НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра системного проектування

«На правах рукопису»
УДК _____

До захисту допущено:
Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інтелектуальні сервіс-орієнтовані розподілені обчислювання»

зі спеціальності 122 "Комп'ютерні науки"

на тему: «Використання семантики і формалізованих знань в інтелектуальній обробці даних»

Виконав: студент VI курсу, групи ДА-91мп
Бражник Максим Русланович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник к.т.н., доцент Булах Б.В _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Розробка стартап-проекту к.т.н., доцент Булах Б.В. _____
(назва розділу) (посада, науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент к.т.н., доцент Тимощук О.Л. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Інститут/факультет Інституту прикладного системного аналізу
(повна назва)

Кафедра Системного проектування
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною (освітньо-науковою) програмою

Спеціальність (спеціалізація) Комп'ютерні науки
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
___ А.І.Петренко
(підпис) (ініціали, прізвище)
«___»_____202_р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Бражник Максим Русланович
(прізвище, ім'я, по батькові)

1. Тема дисертації: Використання семантики і формалізованих знань в інтелектуальній обробці даних
науковий керівник дисертації Булах Богдан Вікторович к.т.н. ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «02» Листопада 2020 р. № 3182-с

2. Строк подання студентом дисертації - 15 грудня, звіт з практики – 25 жовтня

3. Об'єкт дослідження: методи семантичної обробки даних з використання формалізації інформації

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою): реалізація методів семантичного аналізу та формування формалізованої інформації

5. Перелік завдань, які потрібно розробити
1. Підходи до отримання знань в семантичній мережі
2. Технології семантичного data mining

3. Приклади використання інформації з семантичної мережі
4. Розробка прототипу системи ІАД із застосуванням семантики

6. Перелік графічного (ілюстративного) матеріалу: презентація на тему «Використання семантики і формалізованих знань в інтелектуальній обробці даних»

7. Орієнтовний перелік публікацій _____
_____

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Реалізація стартап-проекту	Булах Б.В., доцент		

9. Дата видачі завдання 7 липня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	7 липня 2020	
1	Підходи до отримання знань в семантичній мережі	10 вересня 2020	
2	Технології семантичного data mining	15 вересня 2020	
3	Приклади використання інформації з семантичної мережі	20 вересня 2020	
4	Розробка прототипу системи ІАД із застосуванням семантики	1 жовтня 2020	
5	Розроблення стартап-проекту	25 жовтня 2020	
6	Оцінка результатів та опис практичних рекомендацій	15 листопада 2020	
7	Оформлення дипломної роботи	15 грудня 2020	

Студент _____ М. Р. Бражник
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Б. В. Булах
(підпис) (ініціали, прізвище)

РЕФЕРАТ
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему “ Використання семантики і формалізованих знань в інтелектуальній обробці даних”
студентом Бражником Максимом Руслановичом

Робота виконана на 99 сторінках, містить 25 ілюстрації, 22 таблиці. При підготовці використовувалася література з 31 джерела.

Актуальність. З появою інтернету підхід до технологій кардинально змінився. Насьогодні у всесвітній мережі зберігається велика кількість інформації. Такі масиви даних надзвичайно важко обробляти ручними способами, а з зростаннями ціни на робочу силу, це стає практично неможливо. Зараз набирають великої популярності технології структурування інформації в інтернеті для подальшої машинної обробки. До таких можна перечислити семантичний веб, структуризація за допомогою тегів тощо. Такі підходи до зберігання інформації дозволили застосовувати методи класифікації та кластеризації даних, що зможуть допомогти людині під час роботи або навіть замінити та автоматизувати весь робочий процес.

Мета. Метою даної роботи є дослідження сучасних методів отримання інформації з мережі та подальшим використанням цих даних для кластеризації та видобутку корисної інформації. Це допоможе автоматизувати робочі процеси під час декількох кроків: пошуку структурованої інформації та її наступним використанням методами data mining.

Завдання. Для досягнення поставленої мети необхідно розв’язати наступні завдання:

- проаналізувати існуючі підходи збору структурованої інформації в інтернеті;
- детально розібрати найбільш доцільні методи збору;
- проаналізувати успішний досвід реалізації програмного забезпечення іноземними колегами в даній сфері;

- розробити програмний продукт, що дозволить вирішити проблеми інженерів під час збору інформації в інтернеті, а також надати інструментарій для аналізу та видобутку корисної інформації;
- розробити стратегію стартап-проекту, яка дозволить реалізувати описану технологію в якості конкурентоспроможного продукту.

Об’єкт дослідження. Семантика та структурована інформація.

Предмет дослідження. Взаємодія зі структурованою інформацією та її аналіз.

Наукова новизна. Наукова новизна роботи полягає в дослідженні способів поєднання технологій семантичної мережі та методів інтелектуального аналізу даних, для отримання нових високоякісних процедур аналізу даних.

Практична цінність. Практична цінність роботи полягає у аналізі методів та засобів аналізу структурованої інформації з інтернету, розробка веб-додатку як приклад використання.

Публікації.

Бражник М. Р. Використання семантичних та формалізованих знань в інтелектуальній обробці даних // Міжнародний науковий журнал «Інтернаука». – 2020. - №12

Ключові слова. Семантика, структурована інформація, семантичний веб, K-means, онтологія, семантичні мови, семантичні веб-сервіси.

РЕФЕРАТ
НА МАГИСТЕРСКУЮ ДИСЕРТАЦИЮ

выполненную на тему "Использование семантики и формализованных знаний в интеллектуальной обработке данных"
студентом Бражником Максимом Русланович

Работа выполнена на 99 страницах, содержит 25 иллюстрации, 22 таблицы. При подготовке использовалась литература с 31 источника.

Актуальность. С появлением интернета подход к технологиям кардинально изменился. На сегодняшний день во всемирной сети хранится большое количество информации. Такие массивы данных чрезвычайно трудно обрабатывать ручными способами, а с ростом цены на рабочую силу, это становится практически невозможно. Сейчас набирают большую популярность технологии структурирования информации в интернете для дальнейшей машинной обработки. К таким можно перечислить семантический веб, структуризация с помощью тегов и тому подобное. Такие подходы к хранению информации позволили применять методы классификации и кластеризации данных, смогут помочь человеку во время работы или даже заменить и автоматизировать весь рабочий процесс.

Цель. Целью данной работы является исследование современных методов получения информации из сети и последующим использованием этих данных для кластеризации и добычи полезной информации. Это поможет автоматизировать рабочие процессы при нескольких шагов: поиска структурированной информации и ее последующим использованием методами data mining.

Задание. Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать существующие подходы сбора информации в интернете;
- подробно разобрать наиболее целесообразные методы сбора;
- проанализировать успешный опыт реализации программного

обеспечения иностранными коллегами в данной сфере;

- разработать программный продукт, который позволит решить проблемы инженеров во время сбора информации в интернете, а также предоставить инструментарий для анализа и добычи полезной информации;
- разработать стратегию стартап-проекта, которая позволит реализовать описанную технологию в качестве конкурентоспособного продукта.

Объект исследования. Семантика и структурированная информация.

Предмет исследования. Взаимодействие с структурированная информации и ее анализ.

Научная новизна. Научная новизна работы заключается в исследовании способов сочетания технологий семантической сети и методов интеллектуального анализа данных для получения новых высококачественных процедур анализа данных.

Практическая ценность. Практическая ценность работы заключается в анализе методов и средств анализа структурированная информации из интернета, разработка веб-приложения качестве примера использования.

Публикации.

Бражник М. Р. Использование семантики и формализованных знаний в интеллектуальной обработке данных // Международный научный журнал «Интернаука». – 2020. - №12

Ключевые слова. Семантика, структурированная информация, семантический веб, K-means, онтология, семантические языки, семантические веб-сервисы.

ABSTRACT

ON MASTER'S THESIS

on topic: Use of semantics and formalized knowledge in data mining

student: Maksym R. Brazhnyk

Work carried out on 99 pages containing 25 figures, 22 tables. The paper was written with references to 31 different sources.

Topicality. With the advent of the Internet, the approach to technology has changed dramatically. Today, a large amount of information is stored on the World Wide Web. Such data sets are extremely difficult to process manually, and with rising labor costs, it becomes virtually impossible. Technologies for structuring information on the Internet for further machining are now gaining in popularity. These include the semantic web, tag structuring, and more. Such approaches to information storage have allowed the use of data classification and clustering methods that can help a person at work or even replace and automate the entire workflow.

Purpose. The purpose of this work is to study modern methods of obtaining information from the network and the subsequent use of this data for clustering and extraction of useful information. This will help automate workflows in a few steps: finding structured information and then using it with data mining techniques.

Task. To achieve this goal, it is necessary to solve the following tasks: to analyze existing approaches to collecting information on the Internet:

- analyze existing approaches to collecting information on the Internet;
- analyze in detail the most appropriate methods of collection;
- analyze the successful experience of software implementation by foreign colleagues in this field;
- develop a software product that will solve the problems of engineers when collecting information on the Internet, as well as provide tools for analysis and extraction of useful information;
- develop a startup project strategy that will implement the described

technology as a competitive product.

Object of research. Semantics and structured information.

Subject of research. Interaction with structured information and its analysis.

Scientific novelty. The scientific novelty of the work is to study ways to combine semantic network technologies and methods of data mining to obtain new high-quality data analysis procedures.

Practical value of research. The practical value of the work lies in the analysis of methods and tools for analyzing structured information from the Internet, the development of a web application as an example of use.

Publications.

Brazhnyk M. R. Use of semantics and formalized knowledge in data mining // Internacional scientific journal «Internauka». – 2020. - №12

Keywords. Semantics, structured information, semantic web, K-means, ontology, semantic languages, semantic web services.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

KD (Knowledge Discovery) – процес виявлення знань

RAF (remote access framework) - фреймворк віддаленого доступу

E-learning (Electronic Learning) – електронне навчання

RDF (Resource Description Framework) – середовище опису ресурсів

SPARQL (Protocol and RDF Query Language) – мова запросів до даних

OBIE (Ontology-based Information Extraction) - отримання інформації на основі онтології

IE (Information Extraction) - отримання інформації

ML (Machine Learning) – машинне навчання

SWM (Semantic Web Mining) – семантичний веб майнінг

LSA (Latent semantic analysis) - латентно-семантичний аналіз

DOM (Document Object Model) - об'єктна модель документа

OWL (Web ontology language) – мова веб онтологій

NLP (Natural Language Processing) - обробка природної мови

ВСТУП

Інтелектуальна обробка даних та Data Mining являються нетривіальними задачами, що мають на меті відшукати нову або корисну інформацію, іншими словами – аналіз, найчастіше великих, масивів даних, який необхідний для подальшої структуризації інформації та перетворення її з незрозумілих масивів в корисний інструмент. Головна задача – це пошук необхідної інформації, так званих знань. Знання тут відіграють вирішальну роль. вони можуть бути класифіковані ще з вхідної інформації, звідки вони виявляються за допомогою відповідних алгоритмів та інструментів, або через зовнішні дані, які спочатку повинні бути розглянуті через призму проблеми, наприклад фонової статистики або ж аналітик може побажати уточнити інформацію на свій розгляд.

Два останні випадки - це цікаві можливості для підвищення цінності процесів виявлення знань. Розглянемо наступний випадок: набір даних складається з країн Європи та деяких економічних та соціальних показників. Напевно, є деякі цікаві закономірності, які можна виявити в даних. Однак аналітик, який регулярно працює з такими даними, знатиме, що деякі країни є частиною Європейського Союзу, а інші - ні. Таким чином, можна додати уточнюючу змінну EU_Member до набору даних, що може призвести до нових маніпуляцій з інформацією, наприклад, певні моделі мають місце лише для країн-членів ЄС.

У цьому прикладі знання були додані до даних на розсуд аналітика, але вони могли б також міститися в деяких зовнішніх джерелах знань, таких як Linked Open Data.

