УДК 004.853, 004.55 https://doi.org/10.15407/pp2019.04.028

Ю.В. Рогушина

ВИКОРИСТАННЯ ТЕЗАУРУСІВ ДЛЯ ПОШУКУ СКЛАДНИХ ІНФОРМАЦІЙНИХ ОБ'ЄКТІВ У WEB НА ОСНОВІ ОНТОЛОГІЙ

Запропоновано онтологічну модель взаємодії між об'єктами та суб'єктами семантичного пошуку у Web, охарактеризовано її основні елементи, розглянуто методи її поповнення та застосування для фільтрації інформації, що відповідає персоніфікованим потребам користувачів. Проаналізовано типи відношень між екземплярами та класами цієї моделі та їх характеристики, що можуть впливати на часову складність обробки знань, що подані на основі цієї моделі. Одним з важливих елементів запропонованої моделі є тезауруси, які відображають знання щодо задач, для розв'язання яких користувачі шукають інформацію, та щодо інформаційних ресурсів, в яких такі відомості можуть міститися. Обґрунтовується доцільність застосування окремих випадків онтології – тезаурусів – для знаходження семантично подібних інформаційних об'єктів. Розглянуто види тезаурусів, які застосовуються для семантичного пошуку, наведено джерела їх поповнення та проаналізовано їх характеристики. В роботі запропоновано алгоритм автоматизованої побудови простого тезаурусу, що утворюється на основі онтології предметної області та природномовного опису задачі користувача, та методи генерації складених тезаурусів, що пертинентні новим задачам користувача, за множиною простих тезаурусів, що побудовані користувачем раніше. Оцінюються виразність та обчислювальна складність запропонованих методів, яка залежить від властивостей онтології предметної області та від обсягу опису задачі. Розглянуто методи використання семантично розмічених Wiki-ресурсів як джерела знань для побудови онтологій предметних областей та пов'язаних з ними типових інформаційних об'єктів.

Ключові слова: семантичний пошук, інформаційний об'єкт, онтологія, тезаурус задачі, семантична розмітка.

Вступ

Сучасні інтелектуальні інформаційні системи (ІІС) орієнтовані на роботу в розподіленому середовищі Web, що потребує динамічного отримання актуальних та пертинентних знань з його ресурсів. Великий обсяг та складна структура інформаційних ресурсів (ІР), тенденція поширення великих даних (Big Data) викликають потребу у створенні засобів автоматизованої обробки інформації, які дозволили б аналізувати зміст цих ресурсів та здобувати з них саме ті відомості, що потрібні користувачу для вирішення його поточної задачі.

Один з найбільш перспективних напрямків розв'язання цієї задачі базується на використанні для цього зовнішніх баз знань, тобто із семантизацією як самих ресурсів Web, так і процесу їх пошуку. У використанні ресурсів Web на найбільш високому рівні можна виділити дві основні задачі:

– відбір ІР – пертинентні поточні задачі користувача, що містять інформацію для її розв'язку;

– здобуття з ІР тієї інформації, яка потрібна користувачеві.

Рішення першої задачі є об'єктом для інформаційно-пошукових та рекомендуючих систем і може бути вдосконалено за допомогою застосування фонових знань та інтелектуальних методів їх обробки. Для розв'язання другої задачі використовують Data Mining, Text Mining, методи машинного навчання тощо, що дозволяють здобути з даних неявно представлені в них відомості. Незалежно від того, наскільки досконалим буде розв'язок другої проблеми, вона не буде ефективно вирішена, якщо методи аналізу будуть обробляти не пертинентні дані.

У найбільш широкому розумінні семантизація полягає у встановленні зв'язку між певним інформаційним об'єктом (ІО) та його змістом. Під *семантизацією* ІР будемо надалі розуміти встановлення формалізованих відношень між цими ІР (або їх елементами та метаописами) та формалізованим поданням знань (наприклад, з онтологією, семанти-

чною мережею, фреймом), тобто їх семантичну розмітку на основі обраного подання знань. Така розмітка – це основа для більш ефективної навігації та пошуку в Web.

Під семантичним пошуком [1] зазвичай розуміють такий пошук інформації, коли для задоволення інформаційних потреб користувача, що виникають у процесі розв'язання певної задачі, використовуються зовнішні знання щодо суб'єктів і об'єктів пошукової процедури й методів аналізу цих знань. Це викликає потребу в застосуванні формально представлених зовнішніх відносно пошукової процедури знань щодо основних елементів цієї процедури. Такі фонові знання можуть стосуватися користувача та специфіки його інформаційних потреб (персоніфікація пошуку), структури ІР, серед яких здійснюється пошук, тієї предметної області (ПрО), до якої відносяться ці ІР. В процесі семантичного пошуку співставлення запиту користувача з контентом ІР здійснюється не безпосередньо, а з урахуванням фонових знань співставляються їх формалізовані інформаційні моделі.

Використання типових ІО дозволяє чіткіше визначити інформаційну потребу користувача на змістовному рівні. Це дозволяє категоризувати вміст ІР та пов'язувати елементи контенту з певними поняттями ПрО, які є типовими – мають однакові властивості, відносяться до однакової групи класів, містять подібні за структурою та вмістом елементи. Наприклад, за тими самими умовами користувач може шукати людину, організацію або документ. Якщо пошук здійснюється серед структурованих та класифікованих ІР, визначення типу ІО дозволяє обрати категорію або набір категорій, до якої має відноситися шуканий ІО. Для пошуку серед неструктурованої інформації виникає потреба спочатку отримати з фонових знань ПрО інформацію щодо структури шуканого ІР (його властивості, їх типи та можливі значення, надкласи та підкласи тощо), а потім застосовувати цю інформацію для фільтрації результатів пошуку, отриманих за запитом користувача. Джерелом таких знань можуть бути як онтології ПрО, до яких відносяться шукані ІР, так і довільні семантично структуровані ІР (наприклад, семантичні Wiki-ресурси). Важливо, що інформацію щодо структури та властивостей таких типових ІО користувач може отримувати із зовнішніх джерел знань, а не формулювати самостійно. Це значно спрощує пошук ІО зі складною структурою та дозволяє відфільтровувати необхідну інформацію серед великої кількості ІР, але користувачеві потрібно самостійно обирати з таких наборів знань ту підмножину, яка пертинентна його задачі.

Наприклад, якщо таким ІО є людина, то різним користувачам можуть бути необхідні різні аспекти відомостей – щодо освіти, кваліфікації, здоров'я, сімейного стану тощо.

Щоб використовувати онтологічні знання в процесі семантичного пошуку, потрібно забезпечити: 1) механізми створення онтологічних моделей інформаційних потреб користувачів та ІР, серед яких здійснюється пошук; 2) методи зіставлення таких моделей. Перша проблема пов'язана з формалізацією властивостей основних елементів пошукової процедури, яка виконується із застосуванням фонових знань, а друга може розглядатися як окремий випадок співставлення незалежно створених онтологій, на які накладено деякі специфічні обмеження.

Онтологічна модель взаємодії користувачів та ІР у Web

Щоб проаналізувати методи знаходження в Web ІО зі складною структурою, що відповідають персональним інформаційним потребам користувачів, необхідно побудувати модель пошуку, яка дозволяє чітко та однозначно відобразити властивості основних компонентів пошукової процедури та зв'язки між ними. Такий опис має визначити всі базові терміни, що використовуються для опису задачі семантичного пошуку та характеризують його учасників, вхідні та вихідні дані, а також критерії, за якими оцінюються результати пошукового процесу.

Сьогодні для моделювання різноманітних ПрО широко застосовуються онтологічні моделі. Онтологічна модель семантичного пошуку (*ОМСП)* – це онтологічна модель, яка формалізує відношення між основними суб'єктами пошуку, до яких можна віднести користувачів, експертів, авторів ІР тощо, і його об'єктами (такими, як ІР, ІО, запити та результати їх виконання, описи ПрО тощо). Така модель дозволяє однозначно описати ті взаємини між користувачами та ІР, які виникають в процесі використання знань для задоволення інформаційних потреб користувачів. Для подання моделі може бути використана мова OWL, що дозволяє застосовувати її в різних ІІС, які функціонують в Web і використовують його ІР [2].

Використання ОМСП у задачі семантичного пошуку є основою для інтелектуальної обробки ресурсів Web з використанням онтологічного аналізу. Основна ідея запропонованого підходу полягає у тому, що застосовуються два типи онтологій – зовнішні та внутрішні, відмінність між якими полягає у наступному:

– внутрішня онтологія створюється самими розробниками ІІС відповідно до специфіки тих задач, що вирішуються системою, та формалізує структуру та відношення між основними суб'єктами та об'єктами цієї ІІС, і тому всі характеристики цієї онтології відомі ще до початку роботи з ІІС і дозволяють чітко та однозначно визначити її виразні можливості, обсяг та методи обробки;

– внутрішні онтології здобуваються з ресурсів Web у процесі функціювання ІІС (їх знаходять у зовнішніх репозиторіях, будують відповідно до потреб користувачів, експортують із різноманітних семантичних представлень даних тощо), і тому неможливо оцінити до початку роботи їх властивості та виразну здатність, що безпосередньо визначають складність обробки.

Для задачі семантичного пошуку ОМСП є внутрішньою онтологією, тоді як отримані з різних джерел онтології ПрО, ІР та ІО є зовнішніми.

Цей підхід може застосовуватися для розв'язку інших інтелектуальних задач, пов'язаних з аналізом інформаційних ресурсів Web. Прикладами таких задач є проактивне надання рекомендацій, машинне навчання, створення семантичних порталів. В таких випадках потрібно побудувати відмінну від ОМСП модель взаємодії елементів такої системи (слід зазначити, що багато класів ОМСП – такі, як користувач та ПрО – є досить універсальними, і їх можна переносити до нової моделі тільки з певними доповненнями), доповнену специфічними для задачі класами.

Крім того, ОМСП може бути використана для окремих випадків семантичного пошуку, приклади яких будуть розглянуті далі, – для пошуку фіксованих підмножин ІО (пошук вакансій та навчальних закладів, Web-сервісів) та для пошуку в інформаційному середовищі, що є підмножиною Web (пошук у репозиторіях RDF та OWL, у Wiki-ресурсах).

Основні *суб'єкти* інформаційного пошуку – сутності, які своїми діями можуть ініціювати пошуковий процес або впливати на його результати:

– *користувачі* – ті особи (люди або програмні сутності), які прагнуть за допомогою пошуку (наприклад, за допомогою певної ІПС) отримати доступ до певної інформації;

– *експерти* – ті особи, які здатні певним чином оцінювати об'єкти і суб'єкти пошуку (приміром, надавати кількісну оцінку якості ІР, його відповідності певному запиту, визначати зв'язок між онтологією ПрО та задачею користувача тощо);

– *власники ІР* – особи, що створюють або публікують певну інформацію в Web та можуть визначати її тематику, якість, умови доступу тощо.

