# 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

# ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПОШУКУ ЗНАЧУЩИХ КЛЮЧОВИХ СЛІВ УКРАЇНОМОВНОГО КОНТЕНТУ



**Проведено порівняльне експериментальне дослідження методів пошуку значущих ключових слів україномовного контенту. В основу підходу до автоматичного визначення ключових слів покладено стемінг Портера слів української мови за відстанню Левенштейна, враховано можливості використання тематичного словника та вилучення заблокованих слів. На експериментальній базі зі 100 наукових публікацій технічного спрямування порівняно з авторськими варіантами отримано числові статистичні характеристики точності результатів пошуку.**

**Ключові слова: стемінг Портера, відстань Левенштейна, українська мова, ключові слова, пошук, тематичний словник.**

**This article presents the comparative experimental research of methods of relevant keywords finding in Ukrainian-language content. Based approach to automatic determination keywords Porter stemming for Ukrainian language words by distance Lowenstein, take into account the possibility of using a thematic dictionary and removal of blocked words is incorporated. On an experimental basis with 100 scientific publications of technical direction compared to the author's version received numerous statistical characteristics of precision results.**

**Key words: Porter Stemming, Levenshtein distance, Ukrainian language, keywords, search, thematic dictionary.**

## Вступ. Загальна постановка проблеми

Найефективніші методи залучення потенційних клієнтів та відвідувачів на інформаційні ресурси в інтернет-просторі – це правильне формування та вживання множини ключових слів у контенті, поданому в цих інформаційних ресурсах. Одними із найпростіших способів (але кропітких та розтягнутих у часі) – це опитування постійних користувачів та формування множини вподобань, переваг та недоліків (так зване рейтингування) контенту, розміщеного на інформаційному ресурсі. Наприклад, на сайтах деяких наших рекламодавців для користувачів наведено форми, за допомогою яких вони можуть надіслати свої контактні дані, отримати додаткову інформацію про товари чи послуги рекламодавця та залишити свої коментарі. Особи, які надали відомості за допомогою таких форм на сайті рекламодавця, – це його потенційні клієнти, з якими можна спілкуватися по телефону чи електронною поштою, а також вручну проаналізувати масиви текстового контенту для формування статистичних даних щодо переваг та вподобань потенційної і/або постійної аудиторії.

Іншим популярним способом є використання пакета оптимізації роботи кампаній Google AdWords [7], який допоможе максимально ефективно використати надіслані дані. Так, завдяки Google AdWords розміщують рекламу біля результатів пошуку, щоб збільшити трафік до

інформаційного ресурсу та продажі контенту/товару. Після натискання оголошення відвідувачі розраховують виконати певні дії на цільовій сторінці. Важливо, щоб у відвідувача перед натисканням оголошення склалося правильне уявлення про те, на що можна розраховувати. Щоб досягти цього, оптимізують кампанію AdWords, установлюють ставки для потрібних ключових слів, за необхідності використовують мінус-слова (переглянувши свої звіти про пошукові терміни), використовують лаконічний, але змістовний і привабливий текст оголошення, а також налаштовують відстеження конверсій. Відстеження показника відмов і коефіцієнта конверсії цільової сторінки, а також їх порівняння для різних варіантів тексту оголошення, яке залучає трафік на сторінку, допоможе визначити, яке оголошення було ефективнішим. Залишається головне питання – як оптимально, адекватно та ефективно визначити ключові слова. Для англомовних текстів це вже не є проблемою – відомо багато публікацій на цю тему, розроблено відповідні програмні продукти. Для україномовних текстів аналогічні алгоритми не є ефективними порівняно з англомовними, тому проблема пошуку українських ключових слів нині актуальна.

**Зв'язок висвітленої проблеми із важливими науковими та практичними завданнями**

У статті розв'язана науково-практична задача розроблення методу опрацювання української текстової інформації для автоматичного виявлення значущих ключових слів та рубрикації контенту в інтернет-системах. 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 в межах діяльності науково-дослідного центру прикладної та комп'ютерної лінгвістики. Результати досліджень отримано під час виконання 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 провадилися також відповідно до 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**

**Вимоги до вибору й упорядкування ключових слів з рекламною метою**

Вибір правильного списку ключових слів для кампанії допомагає показувати оголошення цільовим клієнтам, коли вони шукають інформацію за відповідними термінами чи відвідують певні інформаційні ресурси [7]. Ключові слова мають збігатися із запитами, за якими потенційні клієнти шукатимуть товари або послуги [7].

По-перше, необхідно завжди враховувати думку потенційного користувача, для цього треба поставити себе на місце клієнта або проаналізувати опитувальники, які заздалегідь виставлено на інформаційному ресурсі. Необхідно визначити основні категорії контенту інформаційного ресурсу (скласти список рубрик), а також фрази й терміни, які характеризують ці категорії (скласти термінальний словник, де кожне слово належить певній рубриці, або у відсотковому відношенні – кільком рубрикам). У цей список необхідно також внести терміни або фрази, якими б користувачі могли описати контент, продукти чи послуги. Наприклад, для опису спортивного чоловічого взуття використовують множину ключових слів $M$={*чоловіче спортивне взуття*, *чоловічі кросівки*, *чоловіче тенісне взуття*, …}, розширивши цей список як загальними категоріями, які б використали клієнти, так і уживанішими термінами з використанням назв брендів і продуктів.

По-друге, необхідно вибирати загальні або конкретні ключові слова залежно від поставленої мети. Але якщо ключові слова занадто конкретні, це не дасть змоги охопити всю бажану цільову аудиторію. Для максимального охоплення тематики інформаційного ресурсу використовують загальніші ключові слова. Але ключові слова з надто загальним значенням не завжди ефективні, оскільки через них оголошення часто відображається за пошуковими запитами, мало пов'язаними з тематикою інформаційного ресурсу, що зменшує шанси на залучення потенційної аудиторії. Окрім того, за такі ключові слова завжди ведеться гостра конкуренція.

Для збільшення точності конкретизації опису контенту використовують як конкретніші, так і загальніші ключові слова, а потім визначають, наприклад, через Google Analytics, які з них дають кращі результати. Незалежно від рівня загальності чи конкретності ключових слів вони завжди мають максимально відповідати профілю інформаційного ресурсу та загальній тематиці контентного потоку. Це чудовий спосіб уникнути повторюваних ключових слів у обліковому записі, оскільки для певного ключового слова система Google відображає лише одне оголошення від рекламодавця. Наприклад, для опису спортивного чоловічого взуття можна розширити множину ключових слів *M*={*чоловіче взуття для баскетболу*, *дитяче спортивне взуття*, *спортивне взуття для тенісу*, …}, що відповідають різновиду пропонованої продукції. У такому випадку оголошення або в списку як результат роботи пошукової системи відображатиметься, коли хтось шукатиме конкретний тип взуття або переглядатиме відповідний тематичний інформаційний ресурс, наприклад, з баскетболу або тенісу. У список ключових слів можна додати загальні слова (наприклад, *взуття*). У такому випадку оголошення відображатиметься під час пошуку взуття будь-якого типу, а також на інформаційних ресурсах, що стосуються моди.

По-третє, необхідно групувати схожі ключові слова за темами. Якщо додати всі ключові слова й оголошення до однієї групи оголошень, то клієнт, який шукатиме жіноче вечірнє взуття, може побачити оголошення про чоловіче взуття для тенісу. Щоб показувати потенційним клієнтам релевантніші оголошення, необхідно групувати ключові слова й оголошення в групи оголошень на основі продуктів, послуг або інших категорій. До того ж, якщо розподілити ключові слова за тематичними групами, зручніше користуватися обліковим записом. Наприклад, власник магазину взуття може створити дві групи оголошень: одну для бігового взуття й одну для вечірнього. Група оголошень для бігового взуття може містити ключові слова *взуття для бігу* та *кросівки для бігу* й оголошення, призначені для людей, які шукають взуття для бігу. Група оголошень для вечірнього взуття може містити ключові слова *вечірнє взуття* та *модельні черевики* й оголошення, призначені для людей, які шукають вечірнє взуття. Тоді потенційні клієнти бачать оголошення про вечірнє взуття, коли шукатимуть за одним із ключових слів у цій групі оголошень, як-от *модельні черевики*.

По-четверте, необхідно вибрати потрібну кількість ключових слів. Частіше вказують 5–20 ключових слів для групи оголошень, хоча можна і більше. Але кожна створена група оголошень має містити ключові слова, які безпосередньо стосуються теми цієї групи. У групу оголошень не потрібно вводити інші варіанти ключових слів, наприклад, форму множини ключового слова або ключові слова з можливими орфографічними помилками. Найефективніше працюють переважно *фрази*, тобто ключові слова з двох або трьох слів. Можна вибрати до 20 000 окремих варіантів націлювання (разом з ключовими словами) у групі оголошень і до 5 мільйонів – в обліковому записі. Але переважну кількість релевантних кліків ініціює невелика група точно націлених ключових слів. Наприклад, якщо група оголошень містить ключове слово із широкою відповідністю *тенісне взуття*, оголошення відображатимуться, коли хтось здійснюватиме пошук за будь-якою варіацією цього ключового слова: *тенісне взуття*, *придбати тенісне взуття*, *взуття для бігу* або *тенісні кросівки*.

