УДК 004.912

Олексій Кунгурцев, Сергій Ковальчук, Яна Поточняк, Максим Широкоступ

ПОБУДОВА СЛОВНИКА ПРЕДМЕТНОЇ ОБЛАСТІ НА ОСНОВІ АВТОМАТИЗОВАНОГО АНАЛІЗУ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ

Алексей Кунгурцев, Сергей Ковальчук, Яна Поточняк, Максим Широкоступ

ПОСТРОЕНИЕ СЛОВАРЯ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ АВТОМАТИЗИРОВАННОГО АНАЛИЗА ТЕКСТОВ НА УКРАИНСКОМ ЯЗЫКЕ

Alexei Kungurtsev, Serhiy Kovalchuk, Iana Potochniak, Maxim Shirokostup

CREATING THE DOMAIN VOCABULARY ON THE BASIS OF AUTOMATED ANALYSIS OF UKRAINIAN TEXTS

Розроблено метод побудови словника предметної області за допомогою аналізу текстів українською мовою. Розроблено алгоритми для автоматизованого аналізу текстів українською мовою. Представлено алгоритми виділення речень з урахуванням переліків у тексті. Запропоновано метод опису таблиць термінів та процес формування таблиці термінів. Описано процес виділення термінів та занесення їх до сформованої таблиці термінів.

Ключові слова: *інформаційна система, синонім, термін, синтаксичний аналізатор, предметна область.*

Бібл.: 20.

Разработан метод построения словаря предметной области путем анализа текстов на украинском языке. Разработанные алгоритмы для автоматизированного анализа текстов на украинском языке. Представлены алгоритмы выделения предложений с учетом перечней в тексте. Предложен метод описания таблиц терминов и процесс формирования таблицы терминов. Описан процесс выделения терминов та внесение их в сформированную таблицу терминов.

Ключевые слова: *информационная система, синоним, термин, синтаксический анализатор, предметная область.*

Библ.: 20.

Developed the method of creating the domain vocabulary on basis automated analysis of Ukrainian texts. Developed the algorithms for automated analysis of Ukrainian texts. Developed the algorithms for search process of the sentences. Invited the method of description the tables of terms and formation process of the tables of terms. Described the selection process of the terms and saving them in table of terms.

Key words: *information system, synonym, term, parser, domain knowledge.*

Bibl.: 20.

Постановка проблеми. На сьогодні інформаційні системи (ІС) все ширше використовуються в різних сферах діяльності. Відповідно до цього збільшується кількість користувачів, які хочуть отримати доступ до інформації, яка зберігається в ІС [1]. З часом виникла потреба створення способу спілкування між некваліфікованим користувачем і ІС. Використання шаблонних запитів до реляційної бази даних не є повноцінним, оскільки ці шаблони не можуть врахувати всі бажання користувачів [2; 3]. Найкращим варіантом для цього є природня мова (ПМ) в інтерфейсному представленні. ПМ як інтерфейс інформаційних систем має вагомі переваги в порівнянні з іншими формами спілкування користувача з ІС. Ці переваги особливо вагомі для некваліфікованих користувачів [4; 5].

В основі проектування ІС лежить моделювання предметної області (ПрОб). Під моделлю ПрОб розуміється деяка система, що імітує структуру або функціонування досліджуваної ПрОб і відповідає основній вимозі – бути адекватною цій галузі [6]. Словники предметної області необхідні для реалізації систем управління в галузях техніки [7], медицини [8], в експертних системах [9]. Тому основним завданням є створення словника деякої ПрОб, для якої буде розроблятися ІС. Цей словник дозволить коректно записати вимоги, зрозуміти їхню суть та виділити об’єкти в предметній області. В подальшій розробці ІС створений словник ПрОб буде основою для побудови інтерфейсу користувача на ПМ. Побудова словника ПрОб створить можливість знаходити потрібні документи і фрагменти тексту серед раніше проаналізованих документів. Новизна полягає в побудові словника ПрОб для текстів українською мовою та введення індексації термінів у тексті. Терміни предметної області дозволяють формалізувати властивості різних предметних областей [10]. У перспективі індексування термінів дасть можливість пошуку документів і фрагментів тексту на основі статистичних характеристик.

Аналіз останніх досліджень і публікацій. Питання щодо створення інтерфейсу для спілкування користувачів з інформаційними системами на ПМ неодноразово порушувались та обговорювались [11]. Але все ж залишаються проблеми щодо аналізу текстів предметної області і виділення в них ключових слів [12]. Особливо ця проблема притаманна для української мови, оскільки немає інформації про побудову автоматизованого словника ПрОб на українській мові. Відповідно, суть проблеми полягає також у наявності програмних продуктів, які здійснюють синтаксичний аналіз тексту українською мовою. Здебільшого такі програмні продукти представлені в межах великих програмних комплексів. І саме наявність відкритого коду дає змогу застосувати потрібний модуль для синтаксичного аналізу українського тексту і знаходження термінів у тексті [13].

Термін і ключове слово тісно пов'язані, але є різними поняттями. Термін – слово або словосполучення, що визначає чітко й однозначно окреслене поняття і його співвідношення з іншими поняттями в межах спеціальної сфери. Ключове слово – слово або сталий вислів ПМ, яке використовують для вираження деякого аспекту змісту документа. Відповідно, термінів у тексті може бути значно більше, ніж ключових слів. У цій роботі здійснюється пошук термінів. Знаходження ключових слів у наукових текстах за допомогою стеммера Портера вже було розглянуто в роботі [14].

