КОМП’ЮТЕРНА ТА МАТЕМАТИЧНА ЛІНГВІСТИКА

УДК 004.738.5

В. А. Андруник, Р. Р. Бекеш, Л. Б. Чирун
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

СТРУКТУРНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ АНАЛІЗУ ТА СИНТЕЗУ ТЕХНІЧНОГО ТЕКСТУ

Подано застосування породжувальних граматик у лінгвістичному моделюванні. Опис моделювання синтаксису речення застосовують для автоматизації процесів аналізу та синтезу природномовних текстів.

Ключові слова: породжувальні граматики, структурна схема речення, комп’ютерна лінгвістична система.

This paper presents the generative grammar application in linguistic modelling. Description of syntax sentence modelling is applied to automate the processes of analysis and synthesis of texts in natural language.

Key words: Generative grammar, structured scheme sentences, computer linguistic system.

Вступ. Загальна постановка проблеми

Особливістю розвитку сучасної української науково-технічної термінології є посилена зацікавленість в її автентичності, оскільки історично склалося так, що ця термінологія стала недоступною для користувачів. З офіційних словників та підручників цю термінологію було вилучено, а заборонені словники потрапили до спецсховищ бібліотек, і їх видавали лише за спеціальним дозволом. До сьогодні словники 1920–1930 рр. дійшли в поодиноких примірниках або й не дійшли зовсім – їх загублено або знищено. Навіть про саме існування багатьох термінологічних словників тепер відомо лише вузькому колу фахівців [2, 5–11, 17, 19–20].

Підраховано, що близько 90 % нових слів, що з'являються у кожній мові – це терміни. Сучасна українська термінологія також активно поповнюється новими одиницями – переважно запозиченнями з англійської мови або словами-кальками з російської. Незважаючи на те, що українська мова частково асимілює чужі слова, все одно велика кількість запозичених слів створює загрозу для зрозумілості національної терміносистеми і часто негативно впливає на швидкість навчального процесу. Втішно, що до окремих нових запозичень в українській термінології вже виникли власномовні відповідники, наприклад: трастове товариство – довірче товариство, апроксимація – наближення, детектор – виявляч та ін. Якщо така тенденція продовжиться, то більшість "модних" запозичень відійде в пасивний запас – залишаться змістовні необхідні терміни [20]. З однієї мови іншою терміни не перекладають як звичайні слова. Оптимальним є такий шлях перекладання термінів: "поняття -> український термін", а не "іншомовний термін -> український термін", з якої мови не відбувався б переклад (В. Моргунюк). Тобто пошук терміна-відповідника починається з аналізу властивостей нового поняття. На жаль, у більшості випадків переклад термінів українською мовою відбувається «калькуванням» [20, 27].

Зв’язок висвітленої проблеми із важливими науковими та практичними завданнями

Сьогодні існує багато різноманітних модифікацій морфологічних аналізаторів, переважно російських, які успішно експлуатуються в ряді промислових програмних продуктів.

Синтаксичні аналізатори ООО "Діктум" використовують аналізатори російської та англійської мови для морфологічного розбору і витягання граматичної інформації про словоформи перед

аналізом. Система контролю російської орфографії і граматики Пропись 4.0 була першим промисловим застосуванням російського морфологічного аналізатора. В системі застосовано так званий "перший клон", в якому тоді було реалізовано використовуваний і зараз підхід до організації сторінок словника, проте ще не було зафіксовано ідентифікаторів (номерів) лексем, не було можливості синтезувати форми слів за ідентифікаторами лексем і не було поняття ідентифікатора форми слова. Для опису словоформ використовували розширені адитивні граматичні описи, які досі є присутніми в структурі граматичної інформації.

Технологія морфологічного аналізу української мови зараз працює на основі пошукової машини <МЕТА>. Там же можна протестувати морфологічний аналізатор у режимі on-line [19].

Процес розвитку сучасного суспільства характеризується постійно зростаючою роллю інформаційних технологій в науці, виробництві та управлінні. В останні роки значно збільшилися обсяги інформаційних потоків і складність орієнтації в інформаційних ресурсах, що призвело до необхідності пошуку нових способів зберігання, подання, формалізації, систематизації та опрацювання інформації в комп'ютерних системах [27]. Традиційні технології в нових умовах не вирішують на належному рівні завдання навігації інформаційних ресурсів, надання доступу до інформації, пошуку файлів і документів. Одним з результатів досліджень, що проводилися за останні роки з метою подолання зазначених труднощів, стала поява онтологічних технологій та їх використання в інформаційних системах [28]. За своєю суттю онтологія предметної області є формальною моделлю структури понять предметної області [29]. У відомому формулюванні Грубера [27] онтологію визначено як «формальна специфікація концептуалізації, яка має місце в деякому контексті предметної області». Під концептуалізацією розуміємо представлення предметної області через опис множини понять (концептів) предметної області та зв'язків (відношень) між ними. Шляхом створення онтологій формується узгоджене між фахівцями формалізоване представлення структури предметної області [30].

Метою виконання роботи є розроблення інтелектуальної системи для моделювання процесів аналізу та синтезу текстів технічного характеру, а саме: перевірка правильності вживання термінів у статтях відповідно до загальновідомих правил та можливість побудови онтологій цих статей. Створена система основана на алгоритмах морфологічного аналізу, а саме: в її основу покладено модифікований морфологічний аналізатор. Відмінністю створеної системи від вже існуючих морфологічних аналізаторів є її вузька спеціалізація із пошуку термінів у технічних текстах, зокрема статтях. Об’єктом дослідження цієї роботи є морфологічні системи-аналізатори тексту. Предметом дослідження є алгоритми морфологічного аналізу, стемінгу та автоматичної побудови онтологій. Теоретичне значення роботи полягає у аналізі відомих алгоритмів морфологічного аналізу, стемінгу та методів побудови онтологій. Практичне значення отриманих результатів полягає у реалізації композиції методів морфологічного аналізу та стемінгу з метою підвищення ефективності пошуку неправильних слів та словосполучень у тексті та можливісті побудови онтологій.

Аналіз останніх досліджень та публікацій

На сучасному етапі можна простежити 5 підходів до проблем упорядкування української науково-технічної термінології [2, 5–11, 17, 19, 20, 27].

1-й підхід – формальний. Головним для нього є кількісний показник – якнайшвидше видання словника. Поквапливість у термінологічній справі не приносить користі – це в кращому випадку. У гіршому – розхитує терміносистему, подає неправильні орієнтири для користувачів. "Термінологія – це не поле для здобуття слави, для козакування. Це... натомість муравлина, забарна праця, найчастіше зовсім недооцінювана" (А. Вовк).

2-й підхід – етнографічний. Він ґрунтується на ідеї відродження національної термінології Інституту української наукової мови. Творці словників прагнуть повернути до сучасної української термінології майже всі терміни початку століття.

3-й підхід – консервативний. Його прихильники виступають за збереження української науково-технічної термінології у такому вигляді, якого вона набула за радянського часу. Це так званий принцип "реальної мови".

4-й підхід – інтернаціональний. Для нього характерне введення до української науково-технічної термінології великої кількості запозичень із західноєвропейських мов, особливо з англійської.

5-й підхід – поміркований. Він передбачає упорядкування української науково-технічної термінології з урахуванням історичних, національних, політичних чинників і вироблення її оптимального варіанту [20].

Сучасні українські термінологи глибше, ніж їх попередники початку століття, розробляють теорію терміна як мовного знака, теорію термінології як підсистеми загальнолітературної мови.

Вважається, що термінологія, як і загальнолітературна мова, повинна характеризуватися такими чинниками: досконалість, економічність, консонансність. Під досконалістю термінології дослідник розуміє її чітку граматичну структуру, логічність та вмотивованість термінів; під економічністю – її інформативність, легкість у вивченні, короткість терміно-одиниць; під консонансністю – милозвучність, артикуляційну та правописну зручність термінів [20].

Формулюються такі вимоги до терміна:

1. Змістовність – точна відповідність слова поняттю, прозора внутрішня форма терміна;
2. Пластичність, або гнучкість – здатність до творення похідних термінів;
3. Мовна досконалість – короткість, милозвучність, легкість для запам'ятовування;
4. Відповідність міжнародним нормам.

За такими критеріями радять оцінювати термінологічні одиниці Л. Петрух, Б. Рицар та інші дослідники. Під час визначення основних принципів термінотворення українські термінологи спираються на досвід європейської науки в цій справі – над виробленням образу ідеального терміна працювали такі відомі термінологи, як Е. Вюстер, Д. С. Лотте, О. О. Реформатський, Ш. Баллі та ін. З огляду на сказане, а також з урахуванням особливостей функціонування української термінології в останні десятиріччя можна окреслити коло сьогоднішніх проблем, від розв'язання яких залежатиме напрям подальшого розвитку української наукової мови [20]. Особливістю розвитку української науково-технічної термінології є посилений інтерес до термінологічних надбань Наукового товариства імені Тараса Шевченка та Інституту української наукової мови [20].

Автоматичне породження гіпотез про парадигми зміни незнайомих слів дає можливість автоматизувати процес заповнення баз. При переході до нової предметної області постає питання про неповноту морфологічного словника. Кожна предметна область використовує власну лексику. У зв'язку з цим постає питання про поповнення словників. Цей процес може бути автоматизованим, якщо наявний модуль морфологічного аналізу дає змогу передбачати лексичні параметри незнайомих слів. Для цього необхідно виділити всі слова, відсутні в наявному морфологічному словнику, і проаналізувати їх з прогнозом. Результатом аналізу є кортеж словоформи $< f_{nf}, r, P_{const}(r,s) \cup P_{var}(r,s) >$, де $f_{nf} =< s_{nf,r,} P_{var}(r,s) >$ – лексема нормальної форми, r – частина мови словоформи, s і s_{nf} – аналізований токен (рядок слова) і токен нормальної форми, а P – набори параметрів. За результатами аналізу ми можемо об'єднати всі слова, що володіють однаковими токенами нормальної форми, в єдині гіпотези. Висуваючи гіпотези, можна використовувати декілька сильних, але інтуїтивно вірних положень [1]. Існує кілька варіантів алгоритмів стемінгу, які відрізняються своєю точністю та продуктивністю.

Пошук за таблицею. Цей алгоритм використовує принцип пошуку за таблицею, в якій зібрано всі можливі варіанти слів та їх форми після стемінгу. Перевагами цього методу є простота, швидкість та зручність обробки винятків з мовних правил. Недоліками є те, що таблиця пошуку має містити всі форми слів: тобто алгоритм не працюватиме з новими словами (а як відомо, "живі" мови постійно поповнюються новими словами) і розміри такої таблиці можуть бути істотними. Для мов з відносно простою морфологією, таких як англійська, розміри таблиці пошуку доволі скромні, проте у аглютинативних мовах, наприклад, турецькій, кількість варіантів слів з одним коренем може сягати сотень. Цей алгоритм оснований на правилах, за якими можна скорочувати слово. Якщо взяти приклад з алгоритму пошуку за таблицею, то ці правила можуть мати такий вигляд: слово закінчується на "льна" – відсікаємо від слова "ьна"; слово закінчується на "льне" – відсікаємо "ьне"; слово закінчується на "льний" – відсікаємо "ьний"; слово закінчується на "льним" – відсікаємо "ьним".

