https://doi.org/10.31891/csit-2023-3-10
UDC 004.9

Viktor SHYNKARENKO, Inna DEMIDOVICH
Ukrainian State University of Science and Technologies

CONSTRUCTIVE-SYNTHESIZING MODELING OF NATURAL LANGUAGE TEXTS

Means for solving the problem of establishing the natural language texts authorship were developed. Theoretical tools consist of a constructors set was developed on the basis of structural and production modeling. These constructors are presented in this work. Some results of experimental studies based on this approach have been published in previous works by the author, the main results should be published in the next ones.

Constructors developed: converter of natural language text into tagged, tagged text into a formal stochastic grammar and the authors style similarity degree establishment of two natural language works based on the coincidence of the corresponding stochastic grammars (their substitution rules).

In this paper, constructors are developed and presented that model a natural language text in the form of a stochastic grammar that displays the structures of sentences in it. This approach allows you to highlight the syntactic features of the construction of phrases by the author, which is a characteristic of his speech. Working with a sentence as a unit of text for analyzing its construction will allow you to more accurately capture the author's style in terms of the words use, their sequences and speech style characteristic. It allows you not to be tied to specific parts of speech, but reveals the general logic of constructing phrases, which can be more informative in terms of the author's style characteristics for any text.

The presented work is a theoretical basis for solving the problems of the text authorship establishing and identifying borrowings. Experimental studies have also been carried out. The statistical similarity of solutions to the problems of establishing authorship and identifying borrowings was experimentally revealed, which will be presented in the next article of the authors.

The proposed approach makes it possible to highlight the semantic features of the author's phrases construction, which is a characteristic of his speech. Working with a sentence as a unit of text to analyze its construction will allow you to more accurately determine the author's style in terms of the use of words, their sequences and characteristic language constructions. Allows not to be attached to specific parts of speech, but reveals the general logic of building phrases.

It is planned to use the created model in the future to determine the authorship of natural language texts of various directions: fiction and technical literature.

Keywords: natural language texts, constructive-synthesizing modeling, establishing authorship, formal grammars, stochastic grammars, text models

Віктор ШИНКАРЕНКО, Інна ДЕМИДОВИЧ
Український державний університет науки та технологій

КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ МОДЕЛЮВАННЯ ПРИРОДНЬОМОВНИХ ТЕКСТІВ

Розроблені засоби для вирішення задачі встановлення авторства природньомовних текстів. Теоретичні засоби складаються з комплексу конструкторів розроблених на основі конструктивно-продукційного моделювання. Саме ці конструктори представлені в данній роботі. Деякі результати експериментальних досліджень основаних на цьому підході опубліковані в попередніх роботах авторів, основні результати мають бути опубліковані в наступних.

Розроблені конструктори: перетворювач природньомовного тексту на тегований, тегованого тексту у формальну стохастичну граматику та встановлення ступеню схожості стилю авторів двох природньомовних творів за збігом відповідних стохастичних граматик (їх правил підстановки).

У статті розроблено та представлено конструктори, які моделюють текст природною мовою у вигляді стохастичної граматики, що відображає структури речень у ньому. Такий підхід дозволяє виділити синтаксичні особливості побудови фраз автором, що є характеристикою його мовлення. Робота з реченням як одиницею тексту для аналізу його побудови дозволить точніше вловити стиль автора з точки зору вживання слів, їх послідовності та характеристики стилю мовлення. Він дозволяє не прив'язуватися до конкретних частин мови, а розкриває загальну логіку побудови фраз, що може бути більш інформативним з точки зору характеристики стилю автора для будь-якого тексту.

Представлена робота є теоретичним підґрунтям для вирішення проблем встановлення авторства тексту та ідентифікації запозичень. Також були проведені експериментальні дослідження. Експериментально виявлено статистичну схожість розв'язків задач встановлення авторства та ідентифікації запозичень, що буде представлено в наступній статті авторів.

Запропонований підхід дозволяє виділити семантичні особливості побудови фраз автором, що є характеристикою його мовлення. Робота з реченням, як із одиницею тексту для аналізу його побудови, дозволить більш точно визначити авторський стиль у частині використання слів, їх послідовностей і характерних мовних конструкцій. Дозволяє не прив'язуватись до конкретних частин мови, а виявляє загальну логіку побудови фраз.

Створену модель планується використовувати в подальшому для визначення авторства природномовних текстів різного спрямування: художньої та технічної літератури.

Ключові слова: природньомовні тексти, конструктивно-продукційне моделювання, встановлення авторства, формальні граматики, стохастические грамматики, моделі текстів

Introduction

The work develops an approach to the construction formalization proposed by the author by the constructive modeling means. This approach allows you to highlight the semantic features of the author's phrases construction, which is a characteristic of his speech. The use of the developed model is assumed in the field of determining the texts authorship and identifying borrowed ones.

This approach to text analysis is promising, due to the presence of each person's own style and approach to constructing phrases. People communication by means of natural language texts is carried out with the use of not individual words, but expressed, sentences. A sentence as a sequence of individual words meanings is a new unit with a set of semantic values inherent only to it, among which there are also those that are not considered direct derivatives of the existing sentence composition, which is due to the peculiarity of its construction.

Modeling speech communication, in its entirety transmitted with the help of information language, is impossible without studying the features of the sentence structure. This aspect study of syntactic structures is important, in addition to purely linguistic tasks, for understanding the features and regularities of a person's mental activity.

Related works

The main text studies and methods of its formalization mostly work with words or their short sequences, and only a small part of the studies are based on the sentence as a unit of the text.

Widespread methods that investigate the test based on symbols or their sequence of some length [7] or separately words [8] and their sequences [6], work with lemmas [9] or parts of speech [10]. However, such an approach does not always adequately reflect the peculiarities of the text under study and the author's style.

The approach to the sentence as a text unit opens up new opportunities for work, since the peculiarities of its construction and the analysis of the words used within one sentence can serve as an additional source of information about the author [13]. Similar approaches were used for context research [11], working with the author’s mood research [12], in terms of a deeper understanding of the text [14]. Many studies also testify to the importance of considering the sentence as a single unit, rather than a collection of individual words [15], the importance of their order [16] and the general context [17].

Currently, there are many approaches to building models proposed [18, 19]. The most popular methods are the use of trees [20] and the construction of various neural networks [23, 24]. However, the most universal tool has not yet been found [22].

Such approaches are most widespread in the field of work with artificial intelligence in terms of text recognition and understanding, which largely confirms the necessity and relevance of using a separate sentence as a unit of text structure.

The processes of texts authorship identifying using the constructive-production modeling

Generalized designer

Development of a constructive-synthesizing approach to solving the problem of technical text authorship establishing. The generalized constructor is a triple called C_G [2]

$$C_G = \langle M, \Sigma, \wedge \ \rangle,$$

where M – the non-homogeneous carrier of the structure, which expands during the construction process;Σ – is the operations and relations signature, that consisting of binding, substitution and derivation operations, operations on attributes, and a substitution relation; $\wedge$ – construction information support (CIS).

According to $\wedge$ [2] the ${}_w l$ form with attribute w is called a set of terminals and non-terminals that are united by binding operations. The developed constructors use a single binding operation (and relationship) - concatenation. A form that contains only terminals is called a construction. Constructions are formed by derivation from the initial non-terminal, substitution operations and operations on attributes, and generalized partial and full derivation operations.

The operation of partial derivation ($| \Rightarrow \in \Sigma_P$) consists of choosing a suitable substitution rule from their set, performing this substitution and performing operations on the attributes corresponding to the selected rule in a certain sequence.

The operation of complete derivation (or simply derivation, $|| \Rightarrow \in \Sigma_P$) consists of the sequential execution of the operation of partial derivation, starting from the initial nonterminal and ending with the construction.

