Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження методів автоматизованої побудови баз знань в інформаційно-довідкових системах_____
(тема)

Виконав:
студент _2_ курсу, групи _ІУСТм-20-1____
__Кохан Д. А._____
(прізвище, ініціали)

Спеціальність ____122 Комп'ютерні_____
_____науки_____
(код і повна назва спеціальності)
Тип програми ____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Освітня програма _Інформаційні управляючі системи та технології_____
(повна назва освітньої програми)

Керівник ___проф. Чалий С. Ф._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ ___Петров К.Е.__
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____

Кафедра_____Інформаційних управляючих систем_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____122 Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма___Інформаційні управляючі системи та технології_____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Кохану Даніїлу Андрійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи ____Дослідження методів автоматизованої побудови баз знань в інформаційно-довідкових системах_____
_____
затверджена наказом університету від _05_ ____листопада____2021 р. № __1645 Cm__

2. Термін подання студентом роботи до екзаменаційної комісії _09_ _грудня__ 2021 р.

3. Вихідні дані до роботи _Науково-технічні публікації та інтернет джерела з тематики _____
атестаційної роботи_____
_____
_____
_____
_____
_____
_____
_____

4. Перелік питань, що потрібно опрацювати в роботі _вступ, аналіз існуючих методів автоматизованої побудови баз знань для інформаційно-довідкових систем, аналіз структури та властивостей інформаційно-довідкових систем, представлення знань в інформаційно-довідкових системах, дослідження методів автоматизованої побудови баз знань, постановка задачі, удосконалення методу автоматизованої побудови баз знань для інформаційних довідкових систем, методи автоматизованої побудови баз знань в інформаційно-довідкових системах, удосконалений метод побудови зважених правил бз в ідс, технологія вирішення задачі, практичне використання отриманих результатів, розробка програмного засобу автоматизованої побудови зважених правил для баз знань в інформаційно-довідкових системах ,експериментальна перевірка методу, висновки._____
_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та Інтернет-джерел	08.11.21	
2	Постановка задачі	09.11.2021-11.11.2021	
3	Обробка матеріалу	12.11.2021-14.11.2021	
4	Дослідження сучасної структури роботи Інформаційно-довідкових	15.11.2021-20.11.2021	
5	Дослідження методів побудови баз знань в інформаційно-довідкових системах	21.11.2021-25.11.2021	
6	Дослідження особливостей методів реалізації формування правил для баз знань інформаційно-довідкових систем	26.11.2021-28.11.2021	
7	Апробація результатів дослідження на прикладі	29.11.2021-30.11.2021	
8	Написання пояснювальної записки	01.12.2021-04.12.2021	
9	Підготовка презентації	05.12.2021-06.12.2021	
10	Перевірка на плагіат	06.12.2021-07.12.2021	
11	Нормоконтроль	07.12.2021-08.12.2021	
12	Захист	10.12.21	

Дата видачі завдання __08__ _____11_____2021 р.

Студент _____
(підпис)

Керівник роботи _____ _____проф. Чалий С. Ф.____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить: 71 с., 4 розділи, 27 рис., 9 табл., 32 джерел.

БАЗА ЗНАНЬ, ІНФОРМАЦІЙНО-ДОВІДКОВІ СИСТЕМИ, МЕТОД АВТОМАТИЗОВАНОЇ ПОБУДОВИ БАЗ ЗНАНЬ, МЕТОД АВТОМАТИЗОВАНОЇ ПОБУДОВИ ПРАВИЛ ДЛЯ БАЗ ЗНАНЬ, МЕТОД ВИЛУЧЕННЯ ПРАВИЛ ІЗ ТЕКСТУ

У роботі проведено огляд методів автоматизованої побудови баз знань в інформаційно-довідкових системах. Проаналізовано існуючі методи автоматизованої побудови баз знань. На підставі проведеного аналізу запропоновано удосконалення методу автоматизованої побудови баз знань шляхом автоматизованого формування правил в інформаційно-довідкових системах.

В ході дослідження отримані такі результати: визначені компоненти інформаційно-довідкових систем; визначено існуючі моделі представлення баз знань; визначені існуючі методи автоматизованої побудови баз знань в інформаційно-довідкових системах; виконаний опис удосконаленого методу автоматизованої побудови баз знань шляхом автоматизованого формування правил; проведено експериментальну перевірку удосконаленого методу.

ABSTRACT

Explanatory Note to qualifying work contains 71 pages, 4 sections, 27 pictures, 9 tables, 32 sources.

AUTOMATED KNOWLRDGE BASE CONSTRUCTION METHOD, AUTOMATED RULES CONSTRUCTION METHOD FOR KNOWLRDGE BASE, INFORMATION-SHARING SYSTEM, KNOWLEDGE BASE, METHOD OF RULES EXSTRACTION FROM TEXT

The paper reviews the methods of automated knowledge bases construction in information-sharing systems. The existing methods of automated knowledge bases construction are analyzed. Based on the conducted analysis, the method improvement for automated knowledge bases construction by the automated rules formation in information-shared systems is offered.

The following results were obtained during the research: components of information-sharing systems were identified; the existing models of knowledge base presentation are determined; the existing methods of automated knowledge bases construction in information-sharing systems are defined; a description of the improved method of automated construction of knowledge bases by automated rulemaking; an experimental test of the improved method was performed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

БЗ – база знань;

ІДС – інформаційна довідкова система;

НМ – натуральна мова;

РП – робоча пам’ять;

СВЗ – система вилучення знань;

СУБД – система управління базами даних;

ТОВ – товариство з обмеженою відповідальністю;

HTML – hyper text markup language;

KG – knowledge graph;

NELL – never-Ending learning language;

SQL – structured query language;

UDF – user defined function.

ВСТУП

Стрімкий ріст обсягу інформації в інтернеті призводить до того, що процеси пошуку ускладняються, а їх ефективність знижується. Через неконтрольоване поповнення інтернет-джерел збільшується зашумленість інформації та зменшується її надійність. Більша частина інформації зберігається у неструктурованому виді. Вона може бути представлена у відео-форматі, у виді людської мови. Людська мова неструктурована і має багато варіацій. Також у наш час стрімкого розвитку набула розробка проектів з автоматизації різних процесів. Деякі такі проекти вимагають обробки текстів, що для автоматизованого режиму є складною задачею. Ці проблеми можна вирішити побудовою баз знань.

Побудова баз знань дозволяє структурувати дані та надає механізми для отримання інформації з отриманої структури. База знань – це технологія, що дозволяє зберігати інформацію за структурою, що є оптимальною для машинної обробки. У базі знань інформація зберігається у виді сутностей та зв’язків між ними, що дозволяє зменшити обсяг, що займає інформація на диску у порівнянні зі звичайним текстом. Усе це дозволить удосконалити процеси обробки інформації.

База знань є частиною інформаційно-довідкової системи. Такі системи дозволяють швидко отримувати точну інформацію про предмет запиту та сусідні теми до предмету запиту. Інформаційно-довідкові системи мають покривати великий обсяг інформації. Для цього необхідно працювати з великою базою знань. Але мануальний процес побудови такої бази знань є неможливим через великий обсяг даних. Тому доцільно розробляти, удосконалювати та використовувати автоматизовані методи побудови баз знань для інформаційно-довідкових систем. Автоматизація процесу побудови бази знань дозволяє підвищити якість бази знань та швидкість її створення.

На сьогоднішній день існують такі направлення, що націлені на використання методів автоматизованої побудови баз знань, але вони все ще не задовольняють сучасним потребам.

Розглянутий у даний роботі метод використовує готову базу правил для виведення. Це обмежує кількість фактів, що виводяться правилами з вибірки. Запропоноване удосконалення розглянутого методу змінює етап формування правил так, що в автоматизованому режимі виконується побудова правил з тексту, що аналізується.

Об’єктом дослідження є процес побудови баз знань для інформаційних довідкових систем.

Предметом дослідження є метод автоматизованої побудови баз знань для інформаційних довідкових систем.

Мета дослідження: розробка удосконаленого методу автоматизованої побудови баз знань в інформаційно-довідкових системах [1].

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ АВТОМАТИЗОВАНОЇ ПОБУДОВИ БАЗ ЗНАНЬ ДЛЯ ІНФОРМАЦІЙНО-ДОВІДКОВИХ СИСТЕМ

1.1 Аналіз структури та властивостей інформаційно-довідкових систем

Комп'ютерні технології проникають у всі сфери життя. Інформаційно-довідкові системи можуть використовуватися не тільки для отримання довідкових даних, але і при вивченні певних загальноосвітніх чи вузькоспеціальних предметів. Це дозволяє значно полегшити процес навчання школярів, абітурієнтів, студентів, а також надає додаткові можливості самоосвіти. Також подібні системи можуть використовуватися викладачами при розробці методичних матеріалів для лабораторних чи самостійних робіт.

Інформаційно-довідкова система (ІДС) – це програмний засіб, призначений для збору, обробки, зберігання та передачі інформації користувачу різноманітної інформації довідкового змісту. ІДС використовують бази знань, тому інформація завжди структурована. Завдяки цьому, ІДС може ефективно використовуватися для пошуку інформації.

ІДС вирішують усі поставлені завдання щодо забезпечення споживачів нормативною інформацією. ІДС мають такі переваги:

– компактне зберігання великого обсягу інформації;

– структурне відображення інформації, що зберігається;

– швидкий пошук необхідних документів у великих масивах даних.

На рисунку 1.1 представлена узагальнена схема ІДС.

База знань використовується для пошуку та управління знаннями. Над інформацією з неї проводиться логічний висновок.

Механізм висновку автоматизовано виконує висновок зі знань. Він може синтезувати нові зв’язки, які явно не відображені в базі знань [2].

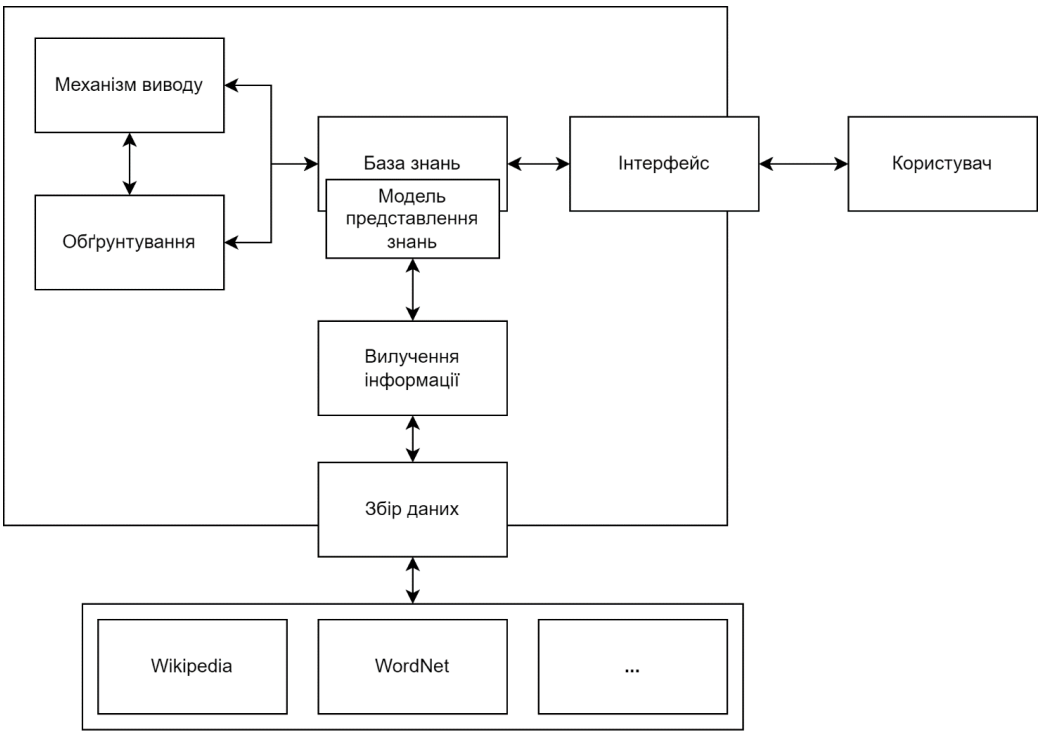


Рисунок 1.1 – Схема інформаційної довідкової системи

Для автоматизованого заселення бази знань виконується обробка даних зі зовнішніх джерел. Для цього вхідні речення розбиваються на елементи. між якими встановлюється зв’язок. Далі елементи зв’язують з існуючими знаннями [3].

БЗ є важливим елементом інтелектуальної інформаційної системи (ІІС), до класу яких відносяться інформаційні довідкові системи (ІДС).

ІІС – це автоматизована інформаційна система, що працює зі знаннями. ІІС представлена комплексом програмного, лінгвістичного, логіко-математичного забезпечення.

Процеси побудови БЗ, що описуються та виконуються зараз є автоматизованими. Для утворення якісної БЗ необхідно обробляти велику кількість темних даних, які мають повторюватися у різних джерелах, щоб підтвердити факти. Без інструментів автоматизації майже неможливо створити якісну БЗ великого масштабу.

Програмна обробка даних найбільш ефективна, коли використовується зі структурованими даними. Методи обробки структурованої інформації зіткаються з меншою кількістю шуму, ніж неструктурованої.

Процес автоматизованої побудови баз знань є частиною ІДС систем. Цей процес дозволяє ефективно підготовлювати інформацію до використання ІДС. Це є дуже актуальним, бо більшість інформації у інтернеті знаходиться у неструктурованому вигляді.

Людська мова у виді тексту зберігається у форматах pdf, word, txt. Наприклад, у медичних закладах звіти пацієнтів зберігаються природною мовою. Людська мова супроводжується багатьма проблемами для програмних систем. Вона неструктурована і має багато варіацій. Більшість програмних систем є системами, заснованими на правилах, і людська мова вимагає великої кількості правил, що не є оптимальним.

У деяких організаціях, таких як лікарні та інженерні фірми, зображення та відео становлять основну частину даних. Програмне забезпечення для обробки зображень в основному засноване на правилах і може розуміти зображення лише до певної міри. Але для вилучення повної інформації потрібні імовірнісні системи.

У колл-центрах людська мова у виді записів формує більшу частину інформації. Людська мова має велику кількість шуму. Якщо привести людську мову у виді записів у структуровану форму, то це значно підвищить якість роботи організацій, що використовують такі дані, бо буде можливість використовувати їх у програмній обробці.

Ще проблема таких даних вкривається в тому, що вони можуть зберігатися в неопрацьованому виді у базах даних та не використовуватися зовсім. Політика компанії зазвичай не дозволяє видалити записи, бо вони мають теоретичну цінність, але якщо такі записи знадобляться в майбутньому, необхідно буде витратити багато ресурсів, щоб проаналізувати їх та вилучити необхідні дані.[4]

Інформація у неструктурованому виді гірше стискається, тому зберігання неструктурованих даних є дорогим через погану ефективність обробки та витратами ресурсів на зберігання такої інформації у базі.

Цю проблему може вирішити побудова бази знань. Саме цей процес включає в себе вилучення структурованої інформації та збереження її у виді, що є зручним для програмної обробки.

Зараз існують проекти, об’єктом яких є створення ІДС. Однією з таких є YAGO (Yet Another Great Ontology)

YAGO – це ІДС на основі бази знань з відкритим вихідним кодом. Система YAGO автоматизовано вилучає дані з вікіпедії та інших джерел та формує базу знань. YAGO розроблюється з ціллю спростити пошук інформації. [5]

Зараз YAGO містить інформацію про понад 10 мільйонів організацій і понад 120 мільйонів фактів про них. Інформація в YAGO вилучається з Вікіпедії, WordNet і GeoNames. YAGO має точність понад 95%. Для можливості інтеграції YAGO було пов’язано з онтологією DBpedia і з онтологією SUMO.

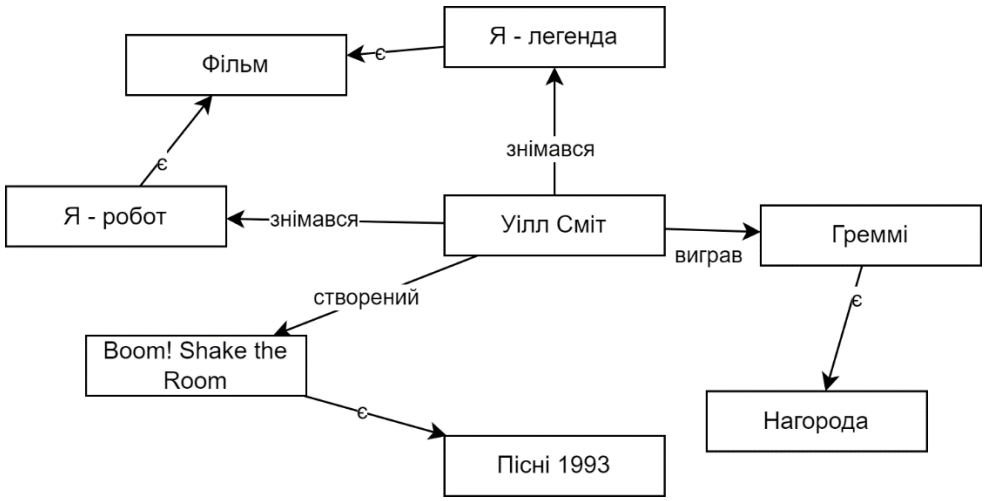


Рисунок 1.2 – Приклад схеми графу представлення знань в YAGO

1.2 Представлення знань в інформаційно-довідкових системах

База знань (БЗ) – це технологія, що може використовуватися для пошуку та управління знаннями. База знань містить структуровану мета-інформацію для опису семантики елементів предметної області. Це дозволяє виводити інформацію, що напряму не вказувалася у базі та є основною різницею між базою даних та базою знань.

БЗ надають можливість оброблювати дані, що зазвичай є неструктурованими та працювати зі семантикою на рівні програми.