Таблиця 1

Фрагмент таблиці пошуку слова "безпритульний"

Слово	Стемінг
безпритульна	безпритул
безпритульне	
безпритульний	
безпритульним	
безпритульними	
безпритульних	
безпритульні	
безпритульній	
безпритульнім	

Відсікання закінчень та суфіксів. Кількість таких правил стемінгу набагато менша за таблицю з усіма словоформами, а тому алгоритм є доволі компактним та продуктивним. Наведені вище 4 правила правильно опрацьовують такі прикметники:

Таблиця 2

Результат роботи алгоритму відсікання закінчень та суфіксів

Слово	Стемінг
безпритульна	безпритул
повільне	повіл
ортогональний	ортогонал
цивільним	цивіл

Проте алгоритм може робити хибні висновки і спотворювати форму стемінгу. Наприклад, слово "пальне" перетвориться на "пал" замість правильної форми "пальн". Тому, враховуючи особливості мови, набір правил із відсікання закінчень та суфіксів може бути доволі складним. До недоліків також належать обробка винятків, коли базові слова мають змінну форму. Наприклад, слова "бігом" та "біжу" повинні мати після стемінгу однаковий вигляд "біг", але простим відсіканням закінчення це неможливо зробити. Алгоритм вимушений враховувати такі ситуації, і це ускладнює правила та врешті-решт негативно впливає на ефективність. Для вирішення цієї проблеми в створеній системі використовуємо більш комплексний підхід, що ґрунтується на визначенні основи слова лематизацією. Першим кроком цього алгоритму є визначення частин мови у реченні – так званий POS tagging. На другому кроці до слова застосовують правила стемінгу відповідно до частини мови. Тобто слова "пальне" та "вітальне" мають проходити через різні ланцюжки правил, тому що "пальне" – іменник, а "вітальне" – прикметник. Теоретично алгоритми стемінгу, основані на лематизації, повинні мати дуже високу якість і мінімальний відсоток помилок, але вони дуже залежні від правильності розпізнавання частин мови.

Одним із сучасних варіантів реалізації безсловникової морфології в чистому вигляді є стемер Портера. У ньому рядок, що подається на вхід, перевіряється на наявність заданих постфіксів, причому постфікси перевіряються в певній послідовності, а частина постфіксів може комбінуватися. Все, що залишилося після послідовного «відкидання», оголошується стемом. Залежно від знайденого постфікса слово можна віднести до тієї чи іншої частини мови, хоча в переважній більшості завдань цього не потрібно. Алгоритм гранично простий, володіє дуже високою швидкодією, проте дає великий відсоток помилок. Крім того, поділ на постфікси є значною мірою спірним. Також алгоритм видає єдиний варіант розбору, повністю приховуючи омонімію слів. Алгоритм Портера дуже слабко враховує той факт, що для різних частин мови і навіть для різних парадигм перед постфіксом можуть стояти різні літери. Цей факт використовується в системі морфологічного аналізу Stemka, де зберігаються не тільки самі постфікси, а й ще дві попередні букви псевдооснови. Самі комбінації букв і постфіксів зберігаються у вигляді кінцевого автомата справа наліво [5]. Суттєвим плюсом безсловникових морфологій є те, що вони можуть видати

результат для будь-яких слів, що зустрічаються в тексті, що дуже зручно при аналізі текстів з незнайомою предметною областю або таких, що містять багато нелітературних або рідко вживаних слів. Проте коректність отримуваних результатів знаходиться на рівні 90–95 %. Це привело до відмови від безсловникових морфологій у задачах, коли точність аналізу повинна переважати над його повнотою, і до переходу до використання словникових морфологій у таких завданнях, як машинний переклад і діалогові системи. Однак на практиці існує велика кількість завдань, що розв'язуються статистичними методами, в яких цілком достатньо приблизного знання про зв'язки між словами. Це завдання рубрикації, інформаційного пошуку, частково завдання реферування, ряд інших завдань. Методи безсловникових морфологій активно використовуються в словникових морфологіях для передбачення нормальної форми і набору параметрів слів, які відсутні в морфологічному словнику, а також для розширення словника [1].

Варіанти стемінгу для української мови існують і використовуються у складі комерційних пошукових систем. На жаль, наразі відсутня вільна реалізація подібних алгоритмів. Слід зазначити, що певні кроки у цьому напрямку вже зроблено, зокрема це модуль Drupal для української мови, який перебуває на стадії розроблення та пошукова система "<META>", в якій використовується модифікований метод стемінгу для відсікання закінчень та суфіксів невідомих слів, тому поява некомерційного алгоритму стемінгу для української мови – це справа часу [19].

Перший закон Ципфа («ранг – частота»). Ще однією особливістю створеної системи є можливість визначення ключових слів статей та порівняння їх із ключовими словами, які вказав автор. Цього вдалося досягти використанням методу, що ґрунтується на законі Ципфа.

Виміряємо кількість входжень кожного слова в текст і візьмемо тільки одне значення з кожної групи, що має однакову частоту. Розташуємо частоти у міру їх зменшення і пронумеруємо, порядковий номер частоти назвемо рангом частоти (позначимо r_i ранг слова *i*). Слова, які найчастіше зустрічаються, матимуть ранг 1, наступні за ними – 2 і так далі.

Тоді очевидно, що ймовірність зустріти довільне, заздалегідь вибране слово дорівнюватиме відношенню кількості входжень цього слова до загальної кількості слів у тексті (n_i –кількість входжень *i* слова, $|N|$ – кількість слів у тексті).

$$p = n_i / |N|. \tag{1}$$

Ципф виявив таку закономірність: добуток ймовірності виявлення слова в тексті і рангу частоти є постійним числом (С):

$$\frac{n_i \cdot r_i}{|N|} = C. \tag{2}$$

Закон показує, що поширеність слова в тексті змінюється за гіперболою, залежно від кількості входжень. Наприклад, друге з використовуваних слів зустрічається приблизно удвічі рідше, ніж перше, третє – утричі рідше, ніж перше, і так далі.

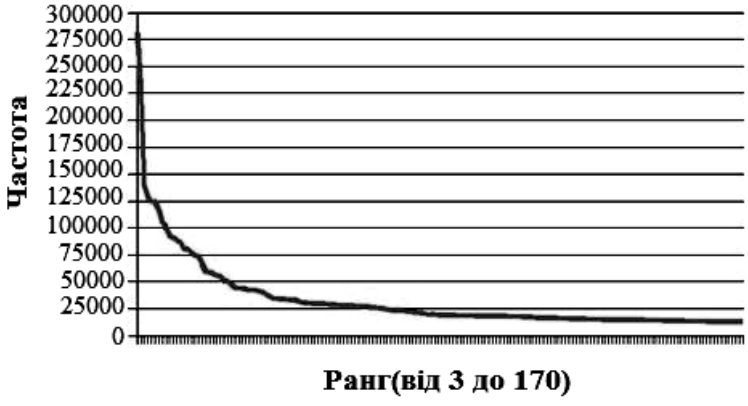

Рис. 1. Залежність частоти вживання слова від його рангу

Значення константи в різних мовах різне, але в межах однієї мовної групи залишається приблизно незмінним, який би текст ми не взяли. Джордж Ципф та інші дослідники встановили, що гіперболічному розподілу підпорядковуються не тільки всі природні мови світу, але й інші явища соціального і біологічного характеру: розподіл вчених за кількістю опублікованих ними статей, міст США за чисельністю населення, населення за розмірами доходу в капіталістичних країнах та інші. Закони Ципфа дають змогу знаходити ключові слова [9].

Скористаємося першим законом Ципфа і побудуємо графік залежності рангу від частоти. Дослідження показують, що найзначущіші для тексту слова знаходяться в середній частині графіка (рис. 2.). Цей факт має просте обґрунтування. Слова, які трапляються дуже часто, є переважно службовими. Також рідко зустрічаються слова, які в більшості випадків не мають вирішального значення для інформації, яку подано у статті. Від встановлення ширини діапазону залежить якість виділення значущих слів. Якщо взяти велику ширину діапазону, то ключовими словами потраплятимуть допоміжні слова; якщо встановити вузький діапазон – можна втратити смислові терміни. Тому в кожному окремому випадку необхідно використовувати ряд евристик, для визначення ширини діапазону, а також методик, що зменшують вплив цієї ширини.

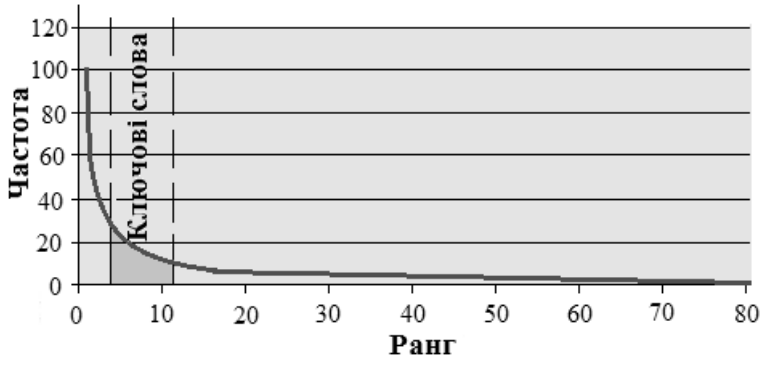

Рис. 2. Розташування ключових слів

Одним із способів, наприклад, є попереднє вилучення з досліджуваного тексту слів, які спочатку не можуть бути значущими і тому є «шумом». Такі слова називаються нейтральними, або стоповими (стоп-словами). Для українськомовного тексту стоповими словами можуть бути всі прийменники, частки, особові займенники. Є й інші способи підвищити точність оцінки значущості слів [9]. Деякі слова можуть зустрічатися майже у всіх документах деякої колекції і, відповідно, мало впливати на приналежність документа до тієї чи іншої категорії, а отже, не бути ключовими для цього документа. Тому очевидно, що, розглядаючи усю колекцію документів, ми підвищимо інформативність виділення ключових слів.

Методи побудови онтологій предметної області такі [29–57]:

- побудова онтологій перетворенням XML-подібних документів;
- використання готових словників;
- застосування лінгвістичного аналізу текстів, написаних природною мовою;
- застосування кластеризації та аналізу формальних концептів.

Автоматичне видобування знань з монологічних текстів з метою побудови онтології передбачає не тільки виявлення термінів, але й пошук у тексті знань про ці терміни. Це означає, що для опису семантичної структури термінології необхідно розпізнати в тексті як терміни, так і семантичні відношення між термінами.