На підставі аналізу загальних тверджень [15], текст характеризувався тільки кількістю входжень ключових слів у тексті. Не можна було виділити фрагмент тексту найбільш інтенсивно використовуваного терміна. Також не можна було виділити окремий документ або фрагмент документа, в якого існує певне поєднання термінів.

Це все приводить до того, що потрібно не тільки знати кількість термінів у тексті, але і їхній розподіл по тексту. Відповідно до цього виникає проблема виявлення позицій термінів. Сучасні рішення пропонують різні методи пошуку текстової інформації для знаходження позицій ключових слів [16]. Всі вони передбачають виконання повторного пошуку по тексту, щоб знайти потрібний термін. Щоб уникнути подібних повторень, застосовується індексація термінів. Це забезпечує тільки одне проходження по тексту з метою виявлення позицій термінів, весь подальший пошук здійснюється за допомогою наявності індексованих термінів. У реляційних базах даних індексація є звичайною практикою, а основи індексування описані [17]. Проте в ІС з використанням інтерфейсу ПМ індексування термінів у текстах використовується не завжди, незважаючи на певні переваги.

Виділення не вирішених раніше частин загальної проблеми. За рахунок індексування (запису позицій) термінів у словнику предметної області буде затрачатися менше часу на пошук у тексті. Для створення словника предметної області з урахування позицій термінів у тексті потрібно вирішити такі проблеми.

1. Проаналізувати вхідний текст синтаксичним аналізатором української мови для знаходження в тексті термінів та приведення їх до нормальної форми. Сформувати список термінів з урахуванням кількості повторень у тексті.

2. Проаналізувати вхідний текст і виділити в ньому всі речення з урахуванням правил переліку значень у тексті та інших особливостей тексту.

3. Проаналізувати кожне речення з метою виявлення в них вже знайдених термінів і здійснення індексації термінів.

4. Побудувати словник предметної області відповідно до знайдених термінів і їхніх позицій у тексті. Урахувати можливість наявності синонімів та тлумачень термінів.

У результаті вирішення виділених проблем буде створений словник предметної області, який буде містити в собі всю необхідну інформацію про терміни. Також буде створений службовий файл для кожного вхідного тексту для того, щоб збільшити швидкість пошуку за рахунок індексації термінів.

Мета статті. Головною метою цієї роботи є розроблення методу побудови словника предметної області на основі автоматизованого аналізу текстів українською мовою, що дасть змогу зменшити час на побудову словника.

Синтаксичний аналіз вхідного тексту українською мовою. Для синтаксичного аналізу тексту українською мовою було використано програмне забезпечення з відкритим вихідним кодом Language Tool, а також модуль для Language Tool, який призначений для роботи з текстом українською мовою language-uk [13]. Цей програмний модуль був модифікований для виконання поставлених цілей. Ця програма підраховує кількість входжень слів, які є іменниками у тексті. Необхідно, щоб на вхід програми поступав текст, а у результаті програма видавала список слів (словосполучень), які є іменниками, із зазначенням кількості входжень кожного слова (словосполучення). Слова (словосполучення) мають бути приведені до початкової форми.

Програмне забезпечення LanguageTool працює на основі словника, тому для слів, які пишуться однаково, неможливо однозначно визначити такі параметри, як рід, число, відмінок тощо. Далі наведено повний список параметрів: *noun* іменник, *adj* прикметник, *adjp* дієприкметник, *adv* прислівник, *prep* прийменник, *numr* числівник, *part* частка, *pron* займенник, *verb* дієслово, *p* множина, *s* однина, *m* чоловічий, *f* жіночий, *n* середній; відмінки: *v_naz, v_rod, v_dav, v_zna, v_oru, v_mis, v_kly*; *actv* активний, *pasv* пасивний, *nv* не відмінюється, *np* без множини, *pres* теперішній час, *3* 3-а особа, *perf* доконане, *imperf* недоконане, *tran* перехідне, *intran* неперехідне, *coord* сурядний (сполучник), *subord* підрядний (сполучник), *compb* базова форма, *def* означальний, *rel* відносний, *int* питальний, *ind* неозначений, *anim* істота (людина/людиноподібна істота).

Через неоднозначність у результаті роботи програма для деяких слів повертає набір можливих форм слів, які отримано у результаті визначення параметрів, а також набір форм, які було отримано у результаті узгодження іменника і прикметника (дієприкметника).

Алгоритм роботи синтаксичного аналізатора української мови:

1. Розбивка тексту на слова (LanguageTool).
2. Видалення слів зі списку, які повинні ігноруватись.
3. Вибирається слово. Проводиться морфологічний розбір слова (LanguageTool).
4. Якщо слово є іменником, виконуємо для нього п. 5. Перевірка чи є попереднє і наступне слово прикметником (дієприкметником), якщо так, то узгоджуємо його із іменником і виконуємо для нього п. 5.
5. Перевірка чи є вхідне слово (словосполучення) унікальним для результуючого списку, якщо так, то добавляємо його у список, якщо ні, то збільшуємо значення змінної, яка відповідає за кількість входжень слова у текст на 1.
6. Якщо у вхідному спискові є ще слова, то виконується перехід до пункту 3.

Виділення термінів. Для виділення термінів на цьому етапі не потрібно виділяти повноцінні речення. Виділення речень відбувається примітивно, враховуючи тільки знаки закінчення речення: «.», «!», «?». Це потрібно для того, щоб не допустити наявність знаків закінчення речення між елементами терміна. На цьому етапі не враховується належність терміна до конкретного речення.

Для того, щоб виділити терміни, потрібно здійснити два проходи по тесту. На першому проході потрібно знайти всі іменники, які будуть термінами на цьому етапі.

