Information Object Storage Model with Accelerated Text Processing Methods

Olesia Barkovska, Daria Pyvovarova, Vladyslav Kholiev, Heorhii Ivashchenko and Dmytro Rosinskyi

Kharkiv National University of Radio Electronics, Nauki ave., 14, Kharkiv, 61166, Ukraine

Abstract

The paper is devoted to the topical problem of structured organization of text documents electronic repositories on the example of electronic library system for storage and access to scientific works of researchers, teachers and students of educational institution. In the course of research, information objects storage model with modified and improved methods of accelerated processing of textual information was proposed, which consists of the following modules: search query pre-processing module; image information search module; keyword in the corpus searching module; database creation and maintenance module. An analysis of pre-processing methods was conducted to identify the possibility of implementation on mass parallelism systems, which showed the possibility and necessity of implementing methods of information search and construction of frequency dictionary on high-performance computer systems, as they have clear data parallelism tendency. The increasing in number and size of information objects makes the issue of source information processing time (classification, annotation, pre-processing) even more relevant than before. To solve this problem, the paper proposes the construction of a frequency dictionary using the computational resource of the graphic processor. The analysis of the obtained results showed that the proposed introduction of term weight sorting on systems with mass parallelism in the constructed frequency dictionary reduces the operating time of the syntactic level module of the proposed model by almost 18%. It is also apparent that for small amounts of data acceleration is almost absent. For large amounts of data, the acceleration is almost 100 times compared to the sequential sort used by default.

Keywords 1

Information object, weight, texts, vectorization, pre-processing, frequency dictionary, graphics processor, sorting, acceleration

1. Introduction

To understand the significance of information in the modern world, it is necessary to remember that its accumulation has been going on since ancient times. From the first years of its existence, humanity has used such natural information technology as language [1]. Later, along with speech, people began to use images and writing to store and transmit information. With the development of language and general culture of peoples there began to appear different types of writing in a "hard copy" format, a substitute for spoken language. The main purpose of writing is the function of storing information. Thus, the main task of writing is to record information on media and transmit it to other people [2, 3].

The increase in the amount of information always continues. This is confirmed by the increase in the number of media, while on the other hand, there was a change in storage technology, which

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine

EMAIL: olesia.barkovska@nure.ua (O. Barkovska); daria.pyvovarova@nure.ua (D. Pyvovarova); vladyslav.kholiev@nure.ua (V. Kholiev); heorhii.ivashchenko@nure.ua (H. Ivashchenko); dmytro.rosinskyi@nure.ua (D. Rosinskyi)
ORCID: 0000-0001-7496-4353 (O. Barkovska); 0000-0002-7251-994X (D. Pyvovarova); 0000-0002-9148-1561 (V. Kholiev); 0000-0003-1027-5262 (H. Ivashchenko); 0000-0002-0725-392X (D. Rosinskyi)

increases the concentration of information stored while reducing the size of the media (papyrus – parchment – birch bark – paper – punch card) [4].

The analysis of the costs and ways of storing and sharing information reveals accelerated pace of modern society development. However, this leads to problems such as reducing the speed of information processing and increasing the cost of information objects (IO) storage media.

An information object will be defined as a set of logically connected information stored on an information medium (paper, magnetic, electronic, laser…)

There are simple and complex information objects. Simple information objects are sound, image, text, number. Complex (structured) information objects include element, table, database, hypertext, and hypermedia [5].

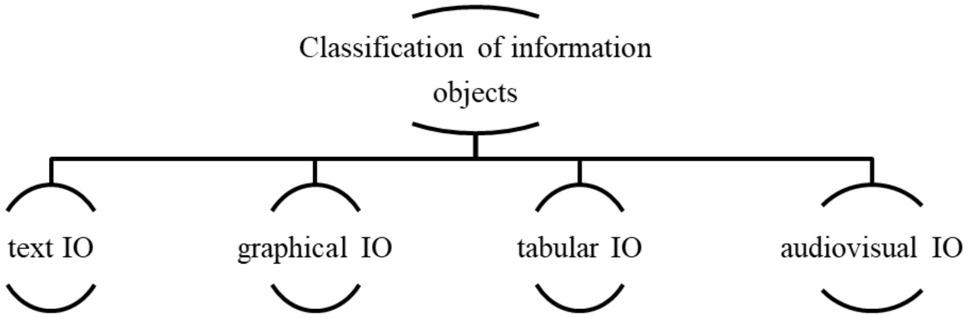


Figure 1: Classification of information objects

IO are divided into the following classes (Figure 1) [5]:

- text information objects – literary work, newspaper article, order;
- graphical information objects – paintings, drawings, diagrams;
- tabular information objects – various documents in tabular form;
- audiovisual information objects – video and music.

Most IOs are complex, i.e. contain information presented in various forms.

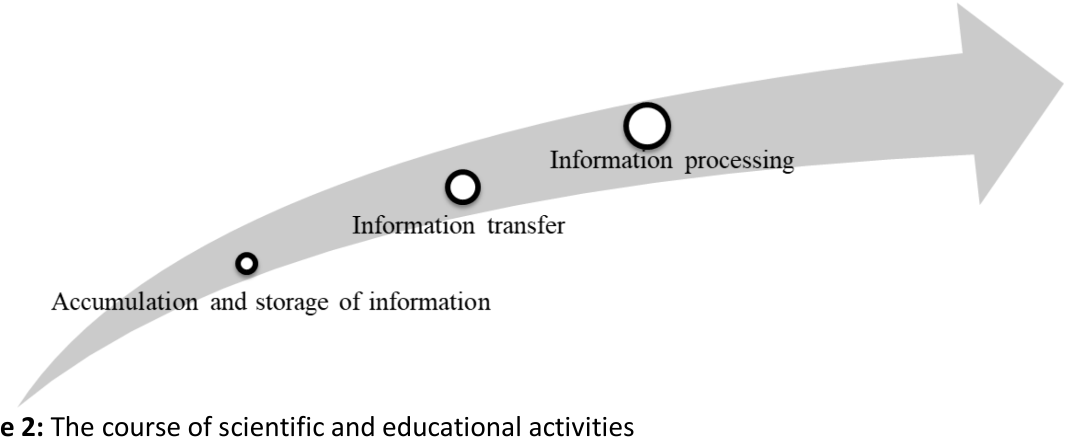


Figure 2: The course of scientific and educational activities

An approximate sequence of ways to spread the information can be represented as: the transfer of information "by word of mouth", through reading books, through the study and exchange of electronic information objects [4].