Виділення проблем

Розглядаючи можливість автоматизації різних стадій автоматичної генерації онтології [29–57], дійшли таких висновків. Підготовчу стадію та стадію серіалізації у всіх випадках можливо повністю автоматизувати, оскільки ці процеси є повністю тривіальними і зводяться до примітивних операцій над рядковими даними або ж перетворення деревоподібних структур даних на деякий

XML-подібний формат. Стадію аналізу також можна ефективно автоматизувати. Побудова концептів, таксономічних відношень між концептами та відношень належності екземплярів до класів автоматизується у разі застосування всіх описаних вище методів генерації онтологій. Можливість повної автоматизації побудови нетаксономічних відношень все ще залишається відкритим питанням, крім того, наявні методи залежать від мови текстових даних, які опрацьовуються. Відкритим є також питання повної автоматизації називання концептів, особливо це стосується методів автоматичної генерації онтологій, основаних на використанні ієрархічної кластеризації та формального аналізу концептів. Цю проблему пропонується вирішити за допомогою ситуаційного підходу з частковим використанням словника [42].

Стадія валідації тією чи іншою мірою потребує втручання експерта. Винятком є випадки, коли онтологія генерується на основі набору XML-документів або ж підмножини записів деякого словника або тезауруса. В цьому контексті засоби на зразок WordNet, без сумніву, заслуговують на особливу увагу через великі можливості для автоматизації валідації. Попри те, що WordNet має занадто широке призначення і не може бути адаптованим до певної предметної області людської діяльності, використання цього методу для валідації та узгодження онтологій є цікавою темою для подальшого розроблення. Стадія розширення також потребує особливої уваги при автоматизації. Особливо це стосується випадку її зведення до узгодження і злиття онтологій. Процес злиття онтологій тісно пов'язаний з узгодженням. Існують методи його реалізації для двох вхідних онтологій, однак одночасне злиття кількох онтологій залишається відкритим питанням. Цю проблему можна вирішити послідовним злиттям, але в такому випадку кінцева онтологія залежатиме від вибору порядку злиття. У деяких випадках додавати нові сутності та зв'язки до онтології можна за методом, відмінним від того, який використовували на початковому етапі [44].

Формулювання мети

Метою є дослідження проблеми автоматизації перевірки правильності вживання термінів у тексті, інтелектуального визначення ключових слів статей, розроблення системи для моделювання процесів аналізу та синтезу текстів технічного характеру, а саме: перевірки правильності вживання термінів у статтях відповідно до загальновідомих правил та формування переліку ключових слів статей з можливістю перевірки правильності переліку вказаних автором ключових слів статті. Створена система ґрунтується на алгоритмах морфологічного аналізу, в її основу покладено модифікований морфологічний аналізатор, побудований за принципами стемінгу та лематизації. Отже, результатом буде розроблення інтелектуальної системи моделювання процесів аналізу та синтезу технічного тексту. Перед проектуванням інтелектуальної системи моделювання процесів аналізу та синтезу технічного тексту необхідно спершу побудувати дерево цілей системи, яке забезпечить можливість виконати послідовні та коректні дії під час проектування інтелектуальної системи [4, 12, 14, 18, 21–23, 25, 26, 28].

Головною ціллю є розроблення інтелектуальної системи моделювання процесів аналізу та синтезу технічного тексту [4, 12, 14, 18, 22, 23]. Досягнення головної цілі можливе лише за умови виконання всіх підцілей. Головна ціль розроблюваної системи поділяється на чотири підцілі (рис. 3).

Першою підціллю є «Зчитати текст». Метою є зчитування вхідних даних, над якими будуть виконуватимуться всі наступні операції. Другою підціллю є «Виконати синтаксичний аналіз тексту». Ця підціль поділяється на дві підцілі: «Знайти стоп-слова» та «Видалити стоп-слова». Метою є «очищення» вхідного тексту від «шуму» (слів, які не несуть жодної смислової інформації). Третьою підціллю є «Виконати аналіз тексту та окремих слів». Ця ціль поділяється на чотири підцілі: «Знайти неправильний термін у тексті», «Опрацювати текст і ключові слова», «Опрацювати термін» та «Виконати морфологічний аналіз терміна». Підціль «Опрацювати текст і ключові слова» поділяється на чотири підцілі: «Побудувати алфавітно-частотний словник», «Знайти ключові слова, вказані автором», «Визначити ключові слова» та «Перевірити правильність вибраних ключових слів». Виконання цих чотирьох підцілей необхідне для пошуку ключових слів, вказаних автором статті, пошуку ключових слів у тексті аналітичним методом та порівняння результатів з метою перевірки правильності вказаних автором ключових слів.

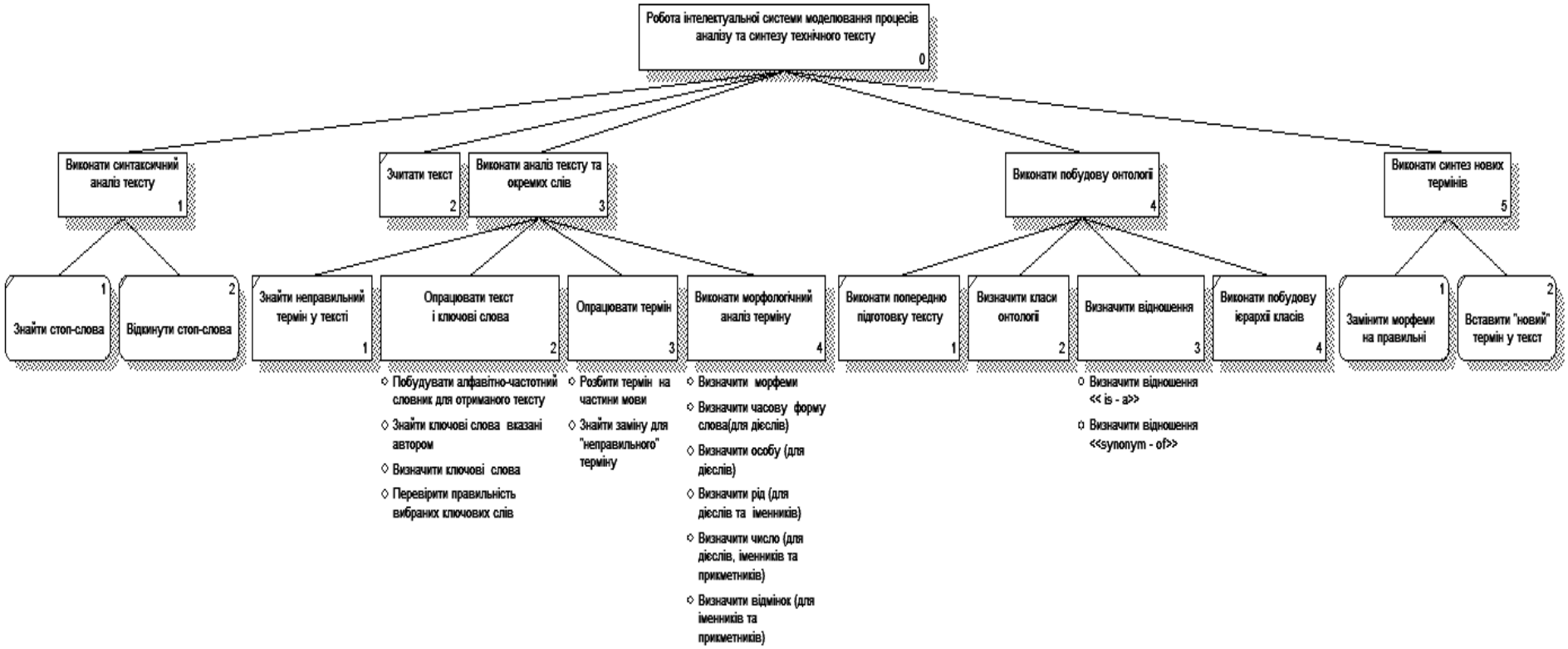

Рис. 3. Дерево цілей функціонування інтелектуальної системи моделювання процесів аналізу та синтезу технічного тексту

Підціль «Опрацювати термін» поділяється на дві підцілі: «Розбити термін на частини мови» та «Знайти заміну неправильного терміна». Досягнення цих цілей гарантує підготовку знайденого терміна перед морфологічним аналізом. Підціль «Виконати морфологічний аналіз терміна» поділяється на шість підцілей: «Визначити морфеми», «Визначити часову форму (для дієслів)», «Визначити особу (для дієслів)», «Визначити рід (для дієслів та іменників)», «Визначити число (для дієслів, іменників та прикметників)», «Визначити відмінок (для іменників та прикметників)». Досягнення цих підцілей забезпечує виконання процесу морфологічного аналізу, який є одним з ключових процесів, необхідних для функціонування всієї системи.

Підціль «Виконати побудову онтології» поділяється на чотири підцілі: «Виконати попередню підготовку тексту», «Визначити класи онтології», «Визначити відношення», «Виконати побудову ієрархії класів». Досягнення цих підцілей забезпечує виконання процесу побудови онтології.

Підціль «Виконати синтез нових термінів» поділяється на дві підцілі: «Змінити морфеми на правильні», «Вставити новий термін у текст». Досягнення цих цілей забезпечує заміну морфем терміна на нові, згідно з отриманими характеристиками введеного користувачем тексту, та вставку назад у текст «нового» відкоригованого правильного терміна.

Досягнення головної цілі не можливе без послідовного виконання кожної з підцілей.

Аналіз отриманих наукових результатів

Контекстна діаграма є першою в ієрархії діаграм нотації IDEF0, на ній зображено функціонування системи загалом (рис. 4) [13, 15, 16, 24]. Цю модель описано з погляду користувача.

Рис. 4. Контекстна діаграма А0

З погляду користувача функціонування цієї системи відбувається так:

- на вхід системи подається фрагмент тексту;
- на виході системи отримуємо відредагований фрагмент тексту, в якому «правильно» вжито технічний термін, перелік ключових слів, обчислений системою, результат перевірки правильності визначених автором статті ключових слів та онтологію;
- система керується правилами чинного українського правопису редакції інституту мовознавства ім. О. О. Потебні НАН України та інституту української мови НАН України від 2007 р., які схвалені Міністерством освіти і науки України;
- ресурсами системи є Адміністратор з правами редагування конфігурації системи та різноманітних правил аналізу, Модератор з правами редагування баз даних, Користувач та середовище програмування.

