КОМП'ЮТЕРНА ТА МАТЕМАТИЧНА ЛІНГВІСТИКА

УДК 004.9

О. В. Бісікало[1], В. А. Висоцька[2]
[1] Вінницький національний технічний університет,
кафедра автоматики та інформаційно-вимірювальної техніки,
[2] Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

МЕТОД ЛІНГВІСТИЧНОГО АНАЛІЗУ УКРАЇНОМОВНОГО КОМЕРЦІЙНОГО КОНТЕНТУ

Розв'язано науково-практичну задачу автоматичного виявлення значущих ключових слів та рубрикації україномовного контенту в інтернет-системах на основі методу лінгвістичного аналізу текстової інформації. Наведено теоретичне та експериментальне обґрунтування методу лінгвістичного аналізу україномовного контенту з використанням стемінгу Портера. Метод спрямовано на автоматичне виявлення значущих ключових слів україномовного контенту на основі запропонованої формалізації складових аналізу – граматичного (графемного), морфологічного, синтаксичного, семантичного, референційного та структурного.

Ключові слова: текст, україномовний, алгоритм, контент-моніторінг, ключові слова, контент-аналіз, стеммер Портера, лінгвістичний аналіз, синтаксичний аналіз.

The scientific and practical problem of automatic detection of meaningful keywords and Ukrainian content categorization in Internet systems on the basis of linguistic analysis of text information is unleashed. The article presents a theoretical and experimental substantiation of linguistic analysis methods for Ukrainian content using Porter stemming. The method is directed at the automatic identification of meaningful keywords in the Ukrainian content, based on the proposed analysis components formalization – the grammatical (grapheme), morphological, syntactic, semantic, structural and referential.

Key words: text, a Ukrainian, algorithm, content monitoring, keywords, content analysis, Porter stemmer, linguistic analysis, parsing.

Вступ. Загальна постановка проблеми

У практичному плані аналіз знакового рівня організації природномовного тексту обмежується виокремленням синтаксичних розділових знаків від слова, виділенням абревіатур, скорочень тощо. Аналіз реальних текстів показує, що вже на рівні знакової організації тексту людина використовує описові можливості семіотичної системи для кодування знань про фрагменти реальної дійсності. Так, використання лапок (наприклад, кінотеатр "Зірка") свідчить, що лексему в лапках не можна розглядати в значенні, поданому в словнику. Власні назви, наведені у тексті, можуть збігатися з написанням загальновживаних слів, але при цьому мати різний зміст (наприклад, група Чорний вересень, асистент Вовк О. Б., викладач Заяць М. М., співачка Катя Чилі, співачка Вінницька Альона, актор Девід Духовний як Fox William Mulder, проспект Свободи, вулиця 1 Травня, акторка Сарі Габріела, акторка Настя Задорожна тощо). Крім того, деякі лексеми в тексті не підпорядковані граматичним правилам мови, а є одиницями знакового рівня (наприклад, число 30, відсоткове значення 15 %, скорочення млн, тис. або кг тощо). Ці особливості природномовного тексту і зумовлюють необхідність розроблення знакового рівня організації тексту як початкового етапу побудови моделі розуміння тексту.

Лінгвістичний метод опрацювання текстової інформації для автоматичного виявлення значущих ключових слів складається з шести етапів:

1. Граматичний (графемний) аналіз текстового контенту, тобто парсинг тексту з врахуванням особливостей графем різних мов.
2. Морфологічний аналіз текстового контенту.
3. Синтаксичний аналіз текстового контенту.
4. Семантичний аналіз текстового контенту.
5. Референційний аналіз для формування міжфразових єдностей.
6. Структурний аналіз текстового контенту.

Вхідними даними етапу графемного (доморфемного) аналізу або парсингу є поточний текстовий файл та апріорні еталонні моделі (рядки і знаки). Виокремленням таких одиниць тексту, як наймення, позначення, назви тощо можна вже на цьому етапі визначити деякі функціональні елементи структури понять. Отже, розв'язання актуальної проблеми формування ефективних процедур розпізнавання знань з предметної області доцільно розпочинати з аналізу тексту із знакового рівня. Інформаційним компонентом при цьому є електронні словники скорочень, географічних назв, імен. Такий підхід зумовлений різноманітністю знакового (графемного) подання лексичних одиниць у тексті, яка визначає їхні різні семантичні функції у тому чи іншому контексті. Для автоматизованого опрацювання природномовної інформації суттєвим є також визначення структури тексту – для виокремлення службової інформації, виділення абзаців, заголовків тощо. Текст при цьому розглядається як певним чином організована послідовність рядків і графем.

Зв'язок висвітленої проблеми із важливими науковими та практичними завданнями

У статті розв'язано науково-практичну задачу автоматичного виявлення значущих ключових слів та рубрикації україномовного контенту в інтернет-системах на основі методу лінгвістичного аналізу текстової інформації. Роботу виконано в межах спільних наукових досліджень кафедри інформаційних систем та мереж Національного університету "Львівська політехніка" на тему "Дослідження, розроблення і впровадження інтелектуальних розподілених інформаційних технологій та систем на основі ресурсів баз даних, сховищ даних, просторів даних та знань з метою прискорення процесів формування сучасного інформаційного суспільства", а також кафедри автоматики та інформаційно-вимірювальної техніки Вінницького національного технічного університету у межах діяльності науково-дослідного центру прикладної та комп'ютерної лінгвістики. Результати досліджень здійснювали у межах держбюджетних науково-дослідних робіт за темами "Розробка методів, алгоритмів і програмних засобів моделювання, проектування та оптимізації інтелектуальних інформаційних систем на основі Web-технологій "ВЕБ" та "Інтелектуальна інформаційна технологія образного аналізу тексту та синтезу інтегрованої бази знань природномовного контенту". Наукові дослідження здійснювали також у межах ініціативної тематики досліджень кафедри ІСМ Національного університету "Львівська політехніка" на тему "Розроблення інтелектуальних розподілених систем на основі онтологічного підходу з метою інтеграції інформаційних ресурсів".

Аналіз останніх досліджень та публікацій

Текстовий контент (стаття, коментар, книга тощо) містить значний обсяг даних природною мовою, частина яких є абстрактною [1–7, 31–32]. Текст подають як об'єднану за змістом послідовність знакових одиниць, основними властивостями якої є інформаційна, структурна та комунікативна зв'язність/цілісність, що відображає змістовну/структурну сутність тексту [8–22, 33–47]. Методом опрацювання тексту є лінгвістичний аналіз змісту (наприклад, коментарі, форуми, статті тощо) [23–30, 48–56]. Процес опрацювання тексту поділяє контент на лексеми за допомогою кінцевих автоматів (рис. 1).

Аналіз отриманих наукових результатів

Як функціонально-семантико-структурна єдність текст відповідає правилам побудови, виявляє закономірності змістовного та формального з'єднання складових . Зв'язність проявляється

через зовнішні структурні показники та формальну залежність компонентів тексту, а цілісність – через тематичну, концептуальну та модальну залежність. Цілісність веде до змістовної та комунікативної організації тексту, а зв'язність – до форми, структурної організації. Тому пропонується під час аналізу досліджувати багаторівневу структуру контенту: лінійну послідовність символів; лінійну послідовність морфологічних структур; лінійну послідовність речень; мережу взаємопов'язаних єдностей (алг. 1).

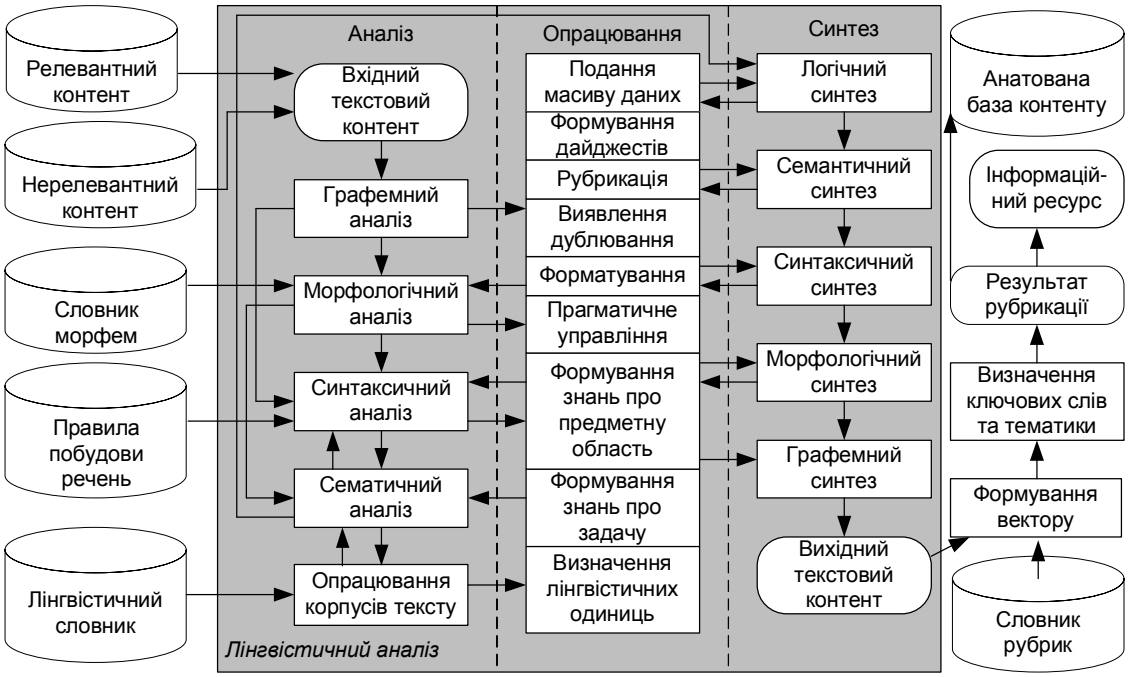


Рис. 1. Структурна схема лінгвістичного аналізу текстового контенту

Алгоритм 1. Лінгвістичний аналіз текстового контенту.

Етап 1. Граматичний (графемний) аналіз текстового контенту C_2.

Крок 1. Поділ текстового комерційного контенту C_2 на речення та абзаци.

Крок 2. Поділ ланцюжка символів контенту C_2 на слова.

Крок 3. Виділення цифр, чисел, дат, незмінних оборотів і скорочень контенту C_2.

Крок 4. Видалення нетекстових символів контенту C_2.

Крок 5. Формування та аналіз лінійної послідовності слів із службовими знаками для контенту C_2.

Етап 2. Морфологічний аналіз текстового контенту C_2.

Крок 1. Отримання основ (словоформ із відрубаними закінченнями).

Крок 2. Для кожної словоформи формується граматична категорія (колекція граматичних значень: рід, відмінок, відмінювання тощо).

Крок 3. Формування лінійної послідовності морфологічних структур.

Етап 3. Синтаксичний аналіз $\alpha_4 : (C_2, U_K, T) \to C_3$ текстового контенту C_2.

Етап 4. Семантичний аналіз текстового контенту C_3.

Крок 1. Слова співвідносяться з семантичними класами із словника.

Крок 2. Відбір потрібних для даного речення морфосемантичних альтернатив.

Крок 3. Зв'язування слів у єдину структуру.

Крок 4. Формування упорядкованої множини записів суперпозицій з базисних лексичних функцій і семантичних класів. Точність результату визначається повнотою/коректністю словника.

Етап 5. Референційний аналіз для формування міжфразових єдностей.

