Structural Modeling of Technical Text Analysis and Synthesis Processes

Rostyslav Bekesh[0000-0003-2074-1212]1, Liliya Chyrun[0000-0003-4040-7588]2, Petro Kravets[0000-0001-8569-423X]3, Andriy Demchuk[0000-0001-5710-9347]4, Yurii Matseliukh [0000-0002-1721-7703]5, Taras Batiuk[0000-0001-5797-594X]6, Ivan Peleshchak[0000-0002-7481-8628]7, Roman Bigun[0000-0002-1235-7475]8, Igor Maiba[0000-0003-0667-7752]9

Lviv Polytechnic National University, Lviv, Ukraine

bekesh@mail.ru[1], lchirun21@gmail.com[2], petro.o.kravets@lpnu.ua[3], Andrii.B.Demchuk@lpnu.ua[4], indeed.post@gmail.com[5], tbatjuk4u@gmail.com[6], peleshchakivan@gmail.com[7], bigunroman@ukr.net[8], igor.maiba@gmail.com[9]

Abstract. The article presents the application of generative Grammars in linguistic modeling. Description of sentence syntax modeling is used to automate the processes of analysis and synthesis of natural texts.

Keywords. Generative grammar, structured scheme sentences, computer linguistic system.

1 Introduction

A feature of the development of modern Ukrainian scientific and technical terminology is the increased interest in its authenticity, since historically; this terminology has become inaccessible to users [1-4]. From official dictionaries and textbooks this terminology was withdrawn, and the forbidden dictionaries got to special storages of libraries, and them issued only by special permission [12-16]. To date, the dictionaries of 1920-1930 came in single copies, or did not come at all-they are lost or destroyed. Even the existence of many terminological dictionaries is now known only to a narrow circle of specialists [2, 5-11, 17-20]. It is estimated that about 90 % of new words appearing in each language are terms. Modern Ukrainian terminology is also actively updated with new units-mainly borrowings from English or words-tracing from Russian. Despite the fact that the Ukrainian language partially assimilates other people's words, still a large number of borrowed words poses a threat to the clarity of the national term system and often negatively affects the speed of the educational process. It is gratifying that in some new borrowings in Ukrainian terminology have already arisen correspondence, for example: *трастове товариство – довірче товариство*, *апроксимація – наближення*, *детектор – виявляч*, etc. If this trend continues, the majority of "fashionable" borrowings will go into passive reserve - will remain meaningful necessary terms [20]. From one language to other terms are not translated like ordinary words. The optimal way is to translate the terms: "concept - > Ukrainian term" and not "foreign language term - > Ukrainian term", from which language the translation would not occur (V. Morgunyuk). That is, the search for an analogue term begins with the analysis of the properties of a new concept. Unfortunately, in most cases, the translation of terms into Ukrainian occurs by "tracing paper" [20-27].

2 Connection of the Problem with Important Scientific and Practical Tasks

To date, there are many different modifications of morphological analyzers, mainly Russian, which are successfully used in a number of industrial software products.

Parsers LLC "Diktum" use analyzers of Russian and English language for morphological analysis and extraction of grammatical information about word forms before analysis. The Russian spelling and grammar control system Propis 4.0 was the first industrial application of the Russian morphological analyzer. The system uses so-called "first clone", where it was then sold and used now the approach to the pages of the dictionary, however, was not yet fixed identifiers (numbers) of tokens, it was not possible to synthesize the ID token, and had no concept of the ID forms of the word. Extended additive grammatical descriptions, which are still present in the structure of grammatical information, were used to describe word forms.

The technology of morphological analysis of the Ukrainian language is now working on the search engine < META>. There you can also test the morphological analyzer in the on-line mode [19]. The process of development of modern society is characterized by the constantly increasing role of information technologies in science, production and management. In recent years, the volume of information flows and the complexity of orientation in information resources have increased significantly, which has led to the need to find new ways of storing, presenting, formalizing, systematizing and processing information in computer systems [27]. Traditional technologies in the new conditions do not solve the problems of navigation of information resources, providing access to information, search for files and documents at the proper level. One of the results of research conducted in recent years to overcome these difficulties was the emergence of ontological technologies and their use in information systems [28]. At its core, domain ontology is a formal model of the structure of domain concepts [29]. In Gruber's famous formulation [27], ontology is defined as "a formal specification of conceptualization that takes place in some domain context". Under conceptualization we understand representations of the subject area through the description of the set of concepts (concepts) of the subject area and the relations (relations) between them. By creating ontology, a formalized representation of the structure of the subject area is formed agreed between specialists [30]. The purpose of the work is to develop an intelligent system for modeling the processes of analysis and synthesis of technical texts, namely: checking the correctness of the use of terms in articles according to well-known rules and the possibility of constructing ontology of these articles. The created system is based on morphological analysis algorithms, namely: it is based on a modified morphological analyzer. The difference between the created system and the existing morphological analyzers is its narrow specialization in the search for terms in technical texts, in particular articles. The object of this work is morphological systems-text analyzers. The subject of the study is the algorithms of morphological analysis, stemming and automatic ontology construction. The theoretical value of the work lies in the analysis of the known algorithms for morphological analysis, stemma and methods for building ontology. The practical significance of the results is the implementation of the composition of methods of morphological analysis and stemming in order to improve the efficiency of the search for incorrect words and phrases in the text and the possibility of constructing ontology.

3 Analysis of Recent Research and Publications

At the present stage it is possible to trace 5 approaches to the solution of problems of ordering of the Ukrainian scientific and technical terminology [2-11, 17, 19-20, 27].

- The 1st approach is formal. The main thing for him is a quantitative indicator-the earliest publication of the dictionary. Haste in terminology is not helpful – it is at best. In the worst-shakes terminosystem, gives wrong reference points for users. "Terminology is not a field for gaining fame, for Cossacks. This... but ant legendary labor, most often quite underestimated" (A. Vovk).
- The 2nd approach is ethnographic. It is based on the idea of reviving the national terminology of the Institute of Ukrainian Scientific Language. The creators of dictionaries seek to return to modern Ukrainian terminology almost all the terms of the beginning of the century.
- The 3rd approach is conservative. Its supporters advocate the preservation of Ukrainian scientific and technical terminology in the form that it acquired during the Soviet period. This is the so-called "real language" principle.
- The 4th approach is international. It is characterized by the introduction into Ukrainian scientific and technical terminology of a large number of borrowings from Western European languages, especially from English.
- The 5th approach is moderate. It provides for the ordering of Ukrainian scientific and technical terminology taking into account historical, national, political factors and the development of its optimal variant [20].

Modern Ukrainian terminologists deeper than their predecessors of the beginning of the century, develop the theory of the term as a language sign, the theory of terminology as a subsystem of the General literary language [31-37].

It is believed that terminology, as well as General literary language, should be characterized by such factors: perfection, economy, consonance. Under the perfection of terminology, the researcher understands its clear grammatical structure, logic and motivation of terms; under economy - its informativeness, ease of study, shortness of time-units; under consonance-euphony, articulation and spelling of terms [20].

The following time requirements are formulated [38-41]:

1. Content is the exact correspondence of the word to the concept, transparent internal form of the term;
2. Plasticity, or flexibility-the ability to form derivative terms;
3. Language perfection-brevity, euphony, ease of memorization;
4. Compliance with international standards.

According to such criteria, L. Petrukh, B. knight and other researchers are advised to evaluate terminological units. When determining the basic principles of term formation, Ukrainian terminologists rely on the experience of European science in this matter – such well-known terminologists as E. Wuster, D. S. Lotte, O. O. reformed, S. Bally and others worked on the development of the image of the ideal term. Given the foregoing, and taking into account the peculiarities of Ukrainian terminology in recent decades, one can identify current problems, the solution of which will depend on the direction of further development of Ukrainian scientific language [20]. The peculiarity of the development of Ukrainian scientific and technical terminology is the increased interest in the terminological achievements of the Taras Shevchenko Scientific society and the Institute of Ukrainian Scientific Language [20].

Automatic generation of hypotheses about paradigms of change of unfamiliar words gives the chance to automate process of filling of bases. In the transition to a new subject area, the question arises about the incompleteness of the morphological dictionary. Each subject area uses its own vocabulary. In this regard, there is a question of replenishment of dictionaries. This process can be automated if the available morphological analysis module allows predictions of lexical parameters of unfamiliar words. To do this, it is necessary to select all the words that are not in the existing morphological dictionary and perform an analysis on them with a forecast. The results of the analysis, is the tuple of the word form

$$< f_{nf}, r, P_{constt}(r,s) \cup P_{var}(r,s) >, \quad (1)$$

where $f_{nf} =< s_{nf}, r, P_{var}(r,s) >$ is the token normal form, r is part of speech word forms, s i s_{nf} is the analyzed token (word string) and the normal form token, and P is parameter sets. According to the results of the analysis, we can combine all words that have the same normal-form tokens into a single hypothesis. In the course of hypotheses, several strong but intuitively correct positions can be used [1]. There are several variants of algorithms stemma, which are characterized by their precision and performance [42-46].

Search in the table. This algorithm uses the principle of search on the table in which all possible variants of words and their forms after stemming are collected [47-51]. The advantages of this method are the simplicity, speed and convenience of handling exceptions to language rules. The disadvantages include the fact that the search table must contain all forms of words: that is, the algorithm will not work with new words (and as you know, "living" languages are constantly updated with new words) and the size of such a table can be significant. For languages with a relatively simple morphology, such as English, the size of the search table is quite modest, but in agglutinative languages, such as Turkish, the number of variants of words with one root can go to hundreds. This algorithm is based on the rules by which you can shorten a word. If we take the example of the search algorithm on a table, these rules can be as follows: the word ends in *льний* - cut off from the word *ьний*; the word ends with *льна* – intercept *ьна*; the word ends with *льне* – intercept *ьне*; the word ends with *льним* – intercept *ьним*.