**Використання планувальника ключових слів, мінус-слів і звіту про пошукові терміни**

1. Планувальник ключових слів використовують для пошуку нових варіантів ключових слів. За допомогою планувальника ключових слів можна знаходити нові варіанти ключових слів і отримувати приблизну оцінку трафіку, що значно допоможе створити кампанію в пошуковій мережі. Цей інструмент також дає змогу дізнатися, наскільки ефективним може бути список

ключових слів, і переглянути середню кількість разів, коли користувачі здійснювали пошук за цими термінами. На основі цих даних вирішують, які ключові слова можуть допомогти збільшити обсяг трафіку для інформаційного ресурсу та підвищити рівень поінформованості щодо рекламованого продукту тощо. Наприклад, якщо в планувальнику ключових слів ввести фразу *бігове взуття*, можуть бути запропоновані такі додаткові ключові слова, як *бігове взуття зі знижкою* або *бігове взуття з керуванням рухом*. Для кожного варіанта ключового слова передбачена статистика, наприклад, оцінка конкурентоспроможності ключового слова або середня кількість пошукових запитів за цим терміном. Ця статистика допомагає вирішити, які ключові слова треба додати до свого списку.

2. Мінус-слова підвищують рейтинг кліків. Інколи виникає необхідність запобігти показу оголошення за термінами, які не відповідають продуктам чи послугам. У такому разі додають мінус-слова, щоб зменшити витрати й активувати показ оголошень лише за потрібними пошуковими термінами. Наприклад, для інформаційного ресурсу, де продається лише *чоловіче взуття для бігу*, можна додати терміни *жінки* та *дівчата* як мінус-слова, щоб оголошення не відображалося, коли люди шукатимуть відповідне взуття.

3. Звіт про пошукові терміни покращує список ключових слів та надає відомості про те, що саме шукали користувачі, коли побачили оголошення та клацнули на ньому. На основі цих даних видаляють неефективні ключові слова й додають до списку нові. Окрім того, визначають мінус-слова.

4. Алгоритми парсеру та стемінгу для визначення ключових слів у множині текстового контенту [11–18, 22–30]. Парсер (англ. Parser) – синтаксичний аналізатор, що, зазвичай програмно, перетворює вхідний текст до структурованого формату. Для контекстозалежних граматик, яким відповідають природні мови, алгоритми парсерингу характеризуються високою складністю та недостатньою якістю – особливо це стосується мов синтетичного типу, зокрема слов'янських. Парсери сучасних лінгвістичних пакетів дають змогу вийти на словникові значення не тільки окремих слів, але й відповідних словоформ, лексем та лем (майже коренів слова).

Стемінг (англ. Stemming) – це процес скорочення слова до основи відкиданням допоміжних частин, таких як закінчення чи суфікс [11–18, 22–30]. Результати стемінгу подібні на визначення кореня слова, але його алгоритми ґрунтуються на інших принципах. Тому слово після опрацювання алгоритмом стемінгу (стематизації) часто відрізнятися від морфологічного кореня слова. Стемінг застосовують у лінгвістичній морфології, контент-аналізі та контент-моніторингу. Пошукові системи використовують стемінг для об'єднання слів, у яких збігаються форми після стематизації, так звані технічні синоніми. Цей процес є злиттям. Під час стемінгу слова *швидко*, *швидкий*, *швидкі* перетворюються до форми *швидк*, а слова *бігом*, *бігаю*, *бігати* – до кореня слова *біг*. Вперше алгоритм стемінгу опубліковано в [20] – це була передова робота, яка справила великий вплив на подальші дослідження у цьому напрямі. Пізніше алгоритм стемінгу написав Мартін Портер та опублікував у липні 1980 р. у журналі Program. Цей алгоритм набув поширення та став де-факто стандартним алгоритмом стемінгу для англійської мови. Існує досить багато реалізацій портерівського алгоритму, що вільно поширюються у програмному забезпеченні, але деякі з них мають певні вади. Як результат, не всі алгоритми стемінгу видають результат, який від них очікують. Щоб зменшити кількість таких помилок, Мартін Портер створив офіційну вільну реалізацію алгоритму в 2000 р. А наступні кілька років займався побудовою Snowball, спеціального середовища для написання алгоритмів стемінгу, яке призначене для вдосконалення стемінгу англійської мови та написання алгоритмів стемінгу ще для кількох мов.

**Використання типів відповідності ключового слова, формування множини ключовиків вручну або за допомогою алгоритмів стемінгу**

– Типи відповідності ключового слова використовують для кращого націлювання оголошень. Наприклад, за точної відповідності оголошення відображатимуться, лише коли користувачі шукатимуть за конкретним ключовим словом або близьким до нього варіантом (з орфографічними помилками чи у формі множини). Ключові слова не чутливі до регістра, тому не важливо, з великої чи з малої літери їх введено. Наприклад, не потрібно вказувати обидва варіанти *взуття для бігу* та *Взуття для бігу* як ключові слова, адже достатньо одного варіанта – *взуття для*

*бігу*. Для кампаній із функцією завантаження додатка система AdWords може розширити діапазон відповідності деяких ключових слів відповідно до специфіки додатків, зокрема:

– **Ключові слова з точною та фразовою відповідністю –** внесення незначних змін (як-от вилучення чи додавання слова) у пошукові терміни для покращення відповідності цільовим ключовим словам.

– **Ключові слова із широкою відповідністю –** використання інформації про категорію додатка для уточнення націлювання та вдосконалення охоплення.

Наприклад, для відображення лише для людей, зацікавлених у придбанні чоловічого взуття для бігу, можна додати терміни *чоловіче взуття для бігу* та *чоловіче бігове взуття* як ключові слова з точною відповідністю. Тоді оголошення відображатиметься, коли користувачі шукатимуть точно за цими термінами або їх близькими варіантами, як-от *чоловіче взуття для бігу*. Воно не відображатиметься, коли пошук здійснюватиметься за такими термінами, як *найкраще чоловіче бігове взуття*, тому що ця фраза містить термін *найкраще*, якого немає у ключовому слові з точною відповідністю та який не є його близьким варіантом.

2. Вибір ключових слів, пов'язаних з додатками або інформаційними ресурсами, які переглядають клієнти. В Інтернеті завдяки списку ключових слів оголошення відображаються у відповідних додатках або на інформаційних ресурсах, які відвідують потенційні клієнти. Тому важливо вибирати ключові слова, пов'язані одне з одним, а також із вмістом, який переглядають потенційні клієнти. Наприклад, система AdWords може розширити діапазон відповідності ключових слів, щоб відобразити оголошення для релевантніших пошукових термінів. Оголошення розміщуються на релевантних інформаційних ресурсах на основі ключових слів. Тому для всіх ключових слів встановлюється лише широка відповідність. Щоб підвищити ефективність ключових слів, деякі з них можна вилучити з груп оголошень. Наприклад, для списку ключових слів з термінами, пов'язаними із взуттям, націлювання на інформаційні ресурси про взуття здійснюватиметься за ключовими словами у цьому списку. Окрім того, можна вилучити терміни *лижі* та *сноуборд*, щоб ваші оголошення не з'являлися на сайтах про зимові види спорту.

3. Автоматичне формування множини ключових слів за допомогою алгоритму стемінгу. Існує кілька варіантів алгоритмів стемінгу, які відрізняються точністю та продуктивністю: пошук за таблицею, відсікання флексій та суфіксів, лематизація, стохастичні алгоритми, гібридний підхід, відсікання префіксів, пошук відповідності, стемінг українською [11–18, 22–30]. У табл. 1 здійснено порівняльний аналіз особливостей, переваг та недоліків відомих алгоритмів стемінгу.

