ARTIFICIAL INTELLIGENCE

On the Problem of Wiki Texts Indexing

A. A. Krizhanovsky and A. V. Smirnov

St.-Petersburg Institute of Informatics and Automation, Russian Academy of Sciences, 14 liniya 39, St.-Petersburg, 199178 Russia

Received February 17, 2009.

A new type of documents called a "wiki page" is winning the Internet. This is expressed not only in an increase of the number of Internet pages of this type, but also in the popularity of Wiki projects (in particular, Wikipedia); therefore the problem of parsing in Wiki texts is becoming more and more topical. A new method for indexing Wikipedia texts in three languages: Russian, English, and German, is proposed and implemented. The architecture of the indexing system, including the software components GATE and Lemmatizer, is considered. The rules of converting Wiki texts into texts in a natural language are described. Index bases for the Russian Wikipedia and Simple English Wikipedia are constructed. The validity of Zipf's laws is tested for the Russian Wikipedia and Simple English Wikipedia.

1.INTRODUCTION

A poll conducted in the USA [1] has shown that more than one third (36%) of adult Internet users consult online texts, Wikipedia encyclopedia. Its popularity is explained by huge amount, diversity, and originality of the material. Another reason for popularity of Wikipedia is its "authority" in retrieval systems. For instance, as Hitwise testifies, more than 70% visits of Wikipedia are provided by transitions from search engines [1].

Wikipedia data can be split into texts and links (internal, external, inter-wikis, categories) Internal links bind pages within a site. Interwikis specify the article that describes a given encyclopedia term, but in another language. The categories classify articles thematically. This allows us to distinguish the following three *types of search algorithms*:

search based on *links analysis*, in which we can distinguish the cases when:

links are explicitly given by hyperlinks (HITS [2], PageRank [3, 4], ArcRank [5], Green [6], and WLVM [7]);

links should be constructed (Similarity Flooding [8], an algorithm for extracting synonyms from the thesaurus [5, 9, 10]);

text analysis with the help of statistical algorithms (ESA [11], similarity of short texts [12], extraction of contextually linked words based on the frequency of word combinations [13], LSA [14]);

analysis of both links and the text [15], [16].

The HITS-algorithm developed earlier and adapted (AHITS) [16, 17] finds semantically close words based on the analysis of internal Wikipedia links. By *semantically close words (SCW),* we mean words close in meaning occurred in the same text. They can be synonyms ("mansion", "palace"), antonyms ("entangle", "untangle"), hypernyms и hyponyms ("aircraft" – "glider", "go" – "hobble"), homonyms and meronyms ("graph" – "vertex", "eye" – "lens"). Many algorithms for searching SCW in Wikipedia do without full-text search [19]. However, the experimental comparison of algorithms [11, 19] has shown that the best results in searching semantically close words are demonstrated by the ESA algorithm, using the full-text search.

Therefore, it was suggested to design a public index database of Wikipedia (in what follows, WikIDF) and software tools for its generation, which ensures full-text search in the encyclopedia and in corpuses of wiki texts. A wiki text is a simplified HTML markup language. For indexing, it is necessary to convert it into a text in a natural language (NL), to provide that keyword search does not take into account symbols and tags of HTML and wiki markups. For indexing wiki texts, in view of sufficiently simple implementation, the TF-IDF approach [20, 21] was chosen.

The designed software resources (database and indexing system) allow users to analyze the obtained index databases (DB) of wikipedias, and make it possible for designers of search engines to use the WikIDF program and to provide searching over wiki resources by accessing

the existing index bases or by generating new ones. For implementation of the indexing system and construction of an index database, it is necessary:

to form the architecture of the system for designing an index database of wiki texts;
to design the structure of tables of the index DB;
to define rules of converting wiki texts into texts in an NL;
to construct a software system for indexing (program interface of access to the index DB);
to conduct experiments.

The structure of the paper suits to the posed problems. The paper is concluded by a discussion of methods for improving the index DB, as well as projects and approaches involving the index DB as an element of solving other problems (data retrieval, etc.)

1. THE ARCHITECTURE OF THE SYSTEM FOR DESIGNING AN INDEX DB OF WIKI TEXTS

To design an index database, it is necessary, first, to develop automatic division of a text into words, and, second, word lemmatization. Note that we chose the approach when for solving each subtask the existing computer programs with open source code are employed, rather than a new program is developed from scratch. For solving the first task, the GATE system was applied [22] (Java -is the tool that allows processing texts in many languages). The second task was solved by the lemmatization program Lemmatizer [23]. To deal with Wikipedia data (here, to extract texts form the Wikipedia database), we used the Synarcher program [17, 18].