To form structures, it is necessary to perform several clarifying transformations of constructors:

– specialization – defines the subject area: the semantic nature of the medium, the finite set of operations and their semantics, the operations attributes, the order of their execution and restrictions on substitution rules;

– interpretation – consists in connecting the operations of the signature with the execution algorithms of some algorithmic designer;

– concretization – expansion of axiomatics by production rules set, specific sets assignment of non-terminal and terminal symbols with their attributes and, if necessary, attribute values;

– implementation – formation of a structure from the constructor carrier elements by performing algorithms related to signature operations.

Constructor-converter of natural language text into a tagged text

The purpose of construction is to convert technical text into tagged text. For each word in a sentence, its attribute is determined in the composition: part of speech (pos), number (num) and gender (gen). Consider the specialization of the designer:

$$C = \langle M, \Sigma, \Lambda \rangle \ {}_S \mapsto C_P = \langle M_P, \Sigma_P, \Lambda_P \rangle,$$

where M_P – a carrier that includes terminal and non-terminal alphabets, initial and tagged texts, as well as a set of production rules Ψ, separate rules ψ_i:<s_i, g_i>, where i – rule number, s_i – is a sequence of substitution relations, g_i – is a sequence of operations on attributes, Σ_P – operations and connections for elements M_P; CIS $\Lambda_P \supset \Lambda$.

The following provisions are included in CIS Λ_P.

Signature Σ_P contains the signature of specific binding operations and attribute operations.

The terminals T include symbols and words of the Ukrainian language, denoted as * - letters that can be used to form words, $\underline{*}$ – space, $\bar{*}$ – end-of-sentence symbols, ⊥ – end-of-text symbol, $W_{i,j}$, - the j-th word in the i-th sentence. The first word of each sentence will additionally store information about its length l, and the first word of the text will store the number of words in the longest sentence, max, and the total number of sentences in the text, S. Non-terminals $N = \{\sigma, \eta, \varepsilon\}$ – auxiliary elements, where ε is the symbol 'empty'.

The following attribute operations are presented.

The operation $\odot$ $(word,\ ends, pos \hookleftarrow word) \in \Sigma_P$ – defines the part of speech pos for the word 'word', which can take one of the following values: verb (v), noun (n), numeral (nume), pronoun (pron), adjective (adj), conjunction (conj), adverb (adv), preposition (prep), verb adverb (v_adv), interjection and particle (frac), verb adjective (v_adj).

The operation $\circledast$ $(word, ends, num \hookleftarrow word) \in \Sigma_P$ – determines the number num for the word 'word', which can be singular (sing) or plural (plur).

The operation $\circledcirc(word,\ ends, gen \hookleftarrow word) \in \Sigma_P$ – determines the gender (gen) for the word 'word', which can take one of the following values: feminine (f), masculine (m), and neuter (n).

Each of these operations compares the word with all elements of 'ends' - corresponding lists of endings [5]. If there is a match with a specific ending, a result is formed, and the parameters pos, num, and gen are assigned corresponding values.

The operation = $(a, b) \in \Sigma_P$ – assigns the value b to the variable a.

The operation $+(c, a, b) \in \Sigma_P$ – adds $c = a + b$.

The Comparison operation *<>(a,b,c,d)* $\in \Sigma_P$ – compares a with b. If a is greater, then the nested operation c is executed; if it is smaller, then d is executed.

Interpretation of the designer

Let's form a constructive system from the SR designer, as an elemental design base, and the SA algorithmic designer, as a model-executor of the design.

$$\langle \mathrm{C_P} = \langle M_P, \Sigma_P, \Lambda_P \rangle, \mathrm{C_A} = \langle M_A, V_A, \Sigma_A, \Lambda_A \rangle \rangle \ {}_{\mathrm{I}}\mapsto \langle \mathrm{C_{PAI}} = \mathrm{M_I}, \Sigma_\mathrm{I}, \Lambda_\mathrm{I} \rangle,$$

where $V_A = \left\{ A_i \ |_{X_i}^{Y_i} \right\}$ – set of forming algorithms in the basic algorithmic structure, X_i and Y_i –possible input and output data of the algorithm $A_i \ |_{X_i}^{Y_i}$, $M_A \supset \bigcup_{A_i \ \in V_A} (X(A_i\ ) \cup Y(A_i\ ))$ – carrier of the algorithmic structure, Σ_A – a set of algorithm binding operations, Λ_A – CIS, $\Omega(C_A)$ – set of algorithms constructed in C_A[2], $\mathrm{M_I} = M_P \cup M_A$, $\Sigma_\mathrm{I} = \Sigma_P \cup \Sigma_A$, $\Lambda_\mathrm{I} = \Lambda_P \cup \Lambda_A \cup \left\{ \left(A_0|_{A_i,A_j}^{A_i \cdot A_j} \hookleftarrow " \cdot " \right); \left(A_1|_{l,s_i}^{l} \hookleftarrow " \Rightarrow " \right); \left(A_2|_{l,\Psi}^{l} \hookleftarrow " | \Rightarrow " \right); \left(A_3|_{\sigma}^{\Omega} \hookleftarrow " || \Rightarrow " \right) \left(A_4|_{word,ends}^{pos \hookleftarrow word} \hookleftarrow " \odot " \right); \left(A_5|_{word,ends}^{num \hookleftarrow word} \hookleftarrow " \circledast " \right); \left(A_6|_{word,ends}^{gen \hookleftarrow word} \hookleftarrow " \circledcirc " \right); \left(A_7|_{b}^{a} \hookleftarrow " = " \right); \left(A_8|_{a,b}^{c} \hookleftarrow " + " \right); \left(A_9|_{a,b}^{c/d} \hookleftarrow " <> " \right) \right\}$

Structure $\mathrm{C_{PAI}}$ contains algorithms for performing operations:

- $A_0|_{A_i,A_j}^{A_i \cdot A_j}$ – is an algorithms composition, $A_i \cdot A_j$ – sequential execution of algorithm A_j after A_i;
- $A_1|_{l,s_i}^{l}$ – is a substitution, where *l* – current form, s_i – s the rule to be executed;
- $A_2|_{l,\Psi}^{l}$ – is a partial output, where Ψ – is the set of production rules to be executed;
- $A_3|_{\sigma,\Psi}^{\Omega}$ – is a complete derivation, where σ – is an axiom, Ψ – is a set of production rules, Ω – is a set of formed constructions;
- $A_4|_{word,ends}^{pos \hookleftarrow word}$ – is a definition for the *word* of its language part *pos;*
- $A_5|_{word,ends}^{num \hookleftarrow word}$ – is a definition for the *word* of its number *num;*
- $A_6|_{word,ends}^{gen \hookleftarrow word}$ – is a definition for the *word* of its gender *gen;*
- $A_7|_{b}^{a}$ – is an assignment of the value *b* to variable *a;*
- $A_8|_{a,b}^{c}$ – is adding $c = a + b$;
- $A_9|_{a,b}^{c/d}$ – execution of action *c* or *d* based on the result of comparing *a* and *b*.

When **specifying** C_{PAI} the following is parameterized:

$$\mathrm{C}_{PAI} \mapsto_\mathrm{K} \mathrm{C}_{PAIK}(\mathrm{TT}) = \langle \mathrm{M_K}, \Sigma_\mathrm{K}, \Lambda_\mathrm{K} \rangle,$$

where $\Lambda_K \supset \Lambda_I \cup \{M_K = T_T \cup N\} \cup \Lambda_1$. TT is a technical text submitted for analysis.

In the substitution rules ψ_i:<s_i, g_i> the sequence of substitution relations s_i consists of the relation $s_{i,1}$ – analysis of the TT, $s_{i,2}$ – formation of a words' set $W_{i,j}$ with their attributes. Operations $g_{i,j}$ are performed after execution of $s_{i,1}$ and before $s_{i,2}$.

Initial construction conditions: σ – a non-terminal from which the derivation begins and the initial values max = 1, i = 1 and j = 1.

Construction completion condition: all incoming text is tagged.

