ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2023. No. 2 (24)*

УДК 004.82 DOI: https://doi.org/10.30837/ITSSI.2023.24.062

О. Каратаєв, І. Шубін

ПРОБЛЕМИ ПОВТОРНОГО ВИКОРИСТАННЯ ЗНАНЬ У ПРОЦЕСІ ПРОЄКТУВАННЯ ПРОГРАМНИХ СИСТЕМ

Останнім часом значна увага приділяється створенню баз знань, що містять мільйони фактів про різні об'єкти реального світу. Одним із ключових аспектів управління знаннями є повторне використання знань, які були набуті раніше. **Предмет дослідження** – процеси повторного використання знань і створення програмних систем на основі баз знань. Інтерпретація знань є одним із підходів до повторного їх застосування, що полягає у виведенні нових знань на основі наявних фактів у базі знань. **Метою** дослідження є підвищення ефективності повторного використання знань в програмних системах на основі баз знань способом автоматичного видобування правил. Для досягнення поставленої мети виконано такі **завдання:** досліджено підходи до структурування наявних у базі даних фактів; проведено якісний аналіз можливості застосування автоматичних методів побудови правил і виведення; розглянуто задачу прогнозування зв'язку між парою сутностей, що визначає наявність відношення для фактів; запропоновано узагальнений підхід для подання фактів, що дає змогу використовувати ефективні алгоритми пошуку правил. Для вирішення перелічених завдань застосовано такі **методи**: алгебра скінченних предикатів і предикатних операцій для подання знань, методи прогнозування зв'язку між парою сутностей на основі репрезентативного навчання для автоматичного видобування правил. Здобуто такі **результати**: розглянуто підхід до формування правил, що дає змогу структурувати наявні факти як сукупність двійкових предикатів та застосувати автоматичні методи побудови правил і виведення; зроблено висновок, що обмеженням повторного використання знань є структура бази знань і програмне забезпечення, яке використовується для її підтримки; сформульовано принципи побудови специфічних предикатів-концентраторів для подання атрибутів, що дає змогу узагальнити предикатне подання фактів та застосовувати автоматичні методи видобування правил, що підвищує ефективність повторного використання знань. **Висновки:** застосування методу й механізму ідентифікації на основі предикатних операцій і специфічних предикатів, що автоматично видобувають атрибути з бази знань, разом з оцінкою якості виведених правил дали змогу запропонувати узагальнений підхід для подання фактів і використати ефективні алгоритми пошуку правил, що допоможе підвищити ефективність повторного застосування знань у програмних системах.

Ключові слова: програмна інженерія; бази знань; повторне використання знань; алгебра скінченних предикатів; факти; правила.

Вступ

У сучасному глобальному світі, де ринки повністю інтернаціоналізовані, середовище швидко змінюється, а конкуренція зростає, компанії борються за переваги та їх збереження. Оскільки інтернет зробив інформацію глобальною та доступною, більшість компаній намагаються використовувати цей інтелектуальний капітал, щоб впливати на результати своєї діяльності й отримувати прибуток. В епоху інтелекту люди усвідомлюють, що знання можуть допомогти в прийнятті рішень, уникнути повторення помилок і втрати часу. Крім того, повторне використання знань є менш дорогим, ніж їх набуття. З цією метою з'явилася концепція управління знаннями. Управління знаннями – це процес збирання, розвитку, спільного та ефективного використання організованих знань [1]. На сьогодні створено різні процеси управління знаннями, щоб уможливити застосування цього ресурсу [2–4]. Проте цінні знання все одно втрачаються.

Оскільки повторне використання знань є одним із найважливіших етапів процесу управління знаннями, у літературі можна знайти кілька досліджень про цей етап [5]. Компанії у всьому світі витрачають час, зусилля та гроші, намагаючись залучити найефективніші системи управління знаннями. Розглядаються не лише програмні засоби, а й культурні та стратегічні аспекти [4]. Однак повторного використання знань не відбувається або принаймні бажані результати не досягаються. Зазвичай припускають, що як тільки знання будуть належним чином збережені, їх повторне застосування відбудеться автоматично.

Отже, питання, на яке потрібно відповісти, полягає в тому, як можна створювати бази знань, щоб їх можна було ефективно використовувати. Для цього необхідно проаналізувати системи повторного використання знань, запропонувати підхід щодо створення бази знань та оцінити запропоноване рішення.

© О. Каратаєв, І. Шубін, 2023

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 2 (24)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

Метою дослідження є підвищення ефективності повторного використання знань у програмних системах на основі баз знань способом автоматичного видобування правил.

Аналіз проблеми й наявних методів

Ще на початку 90-х, коли знання стали визнавати важливим активом для підприємств, з'явилася концепція управління знаннями. Водночас виникало багато нових запитань. Що таке знання? Чи всі отримані знання цінні? Чи дані, інформація та знання одне й те саме? Численні дослідження присвячені тому, щоб знайти загальну відповідь на ці запитання. У літературі можна знайти досить схожі визначення. Але в цій статті звернемо увагу на найбільш поширені дефініції. Зокрема в роботі [6] дані названо "неорганізованими та необробленими фактам", тоді як інформація розглядається як "сукупність оброблених даних, що полегшує прийняття рішень". На відміну від попередніх термінів, "знання – це оцінена та організована інформація, яку можна цілеспрямовано використовувати в процесі вирішення проблем". Існує більш глибоке визначення лексеми "знання" – "це здібність, уміння, досвід маніпулювати, перетворювати, створювати дані, інформацію, ідеї для вмілого виконання, прийняття рішень, вирішення проблем" [7]. Отже, дані та інформацію легко зберігати, описувати та маніпулювати ними, тоді як знання потребують повного процесу розуміння, і тому є чимось активним, що може бути створено, трансформовано й актуалізовано.

Генерація знань вимагає не лише часу, але й значного досвіду щодо теми. Крім того, процес генерації здебільшого відбувається в розумі людини, його не можна спостерігати. Тому важко розпізнати, як формуються нові знання, і задокументувати результати. Отже, процес генерації знань можна розуміти як природний процес оброблення даних у інформацію, а потім і в знання.

