A modified algorithm of the latent semantic analysis for text processing in the Russian language

A A Ivanov and S V Holtzer

1 Institute of Computational Mathematics and Mathematical Geophysics SB RAS. Academician Lavrentiev Avenue, 6, 630090, Novosibirsk, Russia

E-mail: 6ppp@mail.ru

Abstract. The paper presents a methodology for analyzing texts in the Russian language. The methodology is based on the Latent Semantic Analysis (LSA) algorithm. A number of disadvantages of the classical method are considered, and modification methods of extracting N-grams from the text are proposed. The modified method allows one to reduce a number of extracted N-grams and an increasing the meaningfulness of the retrieved collection in comparison with a standard method. The reduction of the collection size leads to a reduced dimension of the TF-IDF matrix and accelerated the execution of the SVD method. The advantages of the developed machine learning algorithm are demonstrated on simple sentences. Owing to discussed ideas it becomes possible to effectively parallelize the text processing at the lemmatization step.

1. Introduction
The rate of data accumulation occurs with exponential growth. The question arises about how to convert them into information in order to synthesize knowledge. Automation of this process will lead to a situation in which the rate of the emergence of a new knowledge will overtake the rate of their development and application. Therefore, the methodological development of the new algorithms for obtaining knowledge from data using automatic tools for finding keywords and machine learning is of the greatest importance that could practically help with finding semantics.

Nowadays, a text is the most common means of transmitting information. Therefore, the use of machine learning algorithms to highlight the meaning of a text is one of the most important tasks, whose solution is in demand in various fields related to human analytical activity.

The LSA was developed in the 80s and was intended to analyze the relationship between documents and terms in the collection, to extract topics in the collection [1, 2]. The active development of computer technology made it possible to use the LSA as the main method for a rapid analysis of huge data arrays of texts (big data). The classic LSA has a number of disadvantages that affect the accuracy of the results to be obtained. It becomes relevant to modify an algorithm in order to increase the accuracy of the method, as well as to develop parallelization algorithms that would allow the method to be applied on a supercomputer.

In this paper, a standard LSA method is described. The disadvantages and limitations of the LSA are indicated. A number of modifications of the method using linguistic algorithms that will clarify the result of using the method were proposed. A mechanism for parallelizing data in order to increase the amount of processed information and reduce the processing time of text is proposed. Examples of the operation of machine learning algorithms based on linguistic are given.
2. The classic LSA algorithm and its limitations

Let n be the number of documents requiring processing, m is the number of different terms collected from a collection of documents by the N-gram method. The LSA method inherits the problems that are encountered when collecting N-grams [3]:

- frequent use of words - “the problem of littering” in periodically repeated phrases in the text,
- defining the boundaries of a topic - the same words in different combinations and contexts can describe different topics,
- homonyms - words in the same form with different meanings,
- synonyms - words in different forms with similar meanings,
- searching semantic connection - words can have different semantic meanings, but at the same time be somehow connected.

Next, the “term-document” matrix $X$ is built, where $m \times n$ are dimensions. The cells $x_{i,j}$ contain the weighting coefficients of each term $t_i$ in the document $d_j$ calculated according to the formula BM25 [3]:

$$
score(D, T) = \sum_{i=1}^{n} idf(t_i, D) \frac{f(t_i, D)(k + 1)}{f(t_i, D) + k (1 - b + b \frac{|T|}{avgdl})}
$$

$score(D, T)$ is the ranking function used by search engines to organize documents according to their relevance to a given search query, $T$ is a bag of keywords consisting of the terms $t_1, t_2, ..., t_m$ of the document $D$. $|T|$ is the total number of words in $D$, $avgdl$ is the average length of the document in the sample. $k, b$ - free odds, most often taken as $k = 2, b = 0.75$. To calculate IDF, a smoothed formula is used: $idf(t_i, D) = log \frac{|D| - n(t_i, D) + 0.5}{n(t_i, D) + 0.5}$, where 0.5 is a coefficient introduced into the equation according to the Robertson probabilistic model to improve the estimate, $tf(t_i, d) = \frac{f(t_i, D)(k + 1)}{f(t_i, D) + k (1 - b + b \frac{|T|}{avgdl})}$.