Figure 1 presents the architecture of the system for indexing wiki texts, where the interaction of the program modules GATE, Lemmatizer, and Synarcher is shown. As a result of operation of the whole system, an index DB is generated at the level of records (*record level inverted index* [20]), containing the list of links to documents for each word (lemma).

It is necessary to specify three groups of input parameters for the program. First, the *language* of Wikipedia texts (one of 265 on 01/03/2009) and one of the three languages (Russian, English, and German) for lemmatization, which is determined by the presence of three DBs available for the lemmatizer [23]. The indication of the Wikipedia language is necessary in order to convert correctly texts in wiki formats into texts in an NL (Fig. 1, the function "Conversion of wiki in a text" of the module "Wikipedia handler"). Second, we should specify *the address of wiki and index DBs*, namely the parameters for connection to a remote DB: IP-address, DB name, user name and login. Third, *indexing parameters*, connected with constraints imposed by the user on the size of the index DB aimed at subsequent search according to the TF-IDF scheme have to be defined. For example, limiting the number of connectives word--page (in experiments from practical considerations, the constraint was given equal to 1000).

The "controlling application" executes consecutively three steps for each article (wiki text) extracted from the Wikipedia DB and converted into an NL (which is the first step). At the first step, using the programs GATE and Lemmatizer and the program interface RussianPOSTagger that joins them, a list of lemmas and the frequency of their occurrence in the given article are determined; more accurately, the total frequency of all word forms in a given lexeme in the given article (and in the whole corpus) for each lemma is calculated. At the third step, the data are saved in the index DB[*]: the obtained lemmas, the frequencies of their occurrence in a given text, the fact whether the lemma belongs to a given wiki text, the frequency of occurrence of lemmas in the corpus (the value of the frequency of lemmas found within a given text is increased).

Note that both functions of the module "Wikipedia handler", specified in Fig. 1, as well as API of the access to the index DB ("TF-IDF Index DB" from the module "TF-IDF Application") are implemented in the Synarcher program. The setting of the input parameters and running of indexation are executed by its module WikIDF, representing a console application written in

[*]Constructed index DBs of Russian Wikipedia and Simple English Wikipedia are available at: http://rupostagger.sourceforge.net (see packages idfruwiki and idfsimplewiki, respectively).

Java.

2. TABLES AND RELATIONS IN THE INDEX DB

To store data in the index DB, a relational data model is used:

data are filled once and then are used *only for reading* (therefore such problems as index update, add-on recording, integrity support are not considered);

data are stored in an uncompressed form, i.e., not archivated.

In the course of indexing, wiki- and HTML-markup is eliminated, the lematization is performed, and lemmas of words are stored in the index DB. This DB does not contain information specifying the position of words in the text. The set of tables in the index DB, their filling, and relations between them were determined in agreement with the problem solved: "Search for texts based on a given word by using the TF*IDF formula (see in what follows)", namely (Fig. 2):

1) *term* is the table containing lemmas of words (the *lemma* field); the number of documents containing the word forms of a given lexeme (*doc_freq*); the total frequency of word forms of a given lexeme in the whole corpus (*corpus_freq*);

2) *page* is the list of names of indexed documents (the field *page_title* exactly corresponds to the field of the table with the same name in the DB MediaWiki); and the number of words in the document (*word_count*);

3) *term_page* is the table that connects lemmas of word forms found in documents with these documents.

The ending "*_id*" in the name of table fields means a unique identifier (Fig. 2). In the bottom part of each table, fields indexed for accelerating the search are listed. Between the table fields, the relation *one to many* is given, between the tables *term* and *term_page* (the term_id field), as well as between the tables *page* and *term_page* (the page_id field). This scheme of the DB allows one to obtain, first, the list of lemmas of words of a given document, whose length may be less than all lemmas of words of the given document, since for words occurring in more than N documents, the $(N+1)$-th record "word--document" is not recorded in the table *term_page*. Second, we can form the list of documents containing the word forms of the lexeme given by its lemma.