Table 1. Fragment of the search table for the word *безпритульний* (homeless)

Ukrainian Word	Stemming	Analog of English Word	Stemming
безпритульна		*homeless*	
безпритульне		*homeless*	
безпритульний		*homeless*	
безпритульним		*homeless*	
безпритульними	*безпритул*	*homeless*	*homel*
безпритульних		*homeless*	
безпритульні		*homeless*	
безпритульній		*homeless*	
безпритульнім		*homeless*	

The cut off endings and suffixes. The number of such rules stemming is much less than with all word forms, but because the algorithm is quite compact and productive. The above 4 rules correctly work out the following adjectives:

Table 2. The result of the algorithm clipping endings and suffixes

Ukrainian Word	Stemming	Analog of English Word	Stemming
безпритульна	*безпритул*	*homeless*	*home*
повільне	*повіл*	*slowness*	*slow*
ортогональний	*ортогонал*	*orthogonally*	*orthogonal*
цивільним	*цивіл*	*civilly*	*civil*

However, the algorithm may make false conclusions and distort the shape stemming. For example, the word *пальне* will turn into *пал* instead of the correct form *пальн*. Therefore, given the peculiarities of the language, the set of rules for cutting off endings and suffixes can be quite complex. The disadvantages should also include the exception handling, when the base words have a variable shape. For example, the words *бігом* and *біжу* must-have after stemma the same біг, but a simple cut off of the end it is not possible to do. The algorithm is forced to take into account such situations-this leads to a complication of the rules, and in the end negatively affects the efficiency. To solve this problem in the created system we use a comprehensive approach based on the definition of the basis of the word by lemmatization. The first step of this algorithm is to determine the parts of speech in the speech, the so-called POS tagging. In the second step, by the word is way stemming according to the rules of the language. That is, the words "fuel" and "welcome" have to go through different chains of rules because "fuel" is a noun and "welcome" is an adjective. Theoretically, stemming algorithms based on lemmatization should have a very high quality and a minimum percentage of errors, but they are very dependent on the correctness of the recognition of parts of speechv[52-59].

One of the modern variants of realization of wordless morphology in pure form is Porter stemmer. In it, the string supplied to the input is checked for the presence of the specified Postfix, and the Postfix is checked in a certain order, and some of the Postfix can be combined. All that is left after a successive "drop" is declared a stem. Depending on the found Postfix, one or another part of speech can be attributed to the word, although in the vast majority of tasks this is not necessary. The algorithm is extremely simple, has a very high speed, but gives a large percentage of errors. In addition, the Postfix division is largely controversial. Also, the algorithm produces a single variant of parsing, completely hiding the homonymy of words. Porter's algorithm takes very little account of the fact that for different parts of speech, and even for different paradigms, different letters may precede the Postfix. This fact is used in the stemmer morphological analysis system, where not only the postfixes themselves are stored, but also the two previous letters of the pseudo base. The letter and Postfix combinations themselves are stored as a right-to-left finite state machine [5, 60-65]. A significant advantage of wordless morphologies is that they can give the result for any words that occur in the text, which is very convenient when analyzing texts with an unfamiliar subject area or containing many non-literary or rarely used words. However, the correctness of the results obtained is at the level of 90-95%. This has led to the abandonment of wordless morphologies in tasks where the accuracy of the analysis should prevail over its completeness, and to the transition to the use of dictionary morphologies in tasks such as machine translation and dialogue systems. However, in practice, there are a large number of problems solved by statistical methods, in which an approximate knowledge of the relationships between words is quite sufficient. This is the task of categorization, information retrieval, partly-the task of abstracting, a number of other tasks. The methods of wordless morphologies are actively used in dictionary morphologies to predict the normal form and set of parameters of words that are absent in the morphological dictionary, as well as to expand the dictionary [1]. Stemming variants for the Ukrainian language exist and are used as part of commercial search engines. Unfortunately, there is no free implementation of such algorithms. It should be noted that certain steps in this direction have already been made, in particular the Drupal module for the Ukrainian language, which is under development and the search engine “<META>", which uses a modified stemming method to cut off endings and suffixes of unknown words, so the emergence of a non-commercial stemming algorithm for the Ukrainian language is a matter of time [19].

Zipf's first law ("rank-frequency"). Another feature of the created system is the ability to determine the keywords of articles and compare them with the keywords specified by the author. This was achieved by using a method based on Zipf's law.

Let's measure the number of occurrences of each word in the text and take only one value from each group having the same frequency. Arrange the frequency as they decrease and number, the serial number of the frequency is called the frequency rank (denote r_i word rank i). The words that are most common will have rank 1, followed by 2 and so on. Then it is obvious that the probability of encountering an arbitrary, pre-selected word will be equal to the ratio of the number of occurrences of this word to the total number of words in the text (n_i is the number of occurrences of words and, $|N|$ is number of words in the text).

$$p = n_i / |N| . \tag{2}$$

CIPF found the following pattern: the product of the probability of finding a word in the text and the rank of frequency, is a constant number (C).

$$\frac{n_i \cdot r_i}{|N|} = C . \tag{3}$$

The law shows that the prevalence of a word in a text varies by hyperbole, depending on the number of occurrences. For example, the second of the words used is about half as common as the first, the third is three times less common than the first, and so on. The meaning of the constant varies in different languages, but within the same language group remains roughly the same, whatever text we take. George Zipf and other researchers found that the hyperbolic distribution is subject not only to all natural languages of the world, but also other phenomena of a social and biological nature: the distribution of scientists by the number of articles published by them, US cities by population, population by income in capitalist countries, and others. Zipf's laws allow finding key words [9]. Let us use Zipf's first law and plot the dependence of rank on frequency. Studies show that the most significant words for the text lie in the middle part of the graph (Fig. 2). This fact has a simple justification. Words that happen very often, mostly turn out to be official. It is also rare to find words that in most cases do not have a decisive meaning for the information that is submitted in the article. The quality of the selection of significant words depends on the width of the range. If you take a large width of the range, the keywords will get auxiliary words; if you set a narrow range - you can lose the semantic terms. Therefore, in each case it is necessary to use a number of heuristics to determine the width of the range, as well as techniques that reduce the impact of this width.

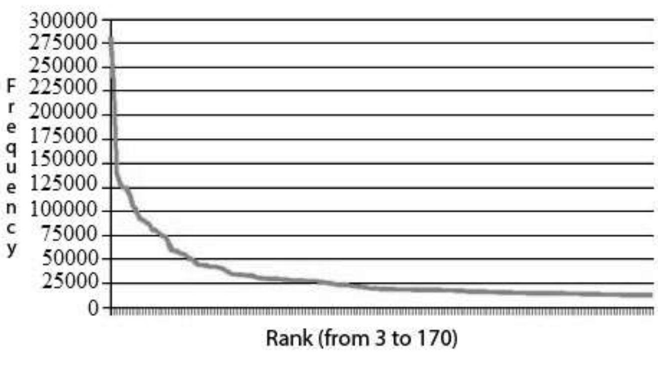

Fig. 1. Dependence of frequency of use of a word on its rank

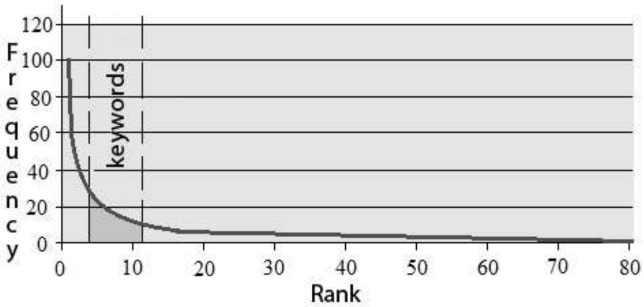

Fig. 2. The location of key words

One of the ways, for example, is the preliminary exclusion of words from the text under study, which initially can not be meaningful and, therefore, is "noise". Such words are called neutral or stop (stop words). For Ukrainian text stop words can be all prepositions, particles, personal pronouns. There are other ways to improve the accuracy of assessing the significance of words [9, 66-71]. Some words can occur in almost all documents of a certain collection and, accordingly, have little influence on the document belonging to a particular category, and therefore not be key to this document. Therefore, it is obvious that, considering the entire collection of documents, we will increase the informativeness of the selection of keywords [72-77].

The methods of constructing domain ontology's are as follows [29-57]:

- Building ontology's by converting XML-like documents;
- Using ready-made dictionaries;
- Application of linguistic analysis of texts written in natural language;
- Application of clustering and analysis of formal concepts.

Automatic extraction of knowledge from monological texts in order to build an ontology involves not only the identification of terms, but also the search in the text of knowledge about these terms. This means that in order to describe the semantic structure of terminology, it is necessary to recognize both the terms and the semantic relations between the terms in the text [78-83].

4 Highlighting Problems

Considering the possibility of automating various stages of automatic ontology generation [29-57], we came to the following conclusions. The preparatory stage and the serialization stage can be fully automated in all cases, since these processes are completely trivial and are reduced to primitive operations on string data or converting tree data structures into some XML-like format. The analysis stage can also be effectively automated. The construction of concepts, taxonomic relations between concepts and relations of belonging of instances to classes is automated using all the methods of ontology generation described above. The possibility of fully automating the construction of non-taxonomic relations is still an open question; in addition, the available methods are dependent on the language of text data that are processed. The question of full automation of concept generation is also open, especially it concerns the methods of automatic ontology generation based on the use of hierarchical clustering and formal analysis of concepts. This problem is proposed to be solved using a situational approach with partial use of the dictionary [42, 82-92].