Намагаючись максимально використати знання компанії, з'явилася дисципліна "Управління знаннями", що розуміється як "процес збирання, розвитку, обміну та ефективного використання організованих знань" [1]. З метою створення цінності для компанії цей системно-орієнтований підхід претендує на те, щоб бути не лише важливим фактором успіху компанії, але й ключем до інновацій. Єдина мета управління знаннями полягає не в тому, щоб стати більш обізнаним, а в тому, щоб краще усвідомити можливі рішення проблем, які вже існують, і те, як до них отримати доступ. Сам процес управління знаннями дослідники поділяють на етапи.

Перший етап – набуття знань. Набуття – це створення нових знань, а також їх пошук. Коли знання вже існують, люди мають шукати їх у певному сховищі; коли йдеться про створення знань, відбувається процес трансформації. Загальновизнано, що знання бувають *явними* (їх можна задокументувати) або *неявними* (для передачі цих знань необхідна особиста взаємодія), і вони мають розглядатися в процесі створення [1, 5]. Трансформація здатна відбуватися за допомогою чотирьох режимів [5]:

– соціалізація: неявні знання перетворюються на нові неявні знання;

– екстерналізація: неявне знання стає явним знанням;

– комбінація: явні знання сприяють новим явним знанням;

– інтерналізація: користувачі інтегрують явні знання у свої процедури, і в такий спосіб вони стають неявними знаннями.

Нові знання можна розрізняти як додаткові або замінні.

Другий етап – документування. На цьому етапі знання фіксуються, документуються, зберігаються та готуються для передачі та подальшого повторного використання.

Третій етап – передача знань. Необхідно проаналізувати складні зв'язки між джерелом знань і метою. Основні види діяльності цього етапу охоплюють визнання потреб у повторному застосуванні, підтримку процесів поширення та сприяння розвитку знань.

Четвертий етап – це повторне використання. Ідеться про перше застосування переданих знань. Люди мають спочатку знати, що вони шукають, бути здатними обрати відповідні знання і, нарешті, застосувати їх.

Останній, п'ятий, етап передбачає всі дії, пов'язані із захистом знань. Про цей етап часто забувають, а деякі автори навіть не розглядають його як частину процесу управління знаннями.

ІТ надають підтримку щодо двох основних підходів до управління знаннями. Вони називаються *кодифікацією* та *персоналізацією* [5]. За допомогою кодифікаційного підходу більш чіткі та структуровані знання кодифікуються й зберігаються в базах знань (електронних сховищах знань); відповідно до підходу персоналізації неявні та неструктуровані знання

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 2 (24)

поширюються завдяки особистому спілкуванню. Реалізація обох підходів є необхідною для управління знаннями.

Однією з найбільш важливих фаз є *повторне використання знань*. Часто, як тільки документ або частину знань знайдено, їх повторне застосування не розглядається. З іншого боку, непросто оцінити вплив повторного використання знань під час певного проєкту, крім того, внесок повторного застосування знань в успіх проєкту неможливо виміряти. Повторне використання знань відбувається як всередині людини, так і зовні. Цікавим фактом є те, що хоча внутрішнє повторне застосування здебільшого успішне, зовнішнє – ні. Внутрішнє повторне використання спирається на найдосконалішу базу знань – мозок людини, де набуття, документування та перетворення знань відбувається автоматично, найбільш інстинктивним способом. Дослідники з'ясували, що людина може швидко знаходити багаторазові предмети у своїй пам'яті, крім того, вона запам'ятовує контекст кожного предмета та розуміє ціле, уможливлюючи більш ефективне повторне використання [5].

Мета статті – розробити підхід до підвищення ефективності повторного застосування знань у програмних системах на основі баз знань способом автоматичного видобування правил. Для досягнення поставленої мети необхідно розглянути підходи до структурування наявних у базі даних фактів; провести якісний аналіз можливості застосування автоматичних методів побудови правил і виведення; проаналізувати задачу прогнозування зв'язку між парою сутностей, що визначає наявність відношення для фактів; запропонувати узагальнений підхід для подання фактів, що дає змогу використовувати ефективні алгоритми пошуку правил.

Вирішення завдання

Під час спроби повторного використання зовнішніх знань виникають певні проблеми. По-перше, людина може не розуміти важливості документування. Документація також часто обмежується формальними знаннями, а аргументація чи контекст, що стоять за прийнятими рішеннями, відсутні. Крім того, важливу роль відіграє відсутність механізмів для визначення, пошуку та відновлення багаторазового використання знань.

Можна зробити висновок, що проблема повторного використання має три основні причини. Перша стосується розуміння контексту та застосованого рішення, друга причина стосується змісту знань, що документуються, і третя – усієї системи бази знань або програмного забезпечення, яке використовується для її підтримки.

Коли говорять про обмін і повторне застосування знань, існує припущення, що знання можна відтворювати та переміщувати з місця на місце; речовина, яку можна отримати від людей-експертів і перенести з однієї комп'ютерної системи чи програми в іншу. Але як можна запропонувати поділитися тим або повторно використовувати те, що не має таких властивостей, як локальність і стійкість? Філософи та інші науковці роками досліджували це питання [4, 8]. Знання розглядається як абстракція, що неможливо записати й ніколи не можна мати в руках. Знання – це те, що спостерігач пояснив би розумному агенту й що дало б змогу агентові раціонально моделювати свою поведінку для досягнення певних цілей, які визначаються відповідно до того, що він дізнався від спостерігача.

Отже, знання можна розглядати як здатність реагувати певним чином, а не як матеріальну субстанцію. Навіть дані, які використовуються для відтворення знань, не можна вважати такими. Правила, символи та фрейми не можуть генерувати розумну поведінку як таку.