Сучасні БЗ будуються на основі вилучення знань з неструктурованих даних. У процесі вилучення знань будуються бази знань. Процес побудови бази знань виконується автоматизовано або автоматично. Це дозволяє охопити великий об’єм даних.

Задача побудови БЗ полягає у виявленні елементів та зв’язків між ними у рамках предметної області у відповідності з обраною моделлю зображення знань [6].

Представлення речень в БЗ можна розділити на два компоненти: базу фактів і базу правил. Факти використовуються для висвітлення основних істин області і є атомарними. Правила використовуються для розширення словникового запасу, виражаючи порядок побудови нових відносин, вони є універсальними умовами. Як факти, так і висновки правил можна отримати за допомогою відповідності відносин.

Наприклад, є простий фрагмент БЗ:

$$\text{Мама}(\text{Джейн}, \text{Біллі}) \tag{1.1}$$

$$\text{Батько}(\text{Джон}, \text{Біллі}) \tag{1.2}$$

$$\text{Батько}(\text{Сем}, \text{Джон}) \tag{1.3}$$

$$\text{Батько}(x, y) \Leftarrow \text{Мама}(x, y) \tag{1.4}$$

$$\text{Батько}(x, y)\ ) \Leftarrow \text{Батько}(x, y) \qquad (1.5)$$

$$\text{Дитина}(x, y) \Leftarrow \text{Батько}(\ y, x) \qquad (1.6)$$

Можна прочитати останнє речення як «х є дочірнім у, якщо у є батьком х». У цьому випадку, якщо ми запитаємо базу знань, чи Джон є батьком Біллі, ми знайдемо відповідь, безпосередньо зіставивши базовий факт. Якщо ми запитаємо, чи Джон є батьком Біллі, нам потрібно буде пройти у зворотному напрямку і запитати БЗ, чи Джон був матір'ю Біллі чи батьком Біллі. Якщо б ми запитали, чи є Біллі дитиною Джона, нам потрібно було б перевірити, чи Джон був батьком Біллі, а потім перейти до перевірки матері та батька. Оскільки правила передбачають ланцюг виведень і можливий виклик інших правил, які, у свою чергу, можуть викликати більше ланцюгів, ключове питання контролю, про яке ми повинні думати, полягає в тому, як найбільш ефективно використовувати правила в БЗ [7].

Методи моделювання БЗ передбачають використання моделей знань (МЗ) з експертами. МЗ є способами перегляду БЗ за допомогою різних форм діаграм і матриць. Експерту показують елементарну МЗ, і йому/їй пропонується змінити помилки та заповнити прогалини.

Усі методи захоплення виконують три основні функції:

– зосереджують експерта на необхідних знаннях;

– допомагають експерту згадати знання, надаючи підказки або підходячи до знань з різних сторін;

– допомагають експерту чітко пояснити те, що він/вона знає.

Аналіз знань – це діяльність, яку виконує інженер з знань після того, як він/вона провів сеанс отримання знань з експертом. Він пов'язаний з визначенням елементів знань, які будуть введені в БЗ для формування її структури та основних компонентів. Ці елементи будуть використовуватися як будівельні блоки для формування всіх МЗ. Під час аналізу знань можна

визначити чотири важливі елементи: поняття, атрибути, значення та відносини. Давайте розглянемо кожен з них [8].

Поняття — це елементи, які утворюють домен. Деякі з основних типів понять:

– частини інформації;

– фізичні поняття;

– люди та ролі;

– джерела інформації;

– організації та групи;

– галузі знань;

– завдання;

– функції;

– фізичні явища;

– проблеми.

Концепції утворюють основну структуру БЗ. Інші елементи БЗ призначені для опису понять. Це робиться двома способами: властивості понять описуються за допомогою атрибутів і значень; За допомогою відношень описуються способи зв'язку пар понять один з одним.

Атрибути - це якості або ознаки, що належать до класу понять. Іншими словами, вони є способами, якими ми бачимо, що поняття відрізняються один від одного. Деякі приклади атрибутів:

– атрибути фізичних об’єктів;

– атрибути людей;

– атрибути інформації;

– атрибути організацій (розмір, товарообіг, асортимент продукції).

Значення – це специфічні ознаки або якості поняття, що відрізняють його від інших понять. Кожне значення завжди пов’язане з атрибутом. Приклади значень атрибутів:

– значення фіолетовий, зелений і жовтий є атрибутом кольору;

– значення важкий і легкий є атрибутом ваги;

– значення 3, 1 і 4 пов'язані з атрибутом числа.

Як видно з цих прикладів, значення можуть бути різних типів. Деякі значення — це прикметники, деякі — числа, а деякі — речення. Інші можуть бути абзацами тексту або фрагментами гіпертекстового коду, які містять гіперпосилання, зображення. Щоб відобразити ці різновиди, атрибути виділяються в класи, наприклад:

– числові атрибути для значень, які є числами;

– атрибути категорії для значень, які є прикметниками;

– атрибути гіпертексту для значень, які є шматками гіпертексту;

– текстові атрибути для значень, які є одним або двома реченнями.

Зв’язки – це елементи, за допомогою два поняття асоціюються один з одним. Наприклад, зв’язок між «сиром» і «їжею» такий, що сир є різновидом їжі. Цей тип відношення говорить, до якого класу належить об’єкт, і зазвичай скорочується до «is a». Основними типами відношень є:

– «є»;

– «виробляє»;

– «має частину»;

– «наслідок»;

– «виконує»;

– «причина»;

– «вимагає».

Кожне відношення також має зворотню форму. Наприклад, оберненим до «виконує» є «виконується», а оберненим до «має частину» є «є частиною». Деякі БЗ можуть включати обернені відношення, а інші — ні. Це залежить від Моделі бази знань, що розроблюється. Коли відношення з’єднує два поняття, то його називають триплетом [9].

Моделювання знань передбачає створення та використання МЗ. МЗ – це способи перегляду знань, що містяться в БЗ. Кожну МЗ слід розглядати як надання іншої точки зору, з якої можна побачити різні аспекти БЗ. МЗ також використовуються в кінцевому продукті для передачі знань кінцевим користувачам. Деякі з основних МЗ – це дерева, матриці, карти, шкали часу, фрейм [10].

Дерево – це діаграма, яка показує розташування вузлів за ієрархією. Кожен вузол представляє концепцію в БЗ, а кожне посилання представляє відношення між парою понять. Є такі типи дерев: дерево концепцій, дерево процесів, дерево композицій, дерево атрибутів, дерево причин, змішане дерево.

Дерево концепцій характеризується тим, що кожне посилання є відношенням «є». Отже, воно показує, якого типу є елементи в БЗ. Ця форма знання називається таксономією і є ключовим аспектом БЗ. Приклад дерева концепцій показано на рисунку 1.3.

Дерево композиції характеризується тим, що всі посилання є зв’язком «має частину». Це використовується, щоб показати компоненти та підкомпоненти концепції, наприклад, складний продукт, документ або організація [11].

Дерево процесів — це особлива форма дерева композиції, в якій всі вузли є процесами. Воно відображає декомпозицію процесів на підпроцеси.

Дерево атрибутів відображає значення та атрибути, які описують властивості певних елементів у БЗ.

Дерево причин містить зв’язки типу «причина». Воно використовується в проектах, які потребують знань про те, як експерти діагностують проблему.

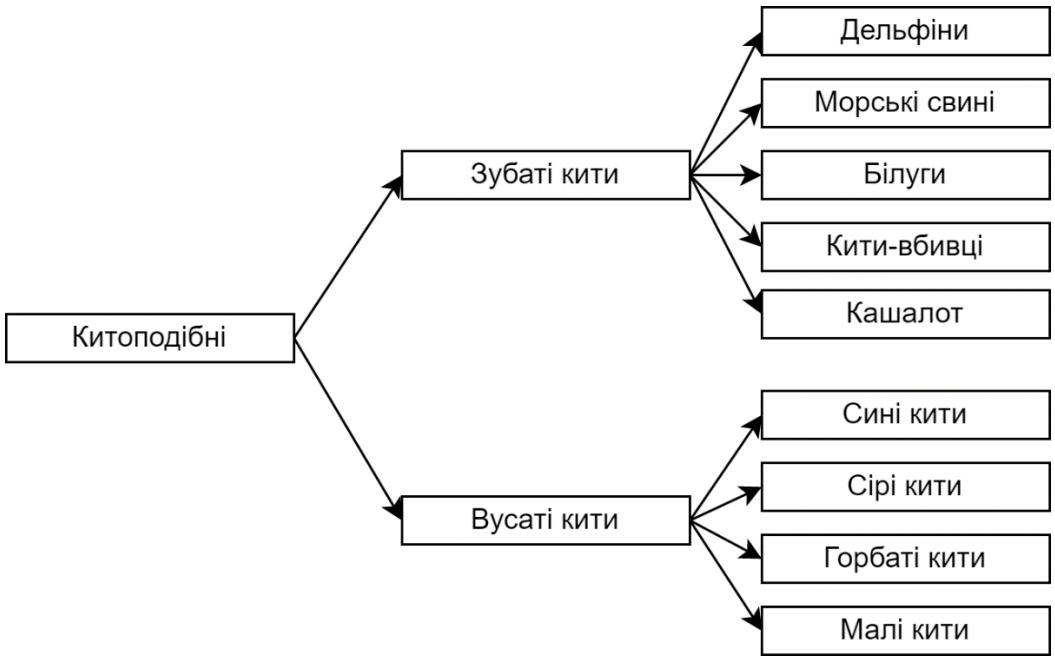


Рисунок 1.3 – Дерево концепцій

Змішане дерево – дерево, що містить не один тип відношень. Змішане дерево представлено на рисунку 1.4. Зв’язки, показані на дереві на рисунку 1.4, є «є» для чорних ліній, «має експонат» для чорних ліній із стрілками і «створено» для пунктирних ліній [12].

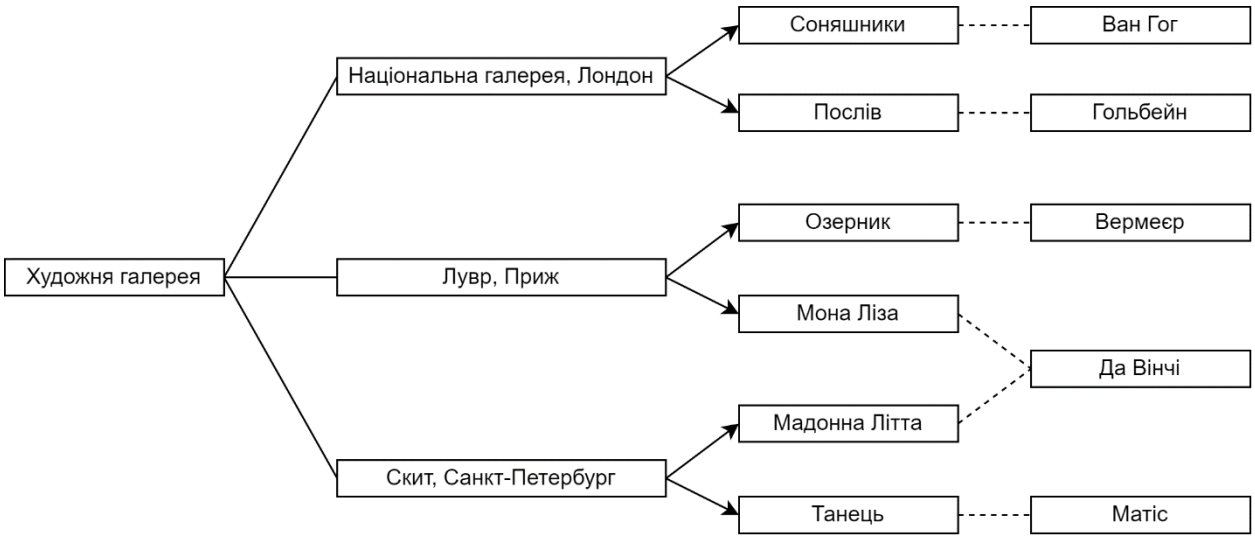


Рисунок 1.4 – Схема змішаного дерева

Можна виділити два основних типи матриць: матриця для атрибутів і матриця для відносин. Матриця атрибутів призначена для відображення властивостей набору понять. У матриці атрибутів відображені поняття на вертикальній осі та атрибути/значеннями вздовж горизонтальної осі. Приклад матриці атрибутів представлено у таблиці 1.1, де показано властивості набору напоїв [13].

Таблиця 1.1 – Приклад матриці атрибутів

		непрозорий			алкоголь		температура		газований	
		повністю	напів	прозорий	так	ні	горяча	холодна	так	ні
Напої	лимонад			+		+		+	+	
	вода			+		+		+		+
	горілка			+	+			+		+
	лікер		+		+			+	+	
	кава	+				+	+			+

Матриця зв’язків відображає два основних набори понять, що пов'язані один з одним за допомогою певного відношення. Елементи в матриці показують, які пари понять мають вказ ане відношення. На таблиці 1.2 наведено приклад матриці відносин. Матриця зв’язків показує людей, що пов’язані із галузями знань, використовуючи відношення «має досвід», наприклад, «Джейн Лентон – має досвід – управління фінансами».

Карта – це діаграма, яка показує розташування вузлів, пов’язаних стрілками. Кожен вузол представляє концепцію в БЗ, а кожне посилання представляє відношення між парою понять [14].

Таблиця 1.2 – Приклад матриці зв’язків

		люди							
		Джейн Лентон	Том Мансфі	Ян Честерф	Дейв Ньювар	Наталія Дача	Маврай с Лідс	Саймон Бредфо	Кері Лінколь
Зони знань	Фінансовий менеджмент	+					+		
	Менеджмент ресурсів		+			+			+
	Зовнішня розробка	+	+	+				+	
	Трансформація бізнесу				+			+	
	Бізнес-аналітика	+	+	+			+		

Отже, карту можна порівняти зі змішаним деревом, яке не обов’язково має бути ієрархічним. Найважливішими типами є концептуальні карти та карти процесів. Можна також використовувати інші формати діаграм, наприклад діаграми стану [15].

Концепційна карта показує різноманітні поняття, пов’язані сумішшю різних відносин. Приклад показано на рисунку 1.5 Концептуальні карти можуть бути різних видів, наприклад, ієрархічні концептуальні карти (подібні до змішаних дерев) і ті, які обмежують поняття та зв’язки.

Карта процесу показує спосіб виконання процесц. Основними елементами на карті процесу є процеси та підпроцеси, що моделюється. Ці підпроцеси розміщені на карті в тому порядку, в якому вони виконуються. Зв’язки між завданнями представляють відношення «слідує». Інші концепції можна відобразити на карті процесу з посиланнями на вузли завдання за допомогою стрілок [16].

До них можна віднести ресурси, необхідні для виконання кожного завдання; продукти кожного завдання, тригери, які викликають запуск

завдання, людей, ролі або речі, які виконують кожне завдання; точки рішення, які впливають на виконання завдань.

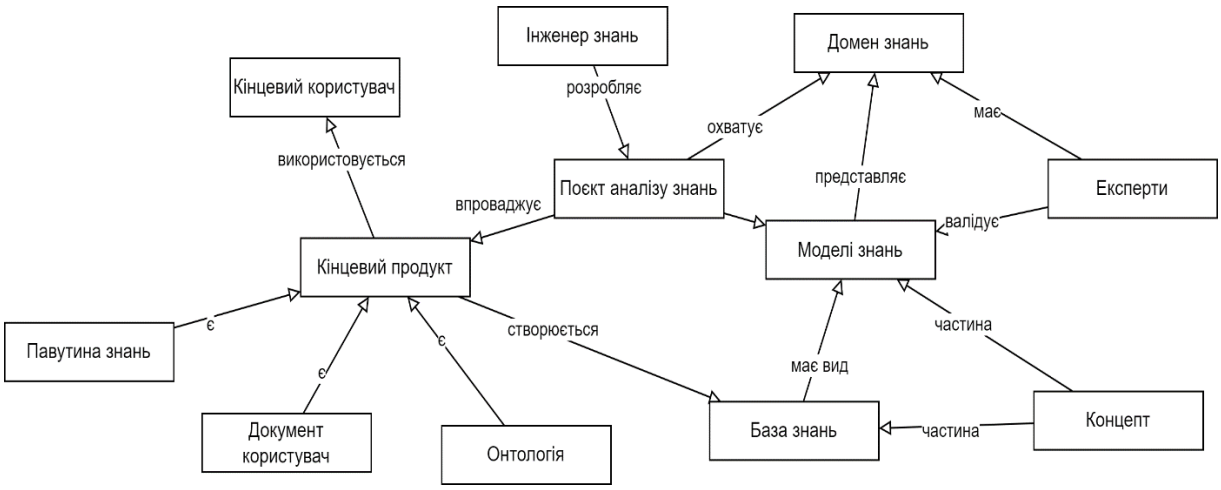


Рисунок 1.5 Приклад концептуальної карти

Часова шкала – це діаграма, яка має поняття та вузли та показує час по горизонтальній осі. Ширина кожного вузла показує, коли концепція починається і закінчується. Це можна використовувати, щоб показати фази проекту або порядок подій чи завдань. Це просте уявлення, яке часто використовується на ранніх етапах виконання захоплення. Приклад шкали часу показано на рисунку 1.6

Фрейм – це проста таблиця з 2 стовпцями, яка показує властивості концепції. Він влаштований так, що атрибути, пов’язані з поняттям, відображаються в лівій колонці, а відповідні значення — у правому. Показує два кадри, один показує атрибути та значення кави, інший показує атрибути та значення горілки [17].

