Conceptual Model of Process Formation for the Semantics of Sentence in Natural Language

Oleg Bisikalo[0000-0002-7607-1943]1, Victoria Vysotska[0000-0001-6417-3689]2, Yevhen Burov[0000-0001-8653-1520]3, Petro Kravets[0000-0001-8569-423X]4

[1]Vinnytsia National Technical University, Vinnytsia, Ukraine
[2-4]Lviv Polytechnic National University, Lviv, Ukraine

obisikalo@gmail.com[1], Victoria.A.Vysotska@lpnu.ua[2], Yevhen.V.Burov@lpnu.ua[3], petro.o.kravets@lpnu.ua[4]

Abstract. The article explains the use of generative grammars in linguistic modeling. Describes syntax modeling of sentence that used to automate the analysis and synthesis of natural-language texts. The article proposes content analysis methods for an online-newspaper.

Keywords. Content, Generative Grammars, Sentence Structure, NLP, Information Resource, Content Analysis, Computer Linguistic System, Content Management System

1 Introduction

Generative grammar theory is an effective tool for syntactic level linguistic modeling. The beginning of this theory is laid in the works of linguist N. Chomsky, where formal analysis of the grammatical structure of phrases is used to distinguish the syntactic structure (constituents) as the basic scheme of a phrase, regardless of its meaning [38-41, 49-57]. A. Gladkiy applied the concepts of dependency trees and component systems to simulate syntactic language level [14-16]. He suggested a way to moderate syntax using syntactic groups that distinguish word components as units of constructing a dependency - a representation that combines the benefits of the method of direct constituents and dependency trees.

2 Analysis of Recent Research and Publications

Active development of the Internet contributes to the creation of various linguistic resources. The need for the implementation of the processes of analysis and synthesis of natural-language texts led to the emergence of appropriate linguistic models of processes of their processing [2-5, 8-9, 12-19, 21-24, 26-29, 32-33, 37-41, 43, 46, 48-59, 61-63, 68]. The need of arises in the development of many linguistic disciplines for the needs of information sciences. Integration processes in most areas of the modern world attract particular attention to the development and creation of automated multilingual information processing systems. Formal generative grammar G is a quadruple $G = (V, T, S, P)$, where V is a finite non-empty set, the *alphabet (dictionary)*; T its subset whose elements are *terminal (main)* symbols, *terminals*; S is the *initial symbol* ($S \in V$); P is a finite set of *productions (transformation rules)* of the form $\xi \rightarrow \eta$, where ξ and η, are chains above V. The set is denoted by N, its elements are non-terminal (auxiliary) symbols, not terminals [14-16]. We will interpret terminal symbols as word forms (of some natural language), non-terminal symbols as syntactic categories, and terminal derived strings as correct sentences of this language [12-16, 38-41, 49-57]. Then the derivation of a sentence is naturally interpreted as its syntactic structure, which is presented in terms of direct constituents, that is, by a method long known in linguistics [10, 44, 46-47]. Research and research by N. Chomsky and A. Gladkiy developed and continued by A. Anisimov [2-3], Y. Apresyan [4-5], N. Bilgaeva [8], E. Bolshakova, E. Klishinsky, D. Lande, A. Noskov, O. Peskov and E. Yagunova [9], I. Volkova and T. Rudenko [11], O. Gakman [12], A. Gerasimov [13], M. Gross and A. Lanten [17], N. Darchuk [18], I. Demeshko [19], V. Inve [21, 65-66], T. Lyubchenko [22], B. Martynenko [23], O. Marchenko [24], E. Paducheva [26], Z. Partiko [27] , A. Pentus and M. Pentus [28], E. Popov [29], G. Potapova [32], N. Rusachenko [33], V. Fomichev [37], S. Sharov [43], Y. Shcherbina [46], Y. Schrader [48], Y. Bar-Hillel and E. Shamir [58], D. Bobrov [59], D. Hays [61], P. Postal [62], L. Tecniere [63], D. Varga [68]. These studies are used to develop tools for NLP, such as text annotation, machine translation systems, information retrieval systems, morphological, educational didactic systems, syntactic and semantic text analysis, linguistic support for specialized software systems, etc. [18-27, 43, 67-89].

3 Formation of the Purpose

Within the article, we will show how to use the generative grammar apparatus to simulate sentence syntax for different natural languages, such as English, German, Russian, and Ukrainian. To do this, let's analyze the syntactic structure of sentences, demonstrate the features of the process of synthesis of sentences of these languages. Consider the influence of the rules and rules of language on the course of constructing grammars [10, 44, 46-47].

4 Analysis of Achieved Scientific Results

Generative grammar G - is a quadruple G = (V, T, S, P), where V - is a finite non-empty set, the *alphabet (dictionary)*; T - its subset whose elements are *terminal (basic)* lexical units, *terminals*; S - is the *initial symbol* ($S \in V$); P - is a finite set of *productions (transformation rules*) of the form $\xi \rightarrow \eta$, where ξ and η, are chains above V. The set is denoted by N, its elements are non-terminal (auxiliary) lexical units, not terminals [14-16]. Grammars are classified by types of products subject to certain restrictions (Table 1) [14-16, 38-41, 49-57]. V vocabulary consists of a finite non-empty set of lexical units [60]. The expression over V is a finite-length chain of lexical units of V. An empty chain that does not contain lexical units is denoted by. Denote the set of all lexical units over V is V^*. Language over V is a subset V^*. The language is given through the set of all lexical units of the language or through the definition of the criterion that lexical units must satisfy in order to belong to the language [14-16, 38-41, 49-57]. Another important way to define a language is through the use of generative grammar. Grammar consists of a set of lexical units of different types and a set of rules or products of expression construction. Grammar has a V dictionary, which is the set of lexical items for constructing language expressions. Some vocabulary (terminal) vocabulary units cannot be replaced by other vocabulary units.

Table 1. Classification of grammars by product type

Grammar	Type	Description
G_0	Unbounded	Here ξ – is an arbitrary chain containing at least one non-terminal sysmbol, η – is an arbitrary chain over V.
G_1	Context-dependent	In many products P is a product of the form $\gamma\xi\delta \rightarrow \gamma\eta\delta$, $\lvert \xi \rvert \leq \lvert \eta \rvert$ (but not in the form of $\xi \rightarrow \eta$), then ξ can be replaced V only in the environment of chains $\gamma \ldots \delta$, i.e. in the relevant context.
G_2	Context-free	The non-terminal A on the left-hand side of the product $A \rightarrow /$ can be replaced by a chain $/$ in an arbitrary environment whenever it occurs, i.e. regardless of context.
G_3	Regular	Only products $A \rightarrow aB$, $A \rightarrow a$, $S \rightarrow \lambda$, may be regular, where A, B – non-terminals, a – terminal, λ – is an empty chain.

A distinctive feature of context-free grammars G_2 - is that at each step of the output is processed *exactly one character*, that is, in no way can be taken into account the presence / absence or properties of different adjacent characters. This may give the impression that grammars G_2 are not suitable for describing natural languages: in ordinary grammars, statements about the choice of forms or variations, or variations, of these or other elements of the expression are usually formulated with contextual conditions in mind. Thus, when describing inflectional forms, they indicate flexion, which must be selected depending on the type of substrate (acting as context); in the description of the use of Ukrainian distinctions indicate that the initial case of the direct supplement is replaced by a generic one in the presence of objection, etc.; oud subject is possible with the verb noun only if the noun is a supplement in the genitive case (*content viewing by the user, but not *viewing by the user*), etc. [7, 10, 18-20, 32-33, 36, 45]. Grammar G_3 is practically capable of generating the vast majority of simple and complex natural language sentences. Therefore, this statement is also true for arbitrary grammars. In almost all cases where the use of context is at first sight inevitable, in fact it can be dispensed with. For example, let there be a class of elements, and in the neighborhood of the elements of a class Y the elements X behave differently than in the neighborhood of the elements of a class Z by the following rules:

$YX \rightarrow YAB$ and $ZX \rightarrow ZCD$ (these rules use context).

We denote by X_1 the element X in the position after Y , and by X_2 - the element X in the position after Z . Then you can go to rules that do not use context:

$$X_1 \to AB \text{ and } X_2 \to CD .$$

That is, more elemental categories are introduced that take into account their position in context. Let us show how the transition to grammar G_2 formulations can be done in the context-based examples above. On the left is the required snippet of the corresponding context-dependent grammar, on the right is an equivalent snippet consisting of context-free rules.

1. Choice of flexion of the case depending on the type of base [7, 10, 18-20, 32-33, 36, 45]:

$Word_{fl.gn} \to O^i F_{fl.gn}$	$Word_{fl.gn} \to O^i F^i_{fl.gn}$
$O^1 F_{fl.gn} \to O^i ds(frien-ds)$	$F^1_{fl.gn} \to ds$
$O^2 F_{fl.gn} \to O^i ys(to-ys)$	$F^2_{fl.gn} \to ys$
$O^3 F_{fl.gn} \to O^i ens(childr-ens)$	$F^3_{fl.gn} \to ens$
$O^4 F_{fl.gn} \to O^i ves(relati-ves)$	$F^4_{fl.gn} \to ves$
$O^5 F_{fl.gn} \to O^i (cars-)$	$F^5_{fl.gn} \to \Lambda$

…………………………………………………………………………………

where $Word$ is wordform, O^i is basis of type i $(i = 1,2,3,...)$, $F_{fl.gn}$ is flexion of the genus. the case of pluralism.