Якщо припустити, що все вищезазначене виконано, то проблема IT-підтримки процесу управління знаннями все ще потребує вирішення. Метою поточних баз знань є сприяння обміну знаннями та повторному використанню. Для подання знань необхідно враховувати значну кількість факторів. Термінології, онтології та методи вирішення проблем – це декілька з них [9]. Проблема полягає в тому, щоб мати змогу поділитися знаннями, які містяться в різних базах знань, оскільки всі ці фактори відрізняються від однієї бази до іншої. Несумісність систем і форматів також унеможливлює об'єднання двох баз знань.

Можна виокремити чотири основні причини.

1. Неоднорідність подання. Тут існує кілька підходів, але один формалізм подання не може бути безпосередньо включений в інший. Не існує універсального формалізму подання знань, який би ідеально відповідав усім вимогам, і тому обмін знаннями передбачає переклад змісту однієї бази в іншу.

2. Мовні діалекти. Обмін знаннями між системами може бути дуже складним, якщо знання були

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 2 (24) ISSN 2524-2296 (online)

закодовані різними діалектами. Це може повністю змінити зміст повідомлення або його інтерпретацію.

3. Відсутність визначених правил спілкування. Теоретично окремі системи можуть спілкуватися одна з одною й у такий спосіб мати користь від обміну знаннями, не маючи спільної бази. Але зазвичай це неможливо, оскільки бракує узгодженого протоколу, який би давав змогу системам взаємодіяти між собою та запитувати одна одну.

4. Невідповідності моделі на рівні знань. Навіть у разі усунення всіх перешкод проблеми з термінологією також стануть на заваді для ефективного спілкування між різними базами. Відсутність спільного словникового запасу не дає змогу зіставити знання однієї бази з іншою.

Отже, повторне використання знань або обмін знаннями, що містяться в різних базах, є дуже складним завданням.

Для подання знань у інформаційних системах застосовуються різні формальні мови, зокрема обчислення предикатів. Воно має однозначну формальну семантику та операційну підтримку у вигляді досконалого механізму виведення. Мовою алгебри предикатів будь-яка множина $U = \{a_1, a_2, ..., a_n\}$ може бути записана у вигляді рівняння $x^{a_1} \vee x^{a_2} \vee ... \vee x^{a_n} = 1$, де x – предметна змінна [10]. Сукупність усіх коренів цього рівняння збігається з множиною U. Рівняння є формальним записом твердження $x \in U$. Методи ідентифікації та подання знань з використанням алгебри предикатів здатні уможливити одноманітне подання знань у вигляді рівнянь алгебри предикатів.

Алгебра предикатів описує лише знання про факти. Алгебра операцій над предикатами або алгебра предикатних операцій має формалізувати операції над знаннями, поданими як відношення на деякому предметному просторі. Алгебра предикатів визначає декларативний складник знань, а алгебра предикатних операцій – процедурний складник знань.

Виокремлюють два типи фактів: перший описує зв'язок двох сутностей, до того ж одна з них буде визначатися як суб'єкт, а друга – як об'єкт предикатної дії [11]. У першому випадку факт – це триплет "*суб'єкт – предикат – об'єкт*", у якому предикат є відношенням, а суб'єкт і об'єкт указують на два предмети. Другий вид факту – триплет "*предмет – атрибут – значення*", де предмет – це об'єкт, про який фіксується факт, атрибут – іменована ознака об'єкта, що заздалегідь має певну властивість, а значення – деяке значення цієї ознаки, область визначення якої може бути в деяких випадках відома. Наприклад, це можуть бути факти атрибутів місця й часу здійснення певної дії. Факти другого типу дають змогу розбивати множину сутностей на класи еквівалентності та звузити простір пошуку шляхів висновку. Для отримання таких трійок, визначення сутностей, що їх утворюють, існує чимало підходів [12, 13]. Завдання ставиться як задача прогнозування зв'язку між парою сутностей, що власне визначає: чи пов'язана пара сутностей через відношення.

Останнім часом значна увага приділяється створенню великих баз знань, що містять мільйони фактів про різні сутності у світі. Ці бази знань виявилися неймовірно корисними для інтелектуального пошуку в інтернеті, розуміння запитань, контекстної реклами, аналізу соціальних медіа та біомедицини. Завдяки новим можливостям такі сучасні бази знань часто називають графами знань. Основним завданням у побудові графа знань є розроблення масштабованих методів для автоматизованого навчання нових сутностей, їх властивостей і зв'язків. Правила є явними знаннями (порівняно з нейронною мережею) і можуть надавати людині зрозумілі пояснення результатів навчання (наприклад, прогнозування зв'язків) на їх основі. Отже, важливо автоматично отримувати правила.

Традиційні методи вивчення правил не можуть бути безпосередньо використані для побудови правил з кількох причин. По-перше, ці методи недостатньо масштабовані, щоб обробити величезну кількість даних, яка міститься в звичайних графах знань. Крім того, графи знань явно не виражають негативних прикладів, які є важливими для багатьох інструментів інтелектуального аналізу даних. З іншого боку, у парадигмі репрезентативного навчання статистичні прогностичні моделі широко застосовуються для вивчення фактів, що явно відсутні в базі знань [12, 14]. Основна ідея полягає в кодуванні реляційної інформації за допомогою вбудовування сутностей і предикатів, а потім у використанні їх для керування вилученням правил і в такий спосіб для зменшення простору пошуку.

Отже, будемо розглядати дві категорії фактів: ті, що є зв'язками сутностей, і ті, що стосуються атрибутів сутностей. Необхідно зауважити, що атрибути сутності часто містять незначну кількість сутностей для подання значень атрибутів, які пов'язані з величезною кількістю інших сутностей. Тобто атрибути сутності можуть бути концентратором

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 2 (24)

і є природною основою для розподілу на класи еквівалентності. Наявні підходи до вивчення правил часто не розрізняють атрибути сутності від зв'язків сутності.

Дотримуючись конвенції в поданні знань, будемо позначати факт як $p(e,e')$, де сутність e пов'язана з іншою сутністю e' через скінченний двійковий предикат p. Розглянемо основні положення щодо визначення правил [14, 15]. Правило r має форму:

$$p_1(x_0,x_1)\wedge\cdots\wedge p_n(x_{n-1},x_n)\wedge a_1(x_{l1})\wedge\cdots\wedge a_m(x_{lm})\rightarrow p(x_0,x_n), \quad (1)$$

де кожен $x_i(0\le i\le n)$ є змінною і $0\le l_j\le n$ для кожного $1\le j\le m$.