Уявімо аналізований текст T у вигляді множини умовних речень S:

$$T = \{S_i\} i = 1, n \ , \tag{1}$$

а кожне речення – у вигляді послідовності елементів e (слів і знаків пунктуації):

$$e_1, ..., e_l, .., e_m \ . \tag{2}$$

Кожен елемент буде характеризуватися текстом N та множиною атрибутів A:

$$e =< N, A > . \tag{3}$$

Визначимо деякі з атрибутів. Нехай $A1$ є частиною мови, $A2$ – число, $A3$ – рід, $A4$ – особа, $A5$ – відмінок, $A6$ – час.

Знаходимо в кожному реченні іменники *noun*. Якщо виконується умова:

$$e_l \to A1 = noun, \quad (4)$$

то як термін приймається $tx = e_l$.

На другому проході шукаємо частоти спільної появи виділених раніше іменників і сусідніх слів. Потрібно врахувати природні обмеження довжини стійкого словосполучення, такі як: розділові знаки; слова, які не можуть входити в термін (наприклад, займенник); обмеження по довжині (не більше 3 слів зліва і з права від раніше виявленого терміна).

Нехай досліджувана група має вигляд:

$$C_{x-3} + C_{x-2} + C_{x-1} + tx_x + C_{x+1} + C_{x+2} + C_{x+3}, \quad (5)$$

де $C_{x-3} - C_{x+3}$ – слова (можливі кандидати на входження в словосполучення).

Здійснюємо прохід вліво. Якщо виконується умова:

$$C_{x-1} \to agj \wedge (tx_x \to A2 = C_{x-1} \to A2) \wedge (tx_x \to A3 = C_{x-1} \to A3) \wedge (tx_x \to A5 = C_{x-1} \to A5), \quad (6)$$

то як термін приймається $tx_x = C_{x-1} + tx_x$. Робимо повторні дії до кінця лівого порогу для слів C_{x-2}, C_{x-3}.

Після закінчення проходу вліво здійснюємо прохід вправо. Якщо виконується умова:

$$C_{x+1} \to agj \wedge (tx_x \to A2 = C_{x+1} \to A2) \wedge (tx_x \to A3 = C_{x+1} \to A3) \wedge (tx_x \to A5 = C_{x+1} \to A5), \quad (7)$$

то як термін приймається $tx_x = tx_x + C_{x+1}$. Робимо повторні дії до кінця правого порогу для слів C_{x+2}, C_{x+3}.

Якщо вказані умови не виконуються, то нові терміни не додаються.

Виявляємо частоти появ можливих термінів: $tx_x + C_{x+1}$, $tx_x + C_{x+1} + C_{x+2}$, $tx_x + C_{x+1} + C_{x+2} + C_{x+3}$, $C_{x-1} + tx_x$, $C_{x-2} + C_{x-1} + tx_x$, $C_{x-3} + C_{x-2} + C_{x-1} + tx_x$, $C_{x-1} + tx_x + C_{x+1}$, $C_{x-2} + C_{x-1} + tx_x + C_{x+1} + C_{x+2}$ тощо.

Встановлюємо поріг мінімальної частоти для словосполучення. У випадку, коли однослівний термін зустрічається як самостійно, так і в словосполученні, то потрібно створити два відповідні терміни. Якщо є конкуруючі словосполучення, це означає, що всі вони є в списку термінів.

Результатом роботи такого синтаксичного аналізатора є два списки: *список № 1* містить у собі всі терміни в початковій формі з вказаною кількість повторень у тексті для кожного; *список № 2* містить у собі всі терміни в тому вигляді, в якому вони зустрічаються в тексті, також біля кожного такого слова записана його початкова форма. Для подальшого аналізу вхідного тексту буде використовуватися *список № 2*.

Для того, щоб дізнатися позиції термінів у тексті, потрібно проаналізувати вхідний текст на наявність перечислень, і врахувати це при розділені тексту на речення. Після цього потрібно розділити вхідний текст на речення, кожне речення записується з нового рядка службового файла.

Формування рядків з урахуванням правил переліку в тексті. Перед початком аналізу тексту потрібно перевести вхідний текст T у формат **.txt* і результат (вихідний текст) Tr_1 теж записується у форматі **.txt*.

Кожен текст може складатися з певних перечислень, які записуються з нового рядка, при цьому не несуть повну інформацію, оскільки не є повноцінними реченнями. Потрібно проаналізувати текст на наявність у ньому переліків і об'єднати їх у повноцінні речення відповідно до знаків пунктуації.

Вхідний текст T може характеризуватися не тільки реченнями (1), а ще і складається з множини рядків R, відповідно $T = \{R_q\} q = 1, b$.

Для початку потрібно проаналізувати всі рядки R_q вхідного тексту T на наявність у тексті різного типу переліків *Type*, а саме простих *Simple* і складних *Complex* переліків.

Кожний рядок R_q складається з множини елементів, $R_q = \{e_t\} t = 1, v$. Здійснюємо прохід по кожному рядку R_q відповідно до розміру $q = v$ вхідного тексту T.

Крок 1. Першим кроком аналізуємо останній символ елементу e_v кожного рядка R_q.

Крок 2. Якщо останній елемент e_v рядка R_q закінчується символом «**:**» e_v =':' і відзначено, що перелік ще не розпочинався, то додаємо рядок R_q в вихідний текст $Tr_1 = \{R_w\} w = 1, x$. Перед додаванням рядків вихідний текст $Tr_1 = \varnothing$. Відзначаємо, що перелік розпочався і наступний рядок R_{q+1} тексту T буде входити до цього переліку (*Type = Simple*), якщо останній елемент e_v рядка R_q не закінчується символом «**:**» $e_v[end] \neq$':', то переходимо до наступного пункту.