*Таблиця 1*

**Основні алгоритми стемінгу**

| Назва | Особливість | Приклад | Переваги | Недоліки |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| Пошук за таблицею | У таблиці зібрані всі можливі варіанти слів та їх форми після стемінгу | Stemming={*інформац*} → Word={*інформаційний*, *інформаційна*, *інформаційне*, *інформаційним*, *інформаційними*, *інформаційних*, *інформаційні*, *інформаційній*, *інформаційнім*, *інформаційного*, *безпритульної*, *інформаційному*, *інформаційною*, *інформаційну*} | Простота, швидкість та зручність опрацювання винятків з мовних правил. Для мов із простою морфологією (англійська) таблиці малі | Таблиця пошуку має містити всі форми слів. Алгоритм не працює з новими словами. Великі розміри таблиці для мов із складною морфологією (аглютинативні, слов'янські) |
| Відсікання флексій та суфіксів | Ґрунтуються на правилах скорочення слова Rules = { Ending (*ційна*) → Cut (*ійна*); Ending (*ційне*) → Cut (*ійне*); Ending (*ційний*) → Cut (*ійний*); Ending (*ційним*) → Cut (*ійним*); Ending (*льне*) → Cut (*ьне*)} | Word={*інформаційний*} → Stemming={*інформац*};<br><br>Word={*цивілізаційний*} → Stemming={*цивілізац*};<br><br>Word={*приватизаційний*} → Stemming={*приватизац*};<br><br>Word={*кульмінаційний*} → Stemming={*кульмінац*};<br>Word={*національне*} → Stemming={*національ*}. | Алгоритм доволі компактний та продуктивний, оскільки кількість правил набагато менша за таблицю з усіма словоформами | Наявні хибні висновки і спотворені форми стемінгу (*пальне* стане *пал* замість *пальн*). Через особливості мови набір правил є складним. Передбачене опрацювання винятків, коли базові слова мають змінну форму (*бігом* та *біжу* повинні мати *біг*, але простим відсіканням це неможливо отримати). Це призводить до ускладнення правил і негативно впливає на ефективність |

*Продовження табл. 1*

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Лематизація | *Step* 1. Визначення частин мови у реченні (POS tagging).<br>*Step* 2. До слова застосовують правила стемінгу відповідно до частини мови | Слова *пальне* (іменник) та *вітальне* (прикметник) проходять через різні ланцюжки правил: Rules = { Ending (*льне*) → Cut (*е*); Ending (*льне*) → Cut (*ьне*)} | Алгоритми мають високу якість і мінімальний відсоток помилок | Алгоритми залежні від правильності розпізнавання частин мови |
| Стохастичні алгоритми | Ґрунтуються на ймовірності визначення основи слова з використанням бази знань. Лематизація має стохастичні властивості, коли частину мови визначають без урахування контексту, в якому це слово було вжито в реченні. Перевага віддається найвірогіднішій частині мови для цього слова | Word={*особистість*}→ Stemming={*особист*}→ End={*ість*}; Word={*спогади*}→ Stemming={*спогад*}→ End={*и*}; Word={*дивними*}→ Stemming={*див*н}→ End={*ими*}, де End – результат навчання алгоритму, тобто Word(*кияни*) → {End(*ість*) = FALSE, End(*и*) = TRUE, End(*ими*) = FALSE}→ Cut (*и*) або Word(*чуйними*) → {End (*ість*) = FALSE, End (*и*) = TRUE, End (*ими*) = TRUE}→ Cut (*и*) OR Cut (*ими*) | Є лише одне логічне правило, за яким від слова відсікаємо останні літери. Алгоритми мають здатність навчатися і чим краща та більша база навчання, тим кращий результат їх роботи. База знань для цих алгоритмів – це набір логічних правил та таблиці пошуку | Після опрацювання слова може з'явитися декілька варіантів основи слова, з яких алгоритм вибере найімовірніший варіант (надати перевагу стемінгу, який скорочує слово – найбільше чи найменше). І як результат – ймовірність помилок стемінгу зростає |
| Гібридний підхід | Використовують комбінацію наведених вище алгоритмів | Наприклад, алгоритм може використовувати метод відсікання закінчень та суфіксів, але на першому етапі виконувати пошук у таблиці | Таблиця містить не всі словоформи, а винятки з правил, які неправильно опрацьовуються алгоритмом відсікання | Ймовірність помилок стемінгу зростає у разі некоректного опису правил та формування таблиці закінчень |
| Відсікання префіксів | Відсікання від слів суфіксів та закінчень, а також за наявності префіксів | Word={*проголошую*, *наголошувати*, *виголошував*} → Stemming={*голошу*} | В [21] детально обґрунтовано важливість такого стемінгу для деяких мов | Ймовірність утворення протилежних за змістом слів, тобто Word={*незалежний*} → Stemming={*залежн*} |
| Пошук відповідності | Використовують базу знань лише з основами слів після стемінгу | KnowledgeBase={*чорн*, *чорняв*} → Word={*чорнява*} → Count={4, 6}→ Stemming={*чорнява*}. Алгоритм вибере довший варіант | Через систему правил (довжина збігу слова та його основи) пошук для найвідповіднішої форми з бази знань | Ймовірність помилок стемінгу зростає у разі некоректного опису правил та формування таблиці закінчень |
| Стемінг різними мовами | Орієнтація на конкурентну мову | Якщо стемінг англійської – це проста задача, то стемінг для арабської чи івриту – на порядок складніша | Перші академічні роботи зі стемінгу присвячені лише англійській, але вже існує багато реалізацій для інших мов | Від особливостей мови залежить складність написання алгоритмів стемінгу |
| Стемінг українською | Варіанти стемінгу для української мови існують [1–2] і використовуються у складі комерційних пошукових систем | - | Певні кроки у цьому напрямі вже зроблені в [6, 11–19] і поява некомерційного алгоритму стемінгу для української є справою часу | Поки що відсутня вільна реалізація таких алгоритмів |

У алгоритмах стемінгу поширені дві типові помилки – надстемінг (англ. *overstemming*) та недостемінг (англ. *understemming*). *Надстемінг* полягає у скороченні двох різних слів до однієї основи. *Недостемінг* полягає в отриманні різних основ для двох однозначних слів з однією спільною основою. Алгоритми стемінгу намагаються мінімізувати такі помилки, проте зменшення помилок одного типу може призвести до зростання кількості помилок іншого.

**Виділення проблем**

Аналіз динаміки потоку контенту та побудова етапів опрацювання інформаційних ресурсів важливі та актуальні. Ефективне розроблення і впровадження тематичних інформаційних ресурсів сьогодні неможливі без коректного визначення множини ключових слів. Крім того, вони мають

бути вживані в текстовому масиві контенту не аби як, а за певним розподілом, щоб пошукові системи не індексували і не класифікували це як спам. Розроблення методу опрацювання україномовної текстової інформації для автоматичного виявлення значущих ключових слів та рубрикації контенту є одним зі стратегічних напрямів розвитку вітчизняного е-бізнесу.

Забезпечення можливості автоматизації процесів опрацювання інформаційних ресурсів для виявлення значущих ключових слів та рубрикації контенту сприяє збільшенню обсягів продажу контенту, товару або послуг постійному користувачу, активному залученню потенційних користувачів та розширенню меж цільової аудиторії. Зокрема, ці принципи і технології в е-комерції активно застосовують для створення систем on-line/off-line продажу та аналізу/обміну/збереження контенту, інтернет-магазину, cloud storage/computing. Відсутність загального стандартизованого підходу до опрацювання української текстової інформації з метою автоматичного виявлення значущих ключових слів та рубрикації контенту, а також проектування функціонала систем підтримки е-комерції призводить до виникнення проблем під час реалізації типової структури таких систем.

Пошук ключовиків, перевірка їх мови та запис їх у відповідну таблицю найчастіше здійснюють так:

1) перегнати слово через словники простим циклом, основний недолік – процес дуже довго працюватиме і, вірогідно, зависатиме, що некоректно для on-line систем;

2) написати правило для мови, якщо є кириличні букви: є, Ї, і, і для певних буквосполучень;

3) якщо тільки українська і англійська, то дізнатись можна через те, що один текст поданий кирилицею, а інший – латиницею;

4) класом PHPLangautodetect можна визначати мову ключового слова;

5) через Google API, але сервіс поки що платний;

6) за допомогою pear-пакета до php: Text_LanguageDetect.