Let us recall the TF-IDF formula, which is used for calculating the weights of keywords, and show that the data in the developed DB scheme (Fig. 2) are sufficient for using this formula. This formula is based on idf (the inverse frequency of the term in documents, the inverse document frequency), which the index of search value of the word (its discriminating ability) [20]. In 1972 Karen Sparck Jones proposed the heuristics: <<the term of a query occurred in a large number of documents has a weak discriminating ability (widely used word), we should assign to it a smaller weight compared with the term rarely occurred in the document of a collection (rare word)>>. This heuristics has shown its advantage in practice and in [21] its theoretical substantiation can be found. Totally, there are D documents in the corpus, the term (lexeme) t_i occurs in DF_i documents (to which the field of the DB *term.doc_freq* corresponds, where *term.doc_freq* is the reduced record pointing the field *doc_freq* of the table *term* of the index DB). For a given term t_i the weight of the document $w(t_i)$ is determined as [21]

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

where TF_i is the number of occurrences of the term t_i in the document (the field *term_page.term_freq*), and idf is used for reducing the weight of high-frequency words. We can normalize TF_i, taking into account the length of the document, i.e., dividing by the number of words in the document (the field *page.word_count*). Thus, the values of fields of the index DB allow one to calculate the inverse frequency of the term t_i in the corpus.

3. CONVERSION OF A WIKI TEXT IN A TEXT IN A NATURAL LANGUAGE USING REGULAR EXPRESSIONS

Wikipedia texts contain wiki markup. There is a daily need in converting wiki texts, namely in elimination or "disclosure" of wiki tags (i.e., extraction of the text component). If we omit this step, then special tags, e.g., "ref", "nbsp", "br", etc., fall in the hundred of the most frequent words of the index DB. In the course of work, questions, such as how and what elements of the markup should be processed, arise. Let us present questions and made decisions in Table 1 as in paper [24]. For some conversions, regular expressions are presented in the table [25]. To transform texts in the wiki format into texts in an NL, we should perform consecutively the steps that can be split into two groups: elimination and transformation of the text.

Step1. The following tags are eliminated (together with the text within them):

1) HTML -- commentaries (<!– ... –>);
2) tags of shutdown of formatting (<pre>...</pre>);
3) tags of the source codes (<source> и <code>).

Step 2. Transformations of wiki tags are performed:

1) the text of footnotes (<ref>) is extracted and added at the end of the whole text;
2) double braces are eliminated, as well as the text within them ({{template}}); (this subfunction is called twice to eliminate {{template in {{template}}}}, deeper nesting is not taken into account in this version);
3) the tables and text are eliminated ({| table 1 \n {| table in table 1 \n|}|});
4) the stress mark is eliminated in texts in Russian (e.g., *Ko@'mop*);
5) the triple apostrophe, surrounding the text and meaning "bold emphasis" is eliminated; the text is retained;
6) the double apostrophe, meaning "cursive" is eliminated; the text is retained;
7) the name is extracted from an image tag, the other elements are eliminated;
8) double square brackets are processed (internal links are disclosed, and interwikis and categories are eliminated);
9) single square brackets, bordering hyperlinks, are analyzed: the text without link is conserved;
10) symbols that are prohibited in XML-parser (XML-RPC protocol of the RuPOSTagger program): <, >, &, " are eliminated (are replaced by a blank); their "XML-safe" analogs <, >, &, " are also eliminated; as well as ', , –, — instead of symbols
,
,
 the carriage return character is used.

This transformer of a wiki text is implemented in the form of one Java-packages of the Synarcher program [18]. Table 2 presents a fragment of an article from Russian Wikipedia "Through thorns to the stars (movie)" and shows the result of complex transformation of the texts according to the rules listed above.

4. API OF THE INDEX DB

At present, there exist the following program interfaces (API) for dealing with Wikipedia data:

FUTEF API for searching in English Wikipedia with account of Wikipedia categories (http://api.futef.com/apidocs.html). The search engine is implemented as a web-service based on Yahoo!, the result is returned in the form Javascript of the object JSON;

the interface for calculating semantic similarity of words in Wikipedia [26], here the query goes from Java through XML-RPC to the Perl-procedure, then by MediaWiki call to the DB is performed;

the interface from Wikipedia to Wiki dictionary [27];

the set of interfaces for dealing with Wikipedia data stored in XML database Sedna (http://wikixmldb.dyndns.org).

The structure of the proposed index DB (Fig. 2) differs from the scheme of the DB MediaWiki (note that for dealing with the DB MediaWiki a sufficient number of necessary functions in the program Synarcher have been already written); therefore the necessity in developing an

"interface" for program control of the index arises. For this purpose, the program interface for dealing with the database WikIDF was designed. The upper-level functions (of the interface WikIDF) allow one, first, to form the list of terms for a given wiki page, arranged according to the value TF-IDF. Second, we can obtain a list of documents containing word forms of the lexeme based on a given lemma; the documents are arranged according to the frequency of the term (TF). The lower-level functions are aimed at dealing with particular tables of the index DB (Fig. 2) and provide reading, saving or deleting records in the table.