Крок 3. Якщо останній елемент e_v рядка R_q закінчується символом «**:**» $e_v[end]$=':' і відзначено, що перелік розпочався, то додаємо рядок R_q у вихідний текст Tr_1 (*Type = Complex*), в іншому випадку переходимо до наступного кроку.

Крок 4. Якщо останній елемент e_v рядку R_q закінчується символом «**;**» $e_v[end]$=';' або символом «**,**» $e_v[end]$=',' за умови, що перелік розпочався, то до рядка R_{w-1} вихідного тексту Tr_1 додається рядок R_q $(R_{w-1} \in Tr_1) = (R_{w-1} \in Tr_1) + (R_q \in T)$, відзначаємо, що триває об'єднання переліку в один рядок. В іншому випадку переходимо до наступного кроку.

Крок 5. Якщо знайдено останній пункт переліку, тобто останній елемент e_v закінчується символом «**.**» $e_v[end]$='.', то до рядка R_{w-1} вихідного тексту Tr_1 додається рядок R_q вхідного тексту T, відзначаємо, що перелік і об'єднання закінчено, в іншому випадку переходимо до наступного кроку.

Крок 6. Якщо перераховуються окремі речення, які закінчуються елементом e_v з останнім символом «**.**» $e_v[end]$='.', то такі пункти не входять до переліку і записуються як окремі рядки R_w вихідного тексту Tr_1.

Крок 7. Якщо ні одна з умов не була виконана, то рядок R_q вхідного тексту T копіюється у вихідний текст Tr_1 без змін і без відміток про початок переліку.

Вихідний текст $Tr_1 = \{R_w\}$ переходить до наступної стадії оброблення.

Виділення речень. Виділення повноцінних речень відбувається за допомогою регулярних виразів.

Для коректного виділення речень у тексті потрібно дотримуватися таких правил:

1. Повноцінно закінченим реченням вважається речення, яке закінчується символом «.» або «!» або «?». Також можна вважати кінцем речення символ кінця рядку (\n).
2. Кожне наступне речення розпочинається з великої літери.
3. Якщо речення розпочинається з нумерації типу «1. Текст.», то нумерацію «1.» потрібно віднести до цього речення.
4. Якщо в реченні зустрічається скорочення типу «і т. д.», то потрібно відносити його до складу цього речення.

5. Якщо в реченні зустрічаються ініціали типу «Т. Г. Шевченко» або «Т. Шевченко», то потрібно відносити їх до складу цього речення.

6. У випадку, коли в реченні зустрічається текст у дужках «( )», потрібно все, що міститься в дужках, віднести до складу цього речення.

Потрібно проаналізувати текст Tr_1 (результат попереднього етапу) і виділити із рядків R_w повноцінні речення, які будуть занесені до результуючого тексту Tr, який складається з множини речень S_j, $Tr=\{S_j\}j=1,z$. До моменту аналізу вхідного тексту Tr_1, $Tr=\varnothing$, $j=0$.

Кожен рядок R_w складається з цілого числа речень S, яких може бути 1 або більше. Виділення речень відбувається для кожного рядка R_w окремо відповідно до їхньої послідовності. Речення може розпочинатися як зі слова з великої літери, так і з нумерації.

Кожен рядок R_w можна також представити у вигляді послідовності символів, а не тільки як послідовність елементів (2),

$$el_1,...,el_r,...,el_u,...,el_h. \quad (8)$$

Введемо деякі позначення для визначень речень:

Велика літера – $((A-Я)\vee(A-Z))\rightarrow Big$.

Маленька літера – $((a-я)\vee(a-z))\rightarrow Small$.

Не літера – $\neg((A-Я)\vee(A-Z)\vee(a-я)\vee(a-z))\rightarrow notBS$.

bg – початок рядка.

en – кінець рядка.

Визначення окремих фрагментів речення:

1. Слово (або літера), яке розпочинається з великої літери:

$A_r\in((el_{r-1}=[bg])\vee(el_{r-1}=[notBS]))\wedge((el_r=[Big])\wedge...\wedge(el_u=[Big\vee Small]))\wedge(el_{u+1}=[notBS]).$

2. Слово (або літера), яке розпочинається з маленької літери:

$A2_r\in(el_{r-1}=[notBS])\wedge((el_r=[Small])\wedge...\wedge(el_u=[Big\vee Small]))\wedge(el_{u+1}=[notBS])$.

3. Закінчення речення:

$B_r\in((el_{r-1}='.')\vee(el_{r-1}='!')\vee(el_{r-1}='?'))\wedge(((el_r='')\wedge A_{r+1})\vee(el_r=en))$.

4. Нумерація:

$C_r\in(el_{r-1}=[bg])\wedge(el_r=[0-9])\wedge...\wedge(el_u=[0-9])\wedge(el_{u+1}='.')\wedge(el_{u+2}='')\wedge A_{u+3}$.

5. Ініціали:

$D_r\in(el_{r-1}='')\wedge(el_r=[Big])\wedge(el_{r+1}='.')$.

6. Скорочення:

$E_r\in(el_{r-1}='.')\wedge((el_r='')\wedge A2_{r+1})$.

7. Текст у дужках:

$F_r\in(el_r='(')\wedge(el_{r+1}=f)\wedge...\wedge(el_u=f)\wedge(el_{u+1}=')')$,

де f – будь-які символи.