Зв’язані дані, LOD - це відкрита взаємопов’язана колекція наборів даних у машинно-інтерпретованій формі, що охоплює різні сфери від наук про життя до державних даних [3]. Таким чином, має бути можливим використання цих знань у даному аналізі даних на різних етапах процесу їх виявлення.

У недалекому минулому було запропоновано багато підходів по використання LOD у процесах інтелектуального аналізу даних для різних цілей, один з них, створення додаткових змінних, як у прикладі вище. Головною задачею являється вирішення питання, як семантичні дані можуть бути використані на різних етапах аналізу інформації. Крім того, потрібно враховувати, як різні характеристики зв’язаних даних, такі як наявність взаємозв’язків між наборами даних та поведінки онтологій, використовуються різними підходами.

1 ПІДХОДИ ДО ОТРИМАННЯ ЗНАНЬ В СЕМАНТИЧНІЙ МЕРЕЖІ

В останнє десятиліття було запропоновано величезну кількість підходів, що поєднують методи видобування даних та виявлення знань через семантичну мережу. Метою цих підходів є підтримка різних завдань з отримання даних або вдосконалення самої семантичної мережі. Такі методи можна розділити на три категорії:

- Використання підходів, що базуються на семантичній мережі, та зв’язаних даних для проведення процесу пошуку знань.
- Використання методів інтелектуального аналізу даних для дослідження Семантичної Мережі, такий процес зветься Semantic Web Mining.
- Використання методів машинного навчання для створення та вдосконалення семантичних веб-даних.

1.1 Семантична мережа

Семантична павутина заснована на баченні Тіма Бернерс-Лі, винахідника інтернету. Великий успіх світової мережі призвів до нової проблеми: величезна кількість даних обробляється лише людьми, а машинна обробка обмежена. Бернерс-Лі наполягає на створенні більшої кількості програм, машин, що зменшать залежність обробки інформації від людського труда. Сучасні пошукові системи вже досить потужні, але все ще занадто часто повертають надто великі або неадекватну результуючу інформацію. Інформація, що обробляється машиною, може спрямовувати пошукову систему на відповідні сторінки, що в свою чергу покращує результати пошуку.

Сьогодні практично неможливо отримати інформацію за допомогою пошуку за ключовим словом, коли інформація розподілена на декількох сторінках. Розглянемо запит для отримання інформації в інтранеті компанії, де єдиною явною інформацією, що зберігається, є стосунки між людьми та

курсами, які вони відвідували, з одного боку, а також між курсами та темами, які вони охоплюють, з іншого боку.

У такому випадку використання правила, згідно з яким люди, які відвідували курс, присвячений певній темі, мають знання про цю тему, може покращити результати пошуку.

Процес побудови семантичної мережі в даний час є популярною темою. Перед побудовою такої мережі спершу необхідно виконати ряд простіших дій. Наступні кроки показують напрямки в яких розвивається семантична мережа:

- Надання загального синтаксису для полегшення обробки знань машинами
- Розробка загальних словникових запасів
- Погодження логічної мови
- Використання мови для обміну доказами

Бернерс-Лі запропонував багатошарову (layers) структуру для семантичної мережі. Така структура відображає кроки[4], перелічені вище. Мається на увазі, що кожен крок сам по собі представляє цінність під час пошуку знань, а це дозволяє реалізовувати таку мережу поступово.

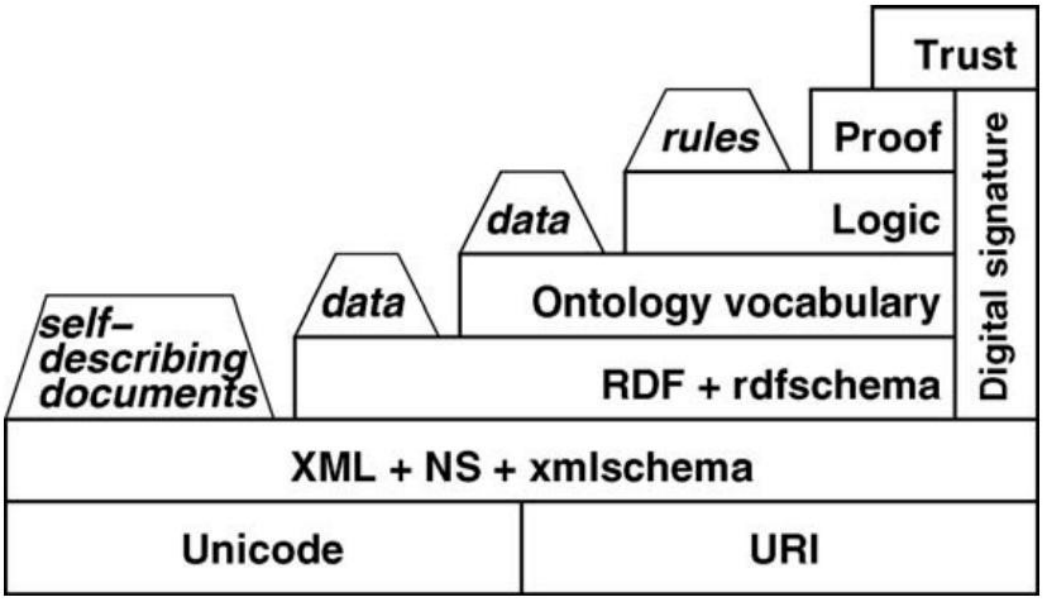

Рис. 1.1 – Шари семантичної мережі

1.1.1 Шари семантичної мережі

На рис. 1 зображені шари семантичної мережі, як було запропоновано в початковій моделі. На перших двох шарах надається загальний синтаксис. Уніфіковані ідентифікатори ресурсів (URI) забезпечують стандартний спосіб посилання на сутності, тоді як Unicode є стандартом для обміну символами. Розширювана мова розмітки (XML) виправляє позначення для опису дерев, а схема XML дозволяє визначати граматики для дійсних документів XML. Документи XML можуть посилатися на різні простори імен, щоб чітко визначити контекст і значення різних тегів. На сьогодні формалізація цих двох рівнів широко використовується, і кількість XML-документів швидко зростає. Хоча XML є одним з кроків, він лише формалізує структуру документа, а не його зміст.

Структуру опису ресурсів (RDF) можна розглядати як перший рівень, де інформація стає зрозумілою для машини: згідно з рекомендацією W3C. RDF – основа для обробки метаданих, забезпечує взаємодію між програмами, які обмінюються машиною незрозумілою інформацією в Інтернеті.

Документи RDF складаються з трьох частин: ресурси, властивості та ствердження. Ресурсами можуть бути веб-сторінки, частини або колекції веб-сторінок, будь-які реальні об'єкти, які не є безпосередньо частиною всесвітньої мережі. У RDF ресурси завжди адресовані через URI. Властивості - це специфічні атрибути, характеристики або відношення, що описують ресурси. Ресурс разом із властивістю, що має значення для цього ресурсу, утворюють оператор RDF. Ствердження можна розглядати як трійку об’єкт-атрибут-значення.

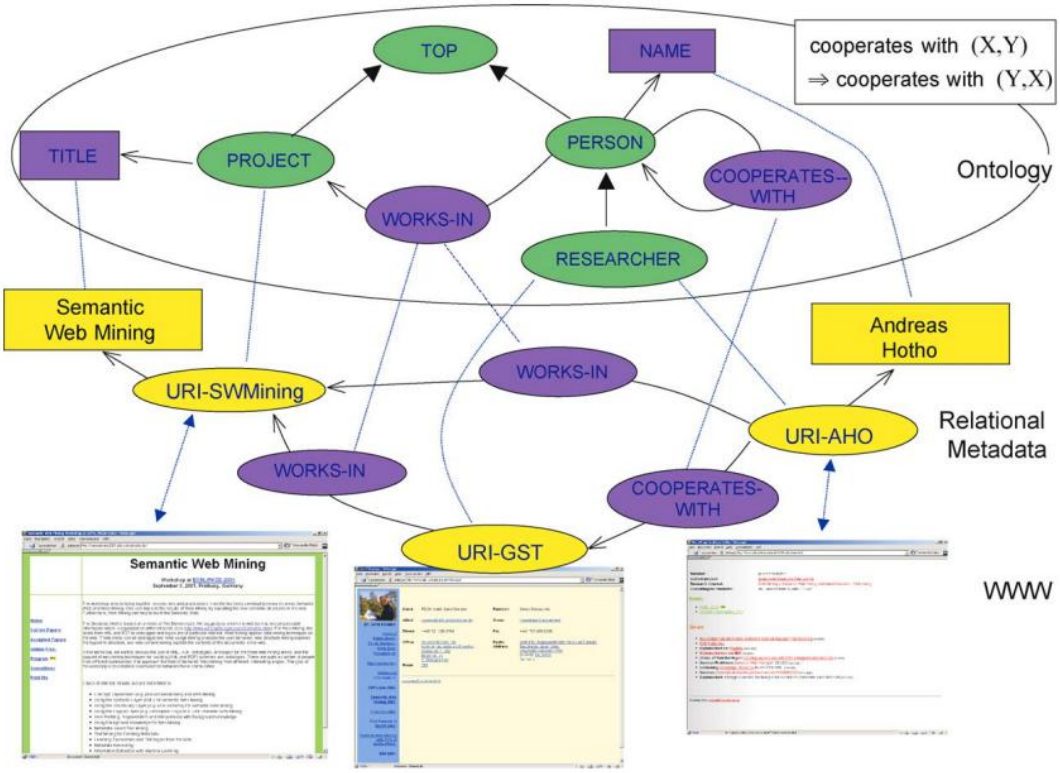

Рис. 1.2 – Відносини між сутностями веб сайтів у форматі онтології

Посередині рис. 2 описаний приклад тверджень RDF. Веб-сторінки представлені у вигляді ресурсів URI-GST та URI-AHO. Твердження внизу праворуч складається з ресурсу URI-AHO та властивості «співпрацює-з» зі значенням URI-GST, що знову ж таки є ресурсом. Ресурс URI-SWMining має значення «Semantic Web Mining» для властивості «title» літералу.

Модель даних, що лежить в основі RDF, є спрямованим графом. Схема RDF описується мовою моделювання, яка включає класи, зв'язки між класами та між властивостями, а також обмеження домену / діапазону для властивостей. RDF пишеться на мовою XML, але не використовує деревоподібну семантику XML.

Схеми XML та XML були розроблені для опису структури текстових документів, таких як документи HTML, Word, StarOffice або LATEX. В XML можна визначати теги для перенесення метаданих, але ці теги не мають формально визначеної семантики, тобто їх значення не буде чітко визначеним. Також важко перетворити один XML-документ в інший без додатково заданої

семантики використовуваних тегів. Призначення XML - групувати об’єкти, але не описувати. Таким чином, XML допомагає в організації документів, надаючи формальний синтаксис. Хоча такий варіант і не можна однозначно назвати «семантичним»[3].

Наступним шаром є словниковий запас онтології. Онтологія – це формальний опис предметної області. Таке визначення абстрактного рівня реалізується різними дослідницькими спільнотами по-різному. Однак більшість із них мають певне спільне розуміння, оскільки більшість із них включає набір понять, ієрархію, аксіоми та відносини між поняттями.

Логіка - наступний шар. Сьогодні більшість досліджень розглядають онтологію та логічний рівень як одне ціле, оскільки більшість онтологій допускають логічні аксіоми. Застосовуючи логічну дедукцію, можна знайти нові знання з інформації. Наприклад, наведена вище аксіома дозволяє логічно зробити висновок, що особа, до якої звертається URI-AHO, співпрацює з особою, до якої звертається URI-GST. Будь який висновок залежить від обраної логіки.

