Computer Linguistic Systems Design and Development Features for Ukrainian Language Content Processing

Victoria Vysotska

Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, 79013, Ukraine

Abstract

The paper describes the developed IT (information technology) processing of Ukrainian-language text content, unlike the existing ones, which supports the modularity principle of the typical CLS (computer linguistic system) architecture for solving a specific NLP (natural language processing) problem and analysing a set of parameters and metrics of the system's functioning by the target audience behaviour. The general structure of CLS for the processing of text content in the Ukrainian language and the conceptual scheme/model of the functioning of a typical CLS based on the modelling of the interaction of the main processes and components of the system were developed, which made it possible to improve IT intellectual analysis of the text flow based on the processing of information resources. The peculiarities of the design and development of computer linguistic systems are analysed based on the definition of the main stages such as grapheme, morphological, lexical, and syntactic-semantic analysis/synthesis of the Ukrainian-language text for a specific NLP problem solution. The formulation of the problem of processing the Ukrainian-language text based on the definition of the functional features of the intellectual analysis of the text flow was made and specified. The general analysis of the problem of analysis of the Ukrainian-language text and the definition of the main problems of the processing of the Ukrainian-language text made it possible to formulate the main stages and requirements for the project of a typical CLS solution of a specific NLP problem. Identification of the main characteristics of CLS and justification of the project implementation of a typical CLS made it possible to determine the expected effects of the corresponding project implementation. Based on the analysis of the input/output streams of the content of the computer linguistic system, the functional requirements for the project of a typical CLS, its software modules, network, software and technical tools of IS software implementation are defined and formulated.

Keywords

Computer linguistic system, intelligent search system, NLP, Ukrainian language, information resource, system performance metrics, machine learning, target audience

1. Introduction

The Internet, mobile applications, information systems, and social networks – bottomless sources of information are constantly present around us. On the one hand, it helps to solve many everyday and professional tasks, but on the other hand, it complicates the life process due to the need to navigate in this chaos of information space. In addition, it is a source of manipulation of people's consciousness through propaganda, fakes both in everyday life (for example, through advertising) [1-3], and in information warfare, etc.

Nowadays, much online information is subject to regional censorship in certain territorial regions due to political, economic, social, religious and other factors, such as to control or manage the opinion of the people of that region. The reasons can be various factors. At the same time, fake information is spread both purposefully and randomly/chaotically in the Internet environment. It is easy for an average person to get lost and navigate in this mass of content flow with opposing facts and causes of events/phenomena. It is unethical, illegal and impractical to control exactly what to show or hide (to censor content) among Internet content to the average user in democratic states. This is one of the first steps in the transition to totalitarianism. But providing

COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

victoria.a.vysotska@lpnu.ua (V. Vysotska)

0000-0001-6417-3689 (V. Vysotska)

information, for example, to journalists about a possible thematic fake for conducting a journalistic investigation or warning the average reader about the possibility of disinformation in this content/resource is, on the one hand, support for freedom of speech, and on the other hand, giving a person the opportunity to choose what to believe and what not to believe. At the same time, it provides an opportunity to gain an understanding of events and orientation in a large flow of information both for solving everyday tasks and adjusting business strategies, etc. Significant and massive dissemination of (dis)information against the background of the war in Ukraine without appropriate analysis potentially leads to panic among the relevant stratum/region of the population, significantly affecting the process of adjusting plans/strategies of business, social services, etc. Against the background of the information war, a lot of time and resources are spent on the appropriate collection, analysis and formation of appropriate conclusions regarding the content of the relevant content. This is also influenced by the language of the information, which may partially/significantly change the content when translated. CLS will not be able to completely replace human activity in this direction. However, it can be a significant helper for quickly forming relevant bases of such content and reacting to local changes or the dynamics of changes in the content flow, marking certain content as potentially fake in a certain percentage. The difficulty lies in the language of the content itself. In comparison with English-language content, Ukrainian/russian languages are quite difficult to automatically process, especially the extraction and analysis of semantics [4-6]. Today, there are many computer linguistic systems for various purposes, even for processing Ukrainian-language textual content. But these are usually commercial projects of a closed type (there are no publications or access to the administrative part) and most often they are foreign projects. There seem to be a lot of publications to understand how the natural language processing process generally works, especially for English texts. However, applying these models, methods, algorithms and technologies directly to Ukrainian-language textual content does not lead to almost any positive result. Already at the level of morphological analysis, a significant conflict arises between the developed methods and the incoming Ukrainian text - the output is not correct. For example, for a simple Porter algorithm (stemming) without a corresponding modification, it will not be correct to separate the base of the word from the inflexion, which will lead to incorrect identification of the keywords of the texts, which in turn affects any NLP task where it is necessary to quickly identify a set of keywords (rubrication, search, annotation, etc.). Determining the main processes and features of the linguistic analysis of Ukrainian-language texts will greatly facilitate the stages of processing the text flow of content such as integration, support and content management. In turn, the adaptation of the processes of intellectual analysis of text content with the identification of functional requirements for the corresponding modules of the CLS will lead to the possibility of developing a typical architecture of such systems based on the principle of modularity (adding components depending on the content of the NLP task and the purpose of the CLS).

2. Related works

To solve most NLP problems, the words of the relevant textual content are processed, analysed and researched as a result of the work of one or more authors in a specific dialect of a certain language (the best measure of the variation of the author's speech characteristics), of a certain style (dialogue/monologue) and genre (an auxiliary measure of variation features of the author's speech) at a certain time, in a certain place, for a certain purpose/function [7-15]. There are more than 7 thousand languages in the modern world. NLP algorithms are most useful when they are applied to many languages. Most NLP tools are usually developed for the official languages of large industrialized countries (English, Chinese, German, russian, etc.) and this is a very limited range of natural languages (out of a couple of dozen). For most of the world's languages, either no NLP tools are developed, or no significant attention is paid (surface development) or highly specialized commercial projects. But usually, most of the content consists of text in more than one language. Therefore, it is advisable to support the development of NLP tools in several languages

according to their purpose, for example, for the classification of text content in the scientific and technical Ukrainian language, it is advisable to use a combination of NLP techniques not only for the analysis of the Ukrainian language but at least English due to the presence of specific terminology and habits speakers to use English analogues from the subject area. In addition, most natural languages have several regional, social or professional dialects or slang/slang. This makes it possible to maintain appropriate dictionaries not only for content classification but also, for example, for identifying the probable author of the corresponding text. At the same time, some languages are constantly developing and changing at different speeds, which significantly affects the quality of processing new modern content. Simply changing the RE-rules will not solve the problem, as all the old contents of the content will not be rewritten. It is then necessary to introduce the concept of classification of old/new RE-rules, for example, the morphological processing of words and the support of relevant dictionaries. Any linguistic text analysis includes the main NLP sub-processes (NLP levels) of linguistic analysis (Fig. 1) [16-21].

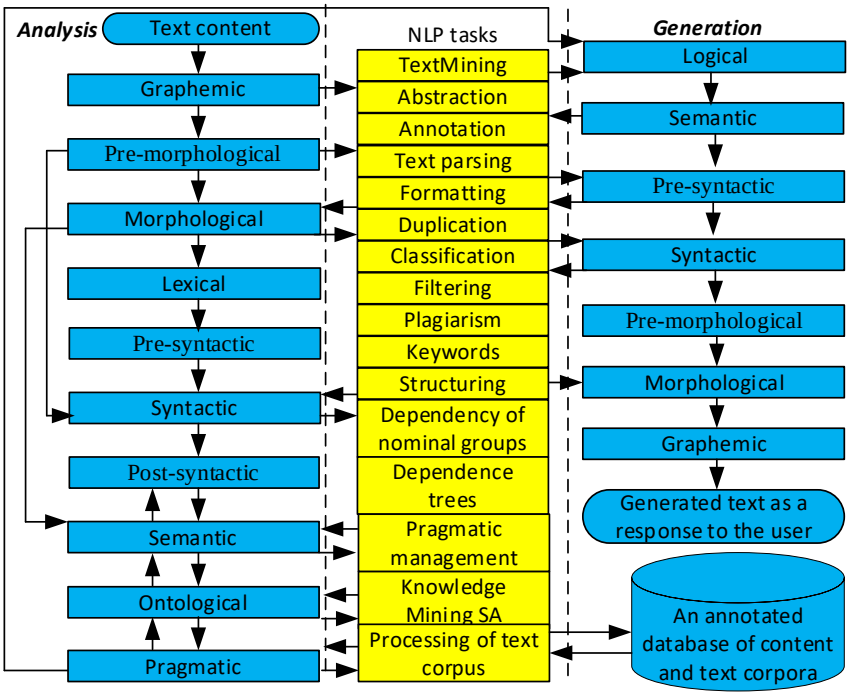

Figure 1: Structural-linguistic scheme of linguistic text analysis

For each language, the difficulty lies in the implementation of syntactic analysis, but there are languages such as Ukrainian, the difficulty lies in the implementation of morphological analysis, on which other NLP levels of linguistic analysis depend (Table 1) [18-24]. The development of full-fledged detailed dictionaries of SA (subject area), the bases of words and their features of declension depending on the part of speech and their features (gender, tense, plural/singular), taking into account the alternation of letters, will greatly facilitate the MA (morphologic analysis) text of the Ukrainian language. This will allow for a more accurate syntactic (sentence structure) and semantic (used concepts) analysis to prepare knowledge extraction from the relevant text through pragmatic analysis (correctness of the purpose of using concepts).

Table 1
Stages of linguistic analysis of textual information [1-24]

N	Analysis	Explanation
1	Graphematic or grapheme (GA)	Selection/combination of syntactic (headings, main text, inserts, footnotes, comments, etc.) and/or structural (paragraphs, sentences, individual words and punctuation marks) units of text content with subsequent filtering
2	Pre-morphological	Separation/combination of inseparable, unchanging, stable word combinations into one linguistic unit:: *_Залізний_Порт_* (_Zalizniy_Port_, city), *_Червона_Калина_*

		(_Chervona_Kalyna_, prospect), *_Нью_-_Йорк_* (_New__York_), *_Івано_-_Франківськ_* (_Ivano__Frankivsk), *_і_так_далі_* (_and_so_on), *_яким_-_небудь_* (_any__any_), *_таким_чином_* (_such_a_way_), *_будь_-_хто_* (_any__who_), etc.
3	Morphological (MA)	Determining the normal form of a word form, and vice versa, generating a word form from the normal form, taking into account the location in the syntactic tree of dependence for matching words in a sentence.
4	Pre-syntactic	Unification of individual lexical units into one syntactic as stable word combinations (for example, idioms and metaphors of how *бити байдики* [to beat idlers]), поділ на окремі (наприклад, *словоформа* [word form], *криптовалюта* [cryptocurrency], *відеомонтаж* [video montage], але не *качкодзьоб* [platypus], *водогін* [water supply], *зорепад* [shooting star], *чорнозем* [black soil]) and segmentation.
5	Syntactic (SA/SYA)	Deploying sentence dependency syntactic trees with word matching. Transforming a tree into a linear order of words with parameters taken into account.
6	Post-syntactic	Normalization of syntactic trees of sentence dependence, taking into account and clarifying the parameters of words and their meaningful load in the expression.
7	Semantic (CEM)	Refinement of word relationships in a tree for knowledge extraction or answer generation taking into account the semantic roles of noun groups and their actions/events.

Logical derivation in the form of a set of natural text content based on linguistic analysis of the input text is a common phenomenon for any statistical text analysis (sentiment analysis, tonality analysis, content analysis, etc.), dialogue systems, QA systems, abstract/annotation generation systems /digests, etc. Natural text is usually partially structured and formalized information with the presence of hints, defaults, abbreviations, incompleteness, noise, inaccuracies, obfuscations, etc., especially for syntactic groups of languages like Slavic languages. Identifying and processing such constructions is a complex process (for example, in Ukrainian *пташка сидить на столі* (the bird sits on the table), although it can *стояти* (stand), *кішка* (the cat) can *сидіти*, *лежати* and *стояти* (sit, lie and stand), *стакан стоїть на столі*, (the glass is on the table), and *тарілка лежить на столі* (the plate lies on the table), etc., in turn, in English it is usually used for all the cases listed, the verb *є* – is). Also, interesting constructions of the spoken Ukrainian language are *шмигати носом* (to swish the nose as have a runny nose), *зробити ноги* (to make legs as run away), *говорити абсурдні речі* (to say absurd things as talk nonsense), *дати прочухана* (to give a thumbs up as quarrel and order the child/animal for something), *золота молодь* (golden youth as young people, whose future was arranged by baddies), *зробити ляпсус* (make a splash/lapsus as make a slip), *пам'ять як у рибки* (to have a memory like a fish or short memory how to forget quickly), *піти по воду* (fetch water), or *піти за водою* (to go for water or go get water as to walk along the course of the river), *стригти купони* (to cut coupons as easy to earn money), *вештатися містом* (to walk around the city as strolling aimlessly through the city), *зелена капуста* (green cabbage as dollars), *теревенити* (*базікати*) *по телефону* (chatter on the phone as talking on the phone for a long time without a purpose), *тримати ніс за вітром* (keep your nose out of the wind as respond to circumstances in a timely and efficient manner), *кмітливий пуцьвірінок* (smart little piglet or on a clever little porcupine as clever little child), *дати телеф* (give the phone as give a phone number), etc.

3. Models and methods

3.1. Grapheme analysis and synthesis of the Ukrainian text

The basis of any grapheme text analysis [6] is the identification of punctuation marks, abbreviations, abbreviations, capital letters in proper names, etc.

An apostrophe in Ukrainian and English is not a delimiter, although there is a similar delimiter - a single quotation mark for separating quotations. In English, it's easier - an apostrophe is found at the end of a noun (determining belonging) or a set of separate symbols near the letter s or to

shorten some verb forms. In the Ukrainian language, the apostrophe is usually found in the roots of words and their variations, in particular, after labial consonants (*б* [b], *в* [v], *м* [m], *п* [p], *ф* [f]) in the roots of some words, after *р* [r] at the end of a syllable, after prefixes before a hard consonant at the beginning of the root and after the first parts of some compound words.

The presence of double quotation marks indicates either a proper name, a quote, or sarcasm. Each of the listed linguistic units carries its content load and is a different engine for generating a parsing syntactic tree and defining key words as stable phrases (*кінотеатр* «*Зірка*» [cinema "Star"], *зірка на кінотеатрі* [star in the cinema], *зірки готелю* [hotel stars] або «*золота рибка*» ["goldfish"] as a human trait (bad memory or fulfilling wishes without mutual benefit depending on the context) or *золота рибк*а [goldfish] as a fish in an aquarium, etc.). Sometimes proper names coincide with commonly used words (група *Мертвий півень* [the Dead Rooster group], students (last name as first name) *Оксана Тарас* [Oksana Taras], *Сергій Семен* [Serhiy Semen], *Тарас Лема* [Taras Lema], *Михайло Сало* [Mykhailo Salo] (*немає сьогодні Михайла Сала*) [Mikhailo Salo is gone today] та *Софія Тесля* [Sofiya Teslia, where last name Teslia is Carpenter as a profession] or *Петро Кравець* [Petro Kravets, where last name Kravets is Tailoras a profess], singers *Катя Чилі* [Katya Chile - last name] and *Альона Вінницька* [Alyona Vinnytska as city in Ukraine], actor *Девід Духовний* [David Dukhovny, where last name translate as Spiritual], проспект *Червоної Калини* [Chervonaya Kalina avenue as Red Viburnum avenue at translation] or проспект *Свободи* [Svoboda avenue as Freedom avenue at translation], вулиця *Перемоги* [Peremoha Street as Victory Street], etc.), but have different meanings. Some lexemes are not subject to grammar (1979, 12%, кг [kg], млн [million], км [km], etc.). Therefore, grapheme analysis allows the labelling and classification of lexemes that go beyond the standard linguistic analysis of grammar. It is not possible to define and supplement dictionaries with such lexemes in advance. Only using the rules of pre-parsing the text using machine learning methods with the teacher. It is impossible to support dictionaries of all possible names, geographical names, abbreviations, numerical values, etc. When identifying a lexeme as a grapheme from unknown, undefined elements, an intermediate dictionary is formed, which must be reviewed by the moderator and labelled accordingly. However, it is easier to maintain a set of rules for identifying non-standard tokens based on grapheme analysis with partial use of dictionaries of frequently used widespread exceptions.

Additional points of grapheme analysis are grapheme identifications in the form of special signs, such as the end of a paragraph, the presence of figures, tables, formulas, etc., the presence of alphabetic characters of another specific language, HTML tags, formatting elements such as headings, alignment, emoticons, etc. [6]. The result of grapheme analysis is the construction of the grapheme structure of the text from the classified sets of grapheme chains and connections between them [25-29].

$$C_\alpha = \alpha(D_\alpha, R_\alpha, X), \quad (1)$$

where X is the input text; C_α is a description of the grapheme structure of the input text; α is grapheme analysis operator (grapheme identification, classification and marking); D_α is dictionaries of punctuation marks, abbreviations, abbreviations, geographical names, etc.; R_α is grapheme analysis rules, including regular expressions.

3.2. Morphological analysis and synthesis of the Ukrainian text

The main goal of MA is to identify the normal word form $f_{\beta i}^{n}$ for any word form $w_{\beta i}^{t}$ in the input text, and the corresponding tuple of descriptive criteria and parameters $c_{\beta i}$ (part of speech, gender, number, case, etc.) [6]:

$$C_\beta = \beta(C_\alpha, D_\alpha, R_\alpha, X),\ C_\beta = \{c_{\beta 1}, c_{\beta 2}, \dots, c_{\beta n},\}, \quad (2)$$
$$c_{\beta i} = (w_{\beta i}^{t}, r_{\beta i}^{w}, f_{\beta i}^{n}, r_{\beta i}^{f}, p_{\beta i}^{w}),\ p_{\beta i}^{w} =< n_{\beta i}^{p}, v_{\beta i}^{p} >,$$

where X is the input text; C_β is a set of tuples of descriptive criteria and parameters for each word $w_{\beta i}^{t}$ of the input text; $c_{\beta i}$ is a tuple of descriptive criteria and parameters for the *i* word of the input text; β is morphological analysis operator; C_α is the result of GA; D_α are dictionaries of

words in normal form or word bases with descriptive parameters; R_α is MA rules; $r_{\beta i}^{w}$ is part of the language of the word $w_{\beta i}^{t}$ of the input text; $f_{\beta i}^{n}$ is the normal form of the word $w_{\beta i}^{t}$ of the input text; $r_{\beta i}^{f}$ is a part of the language of the normal form $f_{\beta i}^{n}$ (for example, for an adverb as a verb form); $p_{\beta i}^{w}$ is a collection of morphological parameters and criteria $w_{\beta i}^{t}$; $n_{\beta i}^{p}$ is the name of the morphological parameter of the word (declension, tense, number, gender, brevity of the adjective form and other parameters of the words of the corresponding natural language); $v_{\beta i}^{p}$ is the specific value of the morphological parameter of the word of the input text of the corresponding natural language.

The variety of dependence in different languages on the location of a specific word form with the corresponding part of the language greatly complicates the linguistic analysis of the text. Preprocessing the words of the input text through MA reduces the list of words that need to be worked on at the next stage (for example, the word *інформація* [information], not all variants, declension and number change formations). Thus, for nouns, they choose to record the form *слово* → <*частина мови*, *рід*, *відмінок*, *істота*, *число*> [word → <part of speech, gender, case, creature, number>] according to different methods, for example, for *донька* [a daughter] they write [6]:

1) 1593 → < 01 0202 0301 0601 0901 >;

2) донька → < і, рід = ж, число = од, відмінок = нз, істота = і >;

3) донька → < ім, ж, од, наз, іст >.