In the first rule, the parsing of the text and the formation of the first element in the tagged text $W_{i,j}$ begins

$$s_1 = \langle \sigma \to \eta, \quad W_{i,j} \to \varepsilon \rangle.$$

Parsing occurs from one character to the next with its rewriting in $W_{i,j}$ for further tagging

$$s_2 = \langle \eta \to * \, \eta, \quad W_{i,j} \to * \, W_{i,j} \rangle.$$

When a space or end-of-sentence mark is reached, a tagging determination is made for the word and the next word is passed

$$s_3 = \langle \eta \to \underline{*} \, \eta, \ \varepsilon \rangle.$$

The operations $\odot$, $\circledast$, and $\circledcirc$ in attribute operations determine the part of speech, number, and gender of a word, respectively. The transition to the next word in the sentence occurs. The flag "done" for each word is set to position 0, and it will be used later for rule formation

$$g_3 = \langle \odot \left(W_{i,j}, pos \downarrow W_{i,j}\right), \ \circledast \left(W_{i,j}, num \downarrow W_{i,j}\right), \ \circledcirc \left(W_{i,j}, gen \downarrow W_{i,j}\right), = \left(done \downarrow W_{i,j}, 0\right), +(j, 1, j) \rangle.$$

The rule s_4 is applied when the end of the sentence is reached, and like the previous rule, the tagging for the word is determined and the next sentence is passed. A transition to the next word is performed. The length of each sentence is calculated and set and stored as an attribute of its first word. The maximum sentence length is determined as an attribute of the very first word in the text. Along with this, to mark the end of the sentence, $\perp(W_{i,j} = \perp)$ will be written to its final position. This is necessary for the correct operation of the following constructor

$$s_4 = \langle \eta \to \overline{*} \, \sigma, \ \rangle,$$

$$g_4 = \langle \odot \left(W_{i,j}, pos \downarrow W_{i,j}\right), \ \circledast \left(W_{i,j}, num \downarrow W_{i,j}\right), \ \circledcirc \left(W_{i,j}, gen \downarrow W_{i,j}\right), \ = \left(l \downarrow W_{i,1}, j\right), <>$$
$$\left(j, max \downarrow W_{1,1}, = \left(max \downarrow W_{1,1}, j\right), \varepsilon\right), = \left(done \downarrow W_{i,j}, 0\right), \ +(j, 1, j), = \left(W_{i,j}, \perp\right), = (j, 1), \ +(i, 1, i) \ \rangle.$$

The last rule is used when the end of the text is reached and is final. The *am* attribute of the first word stores the total number of sentences in the text

$$s_5 = \langle \eta \to \perp, \ \varepsilon \rangle,$$
$$g_5 = \langle = (am \downarrow W_{1,1}, i) \rangle.$$

Realization

The constructor implementation is the language constructions formation from its carrier elements through the algorithm's execution related to signature operations according to the rules of substitution:

$$\mathrm{C}_{\mathrm{PAIK}} \ {}_{R}\!\mapsto \bar{\Omega}(\mathrm{C}_{\mathrm{PAIK}}(\mathrm{TT})),$$

where $\bar{\Omega}(\mathrm{C}_{\mathrm{PAIK}}(\mathrm{TT})) = \Omega(\mathrm{C}_{\mathrm{PAIK}}(\mathrm{TT}))$. $\bar{\Omega}$ – all possible outcomes of the constructor, however, since the generated constructor is based on a specific text, the resulting processed text Ω will be the only possible outcome. As a result of the constructor implementation, the processed text with tagged words as Ω($\mathrm{C}_{\mathrm{PAIK}}$ (TT)) was received.

For example, let's take the sentence «Чорні ґрати розпанахали небо. Червоно-рожеве воно тянуло, манило». The result of the designer's work will look like this:

$$W_{1,1} = {}_{adj,plur,-}\text{Чорні}; \ W_{1,2} =_{n,plur,-} \text{ґрати}; \ W_{1,3} =_{v,plur,-} \text{розпанахал}; \ W_{1,4} =_{n,sing,n} \text{небо};$$
$$W_{2,1} =$$
$${}_{adj,sing,n}\text{Червоно} - \text{рожеве}; \ W_{2,2} =_{pr,sing,n} \text{воно}; \ W_{2,3} =_{v,sing,n} \text{тянуло}; \ W_{2,4} =_{v,sing,n} \text{манило}$$

Tagged text constructor-converter into formal substitution rules set with a probability measure

The purpose of construction is to build a stochastic constructor rule that formalizes the syntactic component of the technical text.

The initial construction condition is the implementation of the $\mathrm{C_P}$ constructor – the tagged text Tg obtained as a result of the constructor implementation $\mathrm{C}_{\mathrm{PAIK}}$ – Ω($\mathrm{C_P}$ (TT)).

Construction completion condition: each sentence of the tagged text is converted into a corresponding set of rules Ω($\mathrm{C_T}$ (R)), which happens under the condition $\tau 5 = true$, which is set when the last word of the longest sentence in the text is reached. This will serve as an indication that all other words in the text have already been processed and the rule building process is complete.

The designer has the following specialization:

$$C = \langle M, \Sigma, \Lambda \rangle \ \ {}_{S}\mapsto C_T(Tg) = \langle M_T, \Sigma_T, \Lambda_T \rangle,$$

where M_T – is a carrier that includes tagged text Tg, Σ_T – operations and relationships on elements M_T and axiomatics Λ_T.

The operation $\circledast$ (r, a, b) – checking that the attributes pos↲a, num↲a, gen↲a of element match the attributes pos↲b, num↲b, gen↲b of element b, where a and b are tagged words. If there is a complete match, the result is 1, otherwise - 0.

The operation $\&(\mathrm{y}, \mathrm{x1}, \mathrm{x2})$ – is a logical and with an unlimited number of operands y = x1 and x2 and …;

A loop operation $\circ$ (a, c) – where a – is a condition, c – is an operation performed while the condition is valid;

The operation $-(\mathrm{c}, \mathrm{a}, \mathrm{b})$ – is equal to c = a - b in infixed form;

The operation $:(\mathrm{c}, \mathrm{a}, \mathrm{b})$ – is equal to c = a : b in infixed form, division of real numbers;

The operation ≤(r,a,b) – is a comparing a ≤ b with saving the result in r.

Interpreting the C_T constructor using the same algorithmic constructor C_{A}:

$$\langle C_T, C_A \rangle \ {}_{\mathrm{I}}\!\mapsto \langle \mathrm{C}_T = \mathrm{M}_{\mathrm{TI}}, \Sigma_{\mathrm{TI}}, \Lambda_{\mathrm{TI}} \rangle,$$

M_{TI} – algorithmic structure for the formation of a stochastic constructor from tagged text, Σ_{TI} – algorithm linking operations, $\Lambda_{\mathrm{TI}} \supset \Lambda_{\mathrm{I}} \cup \Lambda_1 \cup \Lambda_2$.

$\Lambda_2 = \{ (A_{10}|^r_{a,b} \hookleftarrow "\circledast"); (A_{11}|^r_{a,b} \hookleftarrow "\&"); (A_{12}|^c_{a,b} \hookleftarrow "-"); (A_{13}|^c_{a,b} \hookleftarrow ":"); (A_{14}|^r_{a,b} \hookleftarrow "\leq"). \}$.

The $\mathrm{C}_{TA\mathrm{I}}$ structure includes the following algorithms:

- $A_0, A_1, A_2, A_3, A_7, A_8, A_9$ – are similar algorithms of the$\mathrm{C}_{\mathrm{PAI}}$ constructor;
- $A_{10}|^r_{a,b}$ – is a comparison of a and b for their identity;
- $A_{11}|^r_{a,b}$ – is logical "and";
- $A_{12}|^c_{a,b}$ – is a number subtraction;
- $A_{13}|^c_{a,b}$ – is a real numbers division;
- $A_{14}|^r_{a,b}$ – is a comparison of a and b.