5. TESTING THE VALIDITY OF ZIPF'S LAW FOR WIKI TEXTS

The empirical Zipf's law states that the frequency of word usage in the corpus is inversely proportional to its rank in the list of words of this corpus arranged according to frequency [28]; i.e., the word that is the second in frequency occurs in the text two times rarely, than the first one, the third word in the list occurs three times rarely, than the first one, etc. Another formulation of Zipf's law states that if we construct a list of words ranking the words according to the frequency of their occurrence in a *sufficiently large* text, and draw a plot of the logarithm of word frequency depending on the logarithm of the serial number in the list, then we obtain a straight line [21]. Figure 3 presents this plot. The curve, generated by symbols "+", is constructed based on data of the corpus of texts of Russian Wikipedia of February 20, 2008 (RW). Using the least squares method of the package Scilab [29], we calculated *approximating curves* y_{100}^{RW} based on the first 100 most usable words of the corpus (see Fig. 3, dashed and dotted lines) and y_{10K}^{RW} based on the first 10 thousand words (dashed line with long dashes)

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

Data of Simple English Wikipedia of February 14, 2008 (SEW) correspond to symbols "X" in Fig. 3. Approximating curves are drawn in a similar way: y_{100}^{SEW} (dotted line) and y_{10K}^{SEW} (double dotted line)

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

Note that the approximating line y_{10K}^{RW} is more flattened (with the angular coefficient -1.048), than the line y_{10K}^{SEW} (the coefficient is -1.174), corresponding to the sharper decrease of frequencies of English words. The possible explanations are as follows. First, the size of Russian Wikipedia is by an order of magnitude greater, presumably, a wider dictionary for describing a greater number of notions is employed. Second, the authors of Simple English Wilipedia purposefully use simpler words and thus employ a smaller vocabulary.

Figure 3 shows that Zipf's law is valid for Wikipedia texts on the whole, i.e., the curve in the figure with the logarithmic scale can be approximated by a straight line quite well. Note that Simple English Wikipedia data (0.20) correspond to this law a bit better than corpus of Russian texts (0.23). The value 0.20 is the difference between the angular coefficients (degrees of slope) of approximating straight lines constructed based on 100 (0.974) and 10 thousands (1.174) of English words.

Thus Zipf's law is satisfied a bit better for texts in Simple English, which can be explained by either the specific feature of this language or the difference between Russian and English languages. For final understanding of the question, it is necessary to design the index database not for Simple English Wikipedia, but for English Wikipedia.

6. DISCUSSIONS

The disadvantages of the designed system for indexing wiki texts are as follows:

- the index is created once, if the corpus is updated, then the index should be reconstructed; incremental continuous indexing is necessary;
- if the variable doc_freq_max (limiting the size of the index DB) is assigned equal to

100 (and not 1000, for example), short articles have small number of involvement in the table *term*; i.e., for a small number of words of a given article connectives lexeme--page are specified in the table *term_page*;

a single word form may have several lexemes in the base of the Lemmatizer system (lexemes have different ID), to store this information in the DB БД WikIDF, it is necessary to add one more table.

Conclusions and suggestions for improving the indexing system of wiki texts:

The weight *tf-idf* indicates the significance of the word in the whole corpus of texts; therefore the weight of a word, e.g. "bite", will most likely be small in the corpus of texts on biology. It is reasonable to use categories for refining the weight value. Thus the weight depends on the subject domain of wiki texts and the same word may have different weights in texts with different topics.

For the same goal (refinement of the word weight depending on the text topic), it is reasonable to add to the table term (Fig. 2) the field "variation coefficient D"; i.e., the evaluation of the specificity of a word for a particular subject domain [30, p. 347].

A useful additional resource for indexing is a marked corpus of English Wikipedia [31], which makes it possible to perform search taking into account the semantic markup, e.g., using 45 categories of the upper level of the hierarchy of sets of synonyms of WordNet, assigned to words of Wikipedia.

The planned variants of development the WikIDF system are oriented to, first, inclusion of WikIDF (as one of the components) into the Synarcher program for implementing a full-text (but not only based on headers of wiki pages) search for semantically close words. The WikIDF package is included in the Synarcher program; however this search in this version of Synarcher is performed without access to either the WikIDF package or the index DB. Thus, using the full-text search in wiki texts, we will be able to generate the root set of pages in the adapted HITS-algorithm (AHITS) [18], which, presumably, improves the result of searching for semantically close words. Second, the conversion of the Wiki dictionary into a computer form, first of all, the semantic relations of the Wiki dictionary, will make the search in wiki texts more complete and accurate.

