UDC 004.519.7 (045)

A. Vavilenkova
National Aviation University, Kiev, str. Komarova,1

APPLYING OF FORMAL ALGORITHMS IN STRUCTURAL LINGUISTICS

The article analyzes the concept of formal algorithms; it demonstrates an example of using the commands of Turing machine to analyze natural language sentence. The research proposes the algorithm of automated linguistic processing of electronic documents based on constructing logico-linguistic models of sentences.

Keywords – natural language, formal algorithms, semantic structure, logico-linguistic model, text base.

A problem statement

The main subject of research of structural linguistics is the structure of natural language, which is a network of relationships between the elements of the language system and which are arranged in a hierarchical dependence within certain levels. Structural description language provides a textual analysis that will identify generalized invariant units and match them with specific speech segments based on clear rules of realization [1].

Today, structural linguistics has been developing accurate methods for studying natural languages, using a mathematical approach. In particular, special attention should be paid to create such formal algorithms that would analyze text documents by content.

To form an idea about the structure of the text it is possible by making its linguistic analysis, in which the text base is built and it includes semantic representation of the text, and as interrelated proposals it captures what was said in the text. The test base should meet the criteria for local and global connectivity. The first involves the fact that the proposals found in the text must be linked. Global connectivity also provides understanding and identifying the main theme of the text and the role of each phrase within it in terms of disclosure the main theme.

Automated structural analysis of texts can only be done by applying formal rules identify relationships between phrase. This, in turn, is possible by identifying and systematizing the transformations that occur within the underlying lexical and syntactic transformations [2].

Thus, the immediate task is the task to formalize the process of identifying the underlying relationships between simple and complex parts of texts to extract knowledge from text data.

Analysis of recent research and publications

Over the ability to extract content of textual information today there are large number of foreign and domestic scientists work. The process of extracting knowledge from text documents is not possible without the involvement of basic theory of texts and mathematical tools that will help formalize and automate this process (actually what structural linguistics is dealing with).

So, in Philology sphere there are new works of the theory of text organization. For example, the work of Zemska Y.N. «Theory of text» summarizes existing hypotheses of present structural, semantic and communicative-pragmatic organization of texts [3]. J. Layonz [4] finds a match between form, structure and components of the lexical meaning of the words and sentences of natural language. There are also attempts to formalize the structure of texts made in the Filippov's [5] work, where the author outlines the distinct ways of organizing content and meaningful relationships, but without the use of mathematical approaches.

A huge contribution to the automation of process of understanding of electronic documents has become lexicographical dictionary Ukrainian language [6]. It is a thesaurus containing lexical relationships between words, provides comprehensive information on the interpretation of words and their synonyms, grammatical forms, phraseology, etc. The role of conceptual relations in the texts by using lexical networks highlight the authors of «Cognitive linguistics: basic readings» [7]. Special attention to the technology of automatic text analysis, based on linguistic models, is paid within the international conference «Dialogue» [8], where new methods of verification and evaluation of the results of linguistic research and analysis of various texts are tested. The problems of qualitative formal algorithms of analytical processing of text information are raised on the following annual conferences and symposia as International Multiconference on Computer Science and Information, International Conference «Knowledge – Dialogue – Solution» and others.

Despite numerous approaches to linguistic analysis of texts and mass of formal algorithms structuring textual information, the content processing of electronic documents processing is not carried out perfectly and all the theoretical studies in this area are under development.

The formulation of the purpose of Article

Natural language sentences consist of syntactic units which are numerous but finite. A random language system has a finite dictionary of simple units, and a finite set of rules that establish the relationship between two structural levels and reveal which sequence of units are sentences and which are not. So further research is intended for creation and application the formal rules of identification lexical relationships between units of text materials.

Purpose of this article is to analyze the existing formal algorithms of textual information processing and the creation of new algorithms that would be able to process textual information at the semantic level.

