

Establishing Semantic Similarity of the Cluster Documents and Extracting Key Entities in the Problem of the Semantic Analysis of News Texts

Anastasia Nikolaevna Soloshenko¹, Yulia Aleksandrovna Orlova¹, Vladimir Leonidovich Rozaliev¹ & Alla
Viktorovna Zaboileeva-Zotova²

¹ Computer-aided design and search engine design (CAD and SED) Department, Volgograd State Technical University, Volgograd, Russian Federation

² Russian Foundation for Basic Research, Moscow, Russian Federation

Correspondence: Vladimir Leonidovich Rozaliev, Volgograd State Technical University, Lenin Ave., 28, Volgograd, 400005, Russian Federation. Tel: 7-917-336-6988. E-mail: vladimir.rozaliev@gmail.com

Received: December 15, 2015

Accepted: December 20, 2015

Online Published: April 7, 2015

doi:10.5539/mas.v9n5p246

URL: <http://dx.doi.org/10.5539/mas.v9n5p246>

This paper received partial support from the Russian Foundation for Basic Research (projects No. 13-07-00459, 13-07-97042, 13-07-00351, 14-07-97016, 14-07-97017).

Abstract

This paper is dedicated to the problem of establishing semantic similarity for the documents of the news cluster and extracting key entities from the article's text. The existing methods and algorithms for fuzzy duplicate detection texts are briefly reviewed and analysed, such as TF-IDF and its modifications, Long Sent, Megashingles and Log Shingles, and Lex Rand. The shingles algorithm essence and its main stages are described in detail. Several options of the parallel implementation for the shingles algorithm are presented: for multiprocessor heterogeneous computing systems using CUDA and Open CL and for distributed computing systems using Google App Engine. The parameters of the algorithm (operation time, acceleration) applied to the problem of the semantic analysis for news texts are assessed. In addition, the methods and algorithms for extracting key phrases from the news text are reviewed: graph methods, in particular TextRank, building horizontal visibility graphs, the Viterbi algorithm, types of Markov random fields method, as well as a comprehensive context-sensitive algorithm for news text analysis (a combination of statistical algorithms for extracting key words and algorithms for forming semantic coherence of the text blocks). These methods are analysed from the standpoint of applicability to the news articles analysis. Particular attention is paid to the peculiarities of the news text structure. Although the thematic classification and selection of key entities in text documents are powerful text processing tools, these stages of analysis cannot give a complete picture of the news piece semantics. The paper presents a methodology and a comprehensive analysis of news text, based on a combination of semantic analysis and subsequent text abstracting submitting it in a compressed format - so-called mind map.

Keywords: news text, semantic similarity, fuzzy duplicates, shingles, key entities, text graph, semantic analysis, annotation, mind map

1. Introduction

1.1 Introduce the Problem

The problem of establishing semantic similarity for cluster documents and selection of entities that make up the information structure of the text, is one of the most important and difficult problems in the web data analysis and information retrieval on the Internet. The urgency of this problem is determined by a variety of applications requiring contemplation of the news documents semantic component. This involves improving the quality of archives belonging to search engines by removing redundant information, and associating news reports in the stories based on similarity in content of these messages in the semantic analysis task, spam filtering (in e-mail, search engine), establishment of copyright infringement in the illicit copying of information (the problem of

plagiarism or copyright), and several others. However, this paper focuses on the problem of news reports unification in the stories with subsequent news annotating.

1.2 Importance of the Problem

The main obstacle to the successful solution of this task is the huge volume of data stored in databases of modern search engines. This volume makes its "direct" solution by pairwise comparison of text documents into a single aggregation of news with the release of key objects virtually impossible (in a reasonable time).

Unstructured data make up large amount of the information. Thus efficient technologies for automated analysis of the information provided in natural language, are of particular interest not only for many organizations (news feeds, information and library systems, etc.), but also for individuals (network users). In this regard, it is necessary to research the structure of the news text, as well as the methods for its analysis. The big part of attention is paid to the development of methods on reduction the computational complexity of the created algorithms.

1.3 Relevant Scholarship

If talking about the fuzzy duplicate detection algorithms for texts (in other words, methods for determining the semantic similarity of documents), one of the most popular algorithms is the "shingles algorithm" (discussed in detail in Sec. 2.1.5). The initial research in the direction of fuzzy duplicate search may include papers by U. Manber and N. Heintze. The similarity measure for the two documents is the ratio of the common substrings number to the size of the file or the document. In 1997, A. Broder offered new "syntactic" method for assessing the similarity between documents based on the representation of the document as a set of all possible sequences of a fixed length k , consisting of adjacent words. These sequences were called "shingles". Another signature-based approach is no longer on the syntactic, but on lexical principles was proposed by A. Chowdhury in 2002 and improved in 2004. At present, the algorithm is widely applied in the search engines (Martinez-Gil Jorge, 2012).

It is possible to say that in terms of key words allocation from the text, abstracting and semantic analysis, a great contribution to the development of computational linguistics in Russia was made by Apresian Yu. D., Braslavskiy, P., Lande, D., Tarasov, S. D., Gorodetskiy, B. Yu., Kobozeva, I. M., Malkovskiy, M. G., Melchuk, I. A., Narinyani, A. S., Paducheva, E. V., Popov, E. V., Preobrazhenskiy, A. B., and others.

In addition, now, there is a great amount of semantic analysis of texts and various news aggregators. Domestic include TextAnalyst, Content Analyzer, technologies of AOT, RCO, MediaLingua annotator, Yandex News; foreign include Extractor, QDA Miner (WordStat and Simstat packages), systems of Inxight Summarizer (component of AltaVista search index), Intelligent Text Miner (IBM), MEAD, NetSum, Newblaster and others. Furthermore, in the United States, there are already machines (robots) writing news based on the analysis of data from multiple sites. The examples include the project of QuakeBot creating notes about earthquakes, the project of Mapping LA publishing notes based on crime reports. However, bots are able to work only with specific sets of data: the competition results, business performance, stock indices, simply filling in the blanks in the ready-made phrases.

It may be noted that the major systems developed in the West focus exclusively on processing of Western languages, which makes them unsuitable for analysis of texts in Russian. Moreover, few systems are focused on news texts processing. These are the components of the search mechanisms in various systems, but they are implemented with the help of automatic abstract methods for the news clusters (groups of texts with the given topic) rather than individual text selected by the user (Huang et al., 2014).

1.4 State Hypotheses and Their Correspondence to Research Design

In summary, the main problem of establishing the semantic similarity of documents and search of key facts with regard to their connection in the news text is the lack of models and methods that provide an adequate formalization and semantic analysis of news texts in natural language. Explanation for this is the complexity and ambiguity of the solution for the problem of semantic analysis for different types of texts and the formalization complexity of the Russian language.

This paper offers an approach to the development of software for the analysis of news texts, combining the aggregation of news articles (using the shingles method) with comprehensive analysis of the text.

The paper is devoted to a number of problems:

a) Development of a model representation of the news text, taking into account the semantic components based on the identified structural features of the news;

- b) Review and analysis of existing methods for determining semantic similarity of news texts and extracting key entities;
- c) Parallel shingles algorithm implementation using various technologies (CUDA, OpenCL, and Google App Engine);
- d) Development of news aggregation algorithm, which includes the definition of articles relevancy to topics and presenting consolidated text news as a mind map.

2. Method

Method of complex automated analysis of text news is presented in Figure 1. General aspects have been considered in previous researches (Soloshenko et al., 2014a; Soloshenko et al., 2014b), this paper covers the last two stages - the semantic analysis of documents and texts and building a network of appropriate methods and algorithms.

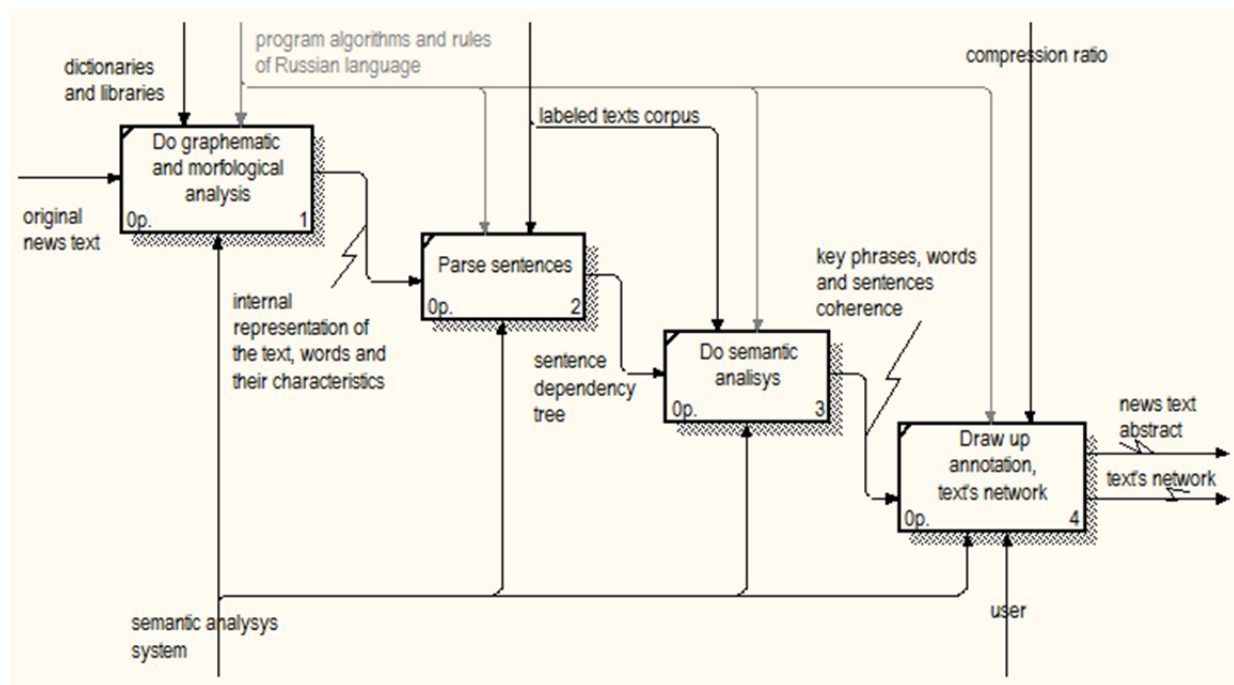


Figure 1. The process of news text semantic analysis

2.1 Methods for Establishing Semantic Similarity of Texts

There are many methods and algorithms for fuzzy duplicate texts detection (in other words, methods for determining the semantic similarity of documents). As part of the paper, such methods as TF-IDF and its modifications, Long Sent, Megashingles and Log Shingles, Lex Rand were considered (Zakharov, V. N., & Khoroshilov, A. A., 2012). However, the shingles algorithm and aspects of its implementation were considered in detail.