The validation stage to some extent requires the intervention of an expert. An exception is when ontology is created based on a set of XML documents or a subset of entries in a dictionary or thesaurus. In this context, tools like WordNet undoubtedly deserve special attention because of the great possibilities for automating validation. Despite the fact that WordNet has too broad a purpose, and can not be adapted to a specific subject area of human activity, the use of this method for validation and harmonization of ontology's is an interesting topic for further development. The expansion stage also requires special attention when automating. This is especially true in the case of its reduction to the harmonization and merging of ontology's. The process of merging ontology's is closely related to reconciliation. Currently, there are a number of methods to implement it for two input ontology's, however, the simultaneous merging of several ontology's remains an open question. This problem can be solved by sequential merging, but in such a case the final ontology will depend on the choice of the merge order. In some cases, the addition of new entities and relationships to the ontology may occur in a method different from the one used at the initial stage [44].

5 Statement of Purpose

The purpose is to study problems of automation verify correct use of terms in the text, intellectual definitions of the key words of articles, development of a system for modeling processes of analysis and synthesis of texts of a technical nature, namely: check the correct use of terminology in articles according to the conventional rules and creation of a list of key words of articles, with the possibility of verifying the correctness of the list mentioned by the author keywords of the article. The created system is based on the algorithms of morphological analysis; it is based on a modified morphological analyzer, built on the principles of stemming and lemmatization. Therefore, the result will be the development of an intelligent system for modeling the processes of analysis and synthesis of technical text. Before designing an intelligent system for modeling the processes of analysis and synthesis of technical text, you must first build a tree of system goals, which will provide the opportunity to perform consistent and correct actions in the design of an intelligent system [4, 12, 14, 18, 21-23, 25-26, 28].

The main goal is to develop an intelligent system for modeling the processes of analysis and synthesis of technical text [4, 12, 14, 18, 22-23]. Achieving the main goal is possible only if all sub-goals are met. The main goal of the developed system is divided into four sub-goals.

The first sub-goal is "read text". The goal is to read the input data on which all subsequent operations will be performed. The second sub-goal is "parse text". This sub-goal is divided into two sub-goals: "find stop words" and "delete stop words". The goal is to "cleanse" the input text of "noise" (words that carry no semantic information). The third sub-goal is "perform analysis of the text and individual words"" this goal is divided into four sub-goals: "find the wrong term in the text" ""work out the text and keywords" "work out the term" and "perform morphological analysis of the term". Pid "Process text and keywords" is divided into four sub-goals: "Build an alphabetical-frequency dictionary", "Find keywords specified by the author", "Identify keywords and Verify the correctness of the selected keywords". Performing these four sub-goals is necessary to search for keywords specified by the author of the article, search for keywords in the text by analytical method and compare the results in order to verify the correctness of the keywords specified by the author.

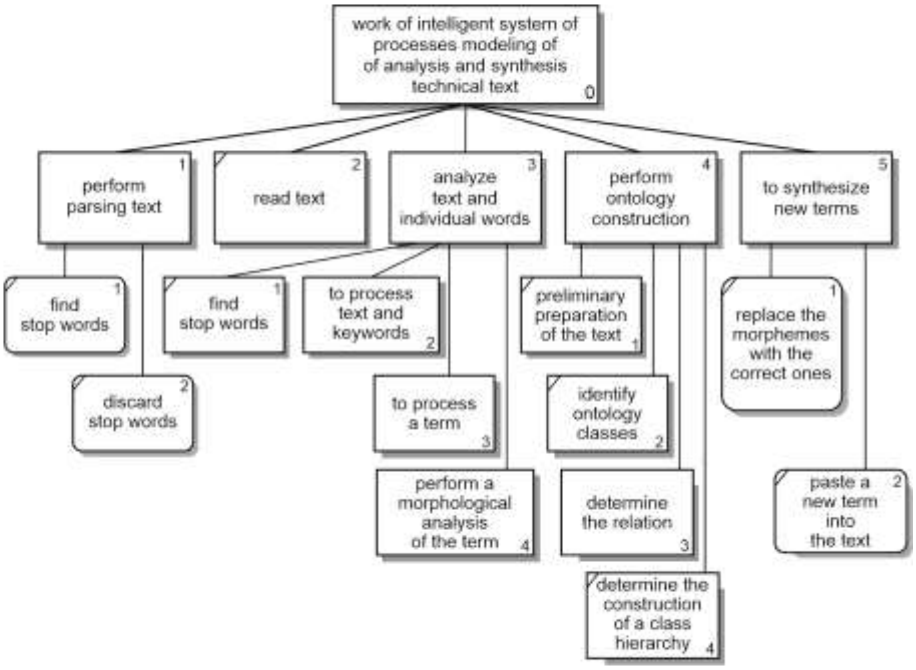

Fig. 3. A tree of system goals

The sub-goal "work out the term" is divided into two sub-goals: "break the term into parts of speech", and "find a replacement for the wrong term". Achieving these goals guarantees the preparation of the found term before morphological analysis. The "perform morphological analysis of the term" sub-goal is divided into six sub-goals: "determine morphemes", "determine temporal form (for verbs)", "determine person (for verbs)", " determine gender (for verbs and nouns)", "determine number (for verbs, nouns and adjectives)", "determine case (for nouns and adjectives)". Achieving these sub-goals ensures that the morphological analysis process, which is one of the key processes necessary for the functioning of the entire system, is carried out.

"Perform ontology construction" is divided into four sub-goals: "perform preliminary preparation of the text", "Define ontology classes", "Determine the relationship", "Perform the construction of the class hierarchy". Achieving these sub-goals ensures that the ontology building process is completed.

The sub-goal "perform synthesis of new terms" is divided into two sub-goals: "change morphemes to the correct ones", "insert a new term into the text". Achievement of these goals provides replacement of morphemes of term on new, according to the received characteristics of the text entered by the user and insertion back into the text of the "new" corrected correct term. Achieving the main goal is impossible without consistent implementation of each of the sub goals.

6 Analysis of the Obtained Scientific Results

The context diagram is the first in the hierarchy of diagrams of IDEF0 notation, it shows the functioning of the system as a whole (Fig. 4) [13, 15-16, 24]. This model is described from the user's point of view.

Fig. 4. Context chart A0

From the point of view of the user the functioning of this system is as follows:

- A text fragment is fed to the system input;
- At the output of the system, we get an edited text fragment in which the technical term is" correctly " used, a list of keywords calculated by the system, the result of checking the correctness of the keywords defined by the author of the article and an ontology;
- The system is guided by the rules of the current Ukrainian spelling edition of the Institute of linguistics. O. O. Potebnya of the national Academy of Sciences of Ukraine and Institute of Ukrainian. national Academy of Sciences of Ukraine 2007, which are approved by the Ministry of education and science of Ukraine;

- System resources are Administrator with rights to edit system configuration and various analysis rules, moderator with rights to edit databases, User and programming environment.

Fig. 5 shows the first step of decomposition. It is possible to examine in more detail the processes of the system, namely "Read the text", "perform text parsing", "analyze the text and individual words" and "perform synthesis of new terms". The "read text" process is the first of the processes running on this system. The input data for this process is the text entered by the user, namely: an article of a technical nature. This text is processed by the programming environment and passed for further processing by the "parse text" and "synthesize new terms" processes».

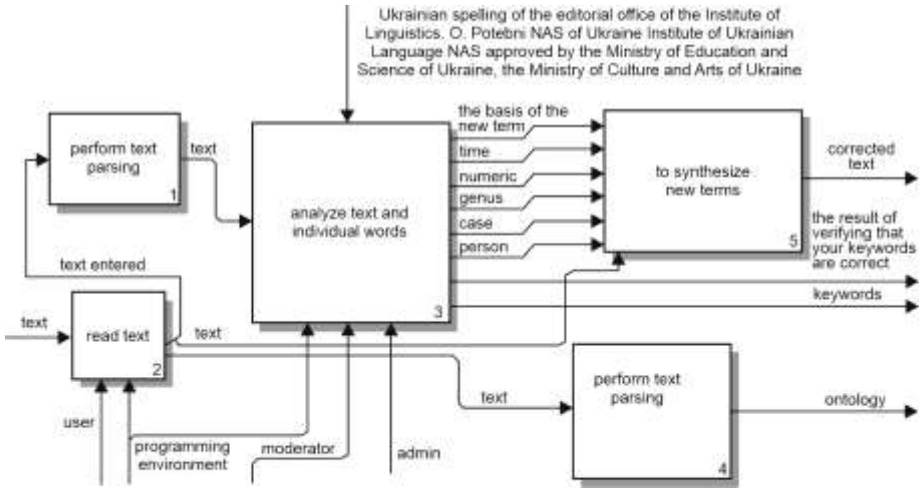

Fig. 5. IDEF0 as decomposition of the system

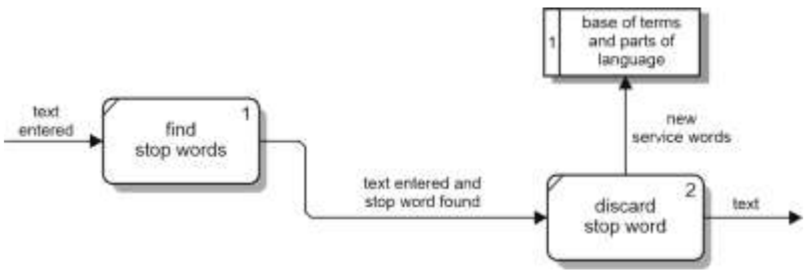

Fig. 6. The decomposition process "to parse the text»

Fig. 6 shows a data flow diagram. This diagram is the final decomposition step of The "parse text" process. It shows the processes of data exchange between the works "Find stop words", "Discard stop words" and the drive "Database of terms and parts of speech". At the input of the work "Find stop words" the text entered by the user is fed, over which stop words search operations are performed, then the found stop words are transferred to the work "Discard stop words", where they are removed from the text and added to the drive "Database of terms and parts of speech" provided that such words are not there. At the output of the work "Discard stop words", we get the text "cleared" of "noise", which is passed on to perform the following operations.