For the first point, each word has its number in the dictionary or it is converted to a number by matching symbols in ASCII tables (for example, the word *донька* [daughter] in the dictionary has the number 1593) [6]. The noun corresponds to the part of speech value 01, the gender parameter corresponds to 02, and the feminine gender is also 02, so we get 0202. Nouns do not change gender, but verbs and adjectives formed from them in the Ukrainian language can change gender depending on the content [6]. Therefore, one-word form can be attributed to several tuples (homonymy), for example:

1) *доньки* → донька → < ім, ж, **од**, **род**, іст >
доньки → донька → < ім, **мн**, **наз**, іст >
2) *мати* → мати → < **ім**, ж, од, наз, іст >
мати → мати → < **д**, перехідне, 1 дієв, недок >
3) *опали* → опал (камінь) → < **ім**, ч, мн, наз, іст >
опали → опасти → < **д**, мин, мн, 3 ос, док. >
4) *ягуари* → ягуар (тварина) → < ім, ч, од, наз, **іст** >
ягуар → ягуар (машина) → < ім, ч, од, наз, **неіст** >
5) *замок* → замок (будівля) → < ім, ч, од, наз, неіст >
замок (інструмент) → < ім, ч, од, наз, неіст >
6) *дракон* → дракон (тварина) → < ім, ч, од, наз, **іст** >
дракон (корабель) → < ім, ч, од, наз, **неіст** >
6) *кішки* → кішка (тварина) → < ім, ж, **од**, **род**, **іст** > або < ім, **мн**, **наз**, **неіст** >
кішки → кішка (частина взуття) → < ім, ж, **од**, **род**, **іст** > або < ім, **мн**, **наз**, **неіст** >
але *кишки* → кишка (частина тіла) → < ім, ж, **од**, **род**, **іст** > або < ім, **мн**, **наз**, **неіст** >
6) *коси* → коса (зачіска) → < ім, **мн**, **наз**, **неіст** > або < ім, ж, **од**, **род**, **іст** >
коси → коса (мілина) → < ім, **мн**, **наз**, **неіст** > або < ім, ж, **од**, **род**, **іст** >
коси → коса (інструмент) → < ім, **мн**, **наз**, **неіст** > або < ім, ж, **од**, **род**, **іст** >
коси → коса (селезіка) → < ім, **мн**, **наз**, **неіст** > або < ім, ж, **од**, **род**, **іст** >

Dictionary morphological analysis is usually used (Fig. 2), that is, a complete dictionary of words is stored. The disadvantages are [6]:

1) it is impossible to work out words that are not in the dictionary;

2) the bulkiness of information (a lot of searches and comparisons) and the excess of information (the presence of several variants of IIS (intelligent information search) results) of the data for processing words in the text.

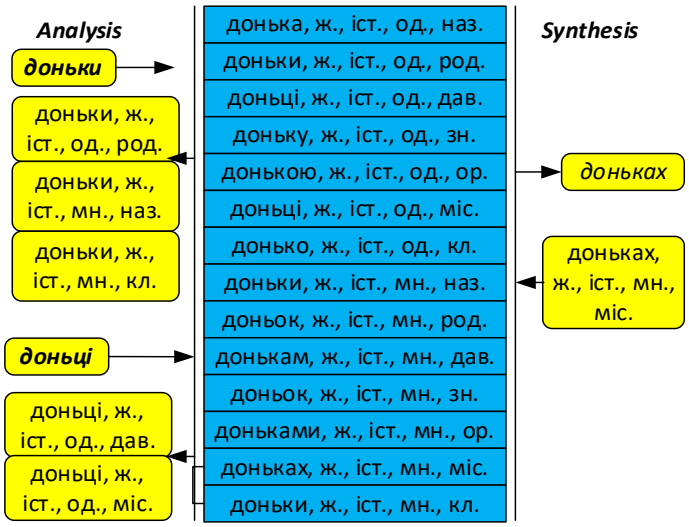

Figure 2: Structural-linguistic scheme of a word presentation example in a dictionary

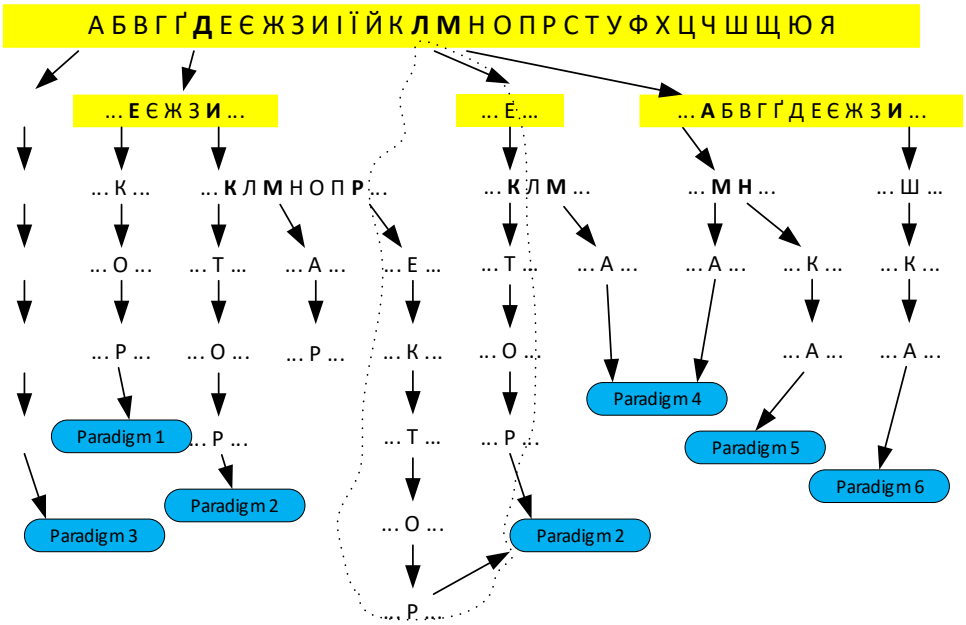

Figure 3: Structural-linguistic diagram of an example of building a prefix tree

The modern Ukrainian language has more than 256 thousand words [4]. The noun has 7 cases, that is, it takes 14 forms, and adjectives - 24, that is, the presence of different inflexions and in some cases the alternation of letters. There are many synonyms, for example, horizon has 12. The number of word forms of adverbs and adjectives as verb forms reaches 300 (about 25 forms per paradigm). All this complicates MA (morphologic analysis). The transition to the tree partially solves this problem (Fig. 3). Usually, MA is carried out symbol by symbol from the tree root. This method is difficult to implement - you have to take into account all possible options from all possible words. Therefore, the best way is to combine these two methods with the parsing of symbols from the end of the word (identification of inflexions by the tree of all possible endings to determine the part of speech, separation of the root and identification of the root in the dictionary). In [30], a static tree of endings for words from the Aspell database (about 1.4 million forms of Ukrainian words) within 1-11 characters was built. Thanks to the author's research [31], inflexions can be ranked by frequency of use and separated into blocks belonging to parts of speech (Table 2) [6]. The majority of inflections with a total specific weight of use of less than 1% belong in most cases to nouns, in particular, ґ [g] (4) in the genitive case and plural – *ґирлиґ* from *ґирлиґа* [girlyga or shepherd's crook as stick, often bent at the end, used by shepherds and old

people], *дзиґ* in the genitive case from *дзиґи* [dziga or spinning top as a toy that maintains balance on a sharp tip by rapidly rotating around its axis], *зиґзаґ* [zigzag], *меґ* [tag] [6]. Similarly, this applies to inflexions ц [ts] (34), ш [shch] (110), ф [f] (214), б [b] (281), п [p] (341), ж [zh] (353), з [z] (581), г [h] (636), л [l] (754), с [s] (914), ч [ch] (959), д [d] (1038), н [n] (2531), р [r] (2709) [30].

Table 2
Static table of common Ukrainian inflections [30]

Inflexion	*всь* [vs'] (10016)	*єте* [yete] (11137)	*ним* [nym] (19093)	*ві* [vi] (22543)	*ій* [iy] (33241)
т [t] (2980)	*ню* [nyu] (10075)	*єш* [yesh] (11138)	*ної* [noyi] (19098)	*ись* [ys'] (22656)	*мо* [mo] (33568)
к [k] (7299)	*вся* [vsya] (10076)	*ють* [yut'] (11222)	*теся* [tesya] (19103)	*ну* [nu] (23125)	*ї* [yi] (34702)
кою [koyu] (7497)	*лась* [las'] (10229)	*ймо* [ymo] (11229)	*тесь* [tes'] (19105)	*ться* [t'sya] (25036)	*му* [mu] (35023)
істю[istyu] (7598)	*лася* [lasya] (10230)	*йте* [yte] (11230)	*еся* [esya] (19105)	*ься* ['sya] (25211)	*ою* [oyu] (39616)
ість [ist'] (7606)	*лось* [los'] (10231)	*є* [ye] (11466)	*ному* [nomu] (19112)	*ося* [osya] (30769)	*х* [kh] (61506)
стю [styu] (7648)	*лося* [losya] (10233)	*ку* [ku] (11624)	*есь* [es'] (19114)	*ось* [os'] (30788)	*ми* [my] (62080)
ості [osti] (7636)	*ася* [asya] (10235)	*шся* [shsya] (11775)	*ш* [sh] (19163)	*ими* [ymy] (31121)	*е* [e] (66988)
сть [st'] (7688)	*ась* [as'] (10239)	*ті* [ti] (12596)	*нім* [nim] (19333)	*их* [ykh] (31127)	*а* [a] (68134)
Юся [yusya] (8044)	*тись* [tys'] (10366)	*ям* [yam] (15717)	*ній* [niy] (19549)	*ий* [yy] (31136)	*й* [y] (77109)
юсь [yus'] (8047)	*лись* [lys'] (10337)	*ів* [iv] (15898)	*ах* [akh] (20023)	*им* [ym] (31166)	*ю* [yu] (80877)
сті [sti] (8731)	*лися* [lysya] (10338)	*ом* [om] (17018)	*ти* [ty] (20025)	*ім* [im] (31343)	*і* [i] (90275)
нням [nnyam] (8975)	*тися* [tysya] (10379)	*ові* [ovi] (17191)	*ами* [amy] (20106)	*ого* [oho] (31389)	*о* [o] (90454)
ння [nnya] (9001)	*ало* [alo] (10465)	*ло* [lo] (17238)	*ам* [am] (20154)	*ої* [oyi] (31421)	*у* [u] (94504)
нню [nnyu] (9054)	*ав* [av] (10547)	*ли* [ly] (17711)	*не* [ne] (20257)	*го* [ho] (31445)	*сь* [s'] (111459)
ням [nyam] (9434)	*ала* [ala] (10610)	*ла* [la] (17945)	*ною* [noyu] (20280)	*ому* [omu] (31585)	*м* [m] (119779)
ня [nya] (9765)	*али* [aly] (10666)	*ний* [nyy] (19042)	*мося* [mosya] (20532)	*ні* [ni] (31679)	*и* [y] (123402)
ями [yamy] (9844)	*ати* [aty] (10819)	*ними* [nymy] (19089)	*мось* [mos'] (20536)	*те* [te] (32651)	*ся* [sya] (148160)
ях [yakh] (9855)	*ка* [ka] (11029)	*ного* [noho] (19090)	*на* [na] (21328)	*в* [v] (32681)	*ь* ['] (151355)
нні [nni] (9909)	*ємо* [yemo] (11136)	*них* [nykh] (19092)	*ися* [ysya] (21940)	*ть* [t'] (33055)	*я* [ya] (164062)

Words are grouped by paradigms (sets of all postfixes based on [6, 30-31] and morphological parameters for all word forms of the corresponding word, for example, the words *лектор* [lecturer] and *професор* [professor]). Then they store a single tape in the postfix tree. Grouping by paradigm depends on the features of words, their morphological parameters and the NLP task. Thus, the words *лектор* [lector] and *вектор* [vector] do not belong to the same paradigm due to different inflexions in the accusative case. But the words *мама* [mother] and *лема* [lemma] can enter the same paradigm, if we consider the meaning of being/non-being for a specific NLP task (it is not necessary for rubrication, but it is necessary for PA – pragmatic analysis). Character-by-character analysis of the word form from the root of the tree requires the preservation of an array of pointers to the next vertex - a specific letter [6, 30-31]. It is necessary to store the alphabet of the language in each vertex. But for the Ukrainian language, more than 46 billion pointers are needed to save all chains of 8 letters. Some of them are cut off (for example, there are no words for a soft sign). Therefore, arrays of letters of the alphabet at the top are densely filled near the root of the tree, and closer to the leaves - sparse. Also, some word postfixes are unique to parts of subtrees, so they are stored as a tape. But all this will not allow us to take into account all possible variants of subtrees and cause unnecessary load on the MA process - trees usually store all words in their normal form. If all options are saved, the declension of words in such trees, taking into account the alternation of symbols, leads to an increase in the excess of data storage. Preserving postfix trees and morphological analysis from the end of words by their inflexions/postfixes will reduce the number of operations [6, 30-31]. For example, to determine keywords, it is enough to consider only words from the noun group (without pronouns), then all endings (postfixes) characteristic of verbs will significantly reduce the number of words that need to be analyzed. For the rubrication of the incoming texts, it is sufficient to identify the noun groups and conduct the corresponding MA.

3.3. Lexical analysis of the Ukrainian-language text

The process of lexical analysis consists of the analytical analysis (segmentation) of the input array of text after a detailed morphological analysis to form collections of tokens (sequences of symbols according to appropriate patterns) as lexemes with subsequent identification of their types [6]. A lexeme is usually a word, word form, or phrase as a meaningful lexical unit of an expression/sentence [6]. Sentence segmentation is another important step in text processing [25-29]. The LA (lexical analysis) module is a scanner, tokenizer, or lexical analyzer, depending on the purpose of the NLP task. Not all tokens are tokens, such as the number 13, a mathematical expression, a punctuation mark, etc. The most useful symbols for segmenting text into sentences are punctuation, such as periods, question marks, exclamation marks, etc. Question/exclamation marks are relatively unambiguous markers of sentence boundaries. Periods, on the other hand, are more ambiguous, such as between a sentence marker and an abbreviation marker such as mln or r. The last contraction illustrated a complex case of this ambiguity, in which the dot marked r is both a contraction of the word year and a sentence boundary marker. For this reason, the tokenization of sentences and words should be done in parallel and simultaneously. In general, sentence tokenization techniques work by building a binary classifier (based on a sequence of rules or machine learning) that decides whether a dot is part of a word or a sentence boundary marker. In making this decision, it helps to find out whether the dot belongs to a commonly accepted abbreviation; thus, a glossary of abbreviations is useful. The most modern methods of sentence tokenization are based on the use of machine learning. Token identification through token type classification in the context of a specific grammar/language. If the lexeme as a language token cannot be identified according to the corresponding grammar, then it is checked with a dictionary of special symbols, mathematical signs, etc. If in this case it cannot be identified, then it is marked as a special error token. A token is a patterned structure with a type/class identifier. Identification takes place in two stages in the form of a finite automaton - scanning for regular expressions and evaluation for further classification by type and transmission to the input to the parser. Sometimes, for simplicity, a parser is combined with a lexical one for some NLP tasks. Then the parsers perform the analysis by parsing the text in two stages (Fig. 4) [6, 25-29]: they identify meaningful lexemes (LA) and generate a sentence parsing tree (dependencies of the identified lexemes).

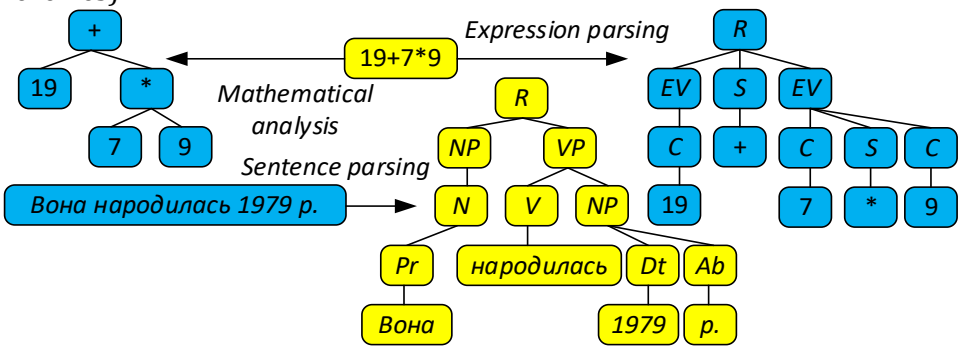

Figure 4: Examples of parsing expressions and generating a dependency tree

A token is an atomic meaningful object from a sequence within [1, N] characters [6, 25-29]. Identifies tokens based on regular expressions and by location in character set/sentence and context. This is not grapheme analysis as separating a group of characters between punctuation marks. Tokens are identified by the rules of the lexer, taking into account already grammatical features from the previous MA step, according to the natural language of the input text, in particular:

- Marking a set of input text characters into a set of tokens;
- Identification of a separate token as a logical linguistic unit of the text (word, mathematical sign, number, punctuation mark, etc.);
- Establishing a relationship between a token and a token - a specific token text (“для” ["for"], "1979", "+", “змінна” ["variable"], “р.” ["y." as year], ";" etc.);

- Identification of additional attributes of the token (for example, a period as a sentence boundary or part of a contraction);
- Formation of the tuple of tokens as input information for SYA.

The lexical analyzer does not check the correctness of the connections in the tuple of tokens, but only identifies, labels and classifies them (Fig. 5) [6]. The lexical analyzer recognizes parentheses, punctuation marks and mathematical symbols as characters, but does not check whether each character "(" corresponds to another – ")", and each mathematical character is between specific two numbers [6]. Such functions are inherent to the syntactic/semantic parser/analyzer in the relevant NLP tasks.

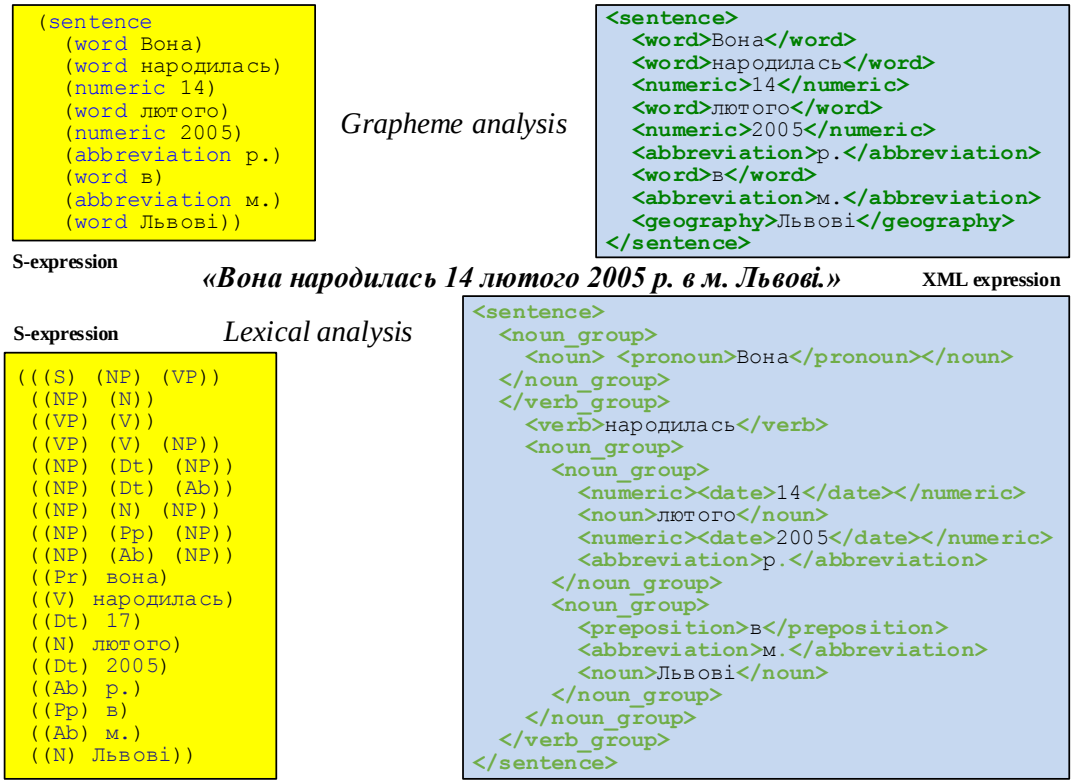

Figure 5: Examples of the results of S/XML expressions for grapheme and lexical analyses of the sentence «Вона народилась 14 лютого 2005 р. в м. Львові.» [«Vona narodylas' 14 lyutoho 2005 r. v m. L'vovi.»] ("She was born on February 14, 2005 in Lviv.")

3.4. Syntactic analysis and parsing of the Ukrainian text

To analyze the syntax of the text, the grammars of N. Chomsky, system grammars of M.A.K Halliday, subordination trees and constituent systems of the researcher A.V. Hladkyi, extensions of the Petri net of transitions are usually used etc. An effective tool for English syntactic modelling (rules for forming sentences from word forms) is generative grammar, which was started in the works of the American linguist N. Chomsky. According to his theory, word forms are denoted by terminal symbols, syntactic categories by non-terminal symbols, and the rules of derivation of sentences (syntactic structure) by production rules and presented in terms of immediate constituents. The scientist applied a formal analysis of the scheme of sentences to distinguish the syntactic scheme of the expression regardless of the meaning. N. Chomsky's research was continued by the linguist A. V. Hladkyi, who used the constituent system and syntactic trees of dependencies for the sentences analysis in natural language. The scientist developed the basics of syntax modelling based on syntactic groups to identify constituent phrases as units for generating a dependency tree. This approach made it possible to combine the advantages of dependency trees and direct components for the Slavic languages processing and analysis. Linguistic studies by N. Chomsky, A.V. Hladkyi, D.V. Lande, A.E. Pentus and M.R. V. Ingve, Yu.A. Schrader, L. Tesniere, P.M. Postal, D.G. Hays, L.W. Tosh, Y. Bar-Hillel and other researchers make

it possible to understand the basic principles of syntactic analysis of text data arrays depending on the specifics of a specific language, including for the Ukrainian language based on relevant research by Ukrainian specialists. During SYA, each sentence is formalized and transformed into a data structure in a tree form of syntax and token dependencies (Fig. 6).