2.1.1 TF-IDF and Its Modifications

The idea of this algorithm and some of its modifications are similar: the entire collection is used to build a dictionary, putting every word in correspondence with a number of documents in which it occurs at least once (df) and then the average length of the document (dl_avg) is determined.

Then the frequency dictionary of the document is formed, and "weight" wt for each word is calculated using the following formula:

$$wt = TF \times IDF \quad (1)$$

where

$$TF = \frac{tf}{2 \cdot (0.25 + 0.75 \frac{dl}{dl_{avg}}) + tf} \quad (2)$$

$$IDF = \log \frac{N - df + 0.5}{df + 0.5} \quad (3)$$

Then 6 words with the highest wt are selected and bonded in alphabetical order in the string. As the signature of the document, CRC32 checksum is calculated for the resulting string (Anisimov et al., 2011).

2.1.2 Long Sent

The document is divided into sentences ordered by descending length, expressed in number of words; in case of equal lengths - in alphabetical order. Then the two longest sentences are selected and bonded into a string in alphabetical order. The signature of the document is the calculated CRC32 checksum of the resulting string.

2.1.3 Megashingles and Log Shingles

In the Megashingles algorithm, 36 shingles, minimizing the value of the corresponding functions, are selected. Then, 36 shingles are divided into 6 groups of 6 and 6 "supershingles" are determined. They are used to make up all sorts of combinations of pairs, called "megashingles". The number of such combinations from 6 of 2 equals to 15. All 15 of these megashingles constitute signature of the document which gave 33% fullness for the threshold difference of 0.75 to 0.80 42% and 50% for 0.85. ($p^6 \sim 0.80^6 \sim 0.26$ (26%); $1 - (1 - 0.26)^6 - 6 \cdot 0.26 \cdot (1 - 0.26)^5 \sim 0.42$ (42%)).

Log Shingles is based on the "supershingling" of logarithmic sampling of the original full set of shingles, leaving shingles divisible by small number's powers. First, the calculation is made for the set of all 5-word shingles of words, which "wrap" to the beginning at the end of the document. Then, shingles divisible by powers of 2 are selected from this set. They constitute the exact signature of the document.

2.1.4 Lex Rand

First, the entire collection is used to build a dictionary, similar to that used in the algorithm A2 from which the words with the highest and lowest IDF values are removed. Then, based on that dictionary, 10 additional dictionaries are generated containing approximately 30% fewer words than the original one. Words are removed randomly. Each document has 11 I-Match signatures built.

Duplicates are the documents with at least one matched signature. It turns out that such an approach sufficiently, in comparison with A2 (more than 2-fold), exceeds the completeness of duplicate detection at reduction of relative precision only by 14% (Zelenkov, Yu. G., & Segalovich, I. V., 2007).

2.1.5 Shingles Algorithm

The algorithm used to compare the texts will be considered in more detail. The algorithm consists of five steps:

1) Text canonization

Text canonization leads the original texts to the unified normal form. The text is cleared of prepositions, conjunctions, punctuation, HTML tags and other unnecessary "junk", which shall not take part in comparison. In most cases, it is offered to remove adjectives, since they carry no semantic value. It is possible to reduce nouns to the nominative case, singular, or leave only their roots.

2) Segmenting into shingles

Shingles are flakes, subsequence of words extracted from articles.

It is necessary to allocate 10 pieces (shingle length) of subsequence of words following each other from the comparable texts. The sample occurs as an overlapping, rather than back-to-back.

Thus, breaking the text on the subsequence, one obtains a set of shingles in an amount equal to the number of words minus the length of shingles plus one ($\text{words_No.} - \text{shingle_length} + 1$).

Actions for each of the items are made for each of the compared texts, the algorithm is shown in Figure 2.

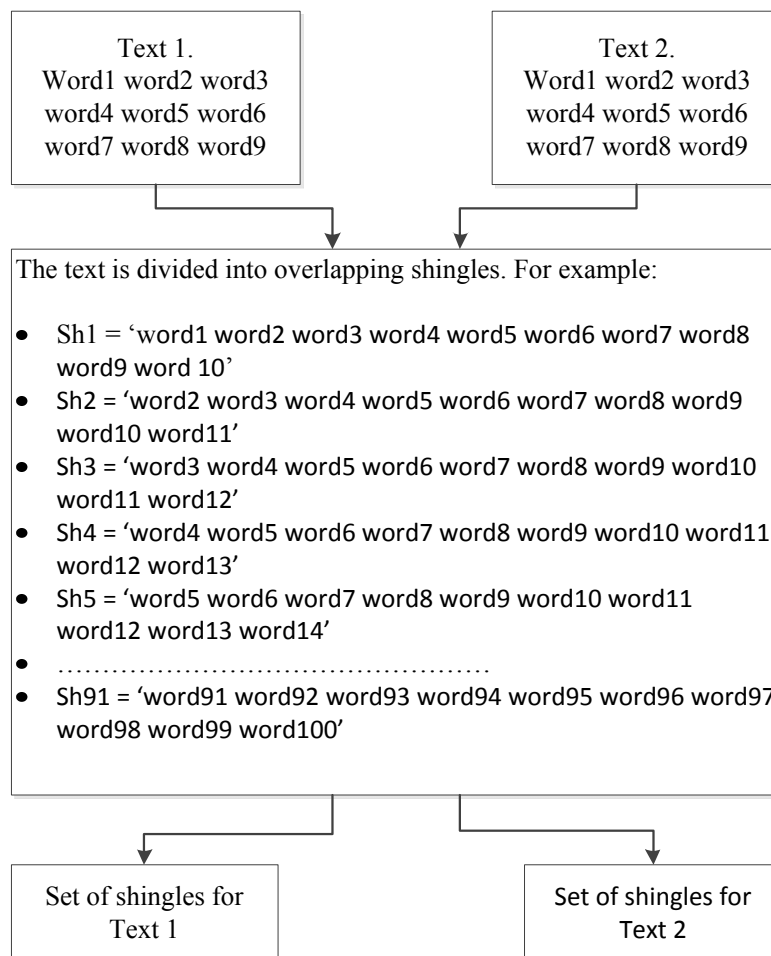


Figure 2. Segmenting text into shingles

3) Calculating shingles hashes using 84 static functions

The principle of the shingles algorithm is to compare the random checksum shingles (subsequences) samples of the two texts between each other.

The problem of the algorithm is the number of comparisons, because it directly affects the performance. Increased number of shingles to compare is characterized by exponential growth of operations having critical impact on performance. This stage can be parallelized.

It is offered to submit the text as a set of checksums calculated through 84 internally unique static hash functions.

84 checksum values are calculated for each shingle through the different functions (e.g., SHA1, MD5, CRC32, etc., 84 functions in total). So, each of the texts will be presented in the form of a two-dimensional array of 84 rows where each row describes the proper function of the 84 checksum functions, as shown in Figure 3.

These sets will be used for random selection of 84 values for each of the texts. These values will be compared with each other according to the checksum function, through which each of them was calculated. Thus, the comparison will require 84 operations.

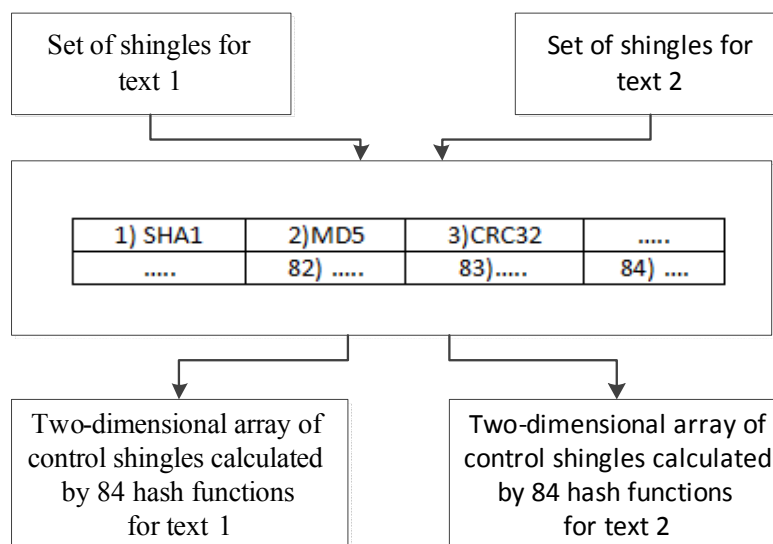


Figure 3. Shingles checksum calculation

4) A random sample of 84 checksum values

Let then choose the most minimum value of each string (Figure 4). The output is the set have minimum shingles checksum values for each of the hash functions.

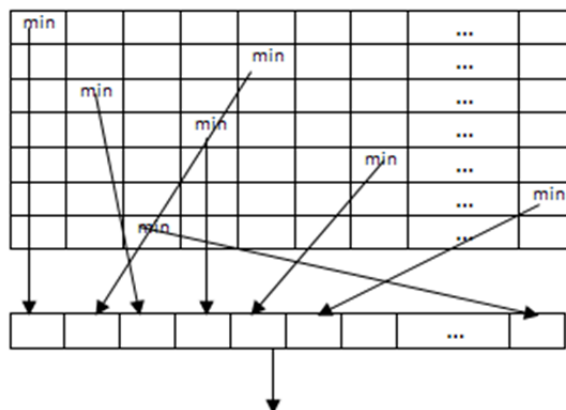


Figure 4. The choice of the minimum checksum values

5) Comparison, the determination of the result (Code Is Art, electronic source)

Further, the paper will present the software implementation of the algorithm with parallelization (para. 3.1 and 3.2).

2.2 Methods to Extract Key Phrases from the News Text

2.2.1 Graph Methods

a) The first considered method is TextRank - the application of the PageRank algorithm for the problems of natural language processing.