The main material

For research in field of automated analytical processing of text information it is important that arbitrary program of calculation function, the domain of definition of which is formed by sentences of natural language, has to be a program that simulated an automaton, no less powerful than a Turing machine. After all, formal algorithms known today primarily include Turing machine, Post machine, normal Markov algorithms, recursive functions and stochastic algorithms. If we consider the text analyzer as a machine with a determined number of states, the process of its work can be divided into:

- operation analysis – performs the function of converting natural language text from informal form into the formalized internal representation;
- analysis – data transformation functions that exist in the internal representation and output on the basis of it a new data and also in a formalized manner;
- synthesis – function of formation of answers on natural language, adequate to internal formalized representation [9].

Try the example of a complex natural language sentence to prove the impossibility of the existing formal algorithms to extract knowledge from text data.

Turing machine – is a ordered group of five:

$$M = \langle A, Q, P, q_1, q_0 \rangle,$$

де $A = \{a_0, a_1, ..., a_n\}$ – finite external alphabet, always assume that $n \geq 1, a_0 = 0, a_1 = 1$.

$Q = \{q_0, q_1, ..., q_m\}$ – finite alphabet of internal states;

$P = T\{i, j),0 \le i \le m,0 \le j \le n$ – an application, consists of instructions $T(i, j)$, each of which is a word kind of: $q_i a_j \to a_l d q_k$ ($0 \le k \le m,0 \le l \le n$), де q_i – internal state of the machine; a_j – character ,which is read out; q_k – new internal state; a_l –a new character, which is written down; d – direction of reading device, L – left, R – right, E – on the spot.
q_0 – initial state;
q_m – finite state.

It is predicted that each pair $q_i a_j$ corresponds to exactly one instruction. The set of instructions is called machine program, which means the program has $m \times (n+1)$ instructions.

So, the use of formal algorithm - Turing machine, to analyze sentences of natural language means, than it should be developed such set of instructions that would allow for a finite number of steps to read out the set of words of sentence in the same order in which they are taken in the sentence. This means that for the sentence that has different structure, type, or even the number of words you need to create a new algorithm.

For example, suppose there is a natural language sentence: «*In cases, when the current legislation provides for the possibility to carry out customs procedures without filing Customs Declaration, legislation applies*». To read out this sentence it is proposed the following set of instructions:

T(0,p)=(0,П,1);
T(1,)=(0,П,2);
T(2,n)=(0,П,1);
T(1,,)=(0,П,1);
T(2,c)=(0,П,1);
T(2,adj)=(0,П,1);
T(2,v)=(0,П,1);
T(2,vadj)=(0,П,1).

Here, the words are replaced by the first letters of parts of speech, they are used to be.

Zero state is considered when the marker is located over the first word of the sentence of natural language. The sentence is considered to be read out if the marker is over the last word.

Process of program check and the current results for this example are shown in Table. 1.

Table 1

The results of the Turing machine

№ state	Current character string	Instruction	The resulting character string
0	***In** cases, when the current legislation provides for the…*	T(0,p)=(0,П,1)	*In_ cases, when the current legislation provides for the…*
1	*In_ cases, when the current legislation provides for the…*	T(1,)=(0,П,2)	*In **cases**, when the current legislation provides for the…*
2	*In **cases**, when the current legislation provides for the…*	T(2,n)=(0,П,1)	*In cases**,** when the current legislation provides for the…*
1	*In cases**,** when the current legislation provides for the…*	T(1,,)=(0,П,1)	*In cases,_ when the current legislation provides for the…*
1	*In cases,_when the current legislation provides for the…*	T(1,)=(0,П,2)	*In cases, **when** the current legislation provides for the…*
2	*In cases, **when** the current legislation provides for the…*	T(2,c)=(0,П,1)	*In cases, when_ the current legislation provides for the…*
1	*In cases, when_ the current legislation provides for the…*	T(1,)=(0,П,2)	*In cases, when **the** current legislation provides for the…*
2	*In cases, when **the** current legislation provides for the…*	T(0,p)=(0,П,1)	*In cases, when the _current legislation provides for the…*
1	*In cases, when the _current legislation provides for the…*	T(1,)=(0,П,2)	*In cases, when the **current** legislation provides for the…*
2	*In cases, when the **current** legislation provides for the…*	T(2,adj)=(0,П,1)	*In cases, when the current_ legislation provides for the…*
1	*In cases, when the current_ legislation provides for the…*	T(1,)=(0,П,2)	*In cases, when the current **legislation** provides for the…*