Any document within itself can contain a number of topics. To determine and rank topics according to their significance, to identify important terms and documents that describe topics, the singular value decomposition (SVD) method is applied to the TF-IDF matrix:

$$
X = U \Sigma V^T
$$

Here $U$ and $V$ are the orthogonal matrices with $m \times r$ dimensions (matrix – «term-topic») and $n \times r$ (matrix – «subject-document»), respectively, $\Sigma$ is the diagonal matrix of $r \times r$ dimension, containing eigenvalues sorted in descending order. Each of $r$ eigenvalues corresponds to one topic outlined in the collection of documents, and its value indicates to their importance.

To filter out data (noise) that are not significant with respect to the entire collection, the parameter $k$ is introduced ($k \leq r$), which determines the dimension of the truncated matrices. For the convenient data analysis, a reconstructed matrix $X'$ is built, which is similar to the original one (its rows and columns are views for the same terms and documents), obtained by multiplying the truncated matrices. Upon multiplication, the weights are adjusted as such that the «noise» is removed from the data, and the relationships between the terms are amplified [4].

3. The modified N-gram method based on the linguistic rules of the Russian language

There are no strict rules for writing sentences in the Russian language, so related words can be far apart in a text. It is proposed to use linguistic rules that take into account the combination parts of speech among themselves regardless of their location in a sentence, punctuation, construction of sentences, dictionaries, ontology, stop words to correctly determine semantic relationships and phrases, improve the accuracy of the classical N-gram method [5].

For parts of a speech, there are a number of semantic properties. A noun is a part of a sentence denoting an object. Two or more adjacent nouns often form a strong semantic connection, are candidates for the key phrase. A verb is a part of a sentence denoting an action or a condition. An adjective denotes an attribute of an object / subject. An adverb denotes a sign of action. The
preposition determines the relationship between an object and a subject, and establishes the relationship of the object to the other words in a sentence. The union in the sentence carries out the transfer of the meaning between the parts of a compound sentence, and can express logical relationships and the causation connecting homogeneous members of the sentence. Interjections in the text convey emotions and most often are stop words for a sequence of N-grams. Pronouns always indicate an item, but do not name it. The problem arises of pronoun resolution. It is proposed to use the following rule for resolving pronoun references: if a personal pronoun is consistent with the last noun encountered in number, gender, and case means, this pronoun refers to this noun, with a high degree of probability.

In linguistics, there is a concept of a predicate (some attribute / property of an object). There are different classifications for them [6]. The developed machine-learning algorithm makes it possible to extract an object / a subject, their predicate composition, relationships with other objects / subjects from a sentence, and map the sentence to a semantic network or a graph in which the vertices correspond to the keywords, and the edges correspond to the predicate relations between them. In addition, to filtering by a predicate type, one can use filtering by questions that are restored using the Rosenthal dictionary [7]. Such a classification allows one to know, for example, what the text is about, highlight who does, what does and how does, etc. The key advantages of the semantic network are a compact data presentation, convenient and quick search, thought this data, construction of the shortest routes from one key term to another and the construction of causal relationships.

Figure 1 presents a table of machine compatibility of parts of the text with each other according to morphological characteristics for constructing N-grams, which are built from nouns or verbs.

| part of speech | noun | noun with prepos. | adject. | noun-participl. | numeral | pronoun | verb | verb-participl. | adverb |
|---------------|------|-------------------|--------|----------------|--------|---------|------|----------------|--------|
| noun          | ✓    | ✓                 | ✓      | ✓              | ✓      | ✓       | ✓    | ✓              | x      |
| noun with prepos. | ✓    | ✓                 | ✓      | ✓              | ✓      | ✓       | ✓    | ✓              | x      |
| adject.       | ✓    | ✓                 | ✓      | ✓              | x      | ✓       | x    | x              | x      |
| noun-participl. | ✓    | ✓                 | x      | x              | x      | ✓       | x    | x              | x      |
| numeral       | ✓    | ✓                 | x      | x              | x      | ✓       | x    | x              | x      |
| pronoun       | ✓    | ✓                 | ✓      | x              | x      | ✓       | x    | x              | x      |
| verb          | ✓    | ✓                 | x      | x              | x      | ✓       | ✓    | ✓              | ✓      |
| verb-participl. | x    | x                 | x      | x              | x      | ✓       | ✓    | ✓              | x      |
| adverb        | x    | x                 | x      | x              | x      | ✓       | x    | x              | x      |

Fig 1. Table of compatibility of parts of speech by morphological characteristics.