The advent of electronic file storage technology and its transport over the Internet has made it possible to create distributed electronic libraries, which in turn has led to the creation of virtual remote universities, where students and teachers can be separated by thousands of kilometers and be on different continents.

Electronic texts, in comparison with printed ones, are characterized by fundamentally new properties. This is due to the modern approach to their storage and distribution. Electronic texts open wide perspectives for linguists. This applies to the processing of large masses of information, taking into account additional classification and new approaches to solving traditional problems. This research area is collectively referred to as "Humanities Computing".

The problem today is the continuous increase in the amount of information, which leads to such requirements as increasing the speed of search engines and systems for categorizing information [7-9].

Among the types of libraries are thematic libraries (legal, medical, military, music, transport, philosophical and art libraries) and specialized-corporate, i.e. those that are relevant and in demand to a group of readers with a certain status, such as student, graduate student, researcher or young scientist (figure 3).

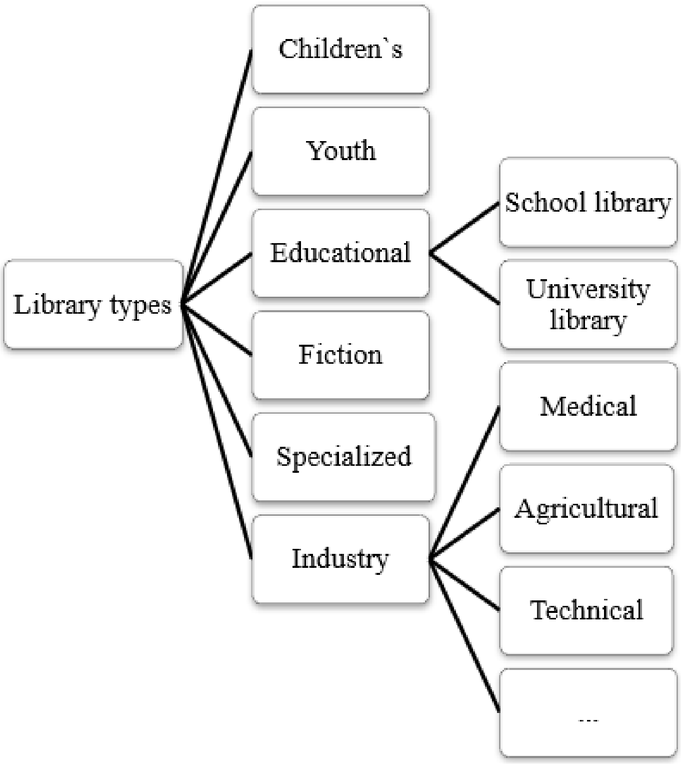


Figure 3: Libraries classification

The aforementioned classification is also relevant for electronic storage, as it provides easy access to target data, namely – scientific and research works of young scientists.

In engineering and linguistic practice, language is defined as the sound form of the text while the text is the written form of language.

2. Related Works

The transition from the traditional to electronic storage form is becoming widespread and is reflected in the functioning of libraries [6]. In this regard, the there is a transformation of libraries into electronic libraries. Accordingly, the functions and processes of their management change as well. Thus, the presentation of information in electronic form (the creation of electronic documents, their organization in the form of electronic publications, various electronic collections and electronic libraries) is a relevant task [7].

The main classes of problems encountered when working with text as a way of presenting information, as well as their practical application, are shown in Figure 4 [10-12]. The figure shows that some methods, such as text vectorization (calculation of TF-IDF measure, compliance with Zipf's and Heaps' laws) are significant in different areas and different tasks, which determines the relevance of the analysis and improvement of these methods.

The relevance of the text proximity detection problem, which is also based on text vectorization and the construction of a frequency dictionary, due to the widespread use of this task in the detection of plagiarism [17, 18], determining document authorship, information retrieval, machine translation, construction of tests and tasks, automatic abstract construction.

In [18] plagiarism detection system that uses artificial neural networks to cancel academic dishonesty with student's homework was created. Both accuracy and recall of plagiarism detection were improved by using ANN for filtering (a single hidden layer with 192 neurons, sigmoid symmetric (tansig) activation function) and for similarity improving (a single hidden layer with 64 neurons, sigmoid (logsig) activation function). All these results justify the relevance of plagiarism and similarity researching, but they don't analyze time of algorithm's execution.

In [17] such representative models as the LDA-based, tf-idf, vectors averaging, and paragraph vector method were researched for improving of F1 and accuracy of the semantic similarity determining for long scientific documents.