У семантичному пошуку додатково можуть використовуватися такі суб'єкти, як *група користувачів* – скінчена невпорядкована множина користувачів, що поєднана за певними спільними властивостями. Приміром, у деяких рекомендуючих системах кожен користувач може визначити склад співтовариства, думки якого в по-

точній ситуації для нього мають певну цінність.

Основні *об'єкти* інформаційного пошуку – сутності, що використовуються в процесі виконання пошукових процедур: ІР; ІО; інформаційне середовище; інформаційно-пошукові системи (ІПС); інформаційні потреби користувачів (ІП); запити, що формалізують ІП користувачів; результати виконання запитів; зовнішні бази знань (БЗ).

ІР – це сукупність даних (документів, файлів тощо), засобів доступу та користування цими даними (бібліотека, архів, база даних тощо). В даній роботі основна увага приділяється ІР, що представлені в електронній формі та доступні за допомогою Web, тобто мають унікальні ідентифікатори (адреси) та характеризуються як за допомогою формальних властивостей (розмір, час створення модифікації, мова подання тощо), так і через їх контент. Також для опису ІР можуть використовуватися метадані, що описують ці властивості певною формальною мовою (приміром, RDF).

ІО – модель певного об'єкту ПрО в інформаційному просторі, що визначає структуру, атрибути, обмеження цілісності і, можливо, поведінку цього об'єкта через контент інформаційних ресурсів. До складу одного ІР може входити кілька ІО. З іншого боку, один ІО може бути описаний за допомогою кількох ІР. Приклади ІО – Web-сервіс, організація, особа, документ. Приміром, сайт організації може складатися з набору окремих Web-сторінок, але на одній з цих сторінок можуть описуватися кілька осіб.

Інформаційне середовище – сукупність усіх доступних ІР, їх властивостей (включаючи їх оцінки користувачами) і зв'язків між ними. У даній роботі під інформаційним середовищем будемо розуміти Web, якому характерні геторогенність, динамічність та великий обсяг інформації, що визначають вимоги та обмеження до методів пошуку інформації, що розробляються. Інші приклади інформаційного середовища, що задають інші специфічні вимоги до пошуку, – корпоративні мережі, сховища даних різних типів, інформаційний вміст локального обчислювального пристрою.

ІПС – засіб, що встановлює за певними критеріями кількісну міру відповідності між запитом користувача та інформацією щодо певної множини ІР або ІО та знаходить серед них підмножину найбільш відповідних.

ІП – усвідомлена необхідність в інформації для розв'язання поставленого завдання за розробленим планом. ІП, для задоволення якої і виконується пошук інформації, може бути формалізована за допомогою запиту (та його контексту), який характеризує поточні інтереси користувача, його задачу та здатність до сприйняття інформації тощо. У більшості випадків інформаційна потреба користувача є надто складною, щоб її формалізація відображала її повністю.

Запит – представлена за допомогою якоїсь мови формалізація інформаційної потреби користувача. Це може бути набір ключових слів – можливо, пов'язаних логічними операторами (такі запити застосовуються найчастіше), природномовне речення або перелік значень властивостей того ІО, який має задовольнити інформаційну потребу (приміром, вхідні та вихідні дані Web-сервісу або адреса організації, назва якої потрібна користувачу).

Результат запиту – це скінчена впорядкована множина ІР або ІО, які ІПС відібрала серед усіх приступних джерел інформації шляхом співставлення інформаційної потреби користувача з інформацією щодо цих ІР або ІО. Результати виконання того самого запиту у різний час можуть різнитися як через зміни в оточуючому середовищі, так і через зміни у профілі користувача.

Зовнішня БЗ – сукупність формалізовано поданих знань, що створена та функціонує незалежно від дій користувачів та розробників пошукової системи, але може бути використана в процесі пошуку.

Крім основних об'єктів процесу інформаційного пошуку, ОСМП описує також додаткові об'єкти, що пов'язані із семантизацією та персоніфікацією пошукових процедур та з підтримкою колабора-

тивного пошуку. Додаткові об'єкти дозволяють більш точно охарактеризувати основні об'єкти цього процесу. До таких об'єктів належать:

- предметна область (ПрО);
- онтології ПрО та ІО;
- тезауруси;
- лексичні онтології;
- теми запитів.

ПрО – деяка підмножина реального світу, що відповідно до якогось набору ознак цікавить користувача у певний час. Це може бути галузь знань, сукупність територіально поєднаних сутностей тощо. ПрО може бути формально представлена через множину понять, їх властивостей, відношень між ними та різноманітних обмежень. Нині у Web-орієнтованих інтелектуальних системах для формалізації опису ПрО часто використовуються її онтології.

Онтологія ПрО – це довільна онтологія [3], представлена на одному з діалектів OWL [4] та придатна для комп'ютерної обробки. Класи цієї онтології відповідають поняттям обраної ПрО, її екземпляри пов'язані з окремими випадками цих понять, а властивості дозволяють визначити зв'язки між поняттями та їх екземплярами. Онтології дозволяють формально описувати як семантику ПрО, що цікавить користувача, і задачі, яку він прагне вирішити, так і семантику тих ІР та ІО, які містять потрібні користувачеві відомості. Слід зазначити, що ці онтології, на відміну від ОМСП, є зовнішніми для задачі пошуку: на відміну від ОМСП, що може в процесі функціювання системи семантичного пошуку тільки поповнюватися новими екземплярами класів та значеннями їх властивостей, ці онтології можуть змінюватися довільним чином – як внаслідок змін у тих ресурсах, за якими вони будуються, так і внаслідок безпосередніх вказівок користувача.

Онтологія ІО – онтологія (часто – таксономія), що формалізує структуру групи ІО, що є суб'єктами пошуку, та їх відношення як одного з одним, так і з іншими об'єктами ПрО, що впливають на обмеження та умови у пошуковому запиті щодо того, які саме типи та екземпляри ІО задовольняють потребам користувача.

Задача користувача – поточна задача, для розв'язку якої користувач потребує отримати певну інформацію з зовнішніх ІР. Може бути описана через природномовне (неструктуроване) або структуроване визначення, приклади, елементи метаданих.

Тезаурус задачі – це окремий випадок онтології ПрО, який містить тільки онтологічні терміни (класи та екземпляри), але не описує (або обмежено описує) семантику відношень між ними з метою аналізу природномовних текстів. Може автоматизовано генеруватися за онтологією ПрО та природномовним описом задачі. Це окремий випадок онтології. *Простий* тезаурус задачі – тезаурус, який базується на термінах однієї онтології ПрО. *Складений* тезаурус задачі – тезаурус, який базується на термінах двох або більш онтологій ПрО.

Тезаурус ІР – це підмножина тезаурусу задачі, який містить тільки ті його терміни, для яких знайдено відповідні фрагменти у контенті цього ІР. Таким чином, склад тезаурусу ІР залежить як від тезаурусу задачі, для якої він будується, так і від методу співставлення контенту ІР із термінами цього тезаурусу.

Лексична онтологія ПрО – онтологія, яка містить формалізовані знання щодо зв'язків між поняттями певної онтології ПрО та пертинентними їм елементами природномовних текстів.

Тема запитів – це скінчена невпорядкована множина запитів одного або кількох різних користувачів, які дозволяють згрупувати їх за певними спільними властивостями або шляхом перерахування для того, щоб спільно обробляти їх параметри або отримані за цими запитами результати. Теми запитів дозволяють структурувати колаборативний пошук та організувати обмін інформацією за визначеними напрямками.

На основі ОМСП створюється інтероперабельний *профіль користувача,* який базується на класі ОМСП "Користувач" та як об'єктні властивості використовує екземпляри інших класів цієї онтологічної моделі. Відомості в цьому профілі можна поділити на кілька груп:

– Реєстраційна інформація:
  · ідентифікатор користувача;
  · пароль для доступу до ІПС.
– Досвід взаємодії ІПС з користувачем:
  · список онтологій, які користувач застосовував для опису своїх інформаційних інтересів;
  · список тезаурусів, що користувач застосовував у пошукових запитах;
  · список раніше виконаних запитів;
  · список результатів виконаних запитів з оцінками користувача для знайдених результатів.
– Відомості, імпортовані з зовнішніх джерел (необов'язкові відомості, їх може й не бути):
  · ідентифікатори користувача в соціальних мережах, що дають змогу динамічно оновлювати відомості про нього;
  · рейтинги користувача в соціальних мережах;
  · адреса користувача у Вікіпедії та інших Wiki-ресурсах;
  · адреса сайту користувача;
  · сфера компетенцій користувача (ключові слова, імпортовані з соціальних мереж);
  · посилання на публікації користувача.

Власні характеристики користувача: сфера компетенцій користувача (список ключових слів, що вводяться користувачем безпосередньо).

– Формальні дані про користувача (необов'язкові відомості, що дають змогу ІПС формувати групи користувачів зі схожими інформаційними потребами): місце проживання; вік; професія, освіта тощо.

Для опису ОМСП пропонується використовувати наступну формальну модель онтології:

$$O = \langle X, R, F, T \rangle,$$

яка більш детально описана в [5]. Ця модель дозволяє формалізувати відношення між елементами процесу пошуку інформації в Web, вона досить добре співставляється з технологічними елементами редактора онтологій Protégé та засобами семантичної розмітки Semantic MediaWiki, використання яких для поповнення онтологій розглядатиметься далі.