Сторінка знань – це проста таблиця з 2 стовпцями, яка показує всі знання, пов’язані з концепцією. Він дуже схожий на фрейм, але показує більше інформації. На додаток до атрибутів і значень, він показує зв’язки з іншими поняттями та менш формальними знаннями, такими як описові абзаци тексту,

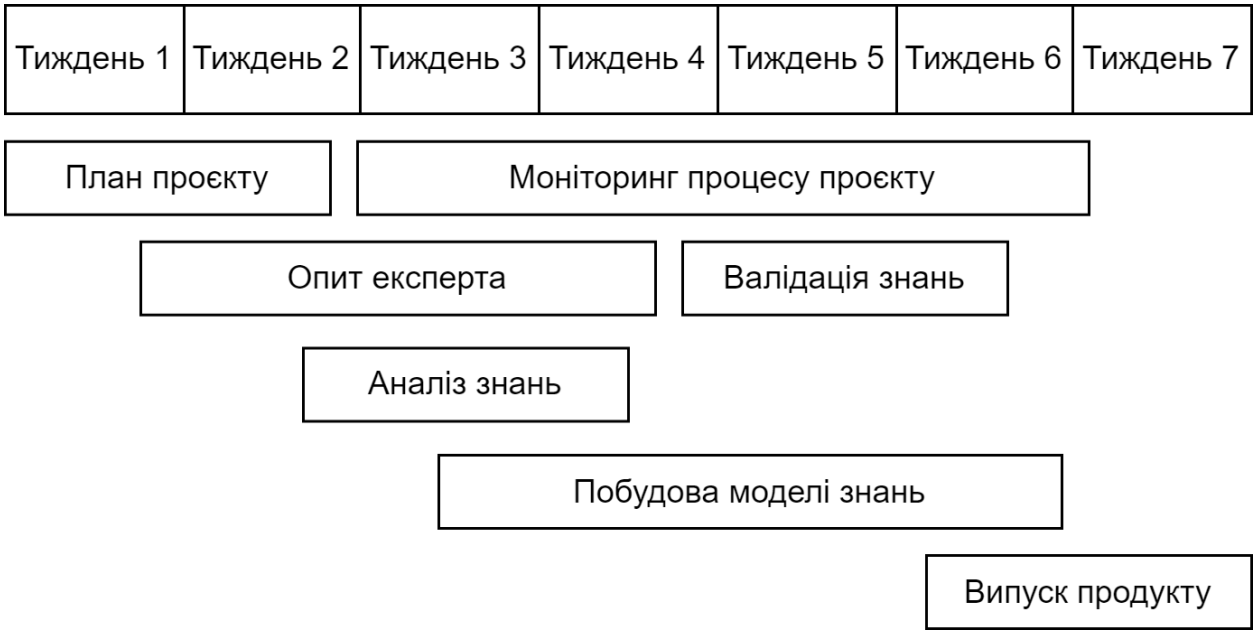


Рисунок 1.6 – Схема часової шкали

малюнки та зображення. Якщо він створений за допомогою веб-інструменту, він містить гіперпосилання на інші поняття в БЗ та на інші файли за межами БЗ (наприклад, документи, файли зображень та відеофайли) [18].

Продукційна система – це система прямих міркувань, яка використовує правила певної форми. Такі правила називаються продукційними і розроблені для представлення загальних знань. Продукційна система підтримує постійну пам'ять тверджень. Така пам’ять називається робочою пам'яттю (РП). РП схожа на базу даних, але РП змінюється набагато частіше під час роботи системи. Правило продукції – це структура, що складається з двох частин: попередній набір умов і наступний набір дій. Зазвичай правила пишуться у такій формі: ЯКЩО умова ТО дія. Попередні умови – це тести, які застосовуються до поточного стану РП. Вони постійно змінюють РП. Основна операція продукційної системи — це цикл із трьох кроків, який повторюється до тих пір, поки у РП більше не будуть застосовні правила, після чого система зупиняється [19].

Кроки ітерації циклу є такими:

– знайти, які правила застосовні, тобто ті правила, попередні умови яких задовольняються поточною робочою пам'яттю;

– серед правил, що знайдені на першому кроці, вибрати таке, що може бути виконане;

– змінити робочу пам'ять, виконавши послідовні дії всіх правил, вибраних на другому кроці. Як зазначено, цей цикл повторюється до тих пір, поки не закінчяться застосовані правила [20].

1.3 Дослідження методів автоматизованої побудови баз знань

Для вирішення задачі автоматизованої побудови бази знань використовують відповідні методи. На рисунку 1.7 представлені методи побудови баз знань.

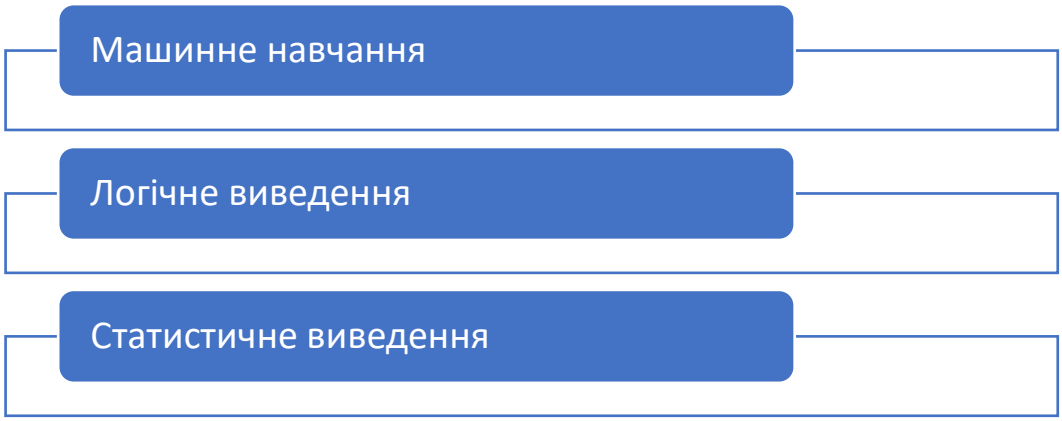


Рисунок 1.7 – Методи для побудови баз знань

Методи машинного навчання використовують напів-закриту систему для побудови бази знань. Вона використовує закономірності на великих масивах даних, щоб вирішити кінцеве виведення. Такі методи добре працюють

з великим масивом даних, але складно виявити помилки, якщо вони трапляються, бо система напів-закрита.

Методи логічного виведення використовують дану множину правил та фактів для логічного висновку на основі зіставлення сутностей фактів та правил. У таких методах легше знаходити та виправляти помилки, але вони обмежені заданим набором правил [21].

Методи статистичного виведення використовують частотні показники фактів та їх зв’язків у множині. Такі методи працюють з великим масивом даних та знаходять найбільш імовірні факти як найбільш часті у тексті.

Задача автоматизованої побудови та управління базами знань полягає у постійному виконанні етапів побудови та застосування бази знань (рисунок 1.8).

На етапі побудови БЗ збираються джерела інформації та виконується їх обробка. Вилучаються сутності та їх залежності.

На етапі доповнення БЗ виконується оновлення БЗ новими фактами та сутностями.

На етапі застосування БЗ виконуються процеси логічного висновку над БЗ для отримання інформації для користувача.

Побудова БЗ – це процес заповнення БЗ інформацією, що була вилучена з неструктурованих джерел. Неструктурованими джерелами можна вважати текстові документи, відео, зображення, звукові записи. Неструктуровані дані дуже складно використовувати у програмній обробці. БЗ впроваджує структуру, до якої приводяться уся інформація з неструктурованих джерел. Таким чином, БЗ є сутністю, яка надає інформацію у вигляді, у якому обробка цієї інформації є простою у порівнянні з неструктурованими джерелами, які майже неможливо використовувати під час обробки [22].

Побудова бази знань є складним процесом, бо він включає декілька процесів вилучення даних із джерел. Процес побудови бази знань зазвичай складається з декількох рівнів [23].

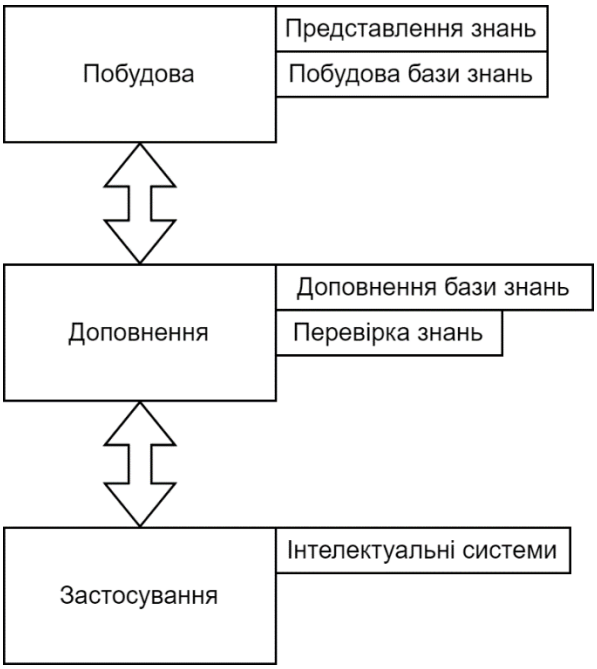


Рисунок 1.8 – Етапи автоматизованої побудови баз знань

До першого рівня входять елементи, що виконують вилучення інформації з неструктурованих джерел. Масив усіх неструктурованих даних, з якими працюють елементи першого рівня називають озером даних.

До другого рівня відносять елементи, що виконують аналіз даних, що зберігаються в графовій базі даних та логіки першого порядку. Елементи цього рівня виконують міркування на основі збережених даних. Знання у базі знань представляють у виді триплетів. Кожен триплет завжди містить два об’єкта та зв’язок між ними. Наприклад «Джон Кармак розробляв Doom» відображається триплетом «Розробляв (Джон Кармак, Doom)» [24].

Триплети у базі знань зазвичай мають імовірнісний характер. Для кожного триплету зберігається імовірність факту, що він описує. Кожен раз, коли система знаходить твердження у новому джерелі, що співпадає з вже існуючим фактом, значення імовірності існуючого триплету підвищується.

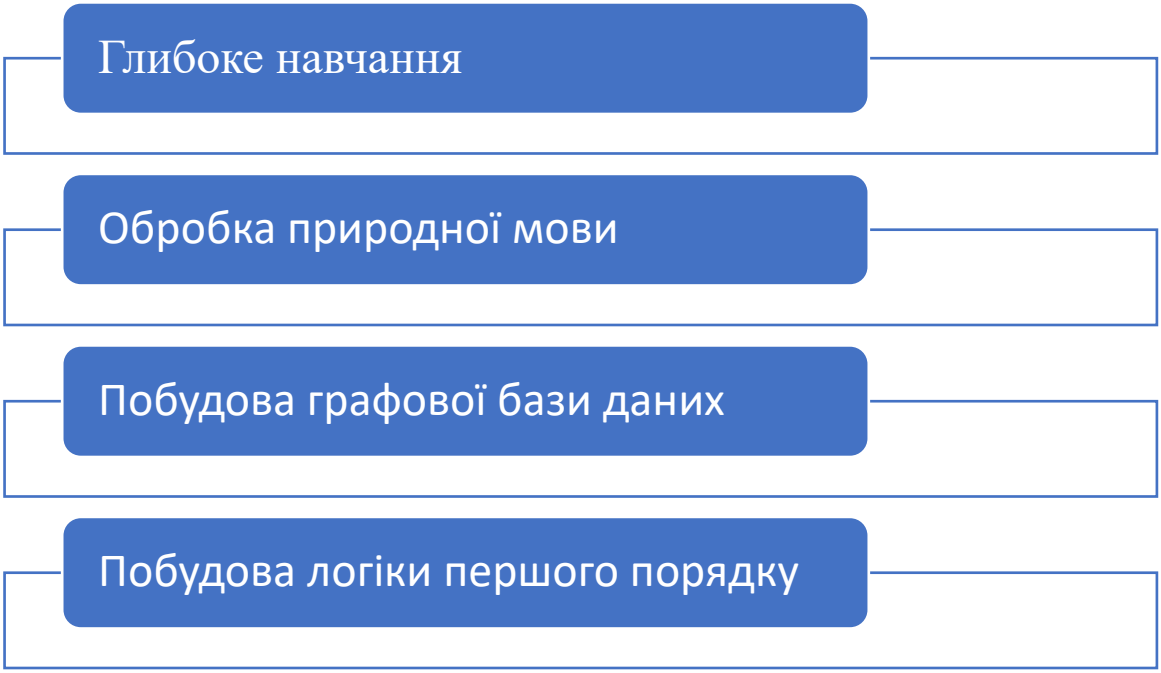


Рисунок 1.9 – Етапи першого рівня

Таким чином, нові факти у системі завжди будуть мати низьку імовірність, яка буде підвищуватися, коли будуть знаходитися нові докази.

Також на другому рівні є системи отримання знань з бази знань та навчання бази знань. Система навчання бази знань полягає у накопиченні нових фактів та підтвердженні старих [25].

До третього рівня належать прикладні системи. До таких систем належать експертні, пошукові, довідкові, діагностичні.

Зараз існують процеси налагодження автоматизованої побудови баз знань.

Однією за перших таких систем є Never Ending Language Learning (NELL). NELL – це ІДС на основі БЗ. Вона розроблена встановлювати базові семантичні зв’язки між заданими категоріями даних. У NELL є недолік, що тільки невеликий відсоток інформації є перевіреним. Під час синтезу фактів існує імовірність помилки. Факти потребують перевірки експертів [8].

Google використовує Knowledge graph (KG) для підвищення якості пошукової системи. KG – це комплекс технологій та семантична база знань. KG використовується, щоб доповнювати результат пошуку. Це дозволяє отримувати інформацію без переходу на посилання.

Document360 – це веб-базоване програмне забезпечення для управління знаннями. Document360 працює за принципом «програмне забезпечення – це сервіс». Платформа впроваджує інструменти для створення, підтримки та користування базою знань для самообслуговування клієнтів і внутрішніх користувачів. Document360 має інструменти:

– текстовий редактор Markdown. Це простий для користувача редактор, що дозволяє стилізувати текстовий документ, використовуючи типові методи форматування, включаючи заголовки, акценти, списки, зображення та посилання;

– менеджер категорій, що дозволяє створювати зміст для навігації по документації. Спрощує користувачам пошук і перетравлення інформації;

– налаштування цільової сторінки, що дозволить персоналізувати базу знань за допомогою власного дизайну, логотипу, зображень заголовків, посилань, домену та десятків інших опцій, включаючи користувацькі CSS;

– ролі користувачів для створення прав доступу. Document360 дозволяє застосовувати обмеження, хто може отримати доступ до заданих функцій;

– засіб контролю версій, що дозволяє вести облік усіх змін в проекті Document360;

– Document360 автоматично створює резервні копії проектів щодня. Це дозволяє відновити весь проект або частину бази знань до попереднього стану з доступної резервної копії;

– підтримка інтеграцій, до дозволяє підключити інші сервіси, або іншу базу знань.

DeepDive – це ІДС на основі БЗ для вилучення структурованої інформації з темних джерел. Вона використовує машинне навчання. Обробка даних у DeepDive виконується в етапах. Етапи представлені на рисунку 1.10.

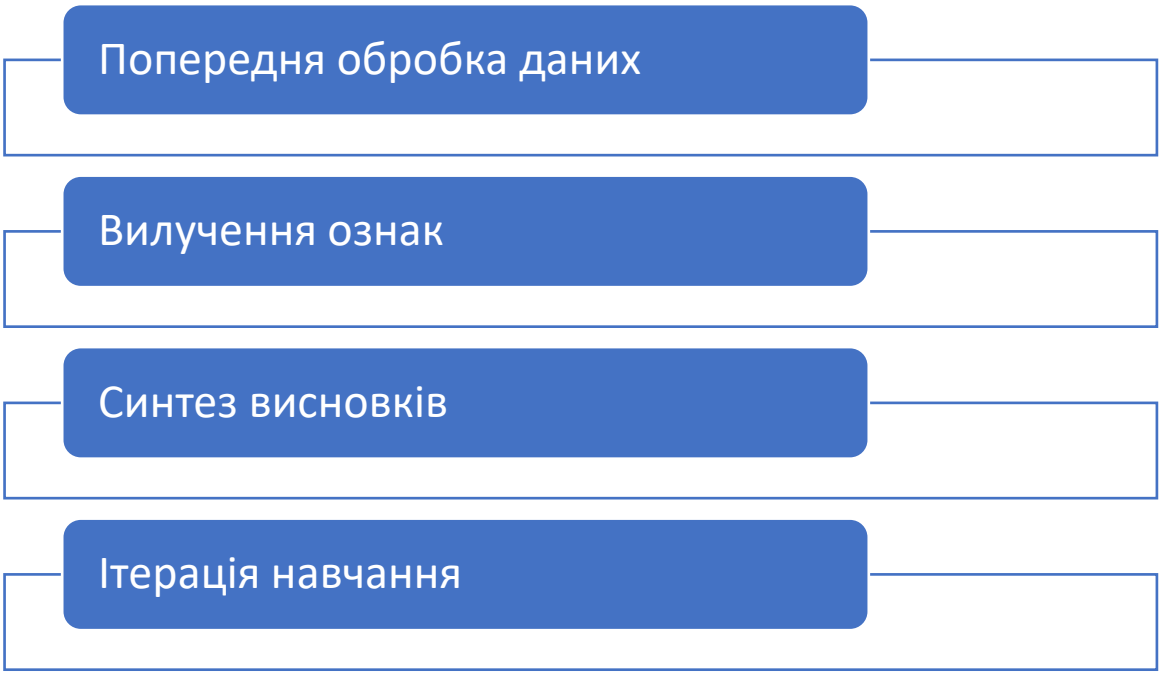


Рисунок 1.10 – Етапи обробки даних в DeepDive

На етапі попередньої обробки даних DeepDive завантажує вхідні дані у текстовому форматі у базу даних і аналізує їх, щоб отримати інформацію на рівні речення, включаючи слова в кожному реченні, теги POS, теги іменованих об’єктів тощо [9].

На етапі вилучення ознак DeepDive перетворює вхідні дані в сигнали відношень, які називаються доказами, запускаючи екстрактори. Докази включають: кандидатів на відносини, лінгвістичні особливості для цих кандидатів.