На основі визначення окремих фрагментів речення можна описати умову виділення речення в тексті:

$$(A_r\vee C_r)\wedge((A_{r+1}\vee A2_{r+1}\vee D_{r+1}\vee E_{r+1}\vee F_{r+!})\wedge...\wedge(A_u\vee A2_u\vee D_u\vee E_u\vee F_u))\wedge B_{u+1}. \quad (9)$$

Кожне знайдене речення в тексті Tr_1 записується в новому рядку вихідного тексту Tr. На виході отримуємо вихідний текст $Tr=\{S_j\}$, який переходить до наступної стадії оброблення.

Процес створення таблиці термінів пропонується розділити на декілька етапів. Спочатку формується початкова форма таблиця термінів, в якій міститься інформація, отримана після роботи синтаксичного аналізатора. На основі таблиці початкової форми знаходяться позиції термінів і зберігаються в ній. Наступним кроком пропонується модифікувати початкову форму таблиці термінів до першої форми таблиці термінів. Це дасть змогу відсіяти всю зайву інформацію і дописати потрібну. Наступним етапом потрібно модифікувати першу форму таблиці термінів до основної таблиці термінів, для того, щоб виділити синоніми як терміни. Це дасть змогу прискорити пошук термінів у подальшій роботі. Останнім етапом буде корекція основної таблиці термінів, для усунення повторень між усіма термінами.

Початкова форма таблиця термінів. На основі результатів виявлених термінів із *списку № 2* формуємо початкову форму таблиці термінів, яка в подальшому буде використана для знаходження позицій термінів (перше проходження) та формування основної форми таблиці. Представимо початкову форму таблиці термінів у вигляді множини записів $TP = \{tp_{pi}\}\, pi = 1, pn$. Кожен запис являє собою кортеж:

$$tp_{pi} =< tx_{pi}, pf_{pi}, inText_{pi}, countT_{pi}, posR_{pi} >, \quad (10)$$

де tx_{pi} – термін (одне або декілька слів);

pf_{pi} – список можливих початкових форм терміну (якщо вони є);

$inText_{pi}$ – список усіх форм терміну, які зустрічаються в тексті;

$countT_{pi}$ – кількість поява терміна в тексті;

$posR_{pi}$ – список позицій, в яких виявлено термін.

Список позицій, в яких виявлено термін $posR_{pi}$, представимо у вигляді послідовності елементів $er_1,...,er_o,...,er_{op}$.

Кожен елемент er_o буде характеризуватися атрибутами:

$$er_o =< Nr, pos, len >, \quad (11)$$

де Nr – номер речення, в якому знаходиться термін;

pos – номер символу, з якого розпочинається термін (позиція терміна в реченні);

len – кількість символів у терміні (довжина терміна).

На початку список позицій $posR_{pi}$ для кожного терміна tx_{pi} є пустим.

Знаходження позицій термінів. Зчитуємо інформацію з початкової форми таблиці TP про термін tx_{pi}. $inText_{pi}$ – являє собою послідовність елементів, які є списком усіх форм терміна, які зустрічаються в тексті $ep_1,...,ep_p,...,ep_{pl}$.

За допомогою алгоритму Бойера – Мура [18] здійснюємо пошук ep_p у множині усіх речень S_j тексту Tr. Алгоритм пошуку стрічки Бойера – Мура, вважається найбільш швидким серед алгоритмів загального призначення, призначених для пошуку підрядка в рядку. Алгоритм порівнює символи шаблону ep_p справа наліво, починаючи з самого правого, один за іншим з символами вихідного рядка S_j. Якщо символи збігаються, проводиться порівняння передостаннього символу шаблону і так до кінця. Якщо всі символи шаблону збіглися з накладеними символами рядка, значить, підрядок знайдено, і пошук закінчено. Що стосується розбіжності будь-якого символу (або повного збігу всього шаблону), він використовує дві попередньо обчислювані евристичні функції, щоб зрушити позицію для початку порівняння вправо.

Якщо елемент ep_p знайдено в реченні S_j, то корегуємо записи в таблиці *TP*. Формуємо список позицій $posT$ терміна, але не заносимо його в таблицю, доки елемент ep_p не буде порівняний з усіма елементами речень S_j. Коли список $posT$ сформований, заносимо його в таблицю $posR_{pi} = posT$ для конкретного терміна.

Здійснюємо повторні дії для кожного терміна tx_{pi} таблиці *TP*. У результаті буде відомо позиції усіх термінів у тексті.

Перша форма таблиці термінів. Для подальшого аналізу потрібно перетворити початкову таблицю термінів *TP* до першої форми, відсіюючи непотрібну інформацію і з додаванням необхідної інформації.

Першу форму таблиці термінів *TF* можна представити у вигляді множини записів $TF = \{tf_{pos}\} pos = \overline{1, len}$. Кожен запис являє собою кортеж записів:

$$tf_{pos} = < tx_{pos}, countT_{pos}, posR_{pos}, synonimT_{pos}, valueT_{pos} >, \quad (12)$$

де $synonimT_{pos}$ – список синонімів терміна;

$valueT_{pos}$ – тлумачення терміна.

Під тлумачним словником предметної області розуміється спеціалізований словник, який дає роз'яснення множині понять, пов'язаних з діяльністю деякої організаційної структури. Кожний запис у словнику представляє одне слово або стійке для цієї предметної області словосполучення, для якого наведене тлумачення, специфічне для цієї предметної області, а також список синонімів.

Метод автоматичної побудови тлумачного словника предметної області розглядався в роботі [19]. Розвитком цієї роботи стало урахування міжфразових зв'язків [20].

Значення таблиці *TP* буде призначено таблиці *TF*, а саме $(tx_{pos} \in TF) = (tx_{pi} \in TP)$, $(countT_{pos} \in TF) = (countT_{pi} \in TP)$, $(posR_{pos} \in TF) = (posR_{pi} \in TP)$. Список синонімів $synonimT_{pos}$ формується відповідно до словників синонімів. Тлумачення терміна $valueT_{pos}$ формується відповідно до тлумачних словників.