The basic idea is to perform the three steps:

1. Building the graph based on the original text in natural language;
2. Approximate calculation of PageRank values to build a graph;
3. Application of obtained vertex weights to retrieve information from the text.

1) Building the graph Weighted undirected graph will be built in the form of $G = (V, E)$, where V is a set of words, E is a set of connections between them.

The set of all unique lemmas of the original text may be taken as V . Since the most of the terms are the named groups, the number of words shall be limited only to lemmas formed from nouns and adjectives.

The E set is built by sequential scanning of the text by the specified window of $N \in [2, 10]$ words. At each iteration, connection value $WC(\omega_1, \omega_2)$ is calculated for a pair of words, which reversely depends on the distance between the words:

$$WC(\omega_1, \omega_2) = \begin{cases} 1 - \frac{d(\omega_1, \omega_2) - 1}{N - 1}, & \text{если } d(\omega_1, \omega_2) \in (0, N) \\ 0, & \text{если } d(\omega_1, \omega_2) \geq N, \end{cases} \quad (4)$$

where ω_1 и ω_2 is the words, $d(\omega_1, \omega_2)$ is the distance between words, N is the window size.

Words for which the value $WC(\omega_1, \omega_2)$ equals to zero are not included in the set of graph vertices. In order to determine the distance between two words, it is sufficient to use a positional measure (Wei et al., 2010):

$$d(\omega_1, \omega_2) = |p(\omega_1) - p(\omega_2)|, \text{ если } \omega_1 \neq \omega_2 \quad (5)$$

where ω_1 и ω_2 is the words, $p(\omega)$ is the serial number of words ω in the text.

The basis for calculating the value $WC(\omega_1, \omega_2)$ is the observation that there is a semantic relation for the two words often standing side by side. This is necessary to ensure the consistency of the text representation in a graph. The higher the distance $d(\omega_1, \omega_2)$, the lower the probability of the existence of such a relation.

When processing a graph, the work is carried out exclusively with single nouns and adjectives. The combination of these words in word groups will be performed at the stage of building word groups.

2) Ranking the graph's vertices. After generating the G graph, it is necessary to calculate TextRank - the value of the stationary distribution of the random walk for each vertex $t \in V$, taking into account the weights of connections:

$$TR(t_i) = (1 - d) + d \cdot \sum_{t_j \in In(t_i)} \frac{\omega_{ji}}{\sum_{t_k \in Out(t_j)} \omega_{jk}} \cdot TR(t_j) \quad (6)$$

where d is attenuation factor, $In(t)$ is the set of vertices belonging to t , $Out(t)$ is the set of vertices emanating from t , ω_{ji} is weight of the edge (t_i, t_j) . An undirected graph takes $In(t) \equiv Out(t)$.

After calculating TextRank, it is necessary to make the C set of candidates in terms of the first T words from the list of vertices, ordered in descending order of TextRank. There are several approaches to the definition of $T \equiv |C|$:

take a constant value of T , use $T = \frac{1}{3} |V|$, etc. (Dongmei Ai et al., 2010).

3) Word groups assembly All sequences of words consisting of C set elements must be extracted from the text. Since $\forall t \in C : \exists ! T R(t) \in R$, then the total weight of the extracted sequence is the sum of the weights of all its constituent words.

When assembling word groups, it is necessary to consider the two points: 1) the sequence must have at least one noun; 2) at detecting embedding of one sequence into another sequence, only the sequence with great weight is considered.

The selected sequences of one or more words can be seen as the terms of the original text (Ustalov, D. A., 2012), (Sheng, T. L., & Fu, Ch. T., 2010). Figure 5 shows an example of such text graph:



Figure 5. Text key words representation as a graph

b) The optimal graph method for work with the news texts is Horizontal Visibility Graph (HVG). This approach also allows building a network structure based on the texts in which individual words or word groups are mapped in a special way by numerical weight values. The function associating with the word number can be, for example, the serial number of a unique word in the text, word length, "weight" of words in texts, common TFIDF assessment (in the canonical form, equal to the multiplication of the word frequency in the text fragment - term frequency - by the binary logarithm of the value reciprocal to the number of fragments in the text where the word is encountered - inverse document frequency) or its variants, as well as other weight assessments.

As TFIDF weighted assessment of the full text consisting of N words, the text is segmented into fragments, comprising a predetermined number of words - M (e.g., $M = 500$). Then, for each word i , included in the text, shall have the number of fragments $df(i)$ counted, in which this word is included, as well as the total number of occurrences of the word in the text $i - n(i)$. Thereafter, according to the formula

$$tfidf(i) = \frac{n(i)}{N} \log \left(\frac{N}{M \times df(i)} \right) \quad (7)$$

the average TFIDF value of the weighted assessment for each word is calculated (Lande et al., 2013).

When building a network of words, dispersion assessment of the words importance is used. The result is the graph shown in Figure 6:

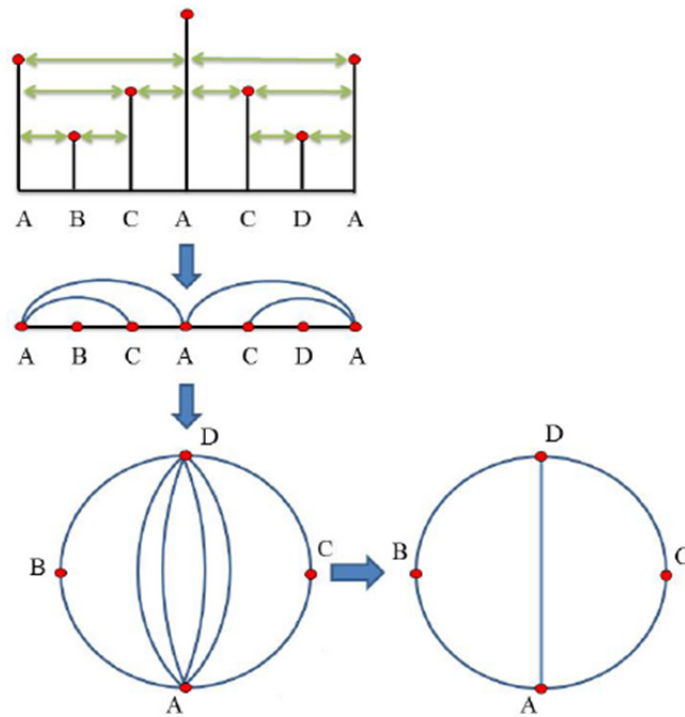


Figure 6. Building horizontal visibility graph

2.2.2 The Viterbi Algorithm

The Viterbi algorithm is a dynamic programming algorithm that searches for the most appropriate list and receives the most likely sequence of events. The algorithm was named in honour of the founder - American engineer of Italian origin Andrew Viterbi.

This algorithm is often used in Natural Language Processing, involving the use of hidden Markov models. Examples of such tasks:

1. POS-tagging;
2. named entity recognition.

Further represents a description of the Viterbi algorithm for hidden Markov models of the second order. For ease of understanding, the terminology of tasks for automatic mapping out parts of speech is used. The following designations are used: $x_1 \dots x_n$ is a sentence, $*$ is a symbol of the beginning of the sentence, $STOP$ is a symbol of the end of the sentence, S is the set of all possible tags, $q(s|u, v)$ is the probability of transition to state s at the previous state (u, v) , $e(x|s)$ is the probability of assigning tag $s \in S$ to a word x (Vasiliev, V. G., 2011).

The input of the algorithm contains the sentence $x_1 \dots x_n$, parameters $q(s|u, v)$ and $e(x|s)$.

The Viterbi algorithm:

```

for  $k = 1 \dots n$ 
  for  $u \in S_{k-1}, v \in S_k$ 
     $\pi(k, u, v) \leftarrow \max_{w \in S_{k-2}} (\pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v))$ 
     $bp(k, u, v) \leftarrow \arg\max_{w \in S_{k-2}} (\pi(k-1, w, u) \times q(v|w, u) \times e(x_k|v))$ 
 $(y_{n-1}, y_n) \leftarrow \arg\max_{u, v} (\pi(n, u, v) \times q(STOP|u, v))$ 
for  $k = (n-2) \dots 1$ 
   $y_k \leftarrow bp(k+2, y_{k+1}, y_{k+2})$ 

```

Output is the sequence of tags $y_1 \dots y_n$.

2.2.3 Types of Markov Random Fields Method

CRF (Conditional Random Fields) method is a type of Markov random fields' method. This method has been widely used in various fields of artificial intelligence, in particular, it is successfully used in text processing tasks.

Markov random field or the Markov chain is the graph model used to represent a set of joint distributions of several random variables. Formally, a Markov random field consists of the following components:

1. undirected graph or a factor graph $G = (V, E)$, where each vertex $v \in V$ is a random variable X , and each edge $\{u, v\} \in E$ represents a dependence between random variables u and v .
2. set of potential functions or factors $\{\phi_k\}$, one for each complete subgraph (it is complete subgraph of an undirected graph G) in the graph. The function ϕ_k puts every possible state of complete subgraph elements association with a certain non-negative real number (Soledad Pera Maria & Yiu-Kai Dennis Ng, 2012).

Non-adjacent vertices shall comply with conditionally independent random variables. Group of adjacent vertices forms a complete subgraph, a set of vertices states is an argument for the corresponding potential function. Set of input lexemes $X = \{x_t\}$ and many corresponding types $Y = \{y_t\}$ combine to form a set of random variables $V = X \cup Y$. In order to solve the task of extracting information from text, it is sufficient to determine the conditional probability $P(Y | X)$.

Then the linear conditional random field is the probability distribution in the following form:

$$p(y | x) = \frac{1}{z(x)} \prod_k \exp(\sum_k \lambda_k f_k(y_y, y_{t-1}, x_t)) \quad (8)$$

where $\exp(\sum_k \lambda_k f_k(y_y, y_{t-1}, x_t))$ is the potential function, $\sum \{\lambda_k\}$ is a real parametric vector,

$\sum \{f_k(y_y, y_{t-1}, x_t)\}$ is a set of feature functions.

The disadvantage of the CRF approach is the computational complexity of the training sample analysis, which makes continuous update of the model difficult when new training data arrives. However, the implementation of the CRF algorithm has good speed, which is very important when dealing with large volumes of information (Leonova Yu. V. & Fedotov A. M., 2013).