На наступному кроці DeepDive використовує докази для створення діаграми факторів. Щоб надати DeepDive, як створити цей факторний графік,

розробники використовують декларативну мову, подібну до SQL, для визначення правил висновку.

На наступному кроці DeepDive автоматично виконує навчання та статистичний висновок на згенерованому факторному графіку. Під час навчання обчислюються значення ваги факторів, зазначені в правилах висновку. Ці ваги представляють довіру до правила. Під час висновку обчислюються граничні ймовірності змінних.

Після висновку результати зберігаються в наборі таблиць бази даних. Розробник може отримати результати за допомогою запиту SQL, перевірити її за допомогою калібрувального графіка та виконати аналіз помилок для покращення результатів.

Перший етап заповнює базу даних за допомогою набору запитів SQL і визначених користувачем функцій. DeepDive зберігає всі документи в базі даних [26].

Після цього кроку завантаження DeepDive виконує розміщення кандидатів через запити SQL, які створюють можливі згадки, сутності та відносини, і вилучення ознак, які асоціюють функції з кандидатами, наприклад, «(об’єкт) і його дружина (об’єкт)».

Такі правила повинні запам’ятовуватися, бо якщо зв’язка кандидатів губить факт, DeepDive не має шансів вилучити його. Далі необхідно вилучити ознаки. Для цього розширюється Марковська модель. По-перше, вводяться визначені користувачем функції. По-друге, вводяться ваги.

Припустимо, що фраза (m1, m2, речення) повертає фразу між двома згадками в реченні, наприклад, «і його дружина» у наведеному вище прикладі. Фраза між двома згадками може вказувати на те, чи одружені дві людини. Це можна записати так: ЗгадкаПроОдруження(m1, m2), що розкладається на КандидатиНаОдруження(m1, m2), Згадка(s, m1), Згадка(s, m2), Речення(s, речення), тоді вага дорівнює фраза (m1, м2, речення) [27].

Це правило говорить, що те, чи вказує текст, що згадки m1 і m2 одружені, залежить від фрази між цими парами згадок. На основі даних про навчання система може зробити висновок про впевненість, що двоє згадок є одруженими. Фраза повертає ідентифікатор, який визначає, які коефіцієнти ваги слід використовувати для даного відношення, згаданого в реченні. Якщо фраза повертає однаковий результат для двох згадок відношення, то вони отримують однакову вагу. Це дозволяє DeepDive підтримувати звичайні приклади функцій, таких як «мішок слів» до контекстно-залежних функцій НМ, до словників та онтологій, що дуже специфічні для домену. На додаток до визначення наборів класифікаторів, DeepDive успадковує здатність Марковських моделі вказувати кореляції між сутностями за допомогою зважених правил. Такі правила особливо корисні для очищення та інтеграції даних [28].

Так само, як і в Марковскій моделі, DeepDive може використовувати навчальні дані або докази про будь-які відносини; зокрема, кожне відношення користувача пов'язане з відношенням доказів з тією ж схемою та додатковим полем, яке вказує, є запис істинним чи хибним.

На етапі навчання та висновку DeepDive генерує факторний графік, подібний Марковскій моделі і виконує розширення статистичного висновку.

DeepDive виконує зазначені вище три фази послідовно, і в кінці навчання та висновку отримує граничну ймовірність для кожного факту-кандидата. Щоб створити остаточну базу даних, користувач вибирає факти, в яких DeepDive впевнений, наприклад, де імовірність більше 0,95. Зазвичай користувачеві потрібно перевірити помилки та повторити попередні кроки, процес, який ми називаємо аналізом помилок. Аналіз помилок – це процес рішення найбільш поширених помилок таких, як неправильне виділення, занадто подрібні ознаки, помилки кандидатів, тощо.

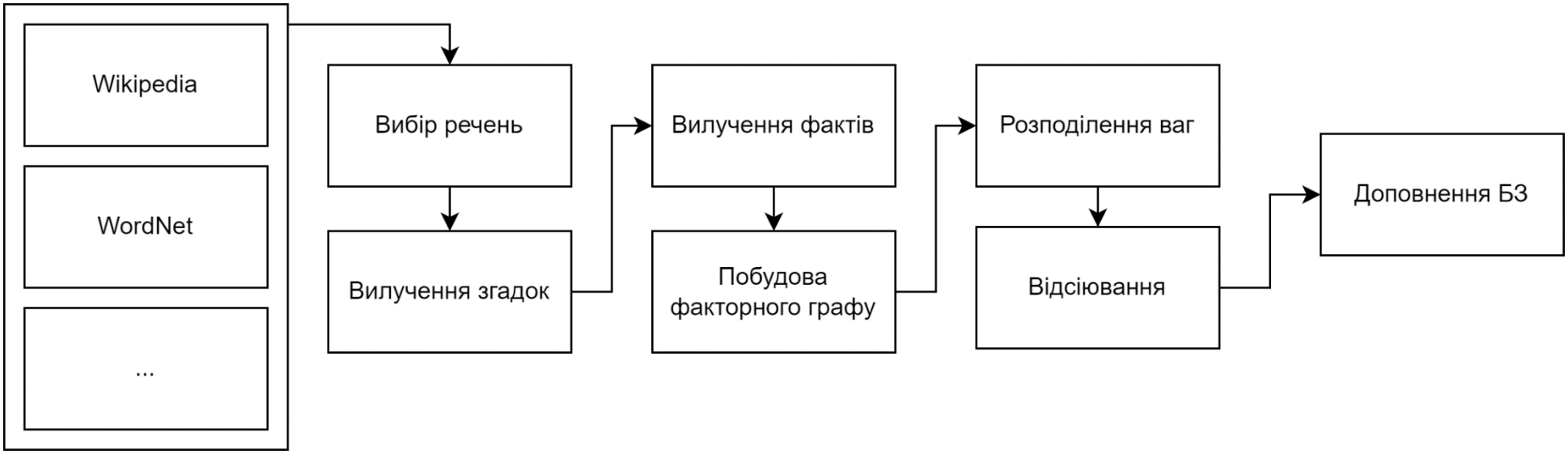


Рисунок 1.11 – Схема процесу обробки даних для представлення їх у базі знань у DeepDive

YAGO витягує дані з WordNet та Wikipedia [29].

WordNet — це семантична база англійської мови. WordNet розрізняє слова, які з’являються в текстах, і їх значення слів. Набір слів, що мають одне значення, називається синсетом. Таким чином, кожен синсет ідентифікує одне значення (тобто семантичне поняття). Слова з кількома значеннями (неоднозначні слова) належать до кількох синсетів.

WordNet надає зв'язки між синсетами, такими як гіпернімія/гіпонімія (тобто зв'язок між підконцептом і суперконцептом) і голонімія/меронімія (тобто відношення між частиною і цілим). Концептуально відношення гіпернімії в WordNet охоплює орієнтований ациклічний граф з одним вихідним вузлом, який називається Entity.

Wikipedia — це багатомовна веб-енциклопедія. Кожна стаття Вікіпедії є окремою веб-сторінкою і описує одну тему. Більшість сторінок Вікіпедії були вручну підписані до однієї або кількох категорій. Сторінка про Альберта Ейнштейна, наприклад, знаходиться в категоріях філософи німецької мови, швейцарські фізики та ще 34 категорії. Категоризація сторінок у Wikipedia та їх структура посилань доступні у вигляді таблиць SQL.

ІДС YAGO розроблена для вилучення онтології з WordNet і Wikipedia. YAGO розроблено з можливістю розширення, тобто до онтології можна додавати нові факти з нових джерел. Для цього кожен факт позначається значенням довіри від 0 до 1. Наразі всі факти позначаються емпіричною оцінкою достовірності, яка знаходиться між 0,90 та 0,98. Факти, отримані іншими методами (наприклад, на основі статистичного навчання), можуть мати менші значення довіри.

Wikipedia має факти про більшу кількість людей, ніж WordNet, інформація про осіб для YAGO взяті з Вікіпедії. Кожна назва сторінки Вікіпедії є кандидатом на те, щоб стати окремою особою в YAGO. Наприклад, назва сторінки «Альберт Ейнштейн» є кандидатом на те, щоб стати об’єктом. Назви сторінок у Вікіпедії унікальні. Щоб встановити для кожної людини свій клас, YAGO використовує систему категорій Вікіпедії. Категорії Вікіпедії

організовані у орієнтований ациклічний граф. Однак ця ієрархія відображає лише тематичну структуру сторінок Вікіпедії. Таким чином, ієрархія мало корисна з онтологічної точки зору. Тому YAGO бере лише листові категорії Вікіпедії та ігнорує всі інші категорії. Потім YAGO використовує WordNet для встановлення ієрархії класів, тому що WordNet пропонує онтологічно чітко визначений систематику синсетів.

Кожний синсет WordNet стає класом YAGO. YAGO відсіює власні іменники, відомі WordNet, які насправді були б особами (наприклад, Альберт Ейнштейн також відомий WordNet, але виключений). Існує приблизно 15 000 випадків, коли сутність надається як WordNet, так і Вікіпедією (тобто синсет WordNet містить загальний іменник, який є назвою сторінки Вікіпедії). У деяких із цих випадків сторінка Вікіпедії описує особу, яка носить загальний іменник як власне ім’я. Однак у переважній більшості випадків сторінка Вікіпедії стосується звичайних іменників. YAGO завжди віддає перевагу WordNet і відкидає Вікіпедію в разі конфлікту. Таким чином YAGO втрачає інформацію про осіб, які носять загальний іменник як ім’я, але це гарантує, що всі загальні іменники є класами, і жодна сутність не дублюється [30].

Ієрархія класів subClassOf взята з відношення гіпонімії з WordNet: клас є підкласом іншого, якщо перший синсет є гіпонімом другого. Тепер нижчі класи, витягнуті з Вікіпедії, мають бути пов’язані з вищими класами, витягнутими з WordNet. Наприклад, «американські люди в Японії» від Вікіпедії потрібно зробити підкласом людини класу WordNet.

Для категорії «американські люди в Японії» основними сутностями є «американці», «люди» та «в Японії». YAGO прив’язує головну складову назви категорії «люди» до її форми однини «людина». Потім перевіряє, чи існує синсет WordNet для сутності та головного з’єднання «американська людина». Якщо це так, клас Вікіпедії стає підкласом класу WordNet. Якщо це не так, YAGO використовує те, що назви категорій у Вікіпедії є майже виключно ендоцентричними складними словами (тобто назва категорії є гіпонімом її головного з’єднання, наприклад, «американська людина» є гіпонімом

«людина»). Головна сутність «особа» має бути зіставлено з відповідним синсетом WordNet. Це зіставлення є нетривіальним, оскільки одне слово може посилатися на кілька синсетів у WordNet.

Для вирішення неоднозначності YAGO застосовує наступний метод. WordNet зберігає в кожному слові частоти, з якими воно посилається на можливі синсети. YAGO зіставляє головну складову з найбільш частим синсетом, це дає правильний синсет у переважній більшості випадків. Таким чином, клас Вікіпедії «американські люди в Японії» стає підкласом класу WordNet «людина».

Деякі об’єкти необхідно виправляти вручну. Усі категорії з головною складовою «capital» у Вікіпедії означають «столицю», але найпоширенішим значенням у WordNet є «фінансовий актив».

Підсумовуючи, YAGO отримує повну ієрархію класів, де верхні класи походять із WordNet, а листи — з Вікіпедії.

Вікіпедія та WordNet також дають інформацію про значення слів. Наприклад, WordNet розкриває значення слів за його синсетами. Наприклад, слова «міський центр» і «мегаполіс» належать до синсету «місто». YAGO використовує цю інформацію двома способами. Спочатку YAGO вводить клас для кожного синсету, відомого WordNet (тобто «місто»). По-друге, YAGO встановлює відношення між кожним словом синсету та відповідним класом (тобто ««мегаполіс», означає, «місто»»).

Вікіпедія надає імена для осіб за допомогою системи перенаправлення: перенаправлення Вікіпедії — це віртуальна сторінка Вікіпедії, яка посилається на справжню сторінку Вікіпедії. Ці посилання служать для перенаправлення користувачів на правильну статтю Вікіпедії. Наприклад, якщо користувач ввів «Ейнштейн, Альберт» замість «Альберт Ейнштейн», то для «Ейнштейн, Альберт» буде віртуальна сторінка перенаправлення, яка посилається на «Альберт Ейнштейн». YAGO використовує сторінки перенаправлення, щоб дати альтернативні імена для сутностей. Для кожного перенаправлення YAGO

вводить відповідний факт засобу (наприклад, («Ейнштейн, Альберт», означає Альберт Ейнштейн)) [31].

1.4 Постановка задачі

Ця робота присвячена вирішенню задачі удосконалення методу автоматизованої побудови баз знань в інформаційно-довідковій системі на основі автоматизованого вилучення правил із вхідних документів. Актуальність даної задачі є наслідком невідповідності можливостей існуючих методів автоматизованої побудови баз знань для інформаційно-довідкових систем, які використовують додаткові зовнішні бази типових правил, та практичними потребами до процесу автоматизованої побудови баз знань безпосереднього аналізу документів.

Метою роботи є дослідження методів автоматизованої побудови баз знань для інформаційно-довідкових систем для підвищення їх ефективності на основі виявлення правил нових типів з тексту довідкових документів.

Об’єкт дослідження: процес побудови баз знань в інформаційно-довідкових системах.

Предмет дослідження: методи автоматизованої побудови баз знань в інформаційно-довідкових системах.

У даній магістерській кваліфікаційній роботі вирішуються наступні задачі:

– аналіз роботи та призначення ІДС;

– аналіз методів представлення знань у базах знань інформаційно-довідкових системах;

– дослідження роботи існуючих інформаційно-довідкових систем з методами автоматизованої побудови баз знань;

– аналіз існуючих методів автоматизованої побудови баз знань для

інформаційних довідкових систем;

– розробка удосконаленого методу автоматизованої побудови баз знань в інформаційно-довідкових системах;

– розробка вимоги до програмної частини;

– розробка програмної частини з використанням удосконаленого методу;

– експериментальна перевірка розробленого методу.

2 УДОСКОНАЛЕННЯ МЕТОДУ АВТОМАТИЗОВАНОЇ ПОБУДОВИ БАЗ ЗНАНЬ ДЛЯ ІНФОРМАЦІЙНИХ ДОВІДКОВИХ СИСТЕМ

2.1 Методи автоматизованої побудови баз знань в інформаційно-довідкових системах

Розглядається метод, що використовується в ІДС ProbKB.

На вході існуюча система вилучення знань (СВЗ), виділяє з тексту сутності правила та факти. З них будується логічна мережа Маркова (ЛММ)

(ЛММ) є математичною моделлю для відображення імовірнісних фактів і правил. ЛММ використовуються для моделювання імовірнісних БЗ, створених СВЗ.

ЛММ — це набір зважених формул першого порядку, де вагові коефіцієнти вказують на те, наскільки вірогідна формула.

Приклад ЛММ:

$$0.96 \text{ народився в}(\text{Рут Губер}, \text{Нью} - \text{Йорк}), \qquad (2.1)$$

$$1.40\ \forall x \in W, \forall y \in P : \text{жив в}(x, y) \leftarrow \text{народився в}(x, y) \qquad (2.2)$$

Приклад ЛММ описує факт, що Рут Грубер народилася в Нью-Йорку, і правило, що якщо письменник x народився в області y, то x живе в y. Однак обидва твердження не є однозначними. Ваги 0,96 і 1,40 визначають, наскільки твердження вірогідні. Обидва вони є частиною ЛММ, але мають різні цілі: твердження є фактом, а правило описує правило висновку, тому їх необхідно розглядати окремо. ЛММ допускають жорсткі правила, які ніколи не повинні порушуватися. Ці правила мають вагу ∞ [13].

Наприклад:

$$\infty\ \forall x \in C, \forall y \in C, \forall z \in W :$$

$$(\text{народився в}(z, x) \wedge \text{народився в}(z, y) \rightarrow x = y). \qquad (2.3)$$

де x, y, z – факти, C, W – класи.

Це твердження говорить, що об’єкт не може народитися в двох різних містах. Факти, що порушують жорсткі правила, розглядаються як помилки і видаляються, щоб уникнути подальшого поширення.

ЛММ можна розглядати як шаблон для побудови факторних графів. Факторний граф — це набір факторів Ф = {φ1, . . . , φN }, де кожен фактор φ_i є функцією $\varphi_i(X_i)$ над стохастичним вектором X_i, що вказує на зв'язки між випадковими величинами в X_i. Ці фактори разом визначають спільний розподіл ймовірностей.

На рисунку 2.1 представлений факторний граф. Квадрати відображають фактори, кола – об’єкти. Зв’язки відображають, які об’єкти який фактор використовує [14].

На рисунку 2.1 у колах відображені цифри, що відповідно:

– народився в(Рут Грубер, Нью-Йорк);

– народився в (Рут Грубер, Бруклін);

– жив в(Рут Грубер, Нью-Йорк);

– жив в(Рут Грубер, Бруклін);

– знаходиться в(Бруклін, Нью-Йорк).

А у квадратах відображені фактори відповідно:

– народився в (Рут Грубер, Нью-Йорк);

– народився в (Рут Грубер, Бруклін);

– ∀x ∈ W ∀y ∈ P (жив в(x, y) ← народився в(x, y));

– ∀x ∈ W ∀y ∈ P (жив в(x, y) ← народився в(x, y));

– ∀x ∈ P ∀y ∈ C ∀z ∈ W (знаходиться в(x, y) ← жив в(z, x) ∧ жив в(z, y));

– ∀x ∈ P ∀y ∈ C ∀z ∈ W (знаходиться в(x, y) ← народився в(z, x) ∧ народився в(z, y)).