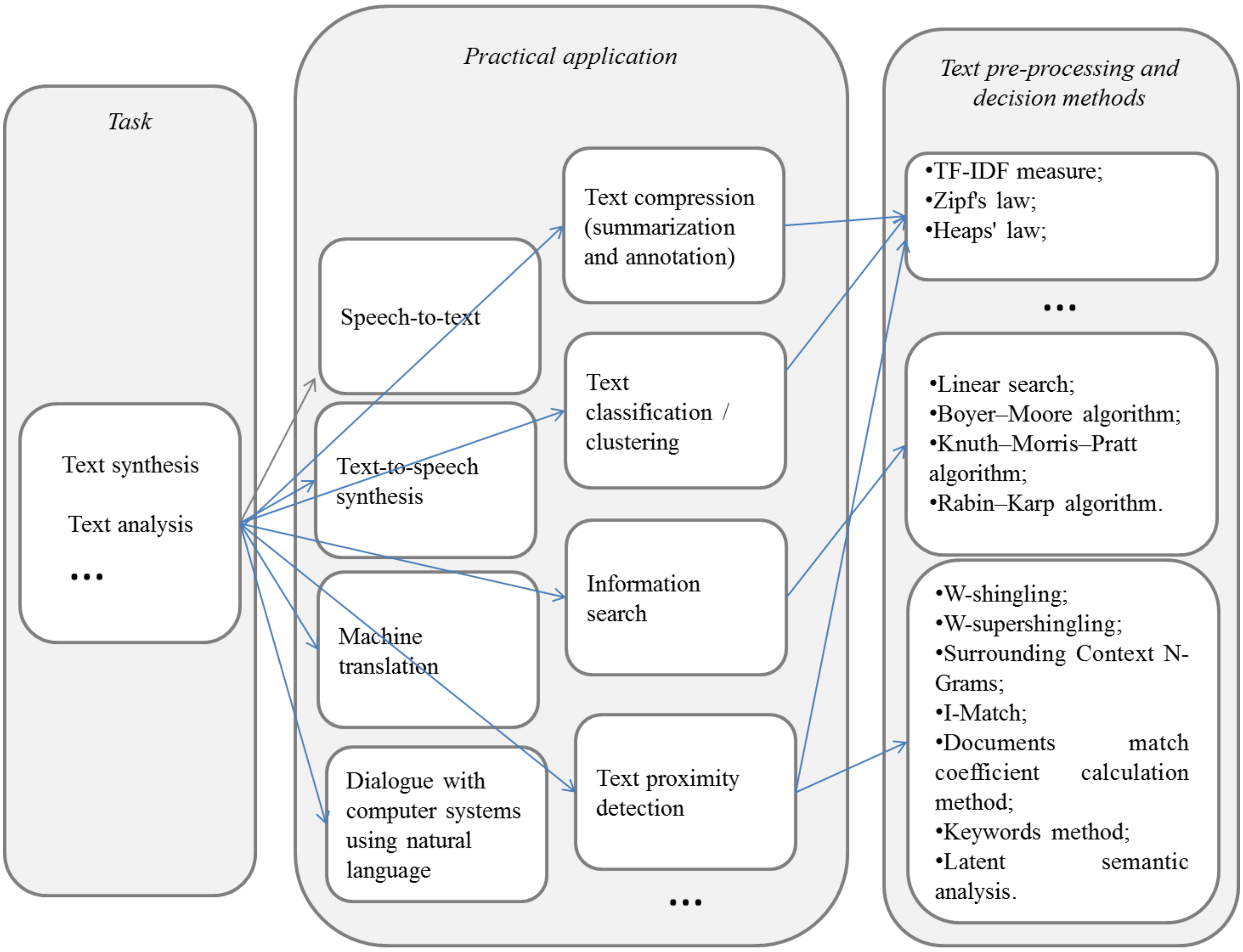


Figure 4: Analysis of problem area

Thus, the representation of texts in the form of vectors from some common to all texts vector space, can be considered one of the main stages of pre-processing, because it allows to consider text not as a set of tokens or symbols, but in a more computer-friendly way – in the form of a vector [13]. This approach is a basic tool in the field of text mining, information retrieval, classification and clustering of text documents.

In the classical vector model proposed Selton and others [13], the weights of terms are a set of local and global parameters. This model is known as tf-idf (term frequency – inverse document frequency).

Nowadays many researchers confirm the effectiveness of use tf-idf model for term weights determining in the text-summarization and sentence similarity tasks on the base of the word frequency and inverse document frequency [19]. According [20], tf-idf measure has a great influence on the sentence similarity calculation method based on multi model nonlinear fusion and the F1 value of the model.

Some words can be found in almost all documents of a collection and, accordingly, have little effect on the pushing a document to a particular category, and therefore aren't key to this document. To reduce the significance of words that occur in almost all documents, the inverse frequency of the term IDF is introduced (inverse document frequency) – this is the logarithm of the ratio of the number of all documents D to the number of documents d containing a word.

The key in this case will be the words with the most weight. Words with low weight, in general, can be ignored in the classification.

Thus, a term will have big weight if it occurs frequently in some texts while rarely in others. On the other hand, for common terms the weights will be small.

3. Aims and Tasks of The Work

The aim of the work is to create an information object storage model with accelerated text processing methods.

To achieve this aim, the following tasks must be solved:

- development of an information object storage model;
- analysis of pre-processing methods to identify the possibility of implementation on mass parallelism systems;
- research of the influence of the characteristics of the computer system on the implementation of a modified method for determining the weight of words in the text corpus;
- analysis of the results.

4. Results and Discussion

The study proposes the organizational model of electronic IO storage for storage and access to scientific works of researchers, teachers and university students (figure 3.2).

The proposed model consists of the following modules:

- search query pre-processing module;
- image information search module;
- keyword in the corpus searching module;
- database creation module.

Organization of storage according to figure 5 can be divided into two stages – information accumulation and access to information. The operation of each stage consists of operation of the individual algorithms described below.

The input of the model receives a large number of documents in different formats (txt, doc, pdf, etc.), the model selects a code library depending on the format of the source document and extracts data, namely – keywords, from the document in the form of updated text. The selected keywords are used as input values at the stage of classification and construction of the cataloger. An important stage of the algorithm is that the previous stage of classification is the filtering of the text on the stop list (short words and punctuation marks that do not carry any semantic load for further analysis), which reduces the volume of text and increases its semantic value.

The importance of the query pre-processing module is to normalize the text. Normalization involves reducing words to a normal form – the canonical form of the word. For example, for nouns the initial form of a word is a singular form in the nominative case, for adjectives it is a singular adjective and in the nominative case without a preposition. This transformation does not cause much loss, because a particular form of the word rarely has useful information (the meaning of the word remains the same). Often there are tasks with a large amount of source data, and therefore it is desirable to reduce the number of properties. By reducing the words to the original form, the number of unique words is also reduced.