ОМСП містить такі основні класи, що пов'язані із типами об'єктів та суб'єктів семантичного пошуку:

– *користувач* – клас, екземпляри якого відповідають описам окремих користувачів, а властивості відповідають параметрам профілю користувача, який описано вище, та зв'язують екземпляри цього класу із екземплярами інших класів ОМСП та константами, що визначають значення певних параметрів із цього профілю;

– *онтологія ПрО,* що містить опис області, до якої належать інформаційні потреби користувача

$$O_{ПрО_i} = \langle X_{ПрО_i}, R_{ПрО_i}, F_{ПрО_i} \rangle, i = \overline{1,n};$$

– *лексична онтологія ПрО* – база знань щодо лексики ПрО, що містить відомості про лексеми природних мов, які відповідають термінам онтології ПрО

$$L_{ПрО_i} = \langle X_{lex_i}, R_{lex_i} = \{r_{lex}\}, \varnothing \rangle, i = \overline{1,n},$$

де

$$X_{lex_i} = X_{ПрО_i} \cup T_{ПрО_i}$$

тобто

$$\forall x_{i_j} \in X_{ПрО_i}, j = \overline{1,m_i}$$

існує набір фрагментів ПМ

$$\{s_{i_{j_p}} \in T_{ПрО_i}\}, p = \overline{1,q_{ij}}, r_{lex}(s_{i_{j_p}}) = x_{i_j} -$$

така онтологія використовується для встановлення зв'язків між елементами природномовних документів і термінами онтології ПрО;

– *тезаурус* – множина термінів Th, що разом із своїми властивостями характеризують певний суб'єкт пошуку, дозволяючи співставляти його з іншими суб'єктами; цей клас у рамках ОМСП має наступні підкласи, екземпляри яких мають додаткові властивості:

– *тезаурус онтології* – множина термінів онтології

$$Th_O = \{x_k \in X\}, k = \overline{1,n};$$

– *тезаурус множини онтологій* – об'єднання тезаурусів множини онтологій

$$O^* = \{X_m\}, m = \overline{1,p},$$

такого, що містить p онтологій, $p \geq 1$,

$$Th_{O^*} = \{x_{k_m} \in X_m\}, k_m = \overline{1,n_m}, m = \overline{1,p},$$

таке, що

$$Th_{O^*} = \bigcup_{m=1}^{p} Th_{O_m};$$

– *тезаурус задачі* – множина термінів з множини X онтології O, сукупність яких характеризує ту конкретну задачу з ПрО, що в цей час розв'язує користувач (визначається шляхом співставлення онтології O з описом задачі),

$$Th_{i_j} = \{th_{k_{i_j}} \in X\}, k = \overline{1,s_{i_j}}\ j = \overline{1,m_i};$$

– *зважений тезаурус задачі* – множина пар, першим елементом яких є термін з тезаурусу задачі, сукупність яких характеризує конкретну задачу з ПрО, а другим – вага (позитивна чи негативна) цього терміна для цієї задачі

$$Tw_{i_j} = \{< th_{k_{i_j}} \in X_i, w_{k_{i_j}} >\},$$

$$k = \overline{1,s_{i_j}}, j = \overline{1,m_i};$$

– *тезаурус IP* – підмножина термінів тезаурусу задачі, яким відповідають певні фрагменти контенту або метаопису IP

$$Th_{IR_q} = \{th_k \in Th_{i_j}\}, k = \overline{1,s_{i_j}}, q = \overline{1,z}$$

(слід відмітити, що для різних задач тезауруси того самого IP можуть значно різнитися);

– *зважений тезаурус IP* – множина пар, першим елементом яких є термін тезаурусу задачі, що містяться в контенті IP або в його метаописі, а другим – вага цього терміну для документа, яка визначається (за різними критеріями) як функція від кількості появ цього терміну в IP, місць його появи та від довжини документа

$$Tw_{IR_q} = \{th_k \in Th_{i_j}, w_k >\},$$

$$k = \overline{1,s_{i_j}}, q = \overline{1,z};$$

– *тезаурус IO* – множина термінів тезаурусу задачі, що містяться в контенті IO або в його метаописі

$$Th_{IO_q} = \{th_k \in Th_{i_j}\}, k = \overline{1,s_{i_j}} –$$

такий опис дозволяє коректно співставляти різні типи IO та IO одного типу , але з різною семантикою із урахуванням їх структури (приміром, розрізняти Web-сервіси, якщо вхідні дані одного подібні до вихідних даних іншого);

– *зважений тезаурус IO* – множина пар, першим елементом яких є терміни тезаурусу задачі, що містяться в контенті IO або в його метаописі, а другим – назва того елементу даного IO (з онтологічного опису IO), в якій зустрічається даний термін

$$Tw_{IO_q} = \{< th_k \in Th_{i_j}, d_w >,$$

$$k = \overline{1,s_q}\}, k = \overline{1,s_{i_j}}, w = \overline{1,x_{IO}};$$

– *зважений тезаурус задачі користувача* – множина пар, першими елементами яких є терміни однієї або різних онтологій, сукупність яких характеризує інформаційні інтереси користувача, а другим – вага цього терміна для опису інтересів користувача

$$Tw_{user_j} = \{< th_{k_{user_j}} \in T_{ПрO_i}, w_{k_{user_j}} >\},$$

$$k = \overline{1,s_{user_j}},$$

де вага терміну визначається як функція (як правило, як сума добутків) від ваги певного ресурсу для користувача та кількості термінів у цьому ресурсі;

– *запит* – множина ключових слів, що характеризують одну з інформаційних потреб користувача, пов'язану з конкретною задачею, за допомогою тезауруса;

– *тема* – множина запитів, пов'язаних з однією інформаційною потребою, що дає змогу поєднувати семантично по-

в'язані запити різних користувачів, які базуються на різних онтологіях і тезаурусах;

– *результат запиту* – множина пар, першим елементом яких є посилання на ІР, а другим – оцінки цих ІР користувачем;

– *група користувачів* – клас, властивостями якого є ідентифікатор групи і список користувачів, які з певних причин об'єднані в одну групу (групи можуть формуватися шляхом вибору користувача безпосередньо чи автоматично на основі відповідності яким-небудь умовам, наприклад, групи користувачів з подібними формальними даними або таких, що виконують схожі запити);

– *ІР* – клас, що описує відомості про відомі ІПС ресурси (ідентифікатор ресурсу, запити, за якими він був виявлений, оцінку користувача, якому він був наданий, і його рівень читабельності для цього користувача) та оцінки цих ресурсів, надані різними користувачами

$$\left\langle U_{url}, \left\{ \left\langle z_i, m_i, q_i \right\rangle, i = \overline{1,n} \right\} \right\rangle;$$

– *ІО* – клас, що описує відомості про відомі ІПС ІО з певною структурою, визначеною користувачем, що містяться в одному чи декількох ІР (ідентифікатор ІО, запити, за якими він був виявлений, оцінку користувача, якому він був наданий, і онтологію, що визначає структуру даного ІО) та оцінки цих ІО, надані різними користувачами

$$\left\langle \mathrm{IO}_{\mathrm{url}}, \left\{ \left\langle \mathrm{z_i}, \mathrm{m_i}, \mathrm{O_i} \right\rangle, \mathrm{i} = \overline{1,\mathrm{n}} \right\} \right\rangle;$$

– *рекомендація* – інформація, що надається користувачеві ІПС проактивно, як наслідок аналізу і персональних відомостей про цього користувача, і колаборативного досвіду системи.

– *агент користувача* – це інтелектуальний програмний агент, що презентує інтереси користувача у взаємодії з ІПС та виконує певні дії в його інтересах.

Застосування такого формалізму, як агент користувача, дасть змогу, з одного боку, уникнути приписування людині-користувачу штучно обмеженої і формально схарактеризованої сфери інтересів, а з іншого – забезпечить засоби та методи прогнозування його вчинків у межах моделі взаємодії користувача та ресурсів у відкритому інформаційному середовищі. Для опису поведінки такого агента використовуються інтенсіональні відношення, за допомогою яких можна формалізувати цілі, наміри й бажання користувача. Таким чином, у ОМСП проводиться відмінність між самим користувачем: клас "користувач" відображає інформацію щодо фактів, пов'язаних з діями користувача, а клас "агент користувача" містить припущення щодо мотивації цих дій.

Для того, щоб описати екземпляри класів ОМСП з $\mathrm{X_{ind}}$, необхідно спочатку формалізувати ті відношення, які для цього використовуються, та задати їх область значення та визначення.

Відношення між елементами ОМСП

Однією з основних переваг, яку забезпечує наявність використання онтологічного підходу до моделювання процесу пошуку, є можливість явно визначити семантику відношень між його основними елементами, тобто задати не тільки імена та визначення цих відношень, але й їх властивості. ОМСП визначає набір таких властивостей та їх характеристики, що впливають на складність даної моделі і визначають ту дескриптивну логіку, що дозволяє описати ОМСП.

Відповідно до специфіки проблеми пошуку, між суб'єктами та об'єктами цієї сфери існують наступні значущі для проблеми зв'язки:

– між ІР і ПрО;

– між ІР і ІО;

– між інформаційними потребами й ПрО;

– між ПрО й задачами користувачів;

– між користувачами та ПрО;

– між користувачами ІПС.

В ОМСП зв'язки відображаються за допомогою відношень з

$$\mathrm{R} = \mathrm{r_{ier\_cl}} \cup \{\mathrm{r_i}\} \cup \{\mathrm{p_j}\},$$

які дозволяють визначити семантику, область значення та її визначення кожного такого зв'язку. Проаналізувавши властиво-

сті цих зв'язків, можна визначити, яка саме дескриптивна логіка лежить в основі ОМСП та, відповідно, наскільки складну мову для подання такої онтології необхідно використовувати. Це, в свою чергу, дозволяє визначити обчислювальну складність задач, які можна вирішувати з використанням такої онтології.

Відношення «клас-підклас»

Ієрархічні відношення «клас-підклас» r_{ier_cl} (приміром, «експерт» є підкласом класу «користувач», «мультимедійний інформаційний об'єкт» є підкласом класу «ІО») є транзитивними та антисиметричними:

– якщо X належить до класу A, а A є підкласом B, то X належить до B;

– якщо A є підкласом B, B є підкласом C, то є підкласом C;

– якщо A є підкласом B, то B не є підкласом A.

За допомогою таких відношень не відображаються мереологічні зв'язки різних типів – приміром, відношення "є членом групи" не можна відображати таким чином, тому що екземпляр класу "користувач" не є екземпляром класу "група користувачів". Це викликає потребу включити до ОМСП інші ієрархічні відношення із специфічною для ПрО специфікою.

Відношенням «клас-підклас» в ОМСП пов'язані такі класи (табл. 1):

Таблиця 1. Ієрархічні властивості в ОМСП

Надклас	Підклас
Онтологія	Онтологія ПрО; Таксономія ІО; Лексична онтологія; Тезаурус; Wiki-онтологія
Тезаурус	Тезаурус задачі; Тезаурус користувача; Тезаурус ІР; Тезаурус ІО
Користувач	Експерт; Член групи
ІР	Природномовний ІР; Мультимедійний ІР; Семантично розмічений ІР
ІО	Людина; Організація; Документ; Web-сервіс

Об'єктні властивості ОМСП

Відношення, специфічні для цієї предметної області – відношення семантичного пошуку, що виражаються через властивості класів, значеннями яких є екземпляри інших класів (приміром, клас «тезаурус» має властивість «побудований на основі», значення якого належить до класу «онтологія ПрО», а клас «тема» має властивість «містить», значення якого належить до класу «запит». В даній ПрО не визначені специфічні відношення, які мають властивості, що можуть застосовуватися для логічного виведення (тразитивність, рефлексивність, симетричність тощо). Такі відношення, в яких і область значення, і область визначення є екземплярами класів ОМСП, з точки зору онтологічного аналізу відповідають об'єктним властивостям $\{r_i\}$ відповідної онтології (табл. 2).