Аналіз та опрацювання текстових масивів даних, контент-аналіз, контентний пошук, SEO, визначення дублів, рубрикація контенту є сьогодні популярними напрямами досліджень. І в кожному з цих напрямів необхідно використовувати ключовики та автоматично визначати мову текстів, що досліджуються. Використовують для цих задач відстань Левенште́йна (також функція Левенштейна, алгоритм Левенштейна, або відстань редагування) – у теорії інформації та комп'ютерній лінгвістиці це міра відмінності двох послідовностей символів (рядків). Обчислюється як мінімальна кількість операцій вставлення, видалення і заміни, необхідних для перетворення однієї послідовності на іншу. Наприклад, функція (<? echo did_you_mean(текст); ?>) виправляє слова за методом Левенштейна [3–5], де DB_MATRIX – змінна таблиця, в якій зберігаються ключові слова, що вирізані за допомогою функції:

```
function did_you_mean ($stext) {  global $locale, $querys;  $i = 0;
 $counter = 0;  $buf = “Ви шукали: “;
 foreach ($querys as $product => $word) { $word = stripinput($querys[$product]);
 $result = dbquery(“SELECT word FROM “.DB_MATRIX.” WHERE SOUNDEX(word) =
SOUNDEX('$word')");
 $max_distance = 100; $near_word ="";
 while ($row = dbarray($result)) {
$distance = levenshtein($row['word'], $word);
 if ($distance < $max_distance && $distance <4) {
 $max_distance = $distance;
 $near_word = $row['word']; } }
 if (!empty($near_word) && $word != $near_word){//$near_words[$i] = $near_word;
if ($counter == 0) {
 $buf .= “<a href='”.BSFURL.“index.php?text=".$near_word.“'>".$near_word."</a>";
 $counter = 1; } else {
 $buf=",<a href='”.BSFURL.“index.php?text=".$near_word.“'>".$near_word."</a>"; }
 $i++;  }  }
 $buf .= “<br><br>";
 if ($i == 0) $buf = “";
return $buf;}
```

Нечіткий пошук є вкрай корисною функцією будь-якої пошукової системи. Разом з тим, його ефективна реалізація набагато складніша за реалізацію простого пошуку за точним збігом. Задачу нечіткого пошуку можна сформулювати так: “За заданим словом знайти в тексті або словнику розміру *n* всі слова, що збігаються з цим словом (або починаються з цього слова) з урахуванням *k* можливих відмінностей”. Наприклад, за запитом “Машина” з урахуванням двох можливих помилок знайти слова “Машинка”, “Махіна”, “Малина”, “Калина” і так далі [4, 8–10]. Також це можна використовувати для рубрикації, виявлення дублювання та формування дайджестів [3, 5], наприклад, алгоритм пошуку ключовиків:

```
function keywords($story) {$keyword_count = 5000; // число ключовиків
 $newarr = array ();
 $quotes = array ("\x22”, “\x60”, “\t”, “\n”, “\r”, “,", “.", “/", “¬", “#", “;", “:",
“@", “~", “[", “]", “{", “}", “=", “-", “+", “)", “(", “*", “^", “%", “$", “<", “>",
“?", “!", '“');
 $fastquotes = array ("\x22”, “\x60”, “\t”, “\n”, “\r”, '“', “\\", '\r', '\n', “/",
“{", “}", “[", “]” );
 $story = str_replace( “ ", “ “, $story );
 $story = strip_tags( $story );
 $story = preg_replace( “#&(.+?);#", “", $story );
 $story = trim(str_replace( “ ,", “", stripslashes( $story )));
 $story = str_replace( $fastquotes, '', $story );
 $metatags['description'] = substr( $story, 0, 190, “utf-8” );
 $story = str_replace( $quotes, ' ', $story );
 $arr = explode( “ “, $story );
 foreach ( $arr as $word ) {
 if( strlen( $word) > 3 ) $newarr[] = strtolower($word); }
 $arr = array_count_values( $newarr ); arsort( $arr );
 $arr = array_keys( $arr ); $total = count( $arr ); $offset = 0;
 $arr = array_slice( $arr, $offset, $keyword_count );
 return implode( “, “, $arr );}
```

Отже, існують відомі алгоритми парсингу, стемінгу та виявлення ключових слів для англомовних текстів, але вони некоректно опрацьовують тексти україномовного контенту. Тому є необхідність в адаптуванні алгоритмів парсингу та стемінгу до текстів українською мовою, причому найперспективнішим, за даними порівняльного аналізу, може бути пошук відповідності у поєднанні зі стохастичними алгоритмами. Корисним інструментом у процесі пошуку ключовиків видається також використання тематичних словників основ слів з паралельною рубрикацією контенту.

**Формулювання мети та ідеї дослідження**

Метою роботи є визначення оптимального методу автоматичного опрацювання множини україномовного текстового контенту для виявлення значущих ключових слів та автоматичної рубрикації контенту.

Опрацювання множини контенту *C* для виявлення значущих ключових слів зазвичай побудовано на принципі знаходження ключових слів за змістом (термами), що ґрунтується на законі Зіпфа і зводиться до вибору слів із середньою частотою появи. Оскільки це прямий та найпростіший спосіб, пропонується застосовувати складніші та виконати відповідне експериментальне дослідження. Експериментальною базою для такого дослідження вибрано 100 наукових публікацій Вісника Національного університету “Львівська політехніка” серії “Інформаційні системи та мережі” (http://science.lp.edu.ua/sisn), двох номерів 783 (http://science.lp.edu.ua/SISN/SISN-2014) та 805 (http://science.lp.edu.ua/sisn/vol-cur-805-2014-2).

Оскільки необхідно не лише організувати пошук ключовиків з множини наперед визначених автором або модератором та відомих системі, ідея дослідження полягає в автоматичному визначенні з текстового масиву даних множини таких слів, які є потенційними ключовиками (відповідають певним умовам та вимогам). Тоді головним критерієм якості визначення ключових слів тексту є потужність множини-перетину множин ключовиків – заданих автором та визначених автоматично. Хоча варто застосовувати мовнонезалежний алгоритм парсеру, алгоритм стемінгу

обов'язково має бути прив'язаний до української мови. Отже, необхідно адаптувати алгоритми парсеру та стемінгу до української мови, використовуючи тематичні словники основ слів шляхом:

1) спочатку за допомогою елементарного алгоритму парсеру (універсально – мова може бути будь-якою) визначається множина слів з тексту, що мають певну частоту появи та потрапляють в певні значущі межі, наприклад, 4–6 відсотків;

2) за допомогою алгоритмів парсеру та стемінгу визначають підмножину значущих слів, блокуючи слова, які вніс модератор до словника заблокованих, наприклад, такі як дієслова, займенники, прийменники, сполучники, частки тощо;

3) нова підмножина порівнюється із вмістом тематичного словника для формування множини ключових слів (словник наперед складено за темами – кожному слову відповідає індикатор рубрики); особливість полягає в тому, що тематичний словник – це словник основ слів для української мови (для англійської мови достатньо простого словника тематичних слів – слова там не відмінюються, проте в слов'янських мовах є надлишковість інформації через відмінювання слів, наявність префіксів, суфіксів та закінчень слів);

4) далі накопичується статистика для різних текстів (художніх, наукових, поетичних, публіцистичних тощо) для формування підмножин ключових слів, які будуть використані для процесу рубрикації текстових масивів даних;

5) порівнюється ефективність розробленого алгоритму, адаптованого для україномовних текстових масивів даних, з відомими за різними категоріями текстів.

## Аналіз отриманих наукових результатів

Для досягнення мети дослідження розроблено систему (рис. 1), що дає змогу вибрати мову або мови, з яких складений текст. Доступ до процесу знаходження множини ключових слів з урахуванням основ тематичних слів можна отримати на інформаційному ресурсі Victana за адресою http://victana.lviv.ua/index.php/kliuchovi-slova. Розроблена інформаційна система знаходження множини ключових слів з урахуванням основ тематичних слів побудована з використанням таких засобів:

1. CMS Joomla! версії 3.4.4 для розроблення е-каркасу інформаційного ресурсу Victana.
2. PHP версії 5.3.20 для реалізації алгоритму знаходження множини ключових слів з урахуванням основ тематичних слів.
3. HTML версії 4.01 для реалізації розмітки Web-сторінок.
4. CSS для опису стилів сторінок.
5. MySQL версії 5.1.63 для зберігання даних (словників).