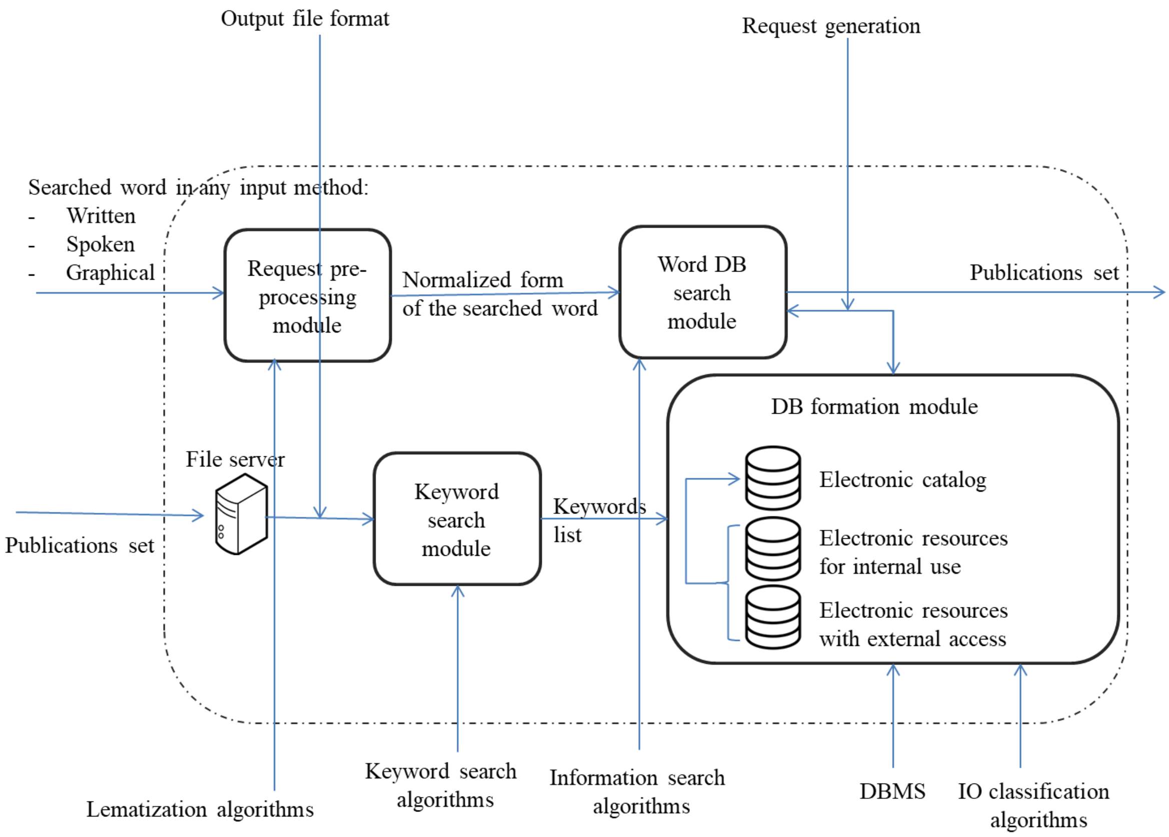

Output file format
Request generation
Searched word in any input method:
- Written
- Spoken
- Graphical
Request pre-processing module
Normalized form of the searched word
Word DB search module
Publications set
DB formation module
File server
Publications set
Keyword search module
Keywords list
Electronic catalog
Electronic resources for internal use
Electronic resources with external access
Lematization algorithms
Keyword search algorithms
Information search algorithms
DBMS
IO classification algorithms

Figure 5: The proposed organizational IO electronic storage model for storing and accessing scientific work of researchers, teachers and university students

The implementation of normalization can be divided into two approaches: stemming and lemmatization.

The task of the database word search module is to determine whether the searched word (term, image), which consists of a number of characters, is included in the text (line, corpus). [14-16] If the word is successfully found, the module gets a reference to the document that contains the word, as well as its annotation.

The keyword search module is an extremely important and necessary step that precedes the direct classification of new documents to fill the catalog. The importance of the module is to reduce the dimension of the property space, which can reduce the effect of overtraining – a phenomenon in which the classifier focuses on random or erroneous characteristics of educational data, rather than on important and significant ones. This stage can significantly reduce the dimension of the problem solving and classification accuracy. To do this, the TF-IDF method can be used [29], on the input of which the document comes in indexed form (numerical model of the text). Word bag, N-gram or Word 2VEC models can be used for indexing.

According to Harris's distributive hypothesis, words with similar meanings will occur in similar contexts.

4.1. Information Accumulation in The Organization of IO Storage

The logical representation of text documents is limited to the use of information available after pre-processing. To do this, the sequence of pre-processing methods includes levels of grapheme (selection of tokens - individual words in the text), morphological (definition of grammatical forms and categories of words) and syntactic analysis (Figure 6).

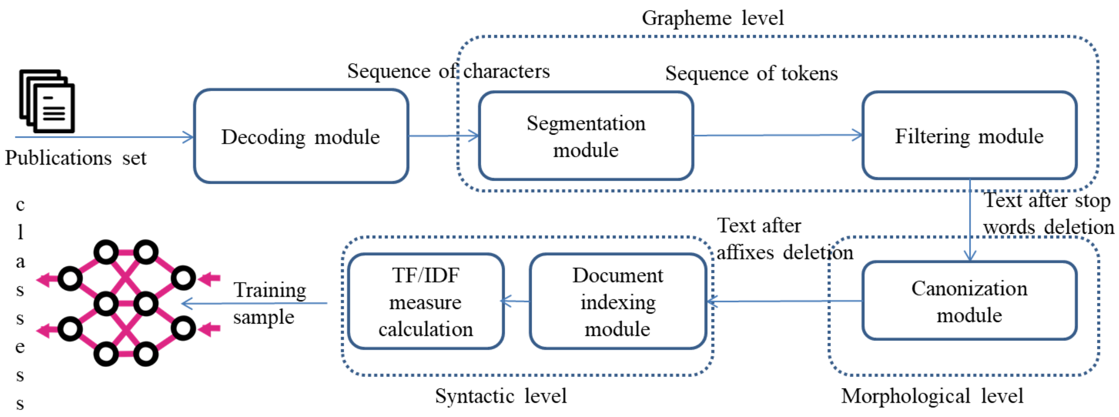
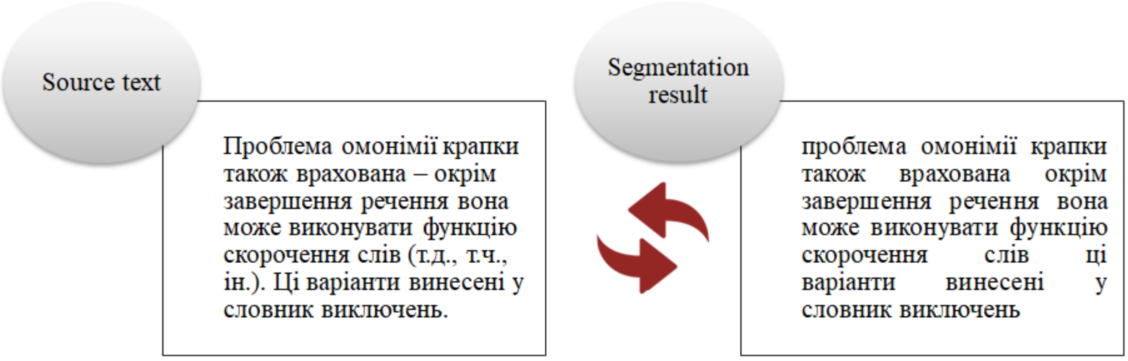