Докази та довіра - це решта шарів. Вони дотримуються розуміння того, що важливо мати можливість перевірити достовірність висловлювань, зроблених у (семантичній) мережі, що довіра до семантичної мережі та способу її обробки інформації зростатиме за наявності підтверджених таким чином тверджень. Тому автор повинен надати доказ, який перевіряє машина. На цьому рівні не потрібно, щоб машина зчитувача сама знайшла доказ, вона просто повинна перевірити доказ, наданий автором.

1.1.2 Схожі галузі досліджень та області застосування

Однією з багатьох галузей досліджень, пов’язаних із семантичною павутиною, є бази даних. За останні кілька років більшість комерційних систем управління базами даних включали можливість зберігання XML-даних для того, щоб зберігати напівструктуровані дані. Оскільки спільнота баз даних вже давно

працює над методами видобутку даних, можна очікувати, що рано чи пізно «видобуток XML» стане активною темою дослідження. Дійсно, існують перші підходи в цьому напрямку [5]. З нашої точки зору, це можна розглядати як окремий випадок семантичного web-mining.

Іншою схожою темою можна назвати тематичні карти, вони дозволяють структурувати зв’язки між об’єктами. Більшість програм для тематичних карт використовує синтаксис XML, на зразок RDF. Насправді, Тематичні карти та RDF тісно пов'язані. Одним з комерційних інструментів в цій області можна назвати «theBrain» дозволяє вибудовувати іменовані відносини між різними об’єктами.

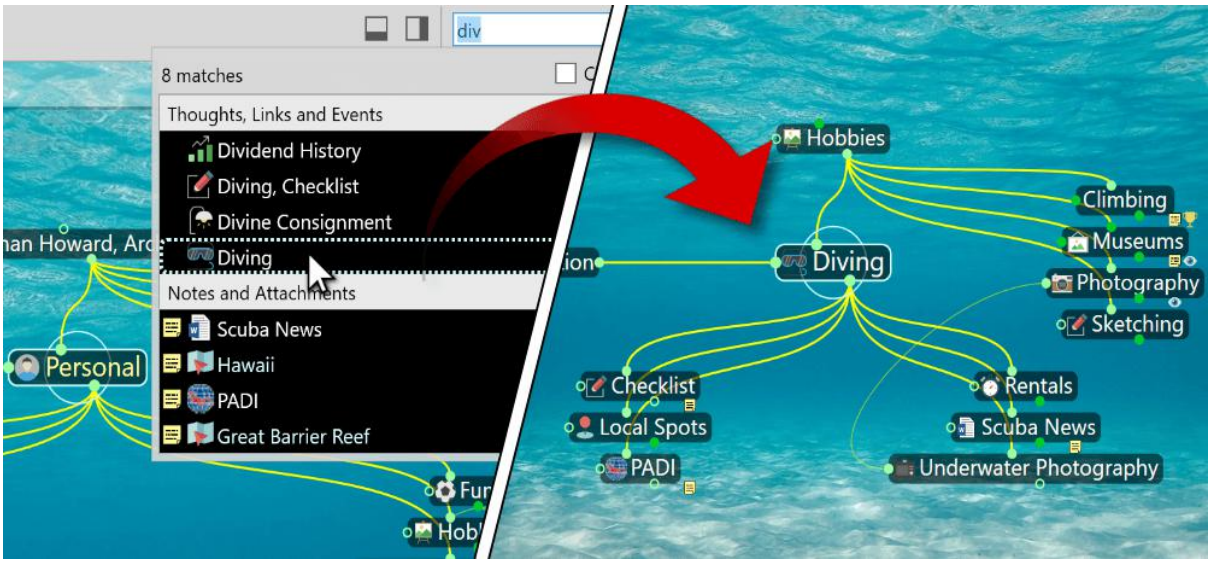

Рис. 1.3 – Приклад роботи інструменту theBrain

Багато різних веб-сайтів можуть отримати переваги від використання як семантичної мережі. Це надає можливість надати інформацію про конкретний домен у інтернеті, що в свою чергу допомагає користувачам знайти відповідну інформацію. Такі системи можна значно покращити, використовуючи магістральну архітектуру та набір інструментів, що базуються на онтології, як це передбачено в SEAL та SEAL-II [1].

Хоча метадані корисні в Інтернеті, вони також залишаються важливими для пошуку ресурсів у однорангових мережах. Приклади включають EDUTELLA, який використовує освітній стандарт LOM, та POOL.

1.2 Web Mining

Web-mining - це застосування методів інтелектуального аналізу даних до вмісту, структури та використаних веб-ресурсів. Таким чином, це нетривіальний процес ідентифікації раніше невідомих та потенційно корисних шаблонів серед великої кількості інформації. Як і інші технолоії для інтелектуального аналізу даних, Web Mining може отримати інформацію з класичної бази даних, але вона також може застосовуватися до напівструктурованих або неструктурованих даних, таких як текст у вільній формі. Це означає, що веб-майнінг – надзвичайно корисний інструмент, необхідний для перетворенні від зрозумілого для людини вмісту до машинозрозумілої семантики. Прийнято виділяти три галузі веб-майнінгу: отримання вмісту, структур та використанньь [6]. У всіх трьох областях застосовується і розробляється широкий спектр загальних методів видобутку даних, зокрема виявлення правил асоціацій, кластеризація, класифікація та отримання послідовностей, що відображають специфічні структури веб-ресурсів та конкретні питання, поставлені перед Web Mining.

1.2.1 Інформація та текст на веб-сторінках

Одна з цілей web mining – аналіз контенту, що знаходиться на веб ресурсах, здебільшого це аналіз тексту. Останні досягнення в мультимедійному аналізі даних обіцяють розширити доступ також до зображення, звуку, відео тощо. Аналіз мультимедійних даних може створювати семантичні анотації, які можна порівняти з тими, що отримані в результаті аналізу тексту. Основними веб-ресурсами, які аналізуються під час web mining, є окремі сторінки.

Пошук інформації - одна з областей, яка забезпечує низку популярних та ефективних, переважно статистичних методів для web mining. Їх можна використовувати для групування, класифікації, аналізу та отримання документів. Ці методи становлять чудову основу для більш складних підходів. Яскравим прикладом є латентно семантичний аналіз. LSA та інші аналітичні

методи виявились цінними для аналізу веб інформації. Однак LSA використовує більш вільне поняття семантики, потрібно багато зусиль, щоб визначити явну концепцію з побудованих відносин.

Окрім стандартних методів аналізу тексту або веб інформація може скористатися перевагами напівструктурованого тексту для веб-сторінки. Теги HTML і розмітка XML несуть інформацію, яка стосується не тільки макета, але і логічної структури.

Аналіз веб інформації спеціально пристосований до характеристик тексту, як це відбувається у веб-ресурсах. Тому основна увага приділяється виявленню шаблонів у великих колекціях документів та часто змінюваних колекціях документів. Це може допомогти під час виявлення певних подій, наприклад нова тема в серед багатьох документів набирає певної популярності, що свідчить про сплеск або зниження інтересу до певних тем.

1.2.2 Структура веб-сторінки

Аналіз структури веб-сторінки зазвичай працює через пошук гіперпосилань. Майнінг фокусується на наборах сторінок, починаючи від окремого веб-сайту і закінчуючи Інтернетом в цілому. Зазвичай аналізується інформація яка часто являється неявною, тобто міститься в структурі гіпертексту.

Такий пошук дозволяє розглянути структуру початкової веб сторінки і рухатися далі в залежності від побудованої топології гіперпосилань, для широкої теми пошуку. Цю інформацію можна знайти на достовірних або «авторитетних» сторінках, які мають посилання на так званих хабах: хаб- це сторінки, які має багато посилань на пов’язані веб-сторінки. Подібним чином пошукова система Google зобов’язана своєму успіху алгоритму PageRank, який стверджує, що релевантність сторінки збільшується із збільшенням кількості гіперпосилань на неї з інших сторінок[7].

Аналіз веб-структури та інформації часто виконуються разом, що дозволяє алгоритму одночасно використовувати вміст і структуру гіпертексту. Дуже часто їх в носять як нероздільні властивості web mining.

1.2.3 Використання веб-сторінок

Під час аналізу важливою інформаією являються записи, залишені користувачами на веб-сайті, найчастіше така інформацію зберігається в формі логів. Зміст та структура веб-сторінок, зокрема, одного веб-сайту, відображають наміри авторів. Фактична поведінка користувачів цих ресурсів може виявити додаткову структуру.

По-перше, взаємозв'язки сутностей можуть бути створені взаємним використанням, навіть у випадках, коли не розроблено жодної конкретної структури. Наприклад, в онлайн-каталозі товарів зазвичай немає певної структури, різні товари просто розглядаються як сутності. Однак, аналіз відвідувань цього сайту, може виявити, що багато користувачів, яких цікавив товар A, також шукали товар B. "Інтерес" може вимірюватися запитами на сторінки з описом товару або розміщенням цього товару в кошику для покупок. Така відповідність між інтересом користувача до різних предметів може бути використана для персоналізації, наприклад, рекомендуючи продукт B, коли переглядається товар A, такий підхід в електронній комерції називається «перехресний продаж».

По-друге, відносини можуть бути ствоерні використанням там, де передбачалося інше відношення. Наприклад, аналіз переходів може показати, що багато користувачів, які перейшли зі сторінки C на сторінку D, робили це шляхом, що вказує на тривалий пошук, часті відвідування довідкової та індексної сторінок, часті зворотні відстеження тощо. Це співвідношення між топологією та використанням може вказувати на проблеми інтерфейсу. Відвідувачі хочуть дістатись від D до C, але їм потрібно здійснити пошук, оскільки немає прямого гіперпосилання або тому, що його важко знайти. Ці

відомості можна використовувати для вдосконалення інформаційної архітектури сайту, а також дизайну сторінки.

По-третє, використання майнінгу може виявляти події у світі швидше, ніж аналіз вмісту інформації. Виявлення та відстеження тем може ідентифікувати події, коли вони знаходять своє відображення в текстах, тобто в поведінці авторів веб-сайтів. Однак пошук інформації часто передує створення певної інформації. Прикладом може бути виявлення початку епідемій або страху перед епідеміями при використанні сайтів медичної інформації[2]. Моніторинг шаблонів дозволяє аналітику вийти за межі аналізу простих часових рядів і відстежувати еволюцію в більш складних шаблонах доступу, таких як правила асоціації або послідовності.

1.3 Аналіз семантики в інтернету

Завдання, яке стоїть перед семантичною мережою, полягає у додаванні машинозрозумілих семантичних анотацій веб-документів, щоб отримати доступ до знань замість неструктурованих матеріалів. Такий підхід дозволяє автоматизвуати процес пошуку знань. Веб-майнінг може допомогти вивчити структури для організації знань, наприклад онтології.

Всі розглянуті тут підходи є напівавтоматизовані. Вони допомагають інженеру аналізувати семантику, але не можуть повністю замінити його роботу. Для отримання якісних результатів не можна замінювати людину в циклі, оскільки в процесі моделювання завжди задіяно багато прихованих знань [8]. Комп’ютер ніколи не зможе повністю врахувати попередні знання, досвід чи соціальні умови. Якби це було так, семантична мережа була б зайвою, оскільки тоді машини, такі як пошукові системи або агенти, могли б працювати безпосередньо на звичайних веб-сторінках. Таким чином, завдання таких систем не в тому, щоб замінити людину, а в тому, щоб надати їй дедалі більшу підтримку.

1.3.1 Семантика створена інформацією та структурою

Витяг онтології з Інтернету є складним завданням. Один із способів - розробка онтології повнстю вручну, але це непомірно дорого. Набагато доцільніше буде використовувати онтологічне навчання для напівавтоматичного аналізу семантики з мережі. Такі методи машинного навчання використовувались для вдосконалення інженерного процесу онтології та зменшення зусиль для інженера.

Онтологічне навчання використовує багато існуючих ресурсів, включаючи тексти, синонімічні словники та бази даних, наприклад WordNet. Аналіз дає проміжні результати, які повинні бути пертворені в машинозрозумілий формат, наприклад, онтологію.