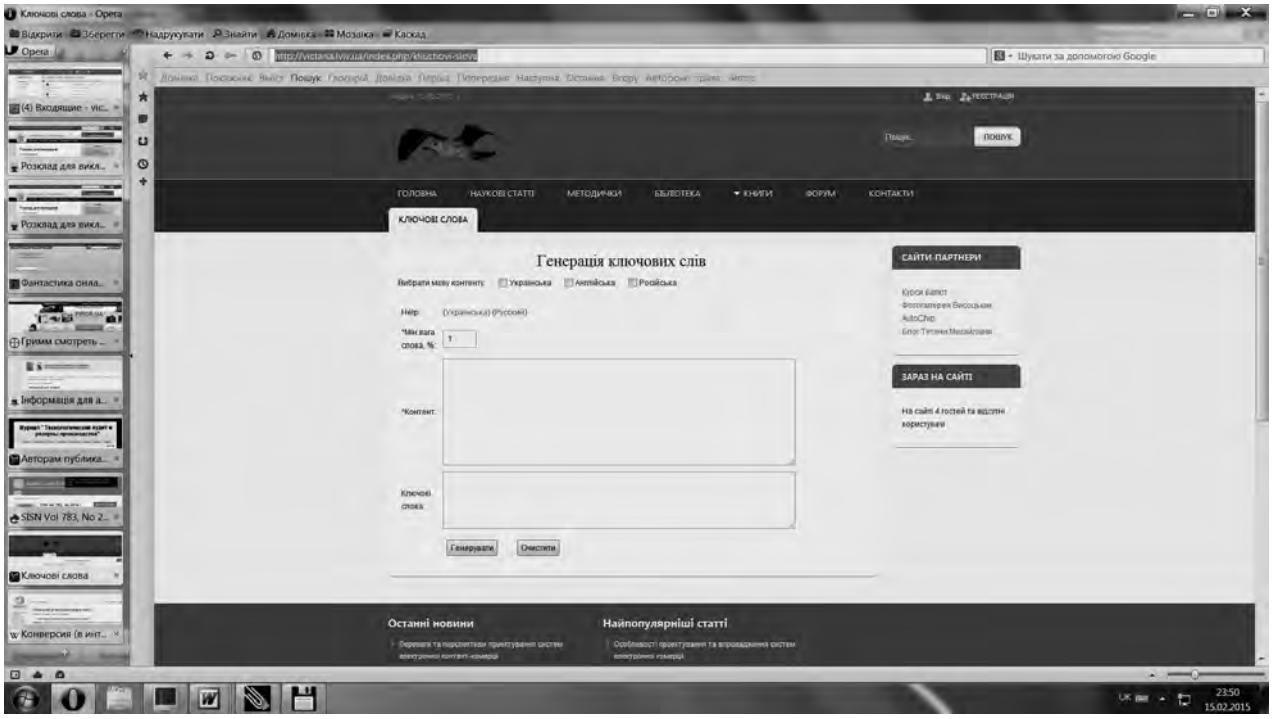



*Рис. 1. Інформаційний ресурс визначення ключових слів з тексту*

Розроблена інформаційна система має такі основні компоненти.

1. Інтерфейс – діалоговий, дружній, користувацький. Кліком мишкою по пункту меню “Ключові слова” переходимо на сторінку “Генерація ключових слів”. Сторінка має розділи (рис. 2):

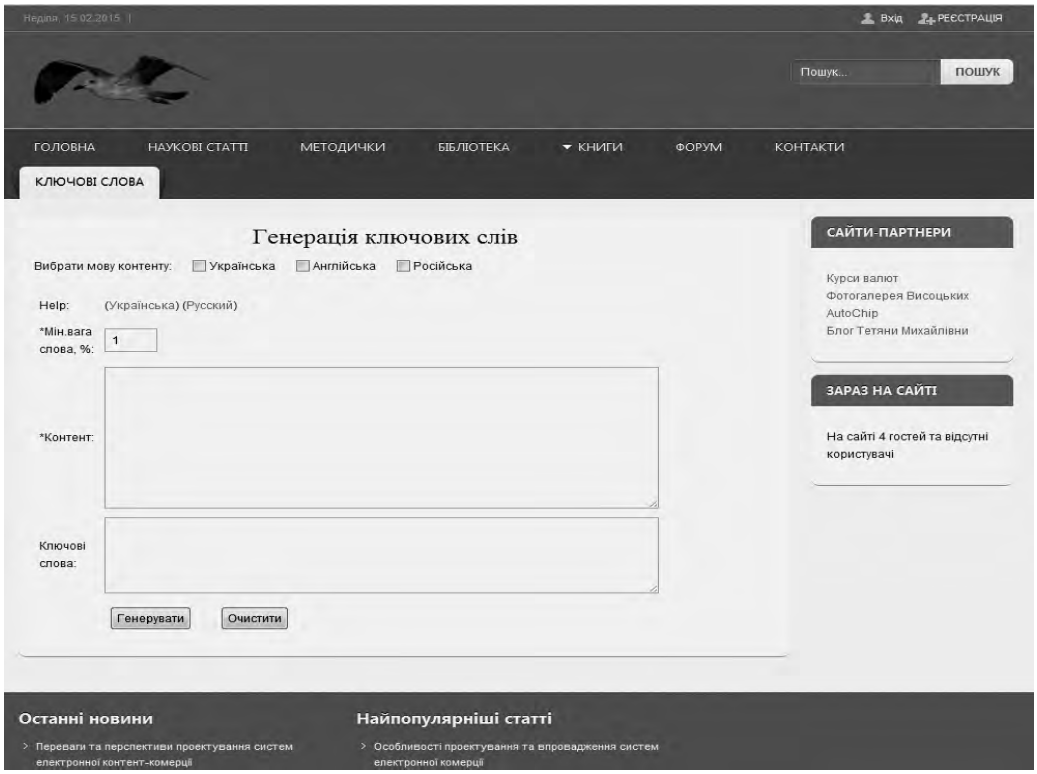



*Рис. 2. Діалогове вікно системи визначення множини ключових слів у текстовому контенті*

– “Вибрати мову контенту” – вибрати одну або декілька мов, якою (якими) написано вхідний текст, що обробляється.

– “Help” – коротка інструкція українською та російською мовами. Текст інструкції відкривається в окремому вікні.

– “Мін. вага слова, %” – поле обов'язкове для заповнення. Формат – XX.XX, значення – 00.01 – 99.99. Відсоток ваги ключового слова до загальної кількості слів тексту, після якого будуть вибиратись ключові слова.

– “Контент” – поле, в яке вставляється текст (стаття) у текстовому форматі.

– “Ключові слова” – у це поле будуть виведені ключові слова після кліку мишкою на кнопці “Генерувати”.

– “Генерувати” – запускає процес генерації ключових слів у разі заповнення обов'язкових полів.

– “Очистити” – очищає поля введення.

– “Повторюваність слів, раз” – кількість повторювань ключового слова в тексті.

– “Рекомендовані рубрики” – назва говорить сама за себе.

2. База даних (БД) – одна. Таблиці (словники) такі:

– основ слів (ключових слів) – для накопичення та зберігання ключових слів;

– заборонених слів – для накопичення та зберігання заборонених слів;

– рубрик – для накопичення та зберігання рубрик;

– правил приведення до основи слова – для накопичення та зберігання правил.

Всі таблиці редагуються з адміністративної частини. Вони ті самі для всіх мов. Віднесення до мови визначається в таблиці.

3. Функції опрацювання тексту написані на PHP:

– функція blocked_words() – формує список заблокованих слів залежно від вибраної мови контексту;

– функція explode_str_on_words() – очищає отриманий контент від заблокованих слів, спецсимволів тощо;

– функція count_words() – підрахування частоти ключових слів;

– функція get_keywords() – формування списку ключових слів;

– функція get_word() – запису правил приведення до основи слова;

– функція set_keywords() – запису ключових слів до БД, якщо вони там відсутні;

– функція function error() – оброблення помилок, надсилання листа адміністратору системи;

– функція recommend_rubric() – формування списку рекомендованих рубрик.

4. Інформаційний ресурс у вигляді веб-сайта – HTML. Сторінка динамічно змінюється залежно від введених користувачем даних та результату їх обробки, частина яких береться із БД (приведені до основи ключові слова, рубрики).

5. CSS використовують для визначення кольору, шрифту, верстання та інших аспектів вигляду сторінки.

Після запуску ресурсу в полі **Мін.вага слова, %* задається ціле число – відсоток, який має перевищувати вживане ключове слово в тексті. В поле **Контент* копіюється текст, який необхідно дослідити. Після натискання кнопки *Генерувати* в полі *Ключові слова* буде розміщена множина визначених ключових слів. Кнопка *Очистити* необхідна для очищення поля **Контент*. Окрім того, після генерування множини контенту внизу під полем *Ключові слова* з'являються дані *Повторюваність слів раз* – перелік знайдених ключовиків з числовими значеннями (кількістю вживання цих слів у тексті).

Аналіз статистики функціонування системи виявлення множини ключових слів зі 100 наукових статей технічного спрямування здійснено за два етапи.

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

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

Окрім того, на кожному етапі роботи системи перевірка відбувалась за два кроки для кожної статті: аналіз всієї статті (рис. 3, *а*) та аналіз статті без початку (назва, автори, УДК, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури (рис. 3, *б*). Такий підхід застосовували для того, щоб визначити похибки точності формування множини ключових слів для різних модифікацій запропонованого методу.