2. The choice of a direct complement, depending on the objection:

$\tilde{R} \to R^i \tilde{N}_d$	$\tilde{R} \to R^i \tilde{N}^1_d$
$\tilde{R} \to \neg R^i \tilde{N}_d$	$\tilde{R} \to \neg R^i \tilde{N}^2_d$
$X R \tilde{N}_d \underset{X \neq \neg x}{\to} X R^i \tilde{N}_з$	$\tilde{N}^1_d \to \tilde{N}_з$ (student studies discipline)
$X \neg R \tilde{N}_d \underset{X \neq \neg x}{\to} X \neg R^i \tilde{N}_p$	$\tilde{N}^2_d \to \tilde{N}_p$ (student does not study discipline)

where $\tilde{R}$ is group of verbs, R^i is transitive verb, $\tilde{N}_d$ is direct complement, $\tilde{N}$ is group of noun, $\neg$ is negation [7, 10, 18-20, 32-33, 36, 45].

3. Opportunity of a subject in case of a verbal noun, depending on the presence of the object *(user viewing content):*

$$\tilde{N} \to \tilde{N}'\tilde{N}^{ob}\tilde{N}^{sb} \qquad \tilde{N} \to \tilde{N}'\tilde{N}^{ob}\tilde{N}^{sb^1} \qquad \tilde{N} \to \tilde{N}'\tilde{N}^{sb}$$

$$\tilde{N} \to \tilde{N}'\tilde{N}^{sb^2} \qquad \tilde{N}^{ob}\tilde{N}^{sb} \to \tilde{N}^{ob}\tilde{N}_o \qquad \tilde{N}^{sb^1} \to \tilde{N}'_o$$

where $\tilde{N}^{ob}$ is the object, $\tilde{N}^{sb}$ is the subject [7, 10, 18-20, 32-33, 36, 45]. In these examples, exactly the same formal technique is used: context information is encoded in new categories. So the fewer contexts you use, the more categories you need to enter, and vice versa. The appeal of switching to context-free rules is that it is difficult to evaluate the degree of complexity of different and meaningful contextual references, whereas in grammar rules, measuring the degree of complexity is the number of categories used.

The context cannot be abandoned, that is, it is impossible to dispense with one character to the left of the rule if the rule is to allow a character shift. Consequently, grammar G_2 cannot produce language that contains strings that cannot be constructed without permutations. Consider, for example, a language that contains all sorts of chains, $a_1a_2a_3qa'_1a'_2a'_3$, $a_2a_1a_2a_3qa'_2a'_1a'_2a'_3$, $a_1a_3a_2a_1qa'_1a'_3a'_2a'_1$, etc. (in general, such chains can be written as xqx') and that does not contain any other chains. Substantive a_1 and a'_1, a_2 and a'_2 the like can be understood as pairs of elements that are in some way consistent with each other. This is not about the characters a_1, a'_1 themselves, etc., but about their corresponding occurrences in the strings. This language is easily generated by grammar containing permutation rules. For this grammar, there is an equivalent grammar G_1 containing permutation rules and can be represented as follows:

$$\left.\begin{array}{l} 1.\ I \to IA_ia'_i, \\ 2.\ a'_iA_j \to A_ja'_i, \\ 3.\ IA_i \to a_iI, \\ 4.\ I \to q. \end{array}\right\} i, j = 1,2,3$$

(a_i, a'_i, q are main characters; A_i is auxiliary characters; I is initial character). Let's show for example how you can draw a chain in this grammar $a_2a_1a_1a_3qa'_2a'_1a'_1a'_3$:

1. I
2. (1) $IA_3a'_3$
3. (1) $IA_1a'_1A_3a'_3$
4. (1) $IA_1a'_1A_1a'_1A_3a'_3$
5. (1) $IA_2a'_2A_1a'_1A_1a'_1A_3a'_3$
6. (2) $IA_2A_1a'_2a'_1A_1a'_1A_3a'_3$
7. – 11. .... (2; 5 times)
12. (3) $a_2IA_1A_1A_3a'_2a'_1a'_1a'_3$
13. – 15. .... (3; 3 times)
16. (4) $a_2a_1a_1a_3qa'_2a'_1a'_1a'_3$

Language {*xqx'*} cannot be generated by any grammar G_2 . This phenomenon occurs in natural languages, that is, they are possible fragments consisting of chains of appearance *xqx'*. Two examples of this kind are described in the literature:

1) Constructions of type: $\overset{a}{\text{Саша}}$, $\overset{b}{\text{Софія}}$, $\overset{c}{\text{Катя}}$, $\overset{d}{\text{Данило}}$, … – $\overset{a'}{\text{спортсмен}}$, $\overset{b'}{\text{співачка}}$, $\overset{c'}{\text{художниця}}$, $\overset{d}{\text{поет}}$, … *in accordance* [7, 10, 18-20, 32-33, 36, 45]. Here the role *x* (*abcd*…) is played by the chain of proper names, and the role *x'* (*a'b'c'd'*…) is played by the chain of professions, which must be reconciled with these names in the genus [58]; *q* is a dash (more precisely, the knot is null).

2) According to [62], the Indian language is widespread sentences, in which the main complement is duplicated by the incorporation of the relevant bases into the verb-adverb: *The artist paints and paints a landscape*. In addition, any verb (including incorporating additions) is easily substantiated and acquires the ability to act as a supplement: in particular, моя дитина вподобала книгочитання (i.e., "my child liked reading books") [7, 10 , 18-20, 32-33, 36, 45]. This process can theoretically be repeated for an unlimited number of times: він книгочитанняцікаводумає про книгочитанняцікавість ("he thinks of an interesting reading books"), i.e.

$$\text{Він}\ \overbrace{\text{книго}}^{a}\overbrace{\text{читання}}^{b}\overbrace{\text{цікавість}}^{c} - \text{думає про} - \overbrace{\text{книго}}^{a'}\overbrace{\text{читання}}^{b'}\overbrace{\text{цікавість}}^{c'}.$$

Here *x'* (=*a'b'c'd'*) is a supplement, *x* (=*abcd*) its duplicate incorporated into a preposition, and a *q* is preposition. Such a construction is only correct when the incorporated duplicate of the supplement exactly matches the complement in the composition and the order of compliance of the basics.

Taking these examples into account, grammars G_2 are not sufficient to describe all natural languages in their entirety. But both examples are peripheral: the first design, although permissible, probably in any language, is very specific and not related to the consumed, and the second, one that is very general and, apparently, sufficiently consumed, known only in one less common language. Therefore, with all the theoretical value of these examples, they can be neglected. If you turn away from them, then grammars G_2 can be considered in principle sufficient means for describing natural languages. This statement, of course, can be rigorously proven; belief in its truth is based on a number of empirical considerations.

1. There are so-called categorical grammars related to recognizable grammars. These grammars were developed and applied to natural languages regardless of grammars G_2 , with no examples of their inadequacy (except for the two mentioned above) so far. However, it has been proved by A. Gladkiy [14-16] that the class of languages described by categorical grammars coincides with the class of context-free languages.
2. More recently, slot machines with the ability to carry both recognition and generation have been offered to describe languages. Chomsky proved that all the languages processed by such automatons are context-free, and vice versa [38-41, 49-57]. Thus, another formal model of natural language, introduced for

independent reasons and without significant fundamental difficulties, is equivalent to grammar G_2 .

3. Within mathematical linguistics, a class of so-called finite-characterized languages that are intuitively close to natural languages is easily distinguished. All finite-character languages are context-free (reverse is wrong!). This again suggests that grammars G_2 are capable of producing natural languages.
4. Finally, there are a number of algorithms for automatic analysis and generation of texts in natural languages that are used as descriptions of the corresponding grammar G_2 languages or equivalent systems. The grammars are based, for example, on syntax analysis algorithms for several languages being developed at the University of Texas [64], a number of algorithms using the so-called Coco method [61], and some other algorithms mentioned in [6, 59, 66].

All this compels grammar to be sufficient for natural languages. In particular, it is worth noting that constructions of the type *abcd…d’c’b’a’* that are not described by grammars can easily be generated by grammar G_3. Yes, it's easy to show that a language that is exactly a strand of a given type (composed of a_1, a_2, a_3, a'_1, a'_2, a'_3) is generated by a grammar G_2 containing only six rules:

$$\left.\begin{array}{l} I \to a_i I a_i' \\ I \to a_i a' \end{array}\right\} i = 1, 2, 3.$$

Now, the following two important points need to be made.

Firstly, what is said does not mean that grammars G_2 produce only natural languages and / or languages close to them: among context-free languages there are those that are not at all similar in structure to natural ones. Secondly, the fact that grammars G_2 are practically sufficient to describe natural languages does not imply that they are always convenient for this purpose, that is, they allow one to describe any natural language construct in a natural way. Grammar G_2 does not, for example, provide a natural (*artificial intelligence*) description for so-called non-design structures, that is, for structures with discontinuous components (*or with intersections* - - - - , - - - - etc., or framed by - - - - , - - - - syntactic arrows) [14-16]. In this case, non-design structures are available in a variety of languages:

Ukr. Наша мова, як і будь-яка інша, посідає унікальне місце. [7, 10, 18-20, 32-33, 36, 45].