The machine compatibility of morphological properties differs from the compatibility in a natural language by the fundamental impossibility of restoring implicit predicative properties of the consciousness of speakers. If there are relatively few restrictions on the compatibility of parts of a speech for a natural language, then for a machine analyzer we are forced to abandon all combinations that imply hidden possibilities of the syntactic-semantic properties of parts of a speech. It is necessary to create more cut offs that are stringent because a machine cannot understand the contextual
semantics of a statement. If incompatible words are encountered during the construction process, a clipping occurs.

When processing the structured documents, for example, contracts, typed certificates, e-mail letters etc., it is possible to extract text metadata according to the model of an original document. In such a case, extracting metadata is the first step in the text processing.

For the effective calculation, it is important to assume that a paragraph is a self-sufficient carrier of meaning. Then, it is important to preserve a sequence and nesting of paragraphs in order to preserve the hierarchy of the meaning when constructing semantic networks. It is convenient and easy to parallelize a text, one paragraph for each computational processor for lemmatization. Lemmatization in the classical sense means the transformation of words into a lemma, that is, into their original vocabulary form. In this paper, lemmatization means not only the transformation of words into a lemma, but also obtaining all the morphological properties of a word. There are a number of libraries that can perform lemmatization, for example, UDPipe [8], TreeTagger [9], myStem [10], etc. The myStem library is used to implement the algorithm. It has the following advantages over other ones:

- the ability to use for commercial purposes without changes,
- the library is supported by Yandex LLC,
- the full support of the Russian and the English languages,
- available for Java, Scala and other languages on the JVM platform,
- the data output format as json or xml.

The "Morphology" library was created for the data management after applying the myStem library. Its main function is to lemmatize a text and provide an API for conveniently extracting specific morphological properties of words [11]. When conducting the lemmatization, the problem of pronoun dereferencing is simultaneously solved.

Considering the punctuation and morphology in an aggregate makes it possible to construct syntactic models of the sentence processing, for example: highlighting the enumeration of words through ",", extracting a direct speech, highlighting homogeneous sentence members, complex-compound sentences, compound sentences, participial and participial sentences, as well as to establish logical, causal relationships between parts of a sentence.

Summarizing the previously mentioned, the following is a modified algorithm for collecting N-grams for the use in machine learning:

1) Extraction of text metadata is performed according to the model, if a text has a given structure.
2) The text is split to paragraphs.
3) Parallel processing of the lemmatization of each paragraph is performed with a simultaneous pronoun resolution.
4) We use syntactic models for parsing sentences in order to highlight causal and logical connections between senses (parts of output format separated by punctuation).
5) A semantic network is built between subjects / objects using predicate relations, while preserving causal relationships and logic between the meanings.
6) Dictionaries are used, with which one can restore additional properties of words for filtering and clustering data.
7) Extraction of N-gammas from a semantic network for the use in further steps in the LSA method, using the rules of compatibility and semantic properties of parts of a speech, stop words.

4. Computational experiments

Figure 2 is a flow chart of the steps of algorithmic text processing using the described rules implemented in Java.

The input to the algorithm is a text that is processed according to syntactic models and divided into paragraphs. At the second stage, paragraphs are enriched with the metadata of words using lemmatization (the process of obtaining the morphological properties of a word). The outcome of the
first and second stages are placed in the repository. At the third stage, paragraphs and their metadata fall into algorithm for extracting the key terms and building a semantic network, based on the implemented rules from Chapter 2. The key terms are extracted from the semantic network and the “term document” matrix is formed, according to the BM25 equation from Chapter 1. Values of the term-document matrix are stored in the tensor form. At the last stage, the SVD method is used to transform the original term-document matrix to two other matrices: term-theme and theme-document. Matrix values are also stored in the tensor form. There are various programs for visualizing tensors, for example, the Google Embedding projector.