Figure 6: The stage of information accumulation in the organization of IO storage

Text segmentation (Figure 7) involves the division of text into sentences. In the simplest case, segmentation is performed on the basis of end-of-sentence markers – dots (three dots), exclamation mark or question mark. In the work, tokenization is performed taking into account that the semicolon also indicates the end of the sentence, as this sign is often used to separate individual independent parts of the sentence. In addition, the selection of simple sentences also occurs in the case of opposing conjunctions – and, but, however, still, however, though. This is an important aspect of segmentation, as simple sentences that are separated by opposing conjunctions are likely to have different tones. The problem of homonymy of a dot is also taken into account – in addition to the completion of a sentence, it can perform the function of abbreviating words (e.g., i.e., etc.). These options are listed in the dictionary of exceptions.



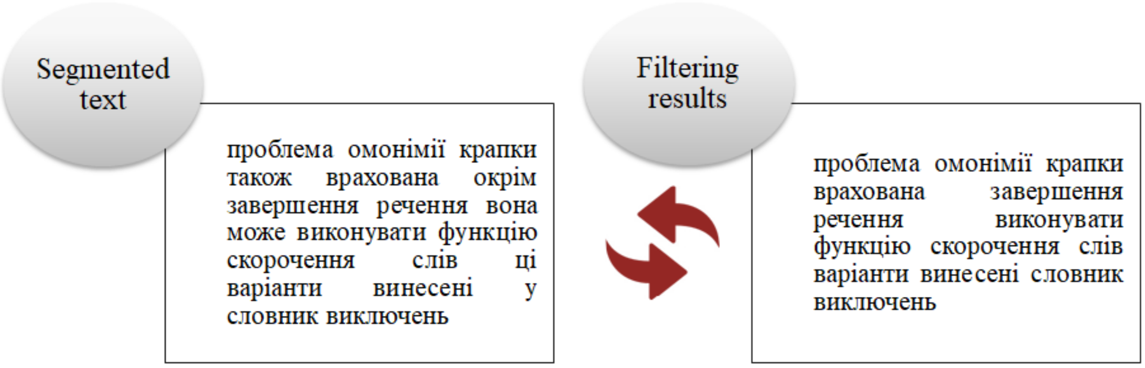
Проблема омонімії крапки також врахована – окрім завершення речення вона може виконувати функцію скорочення слів (т.д., т.ч., ін.). Ці варіанти винесені у словник виключень.

проблема омонімії крапки також врахована окрім завершення речення вона може виконувати функцію скорочення слів ці варіанти винесені у словник виключень

Figure 7: Segmentation results

Further filtering (Figure 8) is to remove stop words which are words that have no informative load on the content of the text. Such words include functional words (which are semantically neutral, such as conjunctions, articles, prepositions…).

Lemmatization brings tokens to a unified form, which allows to get rid of the difference in the spelling of the word (Figure 9). The algorytm is performed on the basis of the formed set of rules Paice/Husk algorythm: the word equals the base + affixes. This algorithm aims to iteratively remove the ending of a word. It uses a table of rules to replace endings and suffixes and relies on the last letter in the word, which makes it effective to search for rules in the general table. The deletion goes on as long as there are rules in the table corresponding to that word.



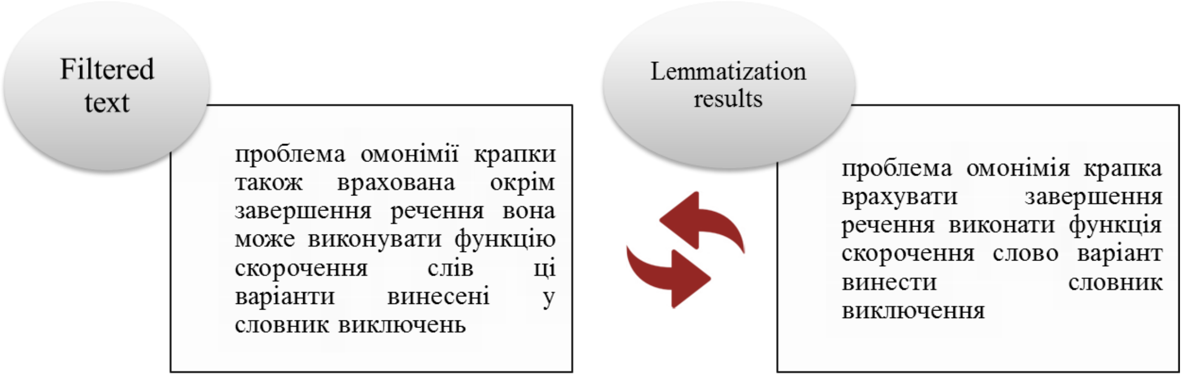
проблема омонімії крапки також врахована окрім завершення речення вона може виконувати функцію скорочення слів ці варіанти винесені у словник виключень

проблема омонімії крапки врахована завершення речення виконувати функцію скорочення слів варіанти винесені словник виключень

Figure 8: Filtering results

By deleting affixes, the word takes the form of a lemma sufficient for further processing, namely, constructing a frequency dictionary and determining the importance of words based on the TF/IDF algorithm, which estimates the importance of the word within a document.