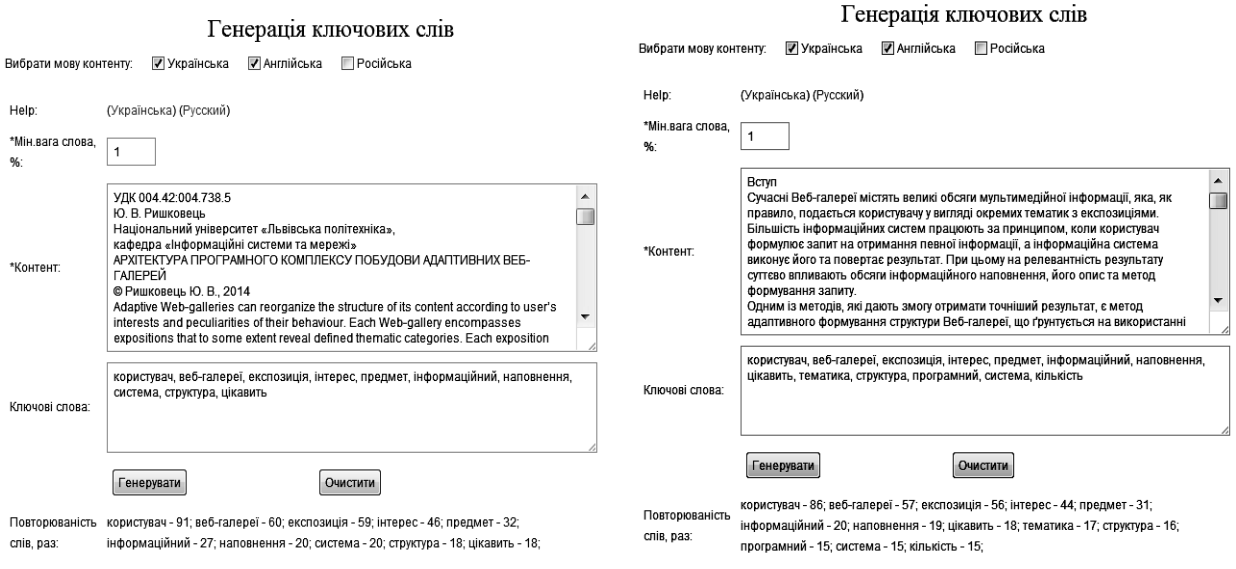



*Рис. 3. Приклади результатів перевірки статті*

Аналіз статистики здійснювався за принципом порівняння множини авторських ключових слів (визначені та прописані в статті самими авторами цих робіт), множини ключових слів, визначених за першим та другим етапами з різними вагами слів (але більше за визначене в опції **Мін.вага слова, %* в межах [1, 5]) з повними та скороченими текстами робіт (табл. 2) за середнього арифметичного значення авторських ключових словосполучень / слів близько 5 (4,77), які в середньому утворені з 10 (9,82) слів. Вага слова розраховується як відносна частота появи основи цього слова у всьому тексті. В табл. 3 позначення такі: *A* (всього ключових слів, визначених системою за заданої ваги слова), *B* (змістовних слів зі списку утворених, тобто без невідомих абревіатур, дієслів, службових слів тощо), *C* (збіг слів з визначеними автором статті), *D* (точність збігу знайдених ключовиків з авторськими ключовими словами), *E* (додаткові ключові слова, визначені системою, але не визначені автором статті).

*Таблиця 2*

**Статистичні дані досліджених обсягів текстів статей**

| Назва частини статті | Крок 1 | | Крок 2 | |
|---|---|---|---|---|
| | Разом | Середнє арифметичне | Разом | Середнє арифметичне |
| Сторінок | 956 | 9,56 | 828 | 8,28 |
| Абзаців | 16497 | 164,97 | 15263 | 152,63 |
| Рядків | 42553 | 425,53 | 36965 | 369,65 |
| Слів | 345580 | 3455,8 | 291247 | 2912,47 |
| Знаків | 2327209 | 23272,09 | 1974773 | 19747,73 |
| Знаків та пробілів | 2674889 | 26748,89 | 2265917 | 22659,17 |

*Таблиця 3*

**Статистичні дані дослідження змісту текстів статей**

| Назва | Вага слова | Етап 1 | | | | | Етап 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *A* | *B* | *C* | *D* | *E* | *A* | *B* | *C* | *D* | *E* |
| Крок 1 | ≥ 1 | 5,46 | 3,92 | 2,51 | 2,08 | 1,74 | 7,43 | 7,03 | 3,27 | 3 | 4,18 |
| | ≥ 2 | 1,08 | 0,88 | 0,63 | 0,59 | 0,26 | 2,67 | 2,64 | 1,65 | 1,54 | 1,12 |
| | ≥ 3 | 0,41 | 0,38 | 0,22 | 0,21 | 0,16 | 1,21 | 1,2 | 0,85 | 0,79 | 0,41 |
| | ≥ 4 | 0,15 | 0,13 | 0,09 | 0,09 | 0,04 | 0,46 | 0,45 | 0,33 | 0,31 | 0,15 |
| | ≥ 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Крок 2 | ≥ 1 | 6,51 | 5,02 | 2,68 | 2,23 | 2,37 | 8,35 | 7,78 | 3,25 | 2,91 | 4,99 |
| | ≥ 2 | 1,34 | 1,11 | 0,74 | 0,72 | 0,39 | 3,12 | 3,07 | 1,81 | 1,67 | 1,43 |
| | ≥ 3 | 0,51 | 0,45 | 0,29 | 0,27 | 0,17 | 1,42 | 1,4 | 0,93 | 0,85 | 0,54 |
| | ≥ 4 | 0,19 | 0,17 | 0,12 | 0,12 | 0,05 | 0,73 | 0,72 | 0,45 | 0,42 | 0,31 |
| | ≥ 5 | 0,11 | 0,1 | 0,06 | 0,06 | 0,04 | 0,33 | 0,32 | 0,25 | 0,23 | 0,1 |

На рис. 4–9 подані діаграми аналізу статистики формування системою множин ключових слів.

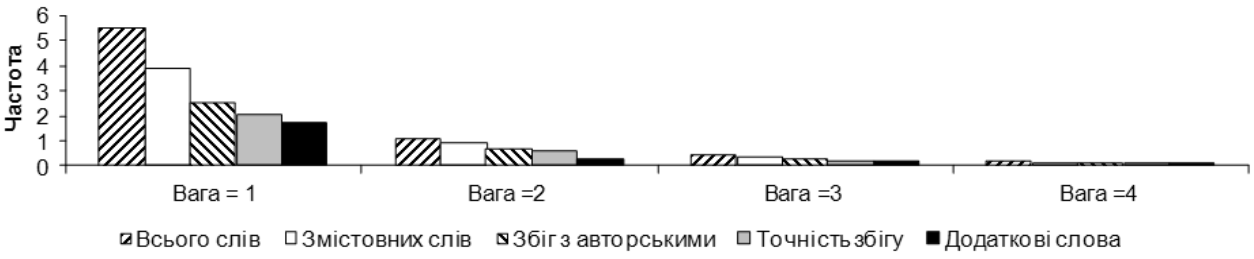



*Рис. 4. Отримання значущих слів під час опрацювання тексту на етапі 1, крок 1*

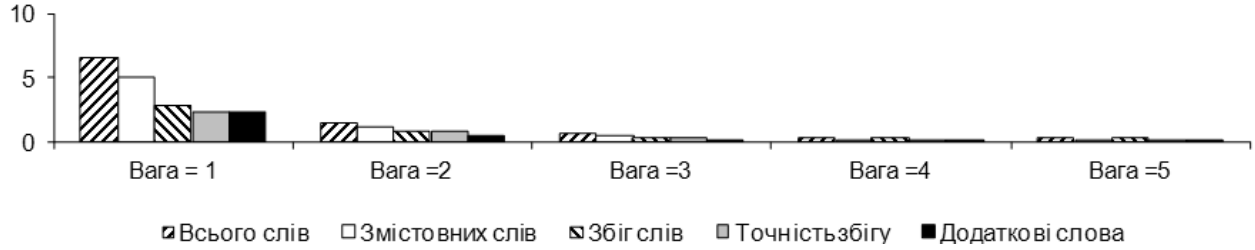



*Рис. 5. Отримання значущих слів під час опрацювання тексту на етапі 1, крок 2*


*Рис. 6. Отримання значущих слів під час опрацювання тексту на етапі 2, крок 1*

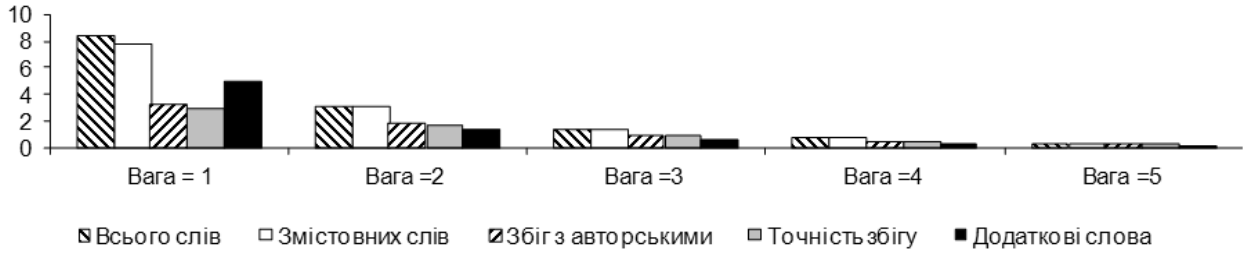
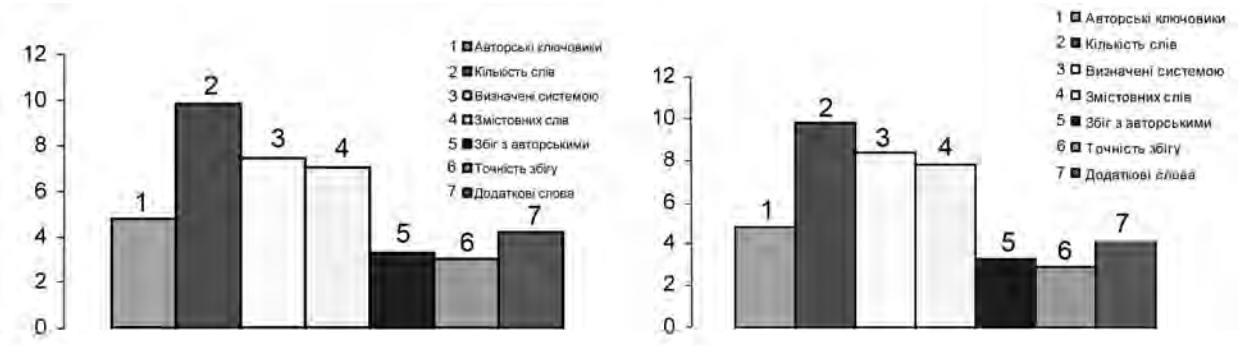