Крок 1. Контекстний аналіз текстового комерційного контенту C_3. За його допомогою реалізується дозвіл локальних референцій (цей, який, його) і виділення висловлювання – ядра єдності.

Крок 2. Тематичний аналіз. Поділ висловлювань на тему і рему виділяє тематичні структури, які використовують, наприклад, при формуванні дайджесту.

Крок 3. Визначають регулярну повторюваність, синонімізацію та повторну номінацію ключових слів; тотожність референції, тобто співвідношення слів з предметом зображення; наявність імплікації, заснованої на ситуативних зв'язках.

Етап 6. Структурний аналіз текстового контенту C_3. Передумовами використання є високий ступінь збігу термінів єдності, дискурсивна одиниця, речення семантичною мовою, висловлювання і елементарна дискурсивна одиниця.

Крок 1. Виявлення базового набору риторичних зв'язків між єдностями контенту.

Крок 2. Побудова нелінійної мережі єдностей. Відкритість набору зв'язків припускає його розширення та адаптацію для аналізу структури текстів C_3.

Детально розглянемо кожний етап запропонованого алгоритму

Етап 1. Граматичний (графемний) аналіз текстового контенту. Графемою називають мінімальну змістовну одиницю писемного тексту. Задачею цього рівня розпізнавання є побудова формалізованого подання графемної структури тексту та розроблення формального апарату виокремлення і класифікації текстових одиниць на множині рядків і графем. Узагальнений алгоритм розпізнавання працює при певних обмеженнях на вхідний текст: відформатований по ширині; не містить переносів; не містить об'єктів як таблиця, рисунок, формула або графічний символ; поданий відомими мовами, наприклад, англійською, українською та німецькою, а не давньоєгипетською, монгольською або ельфійською. Кінцевою метою розпізнавання графемного рівня подання тексту є побудова графемної структури тексту, яка включає виокремлення на множині рядків і графем вхідного тексту таких семантично самостійних одиниць тексту, як фрагменти (дискурси), речень, синтагм, лексем та визначення типів (класів) перерахованих одиниць тексту та встановлення відношень між ними в певному вхідному тексті. Процес розпізнавання на графемному рівні подання тексту передбачає два етапи (рис. 2).

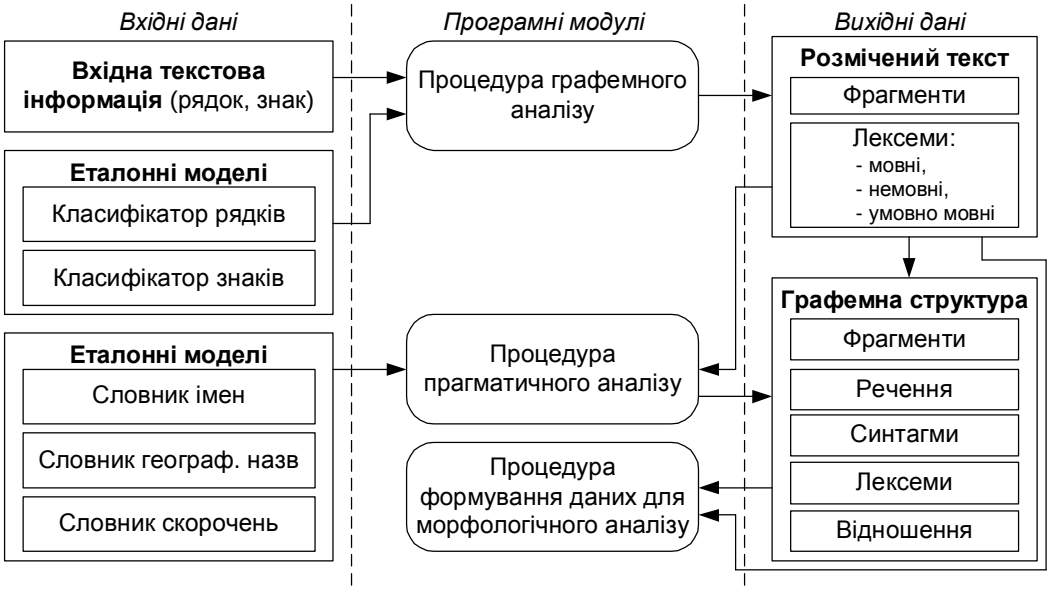


Рис. 2. Структурно-логістична схема розпізнавання знань з предметної області на графемному етапі аналізу текстової інформації

Задачею першого етапу є виокремлення змістовно самостійних фрагментів у тексті, лексем у кожному фрагменті тексту та визначення мови вхідного тексту і/або фрагментів тексту. Вхідними

даними першого етапу є поточний текстовий файл і апріорні еталонні моделі рядків і графем. Класифікатор рядків включає такі вагомі класи: пустий рядок (empty string, *EmpStr*), повний рядок (full string, *FulStr*), неповний праворуч (incomplete right, *IncRgt*), неповний ліворуч (incomplete left, *IncLgt*), симетрично неповний (symmetric incomplete, *SmtInc*). Правила розпізнавання рядків у тексті наведено в табл. 1.

Таблиця 1

Правила розпізнавання рядків в тексті на графемному етапі

№	Назва рядка	Перша позиція	Остання позиція	Всі позиції
1	Пустий рядок (empty string, EmpStr)	–	–	Пробіл
2	Повний рядок (full string, *FulStr*)	Символ	Символ	–
3	Неповний праворуч (incomplete right, *IncRgt*)	Символ	Пробіл	–
4	Неповний ліворуч (incomplete left, *IncLgt*)	Пробіл	Символ	–
5	Симетрично неповний (symmetric incomplete, *SmtInc*)	Пробіл	Пробіл	–

Множину еталонних моделей графем зручно подати нормальною формою Бекуса–Наура (БНФ), скорочені позначення елементів яких подано в табл. 2.

Таблиця 2

Позначення елементів множини еталонних моделей графем

№	Назва	Англійський переклад	Абревіатура
1	2	3	4
1	Граматика	Grammar	G
2	Алфавіт	Alphabet	*V*
3	Термінальні символи	Term	*T*
4	Початковий символ	Initial character	*S*
5	Продукційні правила	Production rules	*P*
6	Символ	Symbol	*Sb*
7	Пробіл	Space	*Sp*
8	Цифра	Digit	*Dgt*
9	Спеціальний символ	Special symbol	*Ssb*
10	Синтаксичний знак	Syntactic sign	*Ssg*
11	Літера	Letter	*Ltr*
12	Латинські літери	Latin letter	*Lat*
13	Літери кирилиці	Cyrillic letter	*Cyr*
14	Англійський алфавіт	English alphabet	*Eng*
15	Німецький алфавіт	German alphabet	*Ger*
16	Польський алфавіт	Polish alphabet	*Pol*
17	Український алфавіт	Ukrainian alphabet	*Ukr*
18	Російський алфавіт	Russian alphabet	*Rus*
19	Службовий символ	Official symbol	*Osb*
20	Дужки	Brackets	*Bsb*
21	Математичний символ	Mathematical symbol	*Msb*
22	Заголовна літера	Capital letter	*Cpl*
23	Мала літера	Small letter	*Sml*
24	Латинська заголовна літера	Latin capital letter	*Lcp*
25	Латинська мала літера	Latin small letter	*Lsm*
26	Заголовна літера кирилиці	Cyrillic capital letter	*Ccp*
27	Мала літера кирилиці	Cyrillic small letter	*Csm*
28	Англійська заголовна літера	English capital letter	*Ecp*
29	Англійська мала літера	English small letter	*Esm*
30	Німецька заголовна літера	German capital letter	*Gcp*

Продовження табл. 2

1	2	3	4
31	Німецька мала літера	German small letter	*Gsm*
32	Польська заголовна літера	Polish letter	*Pcp*
33	Польська мала літера	Polish small letter	*Psm*
34	Українська заголовна літера	Ukrainian capital letter	*Ucp*
35	Українська мала літера	Ukrainian small letter	*Usm*
36	Російська заголовна літера	Russian capital letter	*Rcp*
37	Російська мала літера	Russian small letter	*Rsm*
38	Приголосна літера	Consonant letter	*Cnl*
39	Голосна літера	Vowel letter	*Vwl*
40	Латинська заголовна приголосна літера	Latin capital consonant letter	*Lcc*
41	Латинська мала приголосна літера	Latin small consonant letter	*Lsc*
42	Латинська заголовна голосна літера	Latin capital vowel letter	*Lcv*
43	Латинська мала голосна літера	Latin small vowel letter	*Lsv*
44	Заголовна приголосна літера кирилиці	Cyrillic capital consonant letter	*Ccc*
45	Мала приголосна літера кирилиці	Cyrillic small consonant letter	*Csc*
46	Заголовна голосна літера кирилиці	Cyrillic capital vowel letter	*Ccv*
47	Мала голосна літера кирилиці	Cyrillic small vowel letter	*Csv*
48	Англійська заголовна приголосна літера	English capital consonant letter	*Ecc*
49	Англійська мала приголосна літера	English small consonant letter	*Esc*
50	Англійська заголовна голосна літера	English capital vowel letter	*Ecv*
51	Англійська мала голосна літера	English small vowel letter	*Esv*
52	Німецька заголовна приголосна літера	German capital consonant letter	*Gcc*
53	Німецька мала приголосна літера	German small consonant letter	*Gsc*
54	Німецька заголовна голосна літера	German capital vowel letter	*Gcv*
55	Німецька мала голосна літера	German small vowel letter	*Gsv*
56	Польська заголовна приголосна літера	Polish capital consonant letter	*Pcc*
57	Польська мала приголосна літера	Polish small consonant letter	*Psc*
58	Польська заголовна голосна літера	Polish capital vowel letter	*Pcv*
59	Польська мала голосна літера	Polish small vowel letter	*Psv*
60	Українська заголовна приголосна літера	Ukrainian capital consonant letter	*Ucc*
61	Українська мала приголосна літера	Ukrainian small consonant letter	*Usc*
62	Українська заголовна голосна літера	Ukrainian capital vowel letter	*Ucv*
63	Українська мала голосна літера	Ukrainian small vowel letter	*Usv*
64	Російська заголовна приголосна літера	Russian capital consonant letter	*Rcc*
65	Російська мала приголосна літера	Russian small consonant letter	*Rsc*
66	Російська заголовна голосна літера	Russian capital vowel letter	*Rcv*
67	Російська мала голосна літера	Russian small vowel letter	*Rsv*

Розглянемо граматику $G = < V, T, S, P >$, де алфавіт $V = < Gr, T >$; термінальні символи $T : = <$ A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, Q, P, R, S, T, U, W, V, X, Y, Z, Ä, Ö, Ü, Ą, Ć, Ę, Ł, Ń, Ó, Ś, Ź, Ż, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, w, v, x, y, z, ä, ö, ü, ß, ą, ć, ę, ł, ń, ó, ś, ź, ż, А, Б, В, Г, Д, Е, Ж, З, И, Й, К, Л, М, Н, О, П, Р, С, Т, У, Ф, Х, Ц, Ч, Ш, Щ, Ь, Ю, Я, Є, І, Ї, Ґ, Ы, Э, Ъ, а, б, в, г, д, е, ж, з, и, й, к, л, м, о, н, п, р, с, т, у, ф, х, ц, ч, ш, щ, ь, ю, я, є, і, ї, ґ, ы, э, ъ, ‘> та список кортежів для визначених в табл. 2 елементів множини еталонних моделей графем:

1) $Gr : = < Sb \cup Sp >$ – розпізнаний текстовий контент як набір символів та пробілів;

2) $Sp : = <\_>$ – пробіл як термінальний символ;