Таблиця 2. Об'єктні властивості в ОМСП

Область значень	Відношення	Область визначення
Користувач	Використовує	Онтологія ПрО; Онтологія ІО; Тезаурус задачі; Тезаурус онтології; Тезаурус користувача
Тезаурус ІР; Тезаурус ІО; Тезаурус задачі; Лексична онтологія ПрО; Тезаурус користувача; Запит; Рекомендація	Базується на	Онтологія ПрО; Онтологія ІО; Тезаурус задачі; Запит
Тема; Користувач; Результат запиту	Є об'єднанням	Запит; Група користувачів; ІО; ІР
Результат запиту; Рекомендація	Є результатом	Запит
Агент користувача	Є представником	Користувач; Група користувачів

Через те, що ОМСП створюється для формалізації вже відомих відношень, а не для впорядкування термінології, то недоцільно створювати класи-синоніми: альтернативні назви понять можна вказувати тільки у поясненні або у визначенні класу.

Мереологічні відношення в ОМСП, що відображають різні види специфічних для ПрО зв'язків типу "частина-ціле" (приміром, відношення «входить до складу» пов'язує екземпляри класу «IP» з екземплярами класу «результати пошуку», а екземпляри класу «користувач» з екземплярами класу «група користувачів») в загальному випадку не є транзитивними.

Таким чином, для сфери семантичного пошуку не виявлено важливих транзитивних або симетричних відношень між екземплярами одного класу. Приміром, якщо користувач А вважає експертом користувача В, а користувач В вважає експертом користувача С, то з цього не випливає, що користувач А вважає експертом користувача С. Це пов'язано з тим, що, як правило, екземпляри одного класу не взаємодіють безпосередньо один з одним в процесі пошуку, а їх відношення можуть встановлюватися тільки через відношення з екземплярами інших класів. Приміром, екземпляри класу "користувач" можуть бути пов'язані через екземпляри класу "тезаурус", що використовуються у запиті, або через екземпляри класу "IP", що є результатами пошукової процедури.

Властивості даних ОМСП

Такі відношення, в яких область значення є екземплярами класів ОМСП, а область визначення – іншими типами даних, з точки зору онтологічного аналізу відповідають даних властивостям $\{p_i\}$ відповідної онтології (табл. 3).

Властивості даних в ОМСП дозволяють встановити конкретні значення властивостей екземплярів класів, явно вказавши їх семантику та характеристики. Вказуючи тип значення властивості, можна не тільки задавати стандартні типи даних (число, рядок тощо), але й задати значення із скінченої множини, описавши таким чином всі припустимі варіанти та вказавши відношення між цими значеннями (приміром, часткову впорядкованість або синонімію). Прикладами таких множин можуть бути різні варіанти подання дати або часу, що надалі будуть інтерпретуватися однаково.

Ці відношення не мають додаткових властивостей, які можуть враховуватися в процесі обробки онтології, і тому не впливають на складність ОМСП.

На основі обробки типів цих значень будується множина Т для ОМСП.

Онтологічна модель задачі користувача

Формально проблема побудови онтології задачі користувача полягає у наступному: за онтологією ПрО O_{domain},

$$O_{domain} = \langle X_{domain}, R_{domain}, F_{domain}, T_{domain} \rangle,$$

та набором Wiki-сторінок W_{user}, семантична розмітка яких базується на O_{domain}, побудувати "легковажну" онтологію задачі користувача O_{user}, знання якої є підмножиною знань з O_{domain}. Слід зазначити, що джерела та методи побудови цієї онтології ПрО знаходяться поза сферою розгляду даної роботи – вона може мати довільну структуру та бути сформована як безпосередньо експертами ПрО, так і за допомогою різноманітних засобів здобуття онтологічних знань [6].

$$O_{user} = \langle X_{user}, R_{user}, F_{user}, T_{user} \rangle,$$

така, що

$$X_{user} \subseteq X_{domain},$$

тобто

$$X_{cl_{user}} \subseteq X_{cl_{domain}},$$

$$X_{ind_{user}} \subseteq X_{ind_{domain}};\ R_{user} \subseteq R_{domain},$$

тобто

$$r_{ier\_cl_{user}} = r_{ier\_cl_{domain}},$$

$$\{r_{user_j}\} \subseteq \{r_{domain_i}\}, i = \overline{0,n},\ j = \overline{0,m},\ m \le n;$$

$$\{p_{user_k}\} \subseteq \{p_{domain_l}\}, l = \overline{0,q},\ k = \overline{0,t},\ t \le q,$$

$$F_{user} = \varnothing;\ T_{user} \subseteq T_{domain}.$$

Таблиця 3. Властивості даних в ОМСП

Область значень	Відношення	Область визначення		
		Назва	Тип	Кількість значень
Користувач	Використовує	Ключове слово	Рядок	Обов'язкове, єдине
Користувач	Має ідентифікатор	Ідентифікатор	Рядок	Обов'язкове, єдине
Користувач	Має пароль	Пароль	Рядок	Обов'язкове, єдине
Користувач	Має сайт	Сайт	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має сферу інтересів	Ключове слово	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має професію	Професія	Рядок	Не обов'язкове, єдине або кілька
Користувач	Має освіту	Освітній рівень	Рядок, значення з множини {неповна середня, середня, вища, вчений ступінь}	Не обов'язкове, єдине
Користувач	Рік народження	Рік	Число з 4 знаків	Не обов'язкове, єдине
Користувач	Живе у	Країна Місто Населений пункт	Рядок	Не обов'язкове, єдине або кілька
Користувач	Народився у	Країна Місто Населений пункт	Рядок	Не обов'язкове, єдине або кілька
Онтологія ПрО	Містить	Термін	Рядок	Обов'язкове, єдине або кілька
Тезаурус	Містить	Термін	Рядок	Обов'язкове, єдине або кілька
Запит	Містить	Ключове слово	Рядок	Обов'язкове, єдине або кілька

Така робота має виконуватися в тому разі, якщо користувач починає працювати над великою та досить складною проблемою, рішення якої буде потребувати інформації протягом досить значного часу, значно більшого, ніж час, потрібний на побудову власної онтології (приміром, плануючи дослідження на кілька років, доцільно витратити кілка годин на те, щоб надалі отримувати семантично відфільтровані відомості).

Множина екземплярів класів ОМСП

ОМСП поповнюється екземплярами класів у процесі функціювання системи пошуку. Наприклад, екземпляри кори-

стувачів створюються внаслідок реєстрації користувачів у системі та можуть доповнюватися новими значеннями властивостей у процесі виконання користувачами пошукових запитів, тоді як екземпляри груп користувачів створюються самими користувачами відповідно до їх власних інформаційних потреб.

Використання онтологічних знань у персоніфікованому семантичному пошуку

У загальному випадку співставлення двох незалежних онтологій, які знаходяться в репозиторії онтологій [11], є надзвичайно складною задачею, що потребує багато часу та додаткової обробки. Але в інформаційному пошуку використовуються онтологічні моделі, які мають достатню для задачі, але досить обмежену складність. Такі моделі можуть використовувати знання з довільних онтологій ПрО, але містити тільки обмежену їх підмножину та не застосовувати складний набір відношень між класами та атрибутами (але сам алгоритм побудови таких спрощених моделей за довільними онтологіями може бути досить складним та знання-містким).

Застосування ОМСП у процесі семантичного пошуку

Відомості, що представлені в ОМСП, використовується на різних етапах семантичному пошуку для (табл. 4):

– переформулювання запитів користувачів відповідно до їх реальних інформаційних потреб;

– фільтрації результатів пошуку, отриманих від зовнішніх ІПС, відповідно до їх пертинентності поточним інформаційним потребам користувача;

– використання досвіду співтовариства користувачів з областями інформаційних потреб, що перетинаються, для проактивного пошуку та надання рекомендацій;

– оцінка відповідності рівня складності контенту знадених ІР здатностям користувача до сприйняття інформації в обраній ПрО.

Таблиця 4. Екземпляри класів в ОМСП

Тип операції	Екземпляри класів ОМСП
Переформулювання запитів користувачів	Користувач, Онтологія ПрО, Тезаурус онтології, Лексична онтологія ПрО, Тезаурус задачі, ІО, Тезаурус ІО, Запит, Група запитів
Фільтрація результатів	Користувач, Онтологія ПрО, Зважений тезаурус онтології, Лексична онтологія ПрО, Зважений тезаурус задачі, Зважений тезаурус ІР, Зважений тезаурус ІО
Використання досвіду співтовариства користувачів	Користувач, Група користувачів, Тема , Онтологія ПрО, Тезаурус онтології, Лексична онтологія ПрО, Тезаурус задачі, Тезаурус множини онтологій, Запит, Група запитів, Результат запиту, Агент користувача
Оцінка рівня складності контенту	Тезаурус ІО Тезаурус користувача

Слід враховувати, що така модель інформаційного пошуку орієнтована на користувачів із сталими та досить глибоко усвідомленими інформаційними потребами, тому процес пошуку інформації може розглядатися як циклічний процес, різні етапи якого повторюються у певній послідовності, а наповнення ОМСП екземплярами класів продовжується протягом усієї взаємодії користувача з системою.

Алгоритми такого застосування елементів онтології та їх властивості розглянуті у наступних розділах.

Семантичний пошук на основі зіставлення тезаурусів

Будемо вважати, що сферу інтересів користувача в цілому формально характеризує онтологія відповідної ПрО (або набір таких онтологій, що відповідають різним аспектам діяльності однієї особи), а його поточні інтереси – природномовний опис задачі.

Природномовний опис задачі – це неструктуровані або слабо структуровані дані, аналіз яких потребує попередньої обробки, а онтологія ПрО у загальному випадку має довільний розмір та структуру, що надзвичайно ускладнює її безпосереднє використання у пошуку. Тому за обома цими об'єктами пропонується будувати *тезаурус задачі*, що поєднує їх переваги та дозволяє позбутися недоліків.

Для того, щоб відфільтрувати результати роботи зовнішньої ІПС і отримати тільки ті ІР, що пертинентні інформаційним потребам користувача, необхідно попередньо сформувати тезаурус задачі користувача та ПрО, що цікавить цього користувача, і тезауруси цих ІР, а потім порівняти ці тезауруси.

Алгоритм побудови простого тезаурусу задачі

Простий тезаурус задачі

$$\mathrm{Th} = \langle \mathrm{T}, \mathrm{r}_{\mathrm{ier\_cl}}, \varnothing, \varnothing \rangle$$

будується за обраною користувачем онтологією ПрО та за описом поточної задачі (рисунок).

Опис задачі може бути подано через ПМ-текст, який містить елементи, пов'язані з елементами онтології, або через умови, яким мають задовольняти терміни ПрО, що стосуються цієї задачі. Розглянемо два етапи побудови такого тезаурусу.

Етап 1. Автоматизована генерація простого тезаурусу задачі за описом задачі.

Етап 2. Розширена генерація простого тезаурусу задачі за набором умов, що використовують інші елементи онтології О, крім екземплярів та класів.