There are alternative methods of designing the index DB, with which we can deal in the future work:

module ANNIC of the GATE system, based on the search engine Lucene [32] (see also в in [33] the description of a variant of dealing with wiki from the GATE system);

indexing system MG4J [34];

tool Lemur with capabilities of indexing (English, Chineese, Arabic) and the search engine INDRI (http://www.lemurproject.org).

The promising lines of investigation connected with the index DB are as follows:

determination of meaning of polysemantic words. In [35] it was suggested that marks of subject domains (e.g., med., archit., sport.) make it possible to find semantic relations between the meanings of words. The results of experiments have shown that polysemantic words are actually determined with a high degree of accuracy for a large number of words because of these marks [35]. For experiments, WordNet Domain (extended version of WordNet, see http://wndomains.itc.it) was used, where each synset contained marks of subject domains;

filtering the text flow [36];

finding keywords with account of semantic relations (synonyms, hyponyms, etc.), e.g., in [37] (based on WordNet or CYC) or wiki dictionary;

evaluation of the accuracy of search by the TF-IDF method, obtaining optimal search parameters for the sake of: a) cutting high-frequency words [38]; b) comparison of similarity measures in the vector space of words [38], and c) with account of the additive model of calculating the relevance of the document to the query [39].

CONCLUSIONS

Not only the Internet grows, but also Wikipedia grows, and by recent data [40] the encyclopedia increases and is improved in the following these directions:

the number of languages in which Wikipedia is maintained;

the number of active participants (with time the number of participants grows, but the relative number of high-activity participants, i.e., those who do more than 100 corrections a month, reduces [40]);

the list of thematic directions (every new group of participants, and each language group has its own interests);

the entire number of articles, and in large Wikipedias, the depth of development (formally, this is the size of an article and the number of corrections);

connectedness of pages (i.e., the number of internal links, interwikis, categories);

"embedding" of Wikipedia in the Internet web by increasing the number of external links.

Search systems and wiki resources are cooperated more and more closely. On the one hand, because of a large number of hyperlinks in wiki texts and in view of the specific features of algorithms based on analysis of links (e.g., PageRank [3]), search engines assign a high rating to wiki texts, i.e., put them at the high positions as a result of search [1]. On the other hand, the search within wiki sites is performed both by using search over DB built in MediaWiki and by specialized Wikipedia search systems: Wikia Search, Lucene-search, FUTEF, and in Russian Wikipedia, by Qwika. Note that the future of search engines will probably be based on distributed search using P2P-applications [41]. Text indexing was and will be the important task of search engines.

In this paper, we considered the architecture and implementation of a software system for indexing wiki texts WikIDF. In indexing, a list of lemmas and the frequencies of their occurrence are calculated by the GATE system, the morphological analyzer Lemmatizer, and the module RussianPOSTagger joining them. With the use of the WikIDF system, index DBs for Russian Wikipedia and Simple English Wikipedia were designed.

The parameters of the source DBs of two Wikipedias were presented: Russian Wikipedia and Simple English Wikipedia. The temporal characteristics of indexing DB were presented, and the quantitative properties of the designed index databases were described. A faster growth of English Wikipedia was detected, namely for five months (September 2007 to February 2008); in Simple English Wikipedia, the rate of growth of the number of articles was greater by 14% and by 7% faster, than in Russian Wikipedia.

ACKNOWLEDGMENT

This work was supported by the Russian Foundation for Basic Research (project no. 08-07-00264) and the Program of Basic Research of Presidium RAS (project no. 213 "Intelligent Information Technologies, Mathematical Simulation, Systems Analysis and Automation ").

REFERENCES

1. L. Rainie and B. Tancer, "Wikipedia Users," in Reports: Online Activities & Pursuits (2007), http://www.pewinternet.org/pdfs/PIP_Wikipedia07.pdf.

2. J. J. Kleinberg, ACM **46** (5) (1999).

3. S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998)," http://www-db.stanford.edu/backrub/google.html.

4. S. Fortunato, M. Boguna, A. Flammini, @et al.,@ "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*, Ed. by M. Berry (Springer, New York, 2003).