Fig.7. shows the decomposition of the IDEF0 process "Perform analysis of text and individual words". The input of the processes "find the wrong term in the text" and "work through the text and keywords" is submitted in the form of text that was worked out by the previous process "perform text parsing". Further - "Find the wrong term in the text", which is guided by the rules of Ukrainian spelling, search in the text of the wrong terms. At the output of this process, we obtain data in the form of a word or phrase (term), which is then transmitted for processing by the process "Work term".

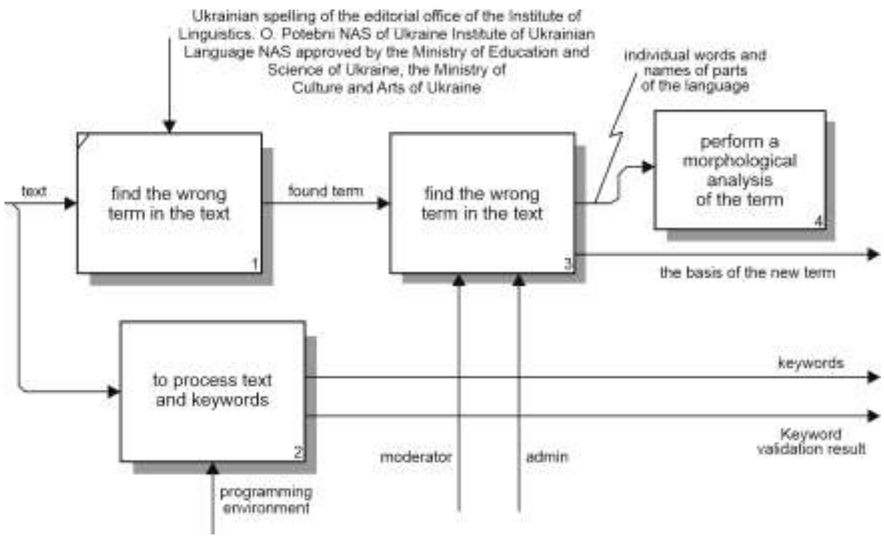

Fig. 7. Decomposition process "perform analysis of text and individual words"

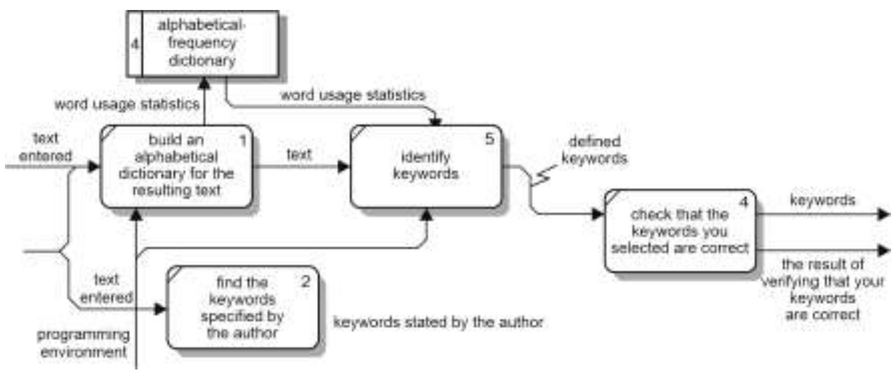

Fig. 8. The decomposition process "to Process the text and key words"

Fig. 8 shows a flow chart that is a decomposition of the process "Process text and keywords". This diagram consists of the processes: "Build an alphabetical-frequency dictionary for the resulting text", "Find keywords specified by the author", "Identify keywords", "Check the correctness of the selected keywords" and the drive "Alphabetical-frequency dictionary". At the input of the works "Build an alphabetical-frequency dictionary for the received text and Find the keywords specified by the author" the text worked out by the previous process is submitted. Work "Find keywords specified by the author" searches for keywords specified by the author according to the rules of registration of articles. At the output of this work we get data in the form of found keywords.

The process "Build an alphabetical-frequency dictionary for the received text" fills the drive "Alphabetical-frequency dictionary" with words and the frequency of their use in the text submitted for input. At the output of the work "Build an alphabetical-frequency dictionary for the received text" we get an unchanged source text, which is then transmitted for processing by the work "Define keywords". The process "Define keywords" receives the input text and on the basis of the existing in the drive "Alphabetical-frequency dictionary" statistical data performs the formation of a list of keywords for the input text. Further, these keywords and keywords obtained as a result of the work "Find keywords specified by the author" are transmitted for processing by the work "Check the correctness of the specified keywords". Process "to check the accuracy of the key words" performs a comparison specified by the author key words with key words found in the received text. At the output of the work "Check the correctness of the specified keywords" we get the result of the check and the list of keywords found by the system based on statistical data.

Fig. 9 shows a diagram of data flows, which is a decomposition of the process "Work time". This diagram consists of the following works: "Break the term into parts of speech", "Find a replacement for the" wrong "term" and accumulators "term analysis Rules", " Base of terms and parts of speech».

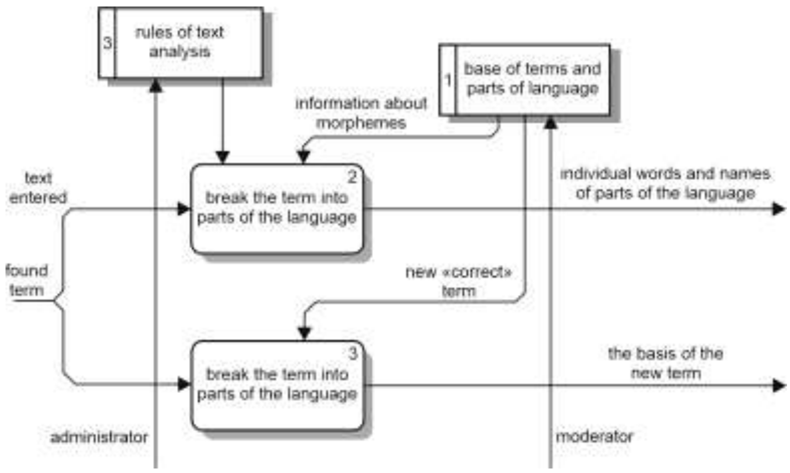

Fig. 9. Decomposition of the process "time to Work"»

At the input of the processes "Break the term into parts of speech" and "Find a replacement for the wrong term" data is submitted in the form of terms found as a result of the process "Find the wrong term in the text". The work "Break the term into parts of speech", based on the rules stored in the storage "Rules of analysis of the term", is divided if the term-a phrase, and the classification of individual words into parts of speech. Only the system Administrator has the rights to edit and add new rules. At the output of the work "break the term into parts of speech" we get a List of words classified by parts of speech, transferred to the next process for processing.

Process "Find a replacement for the" wrong "term" search "wrong" term in the text based on the data contained in the drive "Database of terms and parts of speech". The rights to change and add data to the "database of terms and parts of speech" are granted only to the Moderator of the system.

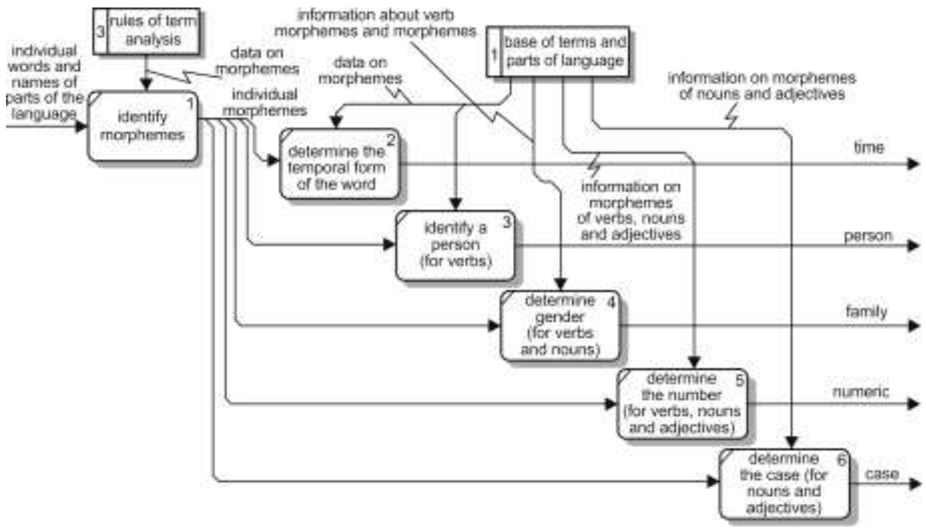

Fig. 10. The decomposition process "to perform a morphological analysis of the term"

Fig. 10 shows a diagram of data flows, which is a decomposition of the process "Perform morphological analysis of the term". This diagram consists of the following processes: "Identify morphemes", "Determine the temporal form of the word(for verbs)", "Determine personality (for verbs)", "Determine gender(for verbs and nouns)", "Determine the number(for verbs, nouns and adjectives)", "Determine the case(for nouns) and the accumulator "database of terms and parts of speech".