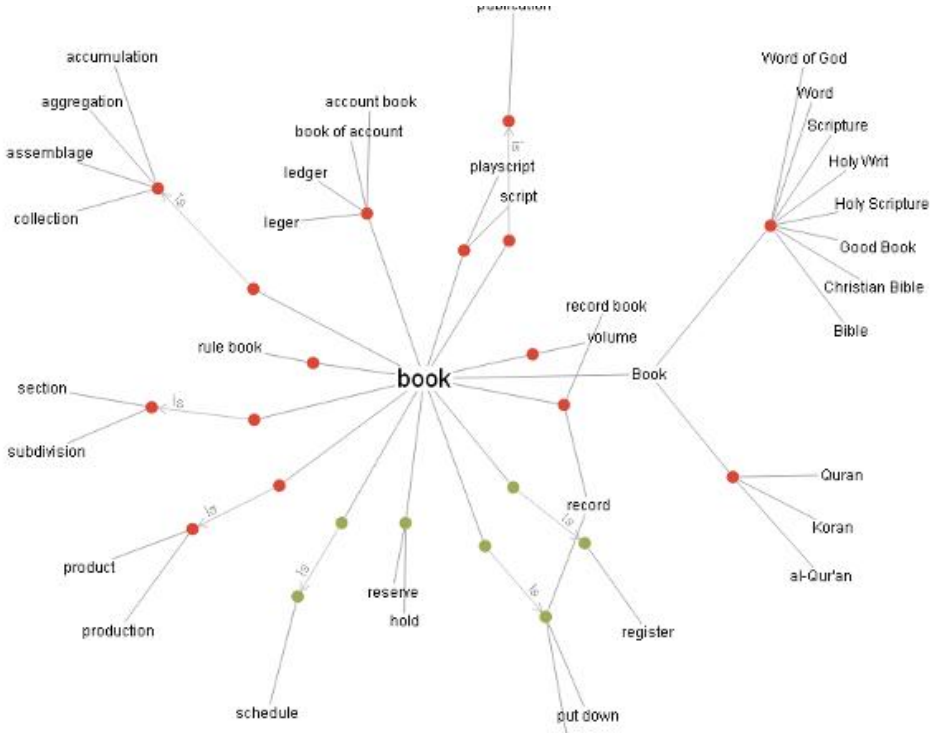

Рис. 1.4 – Приклад можливої структури Word Net

Зростаюче використання онтологій призводить до дублювання знань у спільній області. Онтології, що стосуються доменів, моделюються декількома

авторами. Ці онтології закладають основу для побудови нових специфічних, шляхом збирання та розширення декількох онтологій.

Процес злиття онтологій приймає за вхідні дві або більше онтології та повертає об’єднану онтологію. Ручне злиття онтологій за допомогою звичайних інструментів редагування є складним, трудомістким та багатим на помилки заняттям. Тому нещодавно було запропоновано декілька систем для підтримки інженерів у завданні злиття онтологій [1]. Ці підходи спираються на синтаксичні та семантичні евристики відповідності. Іншим методом є FCA-Merge, який діє знизу та пропонує глобальний структурний опис процесу. Він витягує екземпляри концепцій онтології джерела із заданого набору текстових документів, що стосуються конкретних доменів, застосовуючи методи обробки природної мови. Концептуальна решітка забезпечує кластеризацію концепцій початкових онтологій. Він досліджується та інтерактивно перетворюється в об’єднану онтологію інженером.

Онтологічне відображення - це співвідношення понять однієї онтології та їх екземплярів до понять іншої онтології. Це може бути корисним, наприклад, коли одну з кількох онтологій було обрано правильною для відповідного завдання. Екземпляри можна просто класифікувати з в цільовій онтології.

1.3.2 Навчання на прикладах

Навіть у випадках, коли автор самостійно створює документи і дотримується певної структури, це не позбавляє плутаниці при роботі з існуючими документами. Якщо розглядати питання більш реалістично, то структурування всіх документів вручну практично неможливе. Крім того, деяким користувачам може знадобитися витягти та використовувати іншу або додаткову інформацію від тієї, яку надав автор. Тому для побудови семантичної мережі важливо виробляти автоматичні або напівавтоматичні методи отримання інформації з веб-пов'язаних документів як екземплярів концепцій онтології.

Ряд досліджень показує використання аналізу вмісту для збагачення існуючих концепцій веб-сайту. Наприклад, у роботі пошукової системи Yahoo використовувалися методи категоризації тексту для присвоєння HTML-сторінкам категорії в ієрархії. Це може зменшити ручні зусилля для ведення веб-індексу Yahoo.

Вилучення інформації – information extraction, з текстів є одним з найбільш перспективних напрямків технологій. Такі технології з себе представлють набір автоматичних методів визначення важливих фактів в електронних документах для подальшого використання. Методи IE варіюються від вилучення ключових слів із тексту сторінок з використанням показника TF-IDF, через методи, що враховують синтаксичні структури HTML або природної мови, до технік, які отримають з посилань чітко змодельовану структуру наприклад, онтологію [9].

Вилучення інформації є ідеальною допоміжним інструментом для ідентифікації знань та їх вилучення з веб-документів. Таким чином можна забезпечити користувача автоматичним аналізом документів, так званене неконтрольоване вилучення інформації, або напівавтоматичним, наприклад, пошук відповідних фактів у документах за допомогою виділення конкретної інформації. Прикладом систем IE можна назвати FASTUS та GATE [12]. З появою семантичної сережі GATE була розширена на підтримку онтології, а також навчанням на прикладах. Annotizer OntoMat був розроблений безпосередньо для семантичної мережі.

1.3.3 Використання автоматичних анотацій

Для багатьох сайтів вже існує явна модель домену для створення веб-сторінок. Ці існуючі формалізації можна (повторно) використовувати для семантичної розмітки та видобутку. Наприклад, багато систем управління вмістом генерують веб-сторінки з каталогу товарів за URL-адресами, що відображають шлях до товару в ієрархії каталогів. У прикладі це може

призвести до таких URL-адрес, як Hotels / WellnessHotels / BeachHotel.html, подібні URL-адреси можна знайти в популярних веб-індексах. Класифікація за ієрархією товару є загальновживаною технікою для видобутку веб-даних, див.

Для досягнення загальної схеми онтології та розмітки сторінки можуть генеруватися централізовано одним сервером. У випадку розподіленого авторства використання загальної онтології можна забезпечити за допомогою інтерактивних інструментів, які допомагають окремим авторам розмічати свої сторінки.

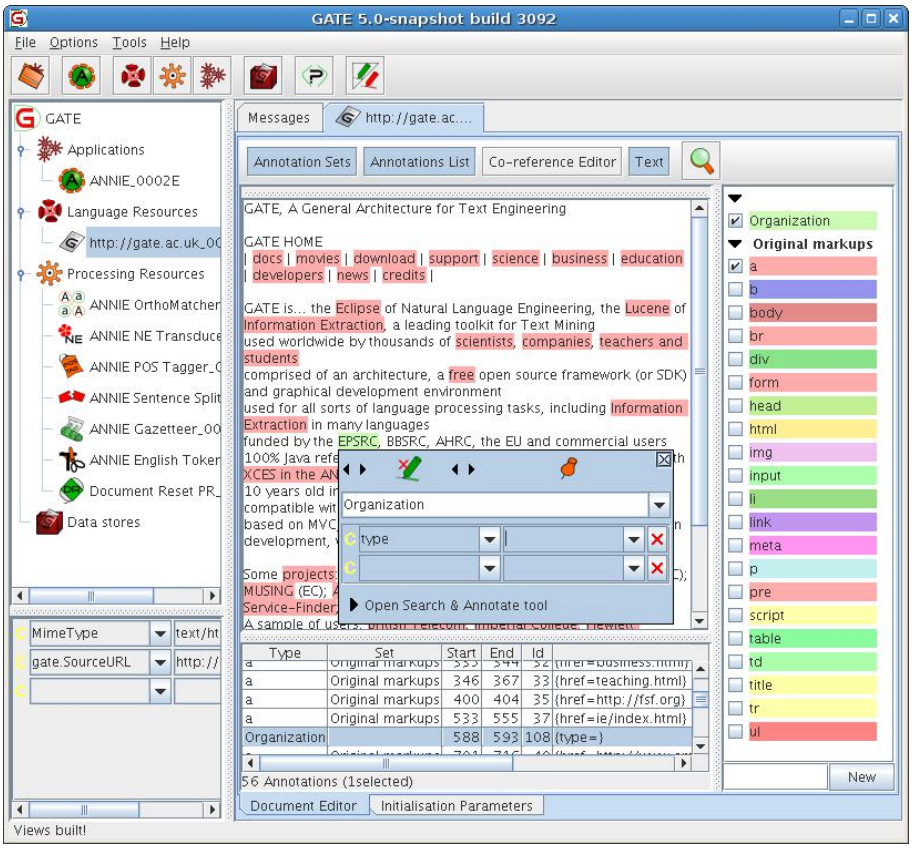

Рис. 1.5 – Приклад роботи системи GATE

1.3.4 Семантика на основі структури

Більшість сучасних веб сайтів має можливість генерації інформації, тобто поява певного тесту буде залежати від інформації, що знаходиться в базі даних. Для таких сайтів аналіз онтології можна провести із внутрішніх джерел, таких

як схеми баз даних, параметри запитів та моделі транзакцій. Така «зворотна інженерія», як правило, передбачає великий обсяг ручної роботи, але її можна автоматизувати використовуючи машинне навчання.

Як згадували в попередніх розділах, через аналіз гіперпосилань можна теж отримати інформацію. Можна зробити висновки про подібність або в якійсь мірі схожості цих ресурсів, на такій основі можна зробити додаток з пошуку схожих сторінок. На основі спостереження, що сторінки, на які часто посилаються одне но одного або на спільні сторінки, ймовірно, будуть пов'язані між собою [5]. Такі методи структурують набір сторінок, але вони не класифікують їх на онтологію. Структуровані гіперпосилання на сторінках піддаються більш безпосередній класифікації. Наприклад, навігаційна сторінка або сторінка «змісту», призначена для орієнтації користувача на сайті, містить багато посилань і мало інформаційного тексту.

Структура розмітки всередині сторінки також може допомогти у аналізі інформації. Концентрація на сегментах сторінки, визначених за допомогою посилання на DOM дерево, об'єктна модель документа або дерево тегів, може служити для ідентифікації основного вмісту сторінки та відокремити його від «шуму», такого як навігаційні панелі, реклама тощо [12].

1.4 Виявлення знань

У своїй фундаментальній роботі від 1996 р. Фаяд представив модель процесу виявлення знань. Модель складається з п’яти кроків, які ведуть від необроблених даних до діючих знань та уявлень, що мають безпосередню цінність для користувача. Весь процес показаний рис. 1.6. Він складається з п’яти етапів:

1. Вибір даних – перший крок, відбувається під час аналізу домену, збір певних знань та визначення цілі аналізу даних з точки зору кінцевого користувача. Виходячи з цього розуміння, можна вибрати

цільові дані, що використовуються в процесі виявлення знань, тобто, вибір відповідної інформації та доречних множин змінних.