Rus. [6, 9, 14-16].

К этой поездке может пробудить интерес только выступление директора.

Eng. A theorem is stated which describes the properties of this function. [1, 34, 38-41, 49-60].

Ger.

... die Tatsache, daß die Menschen die Fähigkeit besitzen, Verhältnisse der objektiven Realität in Aussagen wiederzuspiegeln. [25, 30-31, 35, 42].

Fr. ... la guerre, dont la France portait encore les blessures... [14-16].

Serb-chor. Regulacija procesa jedan je od najstarjih oblika regulacije. [14-16].

Hung. Azt hisszem, hogy késedelmemmel sikerült bebizonyítani. [14-16].

To describe the structure of similar phrases in terms of components (and grammars G_2 describe the syntactic structure exactly), then for natural description it is necessary to use discontinuous components: all words dependent on the same word must form (with it) one component, and this is in the absence of projectivity will necessarily lead to the appearance of discontinuous components *(before this trip ... interest*, and theorem ... which describes the properties of this function, etc.). However, the systems of grammar G_2 components ascribed to grammar phrases and, moreover, to any grammar of immediate components, discontinuous components can not contain.

Consider two special cases of grammar G_2 equivalent to grammar G_3.

The first case. In natural languages it is possible to place dependent words to the right of the main (right subordination) [14-16]:

назва курсу, лист бумаги, une regle stricte, give him,

or to the left of the main (left subordination) [14-16]:

основний курс, белый лист, cette regle, good advice.

Both right and left subordination can be sequential [7, 10, 18-20, 32-33, 36, 45]:

витяг з протоколу звітування з наукової діяльності заступника завідувача кафедри ІСМ інституту ІКНІ Національного університету "Львівська політехніка" міста Львова країни Україна

or жена сына заместителя председателя второй секции эклектики совета по прикладной мистике при президиуме Академии наук королевства Мурак

and досить повільно рухлива черепаха

or очень быстро бегущий олень.

Depending on the language, one or another consecutive right or left subordinated construct may be theoretically unlimited: for example, a consecutive subordinated construct in the Ukrainian language (unlimited right subordination) and a similar construction in Lithuanian (where N_p always preceded by a word which leads to unlimited left subordination). The fact that the languages of the world are different and can be classified by the predominance of right or left subordination and, in particular, depending on the possibility of unlimited consistent subordination in one direction or another, has been noted and investigated in [63]. Also, this problem is in connection with the use of grammars G_2 for the description of natural languages was drawn by V. Ingve [21, 65]. He noted that there are a large number of languages (e.g. English, Ukrainian, French, etc.) in which consistent right subordination is in principle not restricted, and in left subordination the length of the chain is always limited due to the structural features of these languages. B. Ingve's hypothesis [21], which attempts to explain this empirical observation by some general patterns of the structure of the human psyche.

It turns out that the grammar G_2 that generates such a language has the following interesting property: for any terminal chain that is output, there is such a conclusion at each line that all the auxiliary symbols are collected at the right end, occupying no more than K the last places (K is constant, fixed for the given grammar, that is, the same for all conclusions in it). In order for grammar G_2 to have the specified property, there is not enough bounding of consecutive left subordination. It is necessary to fulfill a number of stronger and difficult formulated requirements [26], which imply, for example, the limitation of the right parallel subordination . . . . and the consistent embedding of the type . . . . [48].

If each line of output is divided into two parts: left - one terminal character to the first auxiliary character X - and right of X - inclusive to the end (the right part can also contain terminal characters), then the right part will always contain no more than K characters. The left part of the content is interpreted as an already "issued" piece of the generated chain (in the next steps of the output, this piece is no longer amenable to any processing), and the right - as a working area, which the grammar should keep in memory. Thus, the number K is nothing but the maximum amount of memory required to generate any chain in the given grammar (i.e. there will be a chain that is not generated by the amount of memory $< K$). This number coincides with the maximum chain length of consecutive left subordinates possible in a given language: yes, if there is no more than three consecutive subordinates left in a given language, then, when generating this language, it is possible to construct such an output in which no need arises remember more than three characters at a time. A marked relationship between the allowable depth of the left subordination and the amount of memory was established by V. Ingve [21, 65-66]. Let us illustrate the example, namely, consider the grammar G_2 that gives rise to some nominal groups of the Ukrainian language [7, 10, 18-20, 32-33, 36, 45], in which the right subordination is not limited and the depth of the left does not exceed two.

Example 1 for scheme of grammar G_2

$$\tilde{N}_{x,y,z} \to N_{x,y,z}\tilde{N}_{x',y',p} \qquad \tilde{N}_{x,y,z} \to \tilde{A}_{x,y,z}\tilde{N}_{x,y,z}$$

$$\tilde{A}_{x,y,z} \to \{very, enough, exact, easy, important..., \}A_{x,y,z}$$

$$\tilde{N}_{x,y,z} \to N_{x,y,z} \qquad \tilde{A}_{x,y,z} \to A_{x,y,z} \qquad N_{ж,y,z} \to system_{y,z},...$$

$$N_{ч,y,z} \to request_{y,z}, user_{y,z}, resource_{y,z}, business_{y,z},...$$

$$A_{x,y,z} \to informational_{x,y,z}, simple_{x,y,z},...$$

(The peculiarities of the agreement A with the animated $N_{x,y,н}$ are not taken into account.) Here is an example of the conclusion in grammar G_2 [7, 10, 18-20, 32-33, 36, 45]:

$\tilde{N}_{m,single}$

$\tilde{A}_{m,single}\tilde{N}_{m,single}$

simple $A_{m,single}\tilde{N}_{m,single}$

simple enough $A_{m,single}\tilde{N}_{m,single}$

simple enough informational $\tilde{N}_{m,single}\tilde{N}_{m,single}$

simple enough informational $N_{m,single}\tilde{N}_{m,single}$

simple enough informational request $\tilde{N}_{m,single}$

simple enough informational request $\tilde{N}_{m,single}\tilde{N}_{m,single}$

simple enough informational request $N_{m,single}\tilde{N}_{m,single}$

simple enough informational request (of) *user* $\tilde{N}_{m,single}$

simple enough informational request (of) *user* $\tilde{N}_{m,single}\tilde{N}_{f,single}$

simple enough informational request (of) *user* $N_{m,single}\tilde{N}_{f,single}$

simple enough informational request (of) *user* (of) *resource* $\tilde{N}_{f,single}$

simple enough informational request (of) *user* (of) *resource* $\tilde{N}_{f,single}\tilde{N}_{m,single}$

simple enough informational request (of) *user* (of) *resource* $N_{f,single}\tilde{N}_{m,single}$

simple enough informational request (of) *user* (of) *resource system* $\tilde{N}_{m,single}$

simple enough informational request (of) *user* (of) *resource system* $\tilde{N}_{m,single}$

simple enough informational request (of) *user* (of) *resource system* (of) *business*

Example 2 for scheme of grammar G_2

$\tilde{N}_{x,y,z} \to N_{x,y,z}\tilde{N}_{x',y',p}$ $\quad$ $\tilde{N}_{x,y,z} \to \tilde{A}_{x,y,z}\tilde{N}_{x,y,z}$

$\tilde{A}_{x,y,z} \to \{really, enough, exact, easy, important...\}A_{x,y,z}$

$\tilde{N}_{x,y,z} \to N_{x,y,z}$ $\quad$ $\tilde{A}_{x,y,z} \to A_{x,y,z}$

$N_{ж,y,z} \to school_{y,z},...$ $\quad$ $N_{ч,y,z} \to laugh_{y,z}, pupil_{y,z}, Lviv_{y,z},...$

$N_{c,y,z} \to city_{y,z},...$ $\quad$ $A_{x,y,z} \to joyful_{x,y,z}, childish_{x,y,z},...$

(The peculiarities of the agreement A with the animated $N_{x,y,н}$ are not taken into account.) Here is an example of the conclusion in grammar G_2 [7, 10, 18-20, 32-33, 36, 45]:

$\tilde{N}_{m,single}$

$\tilde{A}_{m,single}\tilde{N}$

very $A_{m,single}\tilde{N}_{m,single}$

very joyful $A_{m,single}\tilde{N}_{m,single}$

very joyful children $\tilde{N}_{m,single}\tilde{N}_{m,single}$

very joyful children $N_{m,single}\tilde{N}_{m,single}$

very joyful children laugh $\tilde{N}_{m,single}$

very joyful children laugh $\tilde{N}_{m,single}\tilde{N}_{m,single}$

very joyful children laugh $N_{m,single}\tilde{N}_{f,single}$

very joyful children laugh (of) *pupil* $\tilde{N}_{f,single}$

very joyful children laugh (of) *pupil* $\tilde{N}_{f,single}\tilde{N}_{f,single}$

very joyful children laugh (of) *pupil* $N_{f,single}\tilde{N}_{n,single}$

very joyful children laugh (of) *pupil* (of) *school* $\tilde{N}_{n,single}$

very joyful children laugh (of) *pupil* (of) *school* $\tilde{N}_{n,single}\tilde{N}_{n,single}$

very joyful children laugh (of) *pupil* (of) *school* $N_{n,single}\tilde{N}_{n,single}$

very joyful children laugh (of) *pupil* (of) *school* (of) *city* $\tilde{N}_{m,single}$

very joyful children laugh (of) *pupil* (of) *school* (of) *city* $N_{m,single}$

very joyful children laugh (of) *pupil* (of) *school* (of) *city Lviv*

In this output, the amount of storage is two: no intermediate chain contains more than two auxiliary characters. The same chain could be generated in a different way by using more memory, for example, first retrieving from the $\tilde{N}_{m,single}$ chain

$$very\ A_{m,single}\ A_{m,single}\ N_{m,single}\ N_{m,single}\ N_{f,single}\ N_{n,single}\ N_{m,single}$$

and from it our terminal chain. For us, however, the amount of memory required is important, which means that it is not possible to get this chain with less volume. It is this volume that is equal to two here.