At the input of the procees "Define morphemes" data are submitted in the form of words classified by parts of speech. Next, based on the data from the drive "rules of analysis of the term", the selection of morphemes is performed. This was found after the morpheme is sent to study one or more of the works "to determine the temporal form of the word (for verbs), Define identity (for verbs)", "to determine the kind (for verbs and nouns)", "to Determine the number(for verbs, nouns and adjectives)", "to

Determine the case (for nouns)" to determine the current characteristics of the words term. Then the received data is transferred to the next process for processing.

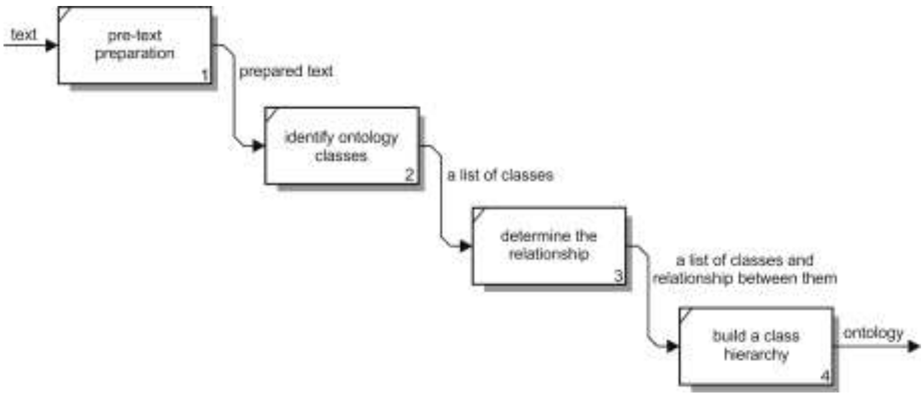

Fig. 11. Decomposition of the process "Perform ontology construction"

Fig.11 shows the IDEF0 decomposition of the "execute ontology build" process. The text of the article for preliminary study is submitted to the input of the process "Perform preliminary preparation of the text". The prepared text is passed to the "Define ontology classes" process to define the main classes. Next, a list of certain ontology classes is passed to the Define relationship process to build relationships. At the output of this process, we get a list of classes and relations between them; this data is passed to the process "Perform the construction of a class hierarchy to build a class hierarchy. Fig.12 shows a diagram of data flows, which is a decomposition of the process "Determine the relationship". This diagram consists of the works: "Define the relation "is-a" and "Define the relation" synonym-of".

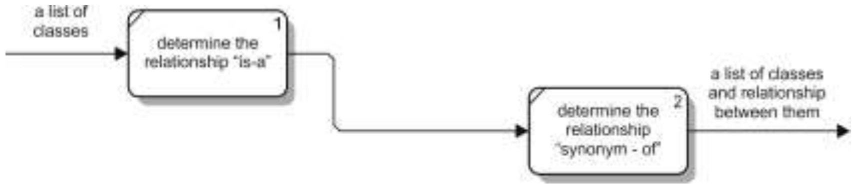

Fig. 12. Decomposition of the process "Determine the relation"

At the input of the work "Determine the relationship "is-a" data is submitted in the form of a list of classes formed in the previous step. Next, you define a relationship of type "is-a" using the appropriate production rule. At the output of the work "Determine the relation" is-a "list of classes and relations between them of type "is-a". The Define synonym-of relationship job defines a synonym-of relationship using the appropriate production rule. At the output of the work "Define the relation" synonym-of" a list of classes and relations between them of type "is-a" and "synonym-of".

Fig. 13 shows a diagram of data flows, which is a decomposition of the process "perform synthesis of new terms". This diagram consists of the works: "Replace morphemes with the correct ones", "Insert" a new "term into the text" and the "database of terms and parts of speech". At the input of the work "Change morphemes to correct" data are submitted in the form of a list of characteristics depending on the part of speech of the individual word and the basis of the word, which should change the wrong term. Further, according to these data, the search is performed and the corresponding morphemes are added to the bases of individual words of the new term from the storage "database of terms and parts of speech". The output of work "to Change the morpheme on the right" re-get "correct" term, which is then transmitted to the input of the work ", Insert "new" term in the text."

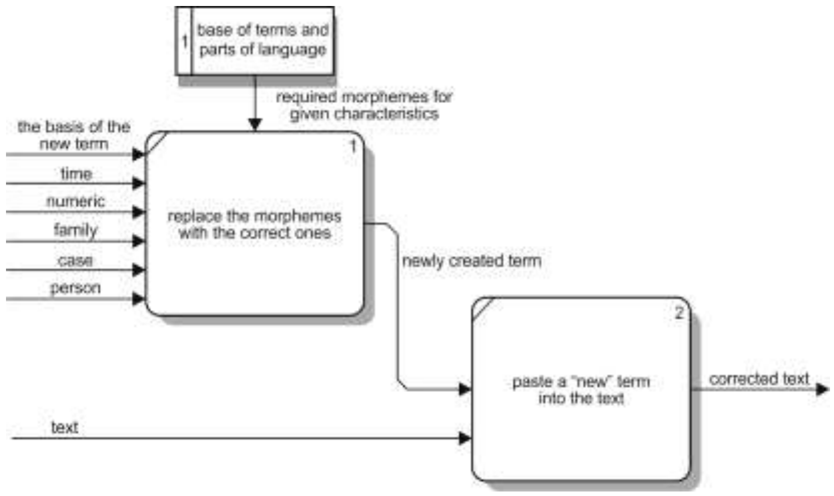

Fig. 13. Decomposition of the process "perform synthesis of new terms"

The insert new term in text job adds the newly created term to the text at the position of the "wrong" term. The output of the work "Insert" a new "term in the text" is one of the outputs of the whole system, namely, the edited text with the "correct" terms. A sentence in a natural language text is some statement. A situational approach is often used to recognize the semantics of statements [45]. Its use is based on the fact that in practice it is difficult to create a consistent and complete knowledge base. In situational semantics, conclusions are drawn only within the context of the situation at the moment. When you move to a different situation, the knowledge base is audited and the statements that were derived earlier are not used to derive new ones.

When implementing the situational approach, the idea of L. Wittgenstein is taken as a basis, with regard to the rules of the use of words: depending on the situation, the word is used one way or another. In fact, he formulated a pragmatic understanding of meaning (meaning is interpreted as an adequate response to emerging situations). If at the same time in the process of reasoning to use logical inference, then, setting syntactic restrictions in the form of a conjunction of facts, which further describe the situation, which is characterized by two semantic properties as consistency and minimalism, you can build a complete and consistent knowledge base. V. N. Vagin [16, 17] explains the idea of L. Wittgenstein as follows . If there are many (contexts) situations, then we have the following mapping F: $S \rightarrow CONT(T)$, where S is many situations, $CONT$ (T) is the set of statements forming the content of the idea, which is denoted by the term T. At the same time $y=F(s) \in CONT(T)$ is aspect of content. Thus, each situation s is associated with some element of the content of y, which corresponds to our intuition: when we use some content, we realize the incorporation of its content (sometimes we take everything from it, and sometimes only some part). Therefore, we seem to react to the situation with which we are dealing. We introduce the previous order relation $\leq$ (transitive and reflexive relation) on the plural $CONT(T)$. Then, for an idea that can be represented by T, some organization of knowledge is constructed $CONT(T)$, which creates the possibility of review or coverage (understanding) in the sense of C. I. Lewis. The development of this approach involves the study of possible situations s, which may be the components of the proposal. A possible situation is one or the other previous order of the components. Performing morphological or syntactic analysis, extracting knowledge about terms from terminological dictionaries or applying other methods of analysis of natural language text, situations are always investigated in which there are morphemes in lexemes, lexemes in sentences, sentences in the text, etc. Thus, methods of processing natural language text are almost always aimed at analyzing the situational context, and, depending on the method presented, the object of this analysis is a text or a fragment of text, or a sentence lexeme or a morpheme of a lexeme. According to the above, for the solution of various problems of study of the monological natural-speech text it is necessary to develop methods of their solution, based on situational modeling. Situational modeling is based on a simple nuclear construction of the language $s_{kk}=xRy$, where x, y is terms, R is semantic relationship between them. Let us now consider the structure of the production rule, which is usually described by the seven [83, 85]:

$$pr = < I, K, O, C, A \rightarrow B, H, E > \tag{4}$$

where I is the unique name of the product; K is the scope of the section or products; O is priority of production execution; C is product applicability condition, which is usually a logical expression; $A \rightarrow B$ is core products; H is aftereffects or postconditions of products, having the form of procedures performed in the event that the core of the product has been realized; E is links to other products.