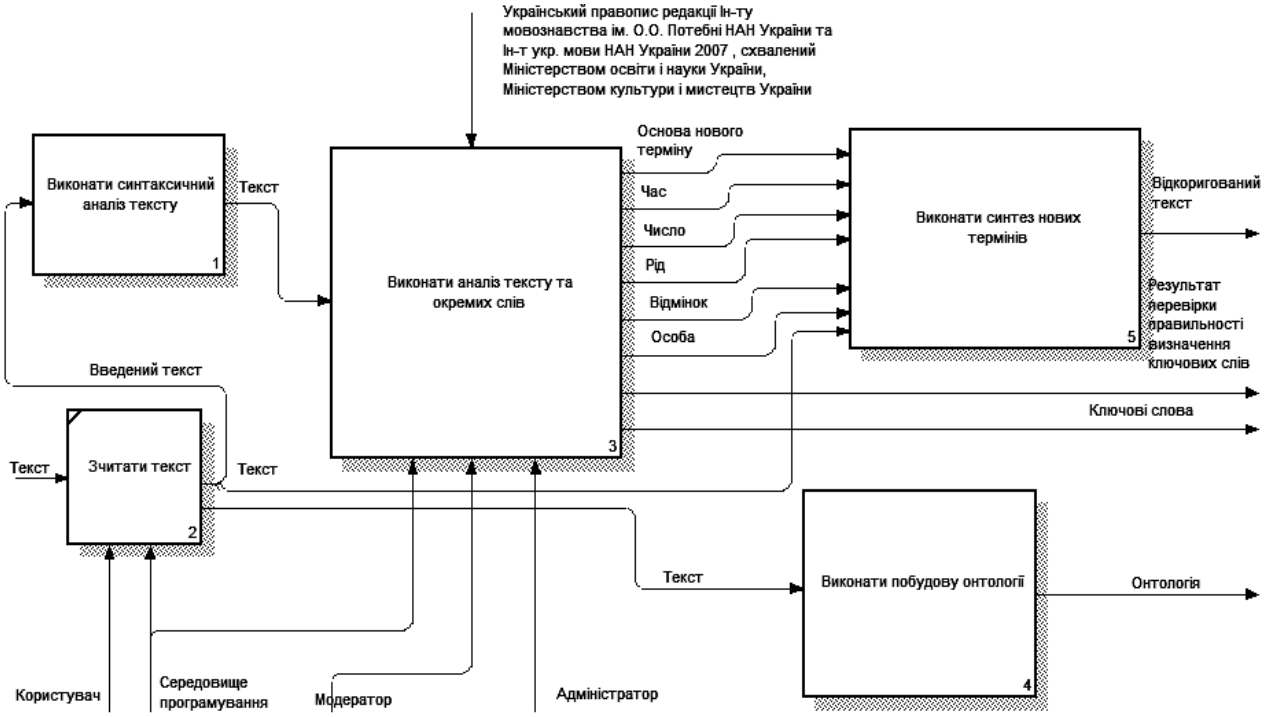

Рис. 5. IDEF0. Діаграма A0. Декомпозиція системи

Діаграма, зображена на рис. 5, є першим кроком декомпозиції. На ній можна детальніше розглянути процеси системи, а саме *«Зчитати текст»*, *«Виконати синтаксичний аналіз тексту»*, *«Виконати аналіз тексту та окремих слів»* та *«Виконати синтез нових термінів»*. Процес *«Зчитати текст»* є першим з процесів, які виконуються у цій системі. Вхідними даними для цього процесу є текст, введений користувачем, а саме: стаття технічного характеру. Цей текст опрацьовується середовищем програмування та передається для подальшого опрацювання процесами *«Виконати синтаксичний аналіз тексту»* та *«Виконати синтез нових термінів»*.

Рис. 6. Діаграма потоків даних A1. Декомпозиція процесу «Виконати синтаксичний аналіз тексту»

На рис. 6 зображено діаграму потоків даних. Ця діаграма є кінцевим кроком декомпозиції процесу *«Виконати синтаксичний аналіз тексту»*. На ній зображено процеси обміну даними між роботами *«Знайти стоп-слова»*, *«Відкинути стоп-слова»* та накопичувачем *«База термінів та частин мови»*. На вхід роботи *«Знайти стоп-слова»* подається введений користувачем текст, над яким виконуються операції пошуку стоп-слів, далі знайдені стоп-слова передаються роботі *«Відкинути стоп-слова»*, де відбувається їх вилучення з тексту та додавання до накопичувача *«База термінів та частин мови»* за умови, що такі слова там відсутні. На виході роботи *«Відкинути стоп-слова»*, отримуємо текст, *«очищений»* від *«шуму»*, який передається далі для виконання наступних операцій.

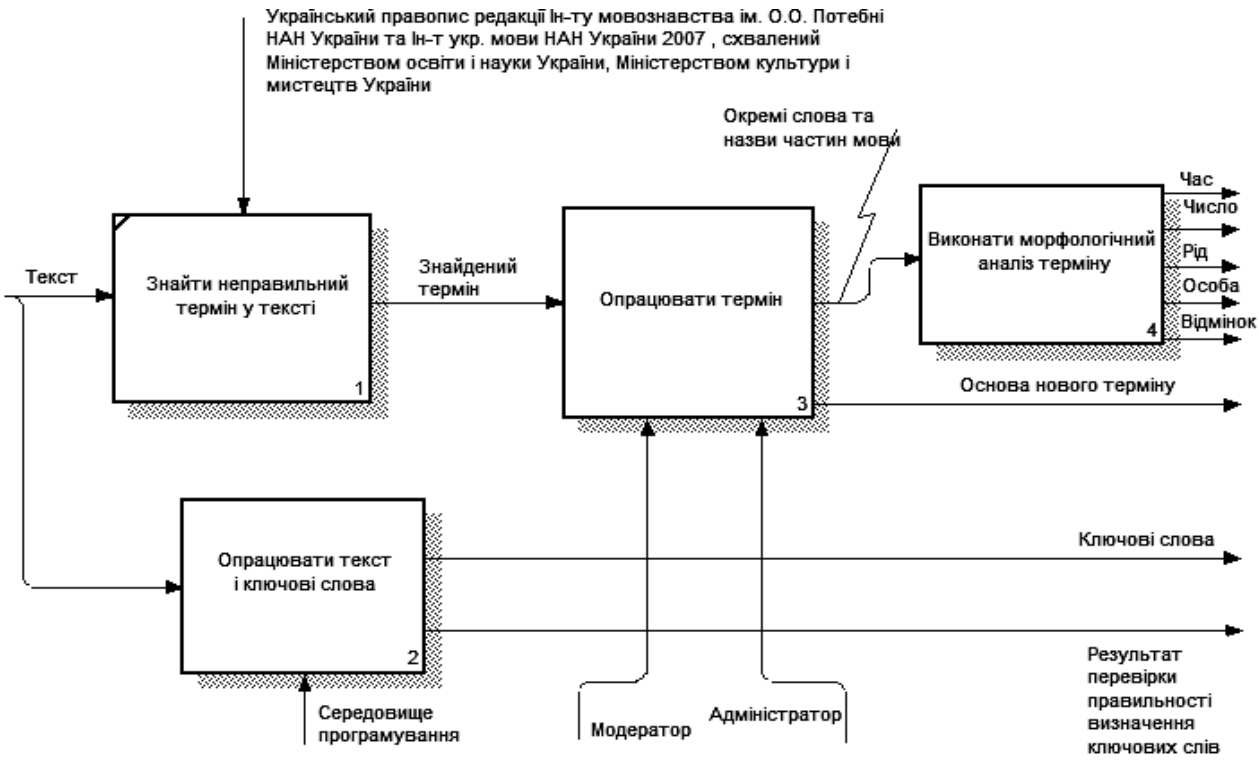

Рис. 7. IDEF0 A3. Декомпозиція процесу «Виконати аналіз тексту та окремих слів»

На рис. 7. зображено IDEF0 декомпозиції процесу «*Виконати аналіз тексту та окремих слів*». На вхід процесів «*Знайти неправильний термін у тексті*» та «*Опрацювати текст і ключові слова*» подаються дані у вигляді тексту, який опрацьований попереднім процесом «*Виконати синтаксичний аналіз тексту*». Далі процесом «*Знайти неправильний термін у тексті*», що керується правилами українського правопису, виконується пошук у тексті неправильних термінів. На виході цього процесу отримуємо дані у вигляді слова або словосполучення (терміна), яке далі передається для опрацювання процесу «*Опрацювати термін*».

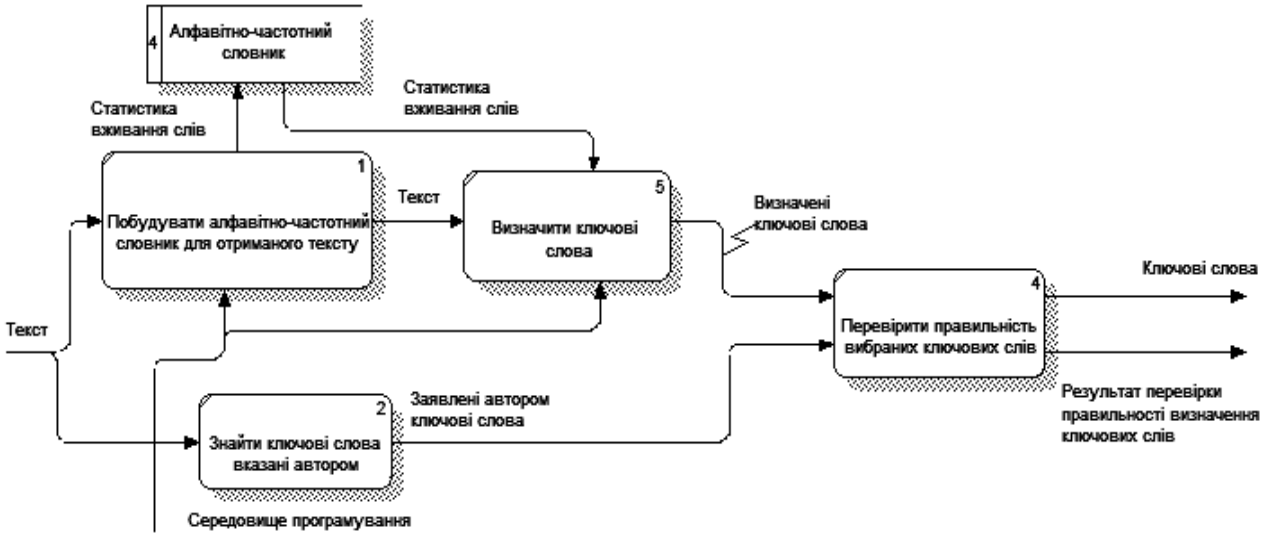

Рис. 8. Діаграма потоків даних A32. Декомпозиція процесу «Опрацювати текст і ключові слова»

На рис. 8 зображено діаграму потоків, яка є декомпозицією процесу «*Опрацювати текст і ключові слова*». Ця діаграма складається з робіт: «*Побудувати алфавітно-частотний словник для отриманого тексту*», «*Знайти ключові слова, вказані автором*», «*Визначити ключові слова*», «*Перевірити правильність вибраних ключових слів*» та накопичувача «*Алфавітно-частотний словник*». На вхід

робіт *«Побудувати алфавітно-частотний словник для отриманого тексту»* та *«Знайти ключові слова, вказані автором»* подається текст, опрацьований попереднім процесом. Роботою *«Знайти ключові слова, вказані автором»* виконується пошук ключових слів статті, вказаних автором згідно із правилами оформлення статей. На виході цієї роботи отримуємо дані у вигляді знайдених ключових слів.

Робота *«Побудувати алфавітно-частотний словник для отриманого тексту»* виконує заповнення накопичувача *«Алфавітно-частотний словник»* словами та частотою їх вживання у поданому на вхід тексті. На виході роботи *«Побудувати алфавітно-частотний словник для отриманого тексту»* отримуємо незмінений вхідний текст, який далі передається для опрацювання роботою *«Визначити ключові слова»*.