You can prove that any terminal chain that is displayed in can be generated G_2 with the amount of memory ≤2. The proof is based on a very simple reasoning: the "good" conclusion should be drawn so that for each noun first its terminal dependents were issued in terminal form, and only then the name group was deployed to the right.

Theorem 1. Grammar G_2 of the type described (with limited memory) is always equivalent to some grammar G_3 [14-16]. This is not easy to prove (the proof is that the right side of the *K* character string is encoded with one new auxiliary character). Thus, in the case of languages with a limited depth of left subordination G_2, grammar with limited memory, equivalent to grammars G_3 and close to them in the construction of conclusions, ie arranged much easier than arbitrary grammars G_2, are not only fundamentally sufficient, but also very convenient - they provide natural description.

There are, however, languages in which not only the right but also the left sequential subordination have unlimited depth. A similar language is, for example, Hungarian, where unrestricted left subordination comes from the prepositional common definitions, and unlimited right subordination at the expense of, for example, the subordinate clauses of which (The house that Jack built) [14-16]. See an example from the novel by G. Feher is a joking toast given in [68].

1. Kivánom, hogy valamint az agyag[23] цlelx karjai[22] közül kibontakozni[21] akary[20] kocsikerйk[19] rettentx nyikorgбsбtyl[18] megriadt[17] juhбszkutya[16] bundбjбba[15] kapaszkody[14] kullancs[13] kidülledt fйlszemйbxl[12] alбcseppent[11] kцnnyeseppben[10] visszatьkrцzxdx[9] holdvilág fйnyйtxl[8] illuminбlt[7] rablylovagvбr[6] felvonyhidjбbyl[5] kiбlly[4] vasszegek[3] kohйziys erejйnek[2] hatбsa[1] évszézadokra összetartja annak materiáját, aképpen tartsa össze ezt a társaságot az igaz szeretet.

2. Я хочу, аби справжнє кохання скріпило цю компанію так, як на століття скріплює матеріал мосту дія[1] єднальної сили[2] цвяхів[3], що торчать[4] з підіймального мосту[5] розбійницького феодального замку[6], освященного[7] місячним світлом[8], що відображається[9] в краплині[10], яка витікає[11] з витріщеного ока[12] клеща[13], що вчепилася[14] в шерсть[15] вівчарки[16], наполоханої[17] жахливим скрипом[18] возових колес[19], що прагнуть[20] вирватися[21] з обіймів[22] грязюки[23] [7, 10, 18-20, 32-33, 36, 45].

This phrase from an artistic text has a depth of 22 and is absolutely correct from a grammatical point of view (to the same extent as its Ukrainian translation). Moreover, nothing prevents the continuation of the ad libitum chain of definitions.

To generate languages with this property, another special type of grammar G_2 can be offered, in some sense more general than the grammar G_2 with limited memory discussed above. First of all, let's state more precisely what languages we have in mind here. These are languages in which an unlimited number of sequentially subordinate structures from left to right $X_1X_2\ldots X_i\ldots$ (unlimited right subordination) is possible, and in each of these structures X_i an unlimited left subordination is possible - a sequence of structures $\ldots X_{ij}\ldots X_{i3}X_{i2}X_{i1}$; however, unlimited X_{ij} deployment is not possible within the structures. With regard to the Hungarian language X_i, it can be understood as a simple sentence, which is each (except the first) additional determinative of the previous one, and X_{ij} - as a prepositional participle.

Consider a grammar $\Gamma' = \langle V', V_1', I', S' \rangle$ whose basic vocabulary V' consists of n symbols $A_1, A_2, ..., A_n$ and whose rules have the form $X \to YA_i$ or $X \to A_i$, where X and Y belong to V_1' [14-16]. Let us put in accordance with each of the symbols A_i some regular grammar $\Gamma_i' = \langle V, V_1^i, A_i, S_i \rangle$, where V is the main vocabulary, common to all Γ_i', V_1^i is the auxiliary vocabulary, which does not contain any characters with V' and V_1' except A_i; A_i is initial symbol; the rules of the S_i scheme are either $C \to dD$ or $C \to c$ (here, as in the other examples, capital letters are denoted by auxiliary characters, and lowercase characters are capitalized). In this case, we assume that the grammar Γ_i' auxiliary dictionaries do not intersect in pairs.

The grammar Γ' is very close to the automaton, differing from it only by the direction of unfolding (the direction of unfolding here refers to the direction in which "terminal" characters, such as $C \to dD$ - the left unfolding) are generated; in fact, it is automatic with a precision to mirror symmetry. So we are dealing with one quasi-regular right-deployment grammar and regular left-deployment grammar.

Consider now the union of all these grammars, more precisely, the grammar Γ in which the main vocabulary is V (the same as all Γ_i'), the auxiliary dictionary - $V_1 = V' \cup V_1' \cup V_1^1 \cup V_1^2 \cup ... \cup V_1^n$ (i.e. the union of auxiliary dictionaries of all grammars Γ' Γ_1' Γ_2', …, Γ_n', and the basic grammar dictionary Γ'), the initial symbol I (same as Γ'), and the scheme is a combination of the schemes of all grammars Γ' Γ'_1 Γ'_2, …, Γ'_n. This grammar Γ is a special *context-free grammar that can be called context-free grammar with independent bilateral deployment*. The fact that this grammar is not automatic is obvious, at least because some of its rules (schema rules S) have two auxiliary characters in the right-hand side. The basic grammar Γ' symbols (i.e. $A_1, A_2, ..., A_n$) in the grammar Γ are auxiliary, so the rules of appearance $X \to YA_i$ within Γ are not "automatic". But grammar Γ is equivalent to automatic. Here is an example (schema) of such grammar.

$$S' = \begin{cases} I \to BA_1 \\ B \to CA_1 \\ C \to BA_2 \\ C \to DA_3 \\ D \to DA_4 \\ D \to A_2 \end{cases} \quad S_1 = \begin{cases} A_1 \to bP_1 \\ P_1 \to aQ_1 \\ Q_1 \to aQ_1 \\ Q_1 \to c \end{cases} \quad S_2 = \{A_2 \to d \quad S_3 = \begin{cases} A_3 \to aP_3 \\ A_3 \to bQ_3 \\ A_3 \to cR_3 \\ P_3 \to a \\ Q_3 \to b \\ R_3 \to dR_3 \\ R_3 \to eR_3 \\ R_3 \to d \end{cases} \quad S_4 = \begin{cases} A_4 \to cP_4 \\ P_4 \to b \end{cases}$$

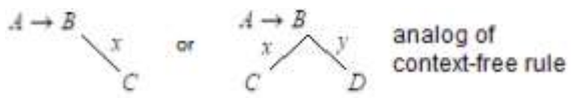
The grammar introduced by the Gladkiy type works like this. Initially, the generated chain is infinitely unfolded from left to right by symbols A_i (which can be interpreted, for example, as syntax groups or sentences S_i); this is done by the rules *S*'. Then any one of the rules A_i can be expanded indefinitely from right to left into a chain of terminal characters (which can be interpreted as words). Such a process of generation is convenient in such cases, for example, as the Hungarian phrases of the type discussed above.

Theorem 2. Each context-free grammar with independent bilateral deployment is equivalent to some regular grammar [14-16].

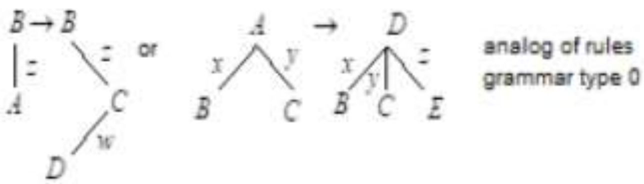
Unlimited grammars of type 0 are only a special case of the general concept of grammar. However, they are certainly sufficient to describe all natural languages in their entirety. Any natural language (set of correct phrases) is an easily recognizable set. This means that there is a fairly straightforward phrase recognition algorithm. If the language is recognized by an algorithm with the specified memory limit, then it can be generated by grammar, where for any terminal chain of the output length there is an output in which no intermediate chain exceeds the length of the number *Kn* (*K* - is some constant). Such grammar is a *grammar with limited stretching,* where the capacitive signal function is no more linear. For any grammar with limited stretching, it is possible to construct an equivalent grammar G_0 that can describe many correct phrases of any natural language, that is, to produce any correct phrases of the given language, without generating any wrong ones. Both constructions, given as examples of the inapplicability of context-free grammars, are easily described by grammar G_0.

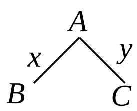
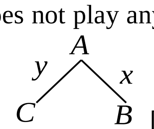
The disadvantages of the *method of grammar G_0 deduction* are reduced to three points.