6. Y. Ollivier and P. Senellart, "Finding Related Pages Using Green Measures: An Illustration with Wikipedia," in *Association for the Advancement of Artificial Intelligence*, Vancouver, Canada (2007).
7. D. Milne, "Computing Semantic Relatedness Using Wikipedia Link Structure," in *Proceedings of New Zealand Computer Science Research Student Conference (NZCSRSC'2007), Hamilton, New Zealand, 2007*, http://www.cs.waikato.ac.nz/~dnk2/publications/nzcsrsc07.pdf.
8. S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity Flooding: a Versatile Graph Matching Algorithm and Its Application to Schema Matching," in *Proceedings of 18th ICDE Conference, San Jose CA, USA, 2002*, http://research.microsoft.com/~melnik/publications.html.
9. V. Blondel and P. Senellart, "Automatic Extraction of Synonyms in a Dictionary," in *Proceedings of SIAM Workshop on Text Mining, Arlington, Texas, USA, 2002*.
10. V. Blondel, A. Gajardo, M. Heymans, @et al.,@ "A Measure of Similarity Between Graph Vertices: Applications to Synonym Extraction and Web Searching," SIAM Review **46** (1) (2004).
11. E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness, Using Wikipedia-Based Explicit Semantic Analysis," in *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, 2007*, http://www.cs.technion.ac.il/~gabr/papers/ijcai-2007-sim.pdf.
12. M. Sahami and T. D. Heilman, "A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets," in *Proceedings of 15th International World Wide Web Conference (WWW), 2006*, http://robotics.stanford.edu/users/sahami/papers-dir/www2006.pdf.
13. P. Pantel and D. Lin, "Word-for-Word Glossing with Contextually Similar Words," in *Proceedings of ANLP-NAACL 2000, Seattle, USA, 2000*.
14. I. Kuralenok and I. Nekrest'yanov, "Automatic Document Classification Based on Latent--Semantic Analysis," in Proceedings of the Conference on *Electronic Libraries: Promising methods and Technologies, Electronic Collections,* St. Petersburg, Russia, 1999, http://www.dl99.nw.ru [in Russian].
15. K. Bharat and M. Henzinger, "Improved Algorithms for Topic Distillation in a Hyperlinked Environment," in *Proceedings of 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.Proc, 21.
16. A. G. Maguitman and F. Menczer, H. Roinestad, *et al.*, Algorithmic Detection of Semantic Similarity, 2005, http://www2005.org/cdrom/contents.htm.
17. A. A. Krizhanovskii, "Automated Search of Semantically Close Words by the Example of Aviation Terminology," Avtomatizatsiya v Promyshlennosti, **64** (4), (2008).
18. A. A. Krizhanovsky, "Synonym Search in Wikipedia: Synarcher," in *Proceedings of the 11th International Conference on Speech and Computer SPECOM'2006, St. Petersburg, Russia, 2006*.
19. A. A. Krizhanovskii, "Evaluation of Search Results of Semantically Closw Words in Wikipedia: Information Content and the Adapted HITS Algorithm," in *Proceedings of Wiki Conference, St. Petersburg, Russia, 2007*, [in Russian].
20. I. V. Segalovich, "How Search Engines Operate," 2004, http://company.yandex.ru/articles/.
21. S. Robertson, "Understanding Inverse Document Frequency: on Theoretical Arguments for IDF," J. Documentation, No. 60 (2004).
22. H. Cunningham, D. Maynard, K. Bontcheva, *et al., Developing Language Processing Components with GATE (User's Guide)*, Technical report. University of Sheffield, UK, 2005.
23. A. V. Sokirko, "Morphological Modules at Site www.aot.ru," in *Proceedings of International conference Dialog 2004 on Computer*

Linguistics and Intelligent Technologies, Moscow, Russia, 2004, [in Russian].

24. D. Vakhitova, "Development of a Corpus of Texts on Corpus Linguistics, 2006, http://matling.spb.ru/files/kurs/Vahitova_Corpus.doc.

25. J. E. F. Friedl, *Regular Expressions* (Piter, St. Petersburg, 2001) [in Russian].

26. S. P. Ponzetto and M. Strube, "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, 2007*.

27. T. Zesch, C. Mueller, and I. Gurevych, "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary," in *Proceedings of Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco, 2008*.

28. C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing* (The MIT Press, 1999).

29. S. Campbell, J.-P. Chancelier, and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos* (Springer, 2006).

30. O. N. Lyashevskaya and S. A. Sharov, "Frequency Dictionary of the National Corpus of Russian Language: Concept and Technique for Development," in *Proceedings of International Conference Dialog 2008 on Computer Linguistics and Intelligent Technologies, Bekasovo, Russia, 2008*, http://www.dialog-21.ru/dialog2008/materials/pdf/53.pdf.

31. J. Atserias, H. Zaragoza, M. Ciaramita, *et al.*, "Semantically Annotated Snapshot of the English Wikipedia," in *Proceedings of Conference on Language Resources and Evaluation, Marrakech, Morocco, 2008*.