2	*In cases, when the current **legislation** provides for the…*	T(2,n)=(0,П,1)	*In cases, when the current legislation_ provides for the…*
1	*In cases, when the current legislation_ provides for the…*	T(1,)=(0,П,2)	*In cases, when the current legislation **provides** for the…*
2	*In cases, when the current legislation **provides** for the…*	T(2,v)=(0,П,1)	*In cases, when the current legislation provides_ for the…*
№ state	Current character string	Instruction	The resulting character string
1	*In cases, when the current legislation provides_ for the…*	T(1,)=(0,П,2)	*In cases, when the current legislation provides **for** the…*
2	*…**for** the possibility to carry out customs procedures without…*	T(2,c)=(0,П,1)	*…for_ the possibility to carry out customs procedures without…*
1	*…for_ the possibility to carry out customs procedures without…*	T(1,)=(0,П,2)	*…for **the** possibility to carry out customs procedures without…*
2	*…for **the** possibility to carry out customs procedures without…*	T(2,c)=(0,П,1)	*…for the_ possibility to carry out customs procedures without…*
1	*…for the_ possibility to carry out customs procedures without…*	T(1,)=(0,П,2)	*…for the **possibility** to carry out customs procedures without…*
2	*…for the **possibility** to carry out customs procedures without…*	T(2,n)=(0,П,1)	*…for the possibility_ to carry out customs procedures without…*
1	*…for the possibility_ to carry out customs procedures without…*	T(1,)=(0,П,2)	*…for the possibility **to** carry out customs procedures without…*
2	*…for the possibility **to** carry out customs procedures without…*	T(2,c)=(0,П,1)	*…for the possibility to_ carry out customs procedures without…*
1	*…for the possibility to_ carry out customs procedures without…*	T(1,)=(0,П,2)	*…for the possibility to **carry** out customs procedures without…*
2	*…for the possibility to **carry** out customs procedures without…*	T(2,v)=(0,П,1)	*…for the possibility to carry_ out customs procedures without…*
1	*…for the possibility to carry_ out customs procedures without…*	T(1,)=(0,П,2)	*…for the possibility to carry **out** customs procedures without…*
2	*…for the possibility to carry **out** customs procedures without…*	T(2,p)=(0,П,1)	*…for the possibility to carry out_ customs procedures without…*
1	*…for the possibility to carry out_ customs procedures without…*	T(1,)=(0,П,2)	*…for the possibility to carry out **customs** procedures without…*
2	*…for the possibility to carry out **customs** procedures without…*	T(2,n)=(0,П,1)	*…for the possibility to carry out customs_ procedures without…*
1	*…for the possibility to carry out customs_ procedures without…*	T(1,)=(0,П,2)	*…for the possibility to carry out customs **procedures** without…*
2	*…for the possibility to carry out customs **procedures** without…*	T(2,n)=(0,П,1)	*…for the possibility to carry out customs procedures_ without…*
1	*…for the possibility to carry out customs procedures_ without…*	T(1,)=(0,П,2)	*…for the possibility to carry out customs procedures **without**…*
2	*…for the possibility to carry out customs procedures **without**…*	T(2,p)=(0,П,1)	*…without_ filing Customs Declaration, legislation applies*
1	*…without_ filing Customs Declaration, legislation applies*	T(1,)=(0,П,2)	*…without **filing** Customs Declaration, legislation applies*
2	*…without **filing** Customs Declaration, legislation applies*	T(2,vadj)=(0,П,1)	*…without filing_ Customs Declaration, legislation applies*
1	*…without filing_ Customs Declaration, legislation applies*	T(1,)=(0,П,2)	*…without filing **Customs** Declaration, legislation applies*
2	*…without filing **Customs** Declaration, legislation applies*	T(2,n)=(0,П,1)	*…without filing Customs_ Declaration, legislation applies*
1	*…without filing Customs_ Declaration, legislation applies*	T(1,)=(0,П,2)	*…without filing Customs **Declaration**, legislation applies*
2	*…without filing Customs **Declaration**, legislation applies*	T(2,n)=(0,П,1)	*…without filing Customs Declaration**,** legislation applies*
1	*…without filing Customs Declaration**,** legislation applies*	T(1,,)=(0,П,1)	*…without filing Customs Declaration,_ legislation applies*
1	*…without filing Customs Declaration,_ legislation applies*	T(1,)=(0,П,2)	*…without filing Customs Declaration, **legislation** applies*