Fig. 2 Algorithmic text processing

The implementation of figure 2 algorithms are easily parallelized assuming that a paragraph is a closed medium of meaning. Then, after breaking the text into paragraphs, it is possible to parallelize the data processing according to the principle of one paragraph - one computing processor, on which the semantic network of the paragraph is collected, and from which the keywords are extracted and the frequency matrix is built. The data of the frequency matrices are sent to the processor collector, on which they merge. Saving a frequency matrix is an intermediate stage, which will subsequently allow the partial addition to the original matrix of new terms obtained already from other documents.

We give examples of the program on simple sentences figure 3. The first column contains sentences in the Russian language. The second column contains the semantic network of each sentence. The third column contains N-grams obtained from the semantic network. The fourth column contains N-grams, which were obtained in a standard way. In the first example, 6 N-grams were generated by the modified method and 19 N-grams by the standard method. In the second example, 5 N-grams were generated by the modified method and 15 N-grams by the standard method. In the third example, 3 N-grams were generated by the modified method and 9 N-grams by the standard method. In the fourth example, 5 N-grams were generated by the modified method and 15 N-grams by the standard method. All N-grams extracted by the modified method have a semantic meaning. In the general context of N-grams extracted in a standard way, there are N-grams that have no semantic meaning. As was mentioned earlier, N-grams take into count the effects of the size of the TF-IDF matrix. Therefore, it is possible to speed up the calculations of the SVD method by reducing the number of N-grams using the modified method by factor of 3. Moreover, the modified collection method improves the meaning of the collected N-grams by eliminating the meaningless ones.
5. Conclusion
The paper presents a modified N-gram method, which is used in the LSA to construct the term-document matrix. Modifications of the method are based on the syntactic and linguistic algorithms that solve a number of problems arising during the application of a standard algorithm. The results of the developed parallel algorithm are presented, which implements the indicated linguistic rules for collecting N-grams, implements the developed models for identifying subjects / objects, identifying predicates, splitting sentences according to syntax, etc. Using simple examples, we have demonstrated the advantages of the method developed in comparison with the standard one, which are as follow:

- a reduced number of extracted N-grams, which further influences the rate of the SVD method due to reduced dimension of the TF-IDF matrix
- improved semantic content of a collection of N-grams by eliminating the meaningless ones.

Acknowledgments
This work was conducted within the framework of the budget project 0315-2019-0002 for ICMMG SB RAS. The reported study was funded by RFBR according to the research project № 18-01-00599
References

[1] Fedyushkin N A, Savinov I A, Fedosin S A. Latent semantic analysis of the text. Saransk, University: Mordovian State University named after N.P. Ogareva, February 11, 2018. http://izron.ru/articles/aktualnye-problemy-tekhnicheskikh-nauk-v-rossii-i-za-rubezhom-sbornik-nauchnykh-trudov-po-igitom-mezh/sektiya-2-informatika-vychislitelnaya-tekhnika-i-upravlenie-spetsialnost-05-13-00/latentno-semantichestkiy-analiz-teksta

[2] Landauer T K, Foltz P W and Laham D 1998 Introduction to Latent Semantic Analysis. Discourse Processes 25 259-284

[3] Fedyushkin N A and Fedosin S A 2017 A brief overview of text mining methods and models—Problems and achievements in science and technology The collection of scientific papers following the results of the international scientific-practical conference No. 4 (Omsk) p 102

[4] Robertson S E and Sparck Jones K 1976 Relevance weighting of search terms. Journal of the American Society for Information Science 27 https://doi.org/10.1002/asi.4630270302

[5] Daniel Jurafsky & James H. Martin. Speech and Language ProcessingCopyright. Draft of October 2, 2019

[6] Seliverstova O N 2004 Transactions on semantics (Moscow: Yaz. Slavs. Culture State Unitary Enterprise Smol. Region. Typ. named after V.I. Smirnov) p 959

[7] Rozental D E, Dzhandzhakova E V, Kabanova N P. Reference book on spelling, pronunciation, literary editing http://www.evartist.narod.ru/text1/20.htm.

[8] Lemmatization library link UDPipe : http://ufal.mff.cuni.cz/udpipe

[9] Lemmatization library link TreeTagger: https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[10] Lemmatization library link myStem: https://yandex.ru/dev/mystem/

[11] Ivanov A A, Petrov A S Certificate of state registration of a computer program №2020613495. «Morphology» library for storing and managing metadata of words in processed text. Registration date: 17.03.2020