1. It is not possible to naturally describe phrases with discontinuous components.
2. Grammar contains only rules for the formation of linguistic expressions, such as word forms or phrases. Grammar sets the correct expressions as opposed to the wrong ones.
3. Grammar G_0 builds sentences at once with exactly the same order of words - with what those sentences should be in their final form. This generates a syntactic structure in the form of an ordered tree, that is, a tree, where, in addition to the subordination relation given by the tree itself, there is also a linear order relation (to the right - to the left). Thus, the syntactic structure of grammar G_0 does not break down two completely different in nature, although related: syntax subordination and linear interposition. But to characterize the syntactic structure is to specify the relation of syntactic subordination. As for the linear order relation, it characterizes not the structure but the phrase itself. The order of words depends on the syntactic structure; it is determined necessarily by its account and thus is in relation to it something derivative, secondary. It is advisable to modify the concept of generating grammar so that the left and right parts of the substitution rules are not linearly ordered chains, but, for example, trees (without linear ordering) depicting syntactic relations [14 - 16]. Then the rules look like this:



or



Index bars represent syntactic links of different types; letters $A,B,C,\ldots$- are syntactic categories. *NB*: the relative arrangement of characters of one level of subordination does not play any role and is accidental  in this scheme; means the same as  [14-16].

The result is a computation of the syntactic structures (not phrases) of the language. This computation is part of the generating grammar. The other part of this grammar is the computation that, for any given syntactic structure, specifies (taking into account any other factors, such as in the Ukrainian language - with the mandatory accounting of logical highlighting, etc.) all possible linear sequences of words for it. Then the problem of discontinuous components is removed. It is impossible to get a natural representation of the structure of the immediate components of that sentence from the regular grammar. That is, regular grammars give some structure to constituents, as in general all grammars of direct constituents, however, these constituents are usually formal. C_1 is different content from different information resources. Text content C_2 (article, commentary, book, etc.) from C_1 contains a considerable amount of data in natural language, some of which is abstract. The text is presented as a unified sequence of character units whose main properties are information, structural and communicative connectivity / integrity, which reflects the content / structure of the text. Linguistic content analysis (such as comments, forums, articles, etc.) is a method of word processing. The text processing process divides the content into tokens using finite state machines. As a functional-semantic-structural unity, the text has rules of construction, reveals patterns of meaningful and formal connection of constituent units. Cohesiveness is manifested through external structural indicators and formal dependence of the text components, and integrity through thematic, conceptual and modal dependence. Integrity leads to a meaningful and communicative organization of text, and coherence to a form, a structural organization. Commercial Content Keyword Detection Operator $\alpha:(C_2,U_K,T)\rightarrow C_3$ is a mapping of commercial content C_2 to a new state that is different from the previous state by having a plurality of keywords that generally describe its content. The analysis investigates the multilevel content structure: linear sequence of characters; linear sequence of morphological structures; linear sequence of sentences; network of interconnected unities (alg. 1). The analysis explores the multilevel structure of textual content: linear sequence of characters; linear sequence of morphological structures; linear sequence of sentences; network of interconnected unities (alg. 1).

The analysis explores the multilevel structure of textual content: linear sequence of characters; linear sequence of morphological structures; linear sequence of sentences; network of interconnected unities (alg. 1).

Algorithm 1. Linguistic analysis of textual commercial content.

Section 1: Grammar analysis of textual content C_2.

Step 1. Divide textual commercial content C_2 into sentences and paragraphs.

Step 2. Divide the content character chain C_2 into words.

Step 3. Allocate numbers, numbers, dates, unchanged turns, and content cuts C_2.

Step 4. Remove non-text content characters C_2.

Step 5. Formation and analysis of linear sequence of words with service marks for content C_2 (alg. 3).

Section 2: Morphological analysis of textual content C_2.

Step 1. Getting the basics (word forms with severed endings).

Step 2. A grammatical category is formed for each wordform (collection of grammatical meanings: genus, case, pronunciation, etc.).

Step 3. Formation of linear sequence of morphological structures.

Section 3: Syntax analysis $\alpha:(C_2,U_K,T)\rightarrow C_3$ of textual content C_2 (alg. 2).

Section 4: Semantic analysis of textual content C_3.
Step 1. Words correlate with semantic vocabulary classes.
Step 2. Selection of morphosemantic alternatives needed for this sentence.
Step 3: Bind the words into a single structure.
Step 4. Generate an ordered set of superposition entries from basic lexical functions and semantic classes. The accuracy of the result is determined by the completeness / correctness of the dictionary.
Section 5: Reference analysis for the formation of interphase unities.
Step 1. Contextual analysis of text commercial content C_3. With its help, the resolution of local references (the one that is, his) is realized and the expression of the expression is the kernel of unity.
Step 2. Thematic analysis. Separation of statements on a theme and rheum distinguishes thematic structures which are used, for example, in the formation of a digest.
Step 3. Determine the regular repetition, synonymization and re-nomination of keywords; the identity of the reference, that is, the ratio of words to the subject of the image; presence of implication based on situational connections.
Section 6: Structural analysis of textual content C_3. The prerequisites for use are a high degree of coincidence of terms of unity, a discursive unit, a sentence in a semantic language, utterance, and an elementary discursive unit.
Step 1: Identify the basic set of rhetorical connections between content unities.
Step 2. Building a nonlinear unity network. The openness of a link set involves its extension and adaptation to analyze the structure of the text .

Parsers work in two stages: identify meaningful tokens and create a parse tree (alg. 2). The text implements the structured activity, which involves the subject and object, process, purpose, means and result, which are reflected in the content-structural, functional, communicative indicators. The units of internal organization of the structure of the text are the alphabet, vocabulary (paradigm), grammar (syntagmatics), paradigms, paradigmatic relations, syntagmatic relations, rules of identification, expression, between phrase unity and fragments-blocks. At the compositional level, there are sentences, paragraphs, paragraphs, sections, chapters, chapters, pages, etc. that, besides sentences, are indirectly related to the internal structure, so are not considered. They use the database (terms / morphemes database and official parts of the language) and defined text analysis rules to search for a term. Parsers work in two stages: identify meaningful tokens and create a parse tree (alg. 2).

Algorithm 2. Commercial Content Syntax.

Section 1: Identification of content tokens $U_{K1} \in U_K$ for commercial content C_2.
Step 1. Define the term chain as a sentence.
Step 2. Identify the group name using the basics dictionary.
Step 3. Identify a verb group using the basics dictionary.
Section 2: Create a parse tree from left to right. The output of a tree is to expand one of the characters in the previous string of a sequence of linguistic variables, or to replace it with another, the other characters are overwritten without change. On de-

ployment, the replaceable / rewritable characters (ancestors) connect directly to the characters that result from the deployment, replacement, or rewriting (descendants), and receive a component tree, or syntax, for commercial content.
Step 1. Deploying a named group. Deploying a verb group.
Step 2. Implementation of syntactic categories with word forms.
Section 3: Determine the plurality of content keywords $\alpha:(C_2, U_K, T) \to C_3$ for C_2.
Step 1. Define the terms $Noun \in U_{K1}$ is nouns, noun phrases, or noun adjectives among the plural words of textual content.
Step 2. Calculation of $Unicity$ uniqueness for terms $Noun \in U_{K1}$.
Step 3. Calculation $NumbSymb \in U_{K3}$ (number of characters without spaces) for $Noun \in U_{K1}$ approx for $Unicity$.
Step 4. Calculation $UseFrequency \in U_{K2}$ is frequency of occurrence of content keywords. For term $NumbSymb \leq 2000$ the frequency $UseFrequency$ is within $(6;8]$%, from $NumbSymb \geq 3000$ is $[2;4)$%, from $2000 > NumbSymb < 3000$ is $[4;6]$%.
Step 5. Calculation - frequency of occurrence of keywords at the beginning of text, $IUseFrequency$ - frequency of occurrence of keywords in the middle of text, $EUseFrequency$ - keywords occurrence frequency at the end of text of content.
Step 6. Compare values $BUseFrequency$, $IUseFrequency$ and $EUseFrequency$ for prioritization. Higher-value keywords $BUseFrequency$ have higher priority than higher-value keywords.
Step 7. Sort your keywords according to their priorities.
Section 4: Fill in the content search engine base C_3, that is attributes $KeyWords \in U_{K4}$ is keywords, $Unicity$ is keyword uniqueness ≥ 80, $Noun$ is term, $NumbSymb$ is number of characters without spaces, $UseFrequency$ is frequency of keywords, $BUseFrequency$ is frequency of keywords at the beginning of text, $IUseFrequency$ is frequency of keywords in the middle of the text, is the frequency of keywords used at the end of the text.

Detecting commercial content C_2 keywords from a text snippet is performed using the processes shown in Figure 1. The text implements structurally submitted activity that involves the subject and object, process, purpose, means and result, which are reflected in the content-structural, functional, communicative indicators. The units of internal organization of the structure of the text are the alphabet, vocabulary (paradigm), grammar (syntagmatics), paradigms, paradigmatic relations, syntagmatic relations, rules of identification, expression, between phrase unity and fragments-blocks. At the compositional level, there are sentences, paragraphs, paragraphs, sections, chapters, chapters, pages, etc. that, besides sentences, are indirectly related to the internal structure, so are not considered. They use the database (terms / morphemes database and official parts of the language) and defined text analysis rules to search for a term. Based on the rules of generative grammar, the term is adjusted according to the rules of its use in context. The sentences set the boundaries of punctuation, anaphoric, and cataphoric references. The semantics of the text are caused by the communicative task of transmitting data. The structure of the text is determined by the internal organization of the text units and the patterns of their interrelation. Through parsing, the text is framed into a data structure, for example, into a tree that matches the syntactic structure of the input sequence, and is best suited for further processing. After analyzing a snippet of text and a term, they synthesize a new term as a content topic keyword, using a base of terms and their morphemes. Next, we synthesize terms to form a new keyword using the base of the official parts of the language. The term keyword detection principle is based on Zipf's law and comes down to medium-frequency word selection (most used words are ignored through stop dictionaries and rare words are ignored).