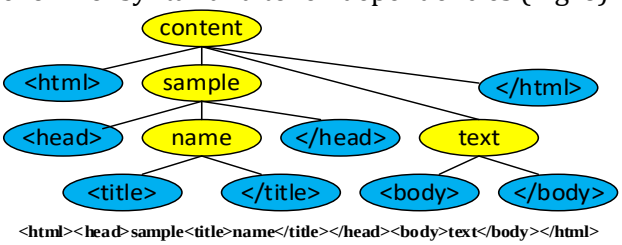

Figure 6: An example of parsing an expression into a token dependency tree

The syntax of sentences is a set of rules of a specific language for forming the dependence of linguistic units to determine the semantic roles and correspondence between entities/objects/phenomena/events/actions in the context of the text based on the operations of the logic of statements. Then syntactic parsing is the process of parsing the input information marked at the previous levels to identify the grammatical structure according to the formal grammar of the corresponding language with the subsequent construction of a dependency tree. This is a rather complicated process for the synthetic type of inflected languages like Ukrainian, where the lexical meaning is synthesized with the grammatical meaning within the lexeme based on supplementivism (generation of grammatical forms of words from different bases, for example, *сказати* [skazaty] (to say) – *говорити* [hovoryty] (to speak), *взяти* [vzyaty] (to take) – *брати* [vzyaty] (to take), etc.), alternation of sounds, formative affixes (a part of a word that changes the meaning of the base, for example, *заїхати* [zayikhaty] (drive-in), *пароплав* [paroplav] (steamship), *лісостеп* [lisostep] (forest-steppe), *заморський* [zamors'kyy] (overseas), etc.) and inflexions. The inflexion of verbs and cases of noun groups determine ways of changing tokens to describe the relationship of tokens to each other within the construction of a sentence to convey meaning. Therefore, sentences in synthetic languages such as Ukrainian are based on word change to describe the structure of token relationships, and do not depend on the location of tokens in the sentence, except only a few moments (for example, a particle does not always precede a negative token, any preposition always precedes a token of the noun group type or noun and do not occur before the verb).

Analytical languages, such as English and German, are relatively limited in morphology, in particular cases, inflexions, and conjugation, but are developed in the use of a variety of prepositions and articles (without them, sentences in such languages fall apart in context). That is, synthetic languages convey context through lexeme relations based on word change within a sentence, and analytic languages use prepositions to form these relations. Sentences of inflectional languages are difficult to programmatically analyze. Natural language often contains ambiguities (tokens that convey many variants of meaning, but only one for a specific context). The correct choice of meaning often depends on the content of the sentence/text, and predicting all possible options is inappropriate. It is difficult to implement structured rules for the implementation of informal events, but due to the identification of the context and the construction of a dependency tree, the list of options can be significantly narrowed down to a minimum. The result of the syntactic analysis is the syntactic structure of the sentence in the form of a parsing/syntax tree and token dependencies. A syntax tree is a graphical representation of the stages of component/dependency parsing of the input text according to the context.

3.5. Semantic and ontological analysis of the Ukrainian text

Semantic analysis forms the structure of the content of the text based on clarifying the relationship of lexemes on SYA and determining the semantic roles of the subjects/objects of the text. Also, SEA (semantic analysis) filters incorrect token values and semantic incoherence. For

the semantic analysis of the text, both Minsky frame models and semantic networks are used, as well as based on ontology, referential and structural analysis to form a set of interphrase units. The result of SEA is an understanding of the content and context of the input text. N.M. Leontieva distinguishes the following types of semantic structures: linguistic structures of text sentences (local understanding), semantic networks of the entire text (global fuzzy understanding), informational structures of the entire test (global generalized understanding), and structures of databases and knowledge (selective special understanding). Case grammars and semantics (the ability of a lexeme to connect other lexemes in the appropriate syntactic way) were proposed for SEA of sentences in [6], thanks to which the semantics of phrases are described through the relationship of the main word with its semantic cases. For example, the main word send is described by the semantic cases of the sender, the addressee and the object of forwarding. To analyze the semantics of the text, predicates (production rules) and semantic networks (labelled graphs, where nodes are definitions, and oriented edges are relations between them) are used. Within the framework of the generative approach, the valences of words (primarily verbs) are described in the form of special frames (subcategorization frames), and within the framework of the approach based on dependency trees – management models. The theory of discourse and pragmatics (elaboration of individual phrases and texts) is based on Van Dijk's research. Anaphoric references and other discourse phenomena are analyzed for the discursive synthesis of connected texts. In the semantic model of the *content↔text* type, a special converter of the given content (invariants of all synonymous transformations of the text) into text and vice versa is considered. The content of a coherent fragment without dismemberment into phrases/word forms is presented in the form of a special semantic structure consisting of two components: a semantic graph and information about the communicative organization of meaning through sememe (semantic unit) and seme (meaningful unit; atom of sememe). A lexeme consists of a seme (a lexical-semantic variant - different meanings) and a sem (a formal variant), whose semantic meanings change due to expansion (increase in meaning), narrowing (concretization of meaning) and displacement (redefinition of meaning). The term seven was introduced by Eric Buyssens and studied by Bernard Pottier.

Semes are the foundation for building SA ontologies. Similar to the sememe, according to Leonard Bloomfield and Kenneth Pike, is an episeme as a unit of tagmeme meaning (the smallest functional element in the grammatical structure of the language). This is an analogue of a morpheme, defined as the smallest meaningful unit of a lexical form. The process of identifying semes in the meaning of words is a component analysis (splitting the meaning of a lexeme into components such as semes, markers or semantic multipliers) based on the construction of binary oppositions. Classical oppositions are equivalent (classification by qualitative difference), gradational (classification by different gradation of the feature) and private (dichotomous classification of elements by the presence/absence of a differential feature). Marcus Solomon also proposed disjunctive (lack of similarity) and null (identical) oppositions. Nikolai Trubetzkoy, in contrast to classical oppositions between members, proposed the system: multidimensional (the relation of which covers other oppositions), isolated (the absence of another opposition with a similar relation) and proportional (the identity of the relations between the members of two oppositions, i.e. the presence correlations to identify a certain speech pattern).

Bernard Pottier and Algirdas Julien Greimas laid the foundations of the component analysis of structural semantics (structuralist semantics) based on the method of Nikolai Trubetzkoy – oppositional phonological analysis through the comparison of phonemes with the identification of their features. The component analysis is directly related to the theory of the semantic field based on the research of Roman Jakobson, Louis Trolle Hjelmslev and other linguists with an emphasis on transferring the principles of phonology by Nikolai Trubetzkoy to grammar (description of case meanings) and semantics (description of the semantic field). In comparison with phonology, here the number of differential features increases significantly and is heterogeneous in terms of the degree of generalization (the more generalized semantic features, the smaller their number, and vice versa, the more specific semantic features, the greater their number. The subject-logical analysis is redundant and ineffective.

Syntagmatic (distributive) and paradigmatic analysis are currently more reliable based on the study of the semantic field (a set of words and their meanings with paradigmatic relations based on a semantic integral feature and distinguished by at least one differential feature). Words and signs of the semantic field form hierarchically organized structures as ontologies, for example, based on the integral sign of kinship and such differential signs as degree, imitation, generation, etc. A semantic feature in different semantic fields has a different hierarchical status (from an element of a category feature to a differential feature). Structural semantics was initiated by the studies of Ferdinand de Saussure and continued in the theory of the lexical field, relational semantics by John Lyons, component analysis (Eugenio Coseriu, Bernard Pottier and Algirdas Greimas), generative linguistics by Noam Chomsky. Ferdinand de Saussure claims that language is a system of interconnected units and structures and that each unit of language is related to others within the same system. Famous developers of structural semantics were Horst Geckeler, Kurt Baldinger, Klaus Heger, Émile Benveniste, Louis Hjelmslev. Carl Hempel, Willard van Orman Quine, and Karl Popper were active in researching the relationships between the meanings of terms in a sentence and how meaning can be composed of smaller elements.

Structuralism is a very effective aspect of semantics, explaining consistency in the meaning of certain words and expressions. The concept of meaningful relations as a means of semantic interpretation is an offshoot of this theory. Structuralism has changed semantics to its present state, and it also helps in understanding other aspects of linguistics. Consequent spheres of structuralism in linguistics are meaningful relations (lexical and phrasal). The content of a coherent fragment of the text without dissection into phrases and word forms is presented in the form of a special semantic representation (ontology), which consists of two components: a semantic graph and values about the communicative organization of the content. Features of the theory: focus on the synthesis of texts (the ability to generate content-correct texts); multi-level and modularity, in particular, the available levels as deep (semantic) and surface (pure) syntax; integrality; saving each level of information by the corresponding module with the transition to the next level; special means of describing the syntax (rules of connecting units) at each of the levels based on a set of lexical functions through the formulated rules of syntactic paraphrasing; emphasis on the dictionary, not on the grammar (the preservation of information of different levels of the language, in particular, for syntactic analysis, word management models describing their syntactic and semantic valences are used). The semantic model of the *content*↔*text* type is based on an explanatory-combinatorial dictionary, in the dictionary article of which, in addition to morphological, syntactic and semantic information (syntactic and semantic valences), information about the lexical connectivity of this word is provided.

Dictionaries of synonyms, paronyms (outwardly similar words that differ in meaning), bases of typical word combinations, thesauruses (semantic dictionary with meaningful relations of words as synonyms, genus-species, part-whole, associations, etc.) and ontologies (sets of semantically dependent concepts following a set of production rules). Ontologies are developed based on the lexicon (linguistic, for example, WordNet, EuroWordNet) and grammar (set of rules for expressing general syntactic properties of words and groups of words) of natural language, the type of which depends on the syntax model. Due to the presence of ambiguity at deeper levels, natural language text analysis within one of the NLP stages often cannot be unambiguously and correctly performed semantic analysis. Then, in such situations, the best option is to generate a set of the most probable analysis results based on intelligent data processing methods. However, the use of such an approach leads to significant computing loads, and optimization due to discarding part of the results leads to the possibility of losing relevant information and the lack of admissible interpretations at the next stages of semantic analysis. Another approach is to use specified structures, where information is presented in an incomplete form at each NLP stage to avoid choosing between different options. The use of feature structures allows you to present information in a specific form in the presence of features without values for the corresponding variables. But ambiguity is in the form of whether one structure of signs is embedded in another or vice versa. The solution is the application of minimal recursion semantics (Minimal recursion semantics, MRS) as a transformation of a nested structure of features (or predicates) into a flat one - a set of structures united by conjunctions. Minimal recursion semantics is the basis for

computer semantics and is implemented in feature structure formalisms, such as Head-driven phrase structure grammar (HPSG), and lexical functional grammar (LFG). Developed by Ivan Sag, Carl Pollard, Dan Flickinger, Ann Copestake for computational language parsing and natural language generation. Allows formulation of grammatical constraints for lexical and phrasal semantics, including principles of semantic composition, for example, in machine translation. The RMRS (Robust Minimal Recursion Semantic) formalism is a development of MRS, the difference of which is in the breakdown of the structure from several signs (multi-argument predicates) to single signs (binary predicates). If feature structures are represented as directed graphs through sets of edges, for each of which the initial and final vertices are specified, and such pointers are represented as constants/variables. In the representation, additional restrictions can be set, for example, requirements for the difference in the value of some variables.

3.6. Setting the problem of processing the Ukrainian-language text

3.6.1. Ukrainian-language text analysis problem

Each natural language has a special structure and a unique collection of linguistic units for generating meaningful content (Table 3), which in turn significantly complicates/impossibility the process of adapting NLP algorithms of one language to another to solve a specific NLP problem [32-39]. Developing new NLP methods for a specific language when solving a specific NLP problem requires a lot of resources, effort and time, which leads to the non-competitiveness of the corresponding projects [40-45]. But the main difficulty usually lies in the lack of native speakers in such projects as specialists at the intersection of the IT, AI and CL fields [46-55], because a non-native speaker is limited in his thinking by the structure and features of his natural language [56-77]. For example, in the Ukrainian language there are linguistic phrases that are incomprehensible to most foreigners [6], in particular, *на столі стакан стоїть* [na stoli stakan stoyit'] (there is [by content Ukrainian - *is standing*] a glass on the table), or *на столі виделка лежить* [na stoli vydelka lezhyt'] (the fork is [by content Ukrainian - *is lying*] on the table). But if you stick the same fork into the table, it will stand. As if it's simple - horizontal things lie, and vertical things stand. But this is not so - *пательня та тарілка стоять на столі* [patel'nya ta tarilka stoyat' na stoli] (the pan and the plate are (by content Ukrainian - are standing) on the table), but *тарілка лежить в пательні* [tarilka lezhyt' v patel'ni] (the plate is (by content Ukrainian – is lying) in the pan). *Кіт на столі* може *лежати*, *сидіти* або *стояти* [Kit na stoli mozhe lezhaty, sydity abo stoyaty] (The cat on the table can lie, sit or stand by content Ukrainian), but *жива пташка* – лише *сидіти* [zhyva ptashka – lyshe sydity] (the live bird can only sit by content Ukrainian), but *іграшка пташки – лежати* [ihrashka ptashky – lezhaty] (the toy bird – lie by content Ukrainian), *опудало пташки – стояти* [opudalo ptashky – stoyaty] (the stuffed bird – stand by content Ukrainian). *Чобіт – сидить на нозі* [Chobit – sydyt' na nozi] (Boot - sitting on the leg by content Ukrainian), but *стоїть/лежить біля столу* [stoyit'/lezhyt' bilya stolu] (standing/lying next to the table by content Ukrainian). *Сукня/спідниця гарно сидить на дівчині* [Suknya/spidnytsya harno sydyt' na divchyni] (The dress/skirt fits well on the girl by content Ukrainian). For a non-native speaker, there is no logic here at all. In English, everything is simple - the object/subject is *на/біля/під* [na/bilya/pid] (on/near/under), etc. the object/subject. This is one of the main reasons why the Ukrainian language is quite difficult and incomprehensible for non-native speakers.

Table 3
Typical structure of natural language

Linguistic analysis of text content	NLP processes				
	Structural			Analytical	
	Graphemic	Morphological	Lexical	Syntactic	Semantic
Writing (spelling) Speaking (phonetics)	letter sound	part	sentence	sentence	corpus

For the full use of language-encoded data, it is necessary and sufficient to consider any natural language not as understandable and natural, but as unlimited and ambiguous. The linguistic unit of textual content analysis is a *lexeme* (a sequence of coded characters/bytes). *Words* are the broader meaning of lexemes, in particular, a meaningful sequence of symbols in the form of a verbal image/sound construction. Lexemes are not words. Words do not have a universally fixed meaning independent of cultural/language contexts. English and Germans use adaptive word forms with suffixes and prefixes that change tense, gender, etc. [6]. The Chinese, on the other hand, recognize a set of pictographic images, where the meaning is identified through the order of the sequence. Unlike the English, Ukrainians use the change of endings, sounds in roots, form-forming affixes, and suppletivism (Table 4) to connect independent linguistic units [6].

Table 4
Comparative features of Ukrainian/English linguistic features

Part of speech	Ukrainian language	English language
Noun	There is a grammatical gender.	There is no grammatical gender.
	The division into male, female and middle genders	Division into people on the one hand by gender, and into phenomena, other living beings and objects.
	Seven cases	Two cases - general and possessive
	Relations through cases	Relations through prepositions.
Article	–	There are two forms - indefinite and definite
Infinitive	A simple form	In addition to simple, as in Ukrainian, there are 5 more complex ones
Pronoun	Division into 9 digits	Division into 7 digits
	2 forms of the 2nd person: singular as *ти*, in the plural as *ви*.	The personal pronoun you is missing (its function is performed by the pronoun – you)
	Personal: it replaces all feminine nouns; he is male; it is neuter.	Personal: he – living beings of the masculine article; she – living creatures of the female gender; it - animals or inanimate objects.
Verb	Expression of completeness or incompleteness of actions, which do not always depend on those factors with the English verb.	Or occurred before some other action in the past, etc. whether it occurs at the moment of speech, during the time that is still ongoing, or the action occurs in general, always, constantly, repeatedly
	–	Often used with adverbs without lexical meaning
Impersonal verbs	Available, for example, in the evening.	–
Gerund	–	6 forms
Adjective	They agree with the noun and change according to cases, numbers, genders	They do not change and do not agree on cases, numbers, genders
Participle	Only one form	2 forms of the participle: present and past, have some adverbial properties
Adverb	2 forms	It is not available in its "pure" form
Numeral	Agree on cases, genders	They do not agree on cases, genders
Service words	Adverbs, prepositions, conjunctions and exclamations have no significant differences	
Sentence	The order of words in the sentence is free.	Order: subject - predicate - other members of the sentence.

English, on the other hand, use for this the order of linguistic units in combination with official words (articles, particles, prepositions). Compared to analytical languages, synthetic languages are more archaic and have a more developed morphology and, therefore more complex semantics. Redundancy, ambiguity and visual associations define natural languages as dynamic, capable of rapid/operational development and conveying the experience of the present. For example, the modern development of emoticons (emograms) makes it possible to translate

children's/adolescent fiction school literature concisely. When a formal grammar is developed and the grammatical/syntactic rules for the use of emoticons are defined, this language will change even more, adapting to the needs of today and the development of IT (changing or increasing the content of specific emoticons, the appearance of new ones and the transformation of others into archaisms, etc.). During the writing of this dissertation, the Ukrainian language underwent some transformations. In particular, on May 22, 2019, the Cabinet of Ministers adopted a new version of the Ukrainian orthography (the change process has been ongoing since June 2015, public discussion since 2018). The transitional stage will last 5 years - until 2024. However, each new change affects the rules for processing Ukrainian-language textual content of CLS. This is the addition of new not only symbols/words and structures to adapt the language to the present, but also definitions/contexts/methods of use. Identifying the meaning of words requires more calculation and analysis than a simple CLS dictionary search.

3.6.2. The main problems in the Ukrainian-language text processing

Ukrainian-language textual content, regardless of style, usually contains a significant amount of unstructured abstract information. It is a meaningful chain of linguistic units with a predetermined structure, integrity and coherence. Correct, operative and full-fledged content analysis of the relevant Ukrainian-language text allows for solving many modern NLP tasks. Parsing Ukrainian textual content into lexemes based on finite automata and Chomsky's grammar is a classic approach. But it does not solve the main problems of processing Ukrainian-language textual content, in particular:

1) Correct matching of all word forms in a sentence, especially when using/generating verbs in complex sentences [6]. The average word length for the English language is about 4.3-4.4 letters (3.5 phonemes) and for the Ukrainian language - about 4.9-5.2 phonemes/letter (depending on the genre). However, the average length of an English-language sentence is longer than a Ukrainian-language one due to the presence of articles and the operative word *of*. If articles are not taken into account (consider that articles are an integral part of most noun groups) and *of* (this is only a connection within a noun group), then the average number of words in an English sentence will decrease significantly. IIS occurs by keywords without taking into account articles and *of*, although the latter significantly affects the result of semantic analysis. There are no such simple hints in Ukrainian language tests - there you have to take into account inflexions to the bases of words and the location of these words with each other, taking into account punctuation marks and other official words. According to, the average number of signs in a Ukrainian sentence is 72.4, in an English sentence - 83.5; 67.7 letters in a Ukrainian-language sentence, and 79.2 in an English-language sentence; words 13.1 and 18.2, respectively. If articles and *of* are not taken into account, the average length of words in an English-language sentence is 10-11. But if we consider only the spoken text (dialogue), then the gap between the corresponding values grows between these indicators. This simplifies the processing of English-language texts and almost does not simplify the processing of Ukrainian-language text dialogues.

2) Presence and coherence of complex sentences. According to [6], the use of complex sentences in English-language texts is approximately 11%, respectively, in Ukrainian-language texts - 15%. Accordingly, the use of complex sentences is 89% and 85% among 300 samples for each respective language among all complex sentences. However, the author did not take into account, even in the examples given by him, that Ukrainian-language complex sentences often contain more than two sentences, compared to the English-language versions. In addition, Ukrainian speakers have more sentences with a combination of subordinate and subordinate clauses not only by number but also by variations of 12% and 9%, respectively, for these two languages. So there should be more processing rules, which affect the complexity of the analysis. In general, the proportion of the use of complex sentences varies between 10-40%, depending on the author's style and the genre of textual content.