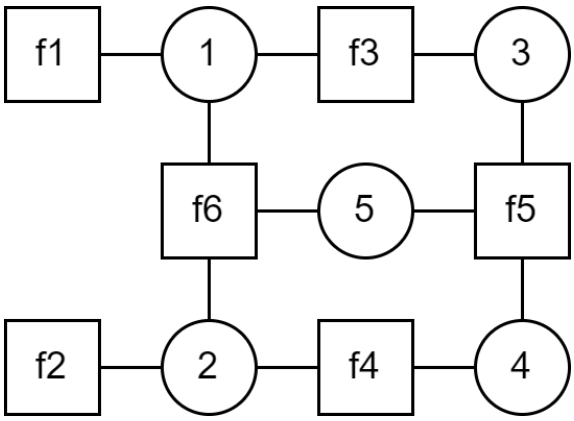


Рисунок 2.1 – Схема факторного графу

Кінцевий факторний граф називається основний факторний граф.

У таблиці 2.1 представлена модель знань для ProbKB. Для сутностей та класів створюється стохастичний вектор X, що містить одну булеву випадкову величину для кожного можливого фактору, що з'являються в правилах та взаємозв'язках. Випадкові величини, створені таким чином, також називають основними атомами. Кожен основний атом має значення 0 або 1, що вказує на його істинність [15].

Імовірнісну БЗ необхідно визначити таким чином:

Імовірнісна база знань є кортежом з п'яти елементів Г=(E,C,R,П,L), де E – набір сутностей. Кожна сутність відноситься до реального об'єкта. C – це набір класів. Кожен клас є підмножиною з E. R – набір відношень. Кожне відношення визначає бінарне відношення на [Ci, Cj]. [Ci, Cj] називається діапазоном R. R(Ci, Cj) використовується для позначення відношення та його діапазону. П – це набір зважених фактів. L – це набір зважених правил.

Модель Маркова підтримує загальні формули першого порядку, але необхідно обмежити правила тільки набором правил, що можуть бути описані диз'юнктом Хорна. Диз'юнкт Хорна — це диз'юнкт тільки з одним позитивним літералом.

Таблиця 2.1 – Модель представлення знань

Сутності (E)	Класи (C)	Зв’язки (R)	Взаємозв’язки (П)
Рут Грубер	W (Письменник) = {Рут Грубер}	народився в(W, P), народився в(W, C)	0,96 народився в(Рут Грубер, Нью-Йорк)
Нью-Йорк	C (Місто) = {Нью-Йорк}	жив в(W, P), жив в(W, C)	0,93 народився в(Рут Грубер, Бруклін)
Бруклін	P (Місце) = {Бруклін}	знаходиться в(P, C)	
Правила (L)			
1,40 $\forall x \in W \; \forall y \in P$ (жив в(x, y) ← народився в(x, y))			
1,53 $\forall x \in W \; \forall y \in C$ (жив в(x, y) ← народився в(x, y))			
0,32 $\forall x \in P \; \forall y \in C \; \forall z \in W$ (знаходиться в(x, y) ← жив в(z, x) ∧ жив в(z, y))			
0,52 $\forall x \in P \; \forall y \in C \; \forall z \in W$ (знаходиться в(x, y) ← народився в(z, x) ∧ народився в(z, y))			
∞ $\forall x \in C \; \forall y \in C \; \forall z \in W$ (народився в(z, x) ∧ народився в(z, y) → x = y)			

Його можна записати як:

$$u \leftarrow p, q, \ldots, t \qquad (2.4)$$

де u, p, q, t – факти.

Це обмежує використання окремих правил, але завдяки масштабу та масштабу набору правил системи Шерлок, звідки отримуються правила, можна робити висновок на великій кількості фактів. Диз'юнкти Хорна надають ряд додаткових переваг:

Диз'юнкти Хорна мають просту структуру, що дозволяє ефективно моделювати їх у реляційних таблицях і розробляти алгоритми висновку на основі SQL.

Для опису реляційної моделі спочатку необхідно ввести позначення. Для кожного елемента БЗ, де X ∈ {C, R, П, H} необхідно позначити відповідне відношення бази даних через T_X. Таким чином, T_X є набором кортежів та таблицею. У реалізації також необхідно використати словникові таблиці D_X, де X ∈ {E, C, R}, щоб уникнути порівняння строкових значень під час SQL запитів [16].

На відміну від інших компонентів, правила не відображаються безпосередньо на реляційну схему, оскільки вони можуть відрізнятися одне від іншого. Підхід до цієї проблеми полягає в тому, що необхідно структурно розділити правила так, щоб виділити деяку кількість типів правил. Для кожного типу правил створюється своя таблиця M_i.

$T_Ф$ визначається як набір кортежів {(I1, I2, I3, w)} і є матрицею, що відображає факторний граф, де I1, I2, I3 – зовнішні ключі до $T_П$(I), а w ∈ R – вага. Кожен кортеж (I1, I2, I3, w) представляє зважене основне правило I1 ← I2, I3. I1 – голова, а I2, I3 є тілом і може бути NULL для факторів розмірів 1 або 2. На рисунку 2.2 показано приклад $T_Ф$. Як кінцевий результат заземлення, він служить проміжним уявленням, яке є вхідними даними в механізм імовірнісного висновку. Механізм імовірнісного висновку визначає достовірність фактів, які використовуються в експериментах.

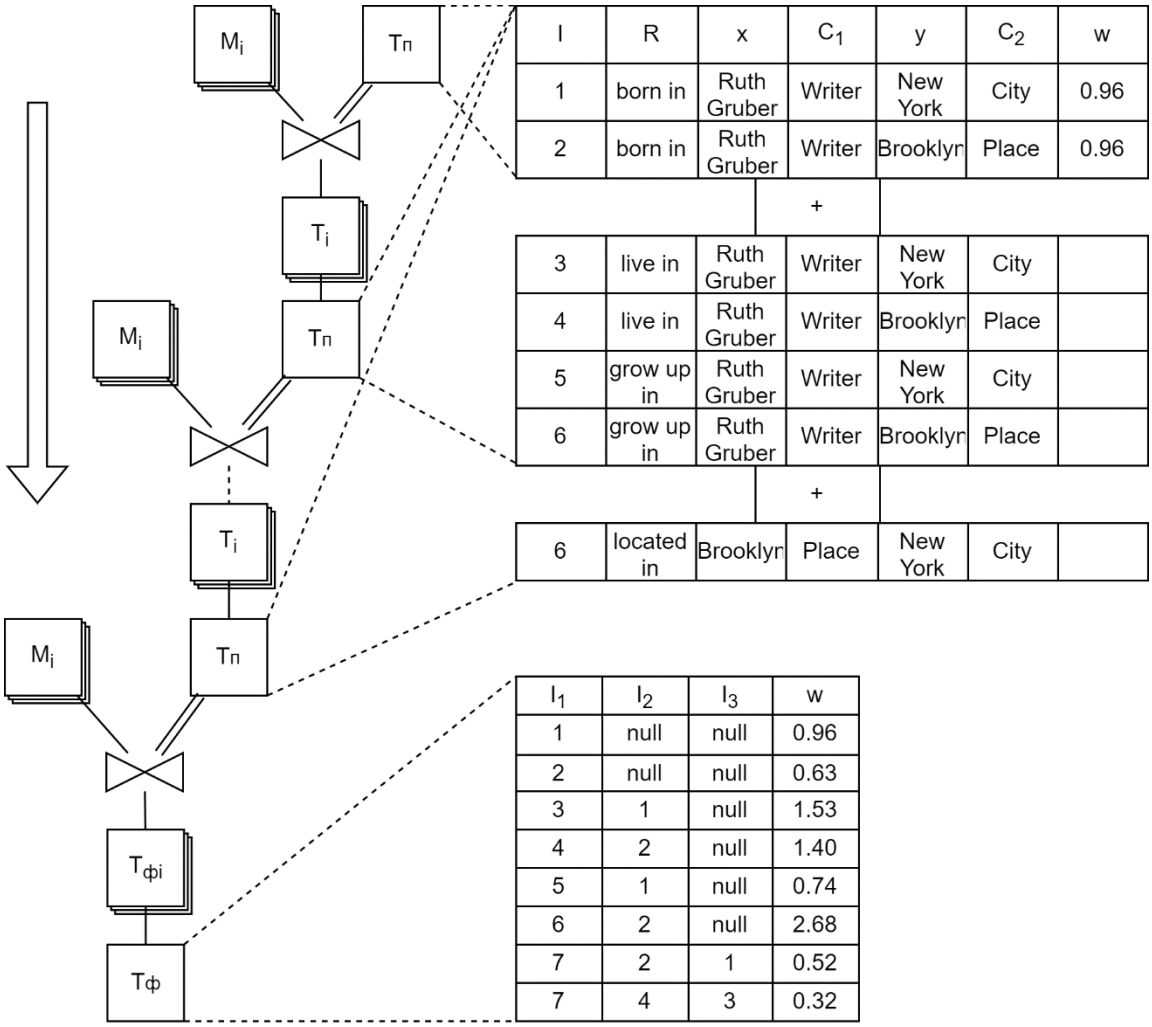


Рисунок 2.2 – Процес побудови факторного графу

Таблиця 2.2 – Базові факти ($T_П$)

I	R	x	C_1	y	C_2	w
1	народився в	Рут Грубер	Письменник	Нью-Йорк	Місто	0.96
2	народився в	Рут Грубер	Письменник	Бруклін	Місце	0.93

Таблиця 2.3 – Правила (M_1)

R_1	R_2	C_1	C_2	w
жив в	народився в	Письменник	Місце	1.40
жив в	народився в	Письменник	Місто	1.53
виріс в	народився в	Письменник	Місце	2.68
виріс в	народився в	Письменник	Місто	0.74

Таблиця 2.4 – Правила (M_3)

R_1	R_2	R_3	C_1	C_2	C_3	w
знаходиться в	жив в	жив в	Місце	Місто	Письменник	0.32
знаходиться в	народився в	народився в	Місце	Місто	Письменник	0.52

Алгоритм побудови факторного графу складається з двох кроків: спочатку необхідно застосувати правила для обчислення основних атомів, поки не отримаємо транзитивне замикання. Потім необхідно застосувати правила для побудови основних факторів [18].

Для забезпечення правильності фактів необхідно визначити правильні висновки. Для цього можна припустити, що факти, отримані з кількох джерел, або висновки з фактів із кількох джерел, будуть правильними з більшою імовірністю, ніж ті, що трапилися один раз. Необхідно розділити базу даних для фактів, переміщуючи вірогідні факти (переконання) в одну таблицю, а невірогідні факти (кандидати) в іншу. Кандидати переводяться до переконань, якщо є впевненість в їх вірності.

Загальні етапи описаного вище методу є такими:

Етап 1 Побудова фактів за допомогою систем вилучення інформації на основі аналізу текстів.

Крок 1.1 Вибір документу

Крок 1.2 Побудова стеммів (стеммінг).

Крок 1.3 Виділення триплетів (сутність, властивість, відношення)

Крок 1.4 Побудова фактів на основі виділених триплетів

Етап 2 Вибір правил, що визначають зв’язки між фактами.

Крок 2.1 Вибір правил, що можуть бути описані диз’юнктом Хорна із сету Sherlok.

Етап 3 Розрахунок ваг правил

Крок 3.1 Формування факторного графу.

Крок 3.2 Розрахунок розподілення за методом Гіббсу

Крок 3.3 Формування зважених правил

Крок 3.4 Упорядкування правил за вагою

Описаний метод в автоматизованому режимі виконує побудову зваженої бази знань. Але він використовує статичний набір правил, що були сформовані системою Sherlok. Це обмежує кількість фактів, що може бути отримано в процесі висновку з базових фактів на основі правил. Пропонується вирішити цю проблему, змінивши етап 2, де ми будемо використовувати вилучення власних правил з джерел, що опрацьовуємо.

2.2 Удосконалений метод побудови зважених правил БЗ в ІДС

Удосконалений метод полягає у тому, щоб вилучати правила для БЗ із вхідного тексту. Для цього необхідно зробити припущення, що якщо факти зустрічаються у одному джерелі, то між ними є зв’язок.

Ми знаходимо множини розміщень двох та трьох фактів з множини усіх фактів джерела. Далі ми припускаємо, що кожний отриманий кортеж є зваженим правилом, що описано диз’юнктом Хорна. Такі правила записуються до окремої таблиці. Чим частіше зустрічається конкретне правило, тим більше вага правила.

Удосконалений метод побудови зважених правил має наступні етапи:

Етап 1 Побудова фактів за допомогою систем вилучення інформації на основі аналізу текстів.

Крок 1.1 Вибір документу

Крок 1.2 Побудова стеммів (стеммінг)

Результатом кроку є множина основ слів

$$Q = \{q_i, p_l\}. \quad (2.5)$$

де Q – це множина слів, p_l – це розділ тексту, q_i – це слово.

Крок 1.3 Виділення триплетів (сутність, властивість, відношення)

$$T = \{(o_i, r_k, s_h): o_i, r_k, s_h \in Q\}. \quad (2.6)$$

де o_i – це об'єкт, r_k – це відношення, s_h - це суб'єкт

Крок 1.4 Побудова фактів на основі виділених триплетів

$$F = \{f_j: f_j = (o_{j,i}, r_{j,k}, s_{j,h}), \forall(m \neq j), o_j, i \neq o_{m,i} \; or \; r_{j,k} \neq r_{m,k} \; or \; s_{j,h} \neq s_{m,h}\}. \quad (2.7)$$

де o_i – це об'єкт, r_k – це відношення, s_h - це суб'єкт

Етап 2 Побудова правил, що визначають зв'язки між фактами.

Крок 2.1 Вибір розділу документу (p_l)

Крок 2.2 Вибір фактів в рамках розділу документу.

$$F_l = \{ f_j: (o_{j,i}, r_{j,k}, s_{j,h}) \in p_l\}. \quad (2.8)$$

де o_i – це об'єкт, r_k – це відношення, s_h - це суб'єкт

Узагальнене представлення правила можна визначити як:

$$G = \{\, g_z(f_j, f_m) \,\}, \tag{2.9}$$

$$g_z(f_j, f_m) = (s_{j,i}, r_{j,k}, s_{m,h}). \tag{2.10}$$

де f_j, f_m – це факти, r_k – це відношення, s_h - це суб'єкт

Загальна кількість правил тоді буде визначатися як:

$$G = G^{(1)} \cup G^{(2)}. \tag{2.11}$$

де $G^{(1)}, G^{(2)}$ – це множини правил першого та другого типів

Крок 2.3 Формування множини розміщень фактів. Множина розміщень по два та три елемента у кортежі буде мати вигляд:

$$A_1 = \{(f_j, f_m) \in p_l \times p_l\}, \tag{2.12}$$

$$A_2 = \{(f_j, f_m) \in p_l \times p_l\}, \tag{2.13}$$

$$A_2 = \{(f_j, f_m, f_n) \in p_l \times p_l\}. \tag{2.14}$$

де f_j, f_m, f_n є фактами, q_i є множиною усіх фактів з розділу документу.

Крок 2.4 Формування правил на основі розміщень фактів. Правила представляються у виді диз'юнкту Хорна. Правила першого, другого та третього типів можна визначити так:

$$G^{(1)} = \{g_z^{(1)}(f_j, f_m) : o_{j,i} = o_{m,i}\,, r_{j,k} = r_{m,k}\}, \tag{2.15}$$

$$G^{(2)} = \{g_z^{(2)}(f_j, f_m) : o_{j,i} = o_{m,i}\,, r_{j,k} \neq r_{m,k}\}, \tag{2.16}$$

$$G^{(3)} = \{g_z^{(3)}(f_j, f_m, f_n) : o_{j,i} = o_{m,i}\,, o_{j,i} = o_{n,i}, s_{j,h} \neq s_{n,h}, s_{j,h} \neq s_{m,h}, r_{j,k} \neq r_{m,k} \neq r_{n,k}\}. \tag{2.17}$$

де u – це консеквент правила, він є фактом, що виходить з інших фактів, x, y – це антецедент правила, факти, що є причиною u.

Ми говоримо, що якщо o_j співпадає з s_j, то таке правило можна вважати обмеженням і його вага буде ∞.