The scope of the products is determined by the nature of the methods. For example, in the extraction of knowledge about the terminological system of the domain, the scope of application of K is the extraction of knowledge from terminological dictionaries. Product priority O is automatically set according to the length of the applicability condition, with the longest condition having the highest priority. The aftereffects of H and the relationship of E to other products are determined during kernel development. The main element of production is the core of the production rule in the form of "If A, then B". By antecedent A and consequent B we mean a number of facts. Structure s_{kk} corresponds to the structure of the fact. To verify this, consider the core of the production rule to identify the qualitative part-to-whole aggregation relation in the sentence "Advance is part of the total contract value". On some given subset of natural language it has this representation:

```
IF      <sentence>           p       has     <term>                t1
   AND  <sentence>           p       has     <term>                t2
   AND  <sentence>           p       has     <SemRelationship>     R
   AND  <SemRelationship>    r       has     <verb>                v
   AND  <SemRelationship>    r       has     <term-satelliteR>     tr
   AND  <verb>               v       has     <value>        ["is"]
   AND  <term-satelliteR>    tr      has     <value>        ["Part"]
   AND  <term>               t       has     <index>        i
   AND  <SemRelationship>    r       has     <index>        (i+1)
   AND  <sign>         h       has     <index>        (i+2)
   AND  <term>         t2      has     <index>        (i+3)
THEN    <term>         t1      has     <type>         ["Part"]
   AND  <<term>        t2      has     <type>         ["All"]
   AND  <SemRelationship> r  belongs to <category>  ["AllPart"].
```

This statement allows you to identify the fact that in the original sentence classified as "All" refers to the term "total contract price", the category "Part" the term "Advance", and that between them there exists a relation "All → Part." As can be seen from the example, the structure of the simple nuclear construction of each statement fact can be seen explicitly, for example, in the first fact " < sentence>*s* contains *<term>* t_1 » : *x* = *«* <sentence>*s s », R = «contains », y = « <Term>* t_1 *»*. Thus, we use the production model of knowledge representation as a method representation model [42]. This means that for each method it is necessary to develop a system of products, which most often has a hierarchical structure and is a declarative form of representation of the method. For example, a production system for recognizing semantic relations *PrSRR* consists of subsystems: recognition of semantic relations " whole → part »*PrSRR_WP*, « Genus-species» *PrSRR_CK* and etc.

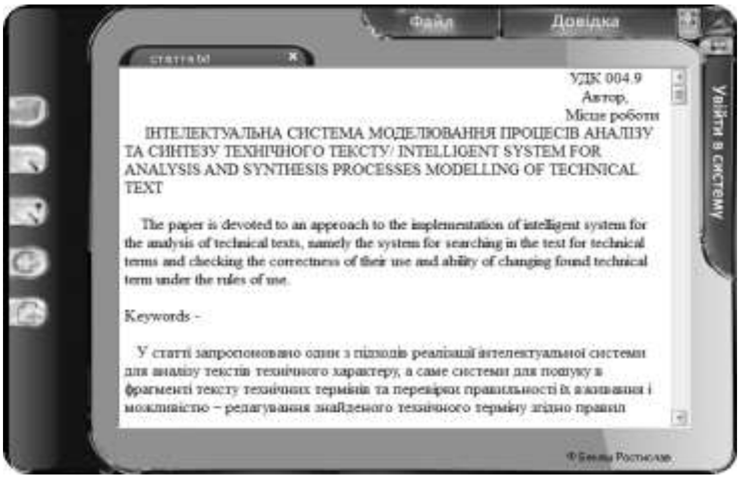

Fig. 14. Example of input data input

However, the creation of a system of products is quite a time-consuming task, it is often difficult for experts to formulate the rules that they use in solving problems, since expert knowledge in most cases is subconscious. It is the subconscious nature of expert knowledge that causes difficulties in the construction of expert systems, and the extraction of expert knowledge is considered a" bottleneck" of artificial intelligence [42]. Therefore, the system provides the ability to add production rules. The input data is a text file with the source text in the format"*.txt,*.doc,*.docx", (Fig.14) and words, phrases and their word forms (Fig. 15).

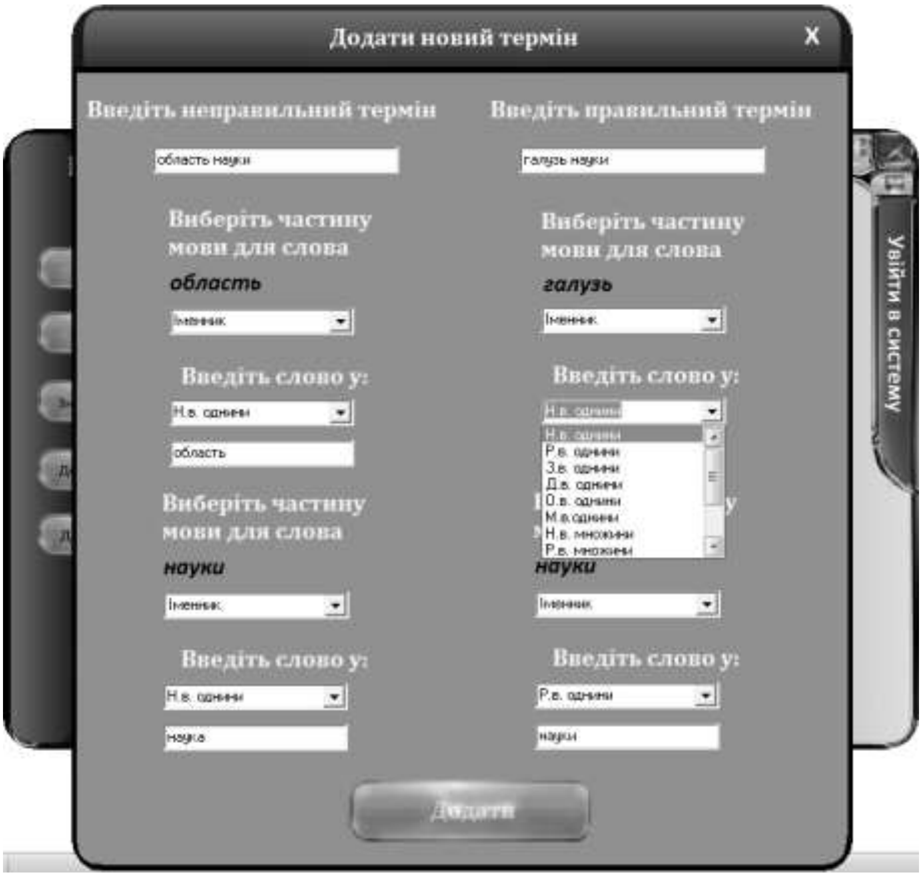

Fig. 15. Example of adding words to a database

The result of the program is: found the wrong term (Fig. 16), a suggestion to replace it with one of the correct terms from the list (list 1, Fig. 16), a list of keywords found, and an RDF file that describes the ontology.

Fig. 16. The result of finding the wrong term and options for choosing the right term

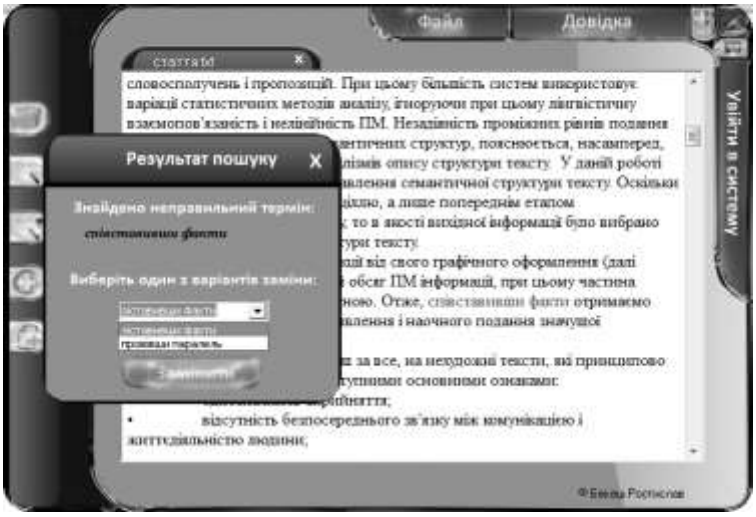

Fig. 17. The result of the process of finding the wrong terms and words

The program searches for incorrect terms and keywords in a text document.

1. Click "Open file".
2. Click the "Check text", "Check" or "Find the wrong term" button.
3. Select one of the options and click "Replace".
4. If necessary, click the "Find keywords" button to display a list of keywords.

Input is fragment of the article "Intelligent system of modeling the processes of analysis and synthesis of technical text".

7 The Work Result of the Program

Fig. 17 shows the system detected the wrong phrase "comparing the facts", also the user is asked to choose one of the options to replace the wrong phrase with the correct one. Fig. 18 is the result of the keyword search engine is shown. The author of the article specified key words: morphological analysis, morphemes, and terms. The system found key words: text, system, approach, mechanism. Therefore, the result: "the keywords Specified by the author are not correct".

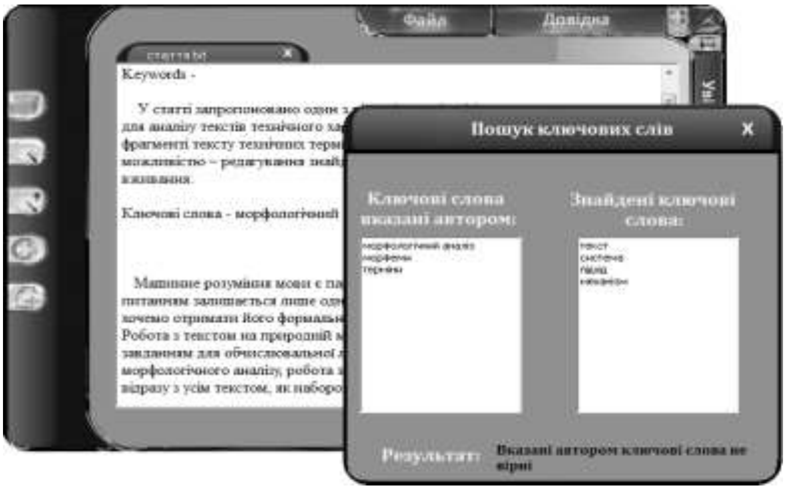

Fig. 18. Keyword search result

8 Conclusions and Prospects for Further Research

An intelligent system for modeling the processes of analysis and synthesis of technical text is developed. For the design and implementation of structural analysis system used by the environment AllFusion Process Modeler 7, which were created a functional diagram system, data flow diagrams, logical and sequence diagram works. All components and functional parts of the system are developed using visual programming environment C++ Builder 6.0, using SQL queries. Designing the database designed in the MySQL Workbench environment. The analysis of literature sources and researches in the field of automatic text processing and morphological analysis was carried out, the existing, similar in functionality, systems were considered. The system of modeling of processes of the analysis and synthesis of the technical text is developed. it searches for wrong words and phrases in texts of technical character, in particular articles. The system additionally provides the function of searching keywords in the text on the basis of Zipf's law. The found keywords can be used to construct a sex ontology in the form of XML documents, which is a very important fact because this format has become in fact the standard for data exchange between applications. To automate the validation stage, it would be more effective to use third-party dictionaries and thesauruses, which accordingly leads to the need to develop Ukrainian-language WordNet correspondences.