32. N. Aswani, V. Tablan, K. Bontcheva, *et al.*, "Indexing and Querying Linguistic Metadata and Document Content," in *Proceedings of RANLP'2005, Borovets, Bulgaria, 2005*.

33. R. Witte and T. Gitzinger, "Connecting Wikis and Natural Language Processing Systems," in *Proceedings of WikiSym'07, Canada, Quebec, 2007*, http://www.wikisym.org/ws2007/_publish/Witte_WikiSym2007_NaturalLanguageProcessing.pdf.

34. P. Boldi and S. Vigna, Efficient Optimally Lazy Algorithms for Minimal-Interval Semantics (2007), http://vigna.dsi.unimi.it/papers.php.

35. B. Magnini, C. Strapparava, G. Pezzulo, *et al.*, "The Role of Domain Information in Word Sense Disambiguation," J. Natural Language Engineering **4** (8) (2002).

36. A. Smirnov and A. Krizhanovsky, "Information Filtering Based on Wiki Index Database," in *Proceeding of FLINS'08, Madrid, Spain, 2008*, http://arxiv.org/abs/0804.2354.

37. M. Shamsfard, A. Nematzadeh, and S. Motiee, "ORank: An Ontology Based System for Ranking Documents," Int. J. Comput. Sci. **3** (1) (2006).

38. M. Meyer, C. Rensing, and R. Steinmetz, "Categorizing Learning Objects Based on Wikipedia as Substitute Corpus," in *Proceedings of LODE'07, Crete, Greece, 2007*, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-311/paper09.pdf.

FIGURE CAPTIONS

Fig. 1. Architecture of the indexing system of wiki texts POS (part of Speech)

Key:

1. Вход->input
2. Язык, Параметры БД, TD-IDF ограничения->Language, DB parameters, TD-IDF constraints
3. Приложения индексирования Википедии->Application for indexing Wikipedia
4. Управляющее приложение-> Controlling application
5. Извлечение текстов из Википедии->Extraction of texts from Wikipedia
6. Анализ текста (порождение лемм)->Analysis of texts (generation of lemmas)
7. Сохранение (лемм, частот)->Saving (lemmas, frequencies)

8. Обработчик Википедии->Wikipedia handler
9. Извлечение страниц вики->Extraction of wiki pages
10. Преобразование в текст->Conversion into text
11. БД->DB
12. Приложение->Application

Fig. 2. Tables and relations in the index DB WikIDF
Key:
1. Леммы слов, найденных в вики-текстах, число документов с леммой, частота леммы в корпусе->Lemmas of words, the number of documents with a lemma, and the frequency of the lemma in the corpus
2. Страницы, содержащие словоформы данной леммы, и леммы словоформ, принадлежащих данной странице->Pages containing word forms of a given lemma, and lemmas of word forms belonging to a given page
3. Заголовки вики-страниц, число слов на странице->Headers of wiki pages, the number of words in a page

Fig. 3. Linear dependence of the decrease of the frequency of word usage in the corpus of the serial number (rank) of the word in the word list, arranges according to frequency, in the scale of logarithm--logarithm for Russian Wikipedia (ruwiki) and Simple English Wikipedia (simplewiki) on February 2008, linear approximation based on 100 and 10 thousands of the most frequently used words.

Key:

1. Частота->Frequency
2. Номер слова->Word number

TABLES

Table 1. Decision on parsing a wiki text

Key:

1. Вопросы->Questions
2. Ответы->Replies
3. Исходный текст->Source text
4. Преобразованный текст->Converted text
5. Заголовки (подписи) рисунков->Headers (captions) of figures
6. *Оставить (извлечь)->Retain(extract)*
7. Интервики->Interwikis
8. *Оставить или удалить (определяется пользователем->Retain or delete (it is determined by the user*
9. Названия категорий->Category names
10. *Удалить->Delete*
11. регулярное выражение->regular expression

12. Категория->Category

13. Шаблоны, цитаты, таблицы->Templates, quotations, tables
14. *Удалить ->Delete*
15. Курсив и "жирное" написание->Italic and "bold"
16. Апострофы удаляются->Apostrophes are deleted