Specification of C_{T}:

$$C_T \mapsto_{\mathrm{K}} C_T(\mathrm{C}_P(Tg)) = \langle \mathrm{M}_{\mathrm{K}}, \Sigma_{\mathrm{K}}, \Lambda_{\mathrm{K}} \rangle,$$

where $\Lambda_{\mathrm{K}} \supset \Lambda_{\mathrm{I}}$, $\Lambda_K \supset \{M_K = T \cup N\}$, the terminals T include all words $W_{m,j}$ with the designation of their place j in the sentence m, $\alpha_{k,j}$, which is a non-terminal of the rule being constructed, σ –is the initial non-terminal and the constructed rule ω_k, which in its attributes will have the left part of the rule L, the right part R and the probability of its operation for the given text prob. The non-terminals N: τ_{i} – is the rule availability attribute.

For each part of speech, its appearance probability (prob) in a certain place of a certain sentence in this text is calculated. The appearance probability of a certain language part in the investigated sequence will allow for a more accurately capturing of the individual author’s writing style characteristic.

The probability of obtaining the entire sentence is defined as its speech parts sequences probabilities product. The resulting constructor will generate a language characteristic of the processed text and structurally similar texts of a certain author.

Thus, each sentence of the presented text will be presented in the form of a chain of rules that will reflect the sequence of used parts of speech and the probability of their appearance in the presented sequence.

Initial conditions: the initial form $W_{1,1}$ – is the first word in the text, where i = 1, n = 2 sentence numbers, j = 1, m = 2 word numbers in them. t = 0 – the number of matches with the selected pair of parameters in a sequence of two words, k = 1 – the number of the rule being built. $\tau 1 = true, \tau 2 = false \ldots \tau 5 = false$ – are conditions for the execution of the rules: if true, it is available for use, if false - not. idone = 1, jdone = 1 are variables equal to the number of the unique element in the layer and the previous layer, respectively, and u = false – a flag for marking already built rules.

Parsing begins with processing the first layer (the first words in the sentences of the text) and searching for a match by attributes among them

$s_1 = \langle W_{i,j} \ {}_{\tau 1}\!\!\rightarrow W_{n,j}, \varepsilon \rangle$,

$$g_{1,1} = \langle == (0, \mathrm{done} \hookleftarrow \mathrm{W}_{\mathrm{i,j}}, \mathrm{x1}), \circledast (\mathrm{W}_{\mathrm{i,j}}, \mathrm{W}_{\mathrm{n,j}}, \mathrm{x2}) \rangle.$$

Searches for matching words with the same attributes in the current layer. If a match with the current word of this current layer is found, and no match was found for this word before, we increase the total number of similar to the searched word (t) and move to the next sentence by increasing n

$$g_{1,2} = \langle \&(\mathrm{y}, \mathrm{x1}, \mathrm{x2}), <> (y, 0, +(t, 1, t), \varepsilon), +(n, 1, n) \rangle.$$

The second rule is used when the end of the sentence is reached, in this case no calculations take place, only the sentence number n is increased to move to the next word in the layer

$$s_2 = \langle W_{i,j} \ {}_{\tau 1}\!\!\rightarrow W_{n,j} \perp, \varepsilon \rangle,$$
$$g_2 = \langle +(n, 1, n) \rangle.$$

The third rule is applied when it is impossible to reach the next word in the layer due to reaching its end

$$s_3 = \langle W_{i,j} \ {}_{\tau 1}\!\!\rightarrow \perp, \varepsilon \rangle.$$

Under this condition, the calculation of the probability of the selected sequence appearing in the text is used, the rules $s_1 - s_3$ become unreachable, and the rules $s_4 - s_7$ become available for processing

$$g_3 = \langle -(n,n,1), :(prob,t,n), = (\tau 1, false), = (\tau 2, true), = (n,1) \rangle.$$

The following rules are responsible for forming rules for the first words of each of the sentences when all corresponding words are repeated

$$s_4 = \langle W_{i,j}\ {}_{\tau 2}\!\rightarrow W_{n,j},\ \omega_k \rangle,$$

$$g_{4,1} = \langle == \left(0, \mathrm{done} \downarrow \mathrm{W_{i,j}}, \mathrm{x1}\right), \circledast \left(\mathrm{W_{i,j}}, \mathrm{W_{n,j}}, \mathrm{x2}\right) \rangle.$$

If the attributes of the words match, a new rule ω_k is built. σ and $W_{i,j}\alpha_{i,j}$ are written in its left and right parts, respectively, where $\alpha_{i,j}$ – is the non-terminal of the newly formed rules, and the probability of its activation for the text $prob$ is written

$$g_{4,2} = \langle \&(\mathrm{y}, \mathrm{x1}, \mathrm{x2},), \quad <> (y, 0, \cdot \left(= \left(L \downarrow \omega_{i,j}, \sigma\right), = \left(R \downarrow \omega_{i,j}, W_{i,j}\alpha_{i,j}\right), = \left(prob \downarrow \omega_{i,j}, prob\right)\right), \varepsilon) \rangle.$$

After the formation of the rule, the flag of the presence of at least one rule on this layer is set $u = true$, the rule receives the uniqueness index in the *idone* layer, and the constructor moves to the next sentence

$$g_{4,4} = \langle +(n,1,n), = \left(done \downarrow \mathrm{W_{i,j}}, idone\right), = (u, true) \rangle.$$

The next rule is similar to rule s_2, does not make calculations and is responsible for increasing the sentence number n to move further through the layer

$$s_5 = \langle W_{i,j}\ {}_{\tau 2}\!\rightarrow W_{n,j} \perp, \varepsilon \rangle,$$

$$g_5 = \langle +(n,1,n) \rangle.$$

If there is only one word in the sentence, rule s_6 is used

$$s_6 = \langle W_{i,j} \perp_{\tau 2}\!\rightarrow W_{n,j},\ \omega_k \rangle.$$

To form a rule in this case, the following checks will be carried out: whether the word is included in another done rule, whether the corresponding attributes of the words match

$$g_{6,1} = \langle == \left(0, \mathrm{done} \downarrow \mathrm{W_{i,j}}, \mathrm{x1}\right), \circledast \left(\mathrm{W_{i,j}}, \mathrm{W_{n,j}}, \mathrm{x2}\right) \rangle.$$

If the sentence is not the first, we form the corresponding rule ω_k. In its left part, write σ, respectively, and only $W_{i,m}$ in the right part

$$g_{6,3} = \langle \&(\mathrm{z}, \mathrm{x1}, \mathrm{x2}, \mathrm{x3}),\ <> \left(z, 0, \cdot (= \left(L \downarrow \omega_{i,j}, \sigma\right), = \left(R \downarrow \omega_{i,j}, W_{i,j}\right), = \left(prob \downarrow \omega_{i,j}, prob\right)), \varepsilon \right) \rangle.$$

And then the same as in $g_{4,4}$ – its creation flag $u = true$, is set, the rule receives the uniqueness index in the idone layer and the executor moves to the next sentence

$$g_{6,4} = \langle +(n,1,n), = (done \downarrow \mathrm{W_{i,j}}, idone), = (u, true) \rangle.$$

And when the end is reached, rule s_7 is triggered

$$s_7 = \langle W_{i,j}\ {}_{\tau 2}\!\rightarrow \perp, \varepsilon \rangle.$$

Reaching the end of a layer means the end of rule formation and transition to the formation of another. For this, the flags $\tau 2 = false$ and $\tau 1 = true$ are changed, which will close the rules $s_4 - s_7$ and open the rules $s_1 - s_3$ to search for other matches and count them. To reflect the operation of another rule in the layer, the rule's uniqueness number for the *idone* layer is increased

$$g_{7,1} = \langle == (u, true, y), <> (y, 0, \cdot \left(= (\tau 2, false), = (\tau 1, true), +(idone, 1, idone), = (t, 0)\right), \varepsilon \rangle.$$

If the work with the layer is completed and rules have been formed for all the words in it, the constructor moves to the next layer, starting again from the first sentence i=1 to search for a match. The calculation of the uniqueness of the rules in the layer also starts from the beginning of idone=1. If the final layer is reached (the last word in the longest sentence $\mathrm{W}_{i,max}$ is processed), the work of the performer with the first layer will be completed $\tau 3 = true$, $\tau 1 = false$, $\tau 2 = false$

$$g_{7,2} = \langle == (u, false, y), <> (y, 0, \cdot (= (i,1), = (idone, 1), = (t, 0), = (\tau 3, true), = (\tau 2, false), = (\tau 1, false)), \varepsilon \rangle.$$