*Рис. 7. Отримання значущих слів під час опрацювання тексту на етапі 2, крок 2*


*а* *б*

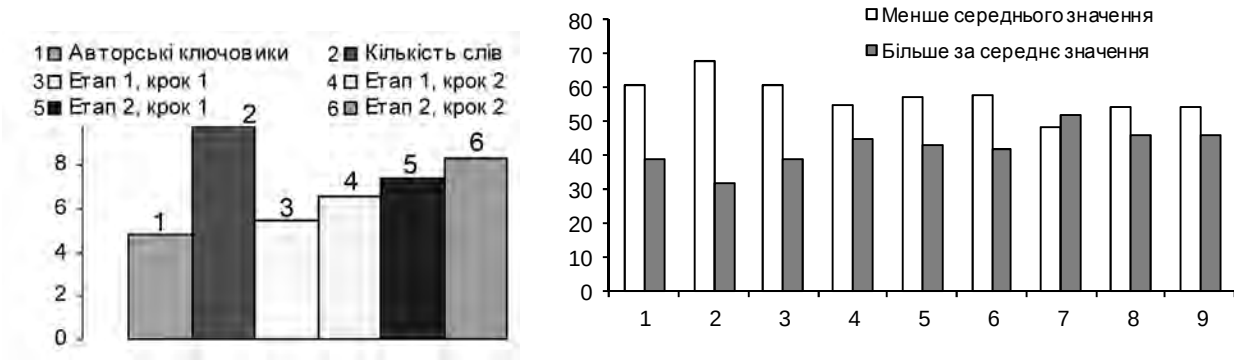
*Рис. 8. Середньоарифметична поява значущих слів у тексті порівняно з авторськими для етапу 1: а – крок 1; б – крок 2*




*а* *б*

*Рис. 9. Середньоарифметична поява значущих слів у тексті порівняно з авторськими для етапу 2: а – крок 1; б – крок 2*

На рис. 10, *а* подана діаграма аналізу статистики формування системою множин всіх потенційних ключових слів порівняно з множиною, яку визначили автори статей. Перший стовпчик – середньоарифметична кількість ключових слів, визначених автором, а другий – середньоарифметична кількість слів, що становлять ці авторські ключові слова. Третій стовпчик – середньоарифметична кількість потенційних ключових слів, визначена системно на етапі 1, крок 1; четвертий – на етапі 1, крок 2; п'ятий – на етапі 1, крок 1; шостий – на етапі 2, крок 2.




*а* *б*

*Рис. 10. Результати перевірки 100 статей*

На рис. 10, *б* подана діаграма аналізу статистики розподілу щільності тексту в аналізованих статтях, де 1 – аналіз кількості сторінок статей відповідно менша та більша за середнє значення, 2 – абзаців у 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. 11 подана діаграма розподілу формування системою множин всіх потенційних ключових слів для кожної статті порівняно з множиною, яку визначили автори статей.

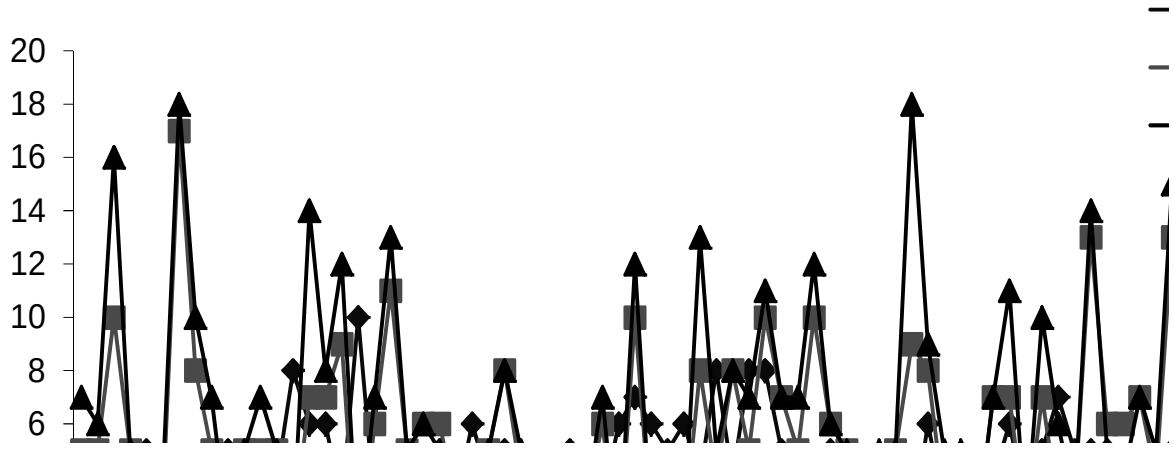


*Рис. 11. Результати перевірки 100 статей*

У табл. 4 подано результати аналізу статистики формування системою множин всіх потенційних ключових слів для кожної статті порівняно з множиною, яку визначили автори 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).

*Таблиця 4*

**Описові статистичні дані формування ключових слів для досліджених змістів текстів статей**

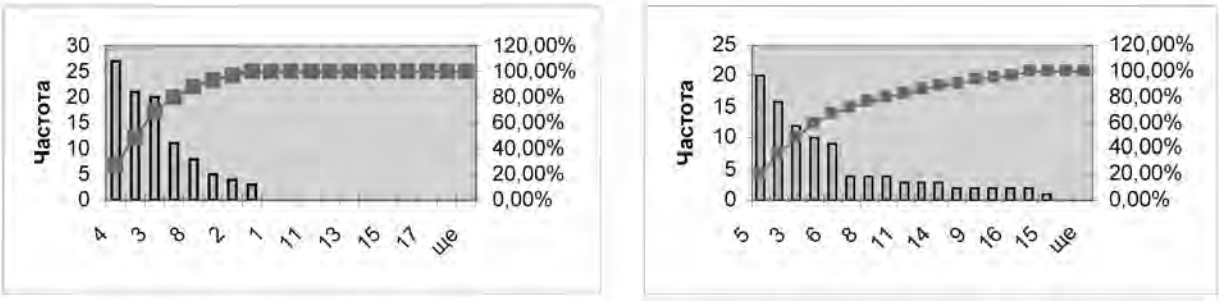
| | А | Б | В | Г | Д |
|---|---|---|---|---|---|
| Середнє | 4,808081 | 5,515152 | 6,565657 | 7,505051 | 8,434343 |
| Стандартна помилка | 0,180859 | 0,310393 | 0,39035 | 0,301297 | 0,324611 |
| Медіана | 4 | 5 | 6 | 7 | 8 |
| Мода | 4 | 5 | 5 | 7 | 8 |
| Стандартне відхилення | 1,799528 | 3,088371 | 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 |

У табл. 5, 6 подані статистичні дані аналізу текстів статей для формування множин ключових слів та побудови відповідних гістограм для груп А-Д (рис. 12–13).