To date, the CRF method is the most popular and accurate way to retrieve objects from the text. For example, it was implemented in the project by Stanford University called "Stanford Named Entity Recognizer".

2.2.4 Context-Sensitive News Text Analysis

1) Contexts of using words

For context of the words, the sentences are segmented into fragments between punctuation marks. The following types of contexts in such fragments are distinguished (Zaboleeva-Zotova, A. V. & Orlova, Yu. A., 2010):

- adjacent adjective or noun to the right or left of the original word (Near);
- fragments, which contain verbs, have adjectives and nouns fixed between them and the original word is a verb (AcrossVerb);
- adjectives and nouns found in the fragments of sentences with the given word, not separated by a verb and not adjacent to the original word (NotN).

In addition, all the adjectives and nouns are distinguished by the remembered words occurring in neighbouring sentences (NS). Sentences for the calculation of this index are taken not in full. Sentence fragments from the beginning to fragment containing the verb (inclusively) are considered only. This allows extracting from the most significant words from the neighbouring sentences.

2) Assembling wordy expressions

An important basis for extracting wordy expressions from the text of the document is the frequency of their occurrence in the text. A cluster is a structure in which many words strings are repeated. Therefore, the main criterion for selection wordy expressions is a significant excess of words occurrence directly next to each other

in comparison with separate occurrence in the sentence fragments:

$$Near > 2 \cdot (AcrossVeb + NotN) \quad (9)$$

In addition, restrictions on the frequency for words occurrence next to each other are used.

View of matching pairs of words (expressions) for bonding is performed in the order of coefficient reduction

$$\frac{Near}{AcrossVeb + NotN}.$$

When finding a suitable pair of words, they are bonded together into a single object, and

contextual relationships are recalculated (Alekseev, A. A. & Lukashevich, N. V., 2011).

3) Features to determine the semantic relationships

In order to determine the semantically related expressions and subsequent construction of topic-based units, a set of six basic similarity features is used.

Context-dependent features:

a) The number of entries in the neighbouring sentences (Neighbouring Sentence Feature, NSF). Based on the hypothesis of global connectivity of natural language texts and its result that the elements of one topic-based unit appear in the neighbouring sentences of certain source documents more often than in the same sentences.

The feature provides a numerical assessment of the occurrences ratio in neighbouring sentences (NS feature) in relation to the number of occurrences in one and the same sentence of the original corpus (AcrossVerb, Near and NotNear features). This feature is based on the following ratio:

$$C = NS - 2 \cdot (AcrossVeb + Near + NotNear) \quad (10)$$

The general formula of the contribution for NSF feature has the following form:

$$NSF = \min \left[0.5, \frac{C}{\text{Avg}(C)} \right] \quad (11)$$

where AVG (C) is the average value of C among all positive values in the entire cluster.

b) Strict Context (SC). This feature is based on a comparison of the strict context of the word use - text templates. Templates are the 4-grams: two words on either side of the expression under review. The more identical templates are shared by the pair of candidates, the greater the similarity on this feature. .

Weight for the strict context template is calculated as follows: each word of the context template n-gram has a weight of 0.25. For example, n-gram (*, * consists, of) will have a weight of 0.5, and n-gram (news, cluster, consists, of) will have a weight of 1.0, which is the maximum weight of the full n-gram template. SC feature value has a real value in the interval [0,1]. Weight of the feature is calculated regarding the weight of the pair with the maximum value of the strict context, proportional to the weight of the strict context for the current pair.

c) Scalar Product Similarity of the context for use on the internal features of the sentence (SPS).

SPS feature value has a real value in the range from 0 to 0.5 (half weight of the feature), and is calculated as the cosine similarity rate for all the contextual features (AcrossVerb, Near and NotNear), bounded above by the 0.5 value (Ferreira Rafael et al., 2014).

Context-independent features:

a) Beginning Similarity, BS. Consideration of BS for expressions is a natural way for detecting semantically related objects. Currently, a simple metric of similarity is applied - the same word beginning. This feature allows finding the similarities between such expressions as Administrator - Administration, the President of Russia - the Russian President, etc.

The total weight of BS feature has real value from the interval of [0.1] and is calculated using the following formula:

$$BS = 1.0 - 0.1 \cdot N_{diff} \quad (12)$$

Where N_{diff} is the number of words with different beginnings.

b) Thesaurus similarity described in the external source (TS).

Currently, there is a wide variety of predefined resources containing additional information about the connections of words and phrases. This information can be used to build topic-based units and make this structure more stable and quality. TS feature calculation is based on the use of information from the Russian language thesaurus RuThes. It covers the following types of connections: synonymy, part - whole genus - species. TS feature value has a real value between 0 and 1 and is calculated inversely proportional to the distance between objects in the thesaurus:

$$TS = 1.0 - 0.2 \cdot N_{rel} \quad (13)$$

Where N_{rel} is the path length in the thesaurus relations (the number of connections).

c) Availability of the similar linguistic expressions (Embedded Objects Similarity, EOS).

The total similarity weight of the contemplated objects pair is calculated as the sum of the weights on individual features of the similarities described above. Thus, each pair receives a weight within the range from 0 (no similarity) to 5 (maximum similarity) obtained based on the six features (three context-dependent and three context-independent) within the range from 0 to 1 (SC, BS, TS, EOS) and from 0 to 0.5 (NSF, SPS) (Alekseev A. A., 2013).

4) Algorithm for constructing a topic-based representation based on the combination of factors

Algorithm for constructing a topic-based representation provides topic-based units of expressions pairs in order of their similarity. The proposed structure of the topic-based unit has the following properties:

- text expression can belong to one or two topic-based units; multiple membership permission provides the ability to represent different aspects of the original text expression, as well as its lexical ambiguity;
- Each topic-based unit has a main element - the centre of the topic-based unit, which can only belong to one cluster topic-based unit; topic-based units centre is the most frequent element among all elements of the topic-based unit (Mashechkin et al., 2011).

Building the topic-based presentation consists of the following steps:

- a pair of text expressions with the largest weight of similarity among all pairs of candidates is considered;
- The most frequent pair absorbs less frequent element with all its text entries and contextual features, and becomes the representative of a given pair of text expressions - the centre of a new topic-based unit;
- the less frequency element of this pair may further join another topic-based unit in a similar way;
- Uniting topic-based units, consisting of several elements is similar to uniting single text expressions; centre of the most frequent topic-based unit becomes the centre of a new, united topic-based unit.

In general, each iteration of the algorithm consists of three main steps:

1. ranking pairs of candidates;
2. choice for the pair for uniting (the largest weight + constraint satisfaction);
3. Uniting procedure (Abramov et al., 2011).

The iterative process continues until there are pairs of candidates for uniting with the similarity weight above the predetermined threshold.

3. Results

Having researched these methods and algorithms, optimal solutions for solving the problem of the semantic analysis for the news texts were selected and implemented.

The problem of establishing semantic similarity for the documents of the news cluster utilises the shingles algorithm considering the following features of the news documents (Kiselev et al., 2005):

- constantly growing collection of documents;
- the same article may cover several stories;
- Different parts of the document shall have a different weight at identifying similarity;
- stories and documents can be cross-referenced to each other.

3.1 Shingles Algorithm Parallel Implementation for Multiprocessor Heterogeneous Computer Systems using CUDA and Open CL

Brief description of the technology:

CUDA: when the application is executed, the CPU performs its portion of the code, and the GPU executes CUDA-code with the heaviest parallel computing. The part designed for the GPU is called the kernel, defining the operations executed on the data.

Open CL: all the threads on the graphics card are divided into Work Groups. The size of groups may differ at various time intervals. This is due to Hyper Threading and power saving features. Their size varies on the devices by different manufacturers, but, for example, executing the application on a video card by NVidia, one can be sure that the maximum size of a single working group will have 48 threads.

The shingles algorithm features by the fact that the main computational load belongs to the calculation cache (CRC32). For this reason, the accelerators cores shall receive the input text converted into shingles, calculate the cache for each pair and compare, returning the result.

We make two versions of the code using Cuda and OpenCL approaches. Then we compose a summary table 1 with the assessment of the time (in seconds) for each implementation. The number of iterations can be regarded as the number of the analysed news texts.

Table 1. Summary table of the time for different approaches

| Iterations | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
|-------------|------|------|------|------|------|------|-------|-------|-------|-------|
| Serial code | 0.39 | 1.14 | 2.23 | 3.69 | 5.54 | 7.74 | 10.29 | 13.21 | 16.50 | 20.15 |
| OpenMP | 0.34 | 0.71 | 1.44 | 2.21 | 3.32 | 5.47 | 7.69% | 9.86 | 11.93 | 14.26 |
| CUDA | 0.13 | 0.23 | 0.51 | 0.73 | 1.10 | 1.42 | 1.89 | 2.21 | 2.82 | 3.42 |
| OpenCL | 0.11 | 0.17 | 0.27 | 0.34 | 0.52 | 0.62 | 0.87 | 1.04 | 1.35 | 1.69 |

Graphs of the time dependency on the number of texts for different technologies:

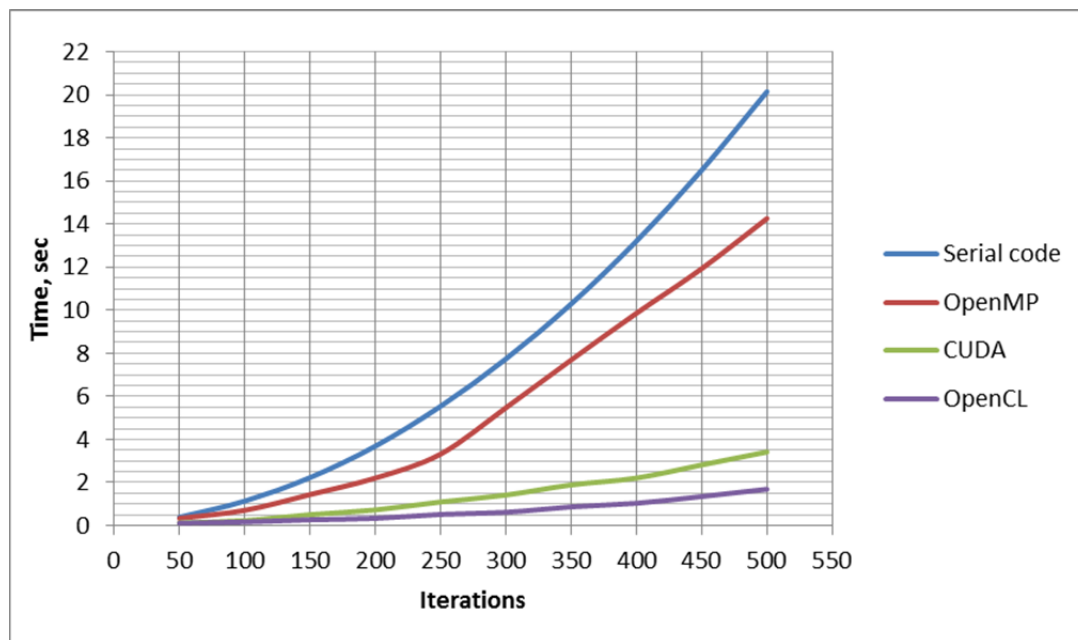


Figure 7. Dependence of the processing time on the number of iterations

Testing was conducted on the following devices: CPU: Intel core i5 3.0 GHz; Cuda: NVIDIA GeForce GTX 650Ti 2GB; OpenCL: AMD Radeon 7870 2GB.