- Suppose there is a collection of three documents:
- The problem of dot homonymy is taken into account;
- The problem of homonymy, word homonymy;
- There is a problem with the dot in the text.



проблема омонімії крапки також врахована окрім завершення речення вона може виконувати функцію скорочення слів ці варіанти винесені у словник виключень

проблема омонімія крапка врахувати завершення речення виконати функція скорочення слово варіант винести словник виключення

Figure 9: Lemmatization results

Frequency dictionary (lexical units are characterized in terms of the degree of their use in a set of texts or for the language as a whole, or for a particular topic, or for a single document) with the included measure IDF will look like table 1.

Table 1
Frequency dictionary

Word	Total	Encountered in documents	IDF
Проблема	3	3	0
Омонімія	3	2	0,18
Крапка	2	2	0,18
Врахувати	1	1	0,47
Слово	1	1	0,47
Текст	1	1	0,47
Виникати	1	1	0,47

Frequency dictionaries are widely used both in computational linguistics (for example, for the text classification) and in traditional linguistics (for example, to compare the vocabulary of different authors, analysis of changes in vocabulary over time, etc.).

Further work lies in classification the source pre-processed documents on the basis of one of the classifiers.

The aforementioned functionality and capabilities of the system determine the relevance of this development, as well as make a number of requirements:

- Scalability of the data storage;
- Guarantee of the integrity, reliability, confidentiality and fault tolerance of the system;
- Implementation of processing of various types of search queries (voice, text, scanned and photo media) and accelerated data search in the system, based on the specified area of user interest;
- A high degree of accuracy and completeness of the division of records into classes with determining the most pressing tasks and problems;
- In-depth analysis of uploaded work to group possible research teams.

4.2. Determining the Frequency Dictionary Sorting Time in Sequential Implementation

Among the vectorization methods, the results of which can be used for further classification of the text, the method of "word bag" was analyzed in conjunction with the TF-IDF model. Often the document is dominated by the words that are very common, but they contain not so much the "information content" of the model, as are more rare, but specific to the subject area. To avoid this problem, it is necessary to determine the IDF measure. By default, the result of the TF-IDF model is a constructed dictionary with the weight of each word – a frequency dictionary. The key in this case will be words that have a measure of IDF in the middle range of values (for words that occur in a large number of documents, IDF will be close to zero (if the word occurs in all documents IDF is zero), which indicates the semantic importance of the word) . The lowest IDF is for commonly used words, the highest for unique words in a document.

The paper made the following thresholds for classification (Table 2).

Table 2

Selection of the number of words that will participate in the classification depending on the word count in the source text

Word count of the source text	Frequency dictionary words involved in classification
Less than 1000	All the words in the dictionary
Less than 10000	Starting from the 100th + 40%
Less than 100000	Starting from the 100th + 20%
More than 100000	Starting from the 100th + 20000 words

Such indicators are justified, because in small texts a sufficient number of words of the ordered frequency dictionary, which are fed to the input of the classifier is 40%. Adherence to such a percentage for large texts is not appropriate, because it significantly increases the operating time of the classification algorithm, while not increasing the accuracy of classification.

Selecting meaningful words in an ordered array leads to the need to quickly sort and organize a large amount of data - words in the text. Thus, the sequence of actions will be shown in Figure 10.

Arranging the IDF values of a large number of terms is a time-consuming operation because a sequential algorithm is used by default. The paper conducts an experimental study on the effect of using different sorting algorithms for keywords selection time.

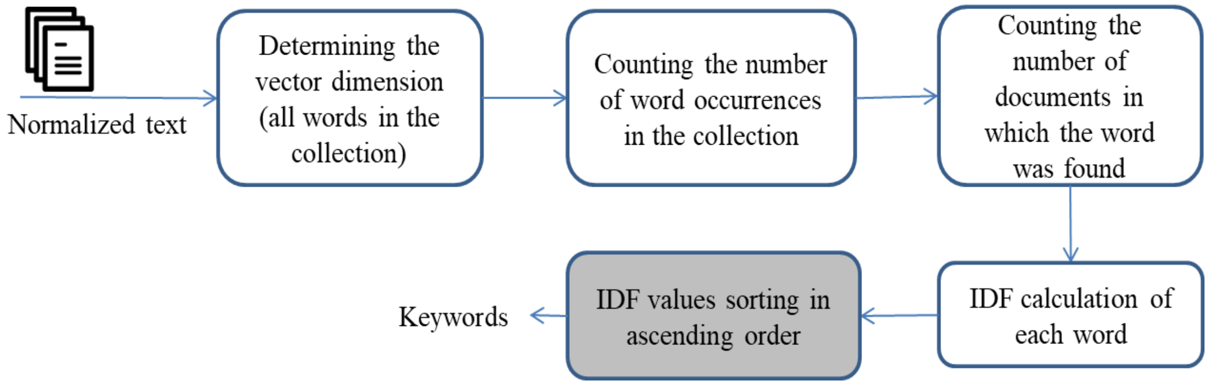

Normalized text
Determining the vector dimension (all words in the collection)
Counting the number of word occurrences in the collection
Counting the number of documents in which the word was found
IDF calculation of each word
IDF values sorting in ascending order
Keywords

Figure 10: Detailed algorithm of syntactic level units of the proposed model

The results of keyword selection for different sized collections with standard ordering of IDF values are shown in table 3.

The table of results shows that the sorting time of the defined weights of terms for large dictionaries is more than 8% of the time. That is why reducing the ordering time can give good results.

Table 3
The execution time of sequential syntactic processing of the document collection in accordance with Figure 10

Collection size	Keywords Count	Sorting time, *sec*	The total algorithm operating time, *sec*
10	10	0	4,8
100	100	0,9	17
1000	1000	4,6	200
10000	From 100th по 4100 word	26,5	509
100000	From 100th to 20100th word	57	2124
1000000	From 100th to 20100th word	407	3654
10000000	From 100th to 20100th word	923	7521

4.3. Determining the Frequency Dictionary Sorting Time in Parallel Implementation on Shared Memory Systems

The results of keyword selection for different sized collections with parallel sorting of IDF values based on the Qsort algorithm are shown in Table 4. The implementation is performed on a system with shared memory and multithreading (OpenMP) platform.