2	*...without filing Customs Declaration,* ***legislation*** *applies*	T(2,n)=(0,П,1)	*...without filing Customs Declaration, legislation_ applies*
1	*...without filing Customs Declaration, legislation_ applies*	T(1,)=(0,П,2)	*...without filing Customs Declaration, legislation* ***applies***

A given set of instructions allows us to pass from the first to the last word of the sentence, so the algorithm is correct. It is not difficult to see that the changes in the structure of the sentence, including the use of punctuation will result in writing additional commands in the instruction. Also reading out natural language sentences using Turing machine leads to time-consuming. In addition, it makes impossible to analyze the sentence by content.

Post machine consists of infinite sting divided by the equal cells that can either be empty – 0 or include the label 1, and reading device (carriage) that can move around on string, check label, erase and record label.

State of Post machine is described by string state and carriage position. The carriage is operated by a program that consists of instructions, each of which has the following syntax: iKj where i – number of instruction, K – the action of the carriage, j – number of next instruction. There are six types of instructions for Post machine:

V_j – to put a label and go to j -string of program ;

X_j – to erase label and go to j -string of program;

$<-j$ – to move left and go to j -string of program ;

$->j$ – to move right and go to j -string of program;

$?\, j_1; j_2$ – if the cell has no tags, then go to j_1-string of program, else go to j_2 string of program;

! – completion of program.

Using Post machine to analyze natural language sentence gives a result similar to the Turing machine.

Formal algorithms also include ***Markov's normal algorithms***, which is a system of consecutive applying of substitutions that implement specific procedures for obtaining new words from basic, built from some of the symbols of the alphabet. Normal algorithms are verbal, that is assigned to apply to words in different alphabets. Determination of normal random algorithm consists of two parts: the definition of the alphabet algorithm and determination of its schema. Simple formula substitutions are called words like $L \to D$, where L and D – two arbitrary words in the alphabet algorithm (respectively left and right side of the formula). $L \to \cdot D$ – the final formula.

Existing formal algorithms are focused on the accurate execution of instructions on a pre-defined pattern. Natural language texts require more flexible algorithms that will work with sentences of different complexity, and will allow them to analyze the content and extract knowledge from texts.

Algorithm that uses formal models for the structural analysis of the text is algorithm of automated linguistic processing of electronic documents based on constructing logico-linguistic models of sentences.

The whole electronic document can be represented as a set of logico-linguistic models of natural language sentences:

$$t = \bigwedge_{g=1}^{N(t)} L_g(S_g), \qquad (1)$$

where $L_g(S_g)$ – logico-linguistic model of sentence S_g, $g = \overline{1, N(t)}$;

$N(t)$ – number of sentences in the text t.

Logico-linguistic model of sentence is [monograph]:

$$L(S) = \bigwedge_{\mu=1}^{\nu(S)} L_\mu(S), \qquad (2)$$

where $L_\mu(S)$ – simple predicate that describes the part of the sentence S which represents the complete content;

$\mu=\overline{1,\ \nu(S)}$, $\nu(S)$ – number of parts of sentence S that represents the completed content.

Algorithm of automated linguistic processing of electronic documents based on forming logico-linguistic models of sentences is performed by the following steps.

1. Building a schematic structure of the document, ie splitting the text into sections, parts, paragraphs, sentences. At this stage, it is forming the set of complex parts of electronic document $F=\{f_1,...,f_j,...,f_m\}$, where $j=\overline{1,m}$, m – number of complex syntactic units, and set of paragraphs $A=\{a_1,...,a_k,...,a_q\}$ – set of paragraphs of text, $k=\overline{1,q}$, q – the number of paragraphs.

2. Formation of stable word combinations (phrases). This stage is reduced to sequential testing of necessary conditions presented in the form of formal rules of formation of natural language phrases [10]. If a rule is executed, then a node forms, for which all syntactic features are initialized; structural parts of speech which were related to words, which are included in the phrase become attributes of nodes.