3.2 Shingle Algorithm Parallel Implementation for Distributed Computational Systems (using Google App Engine-GAE)

GAE is a hosting service for websites and web-based applications on Google's servers with free name <Site_Name>.appspot.com, either with own name, involved using the Google services. Applications based on App Engine must be written in Python, Java, or PHP.

We make the application using GAE. Then test it on different quantity of the compared texts with size up to 50 KB (46 KB). First, start the application using the GAE, then start it locally (computing node - one core of Core i5 3.3 GHz).

Graph for using computing units is shown in Figure 8:

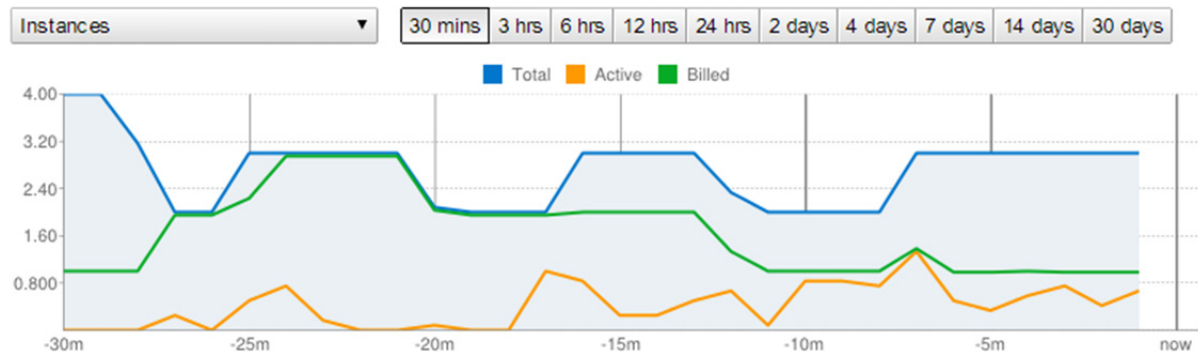


Figure 8. Graph for using computing units

Then measure the time and calculate the acceleration. Data will be given in Table 2.

Table 2. Time and acceleration using App Engine

| Iterations | 5 | 10 | 15 | 20 | 25 | 30 |
|----------------------|--------|--------|--------|--------|--------|--------|
| Time with App Engine | 9.558 | 18.454 | 27.792 | 36.450 | 45.416 | 54.436 |
| Local time | 10.212 | 20.414 | 30.530 | 40.518 | 50.888 | 63.166 |
| Acceleration | 1.068 | 1.106 | 1.099 | 1.112 | 1.120 | 1.160 |

Execution time:

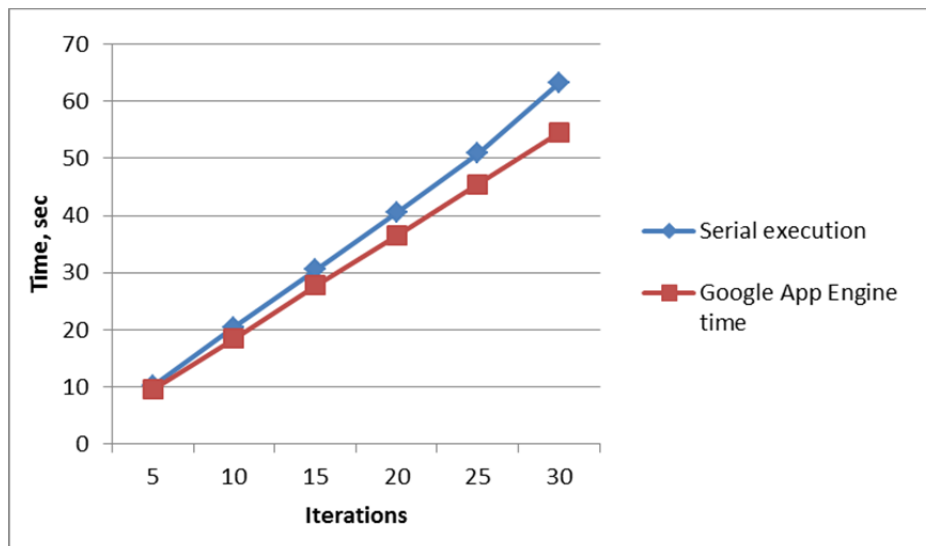


Figure 9. Text processing time

Acceleration

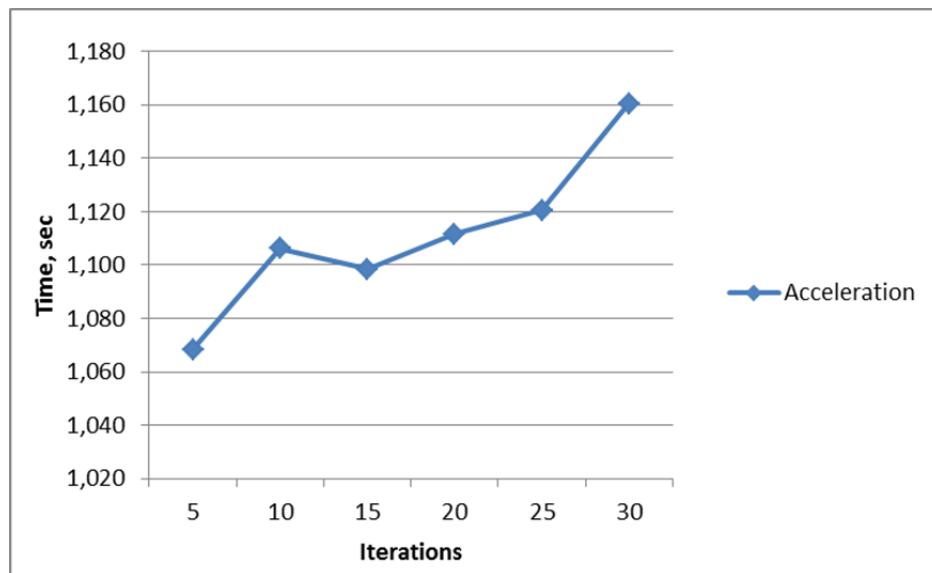


Figure 10. Acceleration using App Engine

3.3 Extraction of Key Phrases from the Text and Representation in the Form of Mind Maps of the News

At this stage, before choosing a key phrases extraction method, one needs to identify common features of news texts structure.

After analysing a number of articles presented at the well-known news sites, such as sites of newspapers Lenta.ru, NewsRu.com, Kommersant.ru, Expert (and others related to the top-30 news portals in RuNet), it is possible to make a generalized structure of the news text (Figure 11).

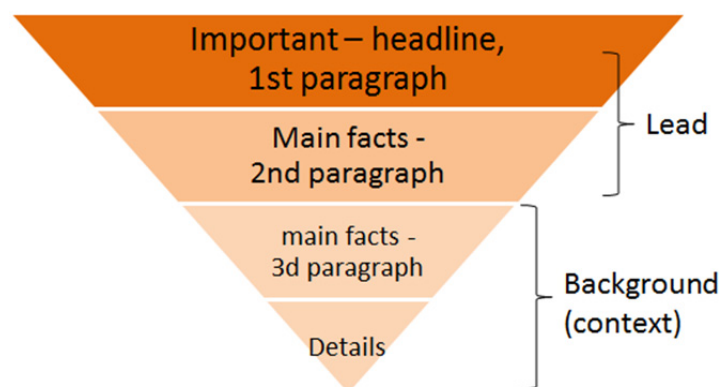


Figure 11. News text structure

It is based on the principle of "inverted pyramid": the basic information is placed at the beginning of the material and its subsequent disclosure is provided further in detail.

- Title of the news reflects its subject and contains no more than 10 words (about 80 characters). Thus, for example, Yandex displays no more than 15 words in the title, Google - up to 70 words.
- Key facts about the event are reflected in paragraphs 1-2, and constitute the so-called text lead (highlights the main topic).
- Third and subsequent paragraphs constitute background of the news (context). As a rule, this part discloses the details of what is going on, and provides the information directly relating to the news.

Thus, for the content of the news has the following formula: (Who? + What? + Where? + Why? + When? + How?). This the so-called "five W and one H" principle, attributed to R. Kipling. If all the news reports were

based on a single structure, the solution of the clustering problem could become significantly easier.

Considering the above presented algorithms, graph methods are optimal due to the peculiarities of construction and convenience in the visual representation of the news. However, the informational structure of the news is usually based on the use of named entities (persons, organizations, locations, etc.).

Therefore, graph method was developed together with own key word search algorithm combining the selection of named entities from the news text (based on morphological analysis and the plugging-in PullEnti SDK module), counting word's weight considering the frequency of its occurrence and location in the text - in the title, lead or context (Figure 12).

Here, candidates for key words are unique words (person, organization, location and other identified by PullEnti module); words that could not be determined using morphological dictionary; words-entities in the Nominative case.