Етап 3 Розрахунок ваг правил

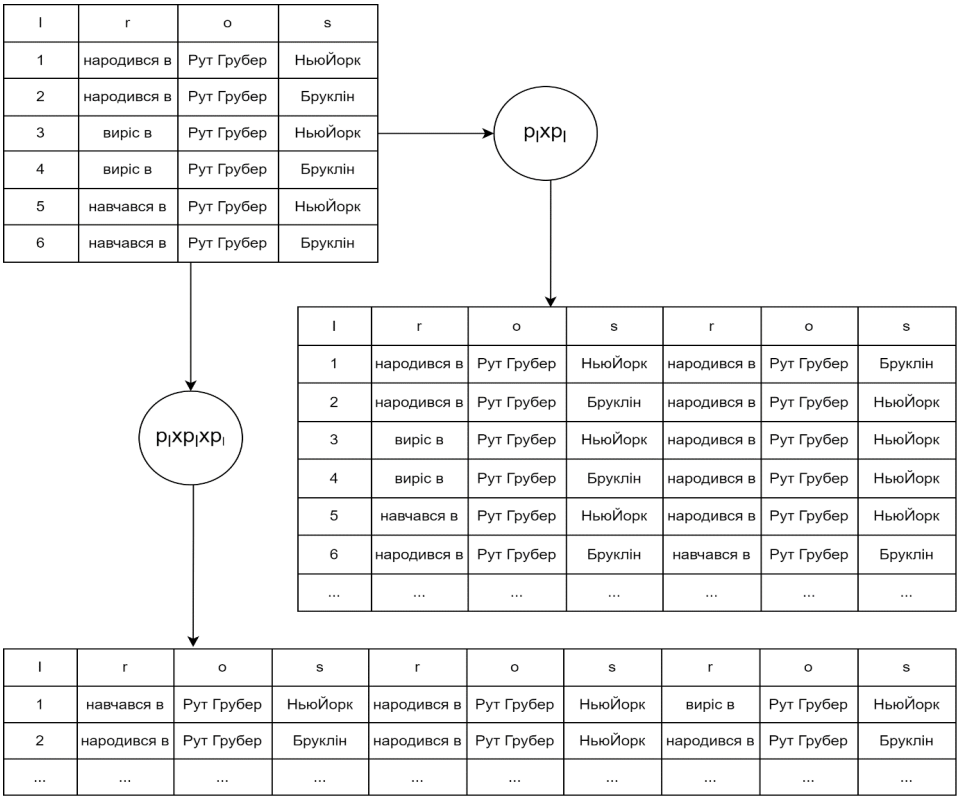
Крок 3.1 Формування факторного графу

Крок 3.2 Розрахунок розподілення за методом Гіббсу

Крок 3.3 Формування зважених правил

Крок 3.4 Упорядкування правил за вагою

Результатом роботи є удосконалений метод. Він відрізняється тим, що було змінено другий етап. Другий етап удосконаленого методу включає в себе генерацію правил з вилучених фактів. Удосконалений метод має переваги: метод надає можливість в автоматизованому режимі формувати нові правила з кожним новим документом.



l	r	o	s
1	народився в	Рут Грубер	НьюЙорк
2	народився в	Рут Грубер	Бруклін
3	виріс в	Рут Грубер	НьюЙорк
4	виріс в	Рут Грубер	Бруклін
5	навчався в	Рут Грубер	НьюЙорк
6	навчався в	Рут Грубер	Бруклін

l	r	o	s	r	o	s
1	народився в	Рут Грубер	НьюЙорк	народився в	Рут Грубер	Бруклін
2	народився в	Рут Грубер	Бруклін	народився в	Рут Грубер	НьюЙорк
3	виріс в	Рут Грубер	НьюЙорк	народився в	Рут Грубер	НьюЙорк
4	виріс в	Рут Грубер	Бруклін	народився в	Рут Грубер	НьюЙорк
5	навчався в	Рут Грубер	НьюЙорк	народився в	Рут Грубер	НьюЙорк
6	народився в	Рут Грубер	Бруклін	навчався в	Рут Грубер	Бруклін
...	...	...	...	...	...	...

l	r	o	s	r	o	s	r	o	s
1	навчався в	Рут Грубер	НьюЙорк	народився в	Рут Грубер	НьюЙорк	виріс в	Рут Грубер	НьюЙорк
2	народився в	Рут Грубер	Бруклін	народився в	Рут Грубер	НьюЙорк	народився в	Рут Грубер	Бруклін
...	...	...	...	...	...	...	...	...	...

Рисунок 2.3 – Приклад формування правил з множини розміщень

3 ТЕХНОЛОГІЯ ПОБУДОВИ БАЗ ЗНАНЬ ДЛЯ ДОВІДКОВИХ СИСТЕМ

Для автоматизованої побудови баз знань для інформаційних довідкових систем пропонується технологія, схема функціональної діаграми якої представлена на рисунку 3.3.

У якості вхідної інформації використовується інформація з документів та інтернет-джерел, що відображаюсь сутності та зв’язки між ними.

На першому етапі виконується вилучення знань з інтернет-джерел. Існуючі системи вилучення знань автоматично вилучають сутності, факти та правила з Інтернету. Через шум корпусу та імовірнісну природу алгоритмів навчання більшість із цих витягів є невизначеними.

На другому етапі застосовуємо удосконалений метод.

Усі вилучені факти використовуємо для розміщення по 2 і по 3 елемента у кортеж. Припускаємо, що отримані кортежі є правилами з невеликою вагою.

Розглядаємо лише правила, що можуть бути записані через диз’юнкт Хорна. Кожен тип правила і має таблицю Rules, в якій записуються предикати, залучені до правил цього типу. Для кожного правила ми маємо кортеж у Rules.

У цих таблицях записуються лише предикати. Порядок аргументів задається правилами.

Більші правила можуть викликати проблеми, оскільки кількість шаблонів правил експоненціально зростає з розміром правила, що робить непрактичним створення таблиці для кожного з них.

Далі виконується класифікація сутностей.

Будується факторний граф. Всі факти та правила записуються в базу даних. Завдяки розробленій реляційній моделі побудова факторного графу виконується декількома операціями JOIN на таблицях фактів. Спочатку виконується запит для виводу фактів. Він виконується циклічно, доки не закінчяться факти. Приклад такого запиту зображений на рисунках 3.1 та 3.2.

Далі інший запит генерує факти. Він зображений на рисунку 3.3.

Запит на рисунку 3.3 повертає матрицю факторів. Він виконує поєднання таблиць правил та фактів будуючи матрицю зв'язків між правилами та фактами, що вони об'єднують.

```
SELECT Rules2.Relationship1 AS R,
Relationships.Entity1 AS x, Relationships.Class1 AS C1
Relationships.Entity2 AS y, Relationships.Class2 AS C2
FROM Rules2
JOIN Relationships ON Rules2.Relationship2 = Relationships.Relation
AND Rules2.Class1 = Relationships.Class1
AND Rules2.Class2 = Relationships.Class2
```

Рисунок 3.1 – Запит для виводу фактів

```
SELECT Rules3.Relationship1 AS R,
Relationships2.Entity2 AS x, Relationships2.Class2 AS C1,
Relationships3.Entity2 AS y, Relationships3.Class2 AS C2
FROM Rules3
JOIN Relationships Relationships2
ON Rules3.Relationship2 = Relationship2.Relation AND
Rules3.Class3 = Relationships2.Class1
AND Rules3.Class1 = Relationships2.Class2
JOIN Relationships Relationships3
ON M3.Relationship3 = Relationships3.Relation
AND Rules3.Class3 = Relationships3.Class1
AND Rules3.Class2 = Relationships3.Class2
WHERE Relationships2.x = Relationships3.x
```

Рисунок 3.2 – Запит для виводу фактів

На третьому етапі виконується відсіювання фактів. Разом з усіма витягнутими сутностями та фактами, ЛММ виз кодує розподіл ймовірностей за всіма виведеними фактами. Таким чином, імовірнісний висновок підтримується запитом цього розподілу. Факти з невеликою імовірністю відсіюються.

```
SELECT Relationships1.idRelationships AS I1,
Relationships2.idRelationships AS I2,
Relationships3.idRelationships AS I3,
Rules3.w AS w
FROM Rules3
JOIN Relationships Relationships1 ON
Rules3.Relationship1 = Relationships1.Relation
AND Rules3.Class1 = Relationships1.Class1
AND Rules3.Class2 = Relationships1.Class2
JOIN Relationships Relationships2 ON
Rules3.Relationship2 = Relationships2.Relation
AND Rules3.Class3 = Relationships2.Class1
AND Rules3.Class1 = Relationships2.Class2
JOIN Relationships Relationships3
ON Rules3.Relationship3 = Relationships3.Relation
AND Rules3.Class3 = Relationships3.Class1
AND Rules3.Class2 = Relationships3.Class2
WHERE Relationships1.x = Relationships2.y
AND Relationships1.y = Relationships3.y
AND Relationships2.x = Relationships3.x;
```

Рисунок 3.3 – Запит для генерації факторів

На четвертому етапі виконується вибіркова перевірка експертом. Для цього вибираються випадкові факти для ручної перевірки. Якщо факти вірні, нічого не змінюється. Якщо знайдена помилка, помилковий факт вилучається та виконується рекурсивна побудова факторного графу на основі фактів, що були пов’язані з помилковим. Ваги таких фактів корегуються або вилучаються самі факти.

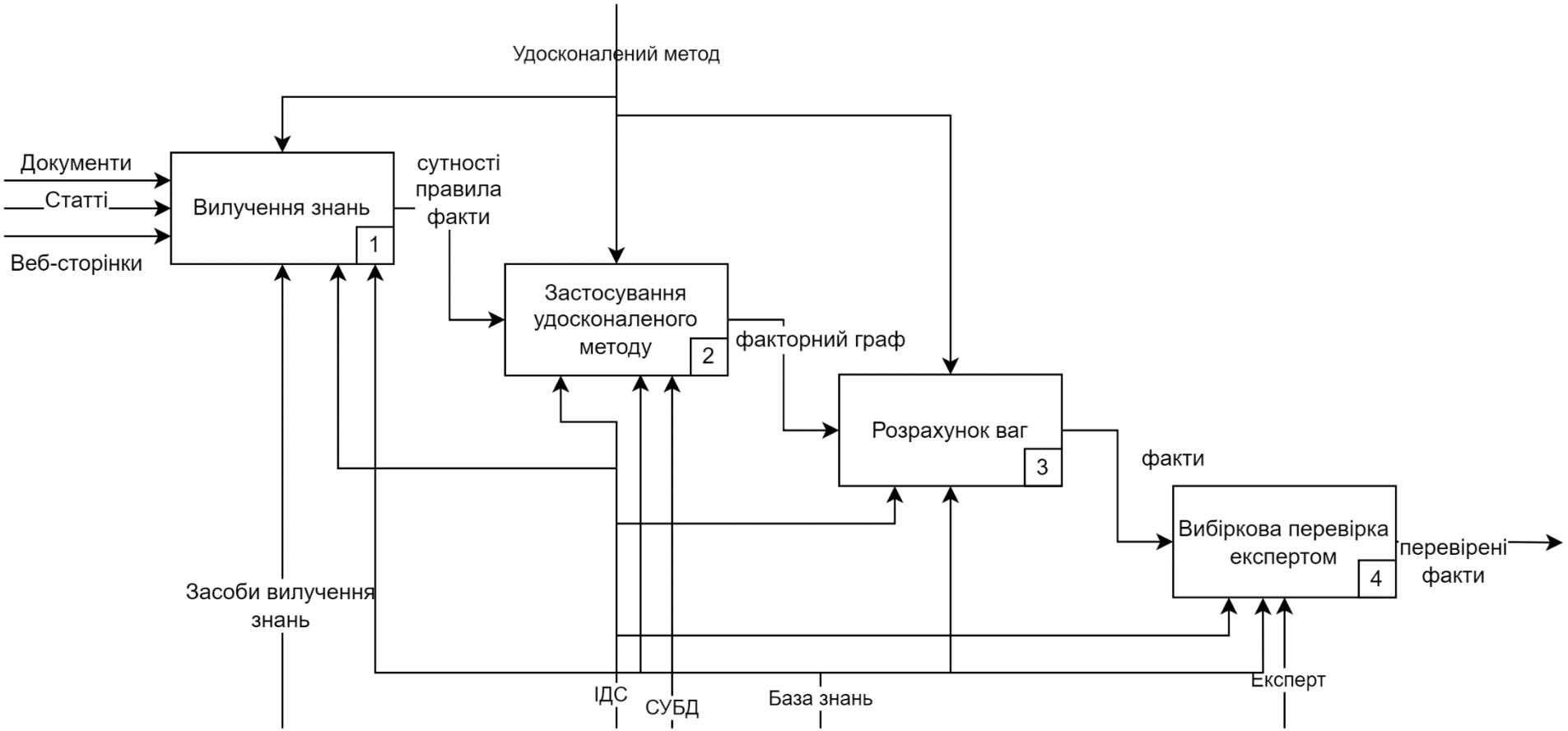
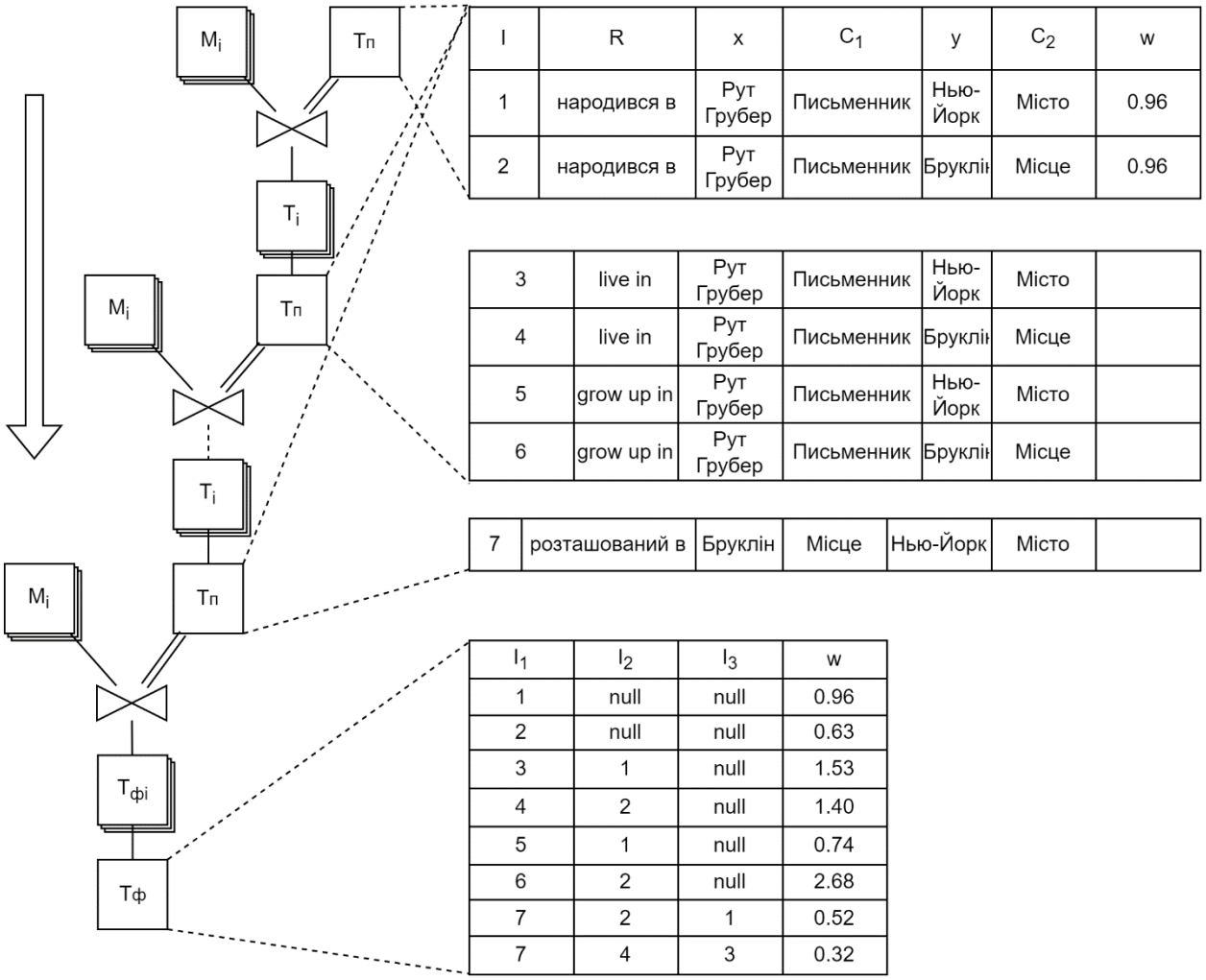


Рисунок 3.4 – схема функціональної діаграми технології для автоматизованої побудови баз знань для інформаційних довідкових систем

4 ПРАКТИЧНЕ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Розробка програмного засобу автоматизованої побудови зважених правил для баз знань в інформаційно-довідкових системах



I	R	x	C_1	y	C_2	w
1	народився в	Рут Грубер	Письменник	Нью-Йорк	Місто	0.96
2	народився в	Рут Грубер	Письменник	Бруклін	Місце	0.96
3	live in	Рут Грубер	Письменник	Нью-Йорк	Місто	
4	live in	Рут Грубер	Письменник	Бруклін	Місце	
5	grow up in	Рут Грубер	Письменник	Нью-Йорк	Місто	
6	grow up in	Рут Грубер	Письменник	Бруклін	Місце	
7	розташований в	Бруклін	Місце	Нью-Йорк	Місто	

I_1	I_2	I_3	w
1	null	null	0.96
2	null	null	0.63
3	1	null	1.53
4	2	null	1.40
5	1	null	0.74
6	2	null	2.68
7	2	1	0.52
7	4	3	0.32

Рисунок 4.1 – Процес побудови факторного графу

Таблиця 4.1 – Базові факти

I	R	x	C_1	y	C_2	w
1	народився в	Рут Грубер	Письменник	New York	Місто	0.96
2	народився в	Рут Грубер	Письменник	Бруклін	Місце	0.93

Таблиця 4.2 – Правила першого типу

R_1	R_2	C_1	C_2	w
жив в	народився в	Письменник	Місце	1.40
жив в	народився в	Письменник	Місто	1.53
виріс в	народився в	Письменник	Місце	2.68
виріс в	народився в	Письменник	Місто	0.74

Таблиця 4.3 – Правила третього типу

R_1	R_2	R_3	C_1	C_2	C_3	w
знаходиться в	жив в	жив в	Місце	Місто	Письменник	0.32
знаходиться в	народився в	народився в	Місце	Місто	Письменник	0.52

Для описаного методу необхідно представили базу знань як реляційну модель. Це дозволить використовувати функціонал, що реалізований СУБД, за допомогою мови запитів SQL. Загальна структура БД представлена на рисунку 4.2.

Таблиця Entities містить у собі інформацію про сутності, а саме ідентифікатор сутності та її клас.

Таблиця Classes містить інформацію про класи в системі. У кожного класа є ідентифікатор та назва.