3) Analysis of nominative sentences (essential, evaluative and indicative) and their features (without the use of verb groups in sentences/statements, the main member is a noun group) in dialogue texts. According to [6], in the analyzed texts of each of the two languages, for English-

language texts, substantive sentences are 25%, and for Ukrainian-language texts - 41%. Accordingly, for the evaluation sentences, 75% and 55%, respectively. Indicatives are mostly characteristic of Ukrainian-language texts - 4%. The only problem is that the author conducted the analysis only among nominative sentences of the respective languages, without taking into account the frequent use of these types of sentences among others in general text arrays of data. I usually use such sentences in poetry. And in contrast to the English language, nominative sentences are often used in Ukrainian-language texts, especially in dialogues – *Зараз тепло. Сьогодні холодно. А ти весела людина. Посміхайся! Знову в школу. Вже кінець літа. Десь попереду.* [Zaraz teplo. S'ohodni kholodno. A ty vesela lyudyna. Posmikhaysya! Znovu v shkolu. Vzhe kinets' lita. Des' poperedu.] (Now it's warm. Today is cold. And you are a cheerful person. Smile! Back to school. It's already the end of summer. Somewhere ahead.).

4) The lack of a clear structure of the sentence, unlike English, which has a fixed (direct) order of linguistic units in the sentence (*subject* → *predicate* → *object* as the core of the sentence – Fig. 7) [6]. For example, for one English sentence *Teenagers like music*, there are 6 variants in Ukrainian, in particular:

Підліткам подобається музика. [*Pidlitkam podobayet'sya muzyka.*] → *Музика подобається підліткам.* [*Muzyka podobayet'sya pidlitkam.*] → *Підліткам музика подобається.* [*Pidlitkam muzyka podobayet'sya.*] → *Музика підліткам подобається.* [*Muzyka pidlitkam podobayet'sya.*] → *Подобається підліткам музика.* [*Podobayet'sya pidlitkam muzyka.*] → *Подобається музика підліткам.* [*Podobayet'sya muzyka pidlitkam.*].

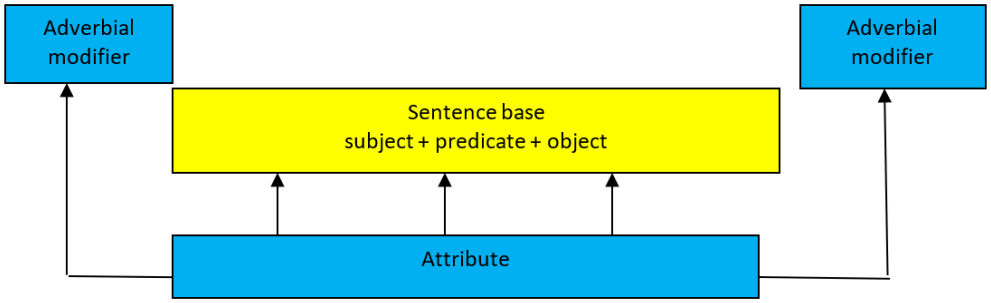

Figure 7: Rules for constructing an English sentence

Interchanging the words *teenagers* and *music* in the English language leads to understanding the sentence so that *music likes teenagers* (absence of meaning) [6]. But for the sentence, *teenagers like singer*, rearranging the words as *singer like teenagers* will lead to the formation of a new meaning of the text. In Ukrainian, thanks to the correspondence of inflexions, it is permissible to rearrange words to avoid the formation of new meanings/nonsense. However, this makes it much more difficult to implement the POST process to identify the meaning.

5) The difficulty of identifying a noun group that can perform various functions, in particular: the subject of a sentence, an adjunct, circumstances simultaneously with a preposition, the meaning or noun part of a complex predicate, including with an adjective, a pronoun, a proper name or an abbreviation without a corresponding display in the dictionary. The nominal group is determined by the set of the relevant meaningful vocabulary of the speaker, taking into account his subjectivism, in particular, words or phrases belong to one of the categories:

1. Direct unambiguous definitions, regardless of the context of the text.
2. The content depends on the specific context of the text (multi-meaning) or has a meaning different from their word-forming components.
3. Newly formed, borrowed or highly specialized words that are not in publicly available dictionaries and whose meaning is ambiguous.

6) The difficulty of identifying an adjective (a quality, feature or property of a noun) in a noun group (not only by its ending in the Ukrainian language and its location - usually before a noun or another adjective). Qualitative, relative and possessive adjectives are distinguished, as well as by the simple or complex form of the highest/highest degree of comparison.

7) Complex identifications of the verb group depending on the possible components of this group (verbs, noun groups as circumstances, participles, adverbs, etc.) and word change depending on the time (future, past, long past and present), the form of the verb (infinitive, personal, participle, impersonal, reflexive and adverb), type (imperfect, perfect), transitivity/intransitiveness (presence/absence of direct object), conjugation (I/II), modes (active, conditional and imperative) and state (active or passive). Prefixes, suffixes, alternation of sounds/letters, stress and various bases are used for the corresponding formations of verbs.

8) The difficulty lies in the presence of a large range of synonyms for describing phenomena/events, etc., morphological analysis of the Ukrainian text and the content of a specific NLP task. For example, the rubrication of a Ukrainian-language text or the determination of the authorship of an article is complicated by the process of identifying a set of keywords (the presence of synonyms and the complexity of MA) and persistent phrases (due to the loose order of words, the presence of several variants of words with the same meaning and the variety of persistent phrases). The task of referencing a Ukrainian-language text is complicated by all NLP stages of analyzes from grapheme to pragmatic.

9) The construction of e-dictionaries, thesauruses and grammars is a voluminous and complex process than the development of a linguistic model and the corresponding NLP module. Automation of the construction of linguistic resources or virtual libraries is one of the promising areas of computer linguistics research, but it is directly related to the correctly constructed previously described levels of analysis of natural language as morphological and syntactic. E-dictionaries are usually generated by converting ordinary text dictionaries, but for their correct construction, collections and corpora of texts of the corresponding SA are additionally used, collected according to a certain principle of categorization (by genre, authorship, etc.) and appropriately marked/marked (annotated) - accentually, morphologically, syntactically, etc.

Typically, labelled corpora are created by linguists and applied to various linguistic research and CLS tuning based on mathematical machine learning techniques, such as IIS, machine translation, error correction, anaphoric reference analysis, speech recognition/synthesis, lexical ambiguity resolution, etc. Text corpora are always limited in their presentation of speech phenomena, and this is a significant drawback. Therefore, the best option is to use text streams of a specific language on the Internet as a linguistic resource as a base of text corpora from reliable sources. But this requires the development of special IT and corresponding CLS.

10) The lack of general rules and standards of typical CLS structures and development stages, in turn, brings disadvantages to the construction of such systems. Therefore, it is necessary to develop NLP models/NLP methods and the general structure of a typical CLS. It will also facilitate the work of defining functional requirements, and typical architecture and recommending the development of appropriate CLS based on modern ML methods.

4. Experiments, results and discussion

4.1. Project of a typical computer linguistic system

4.1.1. Main characteristics of the computer linguistic system

The goal of a typical CLS is the implementation of methods and IT approbation of the intelligent analysis of a text stream for the solution of a specific NLP problem [78-81]. The design of the general structural scheme of CLS causes the specification/typification of IT intelligent analysis of the text stream in CLS through the main stages of integration/management/support for the optimality/quality/efficiency of the solution of a specific NLP problem for a specialized SA [81]. The use of such CLS reduces the total time of processing/analysis of integrated text streams of information resources [81-95], statistics/dynamics of the life cycle of txt content (TCLC) [96-99], activity of regular/potential users, and functioning of CLS (Fig. 8) [100-106] and growing volumes of CLS functionality and permanent/potential target audience.

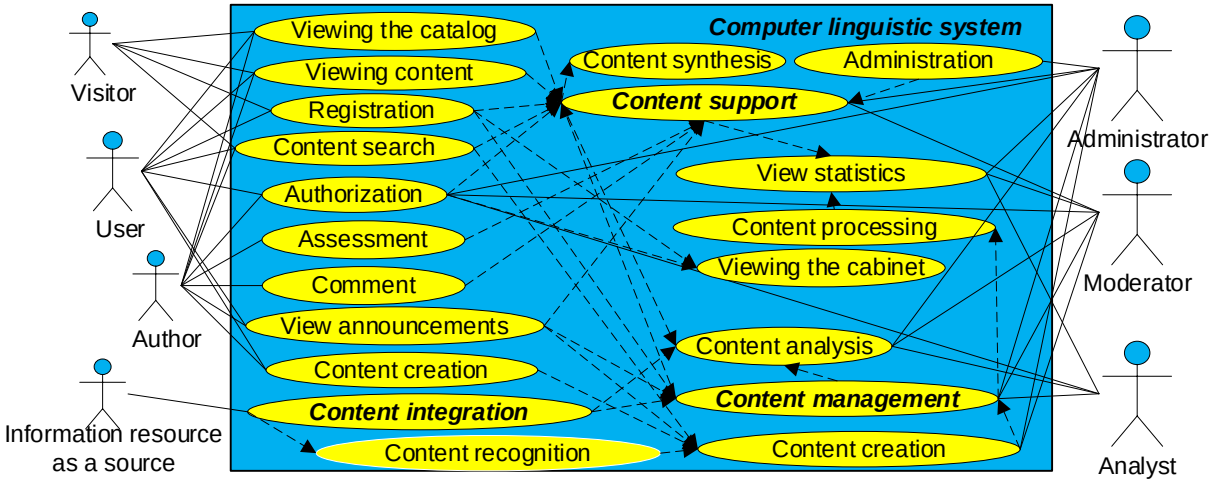

Figure 8: Use case diagram of a typical CLS project

The process of intelligent analysis of text flow in CLS consists of [78-106]:

1. content integration based on text recognition and analysis (collection/creation/formation of text content from various sources, filtering/saving, formatting, structuring, sorting/annotation, clustering and classification, formation/generation relevant filtering/IIS/integration/recognition/analysis rules);
2. content management based on analysis and text processing (filling DB/SD/KB; caching of popular information blocks/Webpage/IIS results; collection/analysis of statistical data on the dynamics of CLS functioning, conversion of user visits and history of transitions according to user requests; generation of Webpage/forms according to user requests; support of interactive interaction with the Website reviews, comments, votes of the permanent audience);
3. content support based on analysis and synthesis of information (generation and updating of information slices/portraits relative to the time intervals of the content flow, potential/permanent personalized users and target audience; identification and updating stories/scenarios of classified content relative to time slots; content ranking/analysts/authors; conversion/actions classification of regular users/visitors, respectively).

Conversion factors K_{wcv} (achievement of the goal by users according to all actions of the relevant content) for CLS are calculated as follows [81-82]:

$$\mathrm{K_{wcv}} = \frac{\mathrm{N_{wcv}}}{\mathrm{N_{vrb}}};\ K_{wcv} = \frac{N_{wcv}}{N_{vtb}};\ K_{wcv} = \frac{N_{wcv}}{N_{wvr}};\ K_{wcv} = \frac{N_{wcv}}{N_{wvt}}; \quad (3)$$

where N_{wcv} is the CLS conversion number, N_{vrb} is the total number of Website users when the relevant conversion is achieved (successful conversion), N_{vtb} is the total number of Website visits when the relevant conversion is achieved, N_{wvr} is the total number of Website users, N_{wvt} is the total number of Website visits.

CLS is used to solve a specific NLP problem according to the relevant requirements/needs of the end user or potential audience, for example, to implement e-business information services based on IT, machine learning and the main stages of NLP. CLS is used for the provision of information services in the relevant spheres of activity of the permanent user and the target audience, for example, for the sale of content through an Internet store, Internet publication, Internet magazine, Internet publishing house, Internet newspaper, Internet marketing, provision of consulting or SEO services, etc. CLS is used as an additional subsystem of the e-commerce system to promote information services/goods, for example, through news agencies, educational institutions, magazines, software development companies, newspapers, publishing houses, etc. The need to use CLS for solving various NLP tasks is associated with the accelerated operational pace of increasing the volumes/scales of the text flow of content in the Internet/e-business and the growth/spread of access to various sources of information, the increase in the set of CLS functionalities and the automation of development due to the variety of NLP tasks, the increase

in demand/needs for actual/relevant/operational information, the development/implementation of IT/software for the processing of texts of the corresponding natural language and the increase in the number of SA applications of NLP technologies to achieve the set goal of the end user or the target audience of computer linguistics systems.

4.1.2. Justification of the implementation of the project of a typical CLS

The lack of standardized, well-known and non-commercialized IT development of a typical CLS and basic modules for the intellectual analysis of text content flows leads to an increase in the number of problems of designing the general structure of the IS solution of a specific NLP problem depending on the natural language itself. Due to the lack of generally accepted standards and detailed typification of CLS and NLP tasks, the process of developing specialized IT/IS/software for the intellectual analysis of textual content streams is problematic. It follows from this that the standardization of the main processes/modules of CLS as support/integration/management of the textual content of a specific language is problematic.

According to the applications on the Website S_{wtm} CLS, there is a module for solving a specific NLP problem M_{dis}, a content support module M_{dmr}, a content integration module M_{dcp} to support the writing of high-quality/effective up-to-date unique content by Website content copywriters, journalists, authors, etc. and a content management module M_{dvm}. For each, their key performance indicator KRI (Key Performance Indicators) is calculated [81-82]:

$$S_{wtm} =< M_{dis}, M_{dmr}, M_{dcp}, M_{dvm} >. \quad (3)$$

Common and popular modern CLSs operate based on methods unknown to most NLP practitioners because these CLSs are closed commercial projects. When new CLSs are developed, NLP specialists create new or modified methods/tools/modules for intellectual analysis of text streams of content and TCLC support. There is quite a lot of material in the public domain about IT based on computational linguistics. But in most cases, they carry a purely theoretical load and almost do not reflect practical recommendations for training specialists in the development of a specific language. Most of these materials are devoted to studying the English language. And almost absent from the Ukrainian language.

There are no widely available publications on the quality/effectiveness of the influence of the presence of implemented TCLC stages on the dynamics of CLS work for the intellectual analysis of text streams of content in the Ukrainian language. Studies of the dynamics of CLS work are practically absent due to the impossibility of organizing the access of a wide range of researchers to the administrative panels of subsystems of modern popular CLS due to their commercialization. The relevance of the implementation of the CLS project lies in the development of the basic structure, unified methods/modules/IT/software for building the CLS and the main TCLC stages. The introduction of the main modules of content integration/management/support in CLS causes a reduction in the stages/time of generating results according to the requests of regular users and, accordingly, of intellectual analysis of text streams of content in the Ukrainian language. At the same time, this encourages the growth of the potential/permanent target audience of CLS users, which allows the accumulation of statistical data on the functioning of CLS for further machine learning based on the analysis of collected big data. This leads to active and operational growth/adaptation of the functionality of the respective CLS. The development of general basic recommendations for the design and development of the CLS architecture based on the main TCLC stages and modules of intellectual analysis of textual content flows will make it possible to effectively/quality/timely/expeditiously support the life cycle of the construction of the corresponding CLS at several levels. In particular, at the developer level, this leads to a reduction in the amount of time/resources for implementation and an increase in the quality/efficiency of CLS functioning, as well as unification/standardization of intellectual content analysis processes (Fig. 9-Fig. 10). At the level of the owner - increase in profitability and interest of the permanent audience. At the user level – increasing the selection of CLS functionality, support/simplification of the interface, and performance/comprehensibility).

Task name	Period	Start	End	Predecessors
CLS development	**37 днів**	**12.03.24**	**01.05.24**	
Collection/clarification of data from the software of a specific NLP task	2 днів	12.03.24	13.03.24	
Formation of a set of software specifications for the corresponding NLP task	1 день	13.03.24	13.03.24	1
Terms of reference of the CLS project of a specific NLP problem	3 днів	14.03.24	18.03.24	2;1
Clarification of TK through interaction with a potential audience	2 днів	16.03.24	19.03.24	3;1
Identification and analysis of a set of functional requirements for CLS	3 днів	15.03.24	19.03.24	1;2;3;4
Analysis and refinement of the information resource architecture of the CLS	1 день	20.03.24	20.03.24	5;2
Identification and analysis of a set of non-functional requirements	2 днів	21.03.24	22.03.24	6;1
Development of a template of the information resource of CLS	2 днів	22.03.24	25.03.24	4;7
Development of a set of typical templates of textual content	2 днів	23.03.24	26.03.24	8;4
Creation and testing of the CLS information resource	4 днів	26.03.24	29.03.24	9;3;5
Analysis and refinement of the CLS architecture of the corresponding NLP problem	6 днів	22.03.24	29.03.24	5;3
Analysis of the architecture of the KLCS content management subsystem	3 днів	28.03.24	01.04.24	11;5
Creation of text content management subsystem	6 днів	30.03.24	08.04.24	12;10
Development of data storage of textual content of CLS	2 днів	29.03.24	01.04.24	11
Creating a knowledge base for managing text content	2 днів	30.03.24	02.04.24	14
Filling the data storage of the textual content of the CLS	5 днів	02.04.24	08.04.24	15
Development of CLS for solving a specific NLP problem	7 днів	06.04.24	16.04.24	16;13
Analysis of the architecture of the integration of content from different sources	3 днів	13.04.24	17.04.24	5;10;13;16
Creation of a text content integration subsystem	5 днів	15.04.24	19.04.24	18
Creating a repository of data sources of textual content	3 днів	17.04.24	19.04.24	19
Creating a knowledge base for text content filters	3 днів	18.04.24	22.04.24	20;17
Analysis of the architecture of the CLS content support subsystem	3 днів	18.04.24	22.04.24	5;19
Creation of a text content support subsystem	5 днів	19.04.24	25.04.24	22;21
Testing of the text content management subsystem	6 днів	07.04.24	15.04.24	13
Testing of the CLS solution of a specific NLP problem	3 днів	14.04.24	17.04.24	17;24
Testing of text content integration subsystem	3 днів	17.04.24	19.04.24	25;19
Elimination of shortcomings of the content management subsystem	6 днів	10.04.24	17.04.24	24
Testing of the text content support subsystem	3 днів	21.04.24	24.04.24	23;26;27
Elimination of deficiencies in the functioning of the CLS	4 днів	18.04.24	23.04.24	25;27
Elimination of shortcomings of the content integration subsystem	3 днів	21.04.24	24.04.24	26;29
Elimination of shortcomings in the operation of text content support	4 днів	24.04.24	29.04.24	27;28;29;30
Preparation of technical documentation developed by CLS	5 днів	25.04.24	01.05.24	24;25;26;27;28;29;30;31

Figure 9: An approximate schedule for a typical CLS design and implementation

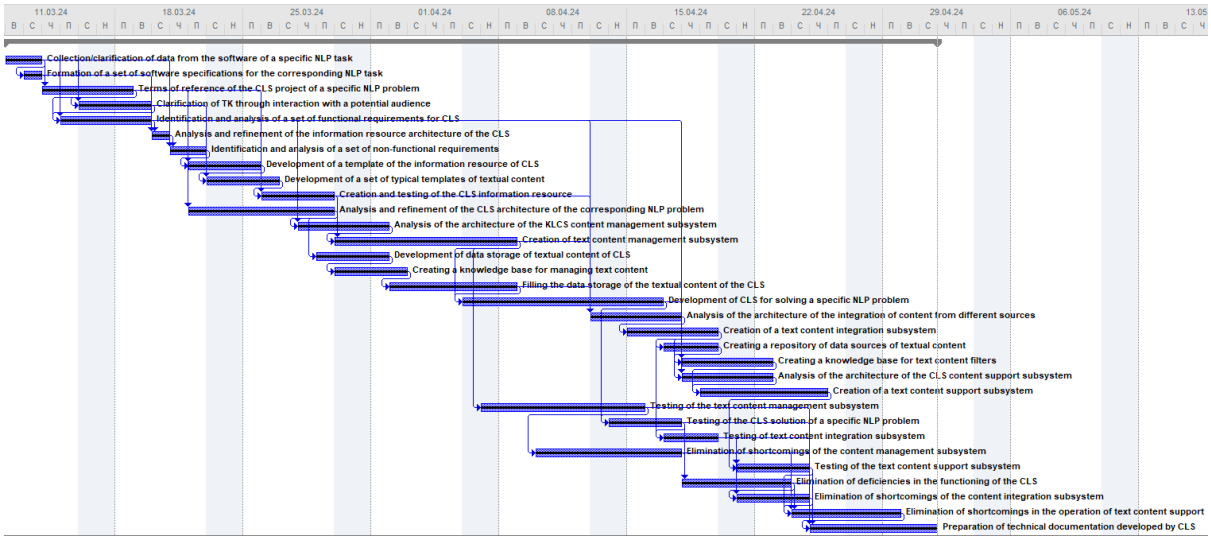

Figure 10: A Gantt chart of the design and implementation of a typical CLS

Fig. 9 shows a general oriented plan for the development of a typical CLS based on the implementation of the main stages of intellectual analysis of text streams of content in the Ukrainian language to simplify the analysis/estimation of financial/time/resource costs. This reduces the time spent on the implementation of the CLS project, reduces the number of NLP specialists and clearly describes the development regulations based on the analysis of the amount of time used for the relevant stages. Fig. 10 shows a Gantt diagram of the design and

implementation of a typical CLS, which allows us to analyze a clear and detailed regulation of the development of a typical CLS as the stages of intellectual analysis of text streams of content in the Ukrainian language and the involvement of relevant NLP specialists at these stages. The results of stage 1 activate stages 2-5 and 7, and stages 4 – 5, 8-9, which allows early redistribution of tasks between the relevant NLP specialists in time and participants between teams, etc. Stage 5 requires output from Stages 1-4, and Stage 5 results activate Stages 12, 18 and 22. The untimely implementation of stage 10 leads to a simultaneous delay in the implementation of stages 13 and 18. Reducing the time for the implementation of stages 11, 13, 16, 17, 19, 24 and 27 will allow early implementation of the CLS project, but will lead to an increase in the occurrence of additional errors that are usually eliminated in stages 24-31.