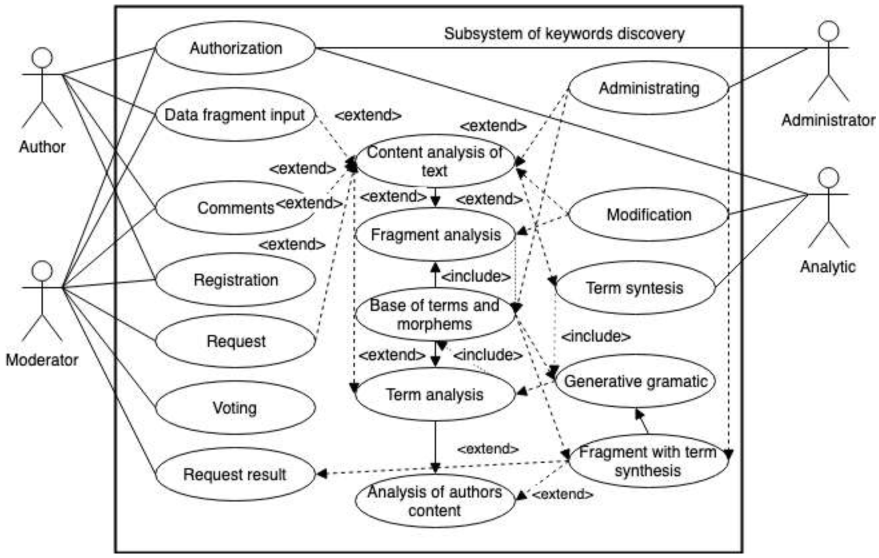


Fig. 1. Use case diagram for the keyword discovery process

The content content analysis is responsible for the process of extracting grammatical data from a word through grapheme analysis and correcting the results of morphological analysis through analyzing the grammatical context of linguistic units (alg. 3).

Algorithm 3. Rubrication of text commercial content

Section 1: Divide the commercial content C_3 into blocks.

Step 1. Submission of commercial content blocks to the tree-building input C_3.

Step 2. Create a new block in the block table.

Step 3. Accumulate characters to a newline character.

Step 4. Check for a period before the newline character. If so, go to step 5, if not, save the sequence to the table, parse the new content block C_3, and go to step 3.

Step 5: Check the end of the text for content C_3. If the end of the text, then the transition to step 6, if not, stores the cached sequence in the table, parsing the new content block C_3 and the transition to step 2.

Step 6. Retrieve the content C_3 block tree as a table $U_{CT}^{B} \in U_{CT}$.

Section 2: Divide the block into sentences with the content structure preserved C_3.

Step 1. A block table is fed to the input $U_{CT}^{B} \in U_{CT}$. Creating a sentence table $U_{CT}^{R} \in U_{CT}$ with a link in the n_to-1 partition_code field with a content block table C_3.

Step 2. Create a new sentence in the sentence table $U_{CT}^{R} \in U_{CT}$.

Step 3. Accumulate a semicolon, semicolon, or newline character.

Step 4. Check for reduction. If it is an abbreviation, then go to step 5, if not, save the sequence in the table, parse the new sentence, and go to step 2.

Step 5. Check the content of the block text for content. If the end of the text, then go to step 6, if not, save the sequence in the table $U_{CT}^{R} \in U_{CT}$, parse the new sentence and go to step 2.

Step 6. Get the output of the sentence tree as a table $U_{CT}^{R} \in U_{CT}$.

Step 7: Check the end of text for content C_3. If the end of the text, then go to step 8, if not, parse the new block and go to step 1.

Step 8. Getting the output of a tree of sentences in the form of tables $U_{CT}^{R} \in U_{CT}$.

Section 3: Divide the sentences into tokens, indicating the belonging to the sentences $U_{CT}^{L} \in U_{CT}$.

Step 1. Formation based on the sentence table of the lexemes table $U_{CT}^{L} \in U_{CT}$ with the fields Codex (unique identifier), Sentence code (number equal to the code of the sentence with the token), Numberx (number equal to the number of the tokens in the sentence), Text (text of the tokens).

Step 2: Log in to parse the sentence tokens from the sentence table $U_{CT}^{R} \in U_{CT}$.

Step 3. Create a new token in the token table $U_{CT}^{L} \in U_{CT}$.

Step 4. Accumulate characters to a point, a space, or the end of a sentence and save it in the token table.

Step 5. Check the end of the sentence. If so, go to step 6, if not, save the accumulated sequence to the table $U_{CT}^{L} \in U_{CT}$, parse the new tokens, and go to step 3.

Step 6. Performing syntax analysis based on raw data (alg. 2).

Step 7. Conduct morphological analysis based on the output data.

Section 4: Identify the topic of commercial content $U_{CT}^{T} \in U_{CT}$.

Step 1. Build a hierarchical structure of the properties $U_{CT}^{T} \in U_{CT}$ of each lexical unit of text containing grammatical and semantic information.

Step 2. Formation of a lexicon with hierarchical organization of types of properties, where each type-descendant inherits and redefines the properties of the ancestor.

Step 3. Unification is the basic mechanism for constructing a syntactic structure.
Step 4. Definition of keywords $KeyWords$ of commercial content $C_4 = \alpha_5(\alpha_4(C_2, U_K), U_{CT})$ at $U_{CT} = \{U_{CT1}, U_{CT2}, U_{CT3}, U_{CT4}\}$, where U_{CT} is collection of terms of rubric, U_{CT1} is set of thematic keywords from the dictionary, U_{CT2} is set of frequencies of usage of keywords in commercial content, U_{CT3} is set of dependencies of use of keywords of different subjects (coefficients are determined by the moderator according to the keyword to specific topics within $[0,1]$), U_{CT4} is the set of frequencies of usage of content keywords in content. (alg. 2).
Step 5. Definitions of $U_{Ct}^T \in U_{Ct}$ with $TKeyWords$ is themed keywords plural for $KeyWords$, $Topic$ is content topic and $Category$ - content category.
Step 6: Determine $FKeyWords$ is frequency of Keyword Usage and $QuantitativeryTKey$ - Frequency of Usage of Topical Keywords in Commercial Content.
Step 7. Definition $Comparison$ is comparison of occurrence of keywords of different topics Calculation $CofKeyWords$ is coefficient of thematic content keywords, $Static$ is coefficient of statistical importance of terms, $Addterm$ is coefficient of availability of additional terms. Comparison of multiple content keywords with key topic concepts, if there is a match, then go to step 9, if not, move to step 8.
Step 8. Formation of a new heading with a set of key concepts of the analyzed C_4.
Step 9. Assign a specific rubric to the analyzed commercial content C_4.
Step 10. Calculation is the coefficient of content C_4 placement in the topic heading.
Section 5: Filling in the search engine base for attributes $Topic$ is content topic, $Category$ is content category, $Location$ is content placement coefficient in the content column, $CofKeyWords$ is content keyword content coefficient, - statistical significance of terms, $Static$ is coefficient of availability of additional terms, $TKeyWords$ is topics of availability of additional terms, $FKeyWords$ is frequency of use of keywords, $Comparison$ is comparison of occurrence of keywords of different subjects, $QuantitativeryTKey$ is frequency of use of thematic keywords in the text of content C_4.

The construction of the content C_4 text is determined by the theme, the expressed information, the terms of communication, the task of the message and the style of presentation. The semantic, grammatical and compositional structure C_4 of the content is related to its stylistic / stylistic characteristics, which depend on the identity of the author and are subordinate to the thematic / stylistic dominance of the text. The process of C_4 categorization in the form of a variant diagram is shown in Fig. 2. The main stages of determining the morphological features U_{CT} of the units of the text C_4: the definition of grammatical classes of words - parts of language and principles of their classification; isolation of a part of word semantics as morphological, substantiation of a set of morphological categories and their nature; a description of the set of formal means assigned to parts of language and their morphological categories. The process of heading $C_4 = \delta(\alpha(C_2, U_K), U_{CT})$ through the automatic indexing of the components of commercial content C_3 is divided into successive blocks: morphological analysis, syntactic analysis, semantic-syntactic analysis of linguistic constructions and variation of the content record of textual content.