To continue forming rules from tagged text, operating the consecutive pairs of words in each sentence. The transition from word to word does not occur along the sentence, but according to the number of words in them. In this way, the constructor considers a pair of consecutive words in a sentence

$$s_8 = \langle W_{i,j}W_{i,m}\ {}_{\tau 3}\!\rightarrow W_{n,j}W_{n,m}, \varepsilon \rangle.$$

To consider an existing pair of words as similar, you need to check the following parameters: the words have not yet been processed; the attributes of the selected sequence of two words (part of speech, gender and number) match the numbered words in the next sentence, the previous string of words must also match, which is checked by jdone

$$g_{8,1} = \langle == \left(0, \mathrm{done} \downarrow \mathrm{W_{i,j}}, \mathrm{x1}\right), \circledast \left(\mathrm{W_{i,j}}, \mathrm{W_{n,j}}, \mathrm{x2}\right), \circledast \left(\mathrm{W_{i,m}}, \mathrm{W_{n,m}}, \mathrm{x3}\right), -(\mathrm{k}, \mathrm{j}, 1), <> \left(\mathrm{j}, 1, == \left(\ \mathrm{done} \downarrow \mathrm{W_{i,k,}} \mathrm{jdone}, \mathrm{x4}\ \right), \mathrm{x4} = \mathrm{true}\right) \rangle.$$

If the attributes match, the pair is counted in the total number of similar sequences and the value of t and the value of n are increased to move to the next sentence in the layer

$$g_{8,2} = \langle \&(\mathrm{y}, \mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}), <> (y, 0, +(t, 1, t), \varepsilon\ ), +(n, 1, n) \rangle.$$

The following rule is triggered when the end of the sentence is reached, the sentence number n is incremented for further viewing of the words in the layer

$$s_9 = \langle W_{i,j} W_{i,m} \ {}_{\tau 3}\!\rightarrow W_{n,j} \ \perp, \varepsilon \rangle,$$
$$g_9 = \langle +(n, 1, n) \rangle.$$

The following rule is executed when it is impossible to move further along the sentences due to reaching the end of the layer. Under this condition, the probability of the appearance of the selected sequence in the text is calculated, rules $s_8 - s_{10}$ become unreachable, and rules $s_{11} - s_{14}$ become available for processing

$$s_{10} = \langle W_{i,j} \ {}_{\tau 3}\!\rightarrow \perp, \varepsilon \rangle,$$
$$g_{10} = \langle -(n, n, 1), : (prob, t, n), = (\tau 3, false), = (\tau 4, true), = (i, 1), = (n, 1) \rangle.$$

The next step is to revisit the current layer and create rules ω_k

$$s_{11} = \langle W_{i,j} W_{i,m} \ {}_{\tau 4}\!\rightarrow W_{n,j} W_{n,m}, \ \omega_k \rangle.$$

To form the appropriate rule, the check from the first rule is repeated and we additionally check whether the word is the first in the sentence (x4) for the correct formation of the initial rules

$$g_{11,1} = \langle == \left(0, \mathrm{done} \downarrow \mathrm{W_{i,j}}, \mathrm{x1}\right), \circledast \left(\mathrm{W_{i,j}}, \mathrm{W_{n,j}}, \mathrm{x2}\right), \circledast \left(\mathrm{W_{i,m}}, \mathrm{W_{n,m}}, \mathrm{x3}\right), -(\mathrm{k}, \mathrm{j}, 1), <> \left(\mathrm{j}, 1, == \left(\mathrm{done} \downarrow \mathrm{W_{i,k,}} \mathrm{jdone}, \mathrm{x4}\right), \mathrm{x4} = \mathrm{true}\right) \rangle.$$

If everything matches and the word is not the first in the sentence, a new rule ω_k is built. $\alpha_{i,j}$ is written in the left part of the rule, $W_{i,m}\alpha_{i,m}$is written in its right part, where $\alpha_{i,j}$ – is the non-terminal of the newly formed rules, and the probability of its activation for the text $prob$ is written.

$$g_{11,2} = \langle \&(\mathrm{y}, \mathrm{x1}, \mathrm{x2}, \mathrm{x3}, \mathrm{x4}),$$
$$<> (y, 0, \cdot \left(= \left(L \downarrow \ \omega_{i,j}, \alpha_{i,j}\right), = \left(R \downarrow \ \omega_{i,j}, W_{i,m}\alpha_{i,m}\right), = \left(prob \downarrow \ \omega_{i,j}, prob\right)\right), \varepsilon) \rangle.$$

After creating a rule, its creation flag $u = true$, is set, the rule receives a unique index in the idone layer, and the constructor moves to the next sentence

$$g_{11,3} = \langle +(n, 1, n), = \left(done \downarrow \mathrm{W_{i,j}}, idone\right), = (u, true) \rangle.$$

The next rule is similar to rule s_9, does not make calculations and is responsible for increasing the sentence number n to move further through the layer

$$s_{12} = \langle W_{i,j} W_{i,m} \ {}_{\tau 4}\!\rightarrow W_{n,j} \ \perp, \varepsilon \rangle,$$
$$g_{12} = \langle +(n, 1, n) \rangle.$$

If the last word in the sentence is in the layer, rule s_{13} is triggered

$$s_{13} = \langle W_{i,j} \ \perp_{\tau 4}\!\rightarrow W_{n,j}, \ \omega_k \ \rangle.$$

To form a rule in this case, the following checks will be carried out: whether the word is included in another done rule, whether the corresponding attributes of the words match, whether the word is the first in the sentence and, if not, whether the previous chain matches

$$g_{13,1} = \langle == \left(0, \mathrm{done} \downarrow \mathrm{W_{i,j}}, \mathrm{x1}\right), \circledast \left(\mathrm{W_{i,j}}, \mathrm{W_{n,j}}, \mathrm{x2}\right), <> \left(\mathrm{j}, 1, == \left(\mathrm{done} \downarrow \mathrm{W_{i,j-1,}} \mathrm{jdone}, \mathrm{x3}\right), \mathrm{x3} = \mathrm{true}\right) \rangle.$$

If the sentence is not the first, we form the corresponding rule ω_k. $\alpha_{i,j}$ is written in the left part of the rule, $W_{i,m}\alpha_{i,m}$is written in its right part, where $\alpha_{i,j}$ – is the non-terminal of the newly formed rules, and the probability of its activation for the text $prob$ is written.

$$g_{13,2} = \langle \&(\mathrm{y}, \mathrm{x1}, \mathrm{x2}, \mathrm{x3}), \ <> (y, 0, \cdot \left(= \left(L \downarrow \ \omega_{i,j}, \alpha_{i,j}\right), = \left(R \downarrow \ \omega_{i,j}, W_{i,j}\right), = \left(prob \downarrow \ \omega_{i,j}, prob\right)\right), \varepsilon) \rangle.$$

Next, just like in $g_{11,3}$ – its creation flag *u = true* is set, the rule receives the uniqueness index in the idone layer and the constructor moves to the next sentence

$$g_{13,3} = \langle +(n, 1, n), = (done \downarrow \mathrm{W_{i,j}}, idone), = (u, true) \rangle.$$

When the end is reached, rule s_{14} is triggered

$$s_{14} = \langle W_{i,j} \ {}_{\tau 4}\!\rightarrow \perp, \varepsilon \rangle.$$

Reaching the end of a layer means the end of rule formation and transition to the formation of another. For this, the flags $\tau 4 = \ false$ and $\tau 3 = \ true$ are changed, which will close the rules $s_{11} - s_{14}$ and open the rules $s_8 - s_{10}$ to search for other matches and count them. To reflect the operation of another rule in the layer, the rule's uniqueness number for the *idone* layer is increased

$$g_{14,1} = \langle = (\tau 4, false), = (\tau 3, true), +(idone, 1, idone), = (t, 0), \leq (r, j, l \ \downarrow \mathrm{W_{i,j}}) \rangle.$$