The result of the system is found the wrong word or phrase, and a list of words or phrases that can replace it, the system also displays a list of keywords specified by the author of the article, a list of keywords found by the system, and the result of checking their coincidence. Despite the functionality of the system, it is not without drawbacks. As all work of system depends on filling of its dictionary, first of all, it is necessary to carry out replenishment of base of words. Depriving the developed system of its shortcomings can be the first step in its further development. In particular, the work on the system should be continued in the direction of improving the algorithms for finding incorrect words and phrases, for example, the implementation of the search algorithm based on neural networks. To improve the efficiency of the keyword search algorithm, it is necessary to implement the ability to search for keywords not in one file, but in several files of similar subjects in order to discard words that are characteristic of texts of the same subject, but are not the key. The difference between the developed systems from the existing systems at the current stage of its development is the narrow specialization of the system on the texts of specific topics, in particular technical articles. So, the system of modeling the processes of analysis and synthesis of technical text is a simple tool for finding the wrong words and phrases, for building ontology and can be used for non-commercial purposes.

References

1. Learning Semantic Textual Similarity from Conversations, https://uk.wikipedia.org/wiki/, last accessed 2020/02/27.
2. Google AI blog, https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html, last accessed 2020/02/27.
3. Li, Y., Xu, L.: Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective. In: Int. J. Conf. on Artificial Intelligence. (2015).
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. arXiv:1310.4546 [cs.CL]. (2013).
5. Lebret, R., Collobert, R.: Word Emdeddings through Hellinger PCA. In: Conference of the European Chapter of the Association for Computational Linguistics (EACL). arXiv:1312.5542. Bibcode:2013arXiv1312.5542L. (2013).
6. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. In: NIPS. (2014).
7. Levy, O., Goldberg, Y.: Linguistic Regularities in Sparse and Explicit Word Representations. In: CoNLL, 171–180. (2014).
8. Globerson, A.: Euclidean Embedding of Co-occurrence Data. In: Journal of Machine Learning Research. (2007).
9. Zipfs law, http://kirsoft.com.ru/freedom/KSNews_394.htm, last accessed 2020/02/27.
10. Nlp Town “Comparing Sentence Similarity Methods”, http://nlp.town/blog/sentence-similarity, last accessed 2016/11/21.
11. ClearNLP Dependency Labels, https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md, last accessed 2016/11/21.
12. TEG-REP: A Corpus of Textual Entailment Graphs based on Relation Extraction Patterns, https://www.researchgate.net/publication/297759246_TEG-

REP_A_Corpus_of_Textual_Entailment_Graphs_based_on_Relation_Extraction_Patterns, last accessed 2016/11/21.
13. Lytvyn, V., Vysotska, V., Demchuk, A., Demkiv, I., Ukhanska, O., Hladun, V., Kovalchuk, R., Petruchenko, O., Dzyubyk, L., Sokulska, N.: Design of the architecture of an intelligent system for distributing commercial content in the internet space based on SEO-technologies, neural networks, and Machine Learning. In: Eastern-European Journal of Enterprise Technologies, 2(2-98), 15-34. (2019)
14. Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., Rishnyak, I.: The Contextual Search Method Based on Domain Thesaurus. In: Advances in Intelligent Systems and Computing, 689, 310-319. (2018)
15. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies. In: Int. Conf. CSIT, 124-128. (2015)
16. Vysotska, V., Chyrun, L., Chyrun, L.: Information Technology of Processing Information Resources in Electronic Content Commerce Systems. In: Computer Science and Information Technologies, CSIT'2016, 212-222. (2016)
17. Vysotska, V., Chyrun, L.: Methods of information resources processing in electronic content commerce systems. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM. (2015)
18. Naum, O., Chyrun, L., Kanishcheva, O., Vysotska, V.: Intellectual System Design for Content Formation. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 131-138. (2017)
19. Vysotska, V.: Linguistic Analysis of Textual Commercial Content for Information Resources Processing. In: Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET'2016, 709–713 (2016)
20. Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., Burov, Y.: Information resources processing using linguistic analysis of textual content. In: Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications, Romania, 573-578. (2017)
21. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 113-116 (2018)
22. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city. In: CEUR Workshop Proceedings, 239-255 (2018)
23. Qureshi, M. A., Greene, D.: EVE: explainable vector based embedding technique using Wikipedia. In: Journal of Intelligent Information Systems. arXiv:1702.06891. (2018).
24. Towson University "Sentence patterns", https://webapps.towson.edu/ows/sentpatt.htm, last accessed 2016/11/21.
25. Stanford Natural Language Processing Group "Neural Network Dependency Parser", https:// nlp.stanford.edu/software/nndep.html, last accessed 2016/11/21.
26. Edit Distance and Postediting, https://www.gala-global.org/blog/edit-distance-and-postediting, last accessed 2016/11/21.
27. Welty C. Towards a Semantics for the Web. Padova, Italy. (2006)
28. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216 (2018)
29. Lytvyn, V., Vysotska, V., Pukach, P., Vovk, M., Ugryn, D.: Method of functioning of intelligent agents, designed to solve action planning problems based on ontological approach. In: Eastern-European Journal of Enterprise Technologies, 3/2(87), 11-17 (2017)

30. Vysotska, V., Lytvyn, V., Burov, Y., Berezin, P., Emmerich, M., Fernandes, V. B.: Development of Information System for Textual Content Categorizing Based on Ontology. In: CEUR Workshop Proceedings, Vol-2362, 53-70. (2019)
31. Lytvyn, V., Vysotska, V., Rusyn, B., Pohreliuk, L., Berezin, P., Naum O.: Textual Content Categorizing Technology Development Based on Ontology. In: CEUR Workshop Proceedings, Vol-2386, 234-254. (2019)
32. Staab S., Studer, R.: Handbook on Ontologies. Springer, Verlag. (2004)
33. Sachenko, S., Rippa, S., Krupka, Y.: Pre-Conditions of Ontological Approaches Application for Knowledge Management in Accounting. In: IEEE International Workshop on Antelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Rende (Cozenza), Italy, 605-608. (2009)
34. Lytvyn, V., Vysotska, V., Dosyn, D., Burov, Y.: Method for ontology content and structure optimization, provided by a weighted conceptual graph. In: Webology, 15(2), 66-85 (2018)
35. Lytvyn, V., Vysotska, V., Dosyn, D., Lozynska, O., Oborska, O.: Methods of Building Intelligent Decision Support Systems Based on Adaptive Ontology. In: International Conference on Data Stream Mining and Processing, DSMP, 145-150. (2018)
36. Yildiz, B., Miksch, S.: Ontology-Driven Information Systems: Challenges and Requirements. In: International Conference on Semantic Web and Digital Libraries. (2007)
37. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. In: Scientific American. (2001)
38. Ruch, P., Gobeil, J., Lovis, C., Geissbtihler, A.: Automatic medical encoding with SNOMED categories. In: BMC Medical Informatics and Decision Making. (2001)
39. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: Classification methods of text documents using ontology based approach. In: Advances in Intelligent Systems and Computing, 512, 229-240. (2017).
40. Burov, Y., Vysotska, V., Kravets, P.: Ontological approach to plot analysis and modeling. In: CEUR Workshop Proceedings, Vol-2362, 22-31. (2019)
41. Lytvyn, V., Vysotska, V., Burov, Y., Demchuk, A.: Architectural ontology designed for intellectual analysis of e-tourism resources. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 335-338 (2018)
42. Haav, H.M.: An Application of Inductive Concept Analysis to Construction of Domain-specific Ontologies. In: Workshop of VLDB2003, 63-67. (2003)
43. Maedche, A., Staab, S.: Discovering Conceptual Relations from Text. In: European Conference on Artificial Intelligence, ECAI, IOS Press, Amsterdam. (2000)
44. Lytvyn, V., Burov, Y., Kravets, P., Vysotska, V., Demchuk, A., Berko, A., Ryshkovets, Y., Shcherbak, S., Naum, O.: Methods and Models of Intellectual Processing of Texts for Building Ontologies of Software for Medical Terms Identification in Content Classification. In: CEUR Workshop Proceedings, Vol-2362, 354-368. (2019)
45. Kravets, P., Burov, Y., Lytvyn, V., Vysotska, V.: Gaming method of ontology clusterization. In: Webology, 16(1), 55-76. (2019)
46. Shu, C., Dosyn, D., Lytvyn, V., Vysotska V., Sachenko, A., Jun, S.: Building of the Predicate Recognition System for the NLP Ontology Learning Module. In: International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS, 2, 802-808. (2019)
47. Alieksieieva, K., Berko, A., Vysotska, V.: Technology of commercial web-resource processing. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)