Таблиця Relations містить інформацію про можливі відношення. Наприклад:

$$\text{народився в}(W, P). \qquad (4.1)$$

де W є ідентифікатором класу «Письменник», а P є ідентифікатором класу

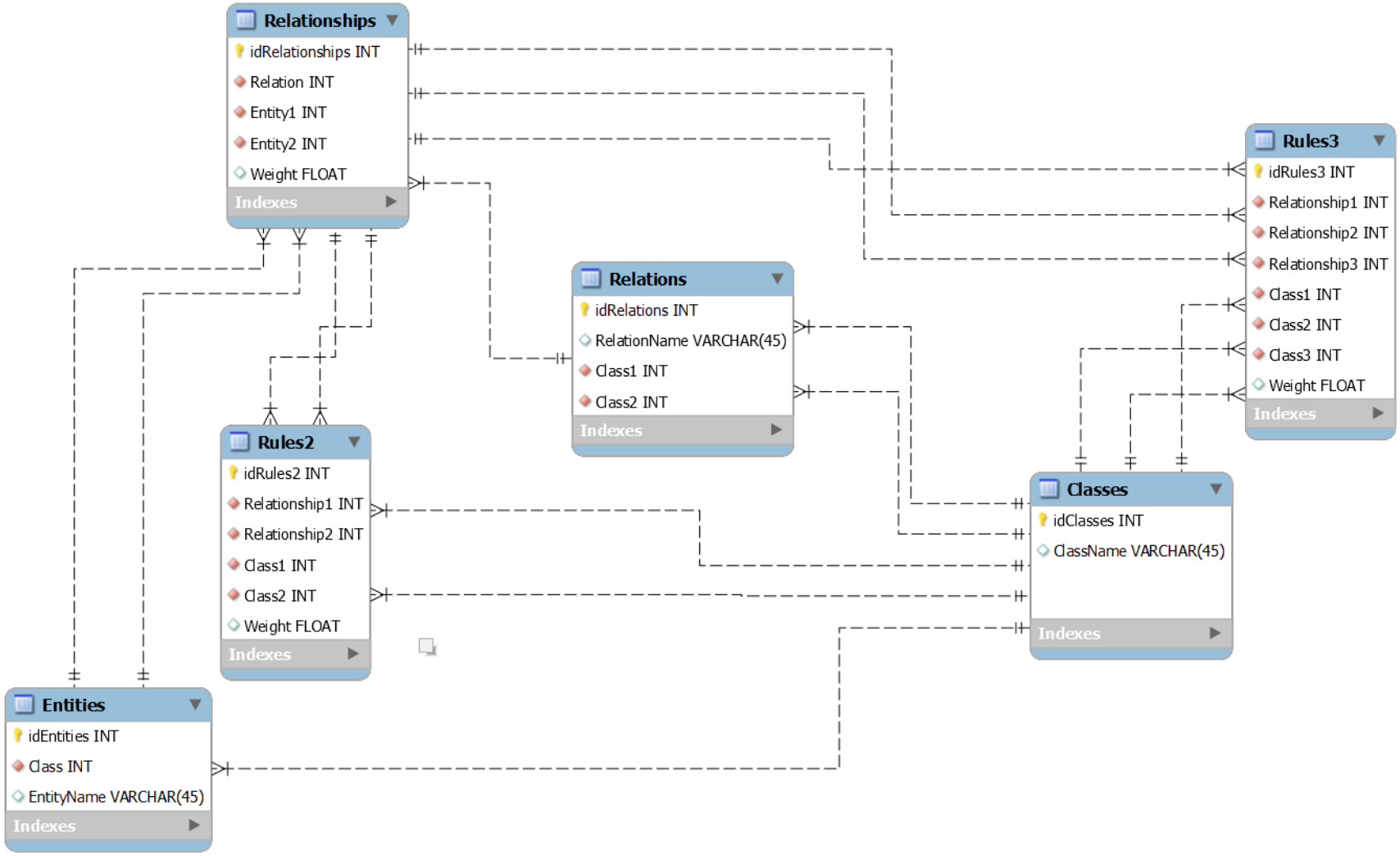


Рисунок 4.2 – Схема реляційної моделі бази знань.

«Місце». Таблиця Relations відображає факти та можливий зв’язок між класами сутностей. Таблиця мстить ідентифікатори класів, що можуть бути застосовані для відображення цього зв’язку.

Таблиця Relationships містить зважену інформацію про факти. У ній містяться ідентифікатор відношення, яке факт відображає та ідентифікатори конкретних сутностей, які беруть участь у факті. Також у цій таблиці є вага, що відображає правдивість факту, що записаний Наприклад:

$$0{,}95\ \text{народився в}(\text{Рут Грубер}, \text{Нью} - \text{Йорк})\ . \qquad (4.2)$$

Таблиця Rules містить інформацію про правила виведення. Таблиця містить атрибути: ідентифікатори фактів, ідентифікатори класів та вагу. Ідентифікатори класів та фактів відображають, які класи та факти беруть участь у правилі. Вага відображає ступінь коректності правила.

Наприклад:

$$1.40\ \forall x \in W\ \forall y \in P\ (\text{жив в}(x, y) \leftarrow \text{народився в}(x, y)). \qquad (4.3)$$

Елемент Rules не є єдиною таблицею. Для кожного з 6 можливих типів правил є своя таблиця. Можливі типи правил представлені нижче:

$$\forall x \in C1, y \in C2\ \big(p(x, y) \leftarrow q(x, y)\big), \qquad (4.1.4)$$

$$\forall x \in C1, y \in C2\ \big(p(x, y) \leftarrow q(y, x)\big), \qquad (4.1.5)$$

$$\forall x \in C1, y \in C2, z \in C3\ \big(p(x, y) \leftarrow q(z, x), r(z, y)\big), \qquad (4.1.6)$$

$$\forall x \in C1, y \in C2, z \in C3\ \big(p(x, y) \leftarrow q(x, z), r(z, y)\big), \qquad (4.1.7)$$

$$\forall x \in C1, y \in C2, z \in C3\ \big(p(x, y) \leftarrow q(z, x), r(y, z)\big), \qquad (4.1.8)$$

$$\forall x \in C1, y \in C2, z \in C3\ (p(x, y) \leftarrow q(x, z), r(y, z)). \qquad (4.1.9)$$

Тож кортежі для кожного з 6 правил будуть відповідно:

$$M_{1-2} = (R1, R2, C1, C2, w), \quad (4.4)$$

$$M_{3-6} = (R1, R2, R3, C1, C2, C3, w). \quad (4.5)$$

Нові сутності та факти, що вилучаються СВЗ, записуються в окремі таблиці, щоб не змішуватися з затвердженими знаннями. Коли робиться оцінка нових фактів, якщо вона задовольняє умовам, нові факти переносяться з тимчасової таблиці до основної.

Представлена реляційна модель описує представлення бази знань. Знання у базі знань представлені з використанням формально-логічної моделі.

Вона містить сутності, що відображають усі об'єкти, що з'являються при вилученні інформації за допомогою СВЗ. Класи представлені як множини, елементами яких є сутності. Зв'язки відображають шаблони фактів та описують зв'язок та які класи використовуються у цих фактах. Взаємозв'язки відображають конкретні факти. Правила відображають правила виведення.

На етапі формування факторного графу виконуються такі запити до бази даних:

Наступні два запита на рисунках повертають об'єднання таблиць за правилом та класом сутності правила. Якщо існує таке правило, де $M.R_2 = T.R$ та $M.C_2 = T.C_2$, то можна сказати, що факт $M.R_1$ э дійсним для сутностями M.x та M.y. Таким чином збираються усі факти, що є у БД та ті, що можуть бути виведені за допомогою правил, що є в БД.

```
SELECT Rules2.Relationship1 AS R,
Relationships.Entity1 AS x, Relationships.Class1 AS C1
Relationships.Entity2 AS y, Relationships.Class2 AS C2
FROM Rules2
JOIN Relationships ON Rules2.Relationship2 = Relationships.Relation
AND Rules2.Class1 = Relationships.Class1
AND Rules2.Class2 = Relationships.Class2
```

Рисунок 4.3 – Запит, що виводить факти з правил першого типу

```
SELECT Rules3.Relationship1 AS R,
Relationships2.Entity2 AS x, Relationships2.Class2 AS C1,
Relationships3.Entity2 AS y, Relationships3.Class2 AS C2
FROM Rules3
JOIN Relationships Relationships2
ON Rules3.Relationship2 = Relationship2.Relation AND
Rules3.Class3 = Relationships2.Class1
AND Rules3.Class1 = Relationships2.Class2
JOIN Relationships Relationships3
ON M3.Relationship3 = Relationships3.Relation
AND Rules3.Class3 = Relationships3.Class1
AND Rules3.Class2 = Relationships3.Class2
WHERE Relationships2.x = Relationships3.x
```

Рисунок 4.4 – Запит, що виводить факти з правил третього типу

Запит на рисунку 4.5 повертає матрицю факторів. Він виконує поєднання таблиць правил та фактів будуючи матрицю зв’язків між правилами та фактами, що вони об’єднують.

```
SELECT Relationships1.idRelationships AS I1,
Relationships2.idRelationships AS I2,
Relationships3.idRelationships AS I3,
Rules3.w AS w
FROM Rules3
JOIN Relationships Relationships1 ON
Rules3.Relationship1 = Relationships1.Relation
AND Rules3.Class1 = Relationships1.Class1
AND Rules3.Class2 = Relationships1.Class2
JOIN Relationships Relationships2 ON
Rules3.Relationship2 = Relationships2.Relation
AND Rules3.Class3 = Relationships2.Class1
AND Rules3.Class1 = Relationships2.Class2
JOIN Relationships Relationships3
ON Rules3.Relationship3 = Relationships3.Relation
AND Rules3.Class3 = Relationships3.Class1
AND Rules3.Class2 = Relationships3.Class2
WHERE Relationships1.x = Relationships2.y
AND Relationships1.y = Relationships3.y
AND Relationships2.x = Relationships3.x;
```

Рисунок 4.5 – Запит, що будує матрицю факторного графу

Схема алгоритму роботи програми представлена на рисунку 4.6. Декомпозиція схеми алгоритму роботи програми представлена на рисунку 4.7.

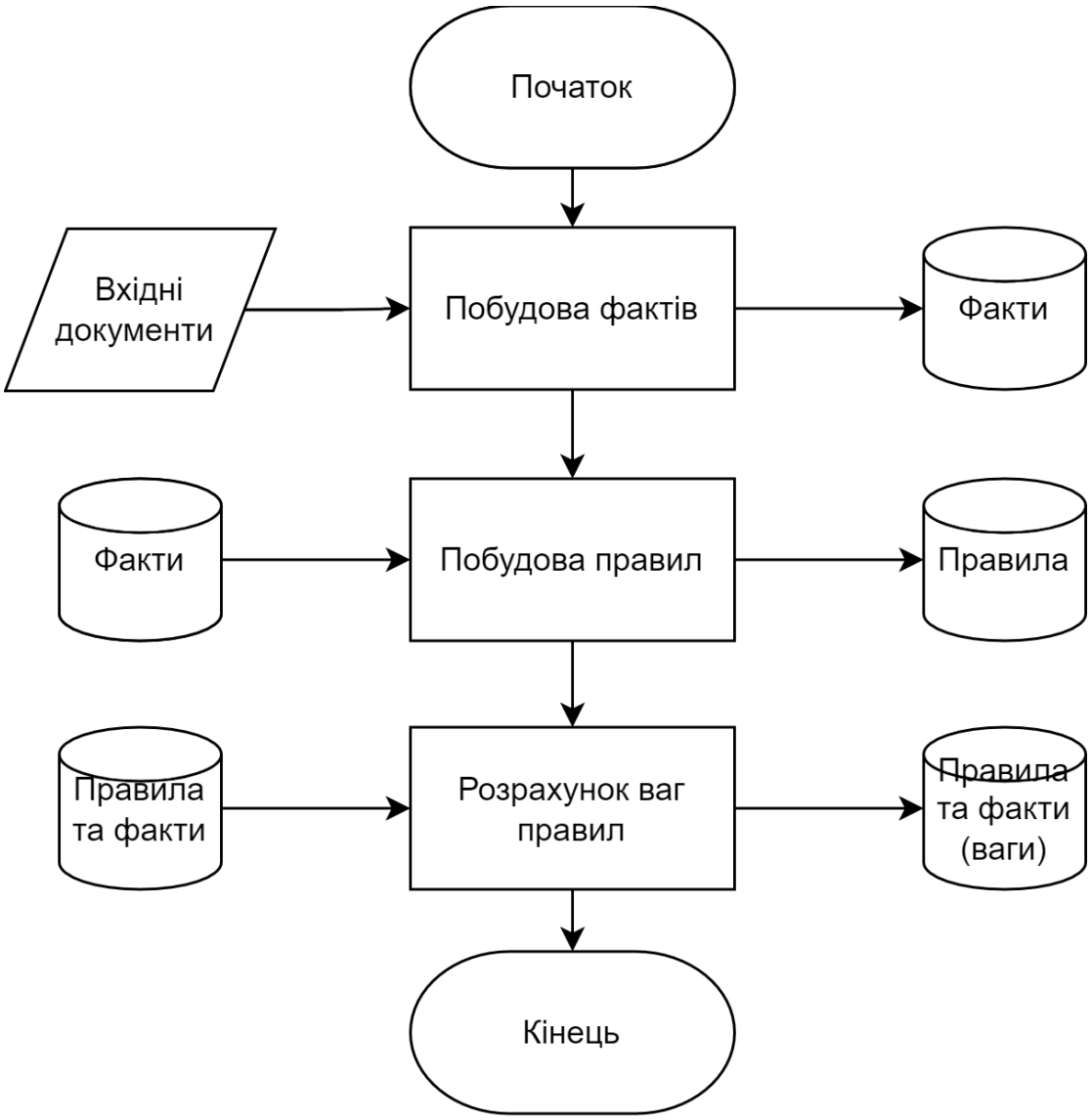


Рисунок 4.6 – Схема загального алгоритму удосконаленого методу

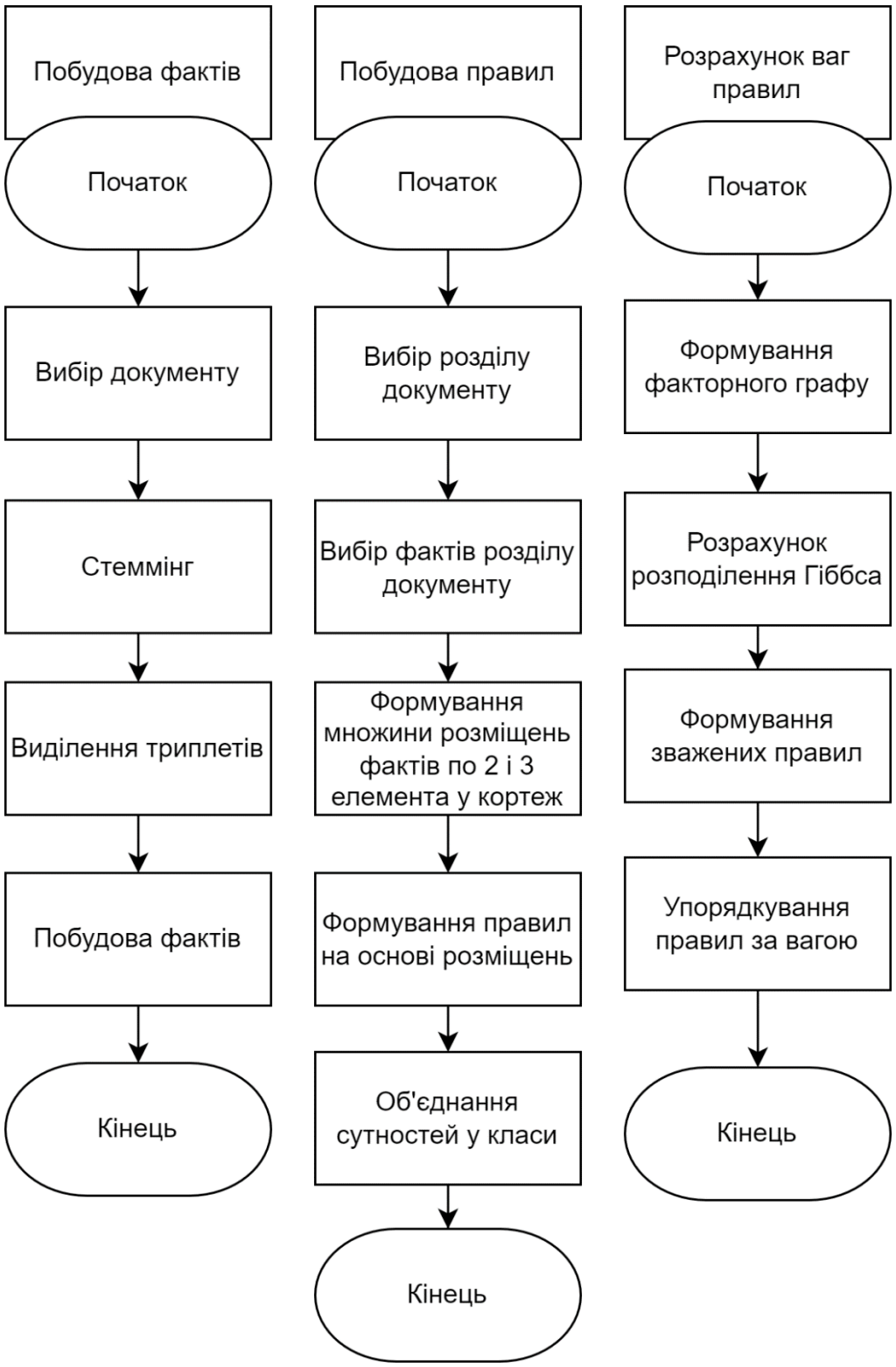


Рисунок 4.7 – Декомпозиція схеми загального алгоритму удосконаленого методу

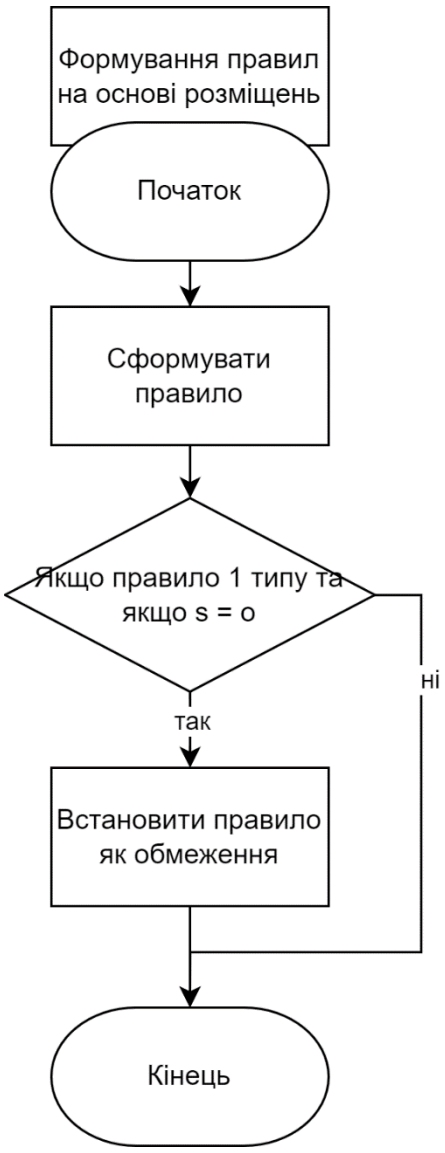


Рисунок 4.8 – Декомпозиція схеми «Формування правил на основі розміщень»

На рисунку 4.9 представлена екранна форма на етапі вилучення знань. Вона відображає поточні результати вилучення знань з джерел. На ній можна ініціювати додавання джерела та їх аналізу. На рисунку 4.10 представлена екранна форма етапу визначення типів відношень. На ній можна відсортувати правила та обрати вибрати необхідні. На цій формі можна ініціювати редагування обраних правил.