2. Попередня обробка - на цьому кроці дані обробляються таким чином, що дозволяє проводити подальший аналіз. Типові дії, вжиті на цьому етапі, включають обробку відсутніх значень, виявлення і потенційно виправлення шуму та помилок у даних, усунення дублікатів, а також узгодження, злиття та вирішення конфліктів для даних, отриманих з різних джерел .
3. Трансформація – третій крок, представляє дані у необхідну форму, над якою можуть працювати алгоритми аналізу даних - у більшості випадків це означає перетворення даних у форму, де кожен екземпляр представлений вектором ознак. Для поліпшення роботи наступних алгоритмів видобутку даних на цьому етапі також можуть застосовуватися методи зменшення розмірності, щоб зменшити кількість змінних.
4. Видобуток даних - після того, як дані представлені в корисному форматі, початкова мета процесу відповідає певним методам, таким як класифікація, регресія або кластеризація. Цей крок включає вирішення, які моделі та параметри можуть бути доречними, наприклад, моделі для категоризованих даних відрізняються від моделей для числових даних, також узгодження методу видобутку даних із загальними критеріями процесу KD, наприклад, кінцевий користувач може бути більше зацікавлена в інтерпретованій, але менш точній моделі, ніж дуже точна, але складна для інтерпретації модель. Після вибору методу та алгоритму інтелектуального аналізу даних відбувається сам аналіз даних: пошук шаблонів, що цікавлять, у певній репрезентативній формі або набори таких екземплярів, як набори правил або дерева.
5. Оцінка та інтерпретація - на останньому кроці розглядаються закономірності та моделі, отримані алгоритмом інтелектуального

аналізу даних, щодо їх обґрунтованості. Крім того, користувач оцінює корисність знайдених знань для даного додатку. Цей крок може також передбачати візуалізацію витягнутих моделей та моделей або візуалізацію даних із використанням витягнутих моделей.

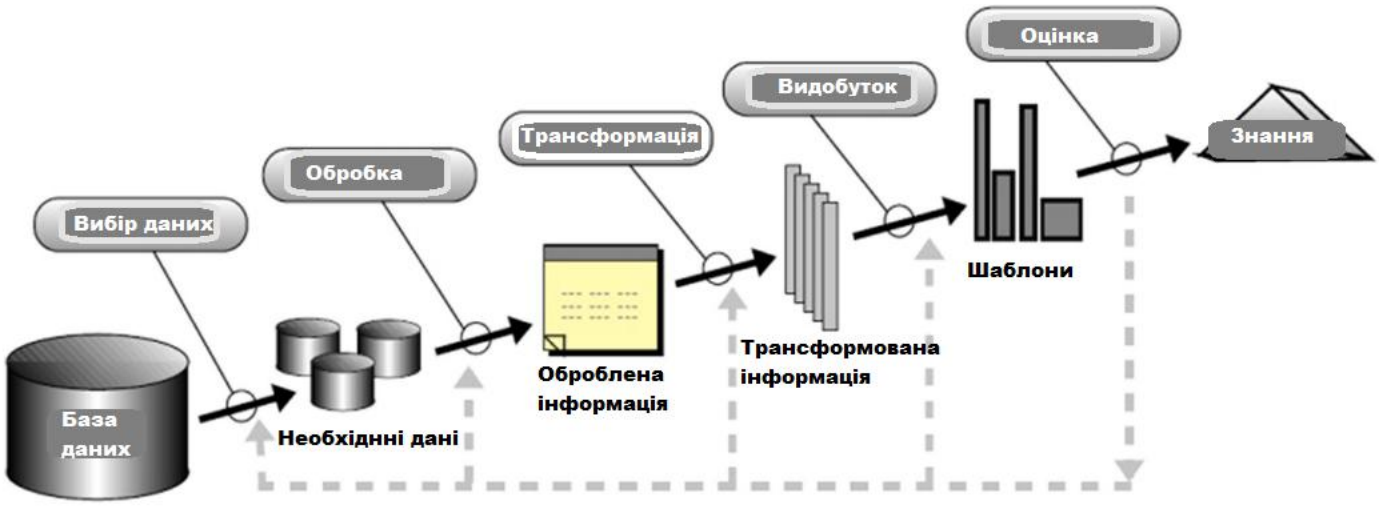

Рис. 1.6 – Схематичне зображення кроків процесу виявлення знань

1.5 Web mining та семантична мережа

Семантику можна використовувати для веб-майнінгу в різних цілях. Деякі з представлених підходів, спираються на порівняльну формалізацію семантики, тоді як інші можуть використати семантичну мережу. Семантична мережа пропонує гарну основу для збагачення веб-майнінгу: типи посилань тепер чітко описані, що дозволяє інженеру отримати глибші уявлення про аналіз веб-структури, а вміст сторінок має формальну семантику, що дозволяє застосовувати технології видобутку інформації, які вимагають більш структурованої інформації. Оскільки різниця у використанні семантики для веб-майнінгу та майнінгом семантичного інтернету дуже мала, ми будемо обговорювати ці питання комплексно.

Першою основною областю застосування є видобуток вмісту. Гіперпосилання на сторінці є частиною тексту цієї сторінки, а на семантично розміченій сторінці вони є елементами сторінки так само, як і текст. Отже, зміст і структура міцно переплітаються, а іноді трактуються як одне ціле [4]. У семантичній мережі різниця між аналізом вмісту та структури повністю зникає. Однак слід зазначити, що розподіл семантичних анотацій усередині сторінки та між сторінками може надати додаткові неявні знання.

1.5.1 Аналіз змісту та структури

Онтології використовуються як базові знання під час попередньої обробки з метою покращення результатів. Ми попередньо обробляємо вхідні дані, наприклад, текст, та застосовуємо евристику, засновану на онтології, для вибору ознак та агрегування ознак. На основі цих тверджень ми обчислюємо множинні результати кластеризації за допомогою k-Means, метод кластеризації. Використовуючи онтологію, ми можемо вибрати результат, який найбільш відповідає нашому завданню.

Аналіз веб-структури також можна вдосконалити. Вже згаданий алгоритм PageRank, співпрацює з алгоритмом аналізу ключових слів, але обидва вони не залежать один від одного. Тож PageRank розглядатиме будь-яку цитовану сторінку як релевантну, незалежно від того, чи вміст цієї сторінки відображає запит. Беручи до уваги також текст прив'язки гіперпосилання, програмний інструмент CLEVER може більш конкретно оцінити відповідність даного запиту. The Focused Crawler покращує це шляхом інтеграції актуального вмісту в модель графіка посилань та завдяки більш гнучкому способу сканування.

Важливою групою методів, які можна легко адаптувати до семантичного аналізу вмісту та структури, являється реляційний видобуток даних, раніше називався Індуктивне логічне програмування - ILP [10]. Реляційний видобуток даних шукає закономірності, що включають кілька відносин у реляційній базі даних. Він включає методи класифікації, регресії, кластеризації та аналізу

асоціацій. Алгоритми можуть бути трансформовані для роботи з даними, описаними в RDF, або за допомогою онтологій. Побудова реляційного аналізу даних вимагає вирішення декількох проблемами. Перша - це розмір наборів даних, що підлягають обробці, а друга - розподіл даних через семантичну мережу. Масштабованість до великих наборів даних завжди була основною проблемою для алгоритмів ILP. З очікуваним зростанням семантичної мережі ця проблема також зростає. Отже, ефективність алгоритмів видобутку повинна бути покращена. Для обробки розподілених даних повинні бути розроблені алгоритми, які виконують майнінг розподіленим способом, таким чином, що замість цілих наборів даних повинні передаватися лише проміжні результати.

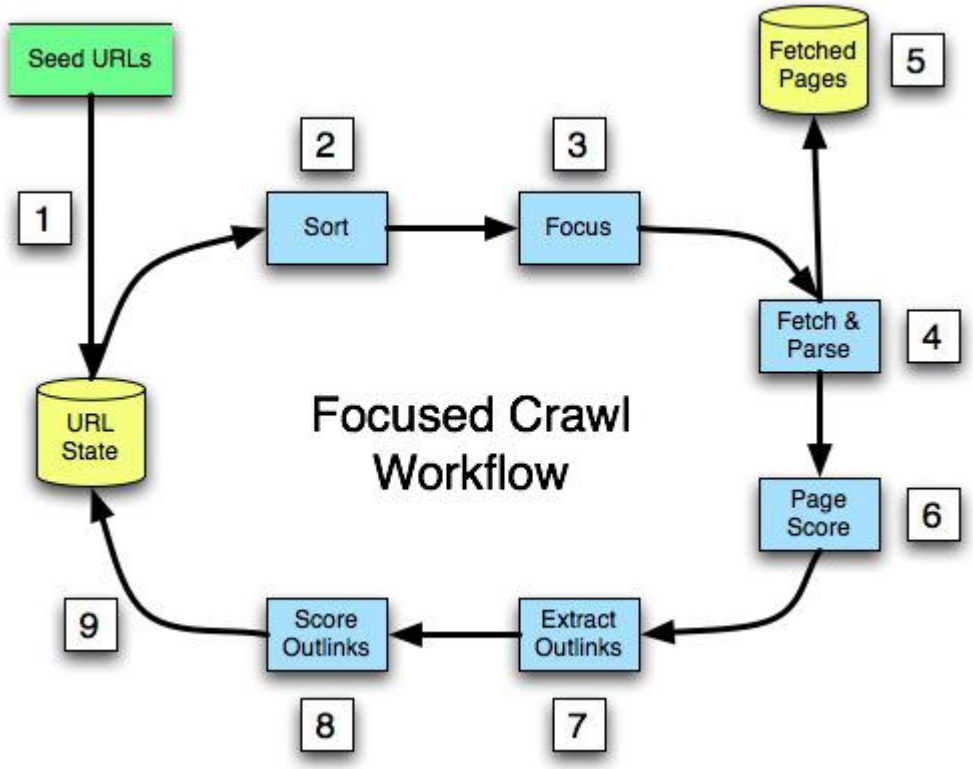

Рис. 1.7 – Focused Crawl

1.5.2 Application events

Події додатків визначаються стосовно домену програми та сайту, що є нетривіальним завданням, яке зводиться до детальної формалізації бізнес-моделі сайту. Наприклад, події на сторінці е-комерції будуть перегляди продукту та кліки, коли користувач виявляє особливий інтерес до конкретного товару, запитуючи більш детальну інформацію, від готелю Beach до переліку його цін у різні сезони. Такі події можна охарактеризувати за змістом, наприклад, готель на пляжі або, загальніше, усі готелі Wellness або просто всі готелі, та послугою, що вимагається при використанні цієї сторінки, функція "пошук готелів за місцем розташування" [2].

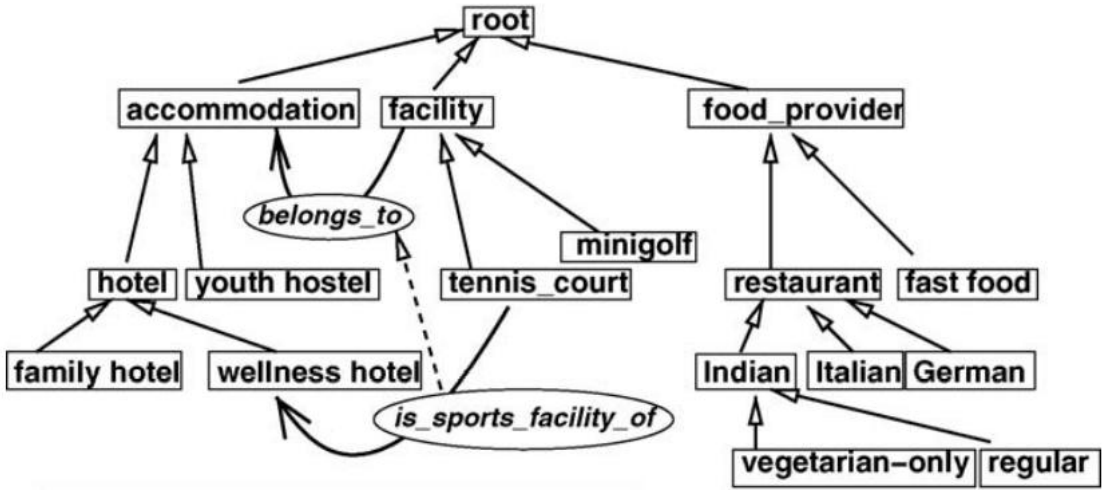

Рис. 1.8 – Приклад можливої онтології для веб-сторінки

На рис. 1.9 показана приклад онтології сервісу для сайту. На сайті відображається інформація, пов’язана з розміщенням, на різних рівнях деталізації: домашня сторінка, сторінки категорій товарів та окремі сторінки товарів. Стратегії пошуку складаються із зазначення одного або кількох параметрів розташування, ціни та назви. Параметри та їх значення задаються за вибором меню або за допомогою набору тексту. У відповідь сервер генерує

сторінку категорії з усіма готелями чи базами, що відповідають заданим специфікаціям.