One element of the array is assigned as a pivot. The elements of the array then are rearranged so that all that are less than the reference are moved to the left of it, and those that are greater – to the right. For each of the subarrays, the operation is repeated recursively. The effectiveness of the algorithm depends greatly on how well the pivot element will be chosen. The ideal case is when the algorithm constantly divides the subarrays equally, but otherwise, the time will be lost on the calculation. Different modifications mainly differ from each other in the way of selecting the pivot element and in the division into subarrays. The study uses the Hoare partition.

The computational complexity of the Qsort algorithm is $(n * \log n)$.

The Intel (R) Core (TM) i5-3210M CPU is used for calculations with four cores loaded.

Table 4
Syntax processing time of a collection of documents with parallel ordering of term weights on a system with shared memory

Collection size	Keywords count	Sorting time, *sec*	The total algorithm operating time, *sec*
10	10	0	4,65
100	100	0,6	15,4
1000	1000	3,8	174
10000	From 100th to 4100th words	18,23	480
100000	From 100th to 20100th word	31,4	2006
1000000	From 100th to 20100th word	206,5	3400
10000000	From 100th to 20100th word	321,8	6354

The results table shows that the proposed sorting reduces the operating time of the syntactic level block of the proposed model, but still takes a long time. It is seen that for small data amounts acceleration is almost absent. For large data amounts, the acceleration is almost 3 times.

4.4. Determining the Frequency Dictionary Sorting Time in Parallel Implementation on Systems with Mass Parallelism

The results of keyword selection for different sized collections with parallel IDF values soring based on the LSD (least significant digit) algorithm with calculation (Radix Sort) are shown in Table 5. Implementation is performed on a system with mass parallelism. The computational complexity of the algorithm is O(n).

Table 5
Execution time of syntactic processing of a documents collection with parallel weight sorting of terms on a system with mass parallelism

Collection size	Keywords count	Sorting time, *sec*	The total algorithm operating time, *sec*
10	10	0	4,71
100	100	0,8	15,5
1000	1000	1,4	169
10000	From 100th to 4100th words	2,06	457
100000	From 100th to 20100th word	2,4	1940
1000000	From 100th to 20100th word	3,56	2987
10000000	From 100th to 20100th word	8,48	6129

The main feature of the algorithm is high efficiency for highly mixed or random data sets. It makes sense to use other algorithms on almost sorted sets, as the gain will not be so significant. Works poorly on small arrays, with less than a couple of hundred elements. It demonstrates that Radix Sort is the best sorter in terms of high-speed applications for systems with mass parallelism, like GPU are.

The computing resource of the NVIDIA GeForce GT 650M graphics processor with Kepler architecture and the maximum number of 384 CUDA cores is used for calculations.

The reduction of the terms sorting time based on sequential and parallel algorithms is shown in Figure 11.

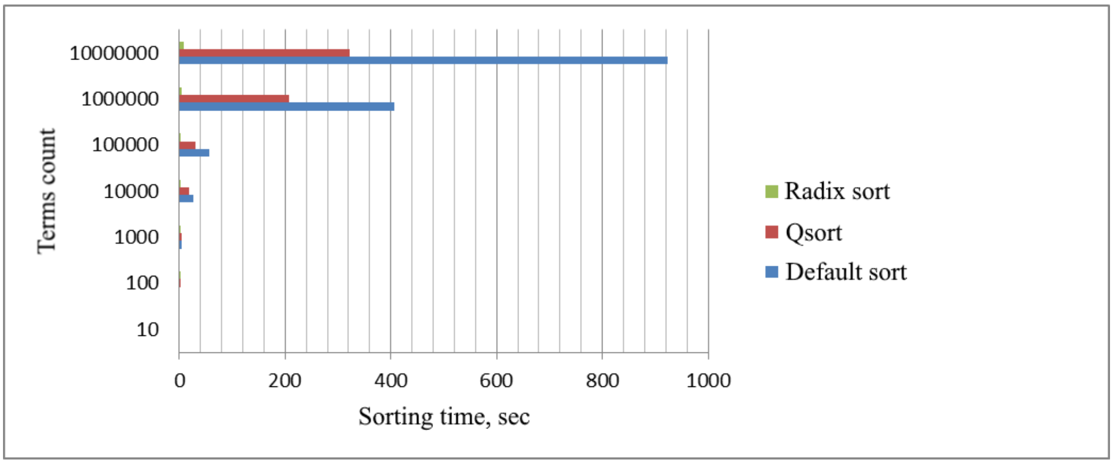


Figure 11: Term sorting time

The resulting acceleration gained for syntactic processing unit is shown in Figure 12.

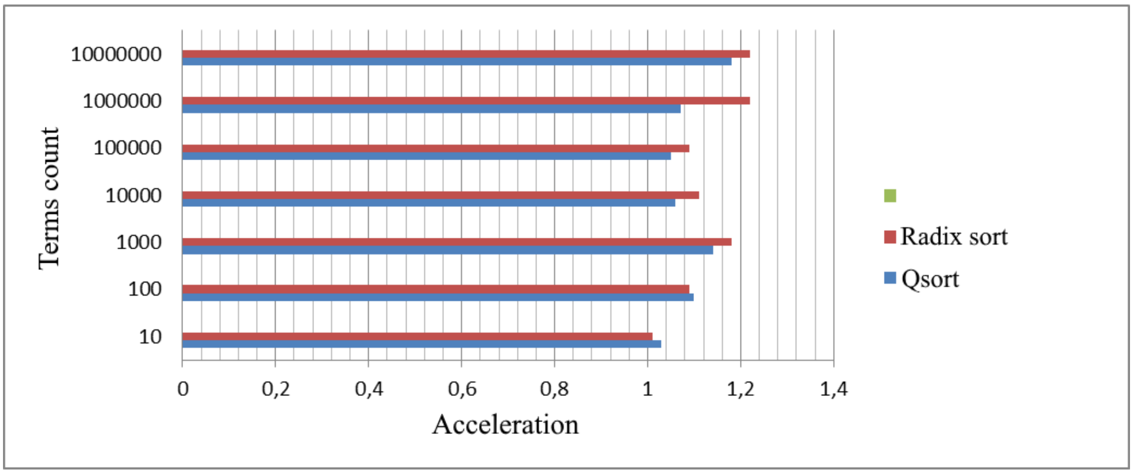


Figure 12: The resulting acceleration of syntactic processing unit

The analysis of the obtained results showed that the proposed term weight sorting on systems with mass parallelism in the constructed frequency dictionary reduces the operating time of the syntactic level block of the proposed model by almost 18%. Meanwhile, small amounts of data showed almost no acceleration. For large amounts of data however, the acceleration is almost 100 times compared to the sequential sort used by default.