3. Construction of logico-linguistic model (LLM) of natural language sentences $L(S)$ that appertain to the electronic document. At this stage, every sentence of natural language is associated with a logical formula [11].

4. Establishing relationships between nodes. It is performing a sequential search of semantic relationships between LLMs, implemented through the consecutive applying of formal rules, using means of cohesion in natural language texts. These are techniques like semantic repetition, discursive words, ellipsis, syntactic parallelism, anaphoric links etc. The result of this phase of the algorithm is the identification and unification in LLM sentences of all text synonyms, common root words and derived words, pronouns restoration and transformation of linking words in corresponding logic operations.

5. Creating corpus linguistic of the document - provides analysis of used formal rules by means of cohesion, logical relationships between paragraphs of text, establishing the type of thematic progressions and rem taken in the text. All the data are compared with data entries that contain pre-registered grammatical and syntactic features of texts of different styles. As a result of this analysis in accordance with each text, you can put a set of logical LLMs that represent semantic relationships and cortege linguistic characteristics of the text:

$$t'=<C,F,B,A>, \qquad (3)$$

where $t'\in T$ -specific electronic text from the entire set of texts;

$C=\{c_1,...,c_i,...,c_n\}$ – set of existing types of texts, $i=\overline{1,n}$, n – number of types;

$F=\{f_1,...,f_j,...,f_m\}$ – set of complex syntactic parts of the text, $j=\overline{1,m}$, m – number of complex syntactic units;

B – text database consists of a set of keywords and associated text suggestions;

$A=\{a_1,...,a_k,...,a_q\}$ – set of paragraphs, $k=\overline{1,q}$, q – number of paragraphs.

Let's analyze the following piece of text: «*Carrying out customs control and customs clearance of goods and vehicles crossing the customs border of Ukraine, it is applying only legal acts at the date of acceptance of the customs declaration by the customs authorities of Ukraine.*

In cases, when the current legislation provides for the possibility to carry out customs procedures without filing customs declaration, it is applied legislation which is current at the date such procedures are being executed».

1. We construct a schematic structure of the document. The set of complex parts of the text consists of one element $F=\{f_1\}$, $m=1$, and the set of paragraphs consists of two elements $A=\{a_1,a_2\}$, $q=2$.

It is purely technical stage that does not set any types of relationships between the structural elements of the text.

2. Formation of stable phrases:
- *customs control;*
- *customs clearance of goods;*
- *clearance of vehicles;*
- *customs border of Ukraine;*
- *legal acts;*
- *current legislation;*
- *customs procedures;*
- *filing declaration.*

3. Construction of logico-linguistic models of sentences that are in the fragment of the text.

L_1= Applying (acts, legal, only, carrying out, control, customs),
L_2= Applying (acts, legal, only, carrying out, clearance, customs, goods),
L_3= Applying (acts, legal, only, carrying out, clearance, customs, vehicles),
L_4= Crossing (vehicles, border, customs, Ukraine),
L_5= Provides (possibility, carry out, procedures, customs, filing, declaration, legislation, current),
L_6= Applied (legislation, current, date, executed, procedures, such, cases, filing, declaration).

$$L_1 = P_{[1]}(x_{1[1]}, c_{11[1]}, x_{2[1]}, x_{3[1]}, x_{4[1]}, c_{41[1]}),$$
$$L_2 = P'_{[1]}(x'_{1[1]}, c'_{11[1]}, x'_{2[1]}, x'_{3[1]}, x'_{4[1]}, c'_{41[1]}, x'_{5[1]}),$$
$$L_3 = P''_{[1]}(x''_{1[1]}, c''_{11[1]}, x''_{2[1]}, x''_{3[1]}, x''_{4[1]}, c''_{41[1]}, x''_{5[1]}),$$
$$L_4 = P'''_{[1]}(x'''_{1[1]}, x'''_{2[1]}, c'''_{21[1]}, x'''_{3[1]}).$$
$$L_5 = P_{[2]}(x_{1[2]}, x_{2[2]}, x_{3[2]}, c_{31[2]}, x_{4[2]}, x_{5[2]}, x_{6[2]} c_{61[2]}),$$
$$L_6 = P'_{[2]}(x'_{1[2]}, c'_{11[2]}, x'_{2[2]}, x'_{3[2]}, x'_{4[2]}, c'_{41[2]}, x'_{5[2]}, x'_{6[2]}, x'_{7[2]}).$$