Кожен $p(u,v)$ називається атомом, а u і v – відповідно аргументами суб'єкта та об'єкта для p, і кожен $a(u)$ є атомом, де a є унарним предикатом. Інтуїтивно правило r стверджує, що якщо $p_1(x_0,x_1),\cdots,p_n(x_{n-1},x_n),a_1(x_{l1}),\cdots,a_m(x_{lm})$ виконується, то $p(x_0,x_n)$ також виконується. Атом $p(x_0,x_n)$ є головою (головним атомом) правила r, і набір атомів $\{p_1(x_0,x_1),\cdots,p_n(x_{n-1},x_n),a_1(x_{l1}),\cdots,a_m(x_{lm})\}$ є тілом правила r. Правило, отримане з r способом вилучення унарних предикатів у тілі, тобто $p_1(x_0,x_1)\wedge\cdots\wedge p_n(x_{n-1},x_n\rightarrow p(x_0,x_n))$, зазвичай називають правилом замкнутого шляху (*closed-path rule*), оскільки послідовність предикатів у тілі правила формує шлях від аргументу суб'єкта до аргументу об'єкта головного предиката [14, 15].

В основних підходах до вивчення правил [15] розглядаються такі положення. Нехай r – правило. Пара сутностей (e,e') задовольняє тіло r, позначене як $body(r)(e,e')$. Якщо існує спосіб замінити змінні в $body(r)$ сутностями в базі знань таким чином: (1) усі атоми в $body(r)$ (після заміни) є фактами в базі знань, і (2) x_0 і x_n замінено на e і e' відповідно. І (e,e') задовольняє голову r, позначену як $p(e,e')$, якщо $p(e,e')$ є також фактом у базі знань. Тоді ступінь підтримки правила r визначається як

$$supp(r)=\#(e,e'):body(r)(e,e')\wedge p(e,e'). \quad (2)$$

Тобто ступінь підтримки правила $supp(r)$ визначається як кількість пар об'єктів, що задовольняють як голову, так і тіло правила r.

Ступінь упевненості правила r $SC(r)$ і покриття голови правила r $HC(r)$ визначаються як форми нормалізації для ступеня підтримки $supp(r)$:

$$SC(r)=\frac{supp(r)}{\#(e,e'):body(r)(e,e')}, \quad (3)$$

$$HC(r)=\frac{supp(r)}{\#(e,e'):p(e,e')}. \quad (4)$$

Отже, $SC(r)$ є нормалізацією $supp(r)$ через кількість пар сутностей, що задовольняють тіло, тоді як $HC(r)$ є нормалізацією $supp(r)$ через кількість пар сутностей, що задовольняють голову.

Метод репрезентативного навчання складається з двох основних кроків: (1) вбудовування сутностей і предикатів бази знань у латентний простір і (2) побудова моделі навчання на основі отриманих убудовувань для передбачення нових фактів [14]. Для побудови вбудовувань можна використовувати такий підхід. Кожна сутність e перетворюється на вектор E і кожен предикат p – у матрицю $\mathbf{P}$. Тоді для кожного заданого факту $p_0(e,e')$ обчислюється наступна функція оцінки:

$$f(e,p_0,e')=E^T\cdot\mathbf{P_0}\cdot E'. \quad (5)$$

Функція оцінки вказує на правдоподібність факту $p_0(e,e')$. Завдання пошуку правил можна звести до завдання пошуку правдоподібних шляхів предикатів. Це досягається за допомогою введення функцій оцінки для всіх можливих шляхів. Щоб визначити таку функцію оцінки, шлях $p=p_1,p_2,\cdots p_n$ розглядається як двійковий предикат між початковою та кінцевою сутністю, а p є правдоподібним, якщо пара сутностей, пов'язаних шляхом, подібна до тих, що асоціюються з цільовим предикатом p_t. Така подібність між p і p_t називається синонімією, коли два предикати пов'язують подібні пари сутностей. Отже, функція оцінки має враховувати довжину шляху та синонімію. Крім цього, важливо оцінити правила за ступенем впевненості $SC(r)$ та покриття голови $HC(r)$.

Як показано в роботі [14], такий підхід дає змогу побудувати алгоритм, здатний визначати правила в базі даних, де факти подано двійковими предикатами, що виражають зв'язки сутностей. Але запропонований підхід має певні обмеження

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 2 (24) ISSN 2524-2296 (online)

щодо фактів другого типу, тобто атрибутів сутностей. Щоб ідентифікувати факти атрибутів $p(e,e')$, потрібно визначити e', який можна використовувати для значень атрибутів, і p, що підходить для подання атрибутів. Тобто потрібно ідентифікувати значення атрибутів e' і предикати атрибутів p.

Важливо зауважити, що значення атрибутів часто пов'язані з величезною кількістю об'єктів і, отже, утворюють своєрідні концентратори. Такі концентратори пропонується ідентифікувати за щільністю їх з'єднання. Визначаємо щільність концентратора $den(e,p)$ у такий спосіб:

$$den_{in}(e,p)=\frac{\#e':p(e',e)}{\#(e',e''):p(e',e'')}, \quad (6)$$

$$den_{out}(e,p)=\frac{\#e':p(e,e')}{\#(e',e''):p(e',e'')}, \quad (7)$$

де $den_{in}(e,p)$ і $den_{out}(e,p)$ – внутрішня та зовнішня щільність відповідно; $\#e'$ – загальна кількість випадків e (як суб'єкта чи об'єкта) у факті $p(e,e')$ або $p(e',e)$.