3) $Sb : = < Ltr \cup Dgt \cup Ssb \cup Ssg >$ – множини літер, цифр, спецсимволів та синтаксичних знаків;

4) $Ltr : = < Lat \cup Cyr \cup Eng \cup Ger \cup Pol \cup Ukr \cup Rus \cup Cpl \cup Sml \cup Cnl \cup Vwl \cup ‘>$ – набір латинських, кириличних, англійських, німецьких, польських, українських, російських літер,

зокрема заголовних і малих літер відповідних мов, приголосних та голосних, а також апострофа як термінального символу;

5) *Dgt* : = < 0 ∪ 1 ∪ 2 ∪ 3 ∪ 4 ∪ 5 ∪ 6 ∪ 7 ∪ 8 ∪ 9 > – множина цифр;

6) *Ssb* : = < *Osb* ∪ *Bsb* ∪ *Msb* > – набір службових символів, дужок та математичних символів;

7) *Ssg* : = < “ ∪ ” ∪ “ ∪ ”∪, ∪ . ∪ : ∪; ∪ – ∪ ? ∪ ! > – термінальні символи синтаксичних знаків;

8) *Cpl* : = < *Lcp* ∪ *Ccp* ∪ *Ecp* ∪ *Gcp* ∪ *Pcp* ∪ *Ucp* ∪ *Rcp* > – набір заголовних літер відповідних мов;

9) *Sml* : = < *Lsm* ∪ *Csm* ∪ *Esm* ∪ *Gsm*∪ *Psm* ∪ *Usm*∪ *Rsm* > – набір малих літер відповідних мов;

10) *Lat* : = < *Lcp* ∪ *Lsm* > – множина латинських літер – як заголовних, так і малих;

11) *Cyr* : = < *Ccp* ∪ *Csm* > – множина кириличних літер – як заголовних, так і малих;

12) *Eng* : = < *Ecp* ∪ *Esm* > – множина англійських літер – як заголовних, так і малих;

13) *Ger* : = < *Gcp* ∪ *Gsm* > – множина німецьких літер – як заголовних, так і малих;

14) *Pol* : = < *Pcp* ∪ *Psm* > – множина польських літер – як заголовних, так і малих;

15) *Ukr* : = < *Ucp* ∪ *Usm* > – множина українських літер – як заголовних, так і малих;

16) *Rus* : = < *Rcp* ∪ *Rsm* > – множина російських літер – як заголовних, так і малих;

17) *Osb* : = < № ∪ % ∪ / ∪ @ ∪ # ∪ $ ∪ & ∪ * ∪ \ > – набір термінальних службових символів;

18) *Bsb* : =< [∪] ∪ { ∪ } ∪ (∪) > – набір термінальних символів дужок;

19) *Msb* : = < + ∪ < ∪ > ∪ = > – набір термінальних математичних символів;

20) *Cnl* : = < *Ecc* ∪ *Esc* ∪ *Gcc* ∪ *Gsc* ∪ *Pcc* ∪ *Psc* ∪ *Ucc* ∪ *Usc* ∪ *Rcc* ∪ *Rsc* > – набір приголосних літер (великих та малих) відповідних мов;

21) *Vwl* : = < *Ecv* ∪ *Esv* ∪ *Gcv* ∪ *Gsv* ∪ *Pcv* ∪ *Psv* ∪ *Ucv* ∪ *Usv* ∪ *Rcv* ∪ *Rsv* > – набір голосних літер (великих та малих) відповідних мов;

22) *Lcp* : = < *Lcc* ∪ *Lcv* ∪ Q ∪ V ∪ X > – латинські заголовні літери;

23) *Lsm* : = < *Lsc* ∪ *Lsv* ∪ q ∪ v ∪ x > – латинські малі літери;

24) *Ccp* :. = < *Ccc* ∪ *Csv* ∪ Ь ∪ Й > – кириличні заголовні літери;

25) *Csm* : = < *Csc* ∪ *Csv* ∪ ь ∪ й > – кириличні малі літери;

26) *Ecp* : = < *Lcc* ∪ *Lcv* ∪ Q ∪ V ∪ X > – англійські заголовні літери;

27) *Esm* : = < *Lsc* ∪ *Lsv* ∪ q ∪ v ∪ x > – англійські малі літери;

28) *Gcp* : = < *Lcc* ∪ *Lcv* ∪ Ä ∪ Ö ∪ Ü ∪ Q ∪ V ∪ X > – німецькі заголовні літери;

29) *Gsm* : = < *Lsc* ∪ *Lsv* ∪ ä ∪ ö ∪ ü ∪ ß ∪ q ∪ v ∪ x > – німецькі малі літери;

30) *Pcp* : = < *Lcc* ∪ *Lcv* ∪ Ą ∪ Ć ∪ Ę ∪ Ł ∪ Ń ∪ Ó ∪ Ś ∪ Ź ∪ Ż > – польські заголовні літери;

31) *Psm* : = < *Lsc* ∪ *Lsv* ∪ ą ∪ ć ∪ ę ∪ ł ∪ ń ∪ ó ∪ ś ∪ ź ∪ ż > – польські малі літери;

32) *Ucp* : = < *Ccc* ∪ *Ccv* ∪ Є ∪ I ∪ Ї ∪ Ґ > – українські заголовні літери;

33) *Usm* : = < *Csc* ∪ *Csv* ∪ є ∪ і ∪ ї ∪ ґ> – українські малі літери;

34) *Rcp* : = < *Ccc* ∪ *Ccv* ∪ Ы ∪ Э ∪ Ъ > – російські заголовні літери;

35) *Rsm* : = < *Csc* ∪ *Csv* ∪ ы ∪ э ∪ ъ> – російські малі літери;

36) *Lcc* : = < B ∪ C ∪ D ∪ F ∪ G ∪ H ∪ J ∪ K ∪ L ∪ M ∪ N ∪ P ∪ R ∪ S ∪ T ∪ W ∪ Z> – термінальні латинські заголовні приголосні літери;

37) *Lcv* : = < A ∪ E ∪ I ∪ O ∪ U ∪ Y> – термінальні латинські заголовні голосні літери;

38) *Lsc* : = < b ∪ c ∪ d ∪ f ∪ g ∪ h ∪ j ∪ k ∪ l ∪ m ∪ n ∪ p ∪ r ∪ s ∪ t ∪ w ∪ x ∪ z > – термінальні латинські малі приголосні літери;

39) *Lsv*: = < a ∪ e ∪ i ∪ o ∪ u ∪ v ∪ y > – термінальні латинські малі голосні літери;

40) *Ccc* : = < Б ∪ В ∪ Г ∪ Д ∪ Ж ∪ З ∪ К ∪ Л ∪ М ∪ Н ∪ П ∪ Р ∪ С ∪ Т ∪ Ф ∪ Х ∪ Ц ∪ Ч ∪ Ш ∪ Щ > – термінальні кириличні заголовні приголосні літери;

41) *Csv* : = < А ∪ Е ∪ И ∪ О ∪ У ∪ Ю ∪ Я > – термінальні кириличні заголовні голосні літери;

42) *Csc* := < б ∪ в ∪ г ∪ д ∪ ж ∪ з ∪ к ∪ л ∪ м ∪ н ∪ п ∪ р ∪ с ∪ т ∪ ф ∪ х ∪ ц ∪ ч ∪ ш ∪ щ > – термінальні кириличні малі приголосні літери;

43) *Csv* : = < а ∪ е ∪ и ∪ о ∪ у ∪ ю ∪ я > – термінальні кириличні малі голосні літери.

Для розпізнавання мови тексту пропонуються такі продукційні правила:

P : = < S → S *Gr*, S → Λ, *Gr* → *Gr Sb*, *Gr* → *Gr Sp*, *Gr* → Λ, *Sp* → _, *Sb* → *Ltr*, *Sb* → *Dgt*, *Sb* → *Ssb*, *Sb* → *Ssg*, *Ltr* → *Lat*, *Ltr* → *Cyr*, *Ltr* → *Eng*, *Ltr* → *Ger*, *Ltr* → *Pol*, *Ltr* → *Ukr*, *Ltr* → *Rus*, *Ltr* → *Cpl*, *Ltr* → *Sml*, *Ltr* → *Cnl*, *Ltr* → *Vwl*, *Ltr* → ‘, *Ssb* → *Osb*, *Ssb* → *Bsb*, *Ssb* → *Msb*, *Cpl* → *Lcp*, *Cpl* → *Ccp*, *Cpl* → *Ecp*, *Cpl* → *Gcp*, *Cpl* → *Pcp*, *Cpl* → *Ucp*, *Cpl* → *Rcp*, *Sml* → *Lsm*, *Sml* → *Csm*, *Sml* → *Esm*, *Sml* → *Gsm*, *Sml* → *Psm*, *Sml* → *Usm*, *Sml* → *Rsm*, *Lat* → *Lcp*, *Lat* → *Lsm*, *Cyr* → *Ccp*, *Cyr* → *Csm*, *Eng* → *Ecp*, *Eng* → *Esm*, *Ger* → *Gcp*, *Ger* → *Gsm*, *Pol* → *Pcp*, *Pol* → *Psm*, *Ukr* → *Ucp*, *Ukr* → *Usm*, *Rus* → *Rcp*, *Rus* → *Rsm*, *Lcp* → *Lcc*, *Lcp* → *Lcv*, *Lcp* → Q, *Lcp* → V, *Lcp* → X, *Lsm* → *Lsc*, *Lsm* → *Lsv*, *Lsm* → q, *Lsm* → v, *Lsm* → x, *Ccp* → *Ccc*, *Ccp* → *Csv*, *Ccp* → Ь, *Ccp* → Й, *Csm* → *Csc*, *Csm* → *Csv*, *Csm* → ь, *Csm* → й, *Ecp* → *Lcc*, *Ecp* → *Lcv*, *Ecp* → Q, *Ecp* → V, *Ecp* → X, *Esm* → *Lsc*, *Esm* → *Lsv*, *Esm* → q, *Esm* → v, *Esm* → x, *Gcp* → *Lcc*, *Gcp* → *Lcv*, *Gcp* → Ä, *Gcp* → Ö, *Gcp* → Ü, *Gcp* → Q, *Gcp* → V, *Gcp* → X, *Gsm* → *Lsc*, *Gsm* → *Lsv*, *Gsm* → ä, *Gsm* → ö, *Gsm* → ü, *Gsm* → ß, *Gsm* → q, *Gsm* → v, *Gsm* → x, *Pcp* → *Lcc*, *Pcp* → *Lcv*, *Pcp* → Ą, *Pcp* → Ć, *Pcp* → Ę, *Pcp* → Ł, *Pcp* → Ń, *Pcp* → Ó, *Pcp* → Ś, *Pcp* → Ź, *Pcp* → Ż, *Psm* → *Lsc*, *Psm* → *Lsv*, *Psm* → ą, *Psm* → ć, *Psm* → ę, *Psm* → ł, *Psm* → ń, *Psm* → ó, *Psm* → ś, *Psm* → ź, *Psm* → ż, *Ucp* → *Ccc*, *Ucp* → *Ccv*, *Ucp* → Є, *Ucp* → I, *Ucp* → Ї, *Ucp* → Ґ, *Usm* → *Csc*, *Usm* → *Csv*, *Usm* → є, *Usm* → i, *Usm* → ї, *Usm* → ґ, *Rcp* → *Ccc*, *Rcp* → *Ccv*, *Rcp* → Ы, *Rcp* → Э, *Rcp* → Ъ, *Rsm* → *Csc*, *Rsm* → *Csv*, *Rsm* → ы, *Rsm* → э, *Rsm* → ъ, *Lcc* → B, *Lcc* → C, *Lcc* → D, *Lcc* → F, *Lcc* → G, *Lcc* → H, *Lcc* → J, *Lcc* → K, *Lcc* → L, *Lcc* → M, *Lcc* → N, *Lcc* → P, *Lcc* → R, *Lcc* → S, *Lcc* → T, *Lcc* → W, *Lcc* → Z, *Lcv* → A, *Lcv* → E, *Lcv* → I, *Lcv* → O, *Lcv* → U, *Lcv* → Y, *Lsc* → b, *Lsc* → c, *Lsc* → d, *Lsc* → f, *Lsc* → g, *Lsc* → h, *Lsc* → j, *Lsc* → k, *Lsc* → l, *Lsc* → m, *Lsc* → n, *Lsc* → p, *Lsc* → q, *Lsc* → r, *Lsc* → s, *Lsc* → t, *Lsc* → w, *Lsc* → x, *Lsc* → z, *Lsv* → a, *Lsv* → e, *Lsv* → i, *Lsv* → o, *Lsv* → u, *Lsv* → v, *Lsv* → y, *Ccc* → Б, *Ccc* → В, *Ccc* → Г, *Ccc* → Д, *Ccc* → Ж, *Ccc* → З, *Ccc* → К, *Ccc* → Л, *Ccc* → М, *Ccc* → Н, *Ccc* → П, *Ccc* → Р, *Ccc* → С, *Ccc* → Т, *Ccc* → Ф, *Ccc* → Х, *Ccc* → Ц, *Ccc* → Ч, *Ccc* → Ш, *Ccc* → Щ, *Csv* → А, *Csv* → Е, *Csv* → И, *Csv* → О, *Csv* → У, *Csv* → Ю, *Csv* → Я, *Csc* → б, *Csc* → в, *Csc* → г, *Csc* → д, *Csc* → ж, *Csc* → з, *Csc* → к, *Csc* → л, *Csc* → м, *Csc* → н, *Csc* → п, *Csc* → р, *Csc* → с, *Csc* → т, *Csc* → ф, *Csc* → х, *Csc* → ц, *Csc* → ч, *Csc* → ш, *Csc* → щ, *Csv* → а, *Csv* → е, *Csv* → и, *Csv* → о, *Csv* → у, *Csv* → ю, *Csv* → я >.