4. Establishing relationships between nodes means the replacement of structural components of the LLMs sentences.

$$L_1 = P_{[1]}(x_{1[1]}, c_{11[1]}, x_{2[1]}, x_{3[1]}, x_{4[1]}, c_{41[1]}),$$
$$L_2 = P_{[1]}(x_{1[1]}, c_{11[1]}, x_{2[1]}, x_{3[1]}, x'_{4[1]}, c'_{41[1]}, x'_{5[1]}),$$
$$L_3 = P_{[1]}(x_{1[1]}, c_{11[1]}, x_{2[1]}, x_{3[1]}, x'_{4[1]}, c'_{41[1]}, x''_{5[1]},$$
$$L_4 = P'''_{[1]}(x''_{5[1]}, x'''_{2[1]}, c_{41[1]}, x'''_{3[1]}).$$
$$L_5 = P_{[2]}(x_{1[2]}, x_{2[2]}, x_{3[2]}, c_{41[1]}, x_{4[2]}, x_{5[2]}, x_{6[2]} c_{61[2]}),$$
$$L_6 = P'_{[2]}(x_{6[2]}, c_{61[2]}, x'_{2[2]}, x_{3[1]}, x_{3[2]}, c'_{41[2]}, x'_{5[2]}, x_{4[2]}, x_{5[2]}).$$

5. Creation corpus linguistic of fragment, its formal record corresponds to cortege; $t' = < \{4\}, \{1\}, B, \{a_1, a_2\} >$ [10].

Conclusions

For automated processing of text data on the semantic level it is necessary to use formal algorithms that are able to process texts based on logical relationships at the level of sentences, and at the level of natural language phrases. Such algorithm is a proposed algorithm of automated linguistic processing electronic documents based on constructing logico-linguistic models of sentences. Corpus linguistics takes into account interphase relationships and grammar-syntactic features of a particular text, and an array of logical-linguistic models allows us to trace the relationship between natural language sentences. The main feature of this formal algorithm is applying formalized rules of using cohesion in natural language texts.

1. Виноградов В.А. Структурная лингвистика // Лингвистический энциклопедический словарь. – М.:1990. 2. Апресян Ю.Д. Лексическая семантика: в 2-х т. Т. 1. / Ю.Д. Апресян. – М.: «Восточная литература», 1995. – 422 с. 3.Земская Ю.Н. Теория текста: учебное пособие / Ю.Н. Земская, И.Ю. 4. Качесова, Л.М. Комиссарова, Н.В. Панченко, А.А.Чувакин. – М.: Наука, 2010. – 132 с.
УРСС, 2004. – 320 с. 5. Филиппов К.А. Лингвистика текста. Курс лекций. / К.А. Филиппов. – Спб.: Издательство С.-Петербургского университета, 2008. – 336с. 6. Український лінгвістичний портал, словники України/ http://lcorp.ulif.org.ua/dictua/. 7. Geeraerts Dirk. Cognitive linguistics: basic readings research / Dirk Geeraerts, Rene Dirven, John R. Taylor. – Berlin-New York: Mouton de cruyter, 2006. – 486 p. 8. Международная конференция по компьютерной лингвистике Диалог 2014 / http://www.dialog-21.ru/dialog2014/. 9. Шулкин Д.Е. Морфологический и синтаксический разбор текстов как конечный автомат, реализованный семантической нейронной сетью, имеющей структуру синхронизированного линейного дерева / Д.Е. Шулкин // Новые информационные технологии. – 2002. – С.74-85. 10. Вавіленкова А.І. Теоретичні основи аналізу електронних текстів: [монографія] / А.І. Вавіленкова, Д.В. Ланде, О.Є. Литвиненко. – К.: НАУ, 2014. – 250с.
11. Вавіленкова А.І. Логіко-лінгвістична модель як засіб відображення синтаксичних особливостей текстової інформації /А.І. Вавіленкова // Математичні машини та системи. – 2010. – № 2. – С. 134–137.