Words algorithm for calculating the threshold for referring the entity as the key one is as follows:

- words-entities are ordered by descending relative frequency (RF);
- then word index is calculated;
- the threshold value of the relative frequency - RF of a word with index equal to $(0.2 \times \text{the number of entities})$.

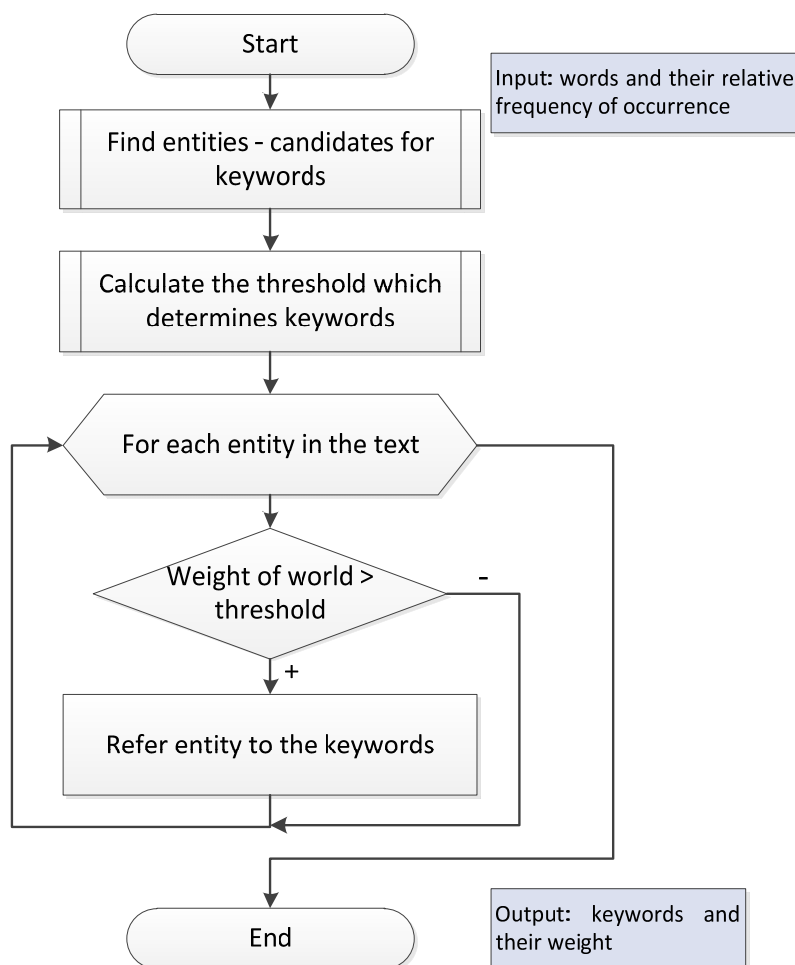


Figure 12. Key words extraction algorithm

Based on the key entities search, it is possible to calculate the weight of sentences by the following formula:

$$W_s = N_{kw} \cdot Rf_{kw} \cdot ParagraphWeight \cdot k \quad (14)$$

where W_s is the weight of the sentence, $N_{kw} \cdot Rf_{kw}$ is the weight of the key entity, N_{kw} is the number of key

word occurrences in the sentence;

Rf_{kw} - relative frequency of the key word;

ParagraphWeight - relative weight of the paragraph in the text, equals to 0.35 for the first paragraph (lead), 0.2

- for the second paragraph, and 0.1 - for others (context); title weight equals to 3;

k - sentence rating within a paragraph, for the first sentence in paragraph, it equals to 1, for the rest - 0.8.

Figure 13 is a screenshot of the main window of the developed system performing a comprehensive analysis of the news text. The article (RIA Novosti, electronic source) was selected from a news web site. Its abstract is made based on the key entities and sentences determining algorithm.

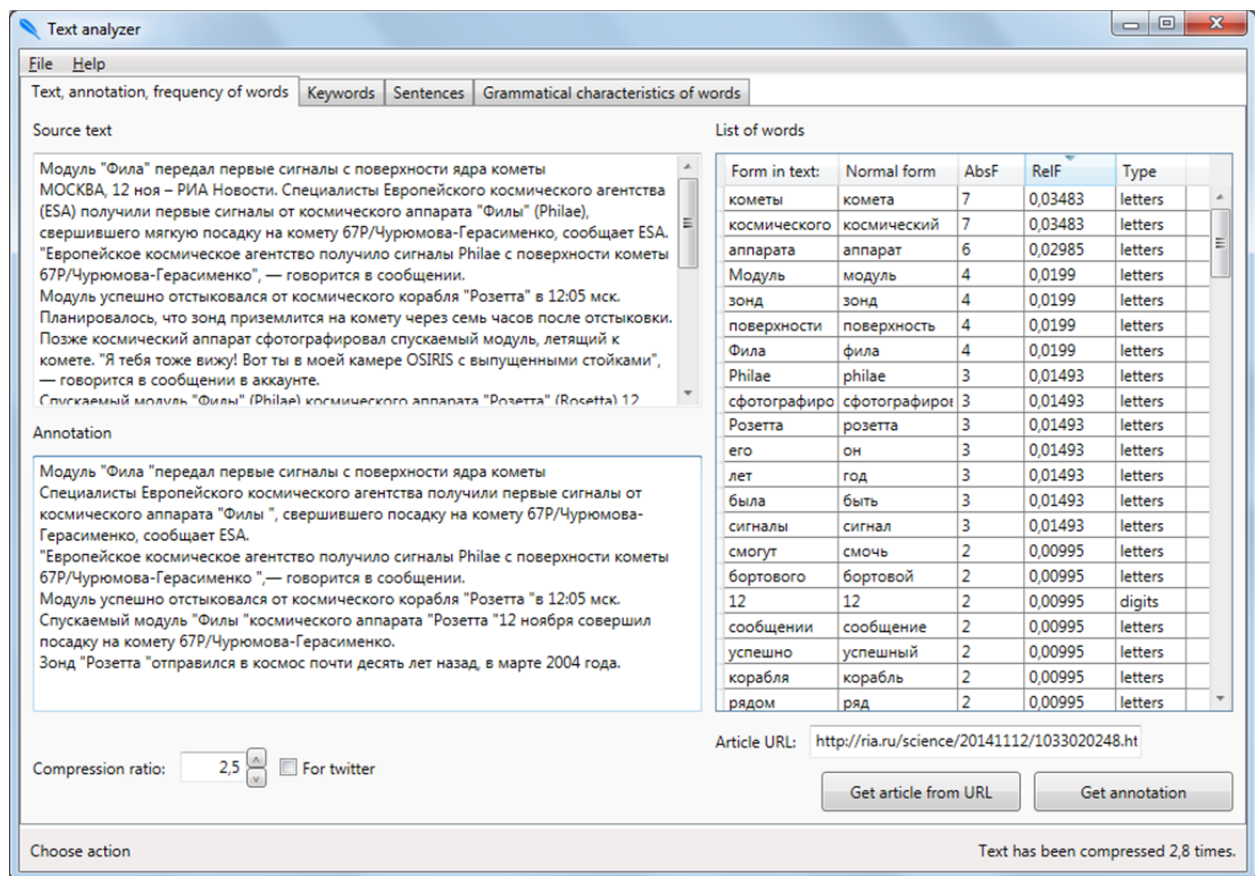
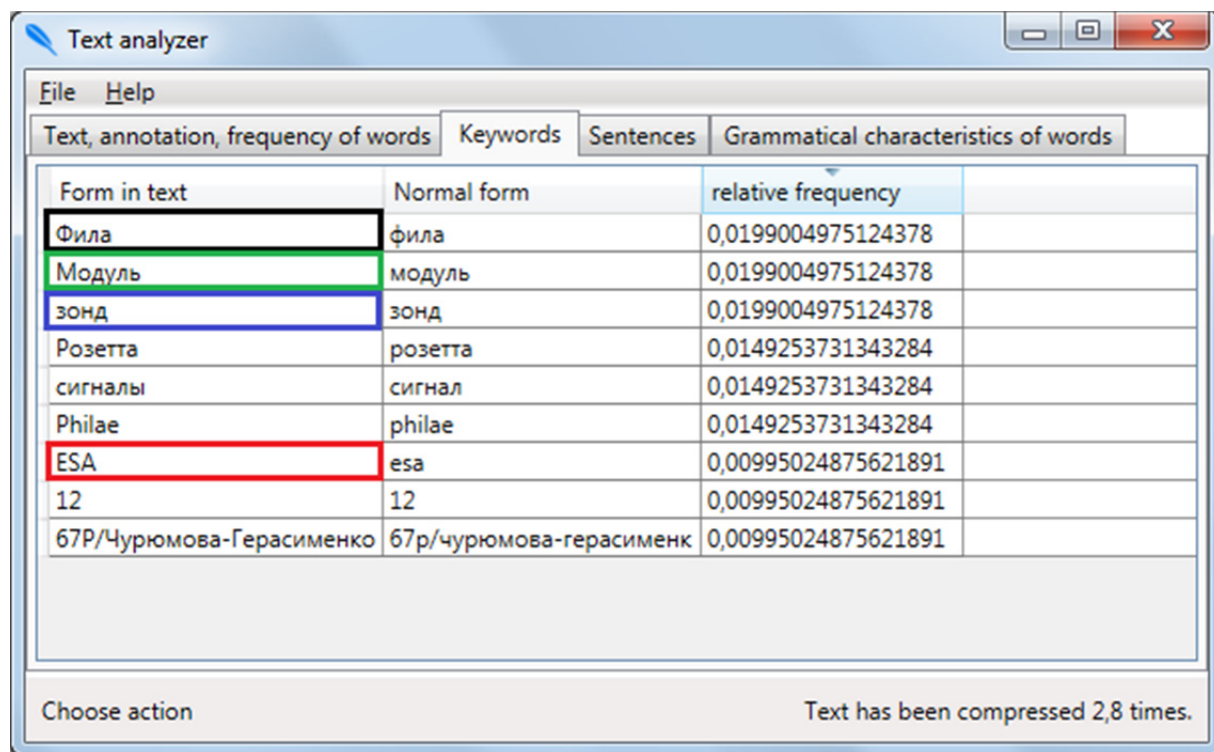


Figure 13. Abstract text creation

Key entities in Russian (topic-based units of the news) are figured out and highlighted in Figure 14.