Ці продукційні правила використовують для виявлення змістовних одиниць аналізу $< U'_C, U'_G >$ текстового комерційного контенту X' (словосполучення, речення, тема, ідея, автор, персонаж, соціальна ситуація, частина тексту, кластеризована за змістом категорії аналізу) (ЕТАП 1 парсинг з врахуванням мови фрагментів текстів) за модифікованим алгоритмом Потера (ЕТАП 2 стемінг). Маємо такі вимоги до вибору лінгвістичної одиниці аналізу: велика для інтерпретації значення; мала, щоб не інтерпретувати багато значень; легко ідентифікується; кількість одиниць велика для виокремлення вибірки.

Етап 2. Морфологічний аналіз текстового контенту полягає у знаходженні основ слів, наприклад, [8] вирізає суфікси, префікси тощо, лишаючи тільки основу слова (стемінг)**.** Існують відомі алгоритми знаходження основ, наприклад, [8] вирізає суфікси, префікси тощо, лишаючи тільки основу слова. Також вирізаються ключовики простою функцією по вибору слів, далі в кожному слові визначають основу і записують в таблицю, наприклад: keywords. Проте маємо недолік – треба врахувати всі правила утворення слів в українській мові (флексії залежно від роду та відмінювання, частини мови, суфікси, префікси, чергування слів в основі при відмінюванні, однина і множина тощо). Наприклад, з такими словами з множини *M* = {пошук**ов**ими, користувач**ам, високо**рейтинг**ово**го, рейтинг**у}** такі алгоритми не працюють (синім кольором позначено

причину, чому не працює – не врахували в правилах). Збільшення правил у геометричній прогресії збільшує навантаження на процеси опрацювання, наприклад, задача перевірки та визначення ключовиків для 100 статей на день вимагає перевірити кожне слово через аналізатор закінчень, суфіксів тощо – складність алгоритму зростає до критичної межі. Для англомовних текстів складність менша – там лише два відмінки та одне закінчення для іменників. Вже для німецької мови складність зростає – 4 відмінки, складені слова разом пишуться з 2, 3 та більше слів тощо. У [8] алгоритм працює для L = {Автомат – Автомат, Автомата – Автомат, Автоматом – Автомат, Ресурсів – ресурс}. Але краще не корінь знаходити відсіканням зайвого, а маючи тематичні словники основ ключовиків знаходити саме в тексті ці основи слова, їх розподіл (більше на початку, чи в кінці, чи в середині тексту) та частоту вживання відносно загального обсягу. І через основу робити статистику, підраховувати кількість однакових основ. Для англомовних текстів є відомий алгоритм – стеммер Портера [9], але для українських текстів він не зовсім коректно працює.

Стемер Портера – алгоритм Стемінг, який опублікував Мартін Портер у 1980 році. Оригінальна версія стемера була призначена для англійської мови і була написана мовою BCPL. Згодом Мартін створив проект “Snowball” і, використовуючи основну ідею алгоритму, написав стемер для поширених індоєвропейських мов, зокрема для російської [10–17]. Алгоритм не використовує баз основ слів, а лише, застосовуючи послідовно ряд правил, відсікає закінчення і суфікси, ґрунтуючись на особливостях мови, у зв’язку з чим працює швидко, але не завжди безпомилково. Алгоритм був дуже популярним і тиражованим, до нього часто вносили зміни різні розробники, причому не завжди вдалі. Приблизно 2000 року Портер прийняв рішення “заморозити” проект і надалі поширювати єдину реалізацію алгоритму (декількома популярними мовами програмування) зі свого сайту [10–17]. Наприклад, цей алгоритм враховує в україномовних текстах наявність лише закінчення, а суфікси – ні. Тоді слова *пошук*, *пошуку* ідентифікує, а *пошукові* – ні. За формою флексій визначають тип слова, наприклад,

```
var $ADJECTIVE =
'/(ими|ій|ий|а|е|ова|ове|ів|є|їй|єє|еє|я|ім|ем|им|ім|их|іх|ою|ими|іми|у|ю|ого|ому|оїй)$/
'; //http://uk.wikipedia.org/wiki/Прикметник + http://wapedia.mobi/uk/Прикметник
 var $PARTICIPLE = '/(ий|ого|ому|им|ім|а|ій|у|ою|ій|і|их|ими|их)$/';
//http://uk.wikipedia.org/wiki/Дієприкметник
 var $VERB = '/(сь|ся|ив|ать|ять|у|ю|ав|али|учи|ячи|вши|ши|е|ме|ати|яти|є)$/';
//http://uk.wikipedia.org/wiki/Дієслово
 var $NOUN =
'/(а|ев|ов|е|ями|ами|еи|и|ей|ой|ий|й|иям|ям|ием|ем|ам|ом|о|у|ах|иях|ях|ы|ь|ию|ью|ю|ия|ь
я|я|і|ові|ї|ею|єю|ою|є|еві|ем|єм|ів|їв|\'ю)$/'; //http://uk.wikipedia.org/wiki/Іменник
```

Особливості алгоритму. Алгоритм працює з окремими словами, тому контекст, у якому вжито слово, невідомий. Також недоступні такі категорії мовознавства як будова слова (корінь, суфікс тощо) та частини мови (іменник, прикметник тощо). Наразі маємо такі прийоми аналізу слів:

- Від слова прибирають закінчення, наприклад, прибирання закінчення *увати* перетворює слово *критикувати* на *критик*.
- Слово має незмінне закінчення. Слова з цим закінченням залишаються без змін. Приклад – *ск* і незмінні слова *блиск*, *тиск*, *обеліск* тощо.
- Слово змінює закінчення. Це правило стосується слів, у яких при відмінюванні випадають певні літери (*ядро* та *ядер* – закінчення *ер* змінюється на *р*) чи змінюються (*чоловік* та *чоловіче* – *к* змінюється на *ч*).
- Слово відповідає регулярному виразу. Це спроба об’єднати декілька правил в одне складне. Можливо, цей прийом не доживе до фінальної версії алгоритму. Але зараз у коді зустрічаються вирази, схожі на:

(ов)*ува(в|вши|вшись|ла|ло|ли|ння|нні|нням|нню|ти|вся|всь|лись|лися|тись|тися)

- Слово не змінюється при стемінгу, але є винятком з правил. Це небажаний випадок для алгоритму. Він змушує утримувати словник слів-винятків. Приклади *віче*, *наче*.
- Слово змінюється при стемінгу, але теж є винятком. Це найгірший випадок для алгоритму, тому що він змушує зберігати у словника одразу дві форми слова: оригинальну і стематизовану. Наприкад, слово *відер* має змінитися на *відр*, хоча інші слова, що закінчуються на *ер*, так не стематизуються (*авіадиспетчер*, *вітер*, *гравер* тощо).
- Короткі слова залишаються незмінними. Службові частини мови (прийменники, сполучники, частки), як правило, дуже короткі слова, які ігноруються алгоритмом (слова до 2-х літер включно).

Всі ці прийоми застосовують групами, які утворюють правила стемінгу. Але це значно ускладнює алгоритм пошуку ключовиків. Тому спочатку пропонується враховувати розповсюджені закінчення – не традиційні закінчення як частини слова, а послідовності літер, якими закінчується слово (табл. 3–4). В табл. 3–4 подані закінчення слів завдовжки від 1 до 4 літер. П'ять і більше літер не враховано, оскільки таких слів доволі мало (для 5-ти максимум *йтесь* (6837), для 6-ти – *ванням* (4656) тощо). Цим створено своєрідну карту для проекту стемінгу. Мета проекту – побудувати статичне дерево закінчень та охопити алгоритмом всі гілки дерева. В загальному випадку можна побудувати більш деталізоване дерево [18–22], проте для комерційного контенту обираємо зважений рівень деталізації – від 500 слів зі спільним закінченням.

Розглянемо докладніше ідею стемера Портера, а саме знаходження основи слова для заданого вихідного слова [23–30]. Алгоритм не використовує баз основ слів, а працює, послідовно застосовуючи ряд правил відсікання закінчень і суфіксів (рис. 3).