4.1.3. Expected effects of implementing a typical CLS project

The predicted economic effect of solving a specific NLP problem depends on the reduction of project creation costs and the general architecture of a typical CLS, the use of additional specialists/specialists/experts/resources, and the availability of clear regulations for the implementation of relevant modules for the intellectual analysis of textual content streams in Ukrainian according to the following factors:

1. The presence of a module for solving a specific NLP problem [81-82] based on the linguistic processing of texts in Ukrainian forms a set of unique target audiences for further analysis and fixation of user needs and corresponding adjustment of e-business goals to increase profits (not only financial but information/resource).

In CLS with a module for solving a specific NLP problem, the value of K_{PI} (KPI) is likely to be larger, since this is usually the main goal of the end user for conversions from IISS (intelligent information search system), social networks, other Websites/banners and direct visits to the Website. It is enough to judge by the indicators of reports from Google Analytics, for example, the number of visitors/users of N_{wvr} (but some K_{PI} must be extracted from other modules for clarification) [81-82], as well as:

$$\text{M}_{\text{dis}} =< \text{N}_{\text{wvr}}, \text{S}_{\text{gcc}}, \text{S}_{\text{gco}}, \text{S}_{\text{gcv}}, \text{S}_{\text{gro}}, \text{P}_{\text{wnv}}, \text{I}_{\text{wnv}} >, \quad (4)$$

where S_{gcc} is the average conversion rate according to Google Analytics calculations, S_{gco} is the average value of orders according to Google Analytics calculations, S_{gcv} is the average cost per visit (usefulness of the visit according to the data of e-commerce transactions) or the average usefulness of the purpose of the visit (based on the usefulness of goals) according to Google calculations Analytics, S_{gro} is average P_{ROI} or average return on investment according to Google Analytics and AdWords calculations, P_{wiv} is the percentage of profit from new visitors to Website CLS, I_{wnv} is the index of new buyers/customers at the first visit to Website CLS.

Performance Indicator for Total Gross Profit P_{ROI} [81-82]:

$$\text{P}_{\text{ROI}} = \frac{\text{N}_{\text{Inc}} - \text{N}_{\text{Exp}}}{\text{N}_{\text{Exp}}}, \quad (5)$$

where N_{Exp} is expenses, N_{Inc} is profit. If P_{ROI}<0, then the cost of attracting users of the target audience is greater than the profit. P_{ROI} does not take into account revenue from the provision of services and the number of users or transactions.

Profit rates according to Google Analytics and AdWords calculations [81-82]:

$$\text{P}_{\text{RR}} = \frac{\text{N}_{\text{Inc}} - \text{N}_{\text{Exp}}}{\text{N}_{\text{Inc}}}. \quad (6)$$

The number of visits required to convince to place an order affects the calculation of P_{wiv}. Therefore, the probability of converting a new visitor into a regular user on the first visit [81-82]:

$$\text{I}_{\text{wnv}} = \frac{\text{P}_{\text{wtv}}}{\text{P}_{\text{wnv}}}, \quad (7)$$

where P_{wnv} is the percentage of new Website users, P_{wtv} is the percentage of transactions from new Website users. When I_{nv}=1, a new user and a repeat user are equally likely to become regular users. When I_{nv}<1, a new user is less likely to become a permanent user than a repeat user. Conversely, if I_{nv}>1, the new one will become permanent with a higher probability than the repeated one.

2. The availability of the text content support module reduces costs for moderators/analysts who collect/analyze statistical data on the dynamics of CLS functioning, the activity of the permanent target audience as a reaction to changes in Website/Web page content, and the formation of rules for analyzing user information portraits and thematic content plots.

To identify the best traffic, the obtained profit and P_{ROI}, costs for the company, conversion rate K_{wcv} are studied. Therefore, the K_{PI} of the content support module meaningfully overlaps with the K_{PI} of the solution module of a specific NLP problem based on data from AdWords. Difference in emphasis not only on order conversion rate but also goals for analyzing/developing relationships with users/visitors who will potentially place an order, including:

$$\text{M}_{\text{dmr}} =< \text{I}_{\text{gyk}}, \text{K}_{\text{gvb}}, \text{P}_{\text{wap}}, \text{P}_{\text{wvk}}, \text{S}_{\text{grk}}, \text{I}_{\text{gck}}, \text{P}_{\text{wck}}, \text{P}_{\text{wvk}}, \text{K}_{\text{wcz}}, \text{P}_{\text{wvz}} >, \quad (8)$$

where I_{gyk} is the index of the quality of the advertising campaign according to AdWords; K_{gvb} is brand recognition coefficient; P_{wap} is the percentage of new/repeated customers; P_{wvk} is the percentage of new/repeated users; S_{grk} is average P_{ROI} by type of advertising campaign; I_{gck} is goal conversion index by type of advertising campaign; P_{wck} is conversion percentage of goals by type of advertising campaign; P_{wvk} is the percentage of visits by type of advertising campaign; K_{wcz} is the conversion rate of goals by type of means; P_{wvz} is the percentage of visits by type of means [81-82].

The index of the quality of the advertising campaign I_{gyk} is related to the quality/efficiency and effectiveness of the targeting of the advertising campaign (attracting targeted traffic to the Website CLS [81-82].

$$\text{I}_{\text{gyk}}(\text{w}) = \frac{\text{P}_{\text{wcv}}(\text{w})}{\text{P}_{\text{wvk}}(\text{w})}, \quad (9)$$

where $P_{wvk}(w)$ is a function for determining the percentage of visits from advertising campaign *w*; $P_{wcv}(w)$ is a function for determining the percentage of conversion of goals for visits from campaign *w*; $I_{gyk}(w)$ is a function for determining the quality index of the advertising campaign *w*. If P_{vk}=50% of users switch from AdWords, but only P_{cv}=20% of conversions match this ad source *x*, then this is ineffective targeting. Another advertising campaign y also generates 50% of traffic and corresponds to 80% of conversions, so this is effective targeting. The value of the index I_{gyk}=1.0 means that a customer from this campaign will convert with the same probability as a customer from any other campaign. A value of I_{gyk}<1.0 means, accordingly, that a customer from this campaign is less likely to convert than a customer from any other campaign. If I_{gyk}>1.0 is the customer will convert with a higher probability than a customer from any other campaign.

Coefficient of brand recognition [81-82]:

$$\text{K}_{\text{gvb}} = \frac{\text{N}_{\text{ubq}}+\text{N}_{\text{utv}}}{\text{N}_{\text{uaq}}+\text{N}_{\text{utv}}}, \quad (10)$$

where N_{uaq} is the total number of IIS user requests (keywords); N_{utv} is the number of direct website visits; N_{ubq} is the number of IIS requests with the brand name.

3. The presence of the text content integration module reduces costs for CLS moderators and content authors by automating/implementing some of their work/functions such as content collection from multiple different reliable sources, its recognition, filtering, saving, formatting, analysis, annotation, clustering, classification etc. [81-82].

For CLS developers, the main goal is the maximum involvement of a permanent target audience, the main key indicators of which are the amount of time/frequency/Webpage for familiarizing with the Website content and increasing user interest. For CLS, an important KPI is the volume of visits/orders for a certain period. For the analysis of time indicators according to a repeated visit, the best time intervals t_1<t_2 are chosen for a specific CLS model. Therefore, for the integration module [81-82]:

$$M_{dcp} =< P_{glt}, P_{gst}, P_{g\square t}, K_{gvb}, K_{uzv}, P_{uav}, P_{uzv}, S_{gnc}, P_{wvv}, S_{gpv}, S_{gtp} >, \quad (11)$$

where P_{glt} is the percentage of repeated visits by the user from the previous visit >t_2 days according to Google Analytics; P_{gst} is the percentage of repeat visits of the user from the previous visit within $[t_1; t_2]$ days when t_1<t_2 according to Google Analytics; P_{ght} is the percentage of repeated visits by the user from the previous visit <t_1 days according to Google Analytics; K_{gvb} is

a brand recognition factor; P_{uav} is the percentage of new/repeated visitors according to Google Analytics; P_{uzv} is the percentage of interest of visitors; S_{gnc} is the average number of clicks on advertising for N_{wvr} visits; P_{wvv} is the rejection rate for Webpage P_{vvp}; S_{gpv} is the average number of web page views per visit according to Google Analytics; S_{gtp} is the average length of stay on a webpage through AdWords.

Bounce rate for one webpage based on data from Google Analytics:

$$\mathrm{P_{vvp}} = \frac{\mathrm{N_{vnp}}}{\mathrm{N_{inp}}}, \tag{12}$$

where N_{inp} is the number of direct visits by users of this webpage; N_{vnp} is the number of one-page visits to this webpage via Google Analytics.

The average number of clicks on advertising for N_{vr} visits [81-82]:

$$\mathrm{S_{gnc}} = \frac{\mathrm{N_{wcr}}}{\mathrm{N_{wav}}} \cdot \mathrm{N_{wvr}}, \tag{13}$$

where N_{wvr} is the number of visits for analysis (often N_{vr}=1000 according to CPM - Cost Per Mille); N_{wav} is the total number of visits according to Google Analytics; N_{wcr} is the average number of clicks on advertising according to AdWords.

Indicator of interest of visitors [81-82]:

$$\mathrm{K_{uzv}} = \frac{\mathrm{N_{wad}}}{\mathrm{N_{wav}}}, \tag{14}$$

where N_{wav} is the total number of visits according to Google Analytics; N_{wad} is the total number of actions on the Website according to AdWords.

Percentage of interest of visitors [81-82]:

$$\mathrm{P_{uzv}} = \frac{\mathrm{N_{wzv}}}{\mathrm{N_{wvk}}}, \tag{15}$$

where N_{wvk} is the total number of users according to Google Analytics; N_{wzv} is the total number of interested users according to AdWords.

With effective ideal implementation/use of CLS [81-82]:

$$\mathrm{P_{ght}} >> \mathrm{P_{gst}} >> \mathrm{P_{glt}}. \tag{16}$$

Periodic analysis of such indicators identifies patterns for adjusting content to maintain at least this ratio.

$$\mathrm{P_{ght}} \geq \mathrm{P_{gst}} \geq \mathrm{P_{glt}}. \tag{17}$$

4. The presence of a text content management module reduces costs for moderators/administrators [81-82] who update the Website/Web page and create caching/IIS rules for popular information blocks.

The content management module is responsible for the continuous and efficient functioning of the Website, controlling the load on the servers (the expected number of user requests), and the frequency of use of typical browsers/languages:

$$\mathrm{M_{dvm}} =< \mathrm{K_{wis}}, \mathrm{P_{wep}}, \mathrm{P_{gum}}, \mathrm{P_{gup}}, \mathrm{P_{gur}}, \mathrm{P_{gus}}, \mathrm{P_{gub}}, \mathrm{P_{gul}}, \mathrm{P_{wep}}, \mathrm{K_{wdu}}, \mathrm{S_{wdu}} >, \tag{18}$$

where K_{wis} is the indicator of internal IIS; P_{wep} is the percentage of Web page publications with an error; P_{gum} is the percentage of mobile users according to Google Analytics; P_{gup} is the percentage of users with high-speed Internet connection; P_{gur} is the percentage of users with low/medium/high display resolution; P_{gus} is an extension of users with a specific operating system; P_{gub} is the percentage of users with a specific browser according to Google Analytics; P_{gul} is the percentage of users with English/Ukrainian language support; K_{wdu} is an indicator of the number of users, views and visits to the webpage. The S_{wdu} indicator is the basic content management module according to Google Analytics [81-82]:

$$\mathrm{S_{wdu}} =< \mathrm{N_{svt}}, \mathrm{N_{sut}}, \mathrm{N_{spt}}, \mathrm{N_{spv}} >, \tag{19}$$

where N_{spv} is the average number of Web page views per visit; N_{spt} is the average number of Web page views for a specific time Δt; N_{sut} is the average number of unique users for a specific time Δt; N_{svt} is the average number of visits for a specific time Δt.

Percentage of Webpage generation with an error (must be minimized):

$$\mathrm{P_{wep}} = \frac{\mathrm{N_{wep}}}{\mathrm{N_{wpp}}}, \tag{20}$$

where N_{wpp} is the total number of viewed Web pages; N_{wep} is the total number of issued Web pages with an error [81-82].

Indicator of internal IIS according to Google Analytics [81-82]:

$$\mathrm{K_{wis}} =< \mathrm{N_{nns}}, \mathrm{P_{uts}}, \mathrm{P_{ksp}}, \mathrm{P_{bus}}, \mathrm{P_{cuss}}, \mathrm{P_{pop}}, \mathrm{P_{ucs}}, \mathrm{S_{vrs}}, \mathrm{P_{uos}}, \mathrm{P_{uns}}, \mathrm{P_{unr}}, P_{uur}, S_{nup}, T_{svs}, P_{uis}, P_{nrp}, K_{wps} >, \quad (21)$$

where N_{nns} is the number of zero IIS results on the Website; P_{uts} is the percentage of users who spent $> t$ time on the Website after IIS was implemented; P_{ksp} is percentage of users who viewed $> k$ Webpage after IIS implementation; P_{bus} is the percentage of purchases made among users using IIS on the Website; P_{cus} is the percentage of buyers among users who use IIS on the Website; P_{pop} is the percentage of rejections after visiting one webpage as a result of IIS; P_{ucs} is the percentage of conversion from users using IIS on the Website; P_{unr} is the percentage of users who do not use IIS on the Website; P_{uur} is the percentage of visitors who use IIS on the Website; S_{nup} is the average number of Web pages viewed by visitors after IIS; T_{svs} is the average time spent on the Website for visits after IIS; P_{uns} is the percentage of visitors who spend several IIS on the Website during the visit (taking into account several IIS for the same keyword); P_{uos} is the percentage of visitors who left the Website after viewing IIS results; S_{vrs} is the average number of IIS results viewed after IIS; P_{uis} is the percentage of visits in which IIS is used on the Website; P_{nrp} is the percentage of zero IIS results on the Website, in particular,

$$\mathrm{P_{nrp}} = \frac{\mathrm{N_{nps}}}{\mathrm{N_{vps}}}, \quad (22)$$

where N_{vps} is the total number of viewed IIS Web pages; N_{nps} is the total number of zero IIS Web page results [81-82].

Indicator of IIS K_{wps} usage by Website as a dependency of visits:

$$\mathrm{K_{wps}} = \frac{\mathrm{N_{wsv}}}{\mathrm{N_{wns}}}, \quad (23)$$

where N_{wns} is visits without IIS on the Website; N_{wsv} is a visit from IIS via the Website.

With the modern gradual increase in the number of Website CLS based on RIA technology, the need to calculate the corresponding K_{PI} is increasing [81-82].

5. The presence of subsystems of intellectual analysis of text content streams reduces the time/costs/personnel/resources for the timely and prompt acquisition of relevant, unique, current text content, which leads to an increase in the volume of the CLS target audience, in particular, contributes to the growth of the economic effect of the implementation of CLS by several points.

Analysts are important statistical data not only about the views of the K_{wdu} Webpage, but the dynamics of a set of constant/potential/recurring events/actions of K_{was} from customers/visitors/users based on interaction with the Website, in particular,

$$\mathrm{K_{was}} =< \mathrm{S_{wcc}}, \mathrm{S_{wtv}}, \mathrm{S_{wnv}}, \mathrm{P_{wuv}}, \mathrm{P_{wnv}} >, \quad (24)$$

where S_{wcc} is the average conversion factor; S_{wtv} is the average length of visit; S_{wnv} is the average number of views per visit; P_{wuv} is the percentage of unique customers/visitors/users; P_{wnv} is the percentage of new Website customers.

According to K_{as} event tracking and interaction with the Website K_{du} analyze:

$$\mathrm{K_{usa}} = \alpha(\mathrm{K_{wdu}}, \mathrm{K_{was}}) =< \mathrm{P_{vcu}}, \mathrm{P_{sau}}, \mathrm{P_{siu}}, \mathrm{I_{wdx}} >, \quad (25)$$

where P_{siu} is the percentage of interaction with the Website (for example, commenting, voting, registration, authorization, subscription, etc.); P_{sau} is the percentage of users who activate various events (for example, click on an ad, start a function, pause, etc.); P_{vcu} is the percentage of users interacting with different types of content presentation (viewing the next communication, panning, zooming, etc.); I_{wdx} is the value of the measure of usefulness, respectively, of Webpage/Website/CLS/content [81-82].

The calculation of a set of different K_{PI} prompts to pay attention to online strategies that are most effective for generating leads, attracting users, and increasing conversions/profits of e-business. This provides an opportunity to optimize the overall structure of the Website when solving a specific NLP problem to increase the efficiency/quality of its application and the volume of regular users and customers. It is also possible to identify a set of ineffective Web pages.

Based on the analysis of data on regular users/customers, the Web page is optimized for the Website when solving a specific NLP task for the efficiency/quality of the visit/stay on it. Usually improve the structure of the Website by changing the URLs of the entry webpage for the corresponding convenient/effective visit by customers/users of the specific webpage, fixing broken links, or adjusting the corresponding content of the webpage to accommodate the necessary advertising block. Algorithm for identifying problem areas of the Website structure for further optimization:

1. Formation of a set of popular Web page entries based on the analysis of user/customer rejection rates.
2. Formation of a set of ineffective Web pages based on the analysis of the degree of usefulness and efficiency/quality to functionality.
3. Analysis of entry sources (direct entries according to the URL from the history of previous visits or the first direct visit, links to/from other Websites, links in e-mail, paid advertising, IISS, transition from social networks or search engines, etc.).
4. Analysis of the keywords of the entry relative to the sources/frequency/time.
5. Visualization of transitions on the Website from the user to achieve the goal/conversion and effectiveness/efficiency/quality of IIS.
6. Research and analysis of the effectiveness of the success of IIS on the Website.

The formation of a set of inefficient Web pages using Web analytics is carried out through the analysis of a set of relevant indicators, in particular [81-82]:

– tree of visualization of dependent sequences (Funnel Visualization);

– a set of popular entry/exit Web pages (Top Landing and Exit Pages);

– the value of the measure of usefulness of a Web page I_{wdx}, which is identified as [81-82]:

$$\mathrm{I_{wdx}} = \frac{\mathrm{R_{wcv}+R_{wec}}}{\mathrm{N_{upv}}}, \tag{26}$$

where N_{upv} is the number of unique Web page views; R_{wec} is profit from e-business; R_{wcv} is the value of the utility measure of the user visit (based on e-business transactions) and the purpose of the user visit (based on the utility of goals).

If Webpage a_i is visited by customers/users with the achievement of the goal b_j, then its usefulness affects the growth of the value of the Web page a_i usefulness. With the increase in the frequency of visits to Webpage a_i by users with the achievement of the goal b_j, and the greater the value of the goal's usefulness, the faster the degree of usefulness of the Web page I_{wdx} increases (the result is not related to conversion and goals). Webpage rating according to I_{wdx} affects the sequence of their optimization. Unexpected Web pages in the set of analyzed (not related to the goals) indicate a problem with the content and structure of the Website (multiple relevant Web pages).

The rejection rate when researching a set of popular ones is the main one. If users visit Webpage c_k through the corresponding entry point and immediately leave the Website, then this is a characteristic of the low involvement of e-business Website customers in solving a specific NLP problem. If the c_k entry webpage has a high rejection value, then the c_k webpage content does not meet the expectations and interests of customers/users/visitors. Then they analyze the sources of transitions both from other sources and in the Website between Webpages to Webpage c_k. Analysis and research of the statistics of low values of these transitions and their regularities prompts to perform relevant specific actions, in particular: improvement of advertising policy and support of Webpage/Website in relevant social networks among the typical target audience, implementation/support of relevant off-line/on-line marketing activities, activation advertising and other campaigns with paid IIS results, support for IIS optimization (SEO).