Після того, як запити були зіставлені з концепціями, перетворені дані готові до видобутку. Групова абстракція часто необхідна для отримання результатів: на сайті з динамічно сформованими сторінками кожна окрема сторінка буде запитуватися настільки рідко, що при аналізі поведінки навігації не буде виявлено жодних закономірностей. Швидше, закономірності можуть існувати на більш абстрактному рівні, що призводить до таких правил, як "люди, які зупиняються в оздоровчих готелях, також, як правило, їдять у ресторанах". Закономірності, отримані з аналізу попередніх даних, не є корисними для таких програм, як рекомендаційні системи, коли нові товари вводяться в каталог продукції: Новий готель X не можна рекомендувати просто тому, що він не був на туристичному сайті до вчора. Знання закономірностей на більш абстрактному рівні може допомогти сформувати рекомендацію готелю X, оскільки це оздоровчий готель.

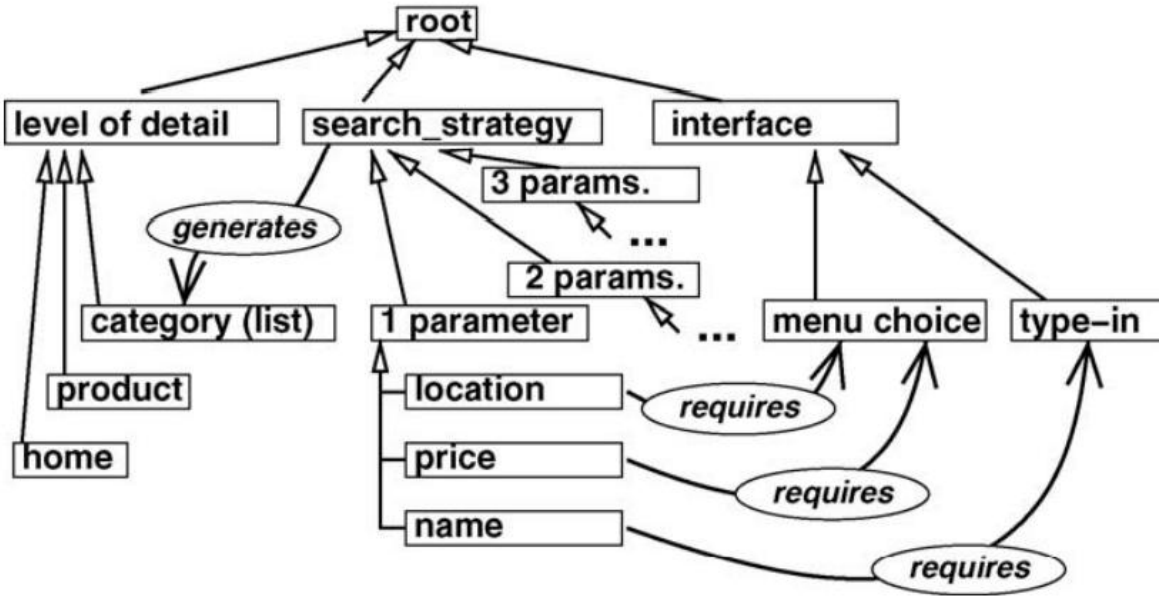

Рис. 1.9 – Онтологія сервісу

Після етапів попередньої обробки, на яких дані доступу були згруповані, наступні методи видобутку можуть використовувати ці групи статично або динамічно. У статичних підходах майнінг оперує концепціями на обраному рівні абстракції, кожен запит відображається рівно з одним поняттям або точно з одним набором понять. Такий підхід, повинен супроводжуватися програмним забезпеченням, щоб аналітик міг повторно відрегулювати обраний рівень абстракції після перегляду результатів.

У динамічних підходах алгоритми визначають найбільш конкретний рівень взаємозв’язків, динамічно вибираючи поняття. Це може призвести до таких правил, як "Люди, які зупиняються в оздоровчих готелях, як правило, їдять у вегетаріанських індійських ресторанах" - пов'язуючи поведінку вибору готелів на порівняно високому рівні абстракції та поведінку вибору ресторану з більш точними характеристиками.

При аналізі та оцінці поведінки користувачів слід мати на увазі, що різні зацікавлені сторони мають різні точки зору щодо використання веб-сайту, що змушує їх досліджувати різні процеси, а також змушує вважати правильними дії різних користувачів. Підводячи підсумок, центральна проблема для майбутніх досліджень у семантичному видобутку веб-матеріалів полягає у розробці, забезпеченні та тестуванні онтологій подій додатків.

1.6 Висновки

У даному розділі ми розглянули різні методи отримання знань: через семантичну мережу, web mining та машинне навчання. Також розглянули покроково стандартні кроки виявлення знань, незалежно від методів. Детально описали принцип роботи з семантичною мережою.

Під час розгляду семантичної мережі були описані її процес створення, що з себе представляє та шари на яких базується. Як альтернативу були запропоновані інші схожі на неї технології. В кінці розділу навели приклад її використання для аналізу поведінки людей під час знаходження в готелі.

Описали принципи роботи Web Mining для веб-сторінок, як відбувається аналіз їхньої структури за допомогою перехресних посилань, доцільність використання змісту для полегшення задач інженерів. Також розглянули аналіз самого тексту з використанням HTML тегів.

Розглянули аналіз веб-сторінок на основі машинного навчання. Такий підхід дозволяє виділяти необхідну інформацію на основі існуючих прикладів та заданої структури, для полегшення аналізу запропонували існуючі системи автоматичної анотації.

2 ТЕХНОЛОГІЇ СЕМАНТИЧНОГО DATA MINING

Дослідження в галузі Семантичної Мережі призвели до досить зрілих стандартів моделювання та структурування знань галузі. Сьогодні онтології семантичної мережі стають ключовою технологією інтелектуальної обробки знань, забезпечуючи основу для обміну концептуальними моделями домену. Для цього широко використовується Інтернет-мова онтології OWL, яка виникла фактичним стандартом для визначення семантичних онтологій Інтернету. Таким чином, семантичні веб-технології, які представляють знання про галузь, можуть створити основи для систематичного включення знань про галузь в інтелектуальне середовище видобутку даних.

2.1 Роль онтологій в семантичному Data Mining

Перспектива та механізм використання онтологій у семантичному аналізі даних різняться залежно від різних систем та додатків. Питання, чому онтологія корисна для сприяння процесу видобування даних, не має єдиної відповіді. Переглядаючи підходи, засновані на онтологіях, ми узагальнюємо наступні три цілі, для яких онтології були введені в семантичний аналіз даних:

- Для подолання семантичного розриву між даними, програмами, алгоритмами інтелектуального аналізу даних та результатами аналізу даних.
- Надати алгоритми аналізу даних, які керують процесом видобутку, або зменшують простір пошуку.
- Надати формальну інструкцію роботи data mining, від попердньої обробки інформації до кінцевого результату аналізу.

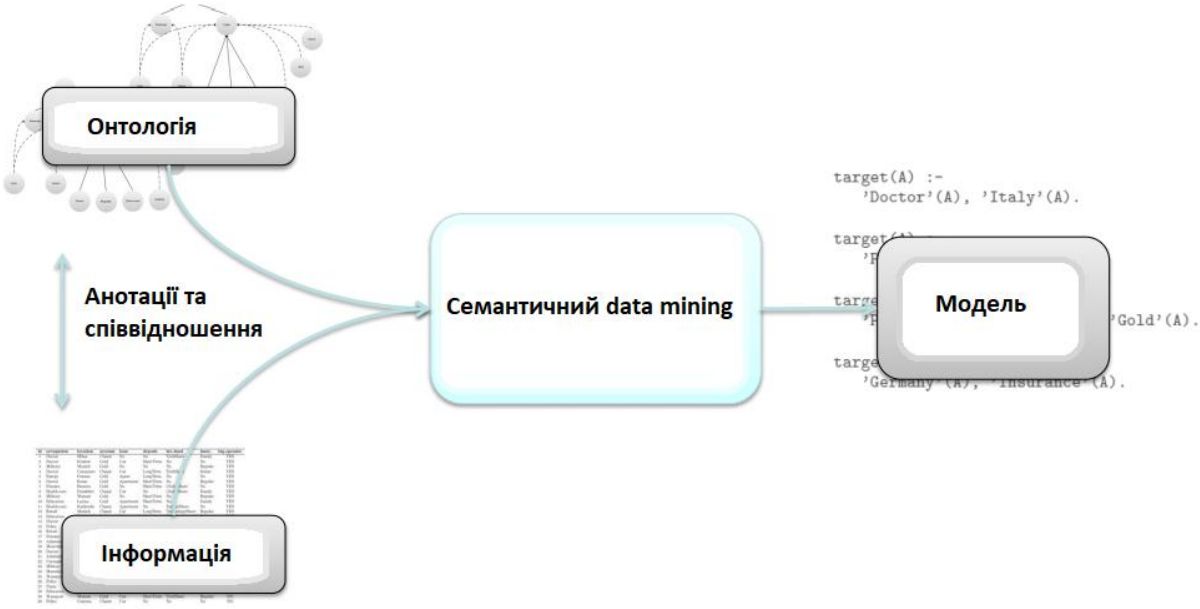

Рис. 2.1 – Схематичне зображення роботи семантичного data mining

2.1.1 Подолання семантичного розриву

Питання, чому знання галузей є корисним в процесі видобутку даних, давно обговорювалось у попередніх дослідженнях семантичного видобутку даних. Дослідники стверджують, що існує розрив у знаннях між даними, алгоритмом видобутку даних та результатами видобутку на всіх етапах видобутку даних, включаючи попередню обробку, виконання алгоритму та генерацію результатів [3].

Попередня обробка даних зазвичай займається очищенням даних, їх нормалізацією та трансформацією. У більшості сценаріїв існують семантичні прогалини на етапах попередньої обробки даних. Без урахування формальної семантики для визначення якості даних використовуються спеціальні або емпіричні методи. Наприклад, правила дефіциту та найближчого сусіда зазвичай приймаються для визначення відхилень та відсутніх значень. На етапі нормалізації та трансформації семантика даних необхідна для розуміння відносин даних. Наприклад, важливо визначити кореляцію між ознаками та атрибутами даних при нормалізації даних. Сильно корельовані атрибути можна звести до одного комбінованого атрибута. На практиці семантичні прогалини

зазвичай заповнюються експертами доменів вручну. Однак онтології виявилися корисними у багатьох завданнях попередньої обробки даних.

Існує семантичний розрив між алгоритмом аналізу даних та даними. Алгоритми видобутку даних зазвичай розроблені для даних, зібраних з різних доменів та сценаріїв. Однак дані з конкретного домену зазвичай несуть специфічну для домену семантику. Алгоритми загального аналізу даних не мають можливості ідентифікувати та використовувати семантику в різних доменах та додатках. Онтології корисні для визначення семантики домену та можуть зменшити семантичну прогалину, додаючи до даних багату семантику. Семантична анотація спрямована на віднесення основного елемента інформаційних посилань до формальних семантичних описів. Такі елементи повинні складати семантику їх джерела. Семантична анотація має вирішальне значення у здійсненні семантичного видобутку даних шляхом залучення формальної семантики до даних. Анотовані дані дуже зручні для наступних етапів семантичного видобутку даних, оскільки дані просуваються до формального та структурованого формату, який пов’язує онтологічні терміни та відношення.