Таблиця 3

Статична таблиця розповсюджених закінчень

я (164062)	тися (10379)	мось (20536)	али (10666)	ному (19112)	ові (17191)	а (68134)	их (31127)
ся (148160)	лися (10338)	лось (10231)	ними (19089)	о (90454)	сті (8731)	на (21328)	ах (20023)
ня (9765)	теся (19103)	тись (10366)	м (119779)	мо (33568)	ості (7636)	ла (17945)	ях (9855)
ося (30769)	лася (10230)	лись (10337)	т (2980)	го (31445)	ю (80877)	ка (11029)	них (19092)
ься (25211)	ь (151355)	тесь (19105)	ім (31343)	ло (17238)	ою (39616)	істю (7598)	ї (34702)
ися (21940)	сь (111459)	лась (10229)	им (31166)	ймо (11229)	ню (10075)	й (77109)	ої (31421)
еся (19105)	ть (33055)	ість (7606)	ам (20154)	ємо (11136)	ною (20280)	ій (33241)	ної (19098)
шся (11775)	ось (30788)	и (123402)	ом (17018)	ого (31389)	кою (7497)	ий (31136)	в (32681)
ася (10235)	ись (22656)	ми (62080)	ям (15717)	ало (10465)	нню (9054)	ала (10610)	ів (15898)
вся (10076)	есь (19114)	ти (20025)	нім (19333)	ного (19090)	стю (7648)	е (66988)	ав (10547)
юся (8044)	ась (10239)	ли (17711)	ним (19093)	і (90275)	у (94504)	те (32651)	ш (19163)
ння (9001)	всь (10016)	ими (31121)	ням (9434)	ні (31679)	му (35023)	не (20257)	єш (11138)
мося (20532)	сть (7688)	ами (20106)	нням (8975)	ві (22543)	ну (23125)	йте (11230)	є (11466)
лося (10233)	юсь (8047)	ями (9844)	ку (11624)	ті (12596)	ній (19549)	єте (11137)	к (7299)
ться (25036)	ють (11222)	ати (10819)	ому (31585)	нні (9909)	ний (19042)	х (61506)	

Таблиця 4

Статичне дерево закінчень, сумарна питома вага яких менша ніж 1 %

р (2709)	ч (959)	г (636)	п (341)	щ (110)
н (2531)	с (914)	з (581)	б (281)	ц (34)
д (1038)	л (754)	ж (353)	ф (214)	ґ (4)

Спочатку введемо деякі визначення:

• **Голосні літери** – *а*, *е*, *і*, *ї*, *о*, *у*, *и*, *е*, *ю*, *я*.

• **RV** – частина слова після першої голосної. Вона є порожньою, якщо голосних в слові немає.

• **R1** – частина слова після першого поєднання *голосна-приголосна*.

• **R2** – частина **R1** після першого поєднання *голосна-приголосна*.

Наприклад, у слові *інформаційний*: RV = *нформаційний*, R1 = *формаційний*, R2 = *маційний*.

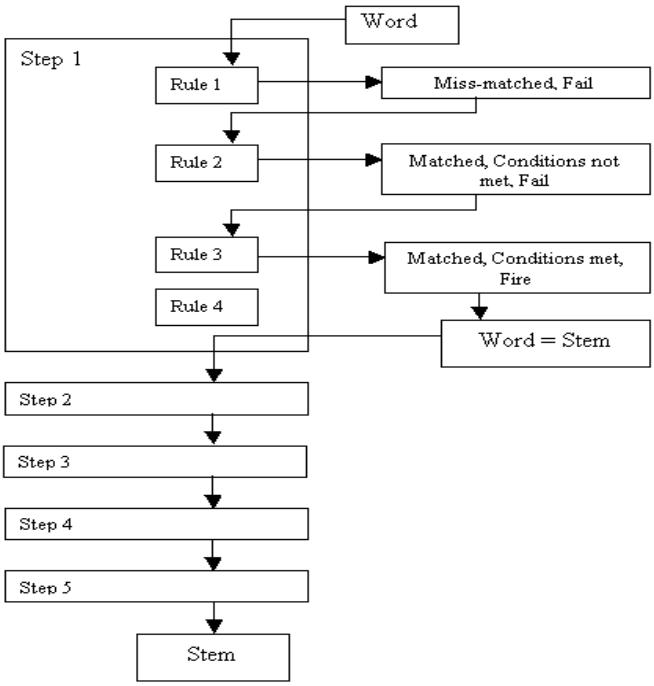


Рис. 3. Структурна схема алгоритму стемера Портера

Тепер визначимо декілька класів закінчень слова, залишивши їх оригінальні назви з вихідного опису алгоритму.

Class 1. PERFECTIVE GERUND

• **Group 1**: *в*, *вши*, *вшися*. Закінченням повинна передувати літера *а* чи *я*.

• **Group 2**: *ив*, *ивши*, *ившися*.

Class 2. ADJECTIVE (прикметник)

а, *е*, *і*, *и*, *ими*, *іми*, *ій*, *ий*, їм, *ім*, *им*, *ього*, *ого*, *ьому*, *ому*, *їх*, *их*, *ую*, *юю*, *ая*, *яя*, *ою*, *єю*.

Class 3. PARTICIPLE (дієприкметник)

• **Group 1**: *вш*, *ювa*, *ува*, *уч*, *юч*, *л*. Закінченням повинна передувати літера *а* чи *я*.

• **Group 2**: *нн*, *н*, *ячи*, *ачи*, *ова*, *ову*, *єм*.

Class 4. REFLEXIVE (рефлексивна флексія)

ся, *сь*.

Class 5. VERB (дієслово)

• **Group 1**: *ла*, *є*, *єте*, *йте*, *ли*, *лою*, *й*, *в*, *єм*, *ємо*, *ний*, *ло*, *ть*, *но*, *ють*, *ні*, *ть*, *єш*. Закінченням повинна передувати літера *а* чи *я*.

• **Group 2**: *ила*, *ела*, *ена*, *йте*, *ите*, *єте*, *юй*, *уй*, їй, *ай*, *ало*, *ив*, *или*, *имо*, *ений*, *ило*, *їло*, *ено*, *ють*, *ать*, *ені*, *ять*, *іть*, *ить*, *иш*, *ую*, *ю*.

Class 6. NOUN (іменник)

а, *ев*, *ов*, *і*, *тя*, *е*, *ами*, *іями*, *ями*, *єї*, *єю*, *ями*, *ям*, *її*, *и*, *ою*, *ій*, *ой*, *ий*, *й*, *им*, *им*, *ім*, *ам*, *ом*, *о*, *у*, *ах*, *ях*, *ую*, *ю*, *ія*, *я*.

Class 7. SUPERLATIVE (найвищий ступінь порівняння – найдовший, миліший, більший)

ш, *іш*.

Class 8. DERIVATIONAL (словотворча флексія – милість, щедрість, малість, крайність)

ість.

Class 9. ADJECTIVAL (ад'єктивне словотворення) визначається як ADJECTIVE або PARTICIPLE + ADJECTIVE. Наприклад: *падюча= пада + юч + а*.

Rules. При пошуку закінчення з усіх можливих вибирають найдовше. Наприклад, у слові *інформація* обираємо закінчення *ія*, а не *я*. Всі перевірки проводяться над частиною **RV**. Так, при

перевірці на PERFECTIVE GERUND попередні літери *а* і *я* також повинні бути всередині **RV**. Літери перед **RV** не беруть участь у перевірках взагалі.

Step 1. Знайти закінчення PERFECTIVE GERUND. Якщо воно існує, то видалити його і завершити цей крок. Інакше, видалити закінчення REFLEXIVE (якщо воно існує). Потім у наступному порядку перевірка та за наявності видалення закінчення: ADJECTIVAL, VERB, NOUN. Як тільки одне з них знайдено, тоді крок завершується.

Step 2. Якщо слово закінчується на *і* – видаляємо *і*.

Step 3. Якщо в **Step 2** знайдеться закінчення DERIVATIONAL, тоді видалити його.

Step 4. Можливий один з трьох варіантів:

1. Якщо слово закінчується на *н*, то видалити останню літеру.

2. Якщо слово закінчується на SUPERLATIVE, то видалити його і знову видалити останню літеру, якщо слово закінчується на *н*.

3. Якщо слово закінчується на ь, тоді видалити його.

Етап 3. Синтаксичний аналіз текстового контенту. Відомо, що синтаксис – це сукупність правил, які дають змогу будувати формули та розпізнавати правильні формули серед послідовностей символів. Для системи символьних обчислень важливим є те, що усі операції логіки висловлювань, окрім однієї, є бінарними. На цьому і буде базуватись синтаксичний аналізатор. Синтаксичним аналізом вважатимемо процес перегляду вхідної послідовності символів з метою розбору граматичної структури відповідно до заданої формальної граматики. Синтаксичний аналізатор (або парсер, англ. parser) – це програма або частина програми, яка виконує синтаксичний аналіз [10]. В загальному (не лише в комп'ютерній галузі) під поняттям синтаксичного розбору розуміють розбиття тексту на складові мови з ідентифікацією їхніх форм, призначення і синтаксичного зв'язку з іншими частинами. Це визначається значною мірою на етапі вивчення відмінків і позиціювання частин конкретної мови, які можуть бути доволі складними для формалізації у флективних мовах [22].

Речення таких мов розібрати програмно зовсім не просто. Для прикладу, у структурі людської мови є суттєві неоднозначності, тобто слова і вирази, які самі по собі можуть передавати зміст у величезній кількості варіантів, але тільки одне зі значень доречне в конкретному випадку. Успіх вибору правильного значення в переважній кількості випадків залежить від багатьох факторів контекстного змісту, а передбачити всі комбінації сенсу практично неможливо. Важко підготувати формальні правила для опису неформальної поведінки, хоча, зрозуміло, існують і строгі правила, множина яких утворює базу граматики, що формує основу синтаксичного аналізатора.

Під час синтаксичного аналізу текст оформлюється у структуру даних, зазвичай у дерево, яке відповідає синтаксичній структурі вхідної послідовності та добре підходить для подальшого опрацювання. Як правило, синтаксичні аналізатори працюють в два етапи: на першому ідентифікуються осмислені лексеми (виконується лексичний аналіз), на другому створюється дерево розбору. Наприклад (рис. 4), для арифметичного виразу 1+ 2*3:

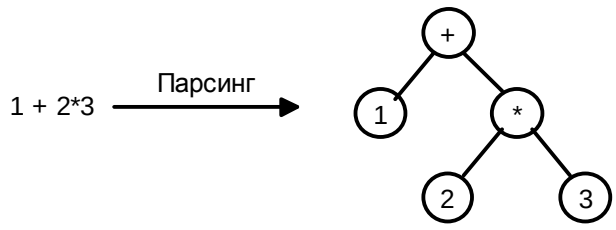


Рис. 4. Приклад розбору виразу в дерево

Токен – це послідовність одного чи більше символів, які виділяють як атомарний об'єкт. Процес формування токенів називається токенізацією, або лексичним аналізом. Виділяються токени на основі базових правил лексичного аналізатора (або лексера), які нерідко відрізняються залежно від області застосування [22]. Токени часто класифікуються за положенням (розташуванням) символів у послідовності знаків чи контексту в потоці даних. Ідеться не про просте виділення групи символів, які обмежуються розділовими знаками з обох боків (пробілами чи

знаками пунктуації). Токени визначаються правилами лексера і включають граматичні елементи мови, що використовується в потоці даних. У природних мовах зазвичай це категорії іменників, дієслів, прикметників або знаків пунктуації. Категорії використовуються в подальшому опрацюванні токенів синтаксичним аналізатором або іншими функціями в програмі.

До завдань лексичного аналізу належать такі [20]:

- Перетворення набору символів тексту на послідовність токенів.
- Виділення кожного токену як логічної частини тексту (ключове слово, ім'я змінної, знак пунктуації тощо).
- Встановлення відповідності між токеном і лексемою – конкретний текст токена ("for" "variable", ";" тощо).
- Виділення додаткових атрибутів токена (наприклад, значення змінної).
- Формування послідовності токенів на виході, яку використовуватиме парсер як вхідні дані.

Лексичний аналізатор зазвичай нічого не робить з комбінацією токенів, які він виділив. Наприклад, типовий лексичний аналізатор розпізнає дужки як знаки, але не перевіряє, чи кожній відкритій" дужці "(" відповідає закрита дужка ")". Це завдання залишається для синтаксичного аналізатора або парсера.