Through a detailed analysis of the entry keywords, the main goals of the users are determined according to the content of the expectations and expectations from the results of the IIS when visiting the Webpage/Website CLS. Demonstration of user transitions between Web pages on Website CLS to achieve the final goal prompts to evaluate the problematic parts of the Website structure as complex/unintelligible/incorrect order fulfilment steps. Often, users/customers use IIS on the Website as an internal technique, replacing the menu/navigation/directory on the Website. For a website with a large number of Web pages, IIS is the best solution for users to

quickly and efficiently find the text content they are looking for. Such an IIS usually uses the same framework/technique as an IISS like Google. The analysis of the success/effectiveness/operation of IIS on the Website consists of the calculation of a set of indicators, in particular[81-82]:

$$K_{iip} =< P_{wuv}, R_{ecc}, S_{wcv}, P_{wip}, P_{wcv}, N_{wvt}, R_{wcv}, R_{wec}, N_{wtr}, N_{wcv}, I_{ssp} >, \quad (27)$$

– utility value of visiting P_{wuv} Website/Webpage CLS [81-82]:

$$P_{wuv} = \frac{R_{wcv}+R_{wec}}{N_{wvt}}, \quad (28)$$

where N_{wvt} is the number of visits; R_{wec} is the usefulness of e-business; vis the utility of the goal.

– the conversion rating in e-business R_{ecc} for the CLS of the corresponding NLP task[82]:

$$R_{ecc} = \frac{N_{wtr}}{N_{wvt}} \cdot 100\%, \quad (29)$$

where N_{wvt} is the number of visits; N_{wtr} is the number of transactions.

– value of average utility S_{wcv} [81-82]:

$$S_{wcv} = \frac{R_{wcv}+R_{wec}}{N_{wcv}+N_{wtr}}, \quad (30)$$

where N_{wtr} is the number of transactions; N_{wcv} is the number of conversions; $R_{wec}$is a utility from e-business, R_{wcv} is the utility of the goal.

– the value of the e-business profit P_{wip} for the CLS of the corresponding NLP problem [82]:

$$P_{wip} = R_{wcv} + R_{wec}, \quad (31)$$

where R_{wec} is a utility of e-business; R_{wcv} is the usefulness of the purpose of the visit.

– the value of the achieved conversion P_{cv} of Website/Webpage CLS visits:

$$P_{wcv} = \frac{N_{wcv}}{N_{wvt}} \cdot 100\%, \quad (32)$$

where N_{wvt} is the number of visits; N_{wcv} is the conversion number [81-82].

Using IIS on the Website to achieve the goal, the user/customer is several times more useful than others. Hence, the creation/implementation of the IIS service on the Website effectively/qualitatively/resultatively influences the indicators of visiting the Website to attract new visitors and increase the volume of the permanent target audience. For this purpose, the calculation of the impact on the income of IIS $I_{ssp}$is used:

$$I_{ssp} = (R_{ssv} - R_{snv}) \cdot N_{ssv}, \quad (33)$$

where N_{ssv} is the number of visits from IIS to the Website; R_{snv} is the usefulness of visiting the Website without IIS; R_{ssv} is the usefulness of visiting from IIS on a Website [81-82].

The I_{ssp} indicator regulates strategies/plans for further investment in the development of the IIS service for Website and CLS as a whole to solve a specific NP problem and should be more than 80% of the monthly income for Website CLS.

IISS Marketing Activities Optimization Process (SEM) [81-82]:

1. Keyword research (for paid/unpaid IIS).
    a. Users visited according to natural IIS results.
    b. Users use the internal IIS on the Website.
2. IPP/Webpage Entry Optimization (SEO) (for all IIS results).
3. Optimization of advertising campaign (paid IIS results).
4. AdWords Ads Optimization (IIS Paid Results) ie:
    a. Positions by visiting a webpage according to the average duration of a stay on a webpage/website for a certain time.
    b. Positions by percentage of new visits (goal 1 conversion rate [for goals 2-4], bounce rate, conversion rate [average usefulness, visit usefulness, transaction, profit, e-commerce conversion rate]).
    c. Positions by time of day/season/month/week in AdWords.
    d. Positions according to the usefulness of visiting a webpage/website.
5. Optimization of AdWords ad versions (IIS paid results).
4. Availability of correctly implemented modules for linguistic processing of content in Ukrainian for effective/quality text analysis when solving a specific NLP problem using the appropriate CLS with TCLC support.

The topic of a set of keywords is one of the main indicators of IIS for identifying specific Web page content. The presence of these words or part of them on a Web page in IIS is not sufficient to add this Web page to the search results for a specific user request. Properly defined keyword themes for IIS significantly improve the quality/efficiency of CLS user visits as a result of IIS. Usually, topics contain 5-10 consistent phrases/phrases/phrases on a webpage with overlapping keywords. The more such expressions, the more difficult it is to determine the topic, which significantly reduces the rating/efficiency/quality of the Web page under IIS. It is better to divide the webpage into several according to identified thematic subsets of keywords. For sets of keywords that increase the conversion value, optimize the investment by increasing the CPC in AdWords. The return on investment value (P_{ROI}) must be positive ($N_{Inc} > N_{Exp}$) [81-82], i.e.:

$$\mathrm{P_{ROI}} = \frac{\mathrm{N_{Inc} - N_{Exp}}}{\mathrm{N_{Exp}}} \cdot 100\% > 0, \tag{34}$$

where N_{Exp} is costs; N_{Inc} is profit. Then P_{ROI} for gross profit [81-82]:

$$\mathrm{P_{ROIvp}} = \frac{(\mathrm{N_{Inc} \cdot A_{Inc}})/100 - \mathrm{N_{Exp}}}{\mathrm{N_{Exp}}} \cdot 100\%, \tag{35}$$

where A_{Inc} is the amount of profit. Then they find how much >q% of funds can be spent on a specific keyword in AdWords without the risk of getting P_{ROI}<0.

To calculate the amount of funds for attracting users, they are used:

$$\mathrm{C_{amax}} = \frac{\frac{\mathrm{N_{Inc} \cdot A_{Inc}}}{100}}{\frac{\mathrm{P_{ROIvp}}}{100} + 1}. \tag{36}$$

To calculate the amount of funds for CPS for a given keyword based on the conversion coefficients for each keyword, use:

$$\mathrm{C_{cmax}} = \mathrm{C_{amax}} \cdot \frac{\mathrm{R_{ecc}}}{100}. \tag{37}$$

Then you don't have to overpay for AdWords keywords. Basic requirements [81-82]:

1. Always consider the interests of Website users for CLS.
2. For advertising/marketing campaigns, use special Web page entries for users according to unpaid/paid IIS results.
3. Webpage entry as a result of the PPI is always next to the call to action.
4. Thematic keywords should be placed in <title> HTML tags.
5. Webpage content should be formed around a specific topic with 5-10 similar keywords for the correctness and effectiveness of IISS.
6. Do not misuse/spam keywords for IISS.
7. Thematic keywords should be meaningfully placed in HTML <a> tags.
8. Place keyword-rich content at the top of the webpage.
9. Control the IISS indexing Webpage list through the robots.txt file.
10. Do not place actual text in pictures/animations, etc.

SEV-algorithm for Website and determination of its efficiency/quality:

5. Formulation and identification of usefulness according to the goals.
6. Activation of e-business reports for CLS according to a specific NLP task:
    a. Define an unlimited number of goals (⍰4 goals for each profile).
    b. Identify the optimal volume of visits/time of the end user/customer for a successful conversion.
    c. Analyze the volume of the contribution of each goal to the total profit.
    d. Combine goals by categories/directions/species.
    e. Form separate sets of transactions as appropriate.
7. Off-line support of current marketing campaigns/customers:
    a. Based on IIS - focus on service/price/convenience etc.
    b. Encoded URLs are a well-known popular NLP service.
    c. Prestigious URLs – host everything on a central domain.
8. Support for the processing of website service content as components of e-business (downloading/saving photos, pdf/txt/xls files, etc.).

The introduction of CLS leads to an *increase in the productivity* of NLP specialists, the volume of the potential/permanent audience of system users, and the quality and efficiency of the

intellectual analysis of text content streams. At the same time, there is a reduction in the amount of time/financial/resource costs for the implementation of CLS and prompt/timely access to unique relevant textual content according to the following factors:

1. The increase in work productivity is caused by the use of automation of content integration/management/support based on the intelligent analysis of text flows and the results of the work of additional special resources such as Google Analytics and NLP specialists, in particular, analysts, programmers, linguists, administrators, moderators and feedback from the target permanent audience.
2. The analysis of the statistics/dynamics of the increase in work productivity causes the formation of a set of influencing factors on the increase in the quality and efficiency of content integration/management/support and the reduction of time/resources/finances for the implementation of CLS and prompt receipt of content by the target audience as a result of successful conversion.
3. The increase in the quality of the intellectual analysis of textual content flows is caused by the effectiveness of the analysis of statistics/dynamics and the main indicators of CLS functioning for a certain period, such as the number of unique visitors, the number of Webpage views per visit, the source of traffic and the number of transitions, new visits, the number of Website/Webpage views, content dynamics, IIS goal achieved, bounce rate, average time spent on Website/Webpage, number of visits, conversion rate, IIS keyword torus, etc.
4. Reduction of time/financial/resource costs for the implementation of CLS and prompt access to unique, relevant, relevant text content is directly proportional to the increase in the quality and efficiency of decision-making by relevant NLP specialists for intelligent text analysis when solving a specific NLP problem:
    a. by administrators for timely operational administration of the Website and CLS and formation of transaction control requests;
    b. moderators for generating relevant rules for integration, recognition, analysis, processing and synthesis of content, in particular, management, support, formatting, filtering, clustering, classification, content caching, etc.;
    c. moderators to form a list of addresses and rules for the integration of current operational data from reliable sources;
    d. by authors to generate unique relevant current text content according to the ranking list of current requests from the target audience according to the current topic;
    e. analysts for analysis of statistics/dynamics of CLS functioning, generation of story identification rules, personalization of work with a permanent audience, and content ranking.

The organizational effect is caused by a number of the following factors:

1. By reducing the number of NLP specialists (1-3 analysts, 1-2 administrators, 1-2 programmers, 1-2 linguists, 1-10 authors, 1-3 moderators, 1-2 SA experts, for example, psychologists) involved in stages of development and implementation of CLS for solving a specific NLP problem;
2. Changing/fixing the organizational structure of the project (functional division between NLP specialists of the project, i.e. a linguist does not perform the work of an analyst, and an expert does not perform the work of a moderator, etc., but it is possible to combine functions in some simple NLP tasks or interchange them);
3. Reduction of the number of functions of NLP specialists of the CLS project (partial automation based on intellectual analysis of text flows);
4. Support for the regulation of intellectual analysis of text content streams for the implementation of decision-making functions based on content integration/management/support modules (integration of information for users/authors, recording/analysis of results/statistics/dynamics of requests/actions of the target audience and other statistical data for moderators /analysts/administrators/linguists/experts).

The technological effect is caused by the reduction/release of resources as NLP specialists, the high-quality/effective application of the modules of intellectual analysis of text content flows in CLS, the relatively fixed distribution of functions between the NLP specialists of the project, as well as the implementation of new IT as integration/management/support of content and organization/ analysis of feedback from the permanent/potential target audience.

The social effect contributes to the growth of the target audience, the number of unique/regular users of the Website, accessibility to relevant content/Webpage/Website, coverage of a wide range of social audiences, etc., based on the regulation of the content/topics of the Website. Support of topically relevant and similar textual content, integration of operational unique text and corresponding management of it through the Website regulates the limits of the volume of CLS's social target permanent audience and helps to predict/regulate these changes.

The advertising effect based on the application of templates for the Website, Webpage/content and integration/generation/creation of unique relevant content contributes to the increase in the number of visits of users with IISS and is a kind of self-promotion of Website CLS, a set of Webpage services/content. The use of Google Analytics/AdWords results significantly facilitates the analysis of e-business indicators, advertising and Website/CLS functioning.

The psychological effect facilitates the organization/implementation of user-friendly interactive interface support for each NLP specialist, user and customer of the Website CLS based on dynamic feedback. This significantly facilitates the performance of duties for linguists, analysts, administrators, moderators, and authors, as well as the collection/analysis of psychological indicators of regular users/customers/visitors of CLS based on the personalization of work with them through a friendly interactive website interface.

The ergonomic effect contributes to the growth of the influence of the results of the operation of CLS and modules of intellectual analysis of textual content flows through support/management/integration of textual content based on the calculation/analysis of the number of traffic sources in %, absolutely unique visitors, new visits (%), Webpage/Webpage views for all /one visit, IIS conversion achieved, rejections (%), visits, as well as content processing dynamics (%), average visit time on the Website (min:c), etc.

4.1.4. Input flow of the content of a computer linguistic system

A classified list of the incoming stream of content with a set of relevant properties/characteristics/parameters helps distinguish project participants through their typification and restriction of access rights depending on the content: regular users, potential visitors, linguists, statistical analysts, website administrators, content/rules moderators, authors of unique content, information resource as a source of content, etc. The typed structure of the content input stream template with a set of relevant properties/attributes/parameters helps to define the main functional requirements for the Website/CLS and its typical structure and delineate the non-functional capabilities, classify the sources, calculate the integration frequencies and the corresponding restrictions/conditions of the integration from the typical source. The input content streams to CLS are typical components:

$$X =< X_a, X_s, X_q, X_f, X_s, X_w, X_b, X_d, X_k, X_v, X_u, X_r, X_t, X_o >, \quad (38)$$

– X_a is Website URLs of the sources for the CLS filter DB;

– X_s is content as a result of integration from various X_a sources according to a predefined list of URLs without a predefined structure in HTML/XML format according to relevant thematic requests

– X_q is thematic requests of visitors/users of Website CLS in the form of a set of keywords or stable phrases;

– X_f is the actual data of persistent users/profiles and the set of rules for allowed actions within the respective CLS user type;

– X_s is statistical data of actions/events/phenomena of CLS subjects/objects of the solution of the corresponding NLP task and the rules of collection/storage/analysis of statistics in certain time intervals of CLS operation;

– X_w is statistical data on the operation of Website CLS, collected with a specified frequency from Google Analytics in the form of XML tables;

– X_b is the contents of content databases/rules/filters/annotations etc. CLS;

– X_d are different types of linguistic dictionaries depending on the purpose of CLS for solving a specific NLP problem;

– X_k is a set of personalized/anonymous feedback/comments of visitors/users to the relevant content of the Website CLS;

– X_v is a tuple of the results of personalized/anonymous votes of regular/potential visitors/users on CLS content;

– X_u is statistical personalized individual actions of CLS users;

– X_r is a set of external/internal advertising of thematic content;

– X_t is thematic stickers of entertainment/informational content (exchange rates, announcements, digests, weather, anecdotes, horoscope, etc.);

– X_o is a tuple of CLS/Website configuration and configuration options tuple.

4.1.5. The output stream of content of a computer linguistic system

The filling of the tuple of the source processed text according to the purpose of CLS for the solution of a specific NLP problem directly depends on the content of the incoming classified stream of content with a predefined set of relevant properties/characteristics/parameters depending on the interaction of the Website of the relevant types of project participants (regular users, potential visitors, linguists, statistical analysts, website administrators, content/rules moderators, authors of unique content, information resource as a source of content, etc.):

$$Y = < Y_c, Y_q, Y_a, Y_v, Y_s, Y_p, Y_t, Y_r, Y_o, Y_k >, \quad (38)$$

– Y_c is text content as an information product or the result of providing a corresponding information service for solving a specific NLP problem on the Website;

– Y_q is a set of meaningfully generated/cached Web pages as a result of thematic requests/IIS of users/visitors of Website CLS;

– Y_a are annotations/digests/abstracts on textual thematic content;

– Y_v is a tuple of statistics of interaction of users/visitors with the Website;

– Y_s is a tuple of the content of the profiles of regular CLS users according to the personalized statistics Y_v for the corresponding generation of an individual portrait of the user/audience at certain time intervals;

– Y_p is a tuple of meaningful recommended content of a Webpage Website, personalized for a specific regular user according to the profile/actions/interaction with CLS at certain time intervals;

– Y_t is multiple content topics/headings with the possibility of renewal according to the results of the latest IIS/requests from regular users of the Website;

– Y_o is a diagram of relationships of textual thematic content according to the appropriate classification (current, relevant, copyrighted, outdated, popular, similar, last-viewed, often-viewed, sequentially most viewed, longest viewed, most viewed from search engines or internal IIS, viewed a typical group of users, etc.);

– Y_r is a set of content ranking results on a predetermined scale within the relevant ranking classification;

– Y_k is a set of marked evaluations/ratings of user comments as the degree of permission to publish on the Website/Web page, if necessary, with a prohibition mark for a specific contributor to write further comments and ranking by the degree of trust of all contributors.

The list of the output stream of content, its main characteristics and the corresponding classification, IT generation/support/analysis helps to define the clear general functional requirements of the CLS implementation for the solution of any NLP problem.

4.2. Functional requirements for the design of a typical CLS

4.2.1. Requirements for software modules of a typical CLS

Functional/non-functional requirements for a typical CLS are the main components for designing and developing software for solving a specific NLP problem. Functional requirements form the direction of development and implementation of a typical CLS, but in most cases, they cannot be calculated and measured (measured as a set of inputs to the CLS and a set of outputs that are checked). Non-functional requirements allow you to dream about the quality of development and the effectiveness of CLS implementation based on feedback from a permanent audience and the rate of growth of the volume of permanent users and the conversion of their actions. The functional requirements for a typical CLS are a set of descriptive instructions regarding the internal functioning of the IS and changing the dynamics of its behaviour depending on the system states through the definition of a set of specific functions/modules for solving a specific NLP problem, in particular, content processing/modification, data manipulation/operation, data integration/calculation, etc. The main typical requirements for CLS are compliance with standards, accuracy/correctness of output for input, security of software and compatibility with different modules/software/IS. General typical requirements for CLS:

- support for dynamic management of CLS/Website transactions;
- support for rapid implementation of WebOLTP applications for CLS;
- prompt and effective interaction between the browser and the back-end DB;
- performance/scalability and quality/efficiency of operation with large volumes of transactions, sessions, users/visitors and simultaneous access of databases/repositories of content/rules, etc.

The following built-in software is used to support the management of basic typical transactions during the operation of CLS/Website:

- calls of distributed elements for timely operational high-quality support of the relationship in the multi-level structure of CLS/Website;
- services for effective operational launch/management of servlets;
- CLS/Website/Webpage quality transaction management web services;
- tools for rapid operational qualitative development/modification and software support for intermediate IS component/module level.