Робота *«Визначити ключові слова»* отримує на вхід текст та на основі сформованих у накопичувачі *«Алфавітно-частотний словник»* статистичних даних формує перелік ключових слів для поданого на вхід тексту. Далі ці ключові слова та ключові слова, отримані внаслідок виконання роботи *«Знайти ключові слова, вказані автором»*, передаються для опрацювання роботою *«Перевірити правильність вказаних ключових слів»* .

Робота *«Перевірити правильність вказаних ключових слів»* виконує порівняння вказаних автором статті ключових слів із ключовими словами, знайденими системою в отриманому тексті. На виході роботи *«Перевірити правильність вказаних ключових слів»* отримуємо результат перевірки та перелік ключових слів, знайдених системою за статистичними даними.

На рис. 9 зображено діаграму потоків даних, яка є декомпозицією процесу *«Опрацювати термін»*. Ця діаграма складається з таких робіт: *«Розбити термін на частини мови»,«Знайти заміну для «неправильного» терміна»* та накопичувачів *«Правила аналізу терміна», «База термінів та частин мови»*.

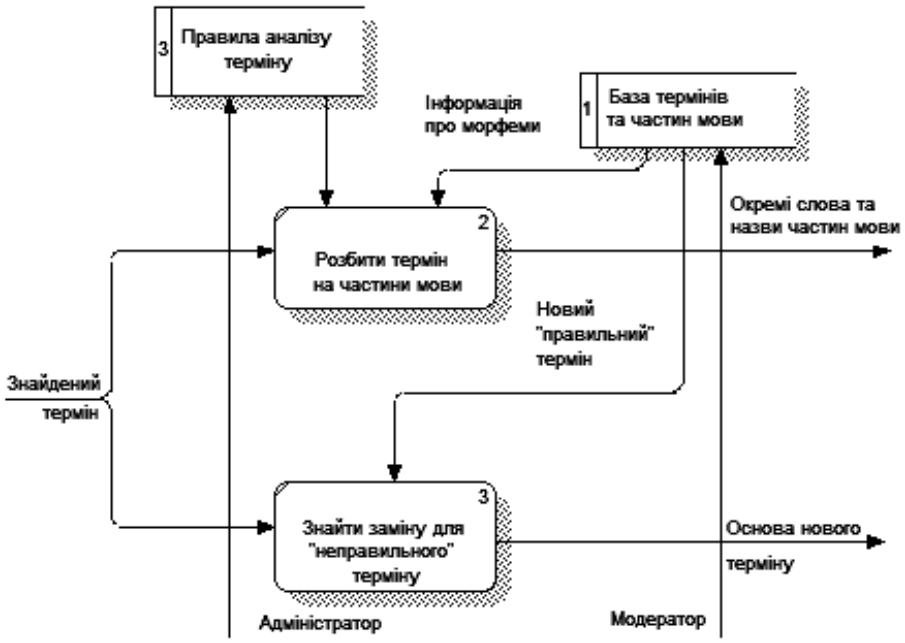

Рис. 9. Діаграма потоків даних А33. Декомпозиція процесу «Опрацювати термін»

На вхід робіт *«Розбити термін на частини мови»* та *«Знайти заміну для неправильного терміна»* подаються дані у вигляді знайдених внаслідок виконання процесу *«Знайти неправильний термін у тексті»* термінів. Роботою *«Розбити термін на частини мови»*, за правилами, збереженими в накопичувачі *«Правила аналізу терміна»*, виконують розбиття у випадку, якщо термін – словосполучення, та класифікацію окремих слів на частини мови. Права для редагування та додавання нових правил має лише *Адміністратор* системи. На виході роботи *«Розбити термін на частини мови»* отримуємо перелік класифікованих за частинами мови слів, що передаються до наступного процесу для опрацювання.

Робота *«Знайти заміну для «неправильного» терміна»* шукає *«неправильний»* термін у тексті за даними, що містяться у накопичувачі *«База термінів та частин мови»*. Права для зміни та додавання даних до накопичувача *«База термінів та частин мови»* надаються лише Модератору системи.

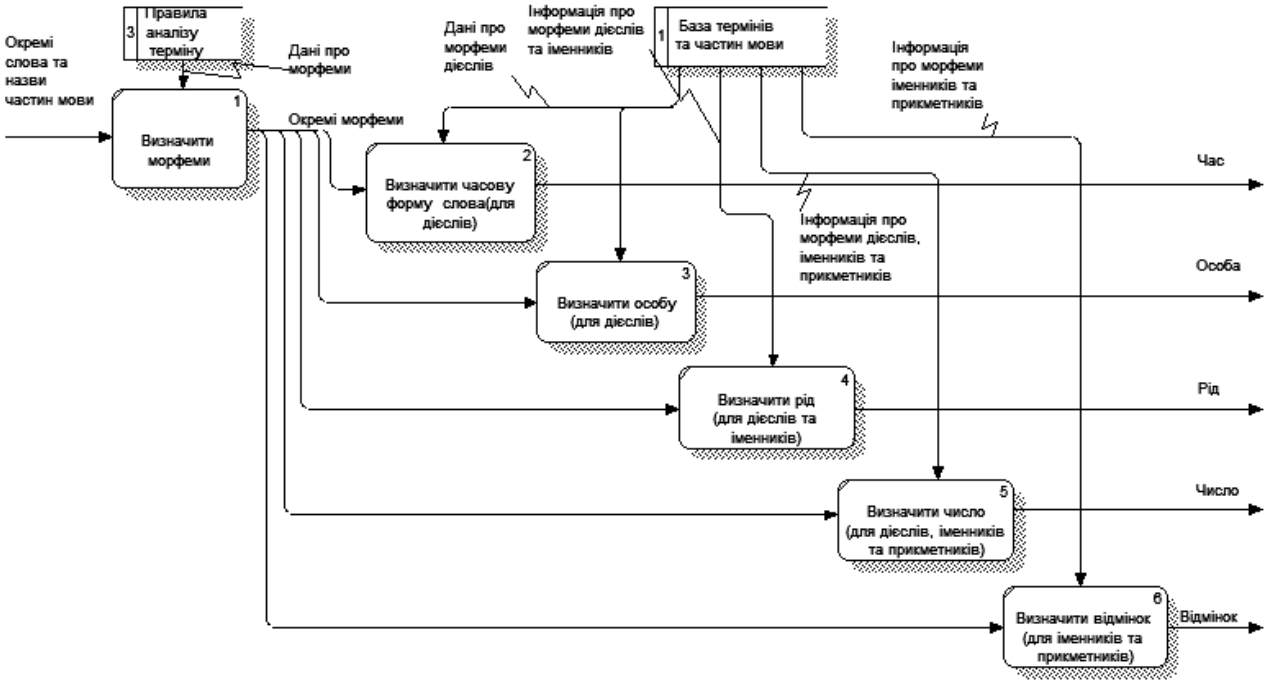

Рис. 10. Діаграма потоків даних А34. Декомпозиція процесу «Виконати морфологічний аналіз терміна»

На рис.10 зображено діаграму потоків даних, яка є декомпозицією процесу *«Виконати морфологічний аналіз терміна»*. Ця діаграма складається з таких робіт: *«Виділити морфеми»*, *«Визначити часову форму слова (для дієслів)»*, *«Визначити особу (для дієслів)»*, *«Визначити рід (для дієслів та іменників)»*, *«Визначити число(для дієслів, іменників та прикметників)»*, *«Визначити відмінок (для іменників)»* та накопичувача *«База термінів та частин мови»*.

На вхід роботи *«Визначити морфеми»* подаються дані у вигляді слів, класифікованих за частинами мови. Потім за даними з накопичувача *«Правила аналізу терміна»* виділяють морфеми. Після цього знайдені морфеми передають для опрацювання однією або декількома з робіт *«Визначити часову форму слова(для дієслів)»*, *«Визначити особу (для дієслів)»*, *«Визначити рід(для дієслів та іменників)»*, *«Визначити число(для дієслів, іменників та прикметників)»*, *«Визначити відмінок (для іменників)»* з метою визначення поточних характеристик слів терміна. Потім отримані дані передаються для опрацювання впродовж наступного процесу.

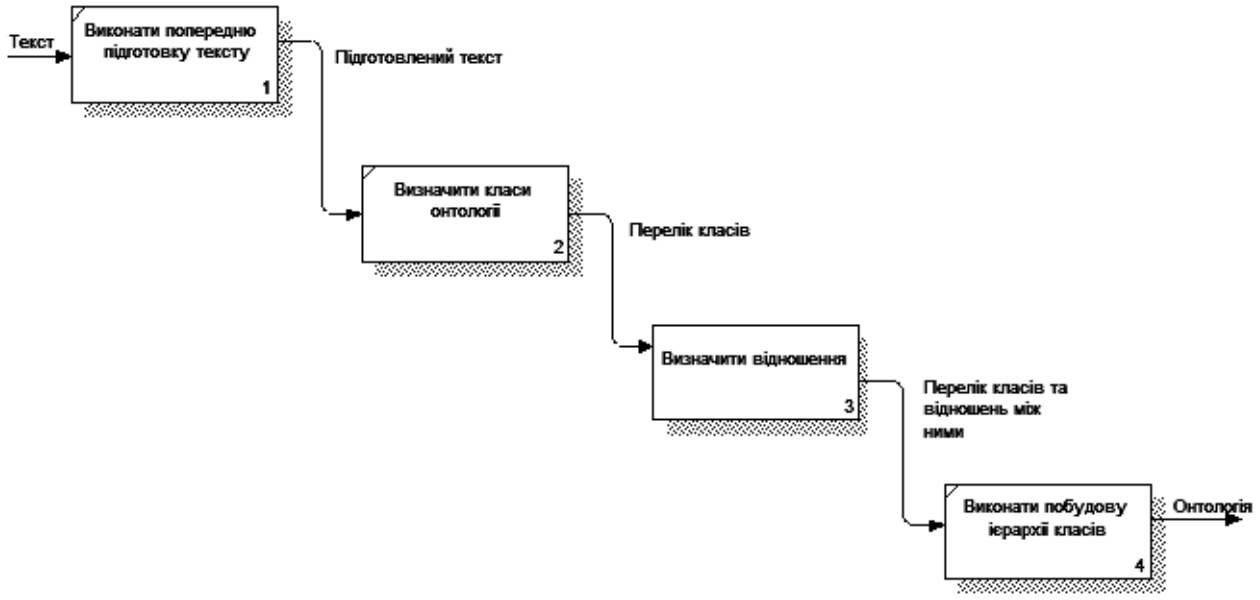

Рис. 11. Діаграма потоків даних А4. Декомпозиція процесу «Виконати побудову онтології»

На рис. 11 зображено IDEF0 декомпозиції процесу «*Виконати побудову онтології*». На вхід процесу «*Виконати попередню підготовку тексту*» подається текст статті для попереднього опрацювання. Підготовлений текст передається процесу «*Визначити класи онтології*» для визначення основних класів. Потім перелік визначених класів онтології передається процесу «*Визначити відношення*» для побудови відношень. На виході цього процесу отримуємо перелік класів та відношень між ними; ці дані передаються процесу «*Виконати побудову ієрархії класів*» для побудови ієрархії класів.