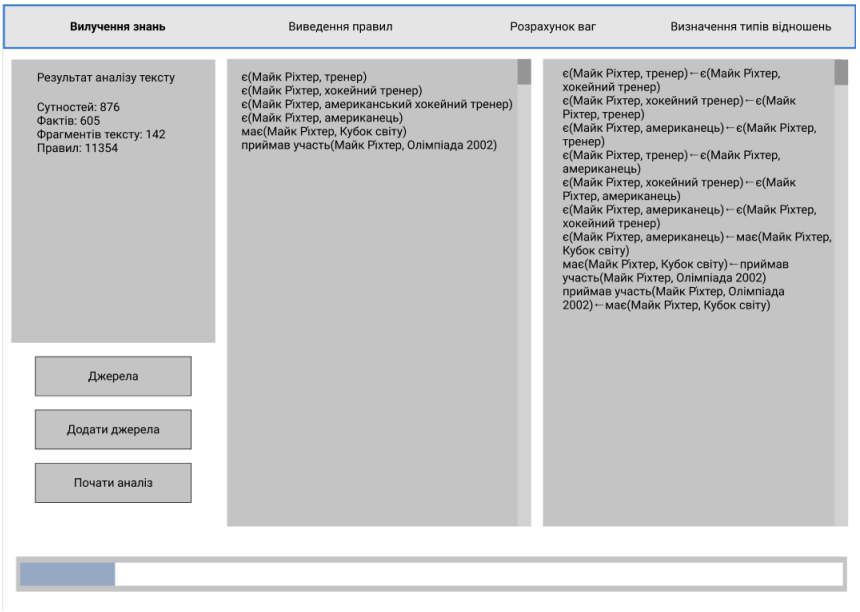


Рисунок 4.9 – Екранна форма на етапі вилучення знань

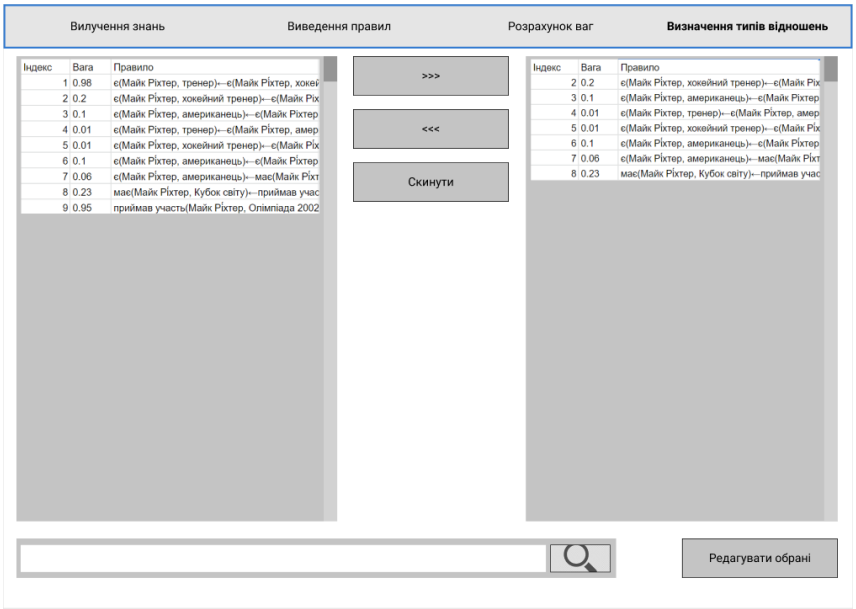


Рисунок 4.10 – Екранна форма на етапі визначення типів відношень

Таблиця 4.4 – Модель представлення знань

Сутності	Класи	Зв’язки	Взаємозв’язки
Рут Грубер	W (Письменник) = {Рут Грубер}	народився в(W, P), народився в(W, C)	народився в(Рут Грубер, Нью-Йорк)
Нью-Йорк	C (Місто) = {Нью-Йорк}	жив в(W, P), жив в(W, C)	народився в(Рут Грубер, Бруклін)
Бруклін	P (Місце) = {Бруклін}	знаходиться в(P, C)	
Правила			
$\forall x \in W\ \forall y \in P$ (жив в(x, y) ← народився в(x, y))			
$\forall x \in W\ \forall y \in C$ (жив в(x, y) ← народився в(x, y))			
$\forall x \in P\ \forall y \in C\ \forall z \in W$ (знаходиться в(x, y) ← жив в(z, x) ∧ жив в(z, y))			
$\forall x \in P\ \forall y \in C\ \forall z \in W$ (знаходиться в(x, y) ← народився в(z, x) ∧ народився в(z, y))			
$\infty\ \forall x \in C\ \forall y \in C\ \forall z \in W$ (народився в(z, x) ∧ народився в(z, y) → x = y)			

4.2 Експериментальна перевірка методу

При порівнянні базового та удосконаленого методів були взяті такі показники: кількість вилучених фактів, кількість вилучених правил. На рисунку 4.11 представлена діаграма порівняння результатів з кількості вилучення фактів у системі ProbKB та при використані удосконаленого методу.

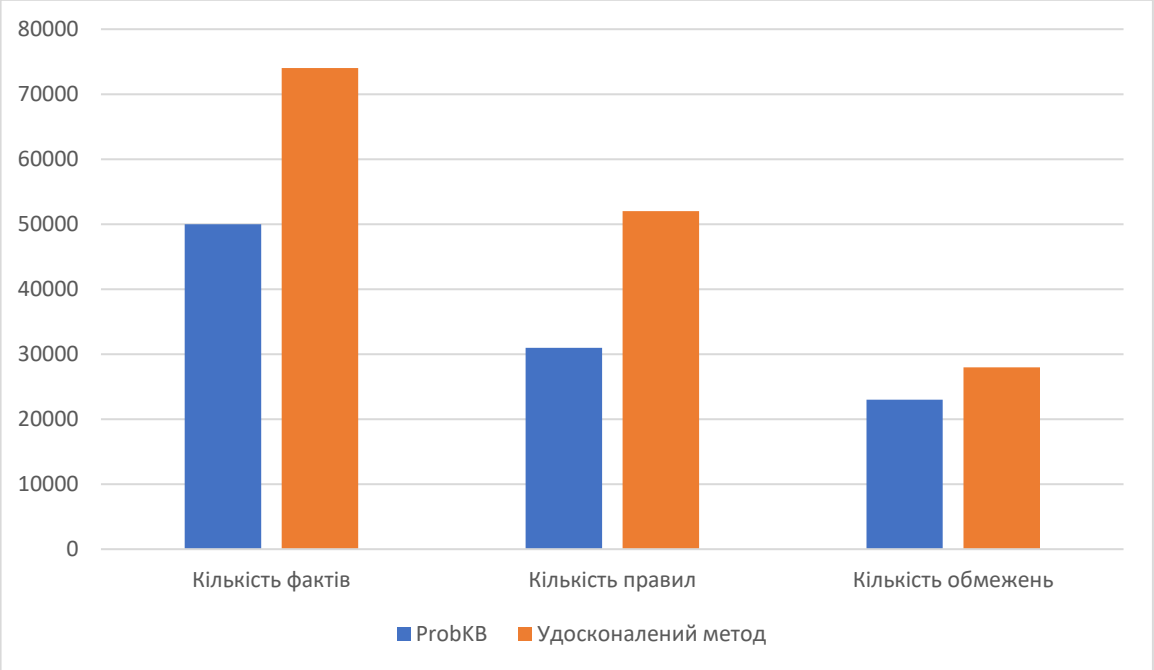


Рисунок 4.11 – Діаграма порівняння методів ProbKB та удосконаленого методу автоматизованої побудови зважених правил.

Діаграма відображає, що при використанні удосконаленого методу використовувалась більша кількість правил та обмежень для виведення, ніж кількість правил у сеті, що використовує система ProbKB. Також, завдяки більшій кількості правил та обмежень спостерігається більша кількість фактів, що були виведені під час роботи удосконаленого методу.

Отже, удосконалений метод показав кращі результати у кількості вилучених фактів, ніж ProbKB. Удосконалений метод виводить правила та обмеження для виведення фактів із текстів, що аналізуються. Це дозволяє збирати усі правила, що можуть бути застосовані до цих фактів, під час аналізу тексту, що в свою чергу, збільшує кількість фактів, що можуть бути виведені в майбутньому.

ВИСНОВКИ

Інформаційно-довідкові системи мають покривати великий обсяг інформації. Для цього необхідно працювати з великою базою знань. Але мануальний процес побудови такої бази знань є неможливим через великий обсяг даних. Тому доцільно розробляти, удосконалювати та використовувати автоматизовані методи побудови баз знань для інформаційно-довідкових систем.

Під час виконання магістерської кваліфікаційної роботи було проаналізовано роботу, призначення та структуру інформаційно-довідкових систем. Були проаналізовані методи представлення знань у базах знань інформаційно-довідкових систем. Розглянута класифікація баз знань та їх основні моделі: семантична, продукційна, фреймова, формально-логічна. Виконане дослідження роботи існуючих інформаційних довідкових систем з методами автоматизованої побудови баз знань. А саме розглянута архітектура та процеси таких систем, як DeepDive та YAGO. На основі дослідження першого розділу була сформована задача атестаційної роботи.

Був проаналізований метод автоматизованої побудови бази знань на основі факторного графу. Метод полягає у тому, щоб будувати факторний граф для розрахунку імовірнісного розподілення за допомогою SQL запитів. Для розрахунку розподілення використовувалося розподілення Гіббса. Розглянутий метод використовує статичний сет правил. Це обмежує можливості висновку інформаційно-довідкової системи на основі цього методу.

Удосконалено метод автоматизованої побудови зважених правил для інформаційно-довідкової системи на основі використання триплетів «об’єкт, відношення, суб’єкт» шляхом виділення залежностей між суб’єктами, для яких співпадає об’єкт та може співпадати відношення між об’єктом та суб’єктом. Ваги правил визначаються частотою виявлення однакових правил в тексті.

Удосконалений метод забезпечує можливість побудови правил безпосередньо з тексту документів без використання типових відношень, що були виведені під час аналізу тексту разом з фактами та зв’язками.

Удосконалений метод реалізовано у рамках інформаційної технології побудови правил. Технологія містить етапи: аналіз тексту; застосування удосконаленого методу; очищення правил та вибіркову перевірку експертом.

Для описаної технології розроблено програмне забезпечення мовою C#. З використанням розробленого програмного забезпечення був проведений експеримент та виконаний порівняльний аналіз системи з використанням удосконаленого методу та методу на основі факторного графу. По результатам експерименту було виявлено, що удосконалений метод може виявляти більше правил, ніж використовується в базовому, а також може виводити більше фактів.

Результати магістерської роботи представлені у матеріалах 25-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» 2021р.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Інформаційні управляючі системи та технології») / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 25 с.

2. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения / Межгосударственный стандарт. М. : Издательство стандартов, 1991. 26 с.

3. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення / Державний стандарт України. Київ : Державний комітет України по стандартизації, метрології та сертифікації, 1996. 29с.

4. https://medium.com/@CereLabs/helplessness-of-software-d83c984ac6a7 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://medium.com/@CereLabs/helplessness-of-software-d83c984ac6a7.

5. YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames / T.Rebele, F. Suchanek, J. Hoffart, J. Biega. // Lecture Notes in Computer Science. – 2016.

6. Brachman R. J. KNOWLEDGE REPRESENTATION AND REASONING / R. J. Brachman, H. J. Levesque. – San Francisco: Elsevier, 2004. – 413 с.

7. Knowledge Base Construction [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@CereLabs/knowledge-base-construction-146177844cc6.

8. https://en.wikipedia.org/wiki/Never-Ending_Language_Learning [Електронний ресурс] – Режим доступу до ресурсу: Never-Ending Language Learning.

9. Ratner A. DeepDive: Declarative Knowledge Base Construction [Електронний ресурс] / A. Ratner, J. Shin, F. Wang. – 2015. – Режим доступу до ресурсу: https://sigmodrecord.org/publications/sigmodRecord/1603/pdfs/16_DeepDive_RH_DeSa.

10. The Dark Data Rises [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@CereLabs/the-dark-data-rises-f911422d80fc.

11. Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction – 209 N. Eighth Street, 2012. – 139 с.

12. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. – Київ, 2006. – 378 с.

13. Chen Y. Knowledge expansion over probabilistic knowledge bases / Y. Chen, D. Zhe Wang. // SIGMOD '14. – 2014.

14. Elementary: Large-scale Knowledge-base Construction via Machine Learning and Statistical Inference. // Semantic Web and Information Systems - Special Issue on Web-Scale Knowledge Extraction. – 2012. – С. 23.

15. Milton N. Knowledge Acquisition in Practice: A Step-by-step Guide / Nicholas Milton. // Knowledge Acquisition in Practice. – 2007. – С. 43.

16. Ji S. A Survey on Knowledge Graphs: Representation, Acquisition and Applications / S. Ji, E. Cambria, P. Marttinen. // JOURNAL OF LATEX CLASS FILES. – 2015. – №8. – С. 27.

17. Rudin F. Falling Rule Lists / F. Rudin, C. Wang. // JOURNAL OF LATEX CLASS FILES. – С. 10.

18. Martinez-Gil J. Automated knowledge base management: A survey / Jorge Martinez-Gil. // Computer Science Review. – 2015.

19. Suchanek F. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia / F. Suchanek, G. Kasneci, G. Weikum. // Track: Semantic Web. – 2007.

20. R. W. Blanning, S. Ram, and R. Y. Wang, «Information technologies and systems», Decision support systems, vol. 13, no. 3–4, pp. 219–221, 1995, doi: 10.1016/0167-9236(93)E0043-D.

21. H. Mintzberg, D. Raisinghani, and A. Theoret, «The structure of «Unstructured» decision processes», Administrative science quarterly, vol. 21, no. 2, pp. 246-248, 1976, doi: 10.2307/2392045.

22. Т.В. Козуля, Н.В. Шаронова, М.М. Козуля, « Формування знанняорієнтованого інформаційного забезпечення досліджень складних 327 систем», Системні дослідження та інформаційні технології, №. 7, с. 63– 72, 2017, https://doi.org/10.20535/SRIT.2308-8893.2017.3.07.

23. H. Uehara, T. Yamaguchi, and Q. Bai, Knowledge management and acquisition for intelligent systems in 17th Pacific rim knowledge acquisition workshop, PKAW 2020, Yokohama, Japan, 2021. Cham: Springe

24. T. H. Davenport и L. Prusak, Working knowledge: How organizations manage what they know. Boston, MA: Harvard business school press, 2000.

25. F. M. Suchanek, J. Lajus, A. Boschin, and G. Weikum, «Knowledge representation and rule mining in entity-centric knowledge bases», in Reasoning Web. Explainable Artificial Intelligence, no. 11810, M. Krötzsch 330 and D. Stepanova, Eds. Cham: Springer International Publishing, 2019, pp. 110–152. doi: 10.1007/978-3-030-31423-1_4.

26. R.K. Bali, N. Wickramasinghe, and B. Lehaney, Knowledge management primer. New York: Routledge, 2009.

27. P. R. Gamble and J. Blackwell, Knowledge management: A state of the art guide. Kogan Page Ltd, 2001

28. S. Wang and D. Liu, «Knowledge representation and reasoning for qualitative spatial change», Knowledge-Based Systems, vol. 30, pp. 161–171 2012, doi: 10.1016/j.knosys.2012.01.009

29. I. Hatzilygeroudis and J. Prentzas, «Knowledge representation requirements for intelligent tutoring systems», in ITS 2004. International Conference on Intelligent Tutoring Systems. Berlin, Heidelberg, 2004, vol. 3220, pp. 87–97.

30. C. Ramirez and B. Valdes, «A general knowledge representation model of concepts», Advances in Knowledge Representation, C. Ramirez, Eds. InTechOpen, 2012. doi: 10.5772/37113

31. Q. Wang, Z. Mao, B. Wang, and L. Guo, «Knowledge Graph Embedding: A Survey of Approaches and Applications», in IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 12, pp. 2724–2743, 2017, doi: 10.1109/TKDE.2017.2754499.

32. A. Felfernig and F. Wotawa, «Intelligent engineering techniques for knowledge bases», Ai Communications, vol. 26, no. 1, pp. 1–2, 2013, doi: 10.3233/AIC-2012-0541.