CLS must support a minimum of 6 interfaces for interaction with a specific type of project participant depending on rights and functionality:

- with limited access for regular/potential Website visitors (Fig. 11) with the ability to quickly find the necessary information;

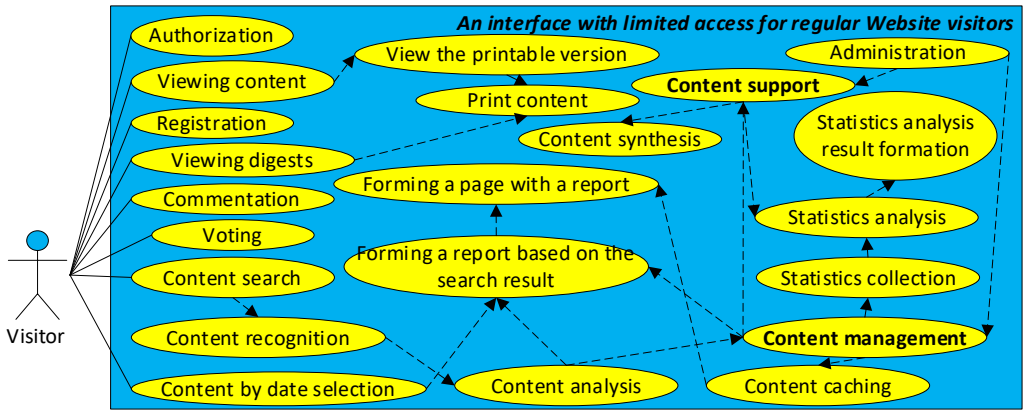

Figure 11: Use case diagram for CLS-restricted visitor access

- with limited personalized access for users (Fig. 12);
- with unrestricted access for the CLS/Website administrator (Fig. 13) with the ability to adjust the Website/CLS structure, relevant Web page/content templates, access rights of participants, and content distribution rules;

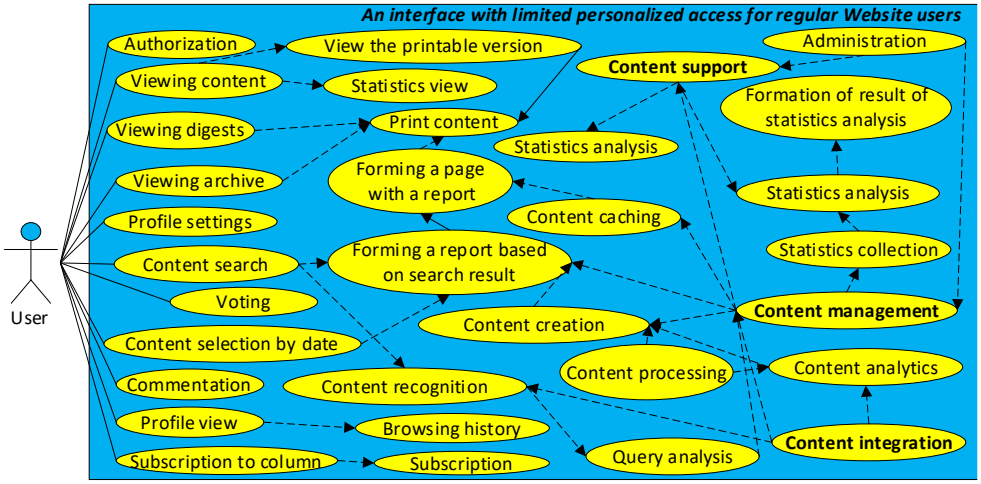

Figure 12: Use case diagram for limited access of Website users

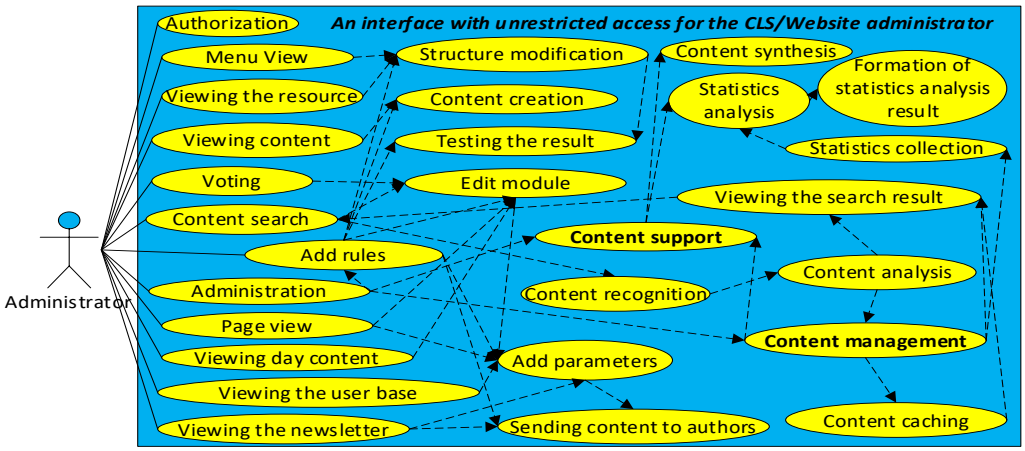

Figure 13: Use case diagram for Website/CLS administrator free access

- with free access to certain CLS modules for the moderator (Fig. 14) with the ability to adjust parameters/rules/configuration of IIS, filtering, analysis, monitoring, categorization, etc. of content;

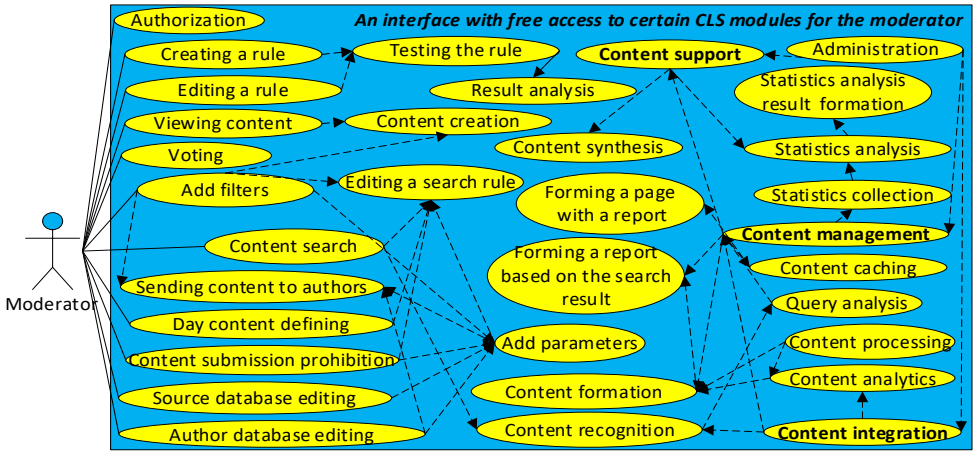

Figure 14: Use case diagram for Website CLS moderator access

- with free access to certain CLS modules for the analyst (Fig. 15);

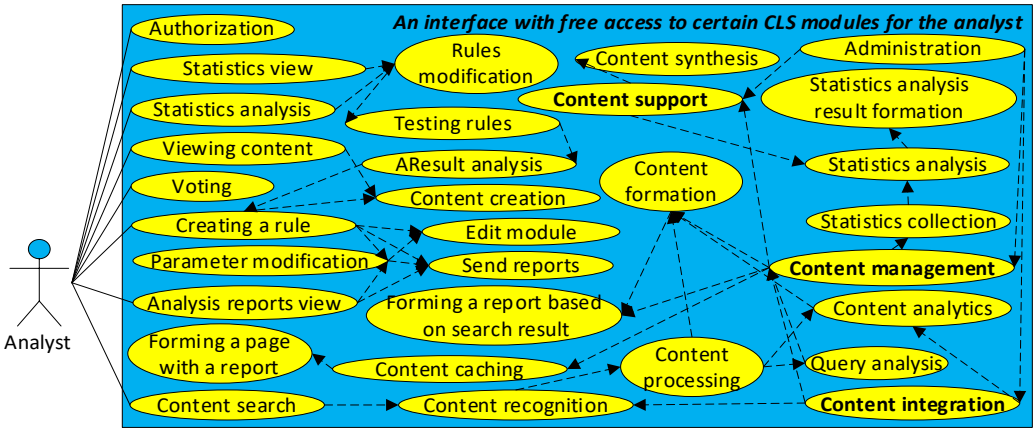

Figure 15: Use case diagram for Website/CLS analysts

- with partial limited access for the author/linguist of Website CLS content (Fig. 16).

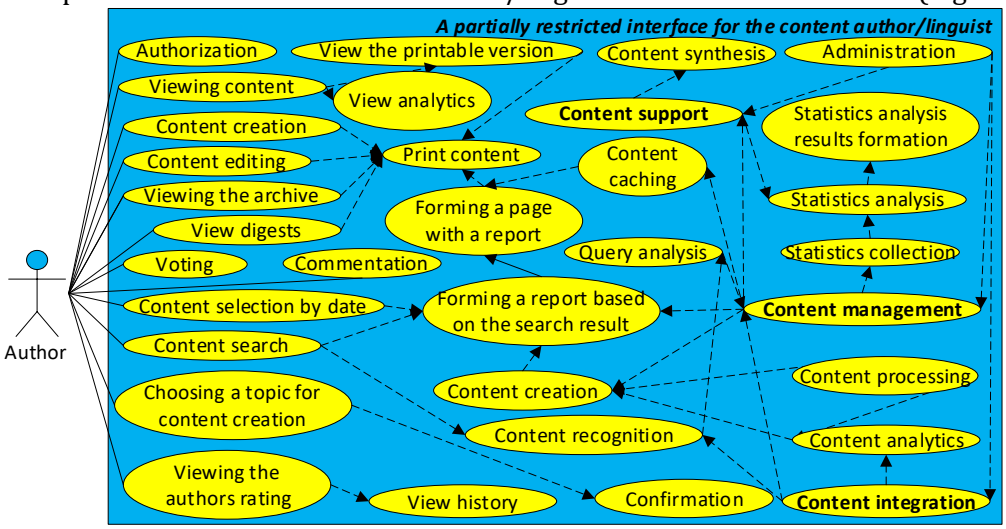

Figure 16: Use case diagram for linguists/content authors Website CLS

The definition of functional requirements for modules of support, management and integration of CLS textual data prompts the development of the general structure of the corresponding IS. A correctly designed website facilitates the interaction of project participants with CLS and, accordingly, supports the possibility of increasing the functionality of corresponding CLS solutions to a specific NLP problem.

The content maintenance module generates a set of relevant irrelevant queries based on statistical data of the interaction of regular users/visitors of the Website for the further generation of a list of thematic subjects/requests of potentially relevant content ranked by popularity. This list is used as input data for the data integration module from various reliable sources (information resources) and for permanent authors of unique content. The author has the opportunity to familiarize himself with such a list to create relevant textual content relevant to the permanent audience/IS/modules in CLS (for example, in media-information systems, recommender systems, systems for analyzing the psychological state of a person, voiceless access interfaces, etc.) based on selected and integrated textual content from various reliable information sources as a basis for research in content generation.

A linguist, in addition to creating unique content, can renew or develop new linguistic e-dictionaries (not only words, keywords and fixed phrases, but morphemes, inflexions, exclusions, bases, etc.), but thematic and other special ones, as well as select text corpora for CLS training.

The moderator develops different rules for processing text content based on the research of a linguist, the needs of the author, statistical data of the analyst regarding the popularity of the IIS results of the thematic content (especially if it is small in volume or absent in terms of the

frequency of refusals from transitions from search engines). Also, the moderator implements content filtering rules when integrating from various sources, internal IIS content based on user requests, annotating and referencing content, identifying duplicates in DB/DS, caching information blocks as a stage of content management, and analyzing personalized user profiles/history of actions and determining thematic plots as a content support stage. If necessary, in cooperation with a linguist, the moderator forms the rules of speech synthesis and recognition, text analytics and text generation, as well as the development/formation of the appropriate text array of data. The analyst develops various rules for the collection/storage/analysis of statistical data on the functioning of the CLS and the actions/events of the permanent target audience in determining the time intervals of a certain periodicity. Also, the analyst generates rules for statistical analysis of the dynamics/frequency of implementation of the TCLC CLS stages for further identification of the thematic/content interest of the permanent (according to the actions of Website users) or potential (according to the actions of unique visitors) target audience. A timely operational response to changes in the interest of the target audience helps to modify the directions of content integration to support the growth of the number of direct/IISS/resource visits with achieved conversion, repeated/unique/regional/thematic visits of CLS, which in turn leads to an increase in the volume of the Website target audience. Also, the rules for collecting/saving/analyzing statistical data of ratings/content headings/authors, website functioning, and periodic activity of website users/visitors following CLS objects are modified.

4.2.2. Basic additional requirements of the network, software and technical tools for the software implementation of a typical CLS

The formation of functional requirements for the module of intellectual analysis of text streams of content in CLS accordingly specifies additional requirements of the network, software and technical environment for the implementation of a typical CLS, in particular, for support/management/integration of Website/CLS/Webpage content (Table 5). The content support module is a supporting tool for Website/CLS administrators and analysts. Content management module – for users, visitors, administrators and Website/CLS moderators. Content integration module – for authors, linguists and Website/CLS moderators.

Table 5
Tools for intellectual analysis of text content streams

Tools	Description
HTTPS, FTP, HTTP, RMI-IIOP, GIOP, IIOP	Communication protocols between the Web server and the user.
SOAP, REST/ Atom	Object access/interaction protocol/rules
SSL, TLS	Domain/Recipient Secure Link Certificates
CGI, Python, R, PHP, Apache, API	Web server integration with content sources.
HTML,CSS,WML,HDML,XML,XHTML,JavaScript	Support for hypertext links.
GifCam,Flash,JavaScript,CSS,audio/video format, VRML	Support for multimedia effects.
IMAP, SMTP, POP3, UDP, LMTP, XML-RPC, CMIP	Support for interactive interaction/communication.
Python, PHP, R, JavaScript	Implementation of NLP-task processes.
Joomla, WordPress, Drupal, LiteDiary, SiMan CMS,	Content management systems.
Django, Tornado, Pyramid, Flask, TurboGears	Web framework on Python
Zend, FuelPHP, CakePHP, Phalcon, Yii, CodeIgniter, Symfony, Laravel	Web framework on PHP
ECM, CMIS, WSDL	WebService content management
EDGE, UMTS, GPRS, WAP, VPN	Support for mobile access/computing.
CORBA, UML, DCOM, COM, ORB, SWIG	Creation of distributed objects.
DBMS of MySQL, filesystem, OC, Oracle	Data storage and processing.

The choice of NLP specialists between CMS and Web framework for the development of the CLS project depends on the results of the analysis of their advantages/disadvantages. The main advantage of Web frameworks is that they have a wide range of tools for the full development/support of any Web application. No need to search/create separate libraries for each separate task and solve compatibility issues. A web framework is like a Lego constructor (Table 6). The Text Mining model of content is directly related to the ML process - finding a model with a collection of functions, an algorithm and hyperparameters that find better results on training data to evaluate previously unknown data. The process consists of creating a training set (corpus), analysis of feature extraction methods, and preliminary processing - converting text into numerical data for further understanding by ML processes based on text classification and clustering. Since ML is applied to Text Mining of content, a programming language with a large number of built-in/additional scientific and computational libraries like Python (Table 7) is needed.

Table 6
Comparison of CMS and Web Framework

Characteristic	CMS	Web framework
Ease of maintenance of the CLS project.	+/-	+
The presence of a set of business processes embedded in the software	+/-	+
It is possible and relatively simple to implement business processes that are not embedded in the software.	+/-	+
CLS projects are easily scalable and modernized	+	+
Solutions work much faster.	-	+
Solutions can withstand heavy loads	-	+
Support for a high level of security.	+	+
The terms of development of typical functionality are short.	+	-
Availability of more than basic application-level business logic components	+/-	-
The need to implement many functions individually for a specific CLS.	+/-	+
Development does not require an understanding of the business processes to be implemented.	+	-
Built-in support for many business processes, such as order processing	+	-
No specialized ranks/skills are required to administer/upgrade CLS	+	+/-

Table 7
Python tools for implementing Text Mining content

Library	Characteristic	Features
Scikit-Learn	An extension of the SciPy (Scientific Python) library to support a program interface (API) for generalized ML.	Based on Cython with support for high-performance C libraries (Boost, LibSVM, LAPACK, etc.), the ScikitLearn extension combines high performance with ease of use for small/medium dataset analysis techniques. The open-source, commercially available extension provides a single interface for many classification, regression, clustering, dimensionality, and cross-validation/hyperparameter tuning models.
Yellowbrick	A set of visual diagnostics tools for analyzing/interpreting ML results, a Scikit-Learn API application.	Provides simple and intuitive visual tools for selecting functions, modelling and setting hyperparameters, and managing the process of selecting models for the most effective description of textual data.
NetworkX	A comprehensive graph analysis package to help create, organize, analyze, and manipulate complex network structures.	Not an ML or Text Mining content library, but the use of graph data structures allows encoding complex relationships that graph algorithms can analyze and find semantic features and is therefore an important tool for text analysis.

Library	Characteristic	Features
spaCy	A tool for implementing a high-quality NLP process based on modern complex algorithms through a simple and convenient API.	Support for preliminary processing of the text within the framework of preparation for deep learning. It is used to create IS information extraction or natural language analysis on large volumes of text.
Gensim	A reliable, effective and simple tool for semantic text modelling without a teacher.	It is designed to search for similarities in texts, supports topic modelling for hidden semantic analysis methods, and has other ML libraries (for example, word2vec).
NLTK	A package of NLP tools (Natural Language ToolKit).	Contains a corpus, lexical resources, grammar, NLP algorithms and pre-trained models for implementing fast processing of textual data from various natural languages.
pandas	Data analysis.	Analysis of numerical data.
TextBlob	NLTK extension	Phrase extraction, PoS tagging, tokenization, sentiment analysis, classification and SYA

Natural language processing is a promising AI branch of artificial intelligence for understanding and interpreting human speech by computers. The application of NLP methods, ML and the best tools for the interpretation of textual data allows CLS to conduct analysis in a timely and efficient manner and make relevant conclusions/forecasts or choose the optimal solution in response to the relevant set of input data. NLP techniques include tokenization, text normalization, and data cleaning. In a standard format, various ML methods are applied for the best interpretation and understanding of the data. For example, this includes applying relevant modelling techniques to classify e-mails as spam/non-spam or to estimate the sentiment of a tweet on Twitter. Newer, more complex methods are also used for topic modelling, keyword extraction, or text generation based on deep learning.

CLS development technology is support for full/partial automation of business processes (including natural language processing) for solving a specific NLP problem. In CLS, based on the support of business processes, tasks, subprocesses, information, messages, documents, content, etc. are transferred for the implementation of relevant actions/events from one type of actor (participant) to the next according to a collection of embedded procedural/associative rules of advanced NLP models from more rich sets of text analysis functions. The content context is presented/implemented as NLP functions and organizes their visual interpretation for analysts/moderators to control the model selection process. Complex relationships extracted from the text are usually analyzed based on graph analysis methods. CLS interprets, implements and manages the workflow (business process) based on the software in the form of modules that analyze and implement the interpretation of the process, interact with the objects/subjects of the workflow and refer to the corresponding modules/tools if necessary.

CLS automates the business process of solving a specific NLP problem and implements the rules of interaction of process objects/subjects. These moments of interaction (dialogue) are the main aspects of losses due to the uncertainty/ambiguity of the interpretation of the input data (understanding the syntactic/semantic analysis of the text and the selection/implementation of the appropriate production/associative rule). Scaling text analysis in multiprocessor CLS using Spark and implementing text analysis through deep learning can be a solution to this problem. The result of the implementation of the NLP project can be not only an independent CLS for solving a specific NLP problem but also a software built-in module in IS such as Internet-publishing, distance learning, Internet-publishing, Internet-magazine, Internet-newspaper, Internet-shop sales of content such as electronic books, audio video, photos, software, etc.

The development of a set of functional requirements for the construction of a typical CLS contributes to the creation for developers and NLP specialists of a generalized IT implementation of the corresponding IS/modules to significantly reduce the amount of time/resources for the design/construction/implementation/modernization/improvement of the corresponding software NLP modules. The requirements for results/regulations of CLS functioning, ways of submission/transmission/saving/modification/interpretation/destruction of textual/service

data depend on the implementation of subsystems of intellectual analysis of textual content flows as support/management/integration of content.

The requirements for compatibility and ways of exchanging/interacting textual/service data with other IS/modules/participants consist of conditions for implementing and supporting the processing of text arrays of content in HTML/XML format.

Support of regulatory and organizational requirements for participants/modules, their qualifications and composition, regulations/time of IS operation, powers and rights for interaction with IS, etc. provide an opportunity to support CLS functioning at an appropriate level, promptly/qualitatively implement/implement CLS, and timely full-scale analysis results of approbation of IS activities and main subsystems of intellectual analysis of text content streams.

The ergonomic requirements for CLS are the comfort of IS management tools, the rational layout of software/interface modules, the convenience/operability of IS service/support/support, and the aesthetic design of the interactive user interface. CLS should provide an appropriate level of protection/security for personal data and other IS components against unauthorized access, destruction, loss, and damage to information.

5. Conclusions

The developed IT processing of Ukrainian-language text content, unlike the existing ones, supports the modularity principle of the typical CLS architecture for solving a specific NLP problem and analysing a set of parameters and metrics of the system's functioning by the behaviour of the target audience. The general structure of CLS for the processing of text content in the Ukrainian language and the conceptual scheme/model of the functioning of a typical CLS based on the modelling of the interaction of the main processes and components of the system were developed, which made it possible to improve IT intellectual analysis of the text flow based on the processing of information resources. The peculiarities of the design and development of computer linguistic systems are analysed based on the definition of the main stages such as grapheme, morphological, lexical, and syntactic-semantic analysis/synthesis of the Ukrainian-language text for the solution of a specific NLP problem. The formulation of the problem of processing the Ukrainian-language text based on the definition of the functional features of the intellectual analysis of the text flow was made and specified. The general analysis of the problem of analysis of the Ukrainian-language text and the definition of the main problems of the processing of the Ukrainian-language text made it possible to formulate the main stages and requirements for the project of a typical CLS solution of a specific NLP problem. Identification of the main characteristics of CLS and justification of the project implementation of a typical CLS made it possible to determine the expected effects of the corresponding project implementation. Based on the analysis of the input/output streams of the content of the computer linguistic system, the functional requirements for the project of a typical CLS, its software modules, network, software and technical tools of IS software implementation are defined and formulated.

References

[1] V. Vysotska, S. Mazepa, L. Chyrun, O. Brodyak, I. Shakleina, V. Schuchmann, NLP Tool for Extracting Relevant Information from Criminal Reports or Fakes/Propaganda Content, in: IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, November, pp. 93-98. IEEE.

[2] A. Mykytiuk, V. Vysotska, O. Markiv, L. Chyrun, Y. Pelekh, Technology of Fake News Recognition Based on Machine Learning Methods, CEUR workshop proceedings 3387 (2023) 311-330.

[3] S. Mainych, A. Bulhakova, V. Vysotska, Cluster Analysis of Discussions Change Dynamics on Twitter about War in Ukraine, CEUR workshop proceedings 3396 (2023) 490-530).