На етапі 1 теж можна виокремити два кроки (на практиці може застосовуватися їх поєднання). Етап 1.1 полягає у тому, що користувач явно та вручну з автоматично побудованого переліку класів та екземплярів X обирає ті, які він вважає пертинентними задачі:

$$\mathrm{T} = \{\mathrm{x}_{\mathrm{t}_1}, ..., \mathrm{x}_{\mathrm{t}_p}\}, 1 \le \mathrm{t}_k \le \mathrm{n}, \forall \mathrm{x}_{\mathrm{t}_k} \in \mathrm{X}.$$

В найпростіших випадках на цьому кроці побудова тезаурусу може завершуватися, але це потребує від користувача більше зусиль.

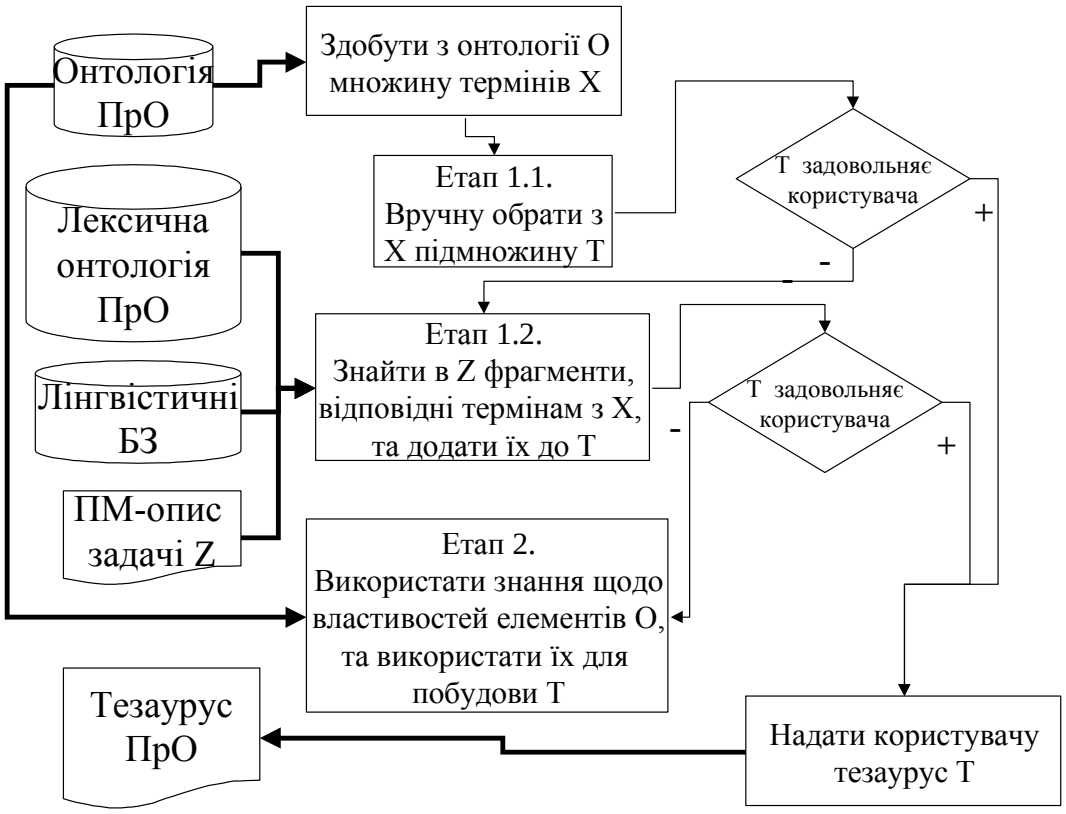

Рисунок. Алгоритм побудови тезаурусу задачі

Етап 1.2 використовує різноманітні методи обробки природномовного опису задачі (лінгвістичний аналіз, статистична обробка, аналіз семантичної розмітки), які дозволяють виявити фрагменти ПМ-тексту, пов'язані з термінами О.

Ті терміни, для яких в описі задачі знайдено відповідні фрагменти, заносяться до простого тезаурусу задачі. На цьому етапі використовується множина X онтології О. Один з методів обробки ПМ-опису задачі Z базується на лексичній онтології $L_{ПрО} = \left\langle X_{lex} = X_{ПрО} \cup T_{ПрО}, \{r_{lex}\}, \varnothing \right\rangle$: якщо

$$s_{j_p} \in T_{ПрО}, p = \overline{1, q_j}$$

є фрагментом ПМ Z, та

$$r_{lex}(s_{j_p}) = x_j \in X,$$

тоді x_j додається до множини Т тезаурусу задачі.

Такий метод високо ефективний у тому разі, якщо вже накопичено великий обсяг лексичної онтології. Для цього можуть використовуватися безпосереднє поповнення лексичної онтології користувачами, експорт лінгвістичних знань з відповідних словників та баз знань, а також аналіз семантично розмічених текстів, що буде розглянуто далі.

Але у багатьох випадках доцільно використовувати у побудові тезаурусу інформацію про інші елементи онтології, які дозволяють враховувати властивості окремих термінів та їх відношення з іншими термінами. В такому разі застосовується етап 2, який спрямований на вдосконалення початково сформованого тезаурусу відповідно до явно сформульованих умов користувача. Ці умови обумовлені специфікою задачі, але не є похідними від її опису. Їх можна розглядати як набір метаправил для опису інформації, яку користувач прагне отримати.

Етап 2 можна представити як функцію, що перетворює онтологію О на простий тезаурус

$$f_{Th}(O) \to Th,$$

є відображенням набору умов, які формулює користувач щодо тих класів та екземплярів класів онтології О, що потрібно включити до тезаурусу задачі.

Набір цих умов можна розглядати як об'єднання (диз'юнкцію) d множин властивостей елементів онтології (класів та об'єктів)

$$f_{Th}(O) = \bigcup_{i=1}^{d} f_{Th_i}(O),$$

що можуть пов'язувати кожен елемент, що задовольняє цим вимогам, їх з фіксованими значеннями або іншими елементами онтології, – кон'юнкція вимог

$$f_{Th_i}(O) = \bigcap_{j=1}^{d_i} f_{Th_{ij}}(O).$$

Потрібно спочатку здобути з онтології набори для всіх

$$f_{Th_i}(O), i = \overline{1, d},$$

а потім побудувати їх об'єднання. Усі $f_{Th_i}(O), i = \overline{1, d}$ будуються наступним чином: оброблюється окремо кожна умова, за нею генеруються d_i наборів елементів (класів та екземплярів) онтології О, після цього побудувати перетин цих множин.

Побудова множини елементів тезауруса для кожної умови

$$f_{Th_{ij}}(O), i_j = \overline{1, d_{ij}}$$

складається з наступних дій, що послідовно враховують інформацію з усіх елементів онтології О.

Множина класів X_{cl} є джерелом тих термінів ПрО, що стосуються поточного набору задачі користувача, і кожен з цих термінів може бути доданий до $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах. Окремо оброблюються негативні умови щодо класів – кожен з цих термінів може бути видалений з $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах.

Множина класів X_{ind} є джерелом тих екземплярів об'єктів ПрО, що стосуються поточного набору задачі користувача, і кожен з цих термінів може бути доданий до $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах. Можуть також додаватися до тезаурусу всі екземпляри певного класу. Окремо обробляються негативні умови щодо екземплярів класів – кожен з цих термінів може бути видалений з $Th_{ij}(O)$ користувачем безпосередньо або через перерахування в умовах.

Множина відношень між елементами онтології

$$R = r_{ier\_cl} \cup \{r_i\} \cup r_{ier\_prop} \cup \{p_j\} \cup p_{ier\_prop}$$

може бути використана для побудови $Th_{ij}(O)$ наступним чином:

– ієрархічні відношення r_{ier_cl} дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ надкласи та підкласи обраних класів, задавши глибину обробки, наприклад, всі підкласи терміну "наукова публікація" на глибину q;

– об'єктні властивості $\{r_i\}$, що встановлюють відношення між екземплярами класів, дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ ті терміни, що пов'язані з обраними термінами і відношеннями з $\{r_i\}$, наприклад, для терміну x включити до $Th_{ij}(O)$ ті терміни, з якими в онтології O цей термін пов'язаний об'єктним відношенням "співпрацює з";

– ієрархічні відношення між об'єктними властивостями r_{ier_prop} дозволяють використовувати підкласи відношень замість самих відношень з $\{r_i\}$, наприклад, якщо для терміну x потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, з якими в онтології O цей термін пов'язаний об'єктним відношенням "співпрацює з" з множини $\{r_i\}$, то потрібно включити до $Th_{ij}(O)$ також ті терміни, з якими в онтології O цей термін пов'язаний об'єктним відношенням "працює в одному відділі", яке є підкласом відношенням "співпрацює з";

– властивості даних $\{p_j\}$, що встановлюють відношення між екземплярами класів та даними, дозволяють користувачу включити (або видалити – для негативних умов) до тезаурусу $Th_{ij}(O)$ ті терміни, що задовольняють певним умовам, в яких задаються значення властивостей даних з $\{p_j\}$, наприклад, включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, які в онтології пов'язані відношенням даних "Рік народження" більшим за 1900;

– ієрархічні відношення між властивостями даних p_{ier_prop} дозволяють використовувати підкласи відношень замість самих відношень з $\{p_j\}$, наприклад, якщо для терміну x потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, що задовольняють певним умовам, в яких задаються значення властивостей даних з $\{p_j\}$, потрібно включити (або видалити – для негативних умов) до $Th_{ij}(O)$ ті терміни, які в онтології задовольняють умовам щодо підкласів цих властивостей, наприклад, якщо є умова щодо відношення даних "Кількість публікацій" більше за 10, то потрібно включити до тезаурусу елементи, для яких "Кількість публікацій Scopus" більше за 10;

– множина характеристик класів онтології F_{cl}, що можуть застосовуватися для логічного виводу, обробляються в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують: якщо два класи x_1 та x_2 в онтології O еквівалентні, і клас x_1 занесено до теза-

урусу $Th_{i_j}(O)$, тоді треба занести до тезаурусу й клас x_2;

– множина характеристик об'єктних властивостей екземплярів класів онтології F_{prop}, що можуть застосовуватися для логічного виводу, обробляються в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують (на практиці це зазвичай не застосовується);

– множина нелогічних правил ПрО M використовується для побудови тезаурусу наступним чином: якщо об'єкт з онтології O (клас або екземпляр класу) o_1 належить до тезаурусу $o_1 \in Th$ та в онтології в M міститься правило "якщо $o_1 \in O$, тоді $o_1 \in O$", треба додати до тезаурусу елемент o_2. Приклад: в тезаурус треба додати усі екземпляри осіб пенсійного віку (тобто тих, вік яких більше певної константи). Якщо вік особи невідомий, але відомо, що вона має дитину пенсійного віку, а за нелогічними правилами ПрО вік батьків більше за вік дитини, тоді треба додати до тезаурусу цей екземпляр класу.