Етап 4. Семантичний аналіз текстового контенту. Латентно-семантичний аналіз – метод опрацювання інформації природною мовою, що дає змогу проаналізувати взаємозв'язок між колекцією документів (повідомлення, статті, тобто текстового контенту) і термінами (ключовими словами), які в них зустрічаються. Зіставляє деякі фактори (теми) зі всіма документами і термами. Спочатку слова співвідносяться з семантичними класами із словника. Потім відбувається відбір потрібних для даного речення морфосемантичних альтернатив. Далі іде зв'язування слів у єдину структуру та формування упорядкованої множини записів суперпозицій з базисних лексичних функцій і семантичних класів. Точність результату визначається повнотою/коректністю словника.

Етап 5. Референційний аналіз для формування міжфразових єдностей. Здійснюється контекстний аналіз текстового контенту C_3. За його допомогою реалізується дозвіл локальних референцій (цей, який, його) і виділення висловлювання – ядра єдності. Потім відбувається тематичний аналіз. Поділ висловлювань на тему і рему виділяє тематичні структури, які використовують, наприклад, при формуванні дайджесту. Визначають регулярну повторюваність, синонімізацію та повторну номінацію ключових слів; тотожність референції, тобто співвідношення слів з предметом відображення; наявність імплікації, основаної на ситуативних зв'язках.

Етап 6. Структурний аналіз текстового контенту. Передумовами використання є високий ступінь збігу термінів єдності, дискурсивна одиниця, речення семантичною мовою, висловлювання і елементарна дискурсивна одиниця. Виявляється базовий набір риторичних зв'язків між єдностями контенту та будується нелінійна мережа єдностей [1–7]. Відкритість набору зв'язків припускає його розширення та адаптацію для аналізу структури текстів C_3.

Існує декілька напрямів використання семантичного аналізу для визначення ключових слів як словосполучення,тобто визначення термів $Noun \in U_{K1}$ – іменників, словосполучень іменників або прикметника з іменником серед множини слів текстового контенту. Наприклад, за правилами:

1. Якщо ключовим словом є прикметник (флексія слова -ий – називний відмінок чоловічого роду). Тоді в тексті знаходять усі слова, що вживані справа від цього прикметника в будь-якому відмінку (пошук іде за основою цього прикметника) та будують для них частотний словник. Ті словосполучення, що вживані більше за певний ліміт (але можуть бути вживані менше за самий прикметник) і є новими ключовими словами. Ліміт визначає модератор.
2. Якщо ключовим словом є іменник (флексія слова не -ий), тоді аналізуються всі слова справа та зліва від нього.

a. Спочатку перевіряються всі слова зліва від нього на наявність флексій -ий. Будується також частотний словник. Визначається множина слів, які зустрічаються найчастіше за певний визначений модератором ліміт – це і є нові ключові слова.

b. Потім аналізуються всі слова справа – вони всі мають бути без флексії -ий. Аналогічно будують частотний словник, за яким визначають множину ключових слів.

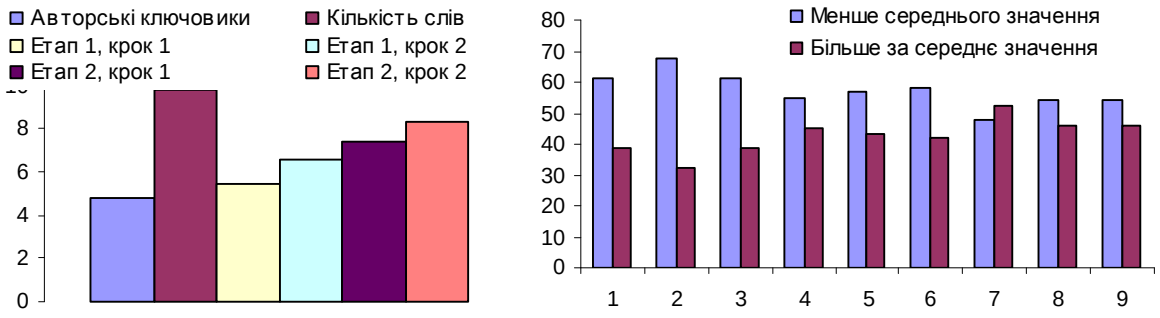
Експериментальні дослідження. Лінгвістичною базою для експериментального дослідження запропонованого методу обрано 100 наукових публікацій Вісника Національного університету "Львівська політехніка" серії "Інформаційні системи та мережі" (http://science.lp.edu.ua/sisn) з двох номерів – 783 (http://science.lp.edu.ua/SISN/SISN-2014) та 805 (http://science.lp.edu.ua/sisn/vol-cur-805-2014-2). Аналіз статистики функціонування системи виявлення множини ключових слів із 100 наукових статей було проведено у два етапи, зокрема:

1. Проаналізовано всі статті із перевіркою загальних заблокованих слів та тематичного словника.

2. Проаналізовано всі статті із перевіркою уточнених заблокованих слів та уточненого тематичного словника (з більшою кількістю запуску системи формується множина невідомих слів (відсутніх і в тематичному словнику і в множині заблокованих).

Окрім того, на кожному етапі перевірка відбувалась в два кроки для кожної статті: аналіз всієї статті (http://victana.lviv.ua/index.php/kliuchovi-slova) та аналіз статті без початку (назва, автори, УДК, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури для того, щоб визначити похибки точності формування множини ключових слів.

На рис. 4, а подано діаграму аналізу статистики формування системою множин усіх потенційних ключових слів порівняно з множиною, визначеною авторами статей. Перший стовпчик – середньоарифметична кількість ключових слів, визначених автором (4,77), а другий – середньоарифметична кількість слів, які складають ці авторські ключові слова (9,82). Третій стовпчик – середньоарифметична кількість потенційних ключових слів, визначена системно на етапі 1, крок 1 (5,46); четвертий – на етапі 1, крок 2 (6,51); п'ятий – на етапі 1, крок 1 (7,43); шостий – на етапі 2, крок 2 (8,35). Позначимо відповідно ці стовпчики $A_1 \div A_6$. Значення A_3 відмінне за значення A_1 на 0,69 (за кількістю, але не за змістом); відповідно A_4 відмінне від A_1 на 1,74; A_5 від A_1 на 2,66; A_6 від A_1 на 3,58. Значення A_2 відмінне за значення A_3 на 4,36; відповідно A_2 від A_4 на 3,31; A_2 від A_5 на 2,39; A_2 від A_6 на 1,47. Отже, автор статті в середньому зазвичай визначає меншу кількість ключових слів, ніж її реально є в цій роботі.



а *б*

Рис. 4. Результати перевірки 100 статей

Налагодження параметрів системи збільшує кількість визначених ключових слів майже удвічі (при аналогічному порівнянні з A_1 значення A_3 більше в 1,144654; A_4 – в 1,36478; A_5 – в 1,557652; A_6 – в 1,750524). Загальний приріст значення, отриманий системою залежно від модерації словників, становить відповідно для A_3 14,46541; A_4 – 36,47799; A_5 – 55,7652; A_6 – 75,05241. Якщо ж послідовно порівнювати A_2 з $A_3 \div A_6$ (у скільки разів значення A_2 більше), то отримаємо відповідно ряд 1,7985; 1,5084; 1,3217; 1,176.

На рис. 4, б подано діаграму аналізу статистики розподілу щільності тексту в аналізованих статтях, де 1 – аналіз кількості сторінок статей (відповідно менша та більша за середнє значення), 2 – абзаців в статті, 3 – рядків з текстом, 4 – слів, 5 – знаків, 6 – знаків і пробілів, 7 – слів на сторінці, 8 – знаків на сторінці, 9 – знаків та пробілів на сторінці.

На рис. 5 подано діаграму розподілу формування системою множин всіх потенційних ключових слів для кожної статті порівняно з множиною, визначеною авторами статей.

Точність визначення ключових слів збільшується в процесі модерації словників. Різниця між кількістю ключовиків, визначених автором та визначеною системою на етапі 1, крок 1 складає 44,39919 % (різниця у відсотках). Точність покращується на етапі 1, крок 2 – 33,70672 %, значно покращується на етапі 2, крок 1 – 24,33809 %, а на етапі 2, крок 2 вже становить 14,96945 %.

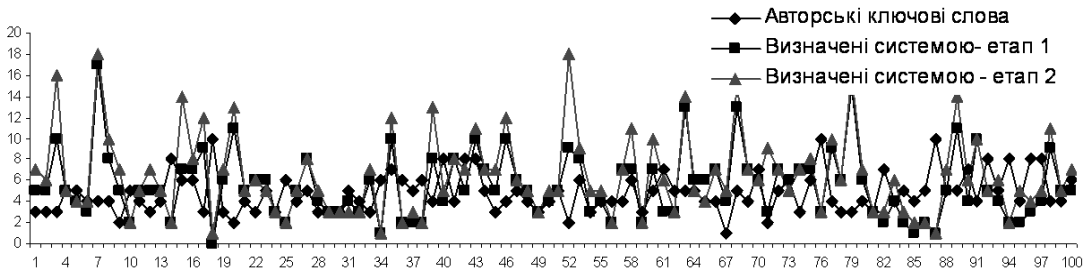


Рис. 5. Результати перевірки 100 статей

У табл. 5 наведено результати аналізу статистики формування системою множин всіх потенційних ключових слів для кожної статті порівняно з множиною, визначеною авторами статей, де А – для авторських ключових слів, Б – для ключових слів, визначених системою на етапі 1 (крок 1), В – для ключових слів, визначених системою на етапі 1 (крок 2), Г – для ключових слів, визначених системою на етапі 2 (крок 1), Д – для ключових слів визначених системою на етапі 2 (крок 2). У табл. 6–7 подано статистичні дані аналізу текстів статей при формуванні множин ключових слів для побудови відповідних гістограм для груп А–Д.

Таблиця 5

Описові статистичні дані формування ключових слів для досліджених текстів

	А	*Б*	*В*	*Г*	*Д*
Середнє	4,808081	5,515152	6,565657	7,505051	8,434343
Стандартна помилка	0,180859	0,310393	0,39035	0,301297	0,324611
Медіана	4	5	6	7	8
Мода	4	5	5	7	8
Стандартне відхилення	1,799528	3,088371	3,883932	2,997869	3,229841
Дисперсія вибірки	3,238301	9,538033	15,08493	8,987219	10,43187
Ексцес	0,652815	1,705273	0,748643	-0,45645	-0,50438
Асиметричність	0,947939	1,125305	1,065716	0,537598	0,517047
Інтервал	8	16	17	12	13
Мінімум	2	1	1	2	3
Максимум	10	17	18	14	16
Сума	476	546	650	743	835
Рахунок	99	99	99	99	99
Найбільший(1)	10	17	18	14	16
Найменший(1)	2	1	1	2	3
Рівень надійності(95,0%)	0,35891	0,615965	0,774637	0,597914	0,64418

Автор наукової статті зазвичай обирає за своїм розсудом кількість ключових слів у діапазоні від 2 до 8 слів (найчастіше – 3–5 ключовиків). Система ж визначає різну кількість слів залежно від стиля написання конкретного автора (існують такі статті, в яких система не знаходить за законом Зіпфа жодного ключового слова). Для групи Б найчастіше система визначила кількість ключовиків 5, 7 та 3 (понад 10), хоча розподіл знайдених ключових слів був в діапазоні від 1 до 18 слів (окрім 17). Для групи В найчастіше система визначила кількість ключовиків також 5, 7 та 3, хоча розподіл знайдених ключових слів був в діапазоні від 1 до 18 слів (окрім 17), але збільшилось кількість знайдених слів та досягнуто найбільшого показника надійності. Для групи С найчастіше система визначила кількість ключовиків 7, 6, 5, 10 та 8, хоча розподіл знайдених ключових слів був в діапазоні від 2 до 14 слів (значно звузився діапазон). Для групи Д найчастіше система визначила кількість ключовиків 8, 5, 7 та 10, хоча розподіл знайдених ключових слів був у діапазоні від 3 до 16 слів (покращилась точність).