Необхідно зазначити, що концентратор e з високою щільністю не обов'язково підходить для значення атрибута. Розглянемо екстремальний випадок, коли всі інші сутності e' пов'язані з e через предикат p, тоді e не є значенням атрибута, яке може відрізнити e' від інших сутностей. З інформаційної теорії такий атрибут не забезпечує жодного інформаційного приросту, оскільки він не виражає жодної відмінної особливості асоційованих сутностей. Отже, нам потрібно обрати концентратори, які мають високу щільність і відмінні особливості. У роботі [15] пропонується обчислити ентропію концентраторів, а потім, разом з оцінкою ступеня дисбалансу, визначити ймовірність факту з концентратором e та предикатом p, що є атрибутом сутності. Тобто $p(e',e)$ виражає факт, що e' є сутністю, а e – значенням атрибута. Це дає змогу ідентифікувати факти атрибутів у базі знань.

Загалом атрибути можна розглядати як унарні предикати, тоді як інші відношення є бінарними. Однак такий унарний факт не може бути безпосередньо оброблений такими алгоритмами, як запропоновано в роботі [14], оскільки вони приймають лише двійкові предикати (тобто трійки) як вхідні дані. Формально кожен факт атрибута виду $p(e,e')$ (або $p(e',e)$) з e', що є значенням атрибута, може бути перетворений на факт $P:p\_E:e'(e,e)$ (відповідно, $E:e'\_P:p(e,e)$), де $P:p\_E:e'$ (відповідно, $E:e'\_P:p$) є новим предикатом. Ідея такого перетворення полягає в тому, щоб подати атрибути як самоцикли в графі знань (подані бази знань), щоб метод навчання правил на основі шляхів можна було легко адаптувати, тоді як атоми атрибутів (як самоцикли) можна буде зручно ідентифікувати у визначених правилах.

Є низка робіт, у яких описано застосування визначених на основі навчання правил до відомих фактів для виведення нових фактів у базі знань. Більшість цих праць присвячена тому, як можна використовувати правила ітераційно [16]. Існують інші дослідження, що розглядають одноразове застосування правил невизначеності в спільноті графів знань [12, 15]. Деякі підходи доповнюються ступенем достовірності, і модуль висновку враховує ці ступені, щоб вивести новий факт (який також доповнюється ступенем достовірності). Щоб контролювати якість і кількість нових висновків, ці системи встановлюють максимальний поріг кількості нових фактів і мінімальний поріг достовірності нових фактів. Проблемою такого підходу є те, що такі методи не враховують вимоги щодо кожного нового факту – вони лише зосереджуються на процесі, результатом якого є нові факти.

У цьому дослідженні, на відміну від наявних підходів до моделювання атрибутів, що використовують атрибути як вхідні дані, запропоновано видобувати атрибутивні правила з бази знань без явного набору атрибутів. Це досягається способом використання двійкового предиката атрибута, який можна легко вбудувати в правила закритого шляху, і механізму ідентифікації, що автоматично видобуває такі атрибути з бази знань. Отже, база знань доповнюється особливими предикатами – концентраторами – та відповідними фактами і може бути оброблена методами репрезентативного навчання для видобування правил, що в сукупності з оцінкою якості виведених правил дасть змогу підвищити ефективність використання баз знань.

Приклад

Розглянемо на прикладі, як можна застосувати запропонований підхід. Нехай $K=(E,F)$ – база

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 2 (24)

знань, де $E=\{e_1,...,e_n\}$ – множина всіх сутностей; $F=\{P_1,\ldots,P_m\}$ – множина всіх предикатів. Можемо подати вхід бази знань як набір A матриць суміжності, де кожна $n\times n$ матриця $A(P_k)$ відповідає предикату P_k у базі знань $(1\le k\le m)$. Зокрема $A[i,j]$ дорівнює 1, якщо факт $P_k(e_i,e_j)$ міститься в K; та 0 в іншому випадку. Отже, $A(P_k)$ є матрицею двійкових значень.

Проілюструємо це. Розглянемо правило $r: P_1(x,z)\wedge P_2(z,y)\to P(x,y)$. Щоб обчислити SC і HC для правила r у базі знань K, нам потрібно обчислити кількість пар сутностей, що задовольняють голову правила r (тобто $\#(e,e'): p(e,e')$), які задовольняють тіло правила r (тобто $\#(e,e'): body(r)(e,e')$), і які задовольняють як голову, так і тіло (тобто $supp(r)$) відповідно. Пари (e_i,e_j), що задовольняють голову, можна безпосередньо прочитати з матриці $A(P)$, тобто де $A(P)[i,j]=1$. Для пар (e_i,e_j), які задовольняють тіло, необхідно зауважити, що вони з'єднані шляхом $p=P_1,P_2$ і можуть бути отримані з добутку $A(P_1)\cdot A(P_2)$ двох матриць $A(P_1)$ і $A(P_2)$. Зокрема елемент $[i,j]$ матриці $A(P_1)\cdot A(P_2)$ подає кількість шляхів правила, які починаються від e_i, проходять через P_1 до іншої сутності та врешті йдуть до e_j через P_2. Але в цьому разі матриця $A(P_1)\cdot A(P_2)$ може мати недвійкові значення. Нехай $A(P_1,P_2)=binary(A(P_1)\cdot A(P_2))$, де $binary(M)$ – це матриця, отримана з матриці M установкою всіх ненульових елементів на 1. Отже, пари (e_i,e_j), що задовольняють тіло, можна побачити з $A(P_1,P_2)$, тобто де $A(P_1,P_2)[i,j]=1$. Покажемо на прикладі.

Нехай $E=\{e_1,e_2,e_3\}$ та $F=\{P_1(e_1,e_2),P_1(e_2,e_1),P_1(e_1,e_3),P_2(e_2,e_3),P_2(e_2,e_1),P_2(e_3,e_3),P(e_1,e_3)\}$. Тоді матриці суміжності для предикатів P_1, P_2 і P записуються таким чином:

$$A(P_1)=\begin{bmatrix}0&1&1\\1&0&0\\0&0&0\end{bmatrix},\ A(P_2)=\begin{bmatrix}0&0&0\\1&0&1\\0&0&1\end{bmatrix},\ A(P)=\begin{bmatrix}0&0&1\\0&0&0\\0&0&0\end{bmatrix},$$

тоді $A(P_1)\cdot A(P_2)=\begin{bmatrix}1&0&2\\0&0&0\\0&0&0\end{bmatrix}\xrightarrow{binary()}A(P_1,P_2)=\begin{bmatrix}1&0&1\\0&0&0\\0&0&0\end{bmatrix}$.