Інші елементи онтології O не використовуються безпосередньо для побудови Th_O, але вони можуть застосовуватися для поповнення та вдосконалення самої онтології O.

Алгоритм побудови складеного тезаурусу задачі

Складений тезаурус задачі будується на основі простих або складних тезаурусів задач. Для цього застосовуються теоретико-множинні операції перетину, об'єднання та різниці.

Якщо

$$Th_1 = \langle T_1, r_{ier\_cl}, \varnothing, \varnothing \rangle$$

та

$$Th_2 = \langle T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$$

– прості тезауруси задачі, і $Th_1 \neq Th_2$, тоді їх об'єднання

$$Th_{об} = \langle T_1 \cup T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle,$$

перетин

$$Th_{перет} = \langle T_1 \cap T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$$

та різниця

$$Th_{різ} = \langle T_1 / T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$$

– складені тезауруси.

Якщо $Th_1 = \langle T_1, r_{ier\_cl}, \varnothing, \varnothing \rangle$ та $Th_2 = \langle T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$ – складені тезауруси задачі, і $Th_1 \neq Th_2$, тоді їх об'єднання $Th_{об} = \langle T_1 \cup T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$, перетин $Th_{перет} = \langle T_1 \cap T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$ та різниця $Th_{різ} = \langle T_1 / T_2, r_{ier\_cl}, \varnothing, \varnothing \rangle$ теж є складеними тезаурусами.

Для того, щоб побудувати складений тезаурус, користувачеві потрібно обрати два раніше створених тезауруси та визначити, яку саме теоретико-множинну операцію треба до них застосувати.

Використання складених тезаурусів дозволяє застосовувати тезаурусне представлення знань ПрО, що цікавить певного користувача, у багатьох задачах, пов'язаних з пошуком, аналізом та структуруванням ресурсів Web, відображаючи персональні уявлення окремого користувача щодо сфери його інформаційних потреб.

На змістовному рівні такий тезаурус – це сукупність термінів ПрО, відомих користувачеві, тобто користувач обирає лише ту підмножину онтологічних термінів різних онтологій, які відповідають його особистим інтересам та уявленням.

Такий тезаурус може застосовуватися не тільки безпосередньо в процесі пошуку, але він є зручним інструментом для розширення функціоналу семантичної ІПС. Наприклад, тезаурус ПрО дозволяє виконувати наступні операції, алгоритми здійснення яких більш детально описані в наступних розділах:

– оцінка складності природномовного тексту для сприйняття конкретним користувачем;

– побудова груп користувачів з подібними інформаційними потребами для рекомендуючих систем;

– виконання теоретико-множинних операцій над тезаурусами, що забезпечують повторне використання тезаурусів для нових задач;

– побудова та використання лексичних онтологій.

Алгоритм побудови зваженого тезаурусу задачі

Зважений тезаурус задачі

$$Tw = \{< t_j, w_j >\}, t_j \in T, j = \overline{1,s}$$

будується за множиною

$$T = \{x_j \in X\}, j = \overline{1,s}$$

тезаурусу задачі Th (простим або складеним) наступним чином: кожному елементу з T ставиться у відповідність вага (позитивна чи негативна) w_j – кількісна характеристика важливості цього терміну для поточної задачі користувача. Ця оцінка може задаватися користувачем явно (якщо зважений тезаурус будується за простим або складеним тезаурусом) або обчислюватися за раніше заданими оцінаки (якщо зважений тезаурус будується як об'єднання двох раніше побудованих тезаурусів, тоді значення оцінок термінів підсумовуються).

Аналіз виразної здатності тезаурусу задачі

Для того, щоб використовувати запропонований підхід, необхідно довести, що виразна здатність таких тезаурусів є задовільною для виконання семантичного пошуку. Незважаючи на досить просту структуру самого тезаурусу задачі, його виразна здатність визначається методом його побудови, який використовує всі ті знання ПрО, що містяться у відповідній онтології та можуть бути застосовані для пошуку у тому випадку, якщо б у співставленні задачі користувача та ІР використовувалися б довільні онтології.

Слід зазначити, що в інших моделях пошуку можуть застосовуватися інші аспекти онтологічних знань, що не використовуються у цій моделі. Крім того, деякі моделі пошуку дозволяють користувачеві явно керувати тим, які саме знання треба враховувати в процесі пошуку (і запропонована модель належить саме до цього класу), тоді як інші моделі не дозволяють користувачу впливати на такий вибір. Переваги та недоліки окремих моделей знаходяться поза розглядом даної роботи.

Можна стверджувати, що певні знання з онтології *зафіксовані* у тезаурусі, якщо їх відсутність в онтології призвела б до таких змін у тезаурусі, побудованому за цією онтологією, які вплинули б на результати пошуку. Але таке визначення не дозволяє оцінити виразну здатність тезаурусу. В процесі пошуку виконується співставлення моделі задачі та тезаурусу задачі. В цьому співставленні аналізуються тільки ті елементи тезаурусу задачі, для яких знайдені певні відповідності у моделі задачі. Тому ті елементи тезаурусів онтологій, які не входять до тезаурусу задачі, не впливають на впорядкування результатів пошуку. Якщо у тезаурусу онтології можна відобразити певний елемент знань онтології, то ці знання можуть бути відображені в тезаурусі задачі.

Тому будемо вважати, що якщо певний елемент онтології може вплинути на вміст тезаурусу, то виразна здатність тезаурусу є достатньою для його відображення стосовно пошукової процедури.

Твердження 1. Алгоритм побудови простого тезаурусу задачі дозволяє використовувати знання щодо структури ПрО з онтології, яку користувач вважає пертинентною його поточній задачі, відповідно до тих умов, які користувач вважає доцільним застосовувати в обраній пошуковій моделі.

Доказ. Проаналізуємо окремо кожен компонент онтології O, його використання для пошуку та те, як цей компонент відображено в простому тезаурусі задачі

$$Th_O = \langle Th_{cl} \cup Th_{ind} \rangle .$$

Розглянемо дане для обох етапів алгоритму побудови простого тезаурусу задачі.

Слід зазначити, що на етапі 1 використовується лише множина X онтології

O. Завдяки цьому обчислювальна складність даного алгоритму не вище, ніж лінійна залежність від кількості класів та екземплярів класів в онтології O.

На етапі 2, крім класів та екземплярів онтології O, інші елементи використовуються не безпосередньо, а для поповнення та вдосконалення самої онтології O, тому обчислювальна складність побудови тезаурусу залежить від кількості та типу умов (наприклад, умов "включити всі екземпляри зі значенням "рік створення" більшим за 2000"), до яких входять ці елементи, та від розміру відповідних множин онтології O (наприклад, в онтології ПрО може бути 3 або 33 властивості даних).

Відповідно до алгоритму побудови простого тезаурусу задачі, для побудови тезаурусу можуть бути застосовані:

– ієрархічні відношення для розширення тезаурусу надкласами та підкласами обраних класів на обрану глибину обробки;

– об'єктні властивості для розширення тезаурусу екземплярами класів, що пов'язані з вже обраними екземплярами класів певними об'єктними відношеннями;

– ієрархічні відношення між об'єктними властивостями для розширення тезаурусу, використовувати підкласи об'єктних відношень замість відношень, визначених у попередньому пункті;

– властивості даних екземплярів класів для розширення тезаурусу тими термінами, що задовольняють певним умовам;

– ієрархічні відношення між властивостями даних розширювати тезаурус, використовуючи підкласи відношень даних замість визначених користувачем відношень даних;

– характеристики класів онтології для використання еквівалентних класи замість класів, обраних користувачем;

– характеристики об'єктних властивостей екземплярів класів для логічного виводу в процесі побудови тезаурусів відповідно до того, яку саме властивість вони фіксують;

– нелогічні правила ПрО для розширення тезаурусу, якщо задовольняються умови цих правил.

Слід відмітити, що для довільної онтології та довільного набору умов користувача досить складно оцінити час побудови тезаурусу, точніше, можна спрогнозувати найгірший варіант, але на практиці час обробки значно менший. Доцільніше аналізувати окремі випадки як онтологій, так і умов. Надалі більш детально буде оцінено обчислювальна складність побудови простого тезаурусу задачі за Wiki-онтологією – онтологією, яка будується на основі семантичної розмітки Wiki-сторінок.

Твердження 2. Алгоритм побудови складеного тезаурусу задачі дозволяє використовувати ту частину знання щодо структури ПрО, які застосовувалися в алгоритмі побудови тезаурусів онтологій, що безпосередньо пов'язані з поточною задачею користувача та можуть бути співставленні з фрагментами її природномовного опису (безпосередньо, з використанням логічного виведення або специфічних для ПрО правил) у тому випадку, коли складений тезаурус будується за простими тезаурусами суттєво різних ПрО.

Доказ. На відміну від операцій перетину, об'єднання та різниці онтологій, алгоритм побудови складеного тезаурусу працює дуже швидко, а його обчислювальна складність залежить лише від розміру простих тезаурусів.

Можна вважати, що простий тезаурус, побудований за об'єднанням онтологій, містить ту саму інформацію для пошуку, що й об'єднання простих тезаурусів, що побудовані за кількома незалежними онтологіями (це визначається алгоритмом побудови простого тезаурусу задачі), але побудова об'єднання простих тезаурусів потребує значно менше часу.

Для випадку, коли поєднуються незалежні онтології, це випливає з алгоритму побудови простого тезаурусу: та частина об'єднаної онтології, що була побудована внаслідок об'єднання, не використовується в операціях поповнення тезаурусу, тому що вона не містить відповідних

термінів і не задовольняє пов'язаних з ними умовам.

Якщо поєднуються пов'язані онтології, то можливі три основні варіанти, суттєві для побудови тезаурусу:

– деякі онтологічні знання дублюються в різних онтологіях, і тому це не впливає на побудову тезаурусу;

– деякі онтологічні знання суперечать і не можуть бути поєднані, тобто в об'єднаній онтології обирається лише один з кількох можливих варіантів зв'язку між термінами, і тоді у побудові тезаурусу використовуються ті знання, які користувач вважає більш пертинентними. Якщо будується об'єднання тезаурусів таких онтологій, то такий спосіб виведення, що привів до додання певного терміну до результуючого тезаурусу, буде присутній хоча б одній з тих онтологій, що об'єднуються;

– деякі онтологічні знання не суперечать між собою, але їх об'єднання надає нові зв'язки між термінами онтології. В такому випадку тезаурус задачі, побудований за об'єднаною онтологією, може містити деякі терміни, що відсутні в усіх тезаурусах, побудованих за тими онтологіями, що об'єднуються. Але через те, що обробка менших онтологій на етапі 1.1 значно простіша, на практиці ці терміни досить часто користувач додає вручну.