The procedure of counting coincidences and calculating the probability of their occurrence for building rules on its basis continues until all words in the layer have been processed. To work with all chains, at each layer pass, the *jdone* uniqueness index is increased to check the calculation condition

$$g_{14,2} = \langle <> \left(am \downarrow \mathrm{W_{1,1}}, i, \circ \left(r, \cdot \left(= (i, 1, i)\right), == (u, true, 1) <> (1{,}0, +(jdone, 1, jdone))\right)\right) \rangle.$$

If the work with the layer is completed and rules have been formed for all the words in it, the constructor moves to the next layer by increasing j and starting again from the first sentence *i = 1* to look for a match. The calculation of the uniqueness of rules in a layer also starts from the beginning of *idone = 1* and *jdone = 1*. On the condition that the final layer is reached (the last word in the longest sentence $\mathrm{W}_{i,max}$ is processed), the work of the constructor will be completed $\tau 5 = \ true$

$g_{14,3} = \langle <> (max \downarrow W_{1,1}, j, \cdot (+(j,1,j), = (i,1), = (idone, 1), = (jdone, 1)), \cdot (= (\tau 5, true), = (\tau 3, true)))\rangle$.

As a result of the work of the constructor-converter with $\Omega(C_{PAIK}(TT))$, we get a set of rules that reflects the style of the author's language in the corresponding text $\Omega(C_T(R))$.

Realization

The implementation of the structure is the formation of language constructions from the elements of its carrier through the execution of algorithms related to signature operations according to the rules of axionomics:

$$C_{P\,K} \;{}_{R}\mapsto \bar{\Omega}(C_{P\,K}),$$

where $\bar{\Omega}(C_{P\,K}) \subset \Omega(C_{P\,K})$.

For example, let's take sentences that have the form:

«Ми були дуже схожі.
Я любила читати книжки.
А ти захоплювався виствами.
Але..
Між нами було й багато різниці».

The tagged text for this example:

$W_{1,1} = {}_{pron,plur}$Ми $W_{1,2} ={}_{v,plur}$ були $W_{1,3} ={}_{adv,sing}$ дуже $W_{1,4} ={}_{adj,plur}$ схожі
$W_{2,1} = {}_{pron,sing}$Я $W_{2,2} ={}_{v,sing}$ любила $W_{2,3} ={}_{v,sing}$ читати $W_{2,4} ={}_{n,plur}$ книжки
$W_{3,1} = {}_{conj}$А $W_{3,2} ={}_{pron,sing}$ ти $W_{3,3} ={}_{v,sing}$ захоплювався $W_{3,4} ={}_{n,plur}$ виставами
$W_{4,1} = {}_{pron}$Але
$W_{5,1} =$
${}_{conj}$Між $W_{5,2} ={}_{pron,plur}$ нами $W_{5,3} ={}_{v,sing}$ було $W_{5,4} ={}_{conj}$ й $W_{5,5} ={}_{adj,sing}$ багато $W_{5,6} ={}_{adj,plur}$ різного.

The result of the designer's work will be presented in the form of relevant rules:

$\sigma \xrightarrow{0.2} W_{1,1}\alpha_{1,1};\ \alpha_{1,1} \xrightarrow{0.2} W_{1,2}\alpha_{1,2};\ \alpha_{1,2} \xrightarrow{0.2} W_{1,3}\alpha_{1,3};\ \alpha_{1,3} \xrightarrow{0.2} W_{1,4};$
$\sigma \xrightarrow{0.2} W_{2,1}\alpha_{1,1};\ \alpha_{2,1} \xrightarrow{0.2} W_{2,2}\alpha_{2,2};\ \alpha_{2,2} \xrightarrow{0.6} W_{2,3}\alpha_{2,3};\ \alpha_{2,3} \xrightarrow{0.4} W_{2,4};$
$\sigma \xrightarrow{0.4} W_{3,1}\alpha_{3,1};\ \alpha_{3,1} \xrightarrow{0.2} W_{3,2}\alpha_{3,2};\ \alpha_{3,2} \xrightarrow{0.6} W_{3,3}\alpha_{3,3};\ \alpha_{3,3} \xrightarrow{0.4} W_{3,4};$
$\sigma \xrightarrow{0.2} W_{4,1};$
$\sigma \xrightarrow{0.4} W_{5,1}\alpha_{5,1};\ \alpha_{5,1} \xrightarrow{0.2} W_{5,2}\alpha_{5,2};\ \alpha_{5,2} \xrightarrow{0.6} W_{5,3}\alpha_{5,3};\ \alpha_{5,3} \xrightarrow{0.2} W_{5,4}\alpha_{5,4};\ \alpha_{5,4} \xrightarrow{1} W_{5,5}\alpha_{5,5};\ \alpha_{5,5} \xrightarrow{1} W_{5,6}$.

Constructor-measurer of the similarity degree

In order to establish the similarity degree of the two texts according to the syntactic style of the author's language, a comparison of the text models is carried out with the help of a constructor-meter.

The purpose of construction is to establish the degree of similarity of texts by comparing stochastic constructors built according to their syntactic structure.

The initial conditions for constructing a model of two texts in the form of a set of substitution rules with the probability of its activation $\Omega(C_T(R_1))$ and $\Omega(C_T(R_2))$, which represent the text of certain technical works $\Omega(C_{PAIK}(TT_1))$ and $\Omega(C_{PAIK}(TT_2))$, which is the result of the execution of previous constructors.

Construction completion condition: $\tau 3 = true$, getting a number from 0 to 1 that reflects the similarity of two works after comparing all rules in two text models.

The designer has the following specialization:

$$C = \langle M, \Sigma, \Lambda\rangle \;{}_S \mapsto C_E = \langle M_E, \Sigma_E, \Lambda_E\rangle,$$

where M_E – is a medium that includes a set of rules describing the language of the author in a certain text R_i, Σ_E – are operations and relations on the elements M_E and CIS Λ_E.

We interpret the structure C_E using the algorithmic structure C_A:

$$\langle C_E, C_A\rangle \;{}_I\mapsto \langle C_E = M_{PI}, \Sigma_{PI}, \Lambda_{PI}\rangle,$$

where $V_A = \left\{A_i^0|_{X_i}^{Y_i}\right\}$ – is the set of forming algorithms of the basic algorithmic structure, X_i and Y_i – are the set of definitions and values of the algorithm $A_i^0|_{X_i}^{Y_i}$, $M_A = \bigcup_{A_i^0 \in V_A}(X(A_i^0) \cup Y(A_i^0))$ – he carrier of the algorithmic structure, Σ_I – the set of operations linking algorithms, Λ_I – the axiomatics of the algorithmic structure, $\Omega(C_A)$ – a set of algorithms constructed in C_A.

Next, the operation on attributes is presented.

The operation min(m, a, b) compares the numbers a and b, and stores the smallest in m;

The operation –(c, a, b) – is subtraction $c = a - b$;

The operation *(c, a, b) – is multiplication $c = a * b$;

$\mathrm{M_{PI}}$ – algorithmic structure for comparing rules, Σ_{PI} – operations of connecting algorithms, $\Lambda_{\mathrm{TI}} \supset \Lambda_{\mathrm{I}} \cup \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$.

$\Lambda_3 = \{ (A_{15}|_{a,b}^{m} \downarrow "min"); (A_{16}|_{a,b}^{m} \downarrow " - "); (A_{17}|_{a,b}^{m} \downarrow " * "); (A_{18}|_{a,b}^{m} \downarrow "max") \}$.