Матриця $A(P_1,P_2)$ показує пари, з'єднані шляхом $p=P_1,P_2$, тобто $\{(e_1,e_2),(e_1,e_3)\}$, і це саме ті пари, які задовольняють тіло правила r. І з огляду на $A(P)$ є тільки одна пара (e_1,e_3), що задовольняє голову r. Тоді можемо легко отримати, що $supp(r)=1$, $HC(r)=1$ і $SC(r)=0,5$.

Необхідно зауважити, що хоча цей приклад базується на правилах довжини, цей розрахунок легко узагальнити до правил будь-якої довжини.

У табл. 1 наведено приклади фактів, що містяться в базі знань як множина трійок у форматі RDF.

Таблиця 1. *Фрагмент бази знань*

Сутність e	**Предикат** p	**Сутність** e'
<Princess_Marie_Louise_of_Parma>	<isMarriedTo>	<Ferdinand_I_of_Bulgaria>
<Sebastian_Faulks>	<created>	<Birdsong_(novel)>
<Sebastian_Faulks>	<created>	<Charlotte_Gray_(film)>
<Sebastian_Faulks>	<created>	<A_Fool's_Alphabet>
<Brian_K._Vaughan>	<created>	<Runaways_(comics)>
<Brian_K._Vaughan>	<created>	<Pride_of_Baghdad>
<August_Aleksander_Czartoryski>	<isMarriedTo>	<Maria_Zofia_Sieniawska>
<Jerzy_Detloff_Fleming>	<hasChild>	<Izabela_Czartoryska>
<Antonina_Czartoryska>	<hasChild>	<Izabela_Czartoryska>
<Kenneth_Waltz>	<influences>	<Stephen_Walt>
<Marie_of_Brabant>	<isMarriedTo>	<_Queen_of_France,Philip_III_of_France>
<Cynthia_Lennon>	<isMarriedTo>	<John_Lennon>
<Ferdinand_I_of_Aragon>	<hasChild>	<Alfonso_V_of_Aragon>
<Azerbaijan>	<hasCapital>	<Baku>
<The_Rolling_Stones>	<created>	<Dirty_Work_(album)>
…	…	…

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 2 (24)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

Спочатку оцінюємо правила-кандидати на основі вибірки K' і обираємо правила, для яких $supp(r) \geq 1$. Ці правила все ще можуть містити велику кількість надлишкових і низькоякісних правил, тому необхідно зробити подальший відбір на основі двох показників – HC та SC. Для обчислення значень HC та SC потрібно перевірити, чи задовільнять усі атоми тіла всіх правил-кандидатів на останній фазі. Іншими словами, потрібно визначити всі релевантні атоми, які можуть запустити правило кандидата. У цьому разі маємо базу знань або вибірку з неї KG і правило замкнутого шляху як вхідні дані.

Наприклад, для наступного правила:

$$hasChild(x,t) \wedge hasChild-1(t,z) \wedge isCitizenOf(z,y) \rightarrow isCitizenOf(x,y).$$

Отримані наведеним методом оцінки $SC = 0,89$ та $HC = 0,13$ показують, що ступінь впевненості високий, тобто ступінь достовірності факту забезпечується правилами з бази знань.

Висновки

Графи знань виявилися корисними для побудови взаємозв'язків між джерелами інформації способом з'єднання об'єктів різних типів, вони надають формальний базис створення та ефективного використання баз знань у процесі проєктування інтелектуальних програмних систем. У штучному інтелекті граф знань подається зазвичай набором трійок: "*суб'єкт – предикат – об'єкт*" або "*об'єкт – атрибут – значення*". Для отримання таких трійок, визначення сутностей, що їх утворюють, існує багато інструментів, але, як показав проведений аналіз, вони мають обмеження щодо масштабування, репрезентативного навчання та інтерпретації правил і механізмів їх виведення. Розглянуто задачу прогнозування зв'язку між парою сутностей, що визначає наявність відношення для обох типів фактів. Запропоновано узагальнений підхід для подання фактів, що дає змогу використовувати ефективні алгоритми пошуку правил.

Методи ідентифікації та подання знань із застосуванням алгебри предикатів можуть допомогти одноманітно подати знання у вигляді рівнянь алгебри предикатів. Будь-яке рівняння алгебри предикатів можна реалізувати програмно як відповідну йому агентно-предикатну структуру. Отже, подальші дослідження варто присвятити створенню агентно-предикатної структури розв'язання рівнянь і пошуку правил.

Список літератури

1. Koenig M. E. D. What is KM? Knowledge management explained. KMWorld Magazine. URL: https://www.kmworld.com/Articles/Editorial/What-Is-.../What-is-KM-Knowledge-Management-Explained-82405.aspx

2. Ma L., Yu H., Wang Y., Chen G. (2012). The Knowledge Representation and Semantic Reasoning Realization of Productivity Grade Based on Ontology and SWRL. *Computer and Computing Technologies in Agriculture V. CCTA 2011. IFIP Advances in Information and Communication Technology*, vol 368. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27281-3_44

3. D. Moshood T., Ebun Rotimi F., O. B. Rotimi J. (2022). An Integrated Paradigm for Managing Efficient Knowledge Transfer: Towards a More Comprehensive Philosophy of Transferring Knowledge in the Construction Industry. *Construction Economics and Building*, 22(3). https://doi.org/10.5130/AJCEB.v22i3.8050

4. Chen Z., Wang Y., et al. Knowledge graph completion: A review. *IEEE Access* Vol. 8. 2020. P. 192435–192456. URL: https://ieeexplore.ieee.org/iel7/6287639/8948470/09220143.pdf