[4] S. Kubinska, R. Holoshchuk, S. Holoshchuk, L. Chyrun, Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining, CEUR Workshop Proceedings 3171 (2022) 315-327.
[5] A. Berko, Y. Matseliukh, Y. Ivaniv, L. Chyrun, V. Schuchmann, The text classification based on Big Data analysis for keyword definition using stemming, in: Proceedings of the IEEE 16th International conference on computer science and information technologies, CSIT-2021, Lviv, Ukraine, 22–25 September 2021, pp. 184–188.
[6] V. Vysotska, S. Holoshchuk, R. Holoshchuk, A comparative analysis for English and Ukrainian texts processing based on semantics and syntax approach, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 311-356.
[7] B. Bengfort, R. Bilbro, T. Ojeda, Applied text analysis with Python: Enabling languageaware data products with machine learning. O'Reilly Media, Inc. (2018).
[8] D. Jurafsky, J. H. Martin, Deep Learning Architectures for Sequence Processing. URL: https://web.stanford.edu/~jurafsky/slp3/9.pdf.
[9] D. Jurafsky, J. H. Martin, Naive Bayes and Sentiment Classification. URL: https://web.stanford.edu/~jurafsky/slp3/4.pdf.
[10] D. Jurafsky, Logistic Regression. URL: https://web.stanford.edu/~jurafsky/slp3/5.pdf.
[11] D. Jurafsky, J. H. Martin, Neural Networks and Neural Language Models. https://web.stanford.edu/~jurafsky/slp3/7.pdf.
[12] N. Shakhovska, I. Shvorob, The method for detecting plagiarism in a collection of documents, in: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 2015, pp. 142-145.
[13] R. Romanchuk, V. Vysotska, V. Andrunyk, L. Chyrun, S. Chyrun, O. Brodyak, Intellectual Analysis System Project for Ukrainian-language Artistic Works to Determine the Text Authorship Attribution Probability, in: Proceedings of the 18th IEEE International Conference on Computer Science and Information Technologies, CSIT 2023, Lviv, Ukraine, October 19-21, 2023. IEEE 2023.
[14] V. Lytvyn, P. Pukach, V. Vysotska, M. Vovk, N. Kholodna, Identification and Correction of Grammatical Errors in Ukrainian Texts Based on Machine Learning Technology. Mathematics 2023, 11, 904. https://doi.org/10.3390/math11040904
[15] K. Shakhovska, et al., An approach for a next-word prediction for Ukrainian language. Wireless Communications and Mobile Computing 2021 (2021) 1-9.
[16] I. Khomytska, I. Bazylevych, V. Teslyuk, I. Karamysheva, The chi-square test and data clustering combined for author identification, in: Proceedings of the IEEE XVIIIth Scientific and Technical Conference on Computer Science and Information Technologies, CSIT 2023, Lviv, Ukraine, 19-21 October 2023.
[17] I. Khomytska, V. Teslyuk, The Multifactor Method Applied for Authorship Attribution on the Phonological Level, CEUR workshop proceedings 2604 (2020) 189-198.
[18] V. Vysotska, Ukrainian Participles Formation by the Generative Grammars Use, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 407-427.
[19] O. Bisikalo, O. Boivan, N. Khairova, O. Kovtun, V. Kovtun, Precision automated phonetic analysis of speech signals for information technology of text-dependent authentication of a person by voice, CEUR Workshop Proceedings 2853 (2021) 276–288.
[20] I. Khomytska, V. Teslyuk, A. Holovatyy, O. Morushko, Development of methods, models, and means for the author attribution of a text, Eastern-European Journal of Enterprise Technologies. 3(2(93)) (2018) 41–46. doi: 10.15587/1729-4061.2018.132052.
[21] O. Bisikalo, V. Vysotska, Linguistic analysis method of Ukrainian commercial textual content for data mining, volume Vol-2608 of CEUR Workshop Proceedings, 2020, pp. 224-244.
[22] T. Batura, A. Bakiyeva, M. Charintseva, A method for automatic text summarization based on rhetorical analysis and topic modeling, International Journal of Computing 19(1) (2020) 118-127. doi: 10.47839/ijc.19.1.1700.

[23] V. Husak, O. Lozynska, I. Karpov, I. Peleshchak, S. Chyrun, A. Vysotskyi, Information System for Recommendation List Formation of Clothes Style Image Selection According to User's Needs Based on NLP and Chatbots, CEUR workshop proceedings 2604 (2020) 788-818.
[24] N. Shakhovska, O. Basystiuk, K. Shakhovska, Development of the Speech-to-Text Chatbot Interface Based on Google API, CEUR Workshop Proceedings 2386 (2019) 212-221.
[25] D. Lande, L. Strashnoy, GPT Semantic Networking: A Dream of the Semantic Web–The Time is Now. URL: https://ela.kpi.ua/server/api/core/bitstreams/299901e4-b9b9-457b-9f07-a0808f3973ba/content.
[26] D. Lande, et al., Link prediction of scientific collaboration networks based on information retrieval, World Wide Web 23 (2020) 2239-2257.
[27] M. Fu, et al. Integration of complete ensemble empirical mode decomposition with deep long short-term memory model for particulate matter concentration prediction, Environmental Science and Pollution Research 28 (2021) 64818-64829.
[28] M. Fu, J. Feng, D. Lande, O. Dmytrenko, D. Manko, R. Prakapovich, Dynamic model with super spreaders and lurker users for preferential information propagation analysis, Physica A: statistical mechanics and its applications 561 (2021) 125266.
[29] D. V. Lande, A. A. Snarskii, E. V. Yagunova, E. V. Pronoza, The use of horizontal visibility graphs to identify the words that define the informational structure of a text, in: IEEE 12th Mexican International Conference on Artificial Intelligence, 2013, November, pp. 209-215.
[30] Senyk M. Project: Static tree of endings for the Ukrainian language. URL: http://www.senyk.poltava.ua/projects/ukr_stemming/ukr_endings.html
[31] Senyk M. The Porter Stemming Algorithm for Ukrainian. URL: http://www.senyk.poltava.ua/projects/ukr_stemming/stemming_about.html
[32] R. Nazarchuk, S. Albota, Tweets about Ukraine during the russian-Ukrainian War: Quantitative Characteristics and Sentiment Analysis, CEUR Workshop Proceedings 3426 (2023) 551-560.
[33] M. Konyk, V. Vysotska, S. Goloshchuk, R. Holoshchuk, S. Chyrun, I. Budz, Technology of Ukrainian-English Machine Translation Based on Recursive Neural Network as LSTM, CEUR Workshop Proceedings 3387 (2023) 357-370.
[34] V. Vysotska, Y. Burov, V. Lytvyn, A. Demchuk, Defining Author's Style for Plagiarism Detection in Academic Environment, in: Proceedings of the International Conference on Data Stream Mining and Processing, DSMP, 2018, pp. 128-133. doi: 10.1109/DSMP.2018.8478574.
[35] V. Lytvyn, et. al., Development of the quantitative method for automated text content authorship attribution based on the statistical analysis of N-grams distribution, Eastern-European Journal of Enterprise Technologies 6(2(102)) (2019) 28–51. doi: 10.15587/1729-4061.2019.186834.
[36] R. Lynnyk, et. al., DDOS attacks analysis based on machine learning in challenges of global changes, CEUR Workshop Proceedings 2631 (2020) 159-171.
[37] O. Barkovska, V. Kholiev, A. Havrashenko, D. Mohylevskyi, A. Kovalenko, A Conceptual Text Classification Model Based on Two-Factor Selection of Significant Words, CEUR Workshop Proceedings 3396 (2023) 244-255.
[38] I. Khomytska, V. Teslyuk, Authorship and Style Attribution by Statistical Methods of Style Differentiation on the Phonological Level, Advances in Intelligent Systems and Computing 871 (2019) 105–118. doi: 10.1007/978-3-030-01069-0_8.
[39] A. Taran, Terminology of Computational Linguistics in Terms of Indexing and Information Retrieval in the System "iSybislaw", CEUR Workshop Proceedings 2870 (2021) 225-234.
[40] N. Kunanets, H. Matsiuk, Use of the Smart City Ontology for Relevant Information Retrieval, CEUR Workshop Proceedings 2362 (2019) 322-333.
[41] K. Nataliia, M. Halyna, Application of Saaty Method While Choosing Thesaurus View Model of the "Smart city" Subject Domain for the Improvement of Information Retrieval Efficiency, in: Proceedings of the IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018, Vol. 2, Art No. 8526656, 2018, pp. 21-25. doi: 10.1109/STC-CSIT.2018.8526656.

[42] E. Fedorov, O. Nechyporenko, Linguistic Constructions Translation Method Based on Neural Networks, CEUR Workshop Proceedings 3396 (2023) 295-306.
[43] V. Lytvyn, Y. Burov, V. Vysotska, Y. Pukach, O. Tereshchuk, I. Shakleina, Abstracting Text Content Based on Weighing the TF-IDF Measure by the Subject Area Ontology, in: Proceedings of the IEEE International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2021. URL: https://ieeexplore.ieee.org/document/9465978.
[44] A. Sartiukova, O. Markiv, V. Vysotska, I. Shakleina, N. Sokulska, I. Romanets, Remote Voice Control of Computer Based on Convolutional Neural Network, in: Proceedings of the IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 07-09 September 2023, pp. 1058-1064.
[45] V. Lytvyn, et al., 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(95)) (2018) 16–28. doi: 10.15587/1729-4061.2018.142451.
[46] V. Lytvyn, et. al. Development of the system to integrate and generate content considering the cryptocurrent needs of users, Eastern-European Journal of Enterprise Technologies 1(2(97)) (2019) 18–39. doi: 10.15587/1729-4061.2019.154709
[47] A. Chiche, H. Kadi, T. Bekele, A Hidden Markov Model-based Part of Speech Tagger for Shekki'noono Language, International Journal of Computing 20(4) (2021) 587-595. doi: 10.47839/ijc.20.4.2448.
[48] S. A. Thorat, K. P. Jadhav, Improving Conversation Modelling using Attention Based Variational Hierarchical RNN, International Journal of Computing 20(1) (2021) 39-45. doi: 10.47839/ijc.20.1.2090.
[49] I. Lauriola, A. Lavelli, F. Aiolli, An introduction to deep learning in natural language processing: Models, techniques, and tools, Neurocomputing 470 (2022) 443-456.
[50] Y. Kang, et. al., Natural language processing (NLP) in management research: A literature review, Journal of Management Analytics 7(2) (2020) 139-172.
[51] L. Hickman, S. Thapa, L. Tay, M. Cao, P. Srinivasan, Text preprocessing for text mining in organizational research: Review and recommendations, Organizational Research Methods 25(1) (2022) 114-146.
[52] D. Hu, An introductory survey on attention mechanisms in NLP problems, in: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), Volume 2, 2020, pp. 432-448.
[53] Gardner, M., Merrill, W., Dodge, J., Peters, M. E., Ross, A., Singh, S., & Smith, N. A. (2021). Competency problems: On finding and removing artifacts in language data. arXiv preprint arXiv:2104.08646.
[54] L. Wu, et al., Graph neural networks for natural language processing: A survey, Foundations and Trends® in Machine Learning 16(2) (2023). 119-328.
[55] M.-A. Lefer, N. Grabar, Super-creative and overbureaucratic: A cross-genre corpusbased study on the use and translation of evaluative prefixation in ted talks and eu parliamentary debates, Across Languages and Cultures 16(2) (2015) 187–208.
[56] D. Jurafsky, J. H. Martin, Speech and Language Processing. URL: https://web.stanford.edu/~jurafsky/slp3/ed3book_sep212021.pdf.
[57] J. Weizenbaum, ELIZA – A computer program for the study of natural language communication between man and machine, CACM 9 (1966) 36–45.
[58] J. Weizenbaum, Computer Power and Human Reason: From Judgement to Calculation. W.H. Freeman and Company. 1976.
[59] ElizaBot. URL: https://www.masswerk.at/elizabot/.
[60] ELIZA: a very basic Rogerian psychotherapist chatbot. URL: https://web.njit.edu/~ronkowit/eliza.html.
[61] D. Jurafsky, J. H. Martin, Regular Expressions, Text Normalization, Edit Distance. URL: https://web.stanford.edu/~jurafsky/slp3/2.pdf.
[62] O. Karnalim, G. Kurniawati, Programming style on source code plagiarism and collusion detection, International Journal of Computing 19(1) (2020) 27-38.

[63] V. Claveau, T. Hamon, S. Le Maguer, N. Grabar, Health consumer-oriented information retrieval, Studies in Health Technology and Informatics 210 (2015) 80–84.
[64] P. Zweigenbaum, S.J. Darmoni, N. Grabar, The contribution of morphological knowledge to French MeSH mapping for information retrieval, in: Proceedings of the AMIA Symposium, 2001, pp. 796–800.
[65] É. Bigeard, F. Thiessard, N. Grabar, Detecting drug non-compliance in internet fora using information retrieval and machine learning approaches, Studies in Health Technology and Informatics 264 (2019) 30–34.
[66] V. Claveau, T. Hamon, S. Le Maguer, N. Grabar, Health consumer-oriented information retrieval, Studies in Health Technology and Informatics 210 (2015) 80–84.
[67] A. Périnet, T. Hamon, Distributional analysis applied to specialized texts. Reduction of data sparseness by context abstractions, TAL Traitement Automatique des Langues 56(2) (2015) 77–102.
[68] V. Trysnyuk, Y. Nagornyi, K. Smetanin, I. Humeniuk, T. Uvarova, A method for user authenticating to critical infrastructure objects based on voice message identification, Advanced Information Systems 4(3) (2020) 11–16. doi: 10.20998/2522-9052.2020.3.02.
[69] A. Medvedyk, M. Lohoida, Z. Rybchak, O. Kulyna, IT Slang: Development of Telegram Chatbot, CEUR Workshop Proceedings 3396 (2023) 152-162.
[70] O. Romanovskyi, et al., Elomia Chatbot: The Effectiveness of Artificial Intelligence in the Fight for Mental Health, CEUR Workshop Proceedings 2870 (2021) 1215-1224.
[71] A. Yarovyi, D. Kudriavtsev, Method of Multi-Purpose Text Analysis Based on a Combination of Knowledge Bases for Intelligent Chatbot, CEUR Workshop Proceedings 2870 (2021) 1238-1248.
[72] T. Basyuk, A. Vasyliuk, Peculiarities of an Information System Development for Studying Ukrainian Language and Carrying out an Emotional and Content Analysis, CEUR Workshop Proceedings 3396 (2023) 279-294.
[73] A. Dmytriv, S. Holoshchuk, L. Chyrun, R. Holoshchuk, Comparative Analysis of Using Different Parts of Speech in the Ukrainian Texts Based on Stylistic Approach, CEUR Workshop Proceedings 3171 (2022) 546-560.
[74] S. Yevseiev, et al., Development of a method for determining the indicators of manipulation based on morphological synthesis, Eastern-European Journal of Enterprise Technologies 117(9) (2022) 22-35.
[75] O. Cherednichenko, O. Kanishcheva, O. Yakovleva, D. Arkatov, Collection and Processing of a Medical Corpus in Ukrainian, CEUR Workshop Proceedings 2604 (2020) 272-282.
[76] A. Dmytriv, V. Vysotska, M. Bublyk, The Speech Parts Identification for Ukrainian Words Based on VESUM and Horokh Using, in: International Conference on Computer Sciences and Information Technologies, CSIT-2021, September 2021, Vol. 2, pp. 21-33.
[77] M. Lupei, et al., Analyzing Ukrainian Media Texts by Means of Support Vector Machines: Aspects of Language and Copyright, in: Computer Science, Engineering and Education Applications, 2023, March, pp. 173-182. Cham: Springer Nature Switzerland.
[78] The free dictionary by Farlex. Linguistic System. URL: https://encyclopedia2.thefreedictionary.com/Linguistic+System.
[79] Glottopedia. Linguistic information system. URL: http://www.glottopedia.org/index.php/Linguistic_information_system.
[80] Lenhart Schubert. Computational linguistics. Stanford Encyclopedia of Philosophy. URL: https://plato.stanford.edu/entries/computational-linguistics/.
[81] V. Vysotska, Analytical Method for Social Network User Profile Textual Content Monitoring Based on the Key Performance Indicators of the Web Page and Posts Analysis, CEUR Workshop Proceedings 3171 (2022) 1380-1402.
[82] B. Clifton, Advanced web metrics with Google Analytics. John Wiley & Sons. 2012.
[83] P. Zhezhnych, A. Shilinh, I. Demydov, Architecture of the Computer-linguistic System for Processing of Specialized Web-communities' Educational Content, CEUR Workshop Proceedings 2616 (2020) 1-11.

[84] V. Vysotska Ukrainian participles formation by the generative grammars use, CEUR Workshop Proceedings 2604 (2020) 407–427.
[85] P. Kravets, The Game Method for Orthonormal Systems Construction, in: Proceeding of the 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics, 2007. doi: 10.1109/cadsm.2007.4297555.
[86] M. Johnson, G. Lakoff, Why cognitive linguistics requires embodied realism, Cognitive Linguistics, 2002. doi: 10.1515/cogl.2002.016.
[87] M. Rehani, W. L. Wolf, Methods and systems for measuring semantics in communications. https://patentimages.storage.googleapis.com/00/d2/da/886c00fc2dce4b/US9269353.pdf.
[88] L. A. Kovbasyuk, I. O. Fritsky, V. N. Kokozay, T. S. Iskenderov, Synthesis and structure of diaqua-bis (ethylenediamine) copper (II) salts with anions of carbamic acids, Polyhedron 16(10) (1997) 1723-1729.
[89] L. A. Kovbasyuk, O. A. Babich, V. N. Kokozay, Direct synthesis and crystal structure of a mixed-valence copper complex,Polyhedron 16(1) (1997) 161-163.
[90] O. Oborska, M. Teliatynskyi, D. Dosyn, V. Lytvyn, S. Kostenko, An Intelligent System Based on Ontologies for Determining the Similarity of User Preferences, CEUR Workshop Proceedings 3403 (2023) 283-292.
[91] D. Dosyn, Y. I. Daradkeh, V. Kovalevych, M. Luchkevych, Y. Kis, Domain Ontology Learning using Link Grammar Parser and WordNet, CEUR Workshop Proceedings 3312 (2022) 14-36.
[92] Y. Burov, K., Mykich, I. Karpov, Intelligent systems based on ontology representation transformations, in: Conference on Computer Science and Information Technologies, 2020, September, pp. 263-275. Cham: Springer International Publishing.
[93] Y. Burov, Knowledge Based Situation Awareness Process Based on Ontologies, CEUR Workshop Proceedings 2870 (2021) 413-423.
[94] Y. Burov, K. Mykich, I. Karpov, Building a versatile knowledge-based system based on reasoning services and ontology representation transformations, in: IEEE 15th International Conference on Computer Sciences and Information Technologies, 2020, pp. 255-260.
[95] Yelpy Insights. URL: https://blog.yelp.com/news/yelpy-insights/.
[96] R. Anita, C. N. Subalalitha, An approach to cluster Tamil literatures using discourse connectives, in: IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP), 2019, pp. 1-4.
[97] O. Tverdokhlib, V. Vysotska, P. Pukach, M. Vovk, Information Technology for Identifying Hate Speech in Online Communication Based on Machine Learning, Data-Centric Business and Applications: Modern Trends in Financial and Innovation Data Processes 1 (2024) 339-369.
[98] D. Nakache, E. Metais, J. F. Timsit, Evaluation and NLP, in: International Conference on Database and Expert Systems Applications. Springer, Berlin, Heidelberg, 2005, pp. 626-632.
[99] M. Tikhonova, A. Gavrishchuk, NLP methods for automatic candidate's cv segmentation, in: IEEE International Conference on Engineering and Telecommunication, 2019. pp. 1-5.
[100] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, J. Li, Dice loss for data-imbalanced NLP tasks, arXiv preprint 2019. arXiv:1911.02855.
[101] Ryu Keun Ho, BioBERT Based Efficient Clustering Framework for Biomedical Document Analysis, in: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computing, October 21–23, 2021, Jilin, China. Springer Nature. p. 179.
[102] N. Rayzmann, H. Aponso, C. Y. Markgraf, P. E. Chappell, SUN-238 Estrogen Modulates Expression Levels of Gonadotropin-Releasing Hormone Receptor (GNRHR) in Immortalized Kisspeptin Neurons in Vitro, Journal of the Endocrine Society 4 (2020) SUN-238.
[103] Y. Tan, et al., Triaging ophthalmology outpatient referrals with machine learning: a pilot study, Clinical & experimental ophthalmology 48(2) (2020) 169-173.
[104] Kim Ju-Ri, Using Markedness Principle for Abstraction of Dependency Relations of Natural Languages, Eurasian Journal of Applied Linguistics 7.2 (2021) 58-67.
[105] D. Heo, W. Lee, B. Jung, J. H. Lee, Quality estimation using dual encoders with transfer learning, in: Proceedings of the Sixth Conference on Machine Translation, 2021, pp. 920-927.
[106] K. Ayre, et al., Developing a natural language processing tool to identify perinatal self-harm in electronic healthcare records, PloS one 16(8) (2021) e0253809.