*Таблиця 5*

**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% |




*а* *б*

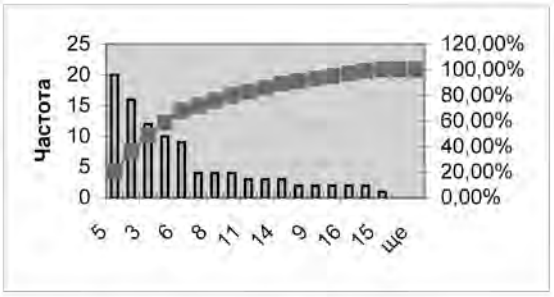
*Рис. 12. Гістограма для:*
*а – вибірки А; б – вибірки Б*

*Таблиця 6*

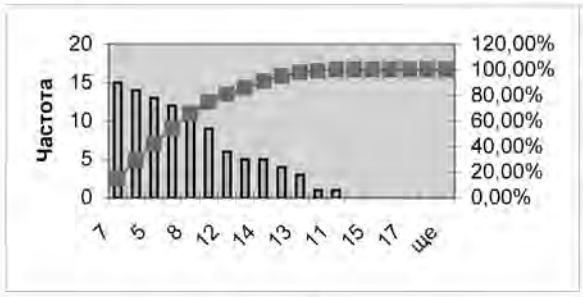
**Статистичні дані побудови гістограми для групи В, групи С та групи Д**

| 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% |

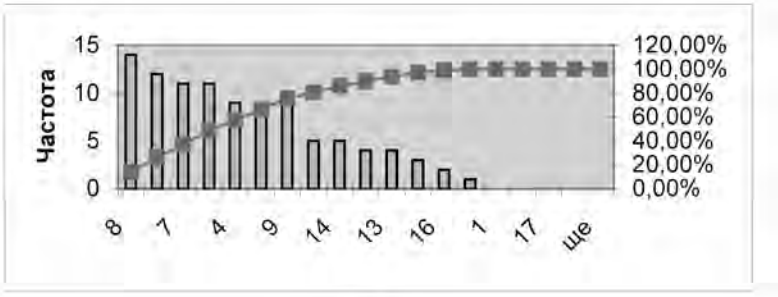
| Д | Частота | Інтегральний % | Д | Частота | Інтегральний % |
|---|---|---|---|---|---|
| 1 | 0 | 0,00% | 8 | 14 | 14,14% |
| 2 | 0 | 0,00% | 5 | 12 | 26,26% |
| 3 | 1 | 1,01% | 7 | 11 | 37,37% |
| 4 | 9 | 10,10% | 10 | 11 | 48,48% |
| 5 | 12 | 22,22% | 4 | 9 | 57,58% |
| 6 | 9 | 31,31% | 6 | 9 | 66,67% |
| 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% |




*а*




*б*




*в*

*Рис. 13. Гістограма для:*
*а – вибірки В; б – вибірки Г; в – вибірки Д*

**Висновки та перспективи подальших наукових розвідок**

У статті проведено експериментальне дослідження методів автоматичного виявлення значущих ключових слів україномовного контенту, побудованих на основі стемінгу Потера за відстанню Левенштейна. Експериментальною базою для такого дослідження вибрано 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). На вибраній експериментальній базі за допомогою реалізованого алгоритму стемінгу Портера на інформаційному ресурсі Victana.lviv.ua отримано статистичні характеристики чотирьох методів:

– аналіз всієї статті із перевіркою загальних заблокованих слів та тематичного словника;

– аналіз статті без початку (назва, автори, УДК, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури із перевіркою загальних заблокованих слів та тематичного словника;

– аналіз всієї статті із перевіркою уточнених заблокованих слів та уточненого тематичного словника (з більшою кількістю запуску системи формується множина невідомих слів, яких немає як у тематичному словнику, так і в множині заблокованих);

– аналіз статті без початку (назва, автори, УДК, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури із перевіркою уточнених заблокованих слів та уточненого тематичного словника.

Виявлено, що для технічних наукових текстів експериментальної бази найкращих результатів досягає четвертий метод аналізу статті (без початку і без списку літератури із перевіркою уточнених заблокованих слів та уточненого тематичного словника). Цей метод визначення ключових слів точніший (за переважною більшістю числових показників) та коректний (знайдені ключові слова точніше описують предметну область статті та визначають рубрику цієї роботи).

Потребує подальшого експериментального дослідження пошук ключових слів у інших категоріях текстів – художніх, публіцистичних, наукових, гуманітарних тощо.

*1. Вероятностный морфологический анализатор русского и украинского языков. – Режим доступу: http://www.keva.ru/stemka/stemka.html. – Назва з титул. екрана. 2. Вірогідний морфологічний аналізатор російської та української. – Режим доступу: http://www.keva.ru/stemka/ stemka.html. – Назва з титул. екрана. 3. Вычисление расстояния Левенштейна между двумя строками. – Режим доступу: http://wm-help.net/lib/b/book/827961078/78. – Назва з титул. екрана. 4. Задача о расстоянии Дамерау-Левенштейна / 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. Левенштейн, который сравнивает строки / Веб-разработка. – Режим доступу: http://dayte2.com/levenshtein. – Назва з титул. екрана. 6. Модуль Drupal для стемінга українською. Новий модуль для алгоритму Стема для українського пошуку з виділенням коренів / Режим доступу: http://drupal.ua/node/1170. – Назва з титул. екрана. 7. Найефективніші методи залучення потенційних клієнтів / 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. 8. Насонов Д. Функция Левенштейна / Д. Насонов. – Режим доступу: http://rain.ifmo.ru/cat/data/theory/unsorted/levenshtein-2006/article.pdf. – Назва з титул. екрана. 9. Нечёткий поиск в тексте и словаре. – Режим доступу: http://habrahabr.ru/post/114997/. – Назва з титул. екрана. 10. Реализации алгоритмов. Расстояние Левенштейна. – Режим доступу: http://ru.wikibooks.org/wiki/Реализации_алгоритмов/ Расстояние_ Левенштейна. – Назва з титул. екрана. 11. Сеник М. Вільний алгоритм стемінгу для української мови / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ ukr_stemming/stemming_about. html. – Назва з титул. екрана. 12. Сеник М. Інструмент для пошуку слів з однаковими закінченнями /*

*М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ ukr_stemming/word_by_ending.html. – Назва з титул. екрана. 13. Сеник М. Статичне дерево закінчень / М. Сеник. – Режим доступу: http://www.senyk.poltava.ua/projects/ukr_stemming/ ukr_endings.html#dyn. – Назва з титул. екрана. 14. Сеник М. Демо стемінгу для української мови / М. Сеник. – Режим доступу: http://www.senyk. poltava.ua/projects/ukr_stemming/demo.html. – Назва з титул. екрана. 15. Стемінг. – Режим доступу: https://uk.wikipedia.org/wiki/Стемінг. – Назва з титул. екрана. 16. Стемінг Портера для української мови. – Режим доступу: http://www.marazm.org.ua/document/stemer_ua/. – Назва з титул. екрана. 17. Стемминг. – Режим доступу: https://ru.wikipedia.org/wiki/Стемминг. – Назва з титул. екрана. 18. Стеммер Потера. – Режим доступу: http://labs.abcvg.com/stemmer/index.php. – Назва з титул. екрана. 19. Hardcoded stemmer for Ukrainian. – Режим доступу: https://github.com/vgrichina/ukrainian-stemmer. – Назва з титул. екрана. 20. Julie Beth Lovins (1968). Development of a stemming algorithm. Mechanical Translation and Computational Linguistics 11:22–31. 21. Jongejan, B. and H. Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike // In the 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. – P. 145–153. – Режим доступу: http://www.aclweb.org/anthology/P/P09/P09-1017.pdf. – Назва з титул. екрана. 22. Moseichuk V. Стемінг Портера для української мови.Стемминг Портера для украинского языка. Porter stemming algorithm for Ukrainian languages / V. Moseichuk. – Режим доступу: http://www. marazm.org.ua/document/stemer_ua/. – Назва з титул. екрана. 23. Perestoronin P. Стеммер Портера для русского языка / P. Perestoronin. – Режим доступу: http://blog.eigene.in/post/49598738049/ snowball. – Назва з титул. екрана. 24. Porter stemmer – реализация алгоритма стеммера Портера для русского языка на чистом функциональном языке Clojure. – Режим доступу: https://github.com/ allaud/porter-stemmer. – Назва з титул. екрана. 25. Porter M. F. An algorithm for suffix stripping (англ.) / M. F. Porter // Program. – 1980. – Т. 14. – № 3. – С. 130–137. (оригинальная публикация Портера). – Режим доступу: http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html. – Назва з титул. екрана. 25. Russian stemming algorithm. – Режим доступу: http://snowball.tartarus. org/algorithms/russian/ stemmer.html. – Назва з титул. екрана. 27. The Porter Stemming Algorithm – Porter's homepage. (англ.). – Режим доступу: http://tartarus.org/~martin/PorterStemmer/. – Назва з титул. екрана. 28. The Porter Stemming Algorithm – Project "Snowball" (англ.). – Режим доступу: http://snowball.tartarus.org/algorithms/porter/stemmer.html. – Назва з титул. екрана. 29. The English (Porter2) stemming algorithm (улучшенная версия алгоритма) – Project "Snowball" (англ.). – Режим доступу: http://snowball.tartarus.org/algorithms/english/stemmer.html. – Назва з титул. екрана. 30. Willett P. The Porter stemming algorithm: then and now (англ.) / P. Willett // Program: Electronic Library and Information Systems. – 2006. – В. 3. – Т. 40. – С. 219– 223. – ISSN 0033-0337. – Режим доступу: http://eprints.whiterose.ac.uk/1434/. – Назва з титул. екрана.*