Список синонімів терміна $synonimT_{pos}$ являє собою послідовність елементів (синонімів) $Sn_1, ..., Sn_{sb}, ..., Sn_{sf}$. Список позицій термінів $posR_{pos}$ являє собою послідовність елементів $p_1, ..., p_{ps}, ..., p_{pf}$.

Основна форма таблиці термінів. Результатом аналізу першої форми таблиці *TF,* буде основна таблиця термінів *TT*. Представимо основну таблицю термінів у вигляді множини записів $TT = \{tt_{\text{int}}\} \text{int} = \overline{1, nt}$. Кожен запис являє собою кортеж:

$$tt_{\text{int}} = < tx_{\text{int}}, Group_{\text{int}}, countT_{\text{int}}, posR_{\text{int}}, valueT_{\text{int}} >, \quad (13)$$

де $Group_{\text{int}}$ – номер групи термінів (спільна група означає, що терміни є синонімами).

У результаті заповнення таблиці термінів *TT* даними з таблиці термінів першої форми *TF*, будемо мати такі результати.

$(tx_{\text{int}} \in TT) = (tx_{pos} \in TF), (tx_{\text{int}+1} \in TT) = (Sn_1 \in TF), ..., (tx_{\text{int}+b} \in TT) = (Sn_b \in TF), ..., (tx_{\text{int}+f} \in TT) = (Sn_f \in TF);$

$(Group_{\text{int}} \in TT) = gr, (Group_{\text{int}+1} \in TT) = gr, ..., (Group_{\text{int}+b} \in TT) = gr, ..., (Group_{\text{int}+f} \in TT) = gr,$

де gr – індивідуальний номер групи;

$(countT_{\text{int}} \in TT) = (countT_{pos} \in TF), (countT_{\text{int}+1} \in TT) = null, ..., (countT_{\text{int}+b} \in TT) = null, ..., (countT_{\text{int}+f} \in TT) = null;$

$(posR_{\text{int}} \in TT) = (posR_{pos} \in TF), (posR_{\text{int}+1} \in TT) = null, ..., (posR_{\text{int}+b} \in TT) = null, ..., (posR_{\text{int}+f} \in TT) = null;$

$$(value_{\text{int}} \in TT) = (value_{pos} \in TF), (value_{\text{int}+1} \in TT) = null, ..., (value_{\text{int}+b} \in TT) = null, ..., (value_{\text{int}+f} \in TT) = null.$$

Наступним кроком потрібно порівняти терміни між різними групами, щоб уникнути повторень. Якщо виявиться, що терміни однієї групи збігаються з термінами іншої, то потрібно дані з другої групи додати до першої групи і видалити другу групу. Групування термінів дозволяє в подальшому використанні здійснювати швидший пошук терміна.

Висновки і пропозиції. Розроблений метод побудови словника предметної області на основі автоматизованого аналізу тексту дозволяє зменшити час на побудову словника. Розроблено метод синтаксичного аналізу тексту української мови. Представлено алгоритми аналізу тесту для можливості здійснення індексації термінів. Реалізовано алгоритм побудови словника предметної області на основі попереднього аналізу тесту. Передбачена можливість додавання синонімів та тлумачень до кожного терміна. Для кожного вхідного тексту передбачено створення службової копії з структурою відповідно до представлених алгоритмів. Відповідно до наведених аналізу й алгоритмів було розроблено програмний продукт, який дав змогу створити словник для предметної області. Дослідження за методом з роботи [19] показали, що час створення словника скоротився орієнтовно в 5 разів у порівнянні зі створенням словника вручну.

Список використаних джерел

1. *Автоматическая* обработка текстов на естественном языке и компьютерная лингвистика / Е. И. Большакова, Э. С. Клышинский, Д. В. Ландэ, А. А. Носков, О. В. Пескова, Е. В. Ягунова. – М. : МИЭМ, 2011. – 272 с.

2. *Bates, M.* Models of natural language understanding. Proceedings of the National Academy of Sciences of the United States of America, Vol. 92, No. 22 (1995), pp. 9977–9982.

3. *Кунгурцев О. Б.* Формування шаблонів запитів до реляційної бази даних з використанням об'єктного словника / О. Б. Кунгурцев, І. В. Барикіна, О. А. Завалін // Наукові праці ОНХАТ. – 2004. – Вип. 27.– С. 233–236.

4. *Акимов О. М.* Интеллектуализация интерфейса базы данных / О. М. Акимов, В. А. Шапцев // Известия Томского политехнического университета. – 2009. – Т. 314, № 5. – С. 137–139.

5. *Steven Bird*, Ewan Klein, Edward Loper. Natural Language Processing with Python. (2009), Published by O'Reilly Media. – 481 p.

6. *Методологии* моделирования предметной области [Электронный ресурс]. – Режим доступа : http://www.intuit.ru/studies/courses/2195/55/lecture/1628.

7. *Рувинская В. М.* Словарь предметной области для разработки экспертной системы / В. М. Рувинская, А. С. Тройнина, Д. А. Силяев // Труды Международной научно технической конференции «Информационные технологи в металлургии и машиностроении» (Днепропетровск, 24–26.03.2015). – Днепропетровск, 2015. – С. 44.

8. *Chernega K. S.* Decision support System for Automated Medical Diagnostics / Chernega K.S., Tymchenko B.I., Komleva N.O. // Electrotechnic and Computer Systems. – Kiev: Science and Technology, 2016. – No. 23 (99). – P. 65–72.