Таблиця 6

Статистичні дані побудови гістограми для групи А та групи Б

А	Частота	Інтегральний %	А	Частота	Інтегральний %	Б	Частота	Інтегральний %	Б	Частота	Інтегральний %
1	0	0,00%	4	27	27,27%	1	2	2,02%	5	20	20,20%
2	4	4,04%	5	21	48,48%	2	10	12,12%	7	16	36,36%
3	20	24,24%	3	20	68,69%	3	12	24,24%	3	12	48,48%
4	27	51,52%	6	11	79,80%	4	4	28,28%	2	10	58,59%
5	21	72,73%	8	8	87,88%	5	20	48,48%	6	9	67,68%
6	11	83,84%	7	5	92,93%	6	9	57,58%	4	4	71,72%
7	5	88,89%	2	4	96,97%	7	16	73,74%	8	4	75,76%
8	8	96,97%	10	3	100,00%	8	4	77,78%	10	4	79,80%
9	0	96,97%	1	0	100,00%	9	2	79,80%	11	3	82,83%
10	3	100,00%	9	0	100,00%	10	4	83,84%	12	3	85,86%
11	0	100,00%	11	0	100,00%	11	3	86,87%	14	3	88,89%
12	0	100,00%	12	0	100,00%	12	3	89,90%	1	2	90,91%
13	0	100,00%	13	0	100,00%	13	2	91,92%	9	2	92,93%
14	0	100,00%	14	0	100,00%	14	3	94,95%	13	2	94,95%
15	0	100,00%	15	0	100,00%	15	1	95,96%	16	2	96,97%
16	0	100,00%	16	0	100,00%	16	2	97,98%	18	2	98,99%
17	0	100,00%	17	0	100,00%	17	0	97,98%	15	1	100,00%
18	0	100,00%	18	0	100,00%	18	2	100,00%	17	0	100,00%
Ще	0	100,00%	Ще	0	100,00%	Ще	0	100,00%	Ще	0	100,00%

Таблиця 7

Статистичні дані побудови гістограми для групи В, групи С та групи Д

В	Частота	Інтегральний %	В	Частота	Інтегральний %	Г	Частота	Інтегральний %	Г	Частота	Інтегральний %
1	2	2,02%	5	20	20,20%	1	0	0,00%	7	15	15,15%
2	10	12,12%	7	16	36,36%	2	1	1,01%	6	14	29,29%
3	12	24,24%	3	12	48,48%	3	5	6,06%	5	13	42,42%
4	4	28,28%	2	10	58,59%	4	9	15,15%	10	12	54,55%
5	20	48,48%	6	9	67,68%	5	13	28,28%	8	11	65,66%
6	9	57,58%	4	4	71,72%	6	14	42,42%	4	9	74,75%
7	16	73,74%	8	4	75,76%	7	15	57,58%	12	6	80,81%
8	4	77,78%	10	4	79,80%	8	11	68,69%	3	5	85,86%
9	2	79,80%	11	3	82,83%	9	4	72,73%	14	5	90,91%
10	4	83,84%	12	3	85,86%	10	12	84,85%	9	4	94,95%
11	3	86,87%	14	3	88,89%	11	1	85,86%	13	3	97,98%
12	3	89,90%	1	2	90,91%	12	6	91,92%	2	1	98,99%
13	2	91,92%	9	2	92,93%	13	3	94,95%	11	1	100,00%
14	3	94,95%	13	2	94,95%	14	5	100,00%	1	0	100,00%
15	1	95,96%	16	2	96,97%	15	0	100,00%	15	0	100,00%
16	2	97,98%	18	2	98,99%	16	0	100,00%	16	0	100,00%
17	0	97,98%	15	1	100,00%	17	0	100,00%	17	0	100,00%
18	2	100,00%	17	0	100,00%	18	0	100,00%	18	0	100,00%
Ще	0	100,00%	Ще	0	100,00%	Ще	0	100,00%	Ще	0	100,00%

Д	Частота	Інтегральний %	Д	Частота	Інтегральний %
1	0	0,00%	8	14	14,14%
2	0	0,00%	5	12	26,26%
3	1	1,01%	7	11	37,37%
4	9	10,10%	10	11	48,48%
5	12	22,22%	4	9	57,58%
6	9	31,31%	6	9	66,67%
7	11	42,42%	9	9	75,76%
8	14	56,57%	11	5	80,81%
9	9	65,66%	14	5	85,86%
10	11	76,77%	12	4	89,90%
11	5	81,82%	13	4	93,94%
12	4	85,86%	15	3	96,97%
13	4	89,90%	16	2	98,99%
14	5	94,95%	3	1	100,00%
15	3	97,98%	1	0	100,00%
16	2	100,00%	2	0	100,00%
17	0	100,00%	17	0	100,00%
18	0	100,00%	18	0	100,00%
Ще	0	100,00%	Ще	0	100,00%

Висновки і перспективи подальших наукових розвідок

У статті наведено теоретичне та експериментальне обґрунтування методу лінгвістичного аналізу україномовного комерційного контенту з використанням стемінгу Портера. Метод спрямовано на автоматичне виявлення значущих ключових слів україномовного контенту на основі запропонованої формалізації складових аналізу: граматичного (графемного), морфологічного, синтаксичного, семантичного, референційного та структурного.

Для реалізації граматичного аналізу запропоновано правила розпізнавання рядків в тексті, визначено множину еталонних моделей графем для 5-ти мов за нормальною формою Бекуса–Наура та відповідну граматику $G = <V, T, S, P>$ для виявлення змістовних одиниць аналізу $< U'_C, U'_G >$ текстового комерційного контенту X'. Морфологічний аналіз реалізовано адаптацією алгоритму Стемінг М. Портера до української мови, зокрема побудовано статичне дерево закінчень та обрано зважений рівень деталізації – від 500 слів зі спільним закінченням, обґрунтовано правила відсікання

закінчень і суфіксів. Визначено основні вимоги та процедури синтаксичного, семантичного, референційного та структурного аналізу україномовного комерційного контенту.

Експериментальне дослідження методу лінгвістичного аналізу проведено на матеріалах 100 наукових публікацій з двох номерів (783 та 805) Вісника Національного університету "Львівська політехніка" серії "Інформаційні системи та мережі" (http://science.lp.edu.ua/sisn). Побудована на основі запропонованого методу система пошуку ключових слів продемонструвала здатність до самовдосконалення формуванням та уточненням множини загальних заблокованих слів і тематичного словника за участю модераторів. Виявлено, що для технічних наукових текстів експериментальної бази автори статей в середньому зазвичай визначають меншу кількість ключових слів, ніж вона реально присутня в цій роботі. За числовими даними статистичного аналізу доведено, що налагодження параметрів системи збільшує кількість визначених ключових слів майже удвічі, не зменшуючи при цьому показників точності та надійності.

Подальшого експериментального дослідження потребує апробація запропонованого методу для визначення ключових слів з інших категорій текстів – наукових гуманітарного профілю, художніх, публіцистичних тощо.

1. Найефективніші методи залучення потенційних клієнтів [Електронний ресурс] / Центр ресурсів якості трафіку оголошень, Google AdWords. – Режим доступу: http://www.google.com/intl/uk_ALL/ads/adtrafficquality/advertisers/best-practices-for-generating-leads.html. – Назва з титул. екрана. 2. Нечёткий поиск в тексте и словаре [Електронний ресурс]. – Режим доступу: http://habrahabr.ru/post/114997/. – Назва з титул. екрана. 3. Реализации алгоритмов. Расстояние Левенштейна [Електронний ресурс]. – Режим доступу: http://ru.wikibooks.org/wiki/Реализации_алгоритмов/Расстояние_Левенштейна. – Назва з титул. екрана. 4. Задача о расстоянии Дамерау-Левенштейна [Електронний ресурс]. – Режим доступу: http://neerc.ifmo.ru/wiki/index.php?title=%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%80%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B8_%D0%94%D0%B0%D0%BC%D0%B5%D1%80%D0%B0%D1%83-%D0%9B%D0%B5%D0%B2%D0%B5%D0%BD%D1%88%D1%82%D0%B5%D0%B9%D0%BD%D0%B0. – Назва з титул. екрана. 5. Насонов, Д. Функция Левенштейна [Електронний ресурс] / Д. Насонов. – Режим доступу: http://rain.ifmo.ru/cat/data/theory/unsorted/levenshtein-2006/article.pdf. – Назва з титул. екрана. 6. Левенштейн, который сравнивает строки [Електронний ресурс] / Веб-разработка. – Режим доступу: http://dayte2.com/levenshtein. – Назва з титул. екрана. 7. Вычисление расстояния Левенштейна между двумя строками [Електронний ресурс]. – Режим доступу: http://wm-help.net/lib/b/book/827961078/78. – Назва з титул. екрана. 8. Стеммер Потера [Електронний ресурс]. – Режим доступу: http://labs.abcvg.com/stemmer/index.php. – Назва з титул. екрана. 9. Moseichuk, V. Porter stemming algorithm for Ukrainian languages [Electronic resource] / V. Moseichuk. – Access mode: http://www.marazm.org.ua/document/stemer_ua/. – Title from the screen. 10. Стемінг [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Стемінг. – Назва з титул. екрана. 11. Russian stemming algorithm [Electronic resource]. – Access mode: http://snowball.tartarus.org/algorithms/russian/stemmer.html. – Title from the screen. 12. Porter stemmer – реализация алгоритма стеммера Портера для русского языка на чистом функциональном языке Clojure [Електронний ресурс]. – Режим доступу: https://github.com/allaud/porter-stemmer. – Назва з титул. екрана. 13. The Porter Stemming Algorithm – Porter's homepage. [Електронний ресурс]. – Режим доступу: http://tartarus.org/~martin/PorterStemmer/. – Назва з титул. екрана. 14. The Porter Stemming Algorithm – Project "Snowball" [Electronic resource]. – Access mode: http://snowball.tartarus.org/algorithms/porter/stemmer.html. – Title from the screen. 15. The English (Porter2) stemming algorithm – Project "Snowball" [Electronic resource]. – Access mode: http://snowball.tartarus.org/algorithms/english/stemmer.html. – Title from the screen. 16. Porter M. F. An algorithm for suffix stripping [Electronic resource] / M. F. Porter // Program. – 1980. – Т. 14. – № 3. – P. 130–137. – Access mode: http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html. – Title from the screen.