Таким чином, запропонований підхід є ефективним тільки для онтологій, що описують різні ПрО (або суттєво різні аспекти ПрО) і тому мають набори термінів (класів та екземплярів), що не перетинаються. Аналогічно оцінюються перетин та різниця онтологій та побудованих за ними тезаурусів.

Твердження 3. Обчислювальна складність першого етапу алгоритму побудови простого тезауруса за онтологією O лінійно залежить від кількості термінів у множині X онтології O, обсягу лексичної онтології та розміру опису задачі.

Доказ. Відповідно алгоритму побудови тезаурусу, на етапі 1.1. користувач проглядає весь перелік термінів онтології O та для кожного з n елементів у множині X онтології O приймає рішення щодо того, чи занести цей елемент до тезаурусу. Від інших параметрів онтології та задачі цей етап не залежить явно. Тому, якщо вважати швидкість прийняття рішення користувачем за постійну величину, то обчислювальна складність етапу 1.1. не перевершує n.

На етапі 1.2 виконується співставлення елементів множини X з природномовним описом задачі за допомогою

$$L_{\Pi pO} = \langle X_{lex} = X \cup T, \{r_{lex}\}, \varnothing \rangle$$

– лексичної онтології, в якій кожному елементу (класу або екземпляру класу) з X $\forall x_j \in X, j = \overline{1,m}$ відповідає скінчена кількість фрагментів ПМ

$$s_{jp} \in T_{\Pi pO_i}, p = \overline{1,q_j}, r_{lex}(s_{jp}) = x_j,$$

що співвідносяться з цим елементом.

Для елементів множини X виконується $\sum_{j=1}^{n} q_j$ перевірок-співставлень для кожного фрагмента, час виконання яких залежить від розміру опису задачі l, якщо $q = \max_{j=1}^{n} q_j$, то обчислювальна складність алгоритму для етапу 1.2 не більш як n*q*l.

Твердження 4. Алгоритм другого етапу побудови простого тезауруса за онтологією O є скінченим, і його обчислювальна складність залежить лінійно від кількості термінів у множинах X, M та R онтології O, кількості вимог та обмежень, за якими елемент з X може бути додано до тезаурусу, рівня вкладеності вимог та кількості елементів.

Доказ. Скінченність алгоритму випливає із скінченності множин елементів та умов щодо них, що перевіряються. Виникненню циклів запобігає те, що кожна перевірка для кожного елемента не виконується більше одного разу: на кожному кроці виконання спочатку визначається множина елементів, які потрібно перевірити, – як об'єднання всіх множин, що задовольняють початковим умовам, а потім

з цієї множини видаляються елементи, що не задовольняють обмеженням.

Для кожного елементу множини X виконується перевірка для кожної з q вимог, що відображають переконання користувача щодо цікавлячої його ПрО відповідно до алгоритму виконання етепу 2.

Більш точно оцінювати обчислювальну складність алгоритму побудови тезаурусу доцільно для окремих випадків онтологій ПрО, що використовуються у практичних задачах, наприклад, для Wiki-онтологій, що будуть розглянуті далі.

Алгоритм побудови тезаурусів ІР

Побудова тезаурусів природномовних ІР дозволяє здобути з неструктурованих текстів відомості, що стосуються тієї задачі, яка цікавить користувача.

Для цього можуть використовуватися лексичні онтології або різноманітні інші засоби лінгвістичного аналізу. Слід зазначити, що лексична онтологія містить відносно невелику підмножину знань щодо ПМ-представлення термінів, пертинентних задачі користувача, і тому час аналізу текста на її основі має обчислювальну складність, що залежить від розміру лексичної онтології, побудованої для тезаурусу задачі (простого або складеного).

Тезаурус ІР

$$Th_{IR} = \langle X_{IR} \subseteq X_{Th}, \varnothing, \varnothing, \varnothing \rangle$$

це підмножина тезаурусу задачі

$$Th = \left\langle X_{Th} \subseteq X, r_{ier\_cl} \in R, \varnothing, \varnothing \right\rangle,$$

який містить тільки ті його терміни, для яких знайдено відповідні фрагменти у контенті цього ІР. Таким чином, склад тезаурусу ІР залежить як від тезаурусу задачі, для якої він будується, так і від методу співставлення контенту ІР із термінами цього тезаурусу.

Алгоритм побудови тезаурусу ІР з використанням лексичної онтології складається з наступних кроків:

$\forall x_j \in X_{Th}, j = \overline{1,q}$ у лексичній онтології

$$L_{ПрО} = \begin{pmatrix} X_{lex} = X_{ПрО_i} \cup T_{ПрО_i}, \\ R_{lex} = \{r_{lex}\}, \varnothing \end{pmatrix}$$

шукати відповідні фрагменти ПМ

$$s_{j_p} \in T_{ПрО}, p = \overline{1,q_j}, r_{lex}(s_{j_p}) = x_j,$$

якщо в контенті ІР знайдено хоча б один з $s_{j_p}\, p = \overline{1,q_j}$, тоді додати додати x_j до множини X_{IR}:

$$\forall x_j \in X_{IR} \exists s_{j_p} : r_{lex}(s_{j_p}) = x_j.$$

Для необхідності аналізу великої кількості ІР для виконання кожного пошукового запиту виникає необхідність використовувати такий алгоритм побудови їх тезауруса, обчислювальна складність якого лінійно залежить від обсягу ІР та від обсягу опису задачі, для якої він будується.

Цей алгоритм застосовується тільки до тих природномовних ІР, що не супроводжуються метаописами. За наявності метаописів (у форматі RDF [12] чи OWL [13]) для довільних ІР (природномовних, мультимедійних, структурованих тощо) аналогічний алгоритм застосовується до цих метаданих: аналізуються елементи метаопису.

Слід зазначити, що запропонований алгоритм виконує співставлення не для всіх елементів лексичної онтології, а лише для тих, що відповідають тезаурусу задачі, що значно зменшує час його виконання через порівняно невелику кількість співставлень. Безпосередньо онтологія ПрО та опис задачі користувача в ньому не використовуються, але відповідні знання з них містяться в тезаурусі задачі, що будується за ними.

Алгоритм побудови зваженого тезаурусу ІР

Зважений тезаурус ІР

$$Tw_{IR} = \{< t_j, w_j >\}, t_j \in T, j = \overline{1,s}$$

будується за множиною X_{IR} тезаурусу ІР Th (простим або складеним) наступним чином: кожному елементу з T ставиться у відповідність вага w_{IR_j} – позитивна кількісна характеристика важливості цього терміну для ІР, що аналізується.

Ця оцінка обчислюється з урахуванням кількості успішних співставлень контенту ІР з тими елементами лексичної онтології, що відповідають цьому терміну. Знаходження відповідностей до терміну в заголовку або метаописі ІР може мати більше значення і тому оцінюється більш високо.

Побудова онтології задачі

Алгоритм побудови онтології задачі наведено в [7]. Для побудови онтології задачі доцільно застосовувати семантично розмічені ІР, що використовують поширені стандарти для такої розмітки. На сьогодні найбільш відомим та вживаним засобом для цього є семантичні Wiki, наприклад, такі ІР, що базуються на Semantic MediaWiki [8]. Використання семантичних Wiki-технологій для створення розподілених інформаційних ресурсів не тільки дозволяє досить легко додавати структурування до неструктурованих даних (НСД), але й є джерелом фонових знань для аналізу довільних природномовних текстів відповідної предметної області. Створення е-ВУЕ як семантизованого Wiki-ресурсу дозволяє вдосконалити процес генерації таких знань. Використання онтологічного аналізу – основа для переходу від неструктурованого контенту [9] до розподіленої бази знань, придатної для повторного використання.

Найпростіше використовувати неспеціалізовані енциклопедії та довідники (такі, як електронна версія Великої української енциклопедії [10]), але, якщо користувач має відомості до більш спеціалізованих ресурсів, то їх застосування може збільшити ефективність роботи.

Етапи побудови онтології за Wiki-ресурсом

Якщо користувач явно визначив множину понять ПрО, що його цікавлять, за допомогою множини Wiki-сторінок, тоді алгоритм побудови онтології ПрО має наступне.

Етап 1. Обрати множину Wiki-сторінок X, що пертинентні ПрО.

Етап 2. Здобути з цих сторінок всі категорії та відібрати ті, що пертинентні ПрО (відкинути службові категорії, зайві для задачі категорії тощо). За множиною цих категорій побудувати множину класів ПрО К.

Етап 3. Проаналізувати множину К та за її структурою додати в онтологію ієрархічні відношення між класами.

Етап 4. Порівняти множини К та X і додати в онтологію ПрО екземпляри класів, що відповідають Wiki-сторінкам з X.

Етап 5. Проаналізувати семантичні властивості сторінок з X, обрати з них ті, що стосуються інших сторінок з X та додати відповідні об'єктні властивості до онтології ПрО.

Етап 6. Проаналізувати посилання (семантичні та звичайні) між сторінками з X та додати до онтології ПрО відповідні відношення між екземплярами класів.

Етап 7. Проаналізувати семантичні властивості, що пов'язують сторінки з X з даними. Додати ці властивості до властивостей даних онтології ПрО, а їх значення – до значень цих властивостей відповідних екземплярів онтології.

Рекурсивне розширення онтології ПрО на основі семантизованого Wiki-ресурсу

У тому випадку, коли користувач задає не всю множину термінів ПрО, що його цікавить, а тільки їх початковий набір, алгоритм побудови онтології розширюється наступними етапами:

Етап 2а. Здобути з множини К всі їх підкатегорії та відібрати ті, що пертинентні ПрО (відкинути службові категорії, зайві для задачі категорії тощо). За множиною цих категорій розширити множину класів ПрО.

Етап 2б. Проаналізувати інші екземпляри категорій з К та запропонувати користувачеві додати їх до X.

Повторювати етапи 2а та 2б доти, поки користувач не буде задоволений термінологічним складом ПрО.

Етап 5а. Проаналізувати семантичні властивості, що пов'язують сторінки з X з іншими Wiki-сторінками. За необхідності додати ці сторінки до X, а самі властивості додати до множини об'єктних властивостей онтології ПрО.

Повторювати етап 5а доти, поки користувач не буде задоволений термінологічним складом ПрО.

Алгоритм фільтрації ІР на основі тезаурусів

Як було вказано вище, через велику кількість ІР, доступ до яких забезпечує Web, основна проблема у пошуку інформації пов'язана не із знаходженням усієї множини ІР I, пертинентних (більше або менше) потребам користувача, а у відборі з цієї множини I тих ІР, що найбільш відповідають цій потребі. В даній роботі розглядається та підзадача пошуку в Web, що стосується фільтрації результатів пошуку за набором ключових слів, отриманих від довільної зовнішньої ІПС.