9. *Ruvinska V.* Rules of Expert System for Safety Monitoring: Checking on Completeness and Consistency / V. Ruvinska, A. Troynina, E. Berkovich, A. Bilovzorov // Праці Одеського політехнічного університету. – 2015. – Вип. 2 (46). – С. 103–110.

10. *Любченко В. В.* О некоторых свойствах моделей предметных областей информационных систем / В. В. Любченко // Труды 14-й Международной научно-практической конференции «Современные информационные и электронные технологии» (27–31 мая 2013 г., Одесса). – Одесса, 2013. – Т. 1. – С. 81–82.

11. *Кунгурцев А. Б.* Интерфейс для общения пользователей с информационными системами на естественном языке / А. Б. Кунгурцев, Я. В. Поточняк // Электротехнические и компьютерные системы. – 2014. – № 14 (90). – С. 74–81.

12. *Необходимость* выделения ключевых слов для свёртывания текста [Электронный ресурс]. – Режим доступа : http://www.scienceforum.ru/2014/476/70.

13. *LanguageTool* [Електронний ресурс]. – Режим доступу : https://languagetool.org/uk/.

14. *Бісікало О. В.* Виявлення ключових слів на основі методу контент-моніторингу україномовних текстів / О. В. Бісікало, В. А. Висоцька // Радіоелектроніка, інформатика, управління. – 2016. – № 1 (36). – С. 74–83.

15. *Ключевые* слова в тексте – как правильно употреблять? [Электронный ресурс]. – Режим доступа : http://dimokfm.ru/klyuchevyie-slova/.

16. *Информационные* технологии управления. Методы поиска текстовой информации [Электронный ресурс]. – Режим доступа : https://refdb.ru/look/2575304-p10.html.

17. *Индексы*. Теоретические основы [Электронный ресурс]. – Режим доступа : http://www.sql.ru/articles/mssql/03013101indexes.shtml.

18. *Алгоритм* Бойера – Мура [Електронний ресурс]. – Режим доступу : http://www.algolib.narod.ru/Search/BoyerMur.html.

19. *Кунгурцев А. Б.* Метод автоматизированного построения толкового словаря предметной области / А. Б. Кунгурцев, Я. В. Поточняк, Д. А. Силяев // Технологический аудит и резервы производства. – 2015. – Вип. 2, № 2 (22). – С. 58–63.

20. *Учет* межфразовых связей при автоматизированном построении толкового словаря предметной области / А. Б. Кунгурцев, А. И. Гаврилова, А. С. Леонгард, Я. В. Поточняк // Информатика и математические методы в моделировании. – 2016. – № 2. – С. 173–183.

References

1. Bolshakova, E.I., Klyshinskii, E.S., Lande, D.V., Noskov, A.A., Peskova, O.V., Yagunova, E.V. (2011). *Avtomaticheskaia obrabotka tekstov na estestvennom iazyke i kompiuternaia lingvistika [Automatic processing of natural language texts and computational linguistics]*. Moscow: MIEM (in Russian).

2. Bates, M. (1995). *Models of natural language understanding*. Proceedings of the National Academy of Sciences of the United States of America, vol. 92, no. 22, pp. 9977–9982.

3. Kunhurtsev, O.B., Barykina, I.V., Zavalin, O.A. (2004). Formuvannia shabloniv zapytiv do reliatsiinoi bazy danykh z vykorystanniam obiektnoho slovnyka [Creating standard queries to the relational database using object dictionary]. *Naukovi pratsi ONKhAT – Proceedings of ONAFT,* issue 27, pp. 233–236 (in Ukrainian).

4. Akimov, O.M., SHaptcev, V.A. (2009). Intellektualizatciia interfeisa bazy dannykh [Intellectualization database interface]. *Izvestiia Tomskogo politekhnicheskogo universiteta – Bulletin of the Tomsk Polytechnic University*, vol. 314, no. 5, pp. 137–139 (in Russian).

5. Steven Bird, Ewan Klein, Edward Loper (2009). *Natural Language Processing with Python*. Published by O'Reilly Media.

6. *Metodologii modelirovaniia predmetnoi oblasti [Methodologies domain simulation]*. Retrieved from http://www.intuit.ru/studies/courses/2195/55/lecture/1628.

7. Ruvinskaia, V.M., Troinina, A.S., Siliaev, D.A. (2015). Slovar predmetnoi oblasti dlia razrabotki ekspertnoi sistemy [Domain vocabulary for the development of an expert system]. Proceedings of the *Trudy Mezhdunarodnoi nauchno tekhnicheskoi konferentcii "Informatcionnye tekhnologi v metallurgii i mashinostroenii" – International scientific conference "Information technologies in metallurgy and machinebuilding"* (Dnepropetrovsk 24–26.03.2015). Dnepropetrovsk (in Russian).

8. Chernega, K.S., Tymchenko, B.I., Komleva, N.O. (2016). *Decision support System for Automated Medical Diagnostics. Electrotechnic and Computer Systems*. Kiev: Science and Technology, no. 23 (99), pp. 65–72.

9. Ruvinska, V., Troynina, A., Berkovich, E., Bilovzorov, A. (2015). Rules of Expert System for Safety Monitoring: Checking on Completeness and Consistency. *Pratsi Odeskoho politekhnichnoho universytetu – Pratsi ONPU*, issue 2 (46), pp. 103–110.