The screenshot shows a window titled 'Text analyzer' with a menu bar (File, Help) and four tabs: 'Text, annotation, frequency of words', 'Keywords', 'Sentences', and 'Grammatical characteristics of words'. The 'Keywords' tab is active, displaying a table with three columns: 'Form in text', 'Normal form', and 'relative frequency'. The table lists several words, with 'Фила', 'Модуль', and 'зонд' highlighted in black, green, and blue respectively. 'ESA' is highlighted in red. At the bottom, there is a 'Choose action' button and a status bar indicating 'Text has been compressed 2,8 times.'

| Form in text | Normal form | relative frequency |
|--------------------------|-------------------------|---------------------|
| Фила | фила | 0,0199004975124378 |
| Модуль | модуль | 0,0199004975124378 |
| зонд | зонд | 0,0199004975124378 |
| Розетта | розетта | 0,0149253731343284 |
| сигналы | сигнал | 0,0149253731343284 |
| Philae | philae | 0,0149253731343284 |
| ESA | esa | 0,00995024875621891 |
| 12 | 12 | 0,00995024875621891 |
| 67P/Чурюмова-Герасименко | 67p/чурюмова-герасименк | 0,00995024875621891 |

Figure 14. Key words of researched text

Next, key words hierarchy is formed, and mind map in Russian of the original article is formed based on this hierarchy (WikIT, electronic source), (Dexi Liu et al., 2013):

Посадка модуля "Филы" (Landing of the "Fila" module)

Зонд Розетта (Rosetta probe)

Исследовательский (Research)

в космосе в 2004 года (in space in 2004)

Модуль (Module)

посадочный (landing)

отделился от "Розетты" (separated from the "Rosetta")

12 ноября в 12:05 по мск (November 12 at 12:05 pm Moscow time)

через 7 часов сел на поверхность кометы (In 7 hours landed the surface of the comet)

Специалисты ESA (ESA experts)

фотография модуля (Pictures of the module)

изучение состава небесного тела (Research of the celestial body composition)

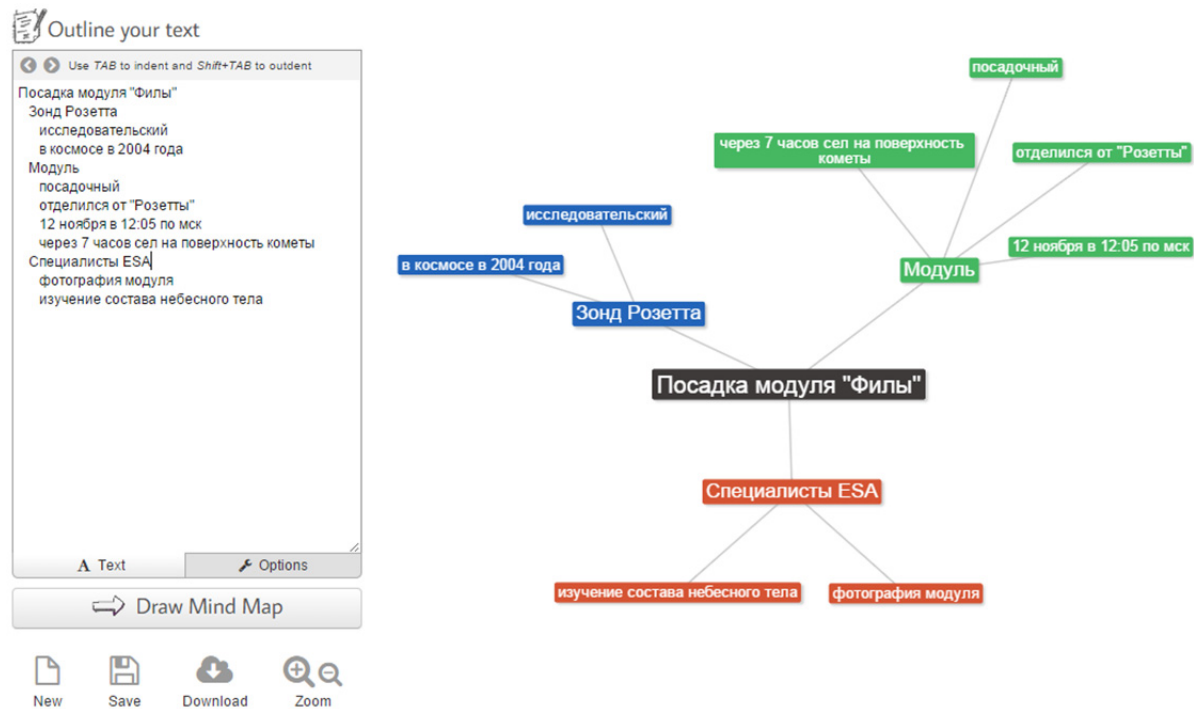


Figure 15. Representation of the text as a mind map

In addition, there is a possibility of aggregating multiple mind maps for articles with similar meaning in one mind map, which is convenient for perceiving information (Figure 16).

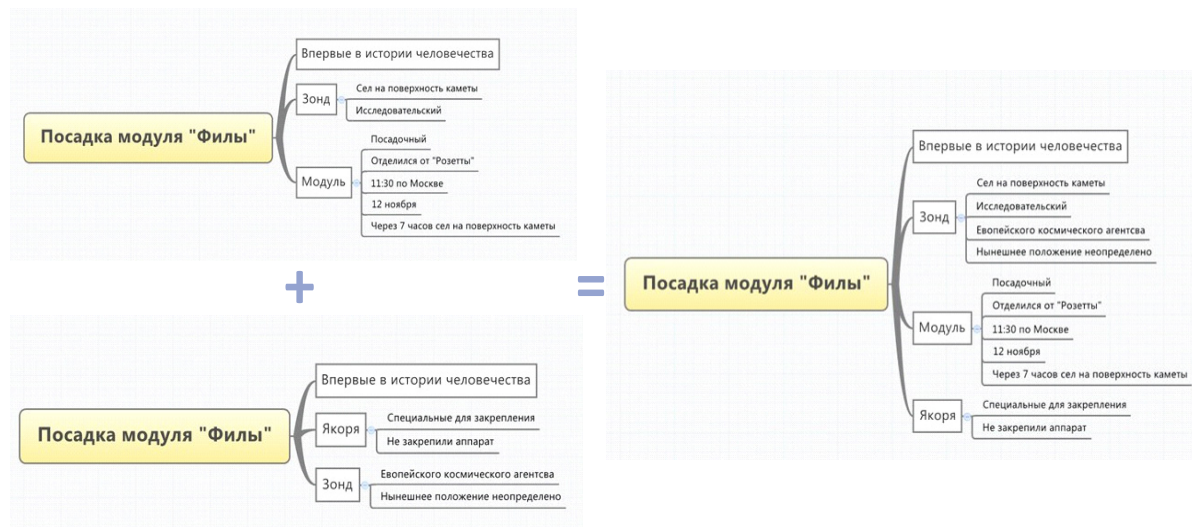


Figure 16. Generalized mind map news

4. Discussion

4.1 Evaluation of the Shingles Algorithm Implementation Results

Let us analyse the obtained results.

1) Shingles algorithm parallel implementation for multiprocessor heterogeneous computer systems using CUDA and Open CL.

The sequential code and Cuda - the average acceleration is 5.10, while hashes counting and comparing were made in parallel, and text normalization was performed sequentially.

The sequential code and OpenCL - the average acceleration is 10.12, and, similarly to the Cuda option, text normalization is not performed in parallel.

Cuda and OpenCL - the average acceleration of 1.93 (OpenCL is faster). In this case, it can be explained by the fact that the tested OpenCL graphics card is slightly better in computing power than its competitor.

2) Shingle algorithm parallel implementation for distributed computational systems (using Google App Engine)

Google App Engine also allows for acceleration of 1.5 times compared with the theoretical. It is necessary to take into account that we were allocated the number of computational nodes of not more than 4. When allocating more computational power to work with large texts, it is possible to get significant acceleration compared to the local machine.

Thus, we can conclude that, depending on the type of task being solved and used hardware technologies, it is possible to use various shingle algorithm parallel implementations for determining similarity of the news texts in the stream. In our case, OpenCL sufficiently exceeds the other options.

4.2 Evaluation of the Results of the Key Phrases Search Modules

The most important step in the task for extracting key phrases is to calculate the information content of their weights, to assess their significance in relation to each other in the document. The use of automated methods for extracting key phrases improved the efficiency of processing online news articles.

The following results were obtained for the processing time and the quality of the selected key phrases (Figures 17 and 18):

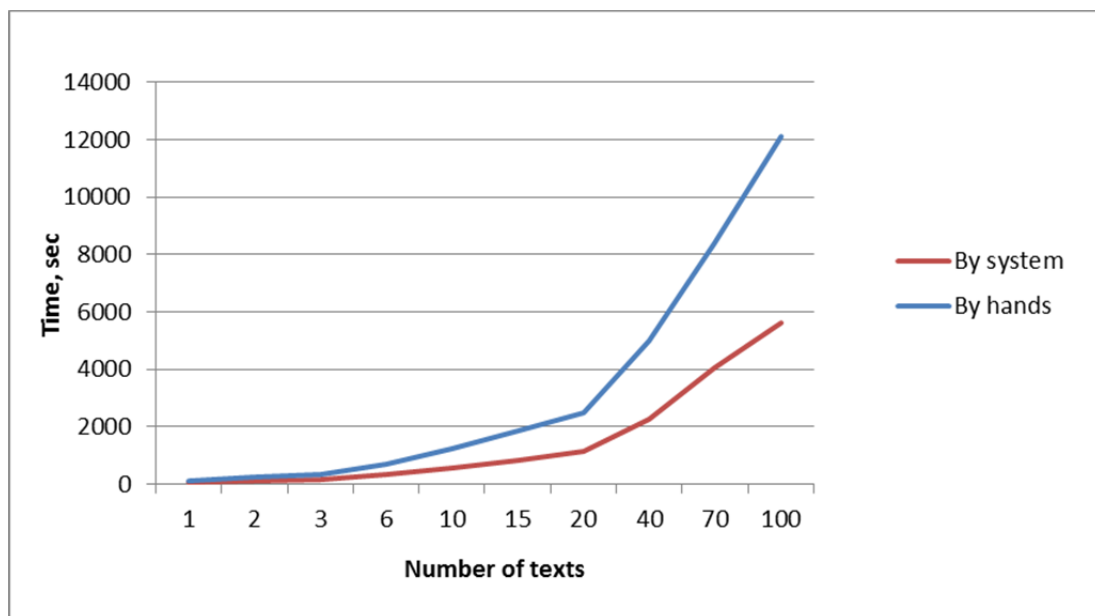


Figure 17. Dependence of the processing time on the number of texts

In comparison with the manual method, the time for determining key entities automatically decreased at least 2-fold. Thus, it shall be noted that it takes into account not only the analysis by the system itself, but also the time required for final correction of texts.