The $\mathrm{C_{PAI}}$ structure includes the following algorithms:

- $A_0, A_1, A_2, A_3, A_6, A_7, A_8, A_9, A_{10}$ – similar algorithms of C_{PAI} and $\mathrm{C_{TAI}}$ structures;
- $A_{15}|_{a,b}^{m}$ – finding the minimum among the numbers a and b;
- $A_{16}|_{a,b}^{c}$ – subtraction *c = a – b;*
- $A_{17}|_{a,b}^{c}$ – multiplication *c = a * b;*
- $A_{18}|_{a,b}^{m}$ – finding the maximum among the numbers a and b.

Specifics $\mathrm{C_T}$:

$$C_{E\ \ K} \mapsto C_E(\Omega(\mathrm{CT}\,(R_1))\,, \Omega(\mathrm{CT}\,(R_2))) = \langle \mathrm{M_K}, \Sigma_{\mathrm{K}}, \Lambda_{\mathrm{K}} \rangle,$$

where $\Lambda_{\mathrm{K}} \supset \Lambda_{\mathrm{I}}$, $\Lambda_K \supset \{M_K = T_T \cup N\}$ the terminals T include all the words in the rules of both constructors that compare ω and $\dot{\omega}$, the non-terminals N – include the auxiliary symbol τ.

In terms of constructive-synthesizing modeling, the set of rules comparing process for the formation of two texts (T_1 and T_2, respectively) and obtaining the final value of their similarity.

The first rule starts by comparing the rules of two constructors $\Omega(\mathrm{C_T}\,(R_1))$ and $\Omega(\mathrm{C_T}\,(R_2)$ describing two texts that are examined for their similarity, i =1, j=1.

If the same rules or rules exist, the degree of their statistical structural similarity will be determined as the product of the minimum difference in the probabilities of applying the corresponding rule

$$\rho(\vartheta_i, \dot{\vartheta}_j) = \prod_{m=1}^{l} min(prob_m - \dot{prob}_m),$$

where ϑ_i – i-th sentence in T_1 text and $\dot{\vartheta}_j$ – j-th sentence in T_2 text.

The degree of statistical structural similarity of T_1 and T_2 texts:

$$\rho(T_1, T_2) = \sum_{i=1}^{N} \rho(\vartheta_i, \dot{\vartheta}_j),$$

Initial conditions: *rule* = 1, i = m = j = n = 1, where i and m are numbers of chains (sentences) in the text, j and n – are numbers of rules in chains. $max \downarrow \omega_{i,1}$, where max = 0 is the product of the difference in probabilities. $max\_ch \downarrow \omega_{i,1}$, max_ch = 0 is the maximum length of the chain, res = 0 – is the total similarity of two texts, $k = n + 1, h = j + 1$, these are the next rules in the chain concerning j and n, respectively. And the flags for triggering s_1 and s_2 $\tau 1 = true, \tau 2 = false$, as well as the flag for completing the comparison $\tau 3 = false$.

The first rule is used to compare the first rules in all strings of text

$$s_1 = \langle \sigma \ {}_{\tau 1}\!\!\rightarrow \ \vartheta_{i,1}; \sigma \ {}_{\tau 1}\!\!\rightarrow \ \vartheta_{m,1} \rangle.$$

For each rule, if their right parts match and the length of the chain is only one rule (that is, the sentence consists of only one word)

$$g_{1,1} = \langle \divideontimes (R \downarrow \vartheta_{i,1}, R \downarrow \dot{\vartheta}_{m,1}, x1), == (l \downarrow W_{i,1} \downarrow \vartheta_{i,1}, 1, x2), == (l \downarrow W_{m,1} \downarrow \vartheta_{m,1}, 1, x3), \&(y, x1, x2, x3) \rangle.$$

If all conditions are met, the product of the difference in their probabilities is calculated, and the result is stored in the first element of the chain. And until the end of the second text is reached, the products are added up in *res*. If the chains from the first text end, the first rule is closed and the second is opened

$$g_{1,3} = \langle <> (y, 0, \cdot\ (* \left(max \downarrow \vartheta_{i,1}, min \left(-\left(r, prob \downarrow \vartheta_{i,h}, prob \downarrow \dot{\vartheta}_{m,k}\right)\right)\right),$$
$$<> (max \downarrow W_{1,1} \downarrow \vartheta_{m,1}, m, \varepsilon, \cdot \left(+\left(res, max \downarrow \vartheta_{i,1}, r\right), +(i, 1, i), = (m, 1)\right),$$
$$<> (max \downarrow W_{1,1} \downarrow \vartheta_{i,1}, i, \varepsilon, \cdot \left(= (\tau 1, false), = (\tau 2, true)\right))), \varepsilon) \rangle.$$

The second rule sequentially traverses all strings longer than one rule, advancing along their length for both texts under investigation. All rules of the second text are reviewed (m varies from 1 to the end of the text). For each sentence, a sequential review of all rules is performed

$$s_2 = \langle \vartheta_{i,j} \ {}_{\tau 2}\!\!\rightarrow \ \vartheta_{i,h}, \ \dot{\vartheta}_{m,n} \ {}_{\tau 2}\!\!\rightarrow \dot{\vartheta}_{m,k} \rangle.$$

To start work and calculate similarities, the right parts of the first rules in both texts are compared and we perform operations on the attributes

$$g_{2,1} = \langle \divideontimes (R \downarrow \vartheta_{i,j}, R \downarrow \dot{\vartheta}_{m,n}, x1), <> \left(l \downarrow W_{i,1} \downarrow \vartheta_{i,j}, j, = (x2, true)), = (x2, false)\right), <> (l \downarrow W_{m,1} \downarrow \dot{\vartheta}_{m,n}, n, = (x3, true), = (x3, false)), == (j, 1, x4) \rangle.$$

If all conditions are met, the first rule in the chain is processed: the length of the chain that matches *ch* is calculated, the product of the difference in the probabilities of the rules from both sim texts is found, and the maximum length of the matching chain and the result of calculating their coincidence are stored in the first element of the chain

$$g_{2,2} = \langle \&(y, x1, x2, x3, x4), <> (y, 0, \cdot \left(+(ch \lrcorner \vartheta_{i,1}, 1, ch \lrcorner \vartheta_{i,1}), \ * \left(rule, min\left(-\left(r, prob \lrcorner \vartheta_{i,j}, prob \lrcorner \dot{\vartheta}_{m,n}\right)\right)\right), \ = \left(sim \lrcorner \vartheta_{i,1}, rule\right), <> (ch \lrcorner \vartheta_{i,1}, maxch \lrcorner \vartheta_{i,1}, \ \cdot \left(= \left(maxch \lrcorner \vartheta_{i,1}, ch \lrcorner \vartheta_{i,1}\right), \ = \left(max \lrcorner \vartheta_{i,1}, sim \lrcorner \vartheta_{i,1}\right)\right), \ +(j,1,j), +(n,1,n)), \cdot (= (j,1), = (n,1), +(m,1,m)), \varepsilon) \rangle.$$

Then all subsequent chains and their rules are processed under the same conditions

$$g_{2,3} = \langle \divideontimes \left(R \lrcorner \vartheta_{i,h}, R \lrcorner \dot{\vartheta}_{m,k}, x1\right), <> \left(l \lrcorner W_{i,1} \lrcorner \vartheta_{i,h}, h, = (x2, true)), = (x2, false)\right), <> \left(l \lrcorner W_{m,1} \lrcorner \dot{\vartheta}_{m,n}, k, = (x3, true), = (x3, false)\right), \&(y, x1, x2, x3) \rangle.$$

If the chain of coincidences is broken, the comparison of the rules of the 2nd text begins already for the next chain of rules of the first text. If the chain has ended, the transition to the next one is performed, and each of the rules in both texts is similarly checked for coincidence. If the rules in the text end, we close the possibility of executing the second rule $\tau 2 = false$ and end the calculations using the flag $\tau 3 = true$

$$g_{2,4} = \langle <> (y, 0, \cdot \left(+(ch \lrcorner \vartheta_{i,1}, 1, ch \lrcorner \vartheta_{i,1}), * \left(rule, min\left(-\left(r, prob \lrcorner \vartheta_{i,h}, prob \lrcorner \dot{\vartheta}_{m,k}\right)\right)\right), \ = \left(sim \lrcorner \vartheta_{i,1}, rule\right), +(h,1,h), +(k,1,k)), <> (max \lrcorner W_{1,1} \lrcorner \vartheta_{m,1}, m, \varepsilon, \cdot \left(+\left(res, max \lrcorner \vartheta_{i,1}, r\right), +(i,1,i), \ = (n,1), = (m,1), +(k,n,1), = (j,1), +(h,j,1)\right), <> (max \lrcorner W_{1,1} \lrcorner \vartheta_{i,1}, i, \varepsilon, \cdot \left(= (\tau 2, false), = (\tau 3, true)\right) \rangle.$$

Note that $\rho(T_1, T_2) = \rho(T_2, T_1)$ $\rho(T_1, T_1) = 1$ is a complete match, $\rho(T_1, T_2) = 0$ – if there are no sentences of the same structure in texts T_1 and T_2.