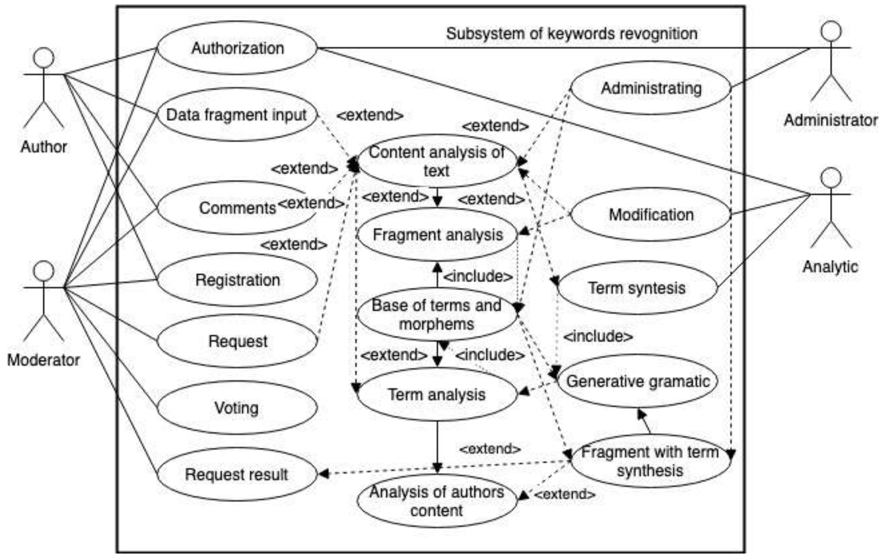


Fig. 2. A diagram of the use cases for the content heading process in the SEC

The following grammatical meanings have been used: synthetic, analytical, analytical, synthetic, and subitive. The grammatical meanings are generalized because of the same characteristics and can be divided into partial meanings. The concept of grammatical category was used to refer to classes of the same grammatical meanings. Morphological values include the categories of genus, number, case, person, time, method, condition, species, combined into paradigms for classifying parts of a text. The object of morphological analysis is the structure of the word, the forms of word exchange, the ways of expressing grammatical meanings. Morphological features of units of text are tools for exploring the connection between vocabulary, grammar, their use in speech, paradigmatics (distinct forms of declining words), and syntagmatic (linear conjunctions of words, conjunctions). The implementation of the automatic encoding of text words, that is, the attribution of grammatical class codes, is associated with grammatical classification. Morphological analysis contains the following steps: selection of the basis in word form; search for the basics in the basics dictionary; comparison of word structure with data in dictionaries of basics, roots, prefixes, suffixes, flexions. In the analysis process, the meanings of words and the syntagmatic relationships between content words are identified. The tools of analysis are the dictionaries of basics / flexions / homonyms and statistical / syntactic word combinations, the removal of lexical homonymy, semantic analysis of nouns, the semantic-syntactic combination of nouns / adjectives and components of adverbials, algorithms for the analysis of algorithms ; system of division of words of the text on a flexion and basis; equivalence thesaurus for replacing equivalent words with one / more concept numbers that serve as content identifiers instead of word bases; a thesaurus in the form of a hierarchy of concepts to provide a search for a given general / associated concept; vocabulary service system. The indexing process depends on the descriptor dictionary or the information retrieval thesaurus. The descriptor dictionary has the structure of a table with three columns: the basics of words; sets of descriptors attributed to each basis; grammatical features of descriptors. Indexing consists of highlighting informative phrases from text; decoding the abbreviation; replacement of words with basic descriptors with the descriptor code; withdrawal of homonymy.

5 Conclusions

The article discusses known methods and approaches to addressing the automatic processing of textual content and highlights the shortcomings and benefits of existing approaches and results in the syntactic aspects of computational linguistics. Generalized conceptual principles of modeling of word-exchange processes in the formation of text arrays on the example of Ukrainian and German sentences, and then, proposing syntactic models and word-classifications of the lexical composition of Ukrainian and German sentences, developed lexicographic rules of syntactic type for automated processing. The application of the technique allows achieving higher reliability indicators in comparison with the known analogues, as well as demonstrating high efficiency in applied applications in the construction of new information technologies of lexicography and the study of the word-exchange effects of natural languages. The work is of practical value, since the proposed models and rules make it possible to effectively organize the process of creating lexicographic systems for processing syntactic textual content.

References

1. Angliyskaya grammatika v dostupnom izlozhenii. In: http://real-english.ru/crash/lesson3.htm, last accessed 2019/11/21.
2. Anisimov, A.V., Marchenko, O.O., Nykonenko, A.O.: Alhorytmichna model′ asotsiatyv-no-semantychnoho kontekstnoho analizu tekstiv pryrodnoyu movoyu. In: Probl. Prohramuv, 2-3, 379-384. (2008)
3. Anisimov, A.V.: Komp'yuternaya lingvistika dlya vsekh: mify, algoritmy, yazyk. Dumka, Kyiv, Ukraine (1991)
4. Apresyan, Y.D.: Idei i metody sovremennoy strukturnoy lingvistiki. Prosveshcheniye, Moscow (1966)
5. Apresyan, Y.D.: Neposredstvenno sostavlyayushchikh metod. In: http://tapemark.narod.ru/les/332a.html, last accessed 2019/11/21.
6. Arsent'yeva, N.G.: O dvukh sposobakh porozhdeniya predlozheniy russkogo yazyka. In: Problemy kibernetiki, 14, 189-218. (1965)
7. Bahmut, A.Y.: Poryadok sliv. In: Ukrayins′ka mova: Entsykl, 3, 675-676. (2007)
8. Bil'gayeva, N.T.: Teoriya algoritmov, formal'nykh yazykov, grammatik i avtomatov. VSGTU, Ulan-Ude (2000)

9. Bolshakova, Y.I., Klyshinskiy, E.S., Lande, D.V., Noskov, A.A., Peskova, O.V., Yagunova, Ye.V.: Avtomaticheskaya obrabotka tekstov na yestestvennom yazyke i komp'yuternaya lingvistika. MIEM, Moscow (2011)
10. 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)
11. Volkova, I.A., Rudenko ,T.V.: Formal'nyye grammatiki i yazyki. Elementy teorii translyatsii. In: Izdatel'skiy otdel fakul'teta vychisli-tel'noy matematiki i kibernetiki MGU im. M.V.Lomonosova, (1999)
12. Hakman, O.V.: Heneratyvno-transformatsiyna linhvistyka N. Khoms′koho yak vyrazhennya yoho linhvistychnoyi filosofiyi. In: Mul′tyversum. Filosofs′kyy al′ma-nakh, 45, 98-114. (2005)
13. Gerasimov, A.S.: Lektsii po teorii formal'nykh yazykov. In: http://gas-teach.narod.ru/au/tfl/tfl01.pdf, last accessed 2019/11/21.
14. Gladkiy, A.V.: Sintaksicheskiye struktury yestestvennogo yazyka v avtomatizirovan-nykh sistemakh obshcheniya. Nauka, Moscow (1985)
15. Gladkiy, A.V. Mel'chuk, I. A.: Elementy matematicheskoy lingvistiki. In: Nauka, Moscow (1969)
16. Gladkiy, A.V.: Formal'nyye grammatiki i yazyki. In: Nauka, Moscow (1973)
17. Gross, M., Lanten, A.: Teoriya formal'nykh Grammatik. Mir, Moscow (1971)
18. Darchuk, N.P.: Komp'yuterna linhvistyka (avtomatychne opratsyuvannya tekstu). In: Kyyivs'kyy universytet VPTS, Kyiv, Ukraine (2008)
19. Demeshko, I.: Typolohiya morfonolohichnykh modeley u viddiyeslivnomu slovotvorenni suchasnoyi ukrayins'koyi movy. In: Zbirnyk naukovykh prats' Linhvistychni studiyi, 19, 162-167. (2009)
20. Zubkov, M.: Ukrayins'ka mova: Universal'nyy dovidnyk. Shkola, Kyiv, Ukraine (2004)
21. Ingve. V.: Gipoteza glubiny. In: Novoye v lingvistike, IV, 126-138. (1965)
22. Lyubchenko, T.P.: Leksykohrafichni systemy hramatychnoho typu ta yikh zastosuvannya v za-sobakh avtomatyzovanoho opratsyuvannya movy. In: Avtoref. dys. kand. tekhn. nauk: spets. 10.02.21, Kyiv, Ukraine (2011)
23. Martynenko, B.K.: Yazyki i translyatsii: Ucheb. Posobiye. In: Izd. 2-ye, ispr. i dop. – SPb.: Izd-vo S.-Peterb. un-ta, (2008)
24. Marchenko, O.O.: Alhorytmy semantychnoho analizu pryrodnomovnykh tekstiv. In: Avtoref. dys. na zdobuttya nauk. stupenya kand. fiz.-mat. nauk: spets. 01.05.01. Kyiv, Ukraine (2005)
25. Noskov, S.A.: Samouchitel' nemetskogo yazyka. Nauka, Kyiv, Ukraine (1999)
26. Paducheva, Y.V.: O svyazyakh glubiny po Ingve so strukturoy dereva pochineniy. In: Nauchno-tekhnicheskaya informatsiya, 6, 38-43. (1967)
27. Partyko, Z.V.: Prykladna i komp'yuterna linhvistyka. In: Afisha, Lviv, Ukraine (2008)
28. Pentus, A.Y., Pentus, M.R.: Teoriya formal'nykh yazykov: Uchebnoye posobiye. In: Izd-vo TSPI pri mekhaniko-matematicheskom f-te MGU, Moscow (2004)
29. Popov, E.V.: Obshcheniye s EVM na yestestvennom yazyke. Nauka, Moscow (1982)
30. Postnikova, O.M.: Nimets′ka mova. Rozmovni temy: leksyka, teksty, dialohy, vpravy. T. 1, A.S.K, Kyiv, Ukraine (2001)
31. Postnikova, O.M.: Nimets′ka mova. Rozmovni temy: leksyka, teksty, dialohy, vpravy. T. 2, A.S.K, Kyiv, Ukraine (2001)
32. Potapova, H.M.: Morfonolohiya viddiyeslivnoho slovotvorennya (na materiali slovotvirnykh hnizd z vershynamy - diyeslovamy ta viddiyeslivnykh slovotvirnykh zon). In: Dys. kand. nauk: 10.02.02, Ukraine (2008).