17. Willett P. The Porter stemming algorithm: then and now [Electronic resource] / P. Willett // Program: Electronic Library and Information Systems. – 2006. – В. 3, т. 40. – P. 219–223. – Access mode: http://eprints.whiterose.ac.uk/1434/. – Title from the screen. 18. Сеник М. Вільний алгоритм стемінгу для української мови [Електронний ресурс] / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ukr_stemming/stemming_about.html. – Назва з титул. екрана. 19. Сеник М. Інструмент для пошуку слів з однаковими закінченнями [Електронний ресурс] / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ukr_stemming/word_by_ending.html. – Назва з титул. екрана. 20. Сеник М. Статичне дерево закінчень [Електронний ресурс] / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ukr_stemming/ukr_endings.html#dyn. – Назва з титул. екрана. 21. Сеник М. Демо стемінгу для української мови [Електронний ресурс] / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ukr_stemming/demo.html. – Назва з титул. екрана. 22. Вероятностный морфологический анализатор русского и украинского языков [Електронний ресурс]. – Режим доступу: http://www.keva.ru/stemka/stemka.html. – Назва з титул. екрана. 23. Стемминг [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Стемминг. – Назва з титул. екрана. 24. Lovins J.B. Development of a stemming algorithm / J. B. Lovins // Mechanical Translation and Computational Linguistics 11:22–31. – 1968. 25. Jongejan B. Automatic training of lemmatization rules that handle morphological changes in pre-, in– and suffixes alike [Electronic resource] / B. Jongejan, H. Dalianis // Proceeding of the ACL-2009, Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Singapore, August 2–7, 2009. – Singapore, 2009. – P. 145–153. – Access mode: http://www.aclweb.org/anthology/P/P09/P09-1017.pdf. – Title from the screen. 26. Вірогідний морфологічний аналізатор російської та української [Електронний ресурс]. – Режим доступу: http://www.keva.ru/stemka/stemka.html. – Назва з титул. екрана. 27. Модуль Drupal для стемінга українською. Новий модуль для алгоритму Стема для Українського пошуку з виділенням коренів [Електронний ресурс]. – Режим доступу: http://drupal.ua/node/1170. – Назва з титул. екрана. 28. Стемінг Портера для української мови [Електронний ресурс]. – Режим доступу: http://www.marazm.org.ua/document/stemer_ua/. – Назва з титул. екрана. 29. Hardcoded stemmer for Ukrainian [Electronic resource]. – Access mode: https://github.com/vgrichina/ukrainian-stemmer. – Title from the screen. 30. Perestoronin P. Стеммер Портера для русского языка [Електронний ресурс] / P. Perestoronin. – Режим доступу: http://blog.eigene.in/post/49598738049/snowball. – Назва з титул. екрана. 31. Берко А. Ю. Системи електронної контент-комерції: монографія / А. Ю. Берко, В. А. Висоцька, В. В. Пасічник. – Львів: Видавництво Національного університету "Львівська політехніка", 2009. – 612 с. 32. Математична лінгвістика. [Книга 1. Квантитативна лінгвістика] : навч. посібник / [В. В. Пасічник, Ю. М. Щербина, В. А. Висоцька, Т. В. Шестакевич]. – Львів : "Новий світ -2000", 2012. – 359 с. – (Серія "Комп'ютинг"). 33. Methods based on ontologies for information resources processing : Monograph / [Vasyl Lytvyn, Victoria Vysotska, Lyubomyr Chyrun, Dmytro Dosyn].- Saarbrücken: LAP Lambert Academic Publishing, 2016. – 324 с. – Access mode: https://www.lap-publishing.com/catalog/details/store/gb/book/978-3-659-89905-8/methods-based-on-ontologies-for-infor-mation-resources-processing?locale=gb. 34. Висоцька В. А. Методи і засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції : автореферат дисертації на здобуття наукового ступеня кандидата технічних наук : 05.13.06 – інформаційні технології / Вікторія Анатоліївна Висоцька ; Національний університет "Львівська політехніка". – Львів, 2014. – 27 с. 35. Berko A. Features of information resources processing in electronic content commerce / Andriy Berko, Victoria Vysotska, Lyubomyr Chyrun // Applied Computer Science. ACS journal. – 2014. – Vol. 10, Number 2. — P. 5–19. – Режим доступу: www.acs.pollub.pl, http://www.acs.pollub.pl/index.php/-current-issue/vol-10-no-122014.html. 36. Vysotska V. Linguistic Analysis of Textual Commercial Content for Information Resources Processing / Victoria Vysotska // Proceedings of the XIIIth International Conference on Modern Problems of Radio Engineering,

Telecommunications and Computer Science (TCSET'2016), February 23–26, 2016, Lviv–Slavske, Ukraine.– Lviv, 2016.– P. 709–713. 37. Бісікало О. В. Виявлення ключових слів на основі методу контент-моніторингу україномовних текстів / О. В. Бісікало, В. А. Висоцька // Науковий журнал Радіоелектроніка. Інформатика. Управління. – Запоріжжя: ЗНТУ, 2016/1. – № 1(36). – С. 74–83. – http://ric.zntu.edu.ua/. 38. Chyrun L. Informational resources processing intellectual systems with textual commercial content linguistic analysis usage constructional means and tools development / L. Chyrun, V. Vysotska, I. Kozak // Econtechmod : an international quarterly journal on economics in technology, new technologies and modelling processes. – Lublin ; Rzeszow, 2016. – Vol. 5, nomber 2. – P. 85–94. 39. Кондратєв Є. Контент-аналіз текстових масивів даних / Євген Кондратєв, Вікторія Висоцька // 4 Міжнародна наукова конференція ІКС-2015 "Інформація, комунікація, суспільство 2015, 20–23 травня 2015, Україна, Львів-Славське. – Львів, 2015. – С. 170-171. 40. Vysotska V. Features of the content-analysis method for text categorization of commercial content in processing online newspaper articles / Victoria Vysotska, Lyubomyr Chyrun // Applied Computer Science. ACS journal. – 2015. – Volume 11, Number 1. – P. 5–19. – Режим доступу: www.acs.pollub.pl, http://www.acs.pollub.pl/index.php/-current-issue/applied-computer-science-volume-11-number-1-2015.html, http://www.acs.pollub.pl/pdf/v11n1/2.pdf. 41. Vysotska V. Linguistic analysis and modelling semantics of textual content for digest formation / Victoria Vysotska, Lyubomyr Chyrun // MEST Journal (Management Education Science & Society Technologie). – 2015. – Vol.3 No.1. – P. 127-148. – Режим доступу: http://mest.meste.org/MEST_1_2015/Sadrzaj_eng.html http://mest.meste.org/MEST_1_2015/5_15.pdf. 42. Висоцька В. А. Особливості моделювання синтаксису речення слов'янських та германських мов за допомогою породжувальних контекстно-вільних грамматик / В. А. Висоцька // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 814: Інформаційні системи та мережі. – С. 246–276. 43. Висоцька В. А. Особливості рубрикації текстового комерційного контенту / В. А. Висоцька // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 826: Комп'ютерні науки та інформаційні технології. – С. 359–367. 44. Застосування контент-аналізу для опрацювання текстових масивів даних / Я. П. Кісь, В. А. Висоцька, Л. Б. Чирун, В. М. Фольтович // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 814: Інформаційні системи та мережі. – С. 282–292. 45. Чирун Л. Б. Особливості методів контент-аналізу текстових масивів даних web-ресурсів в межах регіону / Л. Б. Чирун, В. В. Кучковський, В. А. Висоцька // Вісник Національного університету "Львівська політехніка". – 2015. – № 829: Інформаційні системи та мережі. – С. 296–320. 46. Моделювання семантики речення природною мовою за допомогою породжувальних грамматик / Т. В. Шестакевич, В. А. Висоцька, Л. В. Чирун, Л. Б. Чирун // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 814: Інформаційні системи та мережі. – С. 335–352. 47. Бісікало О. В. Експериментальне дослідження пошуку значущих ключових слів україномовного контенту / О. В. Бісікало, В. А. Висоцька // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 829: Інформаційні системи та мережі. – С. 255–272. 48. Берко А. Ю. Лінгвістичний аналіз текстового комерційного контенту / А. Ю. Берко, В. А. Висоцька, Л. В. Чирун // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 814: Інформаційні системи та мережі. – С. 203–227. 49. Vysotska V. Generative regular grammars application to modeling the semantics of sentences in natural language / Victoria Vysotska // Теорія і практика: Вісник Нац. ун-ту "Львівська політехніка". – 2014. – № 808: Комп'ютерні системи проектування. – С. 43–56. 50. Висоцька В. А. Особливості генерування семантики речення природною мовою за дапомогою породжувальних необмежених та контекстно-залежних граматик / В. А. Висоцька // Вісник Нац. ун-ту "Львівська політехніка". – 2014. – № 783: Інформаційні системи та мережі. – С. 271–292. 51. Шестакевич Т. В. Застосування породжувальних граматик для генерування речень українською мовою / Т. В. Шестакевич, В. А. Висоцька // Східно-Європейський журнал передових технологій. – Харків, 2012. – № 3/2 (57). – С. 51–53. 52. Висоцька В. А. Застосування породжувальних граматик для моделювання синтаксису речення / В. А. Висоцька, Т. В. Шестакевич, Ю. М. Щербина // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 743: Інформаційні системи та мережі. – С. 175–190. 53. Берко А. Ю.

Застосування методу контент-аналізу для формування інформаційних ресурсів в системах електронної контент-комерції / А. Ю. Берко, В. А. Висоцька, М. М. Сороковський // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 743: Інформаційні системи та мережі. – С. 3–15. 54. Висоцька В. А. Утворення речень англійською та німецькою за допомогою породжувальних граматик / В.А. Висоцька, Т.В. Шестакевич, Ю. М. Щербина // Вісник Нац. ун-ту "Львівська політехніка". – 2012. – № 744: Комп'ютерні науки та інформаційні технології. – С. 142–152. 55. Висоцька В. А. Інтелектуальна система розподілу дайджестів між працівниками електронних засобів масової інформації / В. А. Висоцька, О. Ю. Окрушко // Вісник Нац. уні-ту "Львівська політехніка". – 2012. – № 744: Комп'ютерні науки та інформаційні технології. – С. 41–53. 56. Утворення українських дієприкметників за допомогою породжувальних граматик / Ю. М. Щербина, Ю. В. Нікольський, В. А. Висоцька, Т. В. Шестакевич // Вісник Нац. ун-ту "Львівська політехніка". – 2011. – № 715: Інформаційні системи та мережі. – С. 354–369.

УДК 004.912

Н. В. Борисова, З. А. Кочуєва, І. В. Оліфенко
Національний технічний університет "Харківський політехнічний інститут"

МЕТОД АВТОМАТИЗОВАНОЇ ЛЕМАТИЗАЦІЇ ДІЄСЛІВ НІМЕЦЬКОЇ МОВИ

Представлено математичне, алгоритмічне та програмне забезпечення розв'язання задачі автоматизованої лематизації німецьких дієслів з відокремлюваними префіксами.

Ключові слова: лематизація німецьких дієслів, автоматизована лематизація, розробка програм-лематизаторів.

Mathware, algorithmic support and software for problem solution of automated lemmatization of German verbs with separated prefixes are represented in the article.

Key words: lemmatization of German verbs, automated lemmatization, development of lemmatizers.

Вступ. Загальна постановка проблеми

Необхідність у створенні програмного забезпечення для автоматизованої лематизації виникає тому, що процедуру лематизації використовують для вирішення багатьох завдань автоматизованого опрацювання природної мови, а саме:

- при індексуванні веб-документів;
- у пошукових алгоритмах для підвищення релевантності пошуку;
- для визначення унікальності текстового контенту;
- у процесі схематизації веб-документів;
- для попереднього опрацювання текстів при класифікації документів;
- при створенні машинних словників;
- у системах машинного перекладу;
- при розмітці корпусів текстів;
- у системах, що навчають іноземної мови тощо.

Під лематизацією будемо розуміти одну з задач морфологічного аналізу, що полягає у приведенні різних текстових форм слова до його нормальної форми шляхом відкидання від словоформи флективних закінчень і повернення основної або словникової форми слова [2].