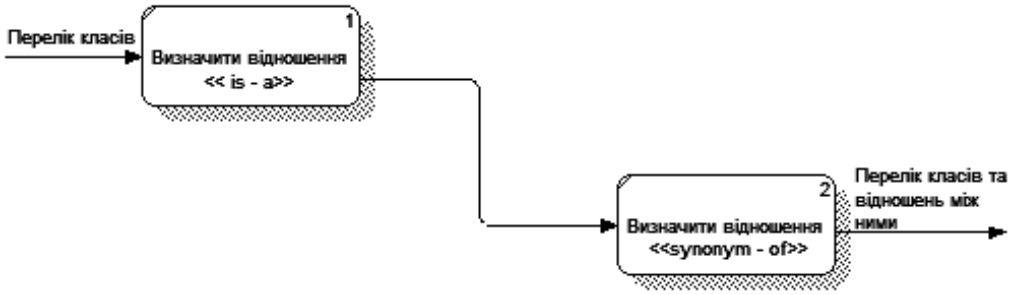

Рис. 12. Діаграма потоків даних А43. Декомпозиція процесу «Визначати відношення»

На рис. 12 зображено діаграму потоків даних, яка є декомпозицією процесу «*Визначити відношення*». Ця діаграма складається з робіт: «*Визначити відношення «is-a»*» та «*Визначити відношення «synonym-of» »*.

На вхід роботи «*Визначити відношення «is-a» »* подаються дані у вигляді переліку класів, який сформований на попередньому кроці. Потім визначаються відношення типу «*is-a*» за допомогою відповідного продукційного правила. На виході *роботи «Визначити відношення «is-a»* перелік класів та відношень між ними типу «*is-a*».

Робота «*Визначити відношення «synonym-of» »* визначає відношення типу «*synonym-of*» за відповідним продукційним правилом. На виході *роботи «Визначити відношення «synonym-of» »* перелік класів та відношень між ними типу «*is-a*» *та «synonym-of»*.

На рис. 13 зображено діаграму потоків даних, яка є декомпозицією процесу «*Виконати синтез нових термінів*». Ця діаграма складається з робіт: «*Замінити морфеми на правильні*», «*Вставити «новий» термін у текст*» та накопичувача «*База термінів та частин мови*».

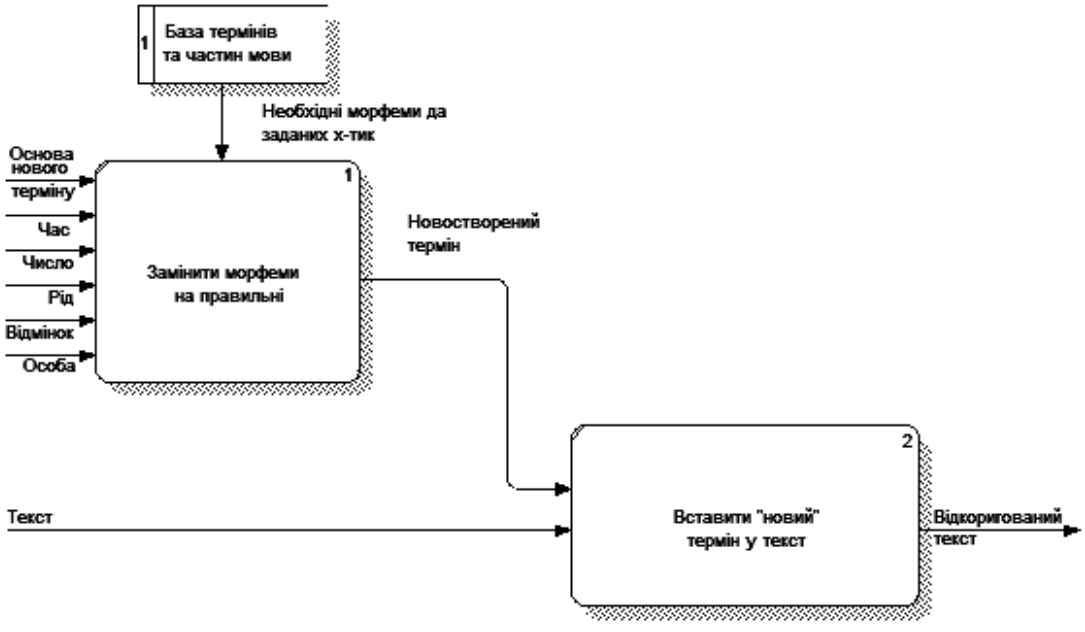

Рис. 13. Діаграма потоків даних А5. Декомпозиція процесу «Виконати синтез нових термінів»

На вхід роботи *«Змінити морфеми на правильні»* подаються дані у вигляді переліку характеристик залежно від частини мови, окремого слова та основи слова, яким слід змінити неправильний термін. Потім за цими даними виконують пошук і додавання відповідних морфем до основ окремих слів нового терміна з накопичувача *«База термінів та частин мови»*. На виході *роботи «Змінити морфеми на правильні»* отримуємо новостворений *«правильний»* термін, який передається на вхід роботи *«Вставити «новий» термін у текст»*.

Робота *«Вставити новий термін у текст»* додає новостворений термін у текст на позицію *«неправильного»* терміна. Вихід роботи *«Вставити «новий» термін у текст»* є одним з виходів всієї системи, а саме, відредагований текст з *«правильними»* термінами.

Речення в тексті природною мовою є деяким твердженням. Для розпізнавання семантики тверджень часто застосовується ситуаційний підхід [45]. Його використання грунтується на тому, що на практиці важко створити несуперечливу і повну базу знань. У ситуаційній семантиці висновки робляться тільки в межах ситуації, що виникла в цей момент. Переходячи до іншої ситуації, ревізують базу знань, і твердження, виведені раніше, не використовують для виведення нових.

При реалізації ситуаційного підходу за основу взято ідею Л. Вітгенштайна щодо правил вживання слів: залежно від ситуації слово вживається так чи інакше. Фактично він сформулював прагматичне розуміння сенсу (сенс інтерпретується як адекватна реакція на ситуації, що виникають). Якщо при цьому в процесі міркувань використати логічний висновок, то, задаючи синтаксичні обмеження у вигляді кон'юнкції фактів, які додатково описують ситуацію, що характеризується двома семантичними властивостями – несуперечністю і мінімальністю, – можна побудувати повну і несуперечливу базу знань. В.Н. Вагін [16, 17] пояснює ідею Л. Вітгенштайна так. Якщо є множина (контекстів) ситуацій, то маємо таке відображення $F : S \to CONT(T)$, де S – множина ситуацій, *CONT (T)* – множина висловлювань, що утворює зміст ідеї, яка позначається термом *T*. При цьому $y = F(s) \in CONT(T)$ – аспект змісту. Отже, кожній ситуації *s* зіставляється деякий елемент змісту *y*, що відповідає нашій інтуїції: коли ми вживаємо деякий термін, то реалізуємо інкорпорацію його змісту (іноді беремо з нього все, а іноді лише деяку частину). Тому ми ніби реагуємо на ту ситуацію, яка виникає. Введемо відношення попереднього порядку $\leq$ (транзитивне і рефлексивне відношення) на множині *CONT (T)*. Тоді для ідеї, яку можна подати за допомогою *T*, побудовано деяку організацію знань *CONT (T)*, яка створює можливість огляду або охоплення (розуміння) в сенсі К. І. Льюіса .

Розвиток цього підходу передбачає дослідження можливих ситуацій *s*, в яких можуть знаходитися компоненти речення. Можливі ситуації – це той чи інший попередній порядок компонентів. Виконуючи морфологічний або синтаксичний аналіз, витягуючи знання про терміни з термінологічних словників або застосовуючи інші методи аналізу природномовного тексту, завжди досліджуються ситуації, за яких у лексемах знаходяться морфеми, в реченнях – лексеми, в тексті – речення і т.д. Отже, методи опрацювання природномовного тексту майже завжди спрямовані на аналіз ситуаційного контексту і залежно від репрезентованого методу об'єктом цього аналізу є або текст, або фрагмент тексту, або лексема речення, або морфема лексеми.

Відповідно до вищевикладеного, для вирішення різних завдань опрацювання монологічного природномовного тексту необхідно розробити методи їх вирішення, грунтуючись на ситуаційному моделюванні. В основу ситуаційного моделювання покладено просту ядерну конструкцію мови *skk=xRy*, де *x*, *y* – терміни, *R* – семантичне відношення між ними.

Розглянемо тепер структуру продукційного правила, яке прийнято описувати сімкою [83, 85] :

$$pr = < I, K, O, C, A \to B, H, E >, \qquad (3)$$

де *I* – унікальне ім'я продукції; *K* – сфера застосування або секція продукції; *O* – пріоритетність виконання продукції; *C* – умова застосовності продукції, яка зазвичай є логічним виразом; *A→B* – ядро продукції; *H* – післядії або післяумови продукції, що мають вигляд процедур, які виконуються в тому випадку, якщо ядро продукції реалізувалося; *E* – зв'язки з іншими продукціями.

Сфера застосування продукцій визначається характером методів. Наприклад, при видобуванні знань про терміносистему предметної області сферою застосування *K* є видобування знань з термінологічних словників. Пріоритетність продукції *O* встановлюється автоматично за довжиною умови застосовності, при цьому найдовша умова має найвищий пріоритет. Післядії *H* та зв'язки *E* з іншими продукціями визначаються під час розроблення ядра. Основним елементом продукції є

ядро продукційного правила у вигляді «Якщо *A*, то *B*». Під антецедентом *A* і консеквентом *B* розуміємо деяку множину фактів. Структура *skk* відповідає структурі факту. Щоб упевнитися в цьому, розглянемо ядро продукційного правила для виявлення кваліативного відношення агрегації «Частина-ціле» в реченні «Аванс становить частину загальної вартості договору». На деякій заданій підмножині природної мови воно має таке представлення:

```
ЯКЩО  <речення>           p     містить     <термін>            t1
   І  <речення>           p     містить     <термін>            t2
   І  <речення>           p     містить          <СемВідношення>     R
   І  <СемВідношення>     r     містить     <дієслово>          v
   І  <СемВідношення>     r     містить     <терм-супутникR>    tr
   І  <дієслово>          v     має         <значення>          ["становить"]
   І  <терм-супутникR>    tr    має         <значення>          ["частина"]
   І  <термін>            t     має         <індекс>            i
   І  <СемВідношення>     r     має         <індекс>            (i+1)
   І  <ознака>            h     має         <індекс>            (i+2)
   І  <термін>            t2    має         <індекс>            (i+3)
ТО    <термін>            t1    має         <тип>               ["Частина"]
   І  <<термін>           t2    має         <тип>               ["Ціле"]
   І  <СемВідношення>     r     відноситься до      <категорія>
      ["ЦілеЧастина"].
```

Це твердження дає змогу виявити те, що у вихідному реченні до категорії «ціле» належить термін «загальна вартість договору», до категорії «частина» – термін «Аванс», а також те, що між ними існує відношення «Ціле → частина». Як видно з прикладу, структуру простої ядерної конструкції кожного факту твердження можна побачити в явному вигляді, наприклад, у першому факті «<*речення*> *s* містить <*термін* > t_1 » : *x* = « <*речення*> *s* », *R* = «*містить* », *y* = «< *Термін* > t_1 ». Отже, як модель подання методу використовуємо продукційну модель подання знань [42]. Це означає, що для кожного методу необхідно розробити систему продукцій, яка найчастіше має ієрархічну структуру і є декларативною формою подання методу. Наприклад, система продукцій для розпізнавання семантичних відношень *PrSRR* складається з підсистем: розпізнавання семантичних відношень « Ціле → частина » *PrSRR_WP*, « Рід-вид » *PrSRR_CK* і т.д.