48. Lytvyn, V., Vysotska, V., Uhryn, D., Hrendus, M., Naum, O.: Analysis of statistical methods for stable combinations determination of keywords identification. In: Eastern-European Journal of Enterprise Technologies, 2/2(92), 23-37. (2018)
49. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 120-123. (2015)
50. Kalyanpur, A.: OWL: Capturing Semantic Information using a Standardized Web Ontology Language. In: Multilingual Computing & Technology Magazine, 15(7). (2004)
51. Gruninger, M., Obrst, L.: An Ontology Framework. In: Ontology Summit NIST, Gaithersburg, MD April 22-23. (2007)
52. Khomytska, I., Teslyuk, V., Holovatyy, A., Morushko, O.: Development of methods, models, and means for the author attribution of a text. In: Eastern-European Journal of Enterprise Technologies, 3(2-93), 41–46. (2018)
53. Khomytska, I., Teslyuk, V.: Authorship and Style Attribution by Statistical Methods of Style Differentiation on the Phonological Level. In: Advances in Intelligent Systems and Computing III. AISC 871, Springer, 105–118. (2019)
54. Korobchinsky, M., Vysotska, V., Chyrun, L., Chyrun, L.: Peculiarities of Content Forming and Analysis in Internet Newspaper Covering Music News, In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 52-57. (2017)
55. Lassila, O., Swick, R.: Resource Description Framework: Model and Syntax Specification. In: W3C Recommendation. (1999)
56. Klyne, G., Carroll, J.: Resource Description Framework : Concepts and Abstract Data Model. In: W3C Working Draft. (2002)
57. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. In: W3C Working Draft. (2002)
58. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136. (2018)
59. Gozhyj, A., Kalinina, I., Vysotska, V., Gozhyj, V.: The method of web-resources management under conditions of uncertainty based on fuzzy logic. In: 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT. 343-346. (2018)
60. Gozhyj, A., Vysotska, V., Yevseyeva, I., Kalinina, I., Gozhyj, V.: Web Resources Management Method Based on Intelligent Technologies. In: Advances in Intelligent Systems and Computing, 871, 206-221. (2019)
61. Vasyl, Lytvyn, Victoria, Vysotska, Dmytro, Dosyn, Roman, Holoschuk, Zoriana, Rybchak: Application of Sentence Parsing for Determining Keywords in Ukrainian Texts. In: Computer Science and Information Technologies, CSIT, 326-331. (2017)
62. Chyrun, L., Vysotska, V., Kis, I., Chyrun, L.: Content Analysis Method for Cut Formation of Human Psychological State. In: International Conference on Data Stream Mining and Processing, DSMP, 139-144. (2018)
63. Chyrun, L., Kis, I., Vysotska, V., Chyrun, L.: Content monitoring method for cut formation of person psychological state in social scoring. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, 106-112. (2018)
64. Lytvyn, V., Pukach, P., Bobyk, I., Vysotska, V.: The method of formation of the status of personality understanding based on the content analysis. In: Eastern-European Journal of Enterprise Technologies, 5/2(83), 4-12. (2016)

65. Lytvyn V., Vysotska V., Pukach P., Nytrebych Z., Demkiv I., Kovalchuk R., Huzyk N.: Development of the linguometric method for automatic identification of the author of text content based on statistical analysis of language diversity coefficients. In: Eastern-European Journal of Enterprise Technologies, 5(2), 16-28. (2018)
66. Lytvyn, V., Vysotska, V., Rzheuskyi, A.: Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. In: CEUR Workshop Proceedings, 2392, 147-171. (2019)
67. Vysotska, V., Kanishcheva, O., Hlavcheva, Y.: Authorship Identification of the Scientific Text in Ukrainian with Using the Lingvometry Methods. In: 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 34-38. (2018)
68. Demchuk, A., Lytvyn, V., Vysotska, V., Dilai, M.: Methods and Means of Web Content Personalization for Commercial Information Products Distribution. In: Advances in Intelligent Systems and Computing, 1020, 332–347. (2020)
69. Vysotska, V., Lytvyn, V., Hrendus, M., Kubinska, S., Brodyak, O.: Method of textual information authorship analysis based on stylometry. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 9-16. (2018)
70. Vysotska, V., Burov, Y., Lytvyn, V., Oleshek, O.: Automated Monitoring of Changes in Web Resources. In: Advances in Intelligent Systems and Computing, 1020, 348–363. (2020)
71. Lytvyn, V., Vysotska, V., Pukach, P., Nytrebych, Z., Demkiv, I., Senyk, A., Malanchuk, O., Sachenko, S., Kovalchuk, R., Huzyk, N.: Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian. In: Eastern-European Journal of Enterprise Technologies, 6(2-96), 19-31. (2018)
72. Vysotska, V., Fernandes, V.B., Lytvyn, V., Emmerich, M., Hrendus, M.: Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship. In: Advances in Intelligent Systems and Computing, 871, 132-151. (2019)
73. Vysotska, V., Burov, Y., Lytvyn, V., Demchuk, A.: Defining Author's Style for Plagiarism Detection in Academic Environment. In: International Conference on Data Stream Mining and Processing, DSMP, 128-133. (2018)
74. Lytvyn, V., Vysotska, V., Burov, Y., Bobyk, I., Ohirko, O.: The linguometric approach for co-authoring author's style definition. In: International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS, 29-34. (2018)
75. Lytvyn, V., Sharonova, N., Hamon, T., Cherednichenko, O., Grabar, N., Kowalska-Styczen, A., Vysotska, V.: Preface: Computational Linguistics and Intelligent Systems (COLINS-2019). In: CEUR Workshop Proceedings, Vol-2362. (2019)
76. Andrunyk, V., Chyrun, L., Vysotska, V.: Electronic content commerce system development. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2015-February. (2015)
77. Lytvyn V., Vysotska V., Peleshchak I., Basyuk T., Kovalchuk V., Kubinska S., Chyrun L., Rusyn B., Pohreliuk L., Salo T.: Identifying Textual Content Based on Thematic Analysis of Similar Texts in Big Data. In: International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT), 84-91. (2019)
78. Vysotska V., Lytvyn V., Kovalchuk V., Kubinska S., Dilai M., Rusyn B., Pohreliuk L., Chyrun L., Chyrun S., Brodyak O.: Method of Similar Textual Content Selection Based on Thematic Information Retrieval. In: International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT), 1-6. (2019)

79. Antonyuk, N., Medykovskyy, M., Chyrun, L., Dverii, M., Oborska, O., Krylyshyn, M., Vysotsky, A., Tsiura, N., Naum, O.: Online Tourism System Development for Searching and Planning Trips with User's Requirements. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, 1080, 831-863. (2020)
80. Lozynska, O., Savchuk, V., Pasichnyk, V.: Individual Sign Translator Component of Tourist Information System. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, Springer, Cham, 1080, 593-601. (2020)
81. Rzheuskyi, A., Kutyuk, O., Voloshyn, O., Kowalska-Styczen, A., Voloshyn, V., Chyrun, L., Chyrun, S., Peleshko, D., Rak, T.: The Intellectual System Development of Distant Competencies Analyzing for IT Recruitment. In: Advances in Intelligent Systems and Computing IV, Springer, Cham, 1080, 696-720. (2020)
82. Rusyn, B., Pohreliuk, L., Rzheuskyi, A., Kubik, R., Ryshkovets Y., Chyrun, L., Chyrun, S., Vysotskyi, A., Fernandes, V. B.: The Mobile Application Development Based on Online Music Library for Socializing in the World of Bard Songs and Scouts' Bonfires. In: Advances in Intelligent Systems and Computing IV, Springer, 1080, 734-756. (2020)
83. Antonyuk N., Chyrun L., Andrunyk V., Vasevych A., Chyrun S., Gozhyj A., Kalinina I., Borzov Y.: Medical News Aggregation and Ranking of Taking into Account the User Needs. In: CEUR Workshop Proceedings, Vol-2362, 369-382. (2019)
84. Kis, Y., Chyrun, L., Tsymbaliak, T., Chyrun, L.: Development of System for Managers Relationship Management with Customers. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 405-421. (2020)
85. Chyrun, L., Kowalska-Styczen, A., Burov, Y., Berko, A., Vasevych, A., Pelekh, I., Ryshkovets, Y.: Heterogeneous Data with Agreed Content Aggregation System Development. In: CEUR Workshop Proceedings, Vol-2386, 35-54. (2019)
86. Chyrun, L., Burov, Y., Rusyn, B., Pohreliuk, L., Oleshek, O., Gozhyj, ., Bobyk, I.: Web Resource Changes Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2386, 255-273. (2019)
87. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O.: Uniform Method of Operative Content Management in Web Systems. In: CEUR Workshop Proceedings, Vol-2136, 62-77. (2018)
88. Chyrun, L., Gozhyj, A., Yevseyeva, I., Dosyn, D., Tyhonov, V., Zakharchuk, M.: Web Content Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2362, 126-142. (2019)
89. Bisikalo, O., Ivanov, Y., Sholota, V.: Modeling the Phenomenological Concepts for Figurative Processing of Natural-Language Constructions. In: CEUR Workshop Proceedings, Vol-2362, 1-11. (2019)
90. Kulchytskyi, I.: Statistical Analysis of the Short Stories by Roman Ivanychuk. In: CEUR Workshop Proceedings, Vol-2362, 312-321. (2019)
91. Shandruk, U.: Quantitative Characteristics of Key Words in Texts of Scientific Genre (on the Material of the Ukrainian Scientific Journal). In: CEUR Workshop Proceedings, Vol-2362, 163-172. (2019)
92. Levchenko, O., Romanyshyn, N., Dosyn, D.: Method of Automated Identification of Metaphoric Meaning in Adjective + Noun Word Combinations (Based on the Ukrainian Language). In: CEUR Workshop Proceedings, Vol-2386, 370-380. (2019)