5. Schacht S., Maedche A. A Methodology for Systematic Project Knowledge Reuse. *Innovations in Knowledge Management. Razmerita, L.* (Eds.), Springer (Berlin). 2016. P. 19-44. DOI:10.1007/978-3-662-47827-1_2

6. Ameri F., Dutta D. Product lifecycle management: closing the knowledge loops. *Computer-Aided Design and Applications* 2 (5). 2005. P. 577–590. DOI:10.1080/16864360.2005.10738322

7. Shubin I. Development of conjunctive decomposition tools. *CEUR Workshop Proceedings (COLINS 2021). Vol. I*, 2021. P. 890–900. URL: https://ceur-ws.org/Vol-2870

8. Milton N. R. Knowledge acquisition in practice: a step-by-step guide. Springer Science & Business Media. 2007. URL: https://www.researchgate.net/publication/234798481_Knowledge_Acquisition_in_Practice_A_Step-by-step_Guide

9. Saavedra C., Villodres T., Lindemann U. Review and Classification of Knowledge in Engineering Design. *Technische Universit chen*. 2017. DOI:10.1007/978-981-10-3521-0_53

10. Martin Ph., Bénard J. Top-level Ideas about Importing. *Translating and Exporting Knowledge via an Ontology of Representation Languages. In Proceedings of the 12th International Conference on Semantic Systems* (2016). Association for Computing Machinery, New York, NY, USA. P. 89–92. DOI: https://doi.org/10.1145/2993318.2993344

11. Khudhair A. T. The intelligence theory mathematical apparatus formal base. *Advanced Information Systems*, 1 (1), 2017. P. 38–43. DOI: https://doi.org/10.20998/2522-9052.2017.1.07

12. Sharonova N. et al. Issues of Fact-based Information Analysis. *International Conference on Computational Linguistics and Intelligent Systems*. 2018. URL: https://ceur-ws.org/Vol-2136/10000011.pdf

13. Omran P. G., Wang K., Wang Z. An Embedding-based Approach to Rule Learning in Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering.* 2021. vol. 33, no. 4. P. 1348-1359. URL: https://doi.org/10.1109/TKDE.2019.2941685

14. Pellissier-Tanon T., Weikum G., Suchanek F. YAGO 4: A Reasonable Knowledge Base. *ESWC.* 2020. URL: https://suchanek.name/work/publications/eswc-2020-yago.pdf

15. Omran P. G., Wang Z., Wang K. Learning Rules With Attributes and Relations in Knowledge Graphs. *AAAI Spring Symposium*: MAKE. 2022. URL: https://ceur-ws.org/Vol-3121/paper10.pdf

16. Omran P. G., Wang Z., Wang K. Scalable rule learning via learning representation. *IJCA.* 2018. URL: https://www.ijcai.org/proceedings/2018/0297.pdf

17. Svato M., Schockaert S., Davis J. STRiKE: Rule-Driven Relational Learning Using Stratified k-Entailment. *ECA.* 2020. URL: https://orca.cardiff.ac.uk/130911/http:/orca.cf.ac.uk/130911/1/ECAI2020_STRiKE.pdf

18. Малєєва Ю. А., Персиянова Е. Ю., Косенко В. В. Информационное и программное обеспечение менеджера по персоналу IT-компании. *Сучасний стан наукових досліджень та технологій у промисловості.* 2018. № 1 (3). С. 22–32. DOI: https://doi.Org/10.30837/2522-9818.2018.3.022

19. Barkovska O. Research into Speech-to-text Transformation Module in the Proposed Model of a Speaker's Automatic Speech Annotation. *Сучасний стан наукових досліджень та технологій у промисловості.* 2022. № 4 (22). С. 5–13. DOI: https://doi.org/10.30837/ITSSI.2022.22.005

References

1. Koenig, M. E. D. (2012), "What is KM? Knowledge management explained. KMWorld Magazine", available at: https://www.kmworld.com/Articles/Editorial/What-Is-.../What-is-KM-Knowledge-Management-Explained-82405.aspx

2. Ma, L., Yu, H., Wang, Y., Chen, G. (2012), "The Knowledge Representation and Semantic Reasoning Realization of Productivity Grade Based on Ontology and SWRL", *Computer and Computing Technologies in Agriculture V. CCTA 2011. IFIP Advances in Information and Communication Technology*, Vol. 368. P. 381–389. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-27281-3_44

3. D. Moshood, T., Ebun Rotimi, F., O. B. Rotimi, J. (2022), "An Integrated Paradigm for Managing Efficient Knowledge Transfer: Towards a More Comprehensive Philosophy of Transferring Knowledge in the Construction Industry", *Construction Economics and Building*, 22(3). DOI: https://doi.org/10.5130/AJCEB.v22i3.8050

4. Chen, Z., Wang, Y., et al. (2020), "Knowledge graph completion: A review", *IEEE Access,* Vol. 8. P. 192435–192456, available at: https://ieeexplore.ieee.org/iel7/6287639/8948470/09220143.pdf

5. Schacht, S., Maedche, A. A. (2016), "Methodology for Systematic Project Knowledge Reuse", *Innovations in Knowledge Management,* Springer (Berlin). P. 19–44. DOI:10.1007/978-3-662-47827-1_2

6. Ameri, F., Dutta, D. (2005), "Product lifecycle management: closing the knowledge loops", *Computer-Aided Design and Applications,* 2 (5). P. 577–590. DOI:10.1080/16864360.2005.10738322

7. Shubin, I. (2021), "Development of conjunctive decomposition tools", *CEUR Workshop Proceedings*, P. 890–900. URL: https://ceur-ws.org/Vol-2870/

8. Milton, N. R. (2007), "Knowledge acquisition in practice: a step-by-step guide", *Springer Science & Business Media,* available at: https://www.researchgate.net/publication/234798481_Knowledge_Acquisition_in_Practice_A_Step-by-step_Guide

9. Saavedra, C., Villodres, T., Lindemann, U. (2017), "Review and Classification of Knowledge in Engineering Design", *Technische Univ chen*, DOI:10.1007/978-981-10-3521-0_53