5. Conclusion

In the course of research, information object storage model with modified and improved methods of accelerated text processing was proposed, which consists of the following modules: search query pre-processing module; image information search module; keyword in the corpus searching module; database creation module. The following tasks were also solved:

- an analysis of pre-processing methods was conducted to identify the possibility of implementation on mass parallelism systems, which showed the possibility and necessity of implementing methods of information search and construction of frequency dictionary on high-performance computer systems, as they have clear data parallelism tendency;
- a study was conducted of the influence of the computer system characteristics on the implementation of a modified method of determining the weight of words in the text body through the use of algorithms for accelerated elements sort.

The analysis of the obtained results showed that the proposed term weight sorting on systems with mass parallelism in the constructed frequency dictionary reduces the operating time of the syntactic level block of the proposed model by almost 18%. Meanwhile, small amounts of data showed almost no acceleration. For large amounts of data however, the acceleration is almost 100 times compared to the sequential sort used by default.

Further research will be related to the development of the model (figure 5) and the expansion of functionality, for example, voice input of a query, the ability to accelerate the search for fragments of scanned documents.

6. References

[1] R. U. Ayres, Information, Entropy, and Progress: A New Evolutionary Paradigm. Front Cover. Aip Press, 1994.

[2] H. P. Yockey, Information Theory, Evolution, and the Origin of Life. Cambridge University Press, Cambridge and New York, 2005.

[3] B. Skyrms, Signals: Evolution, Learning, and Information, Oxford University Press, Oxford and New York, 2010.

[4] J. Avery, Information Theory and Evolution. World Scientific, 2003.

[5] M. Dalrymple, I. Nikolaeva, Objects and information structure, Cambridge University Press, Cambridge 2011.

[6] O. C. Agbonifo, O. S. Adewale, Information revolution through Information and Communication Technology, in: Proceedings of the 2010 Second Region 8 IEEE Conference on the History of Communications, Madrid, Spain, 2010, pp. 1-6, doi: 10.1109/HISTELCON.2010.5735301.

[7] G. Shi, M. Li, M. Lipasti, Accelerating search and recognition workloads with SSE 4.2 string and text processing instructions, in: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (IEEE ISPASS), Austin, TX, USA, 2011, pp. 145-153, doi: 10.1109/ISPASS.2011.5762731.

[8] R. Polig et al., Hardware-accelerated text analytics, in: Proceedings of the 2014 IEEE Hot Chips 26 Symposium (HCS), Cupertino, CA, USA, 2014, pp. 1-24, doi: 10.1109/HOTCHIPS. 2014. 7478822.

[9] R. Takahashi, U. Inoue, Parallel Text Matching Using GPGPU, in: Proceedings of the 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Kyoto, Japan, 2012, pp. 242-246, doi: 10.1109/SNPD.2012.28.

[10] J. Eisenstein, Introduction to Natural Language Processing. The MIT Press. 2019.

[11] Y. Zhang, Zh. Teng, Natural Language Processing. A Machine Learning Perspective. Cambridge University Press, Cambridge, 2021.

[12] V. Sowmya, M. Bodhisattwa, G. Anuj, H. Surana, Practical Natural Language Processing. A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media, 2020.

[13] G. Salton, C. Buckley. Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24 (5). 1988. pp. 513-523.

[14] S. Zaiceva, O. Barkovska, Analysis of Accelerated Problem Solutions of Word Search in Texts, in: Proceedings of The Fourth International Scientific and Technical Conference «COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES». Kharkiv: NURE. 2020. p. 66 DOI: https://doi.org/10.30837/IVcsitic2020201445

[15] O. Barkovska, O. Mikhal, D. Pyvovarova, O. Liashenko, V. Diachenko, M. Volk, Local Concurrency in Text Block Search Tasks, International Journal of Emerging Trends in Engineering Research. Volume 8. 3, March 2020. pp.6 90-694.

[16] O. Barkovska, D. Pyvovarova, V. Serdechnyi, Pryskorenyj alghorytm poshuku sliv-obraziv u teksti z adaptyvnoju dekompozycijeju vykhidnykh danykh. [Accelerated word-image search algorithm in text with adaptive decomposition of input data]. Systemy upravlinnja, navighaciji ta zv'jazku, 4 (56), 28-34. (in Ukrainian)

[17] M. Liu, B. Lang, Z. Gu, A. Zeeshan, Measuring similarity of academic articles with semantic profile and joint word embedding, in: Tsinghua Science and Technology, vol. 22, 6, pp. 619-632, December 2017, doi: 10.23919/TST.2017.8195345.

[18] V. Ljubovic, E. Pajic, Plagiarism Detection in Computer Programming Using Feature Extraction From Ultra-Fine-Grained Repositories, in IEEE Access, vol. 8, pp. 96505-96514, 2020, doi: 10.1109/ACCESS.2020.2996146.

[19] J. Ding, Y. Li, H. Ni, Z. Yang, Generative Text Summary Based on Enhanced Semantic Attention and Gain-Benefit Gate, in IEEE Access, vol. 8, pp. 92659-92668, 2020, doi: 10.1109/ACCESS.2020.2994092.

[20] P. Zhang, X. Huang, Y. Wang, C. Jiang, S. He, H. Wang, Semantic Similarity Computing Model Based on Multi Model Fine-Grained Nonlinear Fusion, in IEEE Access, vol. 9, pp. 8433-8443, 2021, doi: 10.1109/ACCESS.2021.3049378.