Багато дослідницьких зусиль присвячено подоланню семантичного розриву між результатами аналізу даних та користувачами. Результати аналізу даних можуть бути представлені онтологіями у багатофункціональному семантичному форматі, який можна використовувати під час обміну та повторного використання. Наприклад, вилучення інформації (IE) - це завдання автоматичного вилучення структурованої інформації з тексту. Результати аналізу даних / тексту - це набори структурованої інформації та знань щодо домену. За допомогою OBIE (Виділення інформації на основі онтології) інформація, що видобувається, не тільки добре структурована, але й представлена предикатами в онтології, якими легко обмінюватися та використовувати повторно.

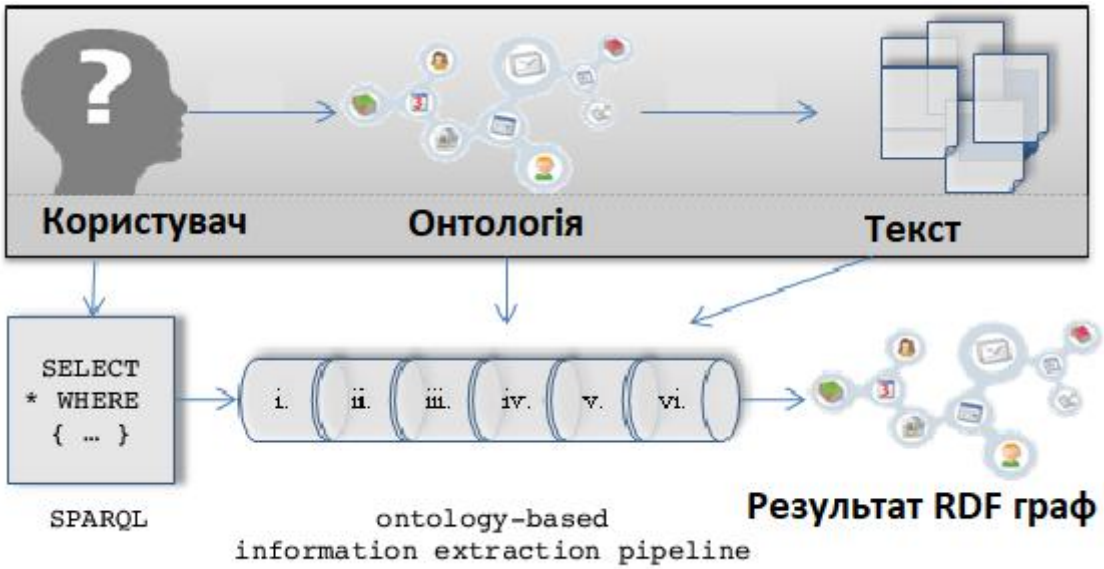

Рис. 2.2 – Приклад роботи OBIE

2.1.2 Створення попередніх знань та обмежень

Визначення та повторне використання попередніх знань є однією з найважливіших проблем для семантичного аналізу даних. Як формальна специфікація понять та відносин, онтологія допомагає створенню формальної семантики попередніх знань. Зашифровані попередні знання можуть допомагати всім етапами процесу видобування даних і впливати на них, починаючи від попередньої обробки і закінчуючи фільтрацією та поданням результатів. Наприклад, гіперграф RDF був розроблений для збору інформації як з онтологій, так і з даних. Онтології включені в графічне представлення даних як знання для зміщення структури графіка, а також для представлення відстані між термінами та поняттями на графіку. Цей підхід перетворює гіперграф та гіперпрограми у дводольний графік, щоб представити як дані, так і онтологію в уніфікованій структурі.

Як сукупність понять і предикатів, онтологія має здатність виконувати логічні міркування і, таким чином, робити висновок про послідовність для цих

предикатів. У семантичному аналізі даних можливість зробити висновок про узгодженість зазвичай представляється як обмеження. Набір обмежень, що забезпечується онтологією, має можливість виявляти суперечливі дані та результати на етапі попередньої обробки, етапі виконання алгоритму та етапі фільтрації та генерації результатів. Онтологія визначає обмеження між кількома завданнями класифікації. Карлсон представив напівконтрольований алгоритм вилучення інформації, який поєднує в собі навчання багатьох екстракторів інформації [7]. Використовуючи онтологію як обмеження набору екстракторів, це дає більш точні результати.

2.1.3 Кластеризація на основі онтології

Кластеризація - це задача інтелектуального аналізу даних, яка групує набір об’єктів в одному кластері, які схожі між собою. Рання робота кластеризації на основі онтології включає використання онтології в задачі кластеризації тексту для попередньої обробки даних, збагачення векторів термінів онтологічними концепціями та сприяння вимірюванню відстані семантикою онтології.

Сонг скористався онтологією на основі тезаурусу для кластеризації тексту із збагаченою концептуальною подібністю. Він запропонував генетичний алгоритм кластеризації тексту з трансформованою прихованою семантичною індексацією з використанням онтології для фіксації асоційованої семантичної подібності.

2.1.4 Вилучення інформації на основі онтології

Вилучення інформації (IE) стосується завдання отримання певних типів інформації з тексту природної мові шляхом їх автоматичної обробки. IE тісно пов'язаний з видобутком тексту. Завдяки цим вказівкам у процесі видобутку, системи OBIE в основному впроваджувались під контрольованим підходом. Хоча дуже мало напівконтрольованих систем IE розглядаються як онтологічні,

вони спираються на випадки відомих зв'язків [4]. Тому ці напівконтрольовані системи також можна розглядати як системи OBIE.

Рання робота OBIE включає вилучення знань із веб-документів та багатофункціональних документів, багатих на дані [15]. Онтологія може забезпечити перевірку узгодженості вилученої інформації в системі IE. Кара представив систему вилучення та пошуку інформації на основі онтології, яка використовує онтологію для перевірки узгодженості. Результат системи IE перетворюється на онтологічні екземпляри через онтологічні сукупності. Результати та перевірка узгодженості базується на онтологічних екземплярах. Карлсон запропонував напівкерований алгоритм вилучення інформації з невеликою кількістю маркованих даних та великою кількістю немаркованих даних. Запропонований алгоритм поєднує в собі кілька екстракторів інформації, що задають обмеження, виключення для різних категорій та відношень. Алгоритм поступово збагачує мітку класифікації, використовуючи найбільш правдиві результати цих екстракторів.

2.1.5 Система рекомендацій на основі онтології

Системи, що рекомендують, або системи рекомендацій - це системи, які призначаються для прогнозування переваг або рейтингів, які користувач повинен надати товару. Системи рекомендацій стали надзвичайно популярними протягом останніх років і застосовуються в різних додатках, включаючи фільми, музику, новини, книги, наукові статті, пошукові запити та соціальні теги. У хорошій системі рекомендацій зазвичай потрібна різноманітна інформація з кількох джерел. Онтологія може інтегрувати використання цієї інформації та керувати перевагами рекомендацій.

Рання робота системи рекомендацій на основі онтології використовує онтологію для профілювання користувачів, персоналізованого пошуку та перегляду веб-сторінок. Нещодавно Пудота та інші запропонували систему рекомендацій, яка автоматично генерує та рекомендує теги для веб-ресурсів.

Веб-документи спочатку коментуються та відповідають термінам в онтології. Потім проводяться аналіз, щоб вивести нові знання з анотованих термінів. Кан і Чой запропонували систему рекомендацій на основі онтології, в якій онтологія використовується для кодування довгострокової та короткострокової інформації про переваги. Онтологія переваг користувача будується на основі концепцій загальної онтології домену разом з документами, які користувач відвідав. Рекомендація складається на основі подібності між онтологічними поняттями та термінами.

Воутер Іджентема розробив систему рекомендацій, Athena, щоб надати онтологічну рекомендацію для системи подачі новин [12]. Він розширює рамки Herme, основу, якої використовується для побудови служби персоналізації новин, за допомогою онтології для визначення семантичних зв'язків між термінами та поняттями. Онтологія використовується для зберігання понять та їхніх зв'язків з новинами. Кантадор запропонував іншу систему рекомендацій новин. Онтології доменів використовуються для забезпечення концептуальної основи вмісту новин та вподобань користувачів. Онтології доменів можуть автоматично анотувати статті новин семантичними поняттями, що відображаються як у текстовому змісті, так і в онтологіях домену.

2.2 Продуктивність сучасних підходів

Онтологія може допомогти у процесі видобутку даних у різних сферах, як формалізована специфікація понять домену та взаємозв’язків. Розумно очікувати збільшення продуктивності підходів, заснованих на онтології, порівняно з підходами до аналізу даних без використання онтологій чи іншої форми знань домену. Багато досліджень із семантичного аналізу даних засвідчили переваги такого підходу. Завдяки добре розробленим алгоритмам, попереднє дослідження допомагає підвищити продуктивності, або виконати завдання з інтелектуального аналізу даних, яких неможливо було досягти без

використання онтологій. У цьому розділі ми дамо короткий підсумок для підвищення ефективності підходів, заснованих на онтології, та їх застосування.

2.2.1 Приріст продуктивності в точності, відкликанні та послідовності результатів інтелектуального аналізу даних

Багато попередніх досліджень, заснованих на онтології, повідомляли про збільшення продуктивності результатів аналізу даних. Як повідомляється, підходи, засновані на онтології, мають кращу точність, ніж традиційні підходи, це стосується багатьох завдань таких як кластеризація тексту, вилучення інформації, передбачення посилань та системи рекомендацій.

Дослідження системи рекомендацій показують, системи, що базуються на онтології, мають кращу точність прогнозування, ніж традиційні методи рекомендацій. Завдяки збагаченій семантиці та зменшенню простору пошуку, у завданні кластеризації генів повідомляється про збільшення швидкості виконання. У завданні видобутку веб-ресурсів та завдання прогнозування наступної сторінки було доведено, що алгоритми послідовного аналізу виконують в 4 рази швидше, ніж звичайні та несемантичні алгоритми.

Онтологічні підходи також покращують узгодженість результатів аналізу даних. Марініка представив подальшу обробку результатів видобутку правил асоціації з використанням онтології для перевірки узгодженості [10]. Семантично суперечливі правила асоціації обрізаються та відфільтровуються за допомогою онтології та логічних міркувань.

2.2.2 Результати обробки даних, багатих на семантику

Онтологія також може допомогти збагатити результати аналізу даних формальною семантикою. Результати обробки даних, багатих на семантику, очікуються від підходів, заснованих на онтології. Наприклад, OBIE здатний витягувати інформацію з подібною або близькою семантикою, яка безпосередньо не відображається у джерелі даних [39].

Не знаючи семантики атрибутів або наборів предметів, видобуток правил асоціацій зазвичай генерує занадто багато правил або навіть суперечливих правил. Видобуток правил асоціації на основі онтології перекриває семантичну прогалину знань домену та алгоритму видобутку правил асоціацій. Це призводить до кращого збору та представлення правил асоціацій шляхом обрізання результатів або зменшення простору пошуку.

За допомогою онтології багаторівневий аналіз правил асоціацій виявить правила на основі концепції, а не на екземплярах [12]. При транзакціях в супермаркетах, наприклад, сир і молоко, хліб та тістечка тощо, традиційні методи видобутку правил асоціацій повинні генерувати правила з цими предметами, тоді як правило багаторівневої асоціації може генерувати правила концептуального рівня, як щоденний список продуктів містить хлібобулочні вироби. Добре контрольована деталізація семантики дозволяє розглянути більш корисні правил асоціації.

2.2.3 Виконання завдання, що неможливо досягти за допомогою традиційних методів аналізу даних

Деякі завдання з видобутку даних, які неможливо досягти традиційними методами видобутку даних, можна виконати за допомогою підходів, заснованих на онтології. Наприклад, традиційне завдання класифікації зазвичай вимагає принаймні розумного обсягу маркованих даних. Використовуючи онтологію як специфікацію попередніх знань, доведено, що завдання класифікації без достатньої кількості маркованих даних має кращі показники порівняно з традиційними методами класифікації [8]. Використовуючи онтологію як концептуальне обмеження узгодженості, модель з немаркованими даними може бути налаштована на ту, яка найкраще відповідає попереднім знанням.