10. Martin, Ph., Bénard, J. (2016), "Top-level Ideas about Importing", *Translating and Exporting Knowledge via an Ontology of Representation Languages. In Proceedings of the 12th International Conference on Semantic Systems,* Association for Computing Machinery, New York, NY, USA, P. 89–92. DOI: https://doi.org/10.1145/2993318.2993344

11. Khudhair, A. T. (2017), "The intelligence theory mathematical apparatus formal base", *Advanced Information Systems,* 1(1), P. 38–43. DOI: https://doi.org/10.20998/2522-9052.2017.1.07

12. Sharonova, N. et al. (2018), "Issues of Fact-based Information Analysis", *International Conference on Computational Linguistics and Intelligent Systems*, available at: https://www.semanticscholar.org/paper/Issues-of-Fact-based-Information-Analysis-Sharonova-Doroshenko/f923b77b8561736202388db853e51df9bb7b9301

13. Omran, P. G., Wang, K., Wang, Z. (2021), "An Embedding-based Approach to Rule Learning in Knowledge Graphs", *IEEE Transactions on Knowledge and Data Engineering,* P. 1348–1359, available at: https://ieeexplore.ieee.org/document/8839576

14. Pellissier-Tanon, T., Weikum, G., Suchanek, F. (2020), "YAGO 4: A Reasonable Knowledge Base", *ESWC*, available at: https://suchanek.name/work/publications/eswc-2020-yago.pdf

Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 2 (24)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

15. Omran, P. G., Wang, Z., Wang, K. (2022), "Learning Rules with Attributes and Relations in Knowledge Graphs", *AAAI Spring Symposium:* MAKE, available at: https://ceur-ws.org/Vol-3121/paper10.pdf

16. Omran, P. G., Wang, Z., Wang, K. (2018), "Scalable rule learning via learning representation", *IJCAI*, available at: https://www.ijcai.org/proceedings/2018/0297.pdf

17. Svato, M., Schockaert, S., Davis, J. (2020), "STRiKE: Rule-Driven Relational Learning Using Stratified k-Entailment", *ECAI*, available at: https://orca.cardiff.ac.uk/130911/http:/orca.cf.ac.uk/130911/1/ECAI2020_STRiKE.pdf

18. Malyeyeva, O., Persiyanova, E., Kosenko, V. (2018), "Information and software for the personnel manager of an IT company", *Innovative Technologies and Scientific Solutions for Industries,* No. 1 (3). P. 22–32. DOI: https://doi.org/10.30837/2522-9818.2018.3.022

19. Barkovska, O. (2022), "Research into Speech-to-text Transformation Module in the Proposed Model of a Speaker's Automatic Speech Annotation", *Innovative Technologies and Scientific Solutions for Industries,* No. 4 (22). P. 5–13. DOI: https://doi.org/10.30837/ITSSI.2022.22.005

Received 16.05.2023

Відомості про авторів / About the Authors

Каратаєв Олександр Анатолійович – Харківський національний університет радіоелектроніки, аспірант кафедри програмної інженерії, Харків, Україна; e-mail: tosanik@gmail.com; ORCID ID: https://orcid.org/ 0009-0007-6654-1327

Шубін Ігор Юрійович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри програмної інженерії, Харків, Україна; e-mail: igor.shubin@nure.ua; ORCID ID: https://orcid.org/ 0000-0002-1073-023X

Karataiev Oleksandr – Kharkiv National University of Radio Electronics, Postgraduate, Kharkiv, Ukraine.

Shubin Ihor – PhD (Engineering Sciences), Kharkiv National University of Radio Electronics, Professor at the Software Department, Kharkiv, Ukraine.

REUSE OF INFORMATION BASED ON THE INTERPRETATION OF KNOWLEDGE

Recently, much attention has been paid to the creation of knowledge bases that contain millions of facts about various objects of the real world. One of the key aspects of knowledge management is the reuse of previously acquired knowledge. The **subject of research** is the processes of knowledge reuse and the creation of software systems based on knowledge bases. Knowledge interpretation is one approach to knowledge reuse, which consists in deriving new knowledge based on existing facts in the knowledge base. **The purpose of the study** is to increase the efficiency of knowledge reuse in software systems based on knowledge bases due to automatic rule extraction. To achieve the goal, the following **tasks** were solved: approaches to structuring the facts available in the database were considered, a qualitative analysis of the possibility of applying automatic methods of rule construction and derivation was carried out. The task of predicting the connection between a pair of entities, which determines the presence of a relationship for facts, is considered. A generalized approach to the presentation of facts is proposed, which allows the use of efficient rule-finding algorithms. The following **methods** are used to solve the given problem: the algebra of finite predicates and predicate operations for knowledge representation, methods for predicting the connection between a pair of entities based on representative learning for automatically obtaining rules. The following **results** were obtained: an approach to rule formation was considered, which allows structuring existing facts as a set of binary predicates and applying automatic methods of rule construction and derivation. It is concluded that the limitation of knowledge reuse is the structure of the knowledge base and the software used to support it. The article formulates the principles of building specific concentrator predicates for the representation of attributes, which allows generalizing the predicate representation of facts and applying automatic methods of rule extraction, which increases the efficiency of knowledge reuse. **Conclusions**: the application of the method and mechanism of identification based on predicate operations and specific predicates, which automatically extracts attributes from the knowledge base, together with the quality assessment of the derived rules, made it possible to propose a generalized approach for presenting facts and use effective rule search algorithms, which allows to increase the efficiency of reuse knowledge in software systems.

Keywords: software engineering; knowledge bases; reuse of knowledge; algebra of finite predicates; facts; rules.

Бібліографічні описи / Bibliographic descriptions

Каратаєв О. А., Шубін І. Ю. Проблеми повторного використання знань у процесі проєктування програмних систем. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 2 (24). С. 62–71. DOI: https://doi.org/10.30837/ITSSI.2023.24.062

Karataiev, O., Shubin, I. (2023), "Reuse of information based on the interpretation of knowledge", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (24), P. 62–71. DOI: https://doi.org/10.30837/ITSSI.2023.24.062