33. Rusachenko, N.P.: Morfonolohichni protsesy u slovozmini ta slovotvori staroukrayins’koyi movy druhoyi polovyny XVI – XVIII st. In: Avtoreferat dysertatsiyi na zdobuttya naukovoho stupenya kandydata filolohichnykh nauk, http://auteur.corneille-moliere.com/?p=history&m=corneille_moliere&l=rus, last accessed 2019/11/21.
34. Torosyan, O.M.: Funktsional′ni kharakterystyky pryslivnykiv miry ta stupenya v suchasniy anhliys′kiy movi. In: avtoref. dys. na zdobuttya nauk. stupenya kand. filol. na-uk, http://disser.com.ua/contents/6712.html, last accessed 2019/11/21.
35. Turysheva, O.O.: Porushennya ramkovoyi konstruktsiyi v suchasniy nimets′kiy movi: funktsional′nyy aspekt, normatyvnyy status. In: Avtoref. dys. kand. filol. nauk: spets. 10.02.04 (2012)
36. Ukrayins’kyy pravopys. In: In-t movoznavstva im. O.O. Potebni NAN Ukrayiny, In-t ukr. movy NAN Ukrayiny, Nauk. dumka, Kyiv, Ukraine (2007)
37. Fomichev, V.S.: Formal'nyye yazyki, grammatiki i avtomaty. In: http://www.proklondike.com/books/thproch/, last accessed 2019/11/21.
38. Chomsky, N.: O nekotornykh formal'nykh svoystvakh grammatik. In: Kiberneticheskiy sbornik, 5, 279-311. (1962)
39. Chomsky, N., Miller, G. A.: Formal'nyy analiz yestestvennykh yazykov. In: Kibernetich-eskiy sbornik, 1, 231-290. (1965)
40. Chomsky, N.: Yazyk i myshleniye. In:Publikatsii OSiPL. Seriya monografiy, 2. (1972)
41. Chomsky, N.: Sintaksicheskiye struktury. In: Sbornik Novoye v lingvistike, 2, 412-527. (1962)
42. Chepurna, Z.V.: Transformatsiya poryadku sliv u prostomu rechenni pry perekladi z nimets′koyi movy ukrayins′koyu. In: Naukovi zapysky, 89(1), 232-236. (2010)
43. Sharov, S.A.: Sredstva komp'yuternogo predstavleniya lingvisticheskoy informatsii. In: http://www.ksu.ru/eng/science/ittc/vol000/002/, last accessed 2019/11/21.
44. 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)
45. Shul′zhuk, K.: Syntaksys ukrayins′koyi movy. In: Akademiya, Kyiv, Ukraine (2004)
46. Babichev, S.: An Evaluation of the Information Technology of Gene Expression Profiles Processing Stability for Different Levels of Noise Components. In: Data, 3 (4), art. no. 48 doi: 10.3390/data3040048 (2018)
47. Babichev, S., Durnyak, B., Pikh, I., Senkivskyy, V.: An Evaluation of the Objective Clustering Inductive Technology Effectiveness Implemented Using Density-Based and Agglomerative Hierarchical Clustering Algorithms. In: Advances in Intelligent Systems and Computing, 1020, 532-553 doi: 10.1007/978-3-030-26474-1_37 (2020)
48. Shreyder, Y.A.: Kharakteristiki slozhnosti struktury teksta. In: Nauchnotekhnicheskaya informatsiya, 7, 34-41. (1966)
49. Chomsky, N.: Three models for the description of language. In: I. R. E. Trans. PGIT 2, 113-124. (1956)
50. Chomsky, N.: On certain formal properties of grammars, Information and Control 2. In: A note on phrase structure grammars, Information and Control 2, 137-267, 393-395. (1959)
51. Chomsky, N.: On the notion «Rule of Grammar». In: Proc. Symp. Applied Math., 12. Amer. Math. Soc. (1961)
52. Chomsky, N.: Context-free grammars and pushdown storage. In: Quarterly Progress Reports, 65, Research Laboratory of Electronics, M. I. T. (1962)
53. Chomsky, N.: Formal properties of grammars. In: Handbook of Mathemati-Mathematical Psychology, 2, ch. 12, Wiley, 323-418. (1963)
54. Chomsky, N.: The logical basis for linguistic theory. In: Int. Cong. Linguists, (1962)

55. Chomsky, N., Miller, G. A.: Finite state languages. In: Information and Control 1, 91-112. (1958)
56. Chomsky, N., Miller, G. A.: Introduction to the formal analysis of natural languages. In: Handbook of Mathematical Psychology 2, ch. 12, Wiley, 269-322. (1963)
57. Chomsky, N., Schützenberger M.P.: The algebraic theory of context-free languages. In: Computer programming and formal systems, North-Holland, Amsterdam, 118–161 (1963)
58. Bar-Hillel, Y., Shamir, E.: Finite state languages: formal representation and adequacy problems. In: Bulletin of the Research Council of Israel, 8F(3), 155-166. (1960)
59. Bobrow, D.G.: Syntactic analysis of English by computer – a survey. In: AFIPS conference proceedings, 24, Baltimore, London, 365-387. (1963)
60. English Verbs (Part 1) – Basic Terms. In: http://sites.google.com/site/englishgrammarguide/Home/english-verbs--part-1----basic-terms, accessed 2019/11/21.
61. Hays, D.G.: Automatic language data processing. In: Computer applications in behavioral sciences, Englewood Cliffs, 394-421. (1962)
62. Postal, P.M.: Limitations of phrase structure grammars. In: The structure of language. Readings in the philosophy of language, Englewood Cliffs, 137-151. (1964)
63. Tesniere, L.: Elements de syntaxe structurale. (1959)
64. Tosh, L.W.: Syntactic translation. The Hague (1965)
65. Yngve, V.H.: A model and a hypothesis for language structure. In: Proceedings of American phylosophical society, 104(5), 444-466. (1960)
66. Yngve, V.H.: Random generation of English sentences. In: Teddington (National physical laboratory, paper 6 (1961)
67. 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)
68. Varga, D.: Yngve's hypothesis and some problems of the mechanical analysis. In: Computational Linguistics, III, 47-74. (1964)
69. 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)
70. 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, doi: 10.1007/978-3-030-01069-0_8 (2019)
71. 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, Eastern-European Journal of Enterprise Technologies, 5(2), 16-28 (2018)
72. Vysotska, V., Lytvyn, V., Hrendus, M., Kubinska, S., Brodyak, O.: Method of textual information authorship analysis based on stylometry, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018, 9-16 (2018)
73. 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, Eastern-European Journal of Enterprise Technologies, 6(2-96), pp. 19-31 (2018)
74. Vysotska, V., Burov, Y., Lytvyn, V., Demchuk, A.: Defining Author's Style for Plagiarism Detection in Academic Environment, Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, 128-133 (2018)

75. Vysotska, V., Fernandes, V.B., Lytvyn, V., Emmerich, M., Hrendus, M.: Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship, Advances in Intelligent Systems and Computing, 871, 132-151 (2019)
76. Lytvyn, V., Vysotska, V., Burov, Y., Bobyk, I., Ohirko, O.: The linguometric approach for co-authoring author's style definition, Proceedings of the 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2018, 29-34 (2018)
77. Vysotska, V., Kanishcheva, O., Hlavcheva, Y.: Authorship Identification of the Scientific Text in Ukrainian with Using the Lingvometry Methods, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 2, 34-38 (2018)
78. 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: 2019 IEEE 14th International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT), 84-91. (2019)
79. 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: 2019 IEEE 14th International Scientific and Technical Conference on Computer Science and Information Nechnologies (CSIT'2019), 1-6. (2019)
80. Cherednichenko, O., Babkova, N., Kanishcheva, O.: Complex Term Identification for Ukrainian Medical Texts. In: CEUR Workshop Proceedings, Vol-2255, 146-154. (2018)
81. Bobicev, V., Kanishcheva, O., Cherednichenko, O.: Sentiment Analysis in the Ukrainian and Russian News. In: First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 1050-1055. (2017)
82. Fedushko, S., Benova, E.: Semantic analysis for information and communication threats detection of online service users. In: The 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN), 160, 254-259. (2019)
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. Chyrun, L., Chyrun, L., Kis, Y., Rybak, L.: Automated Information System for Connection to the Access Point with Encryption WPA2 Enterprise. In: Lecture Notes in Computational Intelligence and Decision Making, 1020, 389-404. (2020)
85. 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)
86. 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)
87. Chyrun, L., Burov, Y., Rusyn, B., Pohreliuk, L., Oleshek, O., Gozhyj, A., Bobyk, I.: Web Resource Changes Monitoring System Development. In: CEUR Workshop Proceedings, Vol-2386, 255-273. (2019)
88. 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)
89. 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)