corpus. // In LODE'07. Greece, Crete, September 18, 2007.
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-311/paper09.pdf

41. *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

42. *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

43. *Wu L.-S., Akavipat R., Menczer F.* 6S: P2P Web index collecting and sharing application. // In: RIAO, 2007.
http://sixearch.org/paper/6S_P2P_Web-1.pdf

CONCERNING THE QUESTION OF WIKI TEXTS INDEXING

A. A. Krizhanovsky, A. V. Smirnov

With the fantastic growth of Internet usage, information search in documents of a special type called a “wiki page”, that is written using a simple markup language, has become an important problem. This paper describes the software architectural model for indexing wiki texts in three languages (Russian, English, and German) and the interaction between the software components (GATE, Lemmatizer, and Synarcher). The rules for parsing Wikipedia texts are illustrated by examples. Two index databases of Russian Wikipedia (RW) and Simple English Wikipedia (SEW) are built.

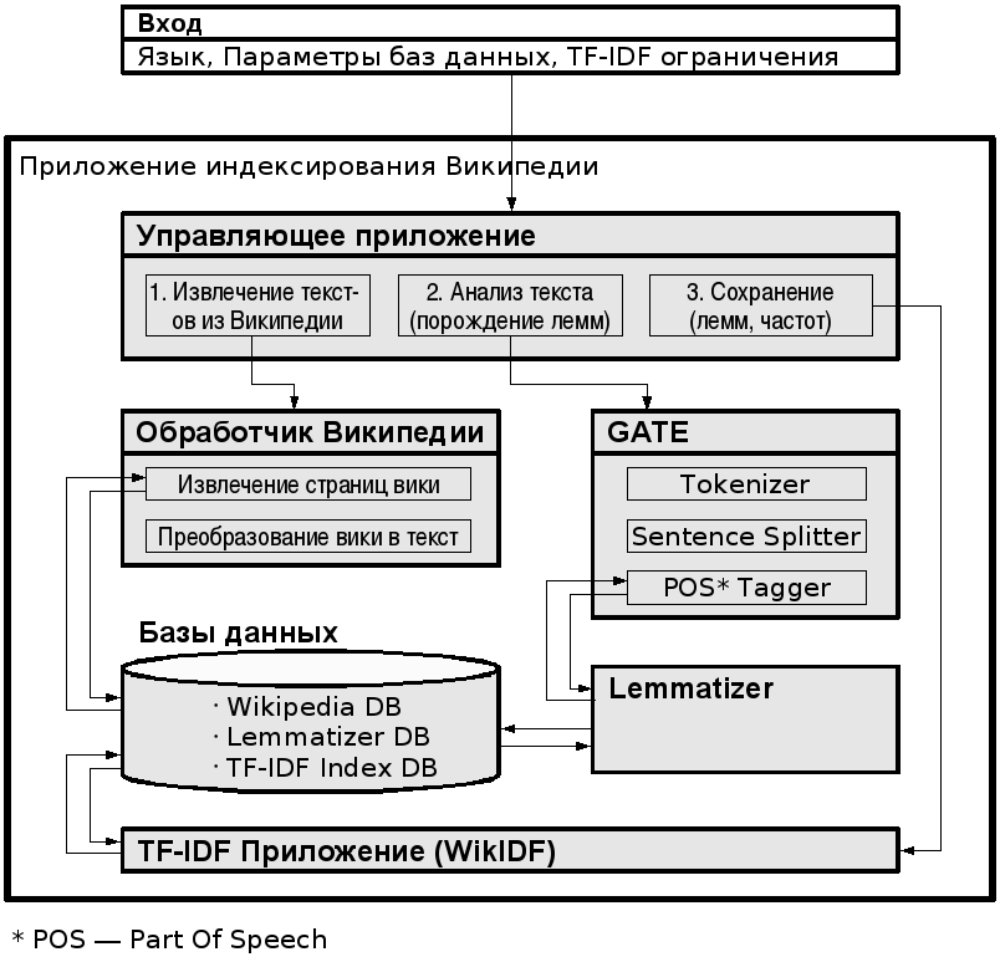

Рис. 1. Архитектура системы индексирования вики-текстов.

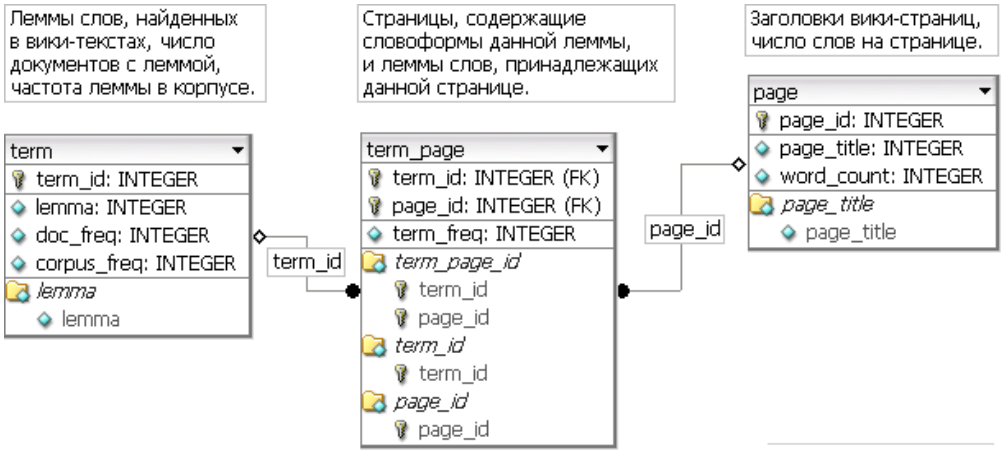

Рис. 2. Таблицы и отношения в индексной базе данных WikIDF.