10. Liubchenko, V.V. (2013). O nekotorykh svoistvakh modelei predmetnykh oblastei informatcionnykh sistem [About some properties of the subject areas of information systems models]. Proceedings of the *Trudy 14-i Mezhdunarodnoi nauchno-prakticheskoi konferentcii "Sovremennye informatcionnye i elektronnye tekhnologii" – 14th International scientific-practical conference "Modern information and electronic technologies"* (May 27-31, 2013, Odessa). Odessa, vol. 1, pp. 81–82 (in Russian).

11. Kungurtsev, A.B., Potochniak, I.B. (2014). Interfeis dlia obshcheniia polzovatelei s informatcionnymi sistemami na estestvennom iazyke [User interface for users communication with

information systems in a natural language]. *Elektrotekhnicheskie i kompiuternye sistemy – Electrical and computer systems,* no. 14 (90), pp. 74–81 (in Russian).

12. *Neobkhodimost vydeleniia kliuchevykh slov dlia svertyvaniia teksta [Selection of keywords for clotting text]*. Retrieved from http://www.scienceforum.ru/2014/476/70.

13. *LanguageTool*. Retrieved from https://languagetool.org/uk/.

14. Bisikalo, O.V., Vysotska, V.A. (2016). Vyiavlennia kliuchovykh sliv na osnovi metodu kontent-monitorynhu ukrainomovnykh tekstiv [Identifying keywords on the basis of content monitoring method in Ukrainian texts]. *Radioelektronika, informatyka, upravlinnia – Radio Electronics, Computer Science, Control*, no. 1 (36), pp. 74–83 (in Ukrainian).

15. *Kliuchevye slova v tekste kak pravilno upotrebliat? [The key words in the text – how to use?]*. Retrieved from http://dimokfm.ru/klyuchevyie-slova/.

16. *Informatcionnye tekhnologii upravleniia Metody poiska tekstovoi informatcii [Information Technology Management. Search methods of textual information]*. Retrieved from https://refdb.ru/look/2575304-p10.html.

17. *Indeksy. Teoreticheskie osnovy [Indexes. Theoretical basis]*. Retrieved from http://www.sql.ru/articles/mssql/03013101indexes.shtml.

18. *Algoritm Boiera – Mura [Boyer–Moore string search algorithm]*. Retrieved from http://www.algolib.narod.ru/Search/BoyerMur.html.

19. Kungurtcev, A.B., Potochniak, Ya.V., Siliaev, D.A. (2015). Metod avtomatizirovannogo postroeniia tolkovogo slovaria predmetnoi oblasti [Method automated construction explanatory vocabulary domain]. *Tekhnologicheskii audit i rezervy proizvodstva – Technology audit and production reserves*, vol 2, no. 2 (22), pp. 58–63 (in Russian).

20. Kungurtcev, A.B., Gavrilova, A.I., Leongard, A.S., Potochniak, Ya.V. (2016). Uchet mezhfrazovykh sviazei pri avtomatizirovannom postroenii tolkovogo slovaria predmetnoi oblasti [Accounting of inter-phrase communication for automated construction the explanatory dictionary of domain knowledge]. *Informatika i matematicheskie metody v modelirovanii – Informatics and Mathematical Methods in modeling*, no. 2, pp. 173–183 (in Russian).

Кунгурцев Олексій Борисович – кандидат технічних наук, професор кафедри системного програмного забезпечення, Одеський національний політехнічний університет (просп. Шевченка, 1, м. Одеса, 65044, Україна).
Кунгурцев Алексей Борисович – кандидат технических наук, профессор кафедры системного программного обеспечения, Одесский национальный политехнический университет (просп. Шевченко, 1, г. Одесса, 65044, Украина).
Kungurtsev Alexei – PhD in Technical Sciences, Professor of Department of System Software, Odessa National Polytechnic University (1 Shevchenka Av., 65044 Odessa, Ukraine).
E-mail: abkun@te.net.ua
Orcid: 0000-0002-3207-7315

Ковальчук Сергій Вікторович – аспірант кафедри системного програмного забезпечення, Одеський національний політехнічний університет (просп. Шевченка, 1, м. Одеса, 65044, Україна).
Ковальчук Сергей Викторович – аспирант кафедры системного программного обеспечения, Одесский национальный политехнический университет (просп. Шевченко, 1, г. Одесса, 65044, Украина).
Kovalchuk Serhiy – PhD student of Department of System Software, Odessa National Polytechnic University (1 Shevchenka Av., 65044 Odessa, Ukraine).
E-mail: serhiy_kovalchuk@mail.ua

Поточняк Яна Володимирівна – аспірант кафедри системного програмного забезпечення, Одеський національний політехнічний університет (просп. Шевченка, 1, м. Одеса, 65044, Україна).
Поточняк Яна Владимировна – аспирант кафедры системного программного обеспечения, Одесский национальный политехнический университет (просп. Шевченко, 1, г. Одесса, 65044, Украина).
Potochniak Iana – PhD student of Department of System Software, Odessa National Polytechnic University (1 Shevchenka Av., 65044 Odessa, Ukraine).
E-mail: yana_onpu@mail.ru

Широкоступ Максим Віталійович – студент, Одеський національний політехнічний університет (просп. Шевченка, 1, м. Одеса, 65044, Україна).
Широкоступ Максим Витальевич – студент, Одесский национальный политехнический университет (просп. Шевченко, 1, г. Одесса, 65044, Украина).
Shirokostup Maxim – student, Odessa National Polytechnic University (1 Shevchenka Av., 65044 Odessa, Ukraine).
E-mail: unikursal@gmail.com

Побудова словника предметної області на основі автоматизованого аналізу текстів українською мовою / О. Кунгурцев, С. Ковальчук, Я. Поточняк, М. Широкоступ // Технічні науки та технології. – 2016. – № 3 (5). – С. 164–174.