Realization

The implementation of the structure is the language constructions formation from the elements of its carrier through the execution of algorithms associated with signature operations according to the rules of axionomics:

$$C_{P\,K} \ {}_{R}\mapsto \bar{\Omega}(C_{P\,K}),$$

where $\bar{\Omega}(C_{P\,K}) \subset \Omega(C_{P\,K})$. As the constructor's work result is a number $\bar{\Omega}(C_{P\,K}) \in [0; 1]$, is obtained, which reflects the degree of similarity of the text.

Conclusions

In this paper, constructors are developed and presented that model a natural language text in the form of a stochastic grammar that displays the structures of sentences in it. This approach allows you to highlight the syntactic features of the construction of phrases by the author, which is a characteristic of his speech. Working with a sentence as a unit of text for analyzing its construction will allow you to more accurately capture the author's style in terms of the words use, their sequences and speech style characteristic. It allows you not to be tied to specific parts of speech, but reveals the general logic of constructing phrases, which can be more informative in terms of the author's style characteristics for any text.

The presented work is a theoretical basis for solving the problems of the text authorship establishing and identifying borrowings. Experimental studies have also been carried out, the results of which are partially presented in [3]. The statistical similarity of solutions to the problems of establishing authorship and identifying borrowings was experimentally revealed, which will be presented in the next article of the authors.

It is planned to use the created model in the future to determine the authorship of natural language texts of various directions: fiction and technical literature.

References

1. Kuropiatnyk, O., Shynkarenko V. Automation of template formation to identify the structure of natural language documents. CEUR Workshop Proceedings. 2021. Vol. 2870: 5th International Conference on Computational Linguistics and Intelligent Systems. Vol. I: Main Conference, COLINS 2021, 22–23 April 2021. P. 179–190.
2. V. I. Shynkarenko, V. M. Ilman. Constructive-Synthesizing Structures and Their Grammatical Interpretations. I. Generalized Formal Constructive-Synthesizing Structure / Cybernetics and Systems Analysis, 2014, Volume 50, Issue 5, pp 655-662.
3. V. I. Shynkarenko, I. M. Demidovich Natural Language Texts Authorship Establishing Based on the Sentences Structure, in: Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2022), Volume I: Main Conference, Gliwice, Poland, May 22-23, 2022, pp. 328-337.
4. I. Demidovich, V. Shynkarenko, O. Kuropiatnyk, O. Kirichenko, Processing Words Effectiveness Analysis in Solving the Natural Language Texts Authorship Determination Task, XVI International Scientific and Technical Conference (CSIT'2021). September 22-25, 2021, Lviv, Ukraine.
5. Плющ М. Я. Граматика української мови. Морфеміка. Словотвір. Морфологія. Підручник / М. Я. Плющ.
6. Viktor Shynkarenko and Olena Kuropiatnyk. Constructive Model of the Natural Language. Acta Cybernetica, 23(4):995-1015, 2018.
7. Foltýnek, Tomáš & Meuschke, Norman & Gipp, Bela. Academic Plagiarism Detection: A Systematic Literature Review. ACM Computing Surveys. 52. 2019. 1-42. 10.1145/3345317.

8. Mohammad Al-Smadi, Zain Jaradat, Mahmoud Al-Ayyoub, and Yaser Jararweh. 2017. Paraphrase identication and semantic text similarity analysis in arabic news tweets using lexical, syntactic, and semantic features. Inf. Process.Manag. 53, 3 (2017), 640–652. DOI:10.1016/j.ipm.2017.01.002.
9. Asli Eyecioglu and Bill Keller. 2015. Twitter paraphrase identication with simple overlap features and SVMs. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval'15). 64–69.
10. Helena Gómez-Adorno, Grigori Sidorov, David Pinto, and Ilia Markov. 2015. A graph based authorship identication approach — Notebook for PAN at CLEF 2015. In Proceedings of the Conference and Labs of the Evaluation Forum and Workshop (CLEF'15).
11. Tenney, Ian et al. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. ArXiv. 1-17. https://doi.org/10.48550/arXiv.1905.06316.
12. Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content Attention Model for Aspect Based Sentiment Analysis. In Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1023–1032. https://doi.org/10.1145/3178876.3186001.
13. Ravfogel, Shauli et al. 2020. Unsupervised Distillation of Syntactic Information from Contextualized Word Representations. BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP.
14. Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19). Association for Computing Machinery, New York, NY, USA, 985–988. https://doi.org/10.1145/3331184.3331303.
15. Wang, Wei, et al. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. arXiv. https://doi.org/10.48550/arXiv.1908.04577
16. Eva Hasler, Felix Stahlberg, Marcus Tomalin, Adri de Gispert, and Bill Byrne. 2017. A comparison of neural models for word ordering. arXiv. https://doi.org/10.48550/arXiv.1708.01809.
17. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv. https://doi.org/10.48550/arXiv.1802.05365.
18. White, L., Togneri, R., Liu, W., Bennamoun, M. 2019. Sentence Representations and Beyond. In: Neural Representations of Natural Language. Studies in Computational Intelligence, vol 783. Springer, Singapore. 93–114. https://doi.org/10.1007/978-981-13-0062-2_5.
19. Liu, Z., Lin, Y., Sun, M. 2020. Sentence Representation. In: Representation Learning for Natural Language Processing. Springer, Singapore. 59–89. https://doi.org/10.1007/978-981-15-5573-2_4.
20. Cheng, Z., Yuan, C., Li, J., Yang, H. 2018. TreeNet: Learning Sentence Representations with Unconstrained Tree Structure. In IJCAI. 4005-4011.
21. Fereshteh Jafariakinabad and Kien A. Hua. 2022. A Self-Supervised Representation Learning of Sentence Structure for Authorship Attribution. ACM Trans. Knowl. Discov. Data 16, 4, Article 68 (August 2022), 16 pages. https://doi.org/10.1145/3491203.
22. Conneau Alexis, Kiela Douwe, Schwenk Holger, Barrault Loïc, and Bordes Antoine. 2017. Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 670–680.
23. Walenski, M., Europa, E., Caplan, D., & Thompson, C. K. 2019. Neural networks for sentence comprehension and production: An ALE-based meta-analysis of neuroimaging studies. Human brain mapping, 40(8), 2275-2304. https://doi.org/10.1002/hbm.24523.
24. Liu, P., Chang, S., Huang, X., Tang, J., Cheung, J. C. K. 2019. Contextualized non-local neural networks for sequence learning. In Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33, No. 01. 6762-6769. https://doi.org/10.1609/aaai.v33i01.33016762.

Віктор Шинкаренко **Viktor Shynkarenko**	DrS, Professor of Computer and information technologies department, Ukrainian State University of Science and Technologies, Dnipro, Ukraine, e-mail: shinkarenko_vi@ua.fm https://orcid.org/0000-0001-8738-7225	Український державний університет науки та технологій
Інна Демидович **Inna Demidovich**	PhD student of Computer and information technologies department, Ukrainian State University of Science and Technologies, Dnipro, Ukraine, e-mail: 2019demidovichinn@gmail.com https://orcid.org/0000-0002-3644-184X	Український державний університет науки та технологій