17. Внутренняя ссылка->Internal link

18. *Оставить текст, видимый пользователю, удалить скрытый текст->Retain the text visible to the user, delete the hidden text*
19. в [[космос|космическом пространстве]]-> in [[space|cosmic space]]
20. *в космическом пространств->in space.*
21. внутренняя ссылка без вертикальной черты->internal link without vertical bar
22. Внешняя ссылка->External link
23. *Оставить текст, видимый пользователю, удалить сами гиперссылки->Retain the text input by the user*
24. сайт->site
25. фан-сайт->fan-site
26. *сайт – фан-сайт->site–fan-site*
27. Имя сайта (без пробелов), содержащее точку '.' хотя бы раз, кроме последнего символа-> Site name (without blanks) containing the dot '.' at least once, except for the last symbol

Table 2. Example of conversion of a wiki text

Key:

1. Исходный текст в вики-разметке->Source text in wiki markup
2. *Преобразованный текст->converted text*
3. {{Фильм | РусНаз = Через тернии к звездам }}->{{Film | RusName = Though thorns to the stars }}
4. [[Изображение:Через-тернии-к-звездам 2.jpg|thumb|Через тернии к звездам]]-> [[Image:Through-thorns-to-the stars 2.jpg|thumb|Through thorns to the stars]]
5. *Через тернии к звездам-> Through thorns to the stars*
6. "'"Через тернии к звездам"' [[научная фантастика|научно-фантастический]] двухсерийный фильм [[режиссер]]а [[Викторов, Ричард Николаевич|Ричарда Викторова]] по сценарию [[Кир Булычсв|Кира Булычева]]-> "'" Through thorns to the stars "' [[science fiction| science-fiction]]diserial film [[producer]]a [Viktorov, Richard Nikolaevich|by Richard Victorov]] by scenario [[Kir Bulychev|by Kir Bulychev]]. .
7. *Через тернии к звездам научно-фантастический двухсерийный фильм режиссера Ричарда Викторова по сценарию Кира Булычева ->Through thorns to the stars science-fiction diserial film by producer Richard Victorov, scenario, by Kir Bulychev.*
8== Сюжет == ->== Plot ==
8. == Сюжет == ->== Plot ==
9. {{plot}}
[[XXIII]] век. [[Звездолет]] дальней разведки обнаруживает в [[космос]]е погибший корабль неизвестного происхождения, на нем - гуманоидных существ, искусственно выведенных путем клонирования. Одна девушка оказывается жива, ее доставляют на [[Земля (планета)|Землю]], где [[ученый]] Сергей Лебедев поселяет ее в своем доме. XXIII век. Звездолет дальней разведки обнаруживает в космосе погибший корабль неизвестного происхождения, на нем гуманоидных существ, искусственно выведенных путем клонирования. Одна девушка оказывается жива, ее доставляют на Землю, где ученый Сергей Лебедев поселяет ее в своем доме. ->[[XXIII]] century. [[Spacecraft]] of remote reconnaissance finds in [[space]] a dead spacecraft of unknown origin, and in it they find humanoid creatures, artificially bred by cloning. A girl appears to be alive, she is taken to the Earth [[the Earth (planet)|to the Earth]], where [[a scientists]] Sergei Lebedev settles her in his house.

10. *[[XXIII]] век. [[Звездолет]] дальней разведки обнаруживает в [[космос]]е погибший корабль неизвестного происхождения, на нем - гуманоидных существ, искусственно выведенных путем клонирования. Одна девушка оказывается жива, ее доставляют на*

[[Земля (планета)|Землю]], где [[ученый]] Сергей Лебедев поселяет ее в своем доме. XXIII век. Звездолет дальней разведки обнаруживает в космосе погибший корабль неизвестного происхождения, на нем гуманоидных существ, искусственно выведенных путем клонирования. Одна девушка оказывается жива, ее доставляют на Землю, где ученый Сергей Лебедев поселяет ее в своем доме. ->XXIII century. Spacecraft of remote reconnaissance finds in space a dead spacecraft of unknown origin, in it, humanoid creatures, artificially bred by cloning. A girl appears to be alive, she is taken to the Earth, where a scientists, Sergei Lebedev, settles her in his house.
11. == В ролях == ->== Roles are played ==
11. *== В ролях == ->== Roles are played ==*
*12. * [[Елена Метулкина]] – "Нийя"->. * [[Elena Metulkina]] – "Niiya"*
13. ** Елена Метулкина Нийя-> Elena Metulkina Niiya*
14. == Ссылки ==->Links
14. == Ссылки ====->Links
15. {{викицитатник}}->{{wikicitations}}
*16. * [http://ternii.film.ru/ Официальный сайт фильма]-> * [http://ternii.film.ru/ Official movie site]*
17. ** Официальный сайт фильма-> Official film site*
18. [[Категория:Киностудия им. М. Горького]] ->Category: M. Gor'kii film studio [[en:Per Aspera Ad Astra (film)]]