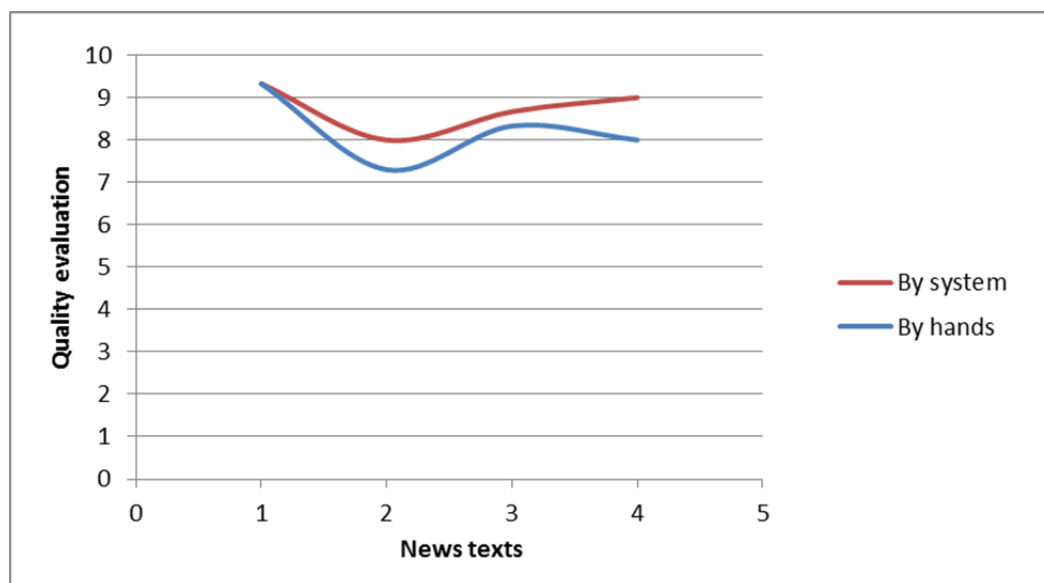


Figure 18. Quality of the obtained key phrases

Quality of the results can be assessed on the following criteria: preservation of key facts, key entities connectivity, maintaining the syntactic structure of the text after removing insignificant parts. Each of these criteria was assessed by experts on a scale from 0 to 10 points, and then in order to assess the quality (adequacy) of the extracted key entity of the abstract, the arithmetic mean of the three indicators for each text was calculated.

Thus, the quality of the news is treated at the same level as in the analysis of the text by a person and the processing time is reduced at least several times.

It is also necessary to note some disadvantages. The research of the described algorithm for extracting key terms demonstrated accuracy dispersion for the texts on various subjects and identified the problems at the stages of part-of-speech tagging. The optimum shingle size on the stage of defining semantic similarity and the threshold for referring the entity as the key were determined experimentally. Therefore, repeated statistical test on a large collection of documents will give a chance to get closer to the results demonstrated in the paper. In addition, deepening of syntactic news text analysis is planned for the construction of a detailed mind map of the text.

Acknowledgments

This paper received partial support from the Russian Foundation for Basic Research (projects No. 13-07-00459, 13-07-97042, 13-07-00351, 14-07-97016, 14-07-97017).

References

- Abramov, V. E., Abramova, N. N., Nekrasova, E. V., & Ross, G. N. (2011). *Statistical analysis of coherence for the texts on social and political topics*. Proceedings of the XIII All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections" - RCDL'2011, Voronezh.
- Alekseev, A. A., & Lukashevich, N. V. (2011). Automatic extraction of entities based on the structure of the news the cluster. *Artificial intelligence and decision-making*, 51-59.
- Alekseev, A. A. (2013). *Topic-based presentation of the news cluster as a basis for automatic abstracting*. Proceedings of the XV All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections" - RCDL'2013, Yaroslavl, 173-182.
- Anisimov, A. V., Marchenko, O. O., & Kysenko, V. K. (2011). A method for the computation of the semantic similarity and relatedness between natural language words. *Cybernetics and Systems Analysis*, 47(4), 515-522. <http://dx.doi.org/10.1007/s10559-011-9334-2>
- Dexi, L., Shihan, W., Yuehua, L., Guoqiang, D., Jiezhao, P., Naixue, X., & Athanasios, V. V. (2013). *A query-oriented XML text summarization for mobile devices*. *Soft Computing*, 17(9), 1585-1593. <http://dx.doi.org/10.1007/s00500-012-0980-8>

- Dongmei, A., Yuchao, Z., & Dezheng, Z. (2010). Automatic text summarization based on latent semantic indexing. *Artificial Life and Robotics*, 15(1), 25-29. <http://dx.doi.org/10.1007/s10015-010-0759-x>
- Ferreira, R., Luciano de Souza, C., & Frederico, F. (2014). A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, 41(13), 5780–5787. <http://dx.doi.org/10.1016/j.eswa.2014.03.023>
- Furu, W., Wenjie, L., Qin, L., & Yanxiang, H. (2010). A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems*, 22(2), 245-259. <http://dx.doi.org/10.1007/s10115-009-0194-2>
- Huang, X. J., Xiaojun, W., & Jianguo, X. (2014). Comparative news summarization using concept-based optimization. *Knowledge and Information Systems*, 38(3), 691-716. <http://dx.doi.org/10.1007/s10115-012-0604-8>
- Kiselev, M. V., Pivovarov, V. S., & Shmulevich, M. M. (2005). *Text clustering method, taking into account the co-occurrence of the key terms, and its application to the thematic structure analysis for the news flow, as well as its dynamics*. Internet-mathematics 2005. Automatic processing of web data. M., p. 412-435.
- Lande, D. V., Snarskiy, A. A., & Yagunova, E. V. (2013). *Using horizontal visibility graphs to identify words defining the information structure of the text*. Proceedings of the XV All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL'2013, Yaroslavl.
- Leonova, Yu. V., & Fedotov, A. M. (2013). *Extraction of knowledge and facts from the texts of dissertations and autoabstracts to study the relations of scientific communities*. Proceedings of the XV All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL'2013, Yaroslavl.
- Martinez-Gil, J. (2012). *An overview of textual semantic similarity measures based on web intelligence*. *Artificial Intelligence Review*. <http://dx.doi.org/10.1007/s10462-012-9349-8>
- Mashechkin, I. V., Petrovskiy, M. I., Popov, D. S., & Tsarev, D. V. (2011). Automatic text summarization using latent semantic analysis. *Programming and Computer Software*, 37(6), 299-305. <http://dx.doi.org/10.1134/S0361768811060041>
- Module "Fila" gave the first signals from the surface of the comet's nucleus*. RIA Novosti. Retrieved from <http://ria.ru/science/20141112/1033020248.html>
- Shingle algorithm for web documents, search for fuzzy duplicate texts, comparing texts similarity*. Retrieved from <http://www.codeisart.ru/part-1-shingles-algorithm-for-web-documents/>
- Soloshenko, A. N., Rozaliev, V. L., & Orlova, Yu. A. (2014a). *Automation of semantic analysis for online news texts*. Open semantic technologies for designing intelligent systems OSTIS-2014: Material of IV Intern. scientific and engineering Conference (Minsk, Feb 20-22, 2014). Belarusian State University of Informatics and Radio Electronics, High-Tech Park Administration. Minsk, p. 435-438.
- Soloshenko, A. N., Rozaliev, V. L., Orlova, Yu. A., & Zaboleeva-Zotova, A. V. (2014b). *Topic-based Clustering Methods Applied to News Texts Analysis* Knowledge-Based Software Engineering: Proceedings of 11th Joint Conference, JCKBSE 2014 (Volgograd, Russia, September 17-20, 2014). In A. Kravets, M. Shcherbakov, M. Kultsova, & Tadashi Iijima (Eds.), Volgograd State Technical University [etc.]. Springer International Publishing, 2014. – P. 294-310. – (Series: Communications in Computer and Information Science ; Vol. 466). http://dx.doi.org/10.1007/978-3-319-11854-3_25
- Sheng-Tun, L., & Fu-Ching, T. (2010). Constructing tree-based knowledge structures from text corpus. *Applied Intelligence*, 33(2), 67-78. <http://dx.doi.org/10.1007/s10489-010-0243-2>
- Soledad, P. M., & Yiu-Kai, D. N. (2012). Using maximal spanning trees and word similarity to generate hierarchical clusters of non-redundant RSS news articles. *Journal of Intelligent Information Systems*, 39(2), 513-534. <http://dx.doi.org/10.1007/s10844-012-0201-z>
- Text to mind map – WikIT*. Retrieved from http://www.informationtamers.com/WikIT/index.php?title=Text_to_mind_map
- Ustalov, D. A. (2012). *Extraction of terms from the Russian texts using the graph models*. Graphs theory and applications: Proceedings of the conference, 62-69.
- Vasiliev, V. G. (2011). *Classification and separation of fragments in the texts on the basis of logical rules*.

Proceedings of the XIII All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL'2011, Voronezh.

Zaboleeva-Zotova, A. V., & Orlova, Yu. A. (2010). *Automation of semantic text analysis on technical specifications: a monograph*. Volgograd: IUNL, 155.

Zakharov, V. N., & Khoroshilov, A. A. (2012). *Automatic assessment of the topic-based content similarity of the text based on a comparison of their formal semantic descriptions*. Proceedings of the XIV All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL'2012, Pereslavl-Zalesskiy.

Zelenkov, Yu. G., & Segalovich, I. V. (2007). *Comparative analysis of methods for fuzzy duplicate detection for Web-documents*. Proceedings of the IX All-Russian Scientific Conference named "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL'2007, 9.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).