Однак, створення системи продукцій – це доволі трудомістке завдання, експертам найчастіше важко сформулювати правила, які вони використовують при вирішенні завдань, оскільки експертне знання в більшості випадків є підсвідомим. Саме підсвідомий характер експертного знання викликає труднощі при побудові експертних систем, а витяг експертних знань вважається «вузьким місцем» штучного інтелекту [42]. Тому для системи передбачено можливість додавання продукційних правил. Вхідними даними є текстовий файл з вхідним текстом у форматі «*.txt, *.doc, *.docx»,(рис. 14) та слова, словосполучення та їхні словоформи (рис. 15).

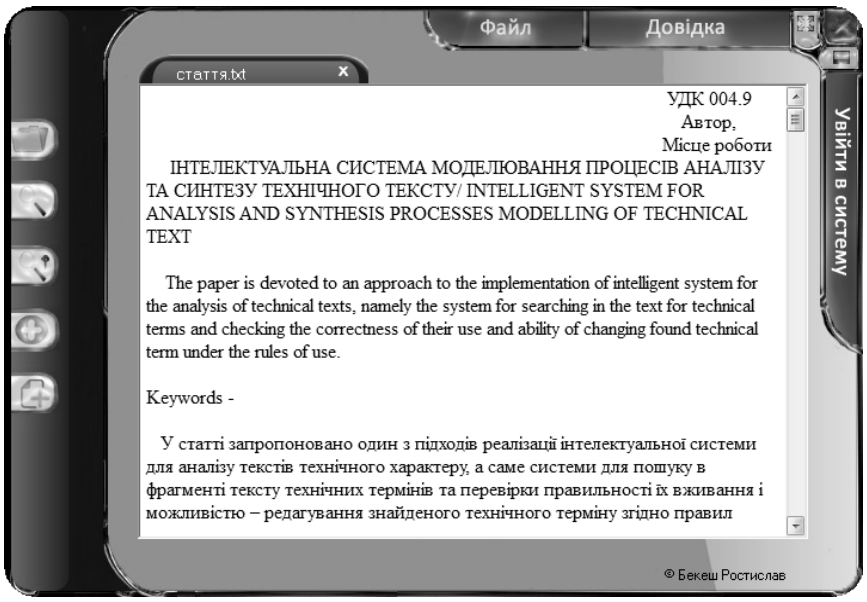

Рис. 14. Приклад введення вхідних даних

Рис. 15. Приклад додавання слів до бази даних

Результатом виконання програми є знайдений неправильний термін (рис. 16), пропозиція замінити його на один із правильних термінів зі списку (список 1, рис. 16), перелік знайдених ключових слів та файл формату RDF, в якому описано онтологію.

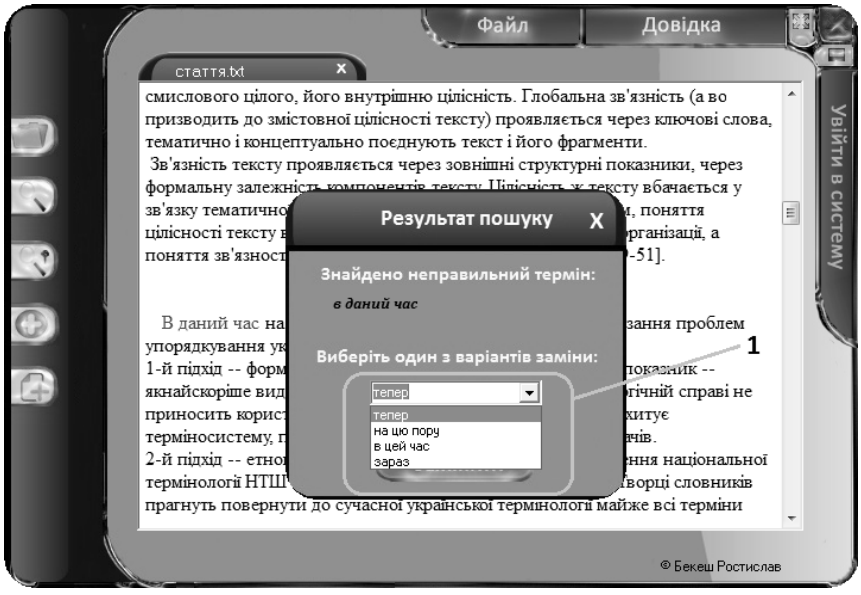

Рис. 16. Результат пошуку неправильного терміна та варіанти вибору правильного терміна

Програма виконує пошук неправильних термінів та ключових слів у текстовому документі.

1. Натиснути кнопку «Відкрити файл».
2. Натиснути кнопку «Перевірити текст», «Перевірити» або кнопку «Знайти неправильний термін».
3. Обрати один із запропонованих варіантів та натиснути кнопку «Замінити».
4. За необхідності натиснути кнопку «Знайти ключові слова» для відображення переліку ключових слів.

Вхідні дані. Фрагмент статті «Інтелектуальна система моделювання процесів аналізу та синтезу технічного тексту».

Результат роботи програми. Як видно з рис. 17, що системою знайдено неправильне словосполучення «зіставивши факти», також користувачу пропонується вибрати один з варіантів для заміни неправильного словосполучення правильним. На рис. 18 зображено результат роботи механізму пошуку ключових слів. Автор статті вказав ключові слова: морфологічний аналіз, морфеми, терміни. Система знайшла ключові слова: текст, система, підхід, механізм. Тому результат: «Вказані автором ключові слова не правильні».

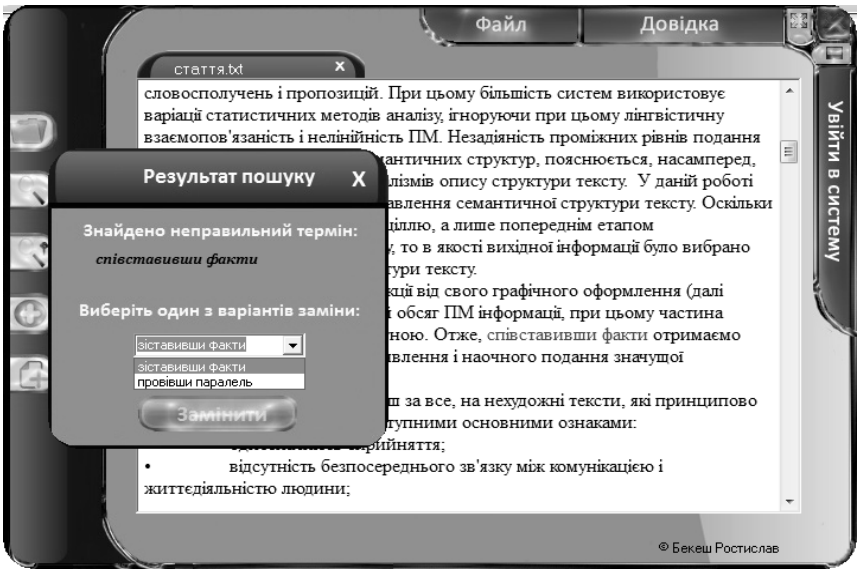

Рис. 17. Результат роботи процесу пошуку неправильних термінів та слів

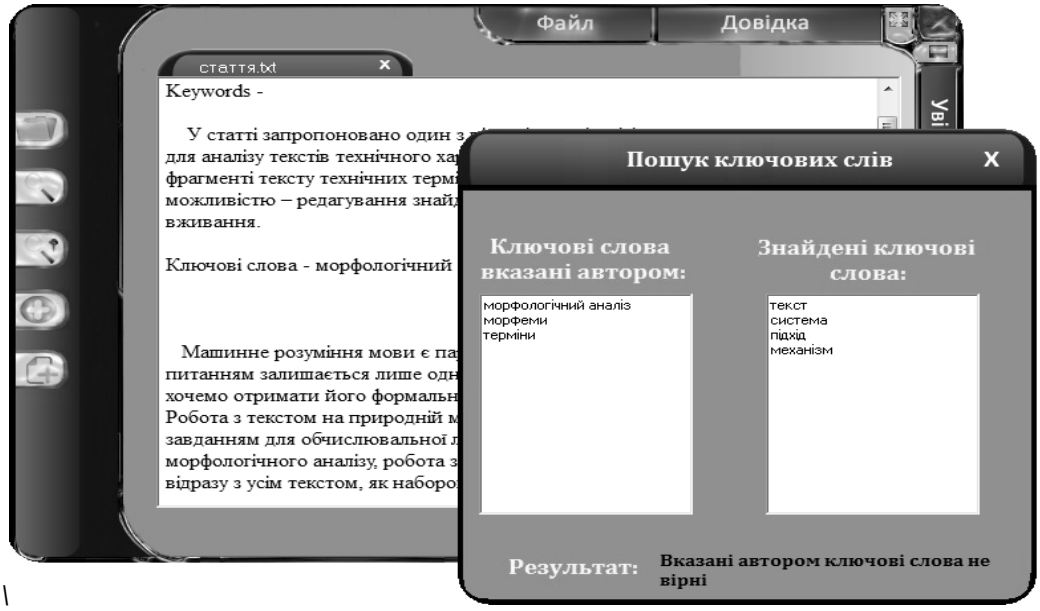

Рис. 18. Результат пошуку ключових слів

Висновки і перспективи подальших наукових розвідок

Розроблено інтелектуальну систему моделювання процесів аналізу та синтезу технічного тексту. Для проектування та виконання структурного аналізу системи використано середовище AllFusion Process Modeler 7, за допомогою якого було створено функціональну діаграму системи, діаграми потоків даних та логічну діаграму послідовності робіт. Всі складові та функціональні частини системи розроблено за допомогою середовища візуального програмування C++ Builder 6.0 з використанням запитів мовою SQL. Проектування бази даних виконано в середовищі MySQL Workbench. Було проаналізовано літературні джерела та дослідження у сфері автоматичного опрацювання текстів та морфологічного аналізу, розглянуто існуючі, схожі за функціональністю, системи. Розроблена система моделювання процесів аналізу та синтезу технічного тексту шукає неправильні слова та словосполучення у текстах технічного характеру, зокрема статтях. У системі додатково передбачено функцію пошуку ключових слів у тексті за законом Ципфа. Знайдені ключові слова можна використати для побудови онтології статті у вигляді XML документів – це важливо, оскільки цей формат став по суті стандартом для обміну даними між прикладними програмами. Для автоматизації стадії валідації ефективніше використовувати сторонні словники і тезауруси, що, відповідно, приводить до потреби розроблення україномовних відповідників WordNet.