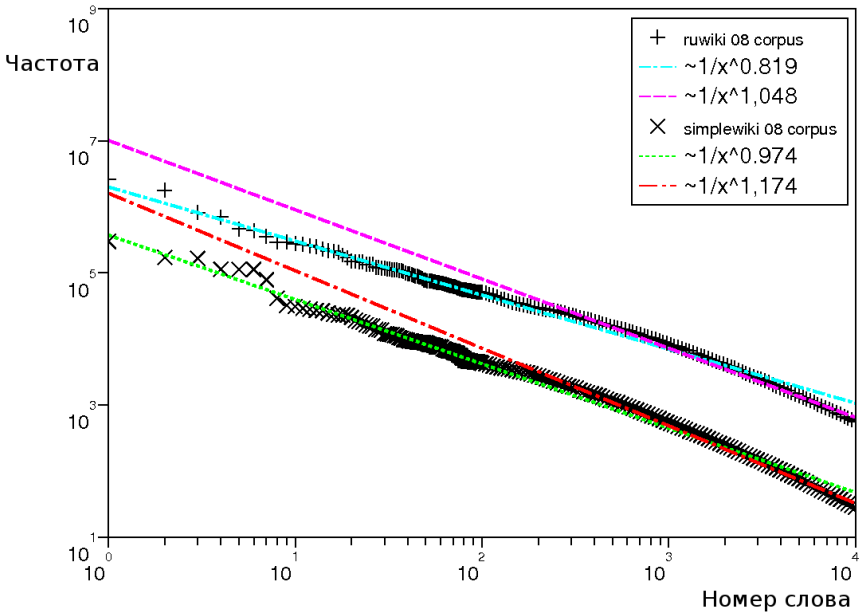

Рис. 3. Линейная зависимость убывания частоты употребления слов в корпусе от порядкового номера (ранга) слова в списке слов, упорядоченных по частоте, в масштабе логарифм-логарифм для Русской Википедии (ruwiki) и Simple Wikipedia (simplewiki) на февраль 2008 г., линейная аппроксимация по 100 и 10 000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

Таблица 1. Решения по парсингу вики-текста.

№	*Вопросы* Исходный текст	*Ответы* Преобразованный текст
1	Заголовки (подписи) рисунков [[Image:Asimov.jpg|thumb|180px|right|[[Isaac Asimov]] with his [[typewriter]].]]	Оставить (извлечь) [[Isaac Asimov]] with his [[typewriter]].
2	Интервики	Оставить или удалить (определяется пользователем)
3	Названия категорий RE (регулярное выражение):	Удалить `\[\[Категория:.+?\]\]`
4	Шаблоны; цитаты; таблицы	Удалить
5	Курсив и ≪жирное≫ написание "'italic"' "bold"	Апострофы удаляются italic bold
6	Внутренняя ссылка [[w:Wikipedia:Interwikimedia_links|text to expand]] [[run]] [[Russian language|Russian]] в [[космос|космическом пространстве]]. RE: внутренняя ссылка без вертикальной черты:	Оставить текст, видимый пользователю, удалить скрытый текст text to expand run Russian в космическом пространстве. `\[\[([^:|]+?)\]\]`
7	Внешняя ссылка [http://example.com Russian] [http://www.hedpe.ru сайт hedpe.ru – фан-сайт] RE: Имя сайта (без пробелов), содержащее точку '.' хотя бы раз, кроме последнего символа:	Оставить текст, видимый пользователю, удалить сами гиперссылки Russian сайт – фан-сайт `(\A|\s)\S+?[.]\S+?[^.]([\s,!?]|\z)`

Таблица 2. Пример преобразования вики-текста.

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≫]]	≪Через тернии к звёздам≫
"'≪Через тернии к звёздам≫"' [[научная фантастика|научно-фантастический]] двухсерийный фильм [[режиссёр]]а [[Викторов, Ричард Николаевич|Ричарда Викторова]] по сценарию [[Кир Булычёв|Кира Булычёва]].	≪Через тернии к звёздам≫ научно-фантастический двухсерийный фильм режиссёра Ричарда Викторова по сценарию 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.	== 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"	== В ролях == * Елена Метёлкина Нийя
== Ссылки == {{викицитатник}} * [http://ternii.film.ru/ Официальный сайт фильма]	== Ссылки == * Официальный сайт фильма
[[Категория:Киностудия им. М. Горького]] [[en:Per Aspera Ad Astra (film)]]	