Алгоритм фільтрації результатів запиту користувача до зовнішнього ІПС:

– користувач обирає ІПС, які забезпечують доступ до ІР (у Web, корпоративній мережі, сховищі даних);

– користувач формулює запит, ідентифікуючи свою інформаційну потребу: за допомогою набору ключових слів, умов запиту, документів-зразків тощо – відповідно до можливостей, що надає обрана ІПС;

– користувач обирає онтологію ПрО та за нею створює (формує або обирає зі вже існуючих) зважений тезаурус задачі

$$Tw_{user} = \{< x_k \in T_{ПрО}, w_k >\}, k = \overline{1,s};$$

– запит передається до зовнішньої ІПС, від якої отримують відповідні до запиту результати його виконання – n посилань на ІР та їхні короткі описи

$$I = \{< ref_j, d_j >\}, j = \overline{0,m},$$

де Ref_j – http-адреса відповідного ІР, знайденого ІПС, а d_j – коротка інформація про цей ІР, що зовнішня ІПС надає користувачеві у відповідь на запит;

– якщо множина I не порожня, тобто ІПС знайшла у відповідь на запит хоча б один ІР ($m \geq 1$), то потрібно встановити порядок, в якому пропонувати користувачеві відомості про знайдені ІР. Тоді для всіх ІР з цієї множини I формуються їх зважені тезауруси $Tw_{IR_j}, j = \overline{1,m}$ та відповідні їм словники термінів X_{IR_j}. Елементи цієї множини з к елементів $x_{j_k} \in X_{IR_j}$ – це терміни ПрО, яким відповідають певні фрагменти з d_j – опису j-го ІР з множини I, запропонованої ІПС. У зваженому тезаурусі ІР фіксується також w_{j_k} – вага кожного терміну з X_{IR_j}, що кількісно характеризує його важливість у цьому ІР (відповідно до місця, де знайдено відповідний фрагмент, та залежно від кількості таких успішних співставлень – відповідно до способу оцінювання, обраного користувачем);

За наявності зважених тезаурусів задачі та ІР визначення семантичної близькості між цими об'єктами вирішується за допомогою обчислення коефіцієнту їх близькості:

$$K(Tw_{IR_j}, Tw_{user}) = \sum_{i=1}^{n} w_{user_i} * w_{IR_{j_i}} * f(x_i).$$

Функція $f(x)$ виконує співставлення термінів тезаурусів задачі та ІР:

$$f(x) = \begin{cases} 1: x \in Tw_{IR_j} \\ 0: x \notin Tw_{IR_j} \end{cases}.$$

Коефіцієнт близькості враховує кількість термінів тезаурусу задачі, що знайдено у тезаурусі ІР, так і в тезаурусі ПрО, важливість цих термінів для задачі користувача та важливість цих термінів для контенту ІР. На практиці можуть застосовуватися різні варіанти цього критерію, наприклад, нормовані значення w_{IR_i}, що дозволяють обробляти ІР різного обсягу, але для аналізу невеликих фрагментів ПМ-тексту приблизно однакового розміру, які надають зовнішні ІПС, достатньо використовувати таку оцінку.

Отримавши оцінки для всіх знайдених ІР, можна виконати впорядкування їх списку за цими оцінками, використовуючи довільний алгоритм сортування масивів. Знайдені ІР впорядковуються залежно від значень K_j.

Часова складність алгоритмів сортування різниться від O(n) до $O(n^2)$. Доцільно застосовувати такі алгоритми сортування з часовою складністю O(n log n), як сортування злиттям, швидке сортування, пірамідальне сортування. Але слід враховувати, що стабільні алгоритми сортування, що працюють за час O(n log n), потребують O(n) додаткової пам'яті. Якщо використовуються алгоритми сортування з часовою складністю O(n), такі як сортування комірками, сортування підрахунком, сортування за розрядами, то вони потребують використання додаткової інформації про елементи, приміром, діапазон значень ключа.

В даному випадку елементи множини, що впорядковуються, містять не тільки ключ, за яким здійснюється сортування, але й інформацію про місцезнаходження відповідного файлу, тобто впорядкування відбувалось не у самому масиві елементів, а в масиві ключів, що є посиланнями на інші дані. Такий підхід не спрямований на аналіз повного контенту ІР , але забезпечує той самий рівень аналізу, який використовують користувачі, вручну проглядаючи такі описи: короткий опис може некоректно відображати вміст самого ІР, але це залежить не від засобів аналізу, а від самого ІР.

Користувачеві надають насамперед ті ІР, що мають найбільші значення K_j – коефіцієнтів близькості до ПрО. Можна обмежити множину ІР, що надаються користувачеві, за двома параметрами, – кількістю ІР (наприклад, перші 20 найближчих ІР) та значенням K_j (наприклад, надавати ІР) з $K_j \geq 5$.

Доцільно використовувати одночасно як позитивні (відомі, бажані, релевантні терміни), так і негативні (незнайомі, незрозумілі, нерелевантні проблемі терміни) тезауруси. Якщо в ІР трапляються терміни з негативною вагою, то зменшує семантичну близькість ІР до задачі.

Запропонований підхід доцільно використовувати, якщо:

– користувач досить глибоко обізнаний у ПрО та близьких до неї областях;

– користувач виконує велику кількість запитів з однієї ПрО, пов’язаних з різними задачами;

– користувач виконує велику кількість запитів, пов’язаних з однією ПрО, що відповідають різним її аспектам або етапам і тому потребують різної інформації;

– користувач досить довго займається пошуком інформації, і тому час, який він витрачує на вибір онтології ПрО та побудову тезаурусу задачі, значно менше за той час, який він витрачував на ручний прогляд результатів кожного запиту у цій сфері.

Таким умовам відповідають наукова діяльність (наприклад, моніторинг публікацій в обраній сфері, пошук аналогів), навчальний процес, аналітичні дослідження тощо.

Висновки

Запропонований у роботі підхід до застосування онтологічної моделі взаємодії між користувачами та ІР у процесі семантичного пошуку забезпечує знаходження ІО із складною структурою, формалізований опис яких міститься у зовнішніх онтологіях. Це дозволяє використовувати фонові знання, що подаються у вигляді тезаурусів, для персоніфікованої фільтрації потрібного користувачам контенту, що особливо актуально із зростанням кількості, обсягу та структурної складності ІР, що оброблюються. Перехід від онтологій до їх окремого випадку – тезаурусів – зменшує обчислювальну складність співставлення ІО. Використання семантичних Wiki-ресурсів як джерела онтологічних знань дозволяє значно точніше описувати ПрО, що цікавлять конкретних користувачів, і внаслідок цього отримувати більш точні результати пошуку за менший час.

Література

1. Baeza-Yates R., A. Raghavan R. Next generation Web search. S. Ceri and M. Brambilla, editors, Search Computing, Springer. 2010. P. 11–23.

2. Рогушина Ю.В. Семантичний пошук у Web на основі онтологій: розробка моделей, засобів і методів. Мелітополь: МДПУ ім. Богдана Хмельницького. 2015. 291 с.
3. Ushold M., Gruninger M. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*. 1996. Vol. 11. N 2.
4. Antoniou G., Van Harmelen F. Web ontology language: Owl. *Handbook on ontologies.* Springer Berlin Heidelberg. 2004. P. 67–92.
5. Рогушина Ю.В. Теоретичні засади застосування онтологій для семантизації ресурсів Web. *Проблеми програмування*. 2018. № 2-3. С. 197–203.
6. Mitchell, T.M. Machine learning. Burr Ridge, IL: McGraw Hill, 45(37). 1997. P. 870–877.
7. Rogushina J.V. Models and Methods of Ontology Use for the Web Semantic search. Proc. of the 11th International Conference of Programming UkrPROG 2018, P. 197–203. http://ceur-ws.org/Vol-2139/197-203.pdf.
8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki.
9. Grimes S. Unstructured Data and the 80 Percent Rule, 2008, Clarabridge, Bridgepoints. http://breakthroughanalysis.com/2008/08/01 /unstructured-data-and-the-80-percent-rule/.
10. Рогушина Ю.В. Використання семантичних властивостей вікі-ресурсів для розширення функціональних можливостей «Великої української енциклопедії». Енциклопедичні видання в сучасному інформаційному просторі: колективна монографія / За ред. Киридон А.М. К.: Державна наукова установа «Енциклопедичне видавництво», 2017. С. 104–115.
11. Гладун А.Я., Рогушина Ю.В. Репозитории онтологии как средство повторного использования знаний для распознавания информационных объектов. *Онтология проектирования*. 2013. № 1 (7).
12. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation, 1999. http://www.w3.org/TR/PR-rdf-syntax
13. OWL 2 Web Ontology Language Document Overview. W3C. 2009. http://www.w3.org/TR/owl2-overview/.

References

1. Baeza-Yates R., A. Raghavan R. (2010) Next generation Web search // S. Ceri and M. Brambilla, editors, Search Computing, Springer, P.11-23.
2. Rogushina J.V. (2015) Semantic retrieval in the Web on base of ontologies: design of methods, means and methods. Melitopol, MDPU. [in Ukrainian].
3. Ushold M., Gruninger M. (1996) Ontologies: Principles, Methods and Applications, // Knowledge Engineering Review, V.11, N 2.
4. Antoniou G., Van Harmelen F. (2004) Web ontology language: Owl. Handbook on ontologies. Springer Berlin Heidelberg, P. 67-92.
5. Rogushina J.V. Theoretical principles of use of ontologies for semantization of the Web resources. Problems in programming. 2018. N 2-3. P. 197–203.
6. Mitchell, T. M. (1997) Machine learning. Burr Ridge, IL: McGraw Hill, 45(37). P. 870–877.
7. Rogushina J.V. Models and Methods of Ontology Use for the Web Semantic search. Proc. of the 11th International Conference of Programming UkrPROG 2018, P.197-203. – http://ceur-ws.org/Vol-2139/197-203.pdf.
8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki.
9. Grimes S. (2008) Unstructured Data and the 80 Percent Rule, , Clarabridge, Bridgepoints. http://breakthroughanalysis.com/2008/08/01 /unstructured-data-and-the-80-percent-rule/.
10. Rogushina J.V. (2017) Use of semantic properties of the Wiki resources for expansion of functional posibilities of “Great Ukrainian Encyclopedia” // Encyclopaedias in the modern information space: collective monograph / Ed. Kyrydon A.M., Kyiv. P. 104–115. [in Ukrainian]
11. Gladun A., Rogushina J. (2013) Ontology repositories as a means of knowledge reuse for recognizing of information objects // Ontology of Design, № 1 (7). [in Russian]
12. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation, 1999. http://www.w3.org/TR/PR-rdf-syntax.
13. OWL 2 Web Ontology Language Document Overview. W3C. 2009. http://www.w3.org/ TR/owl2-overview/.

Одержано 24.10.2019

Про автора:

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 150.
Кількість наукових публікацій в
зарубіжних виданнях – 31.
http://orcid.org/0000-0001-7958-2557.

Місце роботи автора:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.
E-mail: ladamandraka2010@gmail.com