Результатом роботи системи є знайдене неправильне слово чи словосполучення та перелік слів або словосполучень, якими можна його замінити, також система виводить список ключових слів, вказаних автором статті, список ключових слів, знайдених системою, та результат перевірки їхніх збігів. Незважаючи на функціональність системи, вона не позбавлена недоліків. Оскільки вся робота системи залежить від наповнення її словника, то насамперед слід поповнювати бази слів. Позбавлення розробленої системи її недоліків може стати першим кроком у її подальшому розвитку. Зокрема роботу над системою слід продовжити у напрямку удосконалення алгоритмів пошуку неправильних слів та словосполучень, наприклад, реалізації алгоритму пошуку на основі нейронних мереж. Для підвищення ефективності алгоритму пошуку ключових слів необхідно реалізувати можливість пошуку ключових слів не у одному файлі, а у декількох файлах схожої тематики з метою відкидання слів, які є характерними для текстів однакової тематики, але не є ключовими. Відмінністю створеної систем від існуючих систем на поточному етапі її розвитку є вузька спеціалізація системи на текстах конкретної тематики, зокрема технічних статтях . Отже, розроблена система моделювання процесів аналізу та синтезу технічного тексту є простим інструментом для пошуку неправильних слів і словосполучень, для побудови онтологій та може бути використана у некомерційних цілях.

1. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Е. И. Большакова, Э. С. Клышинский, Д. В. Ландэ, А. А. Носков и др. – М.: МИЭМ, 2011. – 272 с. 2. Апресян Ю. Д. Лексическая семантика / Ю. Д. Апресян. – М.: Восточная література, 1995. – 472 с. 3. Артеменко Ю. Н. MySQL Справочник по языку / Ю. Н. Артеменко. – М.: Вильямс, 2004. – 432 с. 4. Ваулин А. С. Языки программирования / А. С. Ваулин. – М.: Школа-Пресс 1993 г. 5. Вероятностный морфологический анализатор русского и украинского языков. – Режим доступу: www.keva.ru/stemka/stemka.htm. 6. Гвенцадзе М. А. Коммуникативная лингвистика и типология текста / М. А. Гвенцадзе. – Тбилиси: Ганатлеба, 1986. – 316 с. 7. Заболеева-Зотова А. В. Анализ семантической структуры и реферирование текстов на естественном языке / А. В. Заболеева-Зотова. – М.: Изд-во Физ-мат литературы, 2003. – 226 с. 8. Заболева-Зотова А. В. Естественный язык в автоматизированных системах. Семантический анализ текстов / А. В. Заболева-Зотова. – Волгоград:РПК «Политехник», 2002. – 228 с. 9. Закон Ципфа. Условная энтропия. Свойства иерархической аддитивности. – Режим доступу: http://kirsoft.com.ru/freedom/KSNews_394.htm. 10. Иорданская Л. Н. Морфологические типы основ

русского языка / Л. Н. Иорданская // Проблемы кібернетики. – М., 1961. С. 120–151. 11. Использование ЭВМ в лингвистических исследованиях / [отв. ред. В. И. Перебейнос]. – К.: Наукова думка, 1990. 12. Калверт Ч. Borland C++ Builder. Энциклопедия программиста / Ч. Калверт, К. Рейсдорф. – К.: ДиаСофт, 2001. – 944 с. 13. Катренко А. В. Системний аналіз об'єктів та процесів комп'ютеризації: навч. посібник / А. В. Катренко. – Львів: Новий світ-2000, 2003. – 424 с. 14. Керниган Б. В. Язык программирования С / Б. В. Керниган, Д. Ритчи, А. Фьюэр. – М.: Невский диалект, 2003. – 355 с. 15. Маклаков С. В. Моделирование бинес-процессов с AllFusion Process Modeler / С. В. Маклаков. – М.: Диалог-МИФИ, 2002. – 240 с. 16. Маклаков С. В. BPwin ERwin CASE-средства разработки информационных систем / С. В. Маклаков. – М.: Диалог- МИФИ, 2001. – 304 с. 17. Мельчук И. А., Морфологический анализ при машинном переводе / И. А. Мельчук // Проблемы кибернетики. – М., 1961. – С. 207–276. 18. Мова програмування С++. – Режим доступу: applmath.lviv.ua/fileadmin/studworks/2010/Andrii_Chukhrai/index.html. 19. Морфологічний аналіз української та російської мови. – Режим доступу: victoria.lviv.ua/html/oio-l/6.html. 20. Наконечна Г. В. Українська науково-технічна термінологія. Історія і сьогодення / Г. В. Наконечна. – Львів: Кальварія, 1999. – 278 с. 21. Применение ADO для работы с БД. – Режим доступу: www.codenet.ru/progr/bcb/ado/. 22. Програмування на С і С++: теорії і гіпотези. – Режим доступу: www.iatp.kharkov.ua/statti/cplusplus5/. 23. Холингвэрт Дж. С++Builder. Руководство разработчика/ Дж. Холингвэрт, Д. Баттерфилд, Б. Сворт. – М.: Вильямс, 2001. – 382 с. 24. AllFusion Process Modeler 7 (BPwin).Средство функционального моделирования бизнес-процессов. – Режим доступу: www.interface.ru/home.asp?artId=102. 25. Chapter 1. MySQL Workbench Introduction. – Режим доступу: dev.mysql.com/doc/workbench/en/wb-intro.html. 26. MySQL. Установка. Настройка. Программирование. – Режим доступу: www.gsub.kiev.ua/Arts/?aid=471&action=view. 27.Welty C. Towards a Semantics for the Web / C. Welty, Padova, Italy. 28. Когаловский М. Р. Абстракции и модели в системах баз данных / М. Р. Когаловский // Журнал «СУБ», Издательский дом «Открытые системы», 4-5/1998. 29. Онтологии в системах искусственного интеллекта: способы построения и организации (часть 1) / А. В. Смирнов, М. П. Пашкин, Н. Г. Шилов, Т. В. Левашова // «Новости искусственного интеллекта» . – 2002. – № 1 (49). 30. Клещев А. С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия "онтология" / А. С. Клещев, И. Л. Артемьева // НТИ. Сер. 2, 2001. 31. Норенков И. П. Интеллектуальные технологии на базе онтологій / И. П. Норенков // Информационные технологии. – 2010. – № 1. – С. 17–23. 32. Staab S. Handbook on Ontologies/ Staab S., Studer R. // Springer—Verlag, 2004. 33. Рубашкин В. Ш. Онтологии концептуальные границы, проблемы и решения. Точка зрения разработчика / В. Ш. Рубашкин // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции "Диалог 2007". – М.: Издательский центр РГГУ. – 2007. – С. 481–485. 34. Евгенев Г. Б. Онтология инженерных знаний / Г. Б. Евгенев // Информационные технологии. – 2001. – № 6. – С. 2–5. 35. Смирнов С. В. Онтологический анализ предметных областей моделирования / С. В. Смирнов; Известия Самарского научного центра РАН. – 2001. – Т. 3. – № 1. – С. 62–70. 36. Yildiz B. Ontology-Driven Information Systems: Challenges and Requirements / B. Yildiz, S. Miksch // Proceedings of the International Conference on Semantic Web and Digital Libraries, 2007. 37. Berners-Lee T. The Semantic Web / T. Berners-Lee, J. Hendler, O. Lassila // Scientific American. – 2001. 38. Ruch P. Automatic medical encoding with SNOMED categories / P. Ruch, J. Gobeil, C. Lovis, A. Geissbtihler // BMC Medical Informatics and Decision Making, – 2001. 156 с. 39. Гаврилова Т. А. Базы знаний интеллектуальных систем: учебное пособие для вузов / Т. А. Гаврилова, В. Ф. Хорошевский // СПб.: Питер, 2000. – 382 с. 40. Найханова Л. В. Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования : монография / Л. В. Найханова; Федеральное агентство по образованию, ГОУВПО ВСГТУ. Улан-Удэ: БНЦ СО РАН, 2008. – 244 с. 41. Рабчевский Е. А. Автоматическое построение онтологий / Е. А. Рабчевский, Г. И. Булатова //

Научно-технические ведомости СПбГПУ. 2007. – 4. – СПб.: Изд-во политехнического университета, 2007. 42. Haav H.M. An Application of Inductive Concept Analysis to Construction of Domainspecific Ontologies / Haav H.M.; In Proceedings of the Workshop of VLDB2003, 2003. – P. 63–67. 43. Maedche. A. Discovering Conceptual Relations from Text / A. Maedche, S. Staab // ECAI 2000. Proceedingsof the 14th European Conference on Artificial Intelligence, IOS Press, Amsterdam, 2000. 44. Добров Б. В. Онтологии и тезаурусы: модели, инструменты, приложения / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич // БИНОМ, Интернет-университет информационных технологий-ИНТУИТ.ру, 2009. – 173 с. 45. Клещёв А. С. Математические модели онтологий предметных областей / А. С. Клещёв, И. Л. Артемьева // Сер. 2. 2001. – № 2. – С. 20–27. 46. Клещёв А. С. Математические модели онтологий предметных областей / А. С. Клещёв, И. Л. Артемьева // НТИ. Сер. 2. 2001. – № 3. – С. 19–29. 47. Клещёв А. С. Математические модели онтологий предметных областей. Ч. 3. Сравнение разных классов моделей онтологий / А. С. Клещёв, И. Л. Артемьева // НТИ. Сер. 2. 2001. – № 4. – С. 10–15. 48. Декер С. Semantic Web: роли XML и RDF / С. Декер, С. Мельник // Открытые системы, 2001. – № 09. – С. 23–27. 49. Бездушный А. А. RDFS как основа среды разработки цифровых библиотек и Web-порталов / А. А. Бездушный, А. Н. Бездушный, А. К. Нестеренко, В. А. Серебряков, Т. М. Сысоев // Электронные библиотеки – 2003. 50. Kalyanpur A. OWL: Capturing Semantic Information using a Standardized Web Ontology Language / A. Kalyanpur // Multilingual Computing & Technology Magazine, Vol. 15, issue 7, Nov 2004. 51. Gruninger M. An Ontology Framework / M. Gruninger, L. Obrst // Ontology Summit NIST, Gaithersburg, MD April 22–23, 2007. 52. Слета В. Д. Построение и эволюционное развитие онтологий / В. Д. Слета, А. С. Сергеев // Известия ПГПУ им. В. Г. Белинского. 2010 № 18 (22). – С. 196–200. 53. Пронина В. А. Использование отношений между атрибутами для построения онтологии предметной области / В. А. Пронина, Л. Б. Шипилина. – 2009, № 1. – С. 27–32. 54. Декер С. Semantic Web: роли XML и RDF / С. Декер, С. Мельник, Ф. Хермелен // Открытые системы. – 2001. 55. Lassila O. Resource Description Framework : Model and Syntax Specification / O. Lassila, R. Swick // W3C Recommendation, 1999. 56. Klyne G. Resource Description Framework : Concepts and Abstract Data Model / G. Klyne, J. Carroll // W3C Working Draft, 2002. 57. Brickley D. RDF Vocabulary Description Language 1.0: RDF Schema / D. Brickley, R. V. Guha. – W3C Working Draft, 2002.