2.3 Інші підходи семантичного видобутку даних

Незважаючи на те, що онтологія є одним із найпоширеніших способів формалізованого представлення знань про домен, також існують інші підходи їхнього представлення у семантичному аналізі даних. Ранні дослідженні з семантичного видобутку даних використовували концепцію ієрархії як дуже важливе представлення знань домену. Алгоритми, засновані на концепції ієрархії, в основному зосереджуються на використанні її узагальнюючої здатності, для подальшої обробки початкових даних на концептуальному рвіні.

Пізніше використовували бази знань для завдань семантичного аналізу даних, включаючи Вікіпедію та Freebase, які не являються точно формальними онтологіями. Габрилович і Маркович обчислили семантичну спорідненість, використовуючи семантичний аналіз, заснований на Вікіпедії, в якому підтверджуються суттєві вдосконалення в обчисленні спорідненості слів та тексту [10]. Мілн та Віттен використовували Вікіпедію як зовнішню базу знань для завдання кластеризації документів [7]. Значного покращення продуктивності було досягнуто за допомогою інформації про поняття та категорії у Вікіпедії для анотації документів збагаченою інформацією про семантику.

Недавно, Meta-path розробли представлення для завдань семантичного аналізу даних. Мета-шлях - це шлях, який визначає композицію відношень між сукупністю термінів на шляху [6]. Зазвичай його визначають на основі графіку мережевої схеми пов'язаних термінів та концепцій видобутку даних. Порівняно з онтологією, кожна мета-шлях може стосуватися декількох понять, тоді як кожен предикат в онтології OWL зазвичай пов'язаний з двома поняттями. Тип мета-шляху визначається типом сутностей у мета-шляху, тоді як тип предиката в онтології визначається відповідними поняттями. Останні дослідження мета-

шляху успішно продемонстрували її здатність досліджувати ефективний алгоритм семантичного аналізу даних з багатьох точок зору.

2.4 Висновки

В даному розділі ми дослідили тему семантичного data mining. З’ясували, що онтологій можуть допомогти в подоланні семантичного розриву, створенні формальної інструкції під час виконання процесу аналізу, та наданні попередніх знань і обмежень. Також розглянули приклад системи рекомендацій для різних комерційних сайтах.

Під час дослідження було знайдено велику кількість альтернативних рішень або реальних прикладів впровадження тих чи інших підходів іноземними колегами, все це було описано та запропоновано до розгляду.. Також дослідили продуктивність сучасних підходів та альтернативні підходи видобутку даних, спираючись на існуючі дослідження та експерименти.

3 ПРИКЛАДИ ВИКОРИСТАННЯ ІНФОРМАЦІЇ З СЕМАНТИЧНОЇ МЕРЕЖІ

3.1 Медична система iASiS

iASiS - проект H2020-RIA по перетворенню великих клінічних даних та фармакогеноміки на корисні знання для персоналізованої медицини та прийняття рішень. iASiS націлений на інтеграцію різнорідних джерел великих даних у графік знань. Джерела даних включають клінічні записки, медичні зображення, геноміку, ліки та наукові публікації. Для того, щоб створити графік знань, iASiS пропонує уніфіковану схему, здатну представити знання, закодовані в неоднорідні джерела великих даних. Крім того, для подолання конфліктів неоднорідності між неоднорідними джерелами, інфраструктура iASiS використовує різноманітні методи аналізу даних. Наприклад, технології обробки природної мови та видобування тексту використовуються для перетворення клінічних нотаток у корисні дані, найсучасніші методи машинного навчання використовуються для аналізу зображень та інструменти геномного аналізу для прогнозування посилань.

Інфраструктура iASiS спирається на онтології, щоб семантично описувати реальні сутності, наприклад, ліки, методи лікування, публікації, гени та мутації. Ці анотації забезпечують основу для семантичної інтеграції цих сутностей. Графік знань iASiS пов'язаний з існуючими графіками знань, наприклад, DBpedia та Bio2RDF, а також застосовуються методи обробки запитів та виявлення знань для вивчення та розкриття закономірностей в графіках знань.

3.1.1 Проблеми опису даних

Сама природа біомедичних джерел даних, зокрема, різноманітність, породжує конфлікти взаємодії між джерелами даних, на які потрібно звернути увагу перед тим, як інтегрувати їх у графік знань:

Структура - проблема, коли джерела даних описуються на різному рівні структурованості, наприклад, структуровані, напівструктуровані та неструктуровані. Структуровані - всі представлені сутності описуються з точки зору фіксованої схеми та атрибутів. Напівструктуровані - на відміну від структурованих даних, кожна змодельована сутність може бути представлена за допомогою різних атрибутів. Нарешті, неструктуровані джерела даних представляють дані без дотримання будь-якої структури або використання моделі даних.

Схематичність – включає ряд проблем:

1) різні атрибути, що представляють одне і те ж поняття в різних джерелах
2) різні типи використовуються для представлення одного і того ж поняття, наприклад, текстовий тип або число
3) різні імена використовуються для моделювання одного і того ж поняття
4) різні онтології використовуються для анотації одного і того ж об’єкта, наприклад, UMLS, SNOMED-CT.

Галузь – проблема опису однієї галузі різними інтерпретаціями. Різні тлумачення включають:

1) омонім - одна і та ж назва використовується для представлення понять з різним значенням
2) синонім - різні назви використовуються для моделювання одного і того ж поняття
3) акронім - різні скорочення для одного і того ж поняття
4) семантичне обмеження - різні обмеження цілісності для моделювання характеристик концепції.

Представлення понять – різний опис, що використовується для моделювання одних і тих самих понять. Сюди входять: різні масштаби або одиниці, різні значення точності, неправильне написання.

Мова – проблема виникає, коли для представлення даних або метаданих використовуються різні мови.

Деталізація – сюди відносяться такі проблеми:

1) дані, змодельовані з використанням різної точності
2) одне і те ж саме вимірювання спостерігаються з різною частотою часу
3) різні критерії

3.1.2 Технології отримання даних

Методи отрмання знань знаходять інформацію, яка закодована в неструктурованих джерелах даних, і представляють отримані знання за допомогою біомедичних онтологій або словників. Таким чином, більшість конфліктів, що існують у джерелах біомедичних даних, вирішуються під час аналізу знань.

Аналіз тексту електронної медичної картки (EHR): Напівавтоматичні методи отримання даних використовуються для забезпечення якості даних, наприклад, видалення дублікатів, вирішення двозначностей та заповнення відсутніх атрибутів. Методи обробки природної мови, розроблені Ернестіною Менасальвас застосовуються для отримання відповідних об'єктів з неструктурованих областей, тобто клінічних приміток або результатів лабораторних випробувань. Методи NLP покладаються на медичні словники, наприклад, UMLS або HPO, а також на інструменти NLP, наприклад, лематизація або розпізнавання іменних сутностей, для анотації понять термінами з медичних словників.

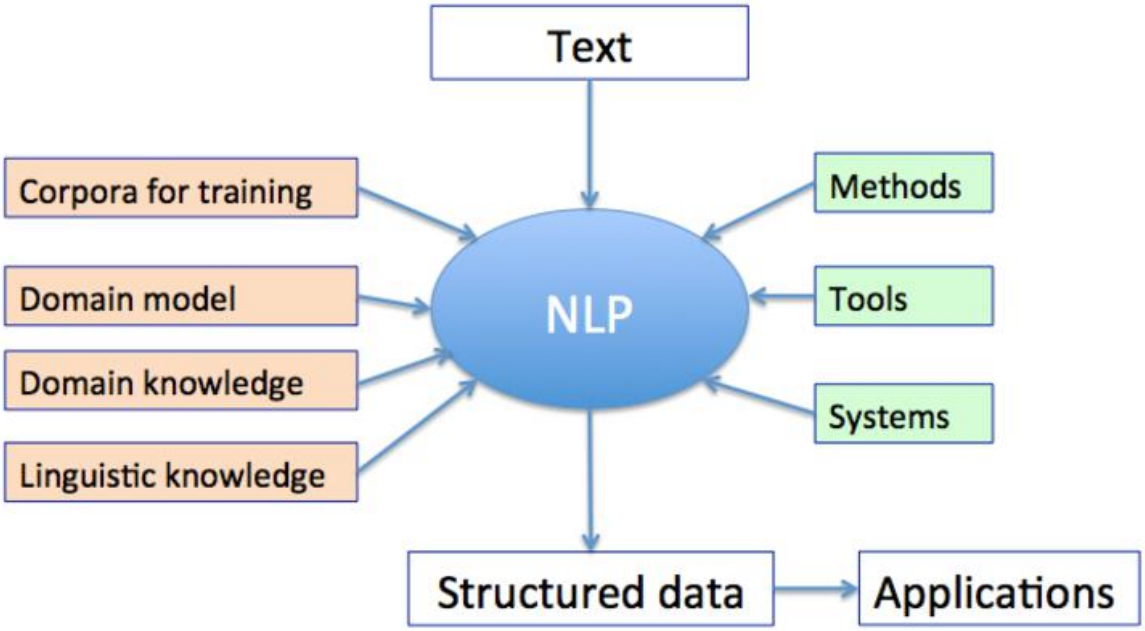

Рис. 3.1 – Приклад NLP

Інструменти видобутку даних, наприклад, catRapid використовуються для виявлення асоціацій білка РНК з високою точністю. Для інтеграції з транскриптомичними даними використовуються загальнодоступні набори даних, наприклад, дані з GTEx, GEO та ArrayExpress. Цей компонент покладається на онтологію генів для визначення ключових генів раку легенів та взаємодії між ними. Крім того, гени анотуються ідентифікаторами з різних баз даних, наприклад, HUGO або Uniprot / SwissProt, а також з HPO.

Алгоритми машинного навчання, розроблені Optiсom, застосовуються для вивчення прогнозних моделей, здатних класифікувати медичні зображення та виявляти сфери інтересів, наприклад, пухлини раку легенів або біомаркери візуалізації [15]. Крім того, методи анотації зображень семантично описують ці сфери інтересів за допомогою онтологій.

NLP та методи мережевого аналізу дозволяють розробити семантичну анотацію об’єктів із біомедичних джерел даних, використовуючи біомедичні онтології та медичні словники, наприклад, UMLS або HPO. Джерела даних включають PubMed15, COSMIC16, DrugBank17 та STITCH18. Анотовані набори даних включають такі сутності, як мутації, гени, наукові публікації,

біомаркери, побічні ефекти, транскрипти, білки та ліки, а також відносини між цими сутностями. Крім того, інструменти зв'язування сутностей, такі як DBpedia Spotlight та TagMe, вирішують завдання вилучення сутності, неоднозначності та зв'язування. Вони використовуються для анотування неструктурованих атрибутів джерел даних, наприклад, назв ліків, генів або мутацій за допомогою постійних веб-посилань, наприклад, у DBpedia або Wikipedia.

3.2 Система NASS

У засобах масової інформації мета відділу документації - допомогти журналістам знаходити інформацію в заархівованих новинах, щоб повторно використовувати її в новій статті. З цією метою ці відділи повинні щодня позначати новини, типовий спосіб це зробити - використання тезаурусу: набір предметів, слів або фраз, що використовуються для класифікації речей. Зазвичай він має структуру ієрархічного списку унікальних термінів.

NASS (News Annotation Semantic System), забезпечує новий метод отримання тегів тезаурусу за допомогою семантичних інструментів та технологій вилучення інформації.

3.2.1 Принцип роботи NAAS

Подібна інформація, різні новини, може бути класифіковані в OBIE. Основними елементами системи OBIE є препроцесор, який працює над вхідним текстом, модуль вилучення інформації, який зазвичай керується семантичним лексиконом та створеною людиною онтологією та база даних знань, що використовується для зберігання результату. Принцип роботи системи представлений на малюнку.

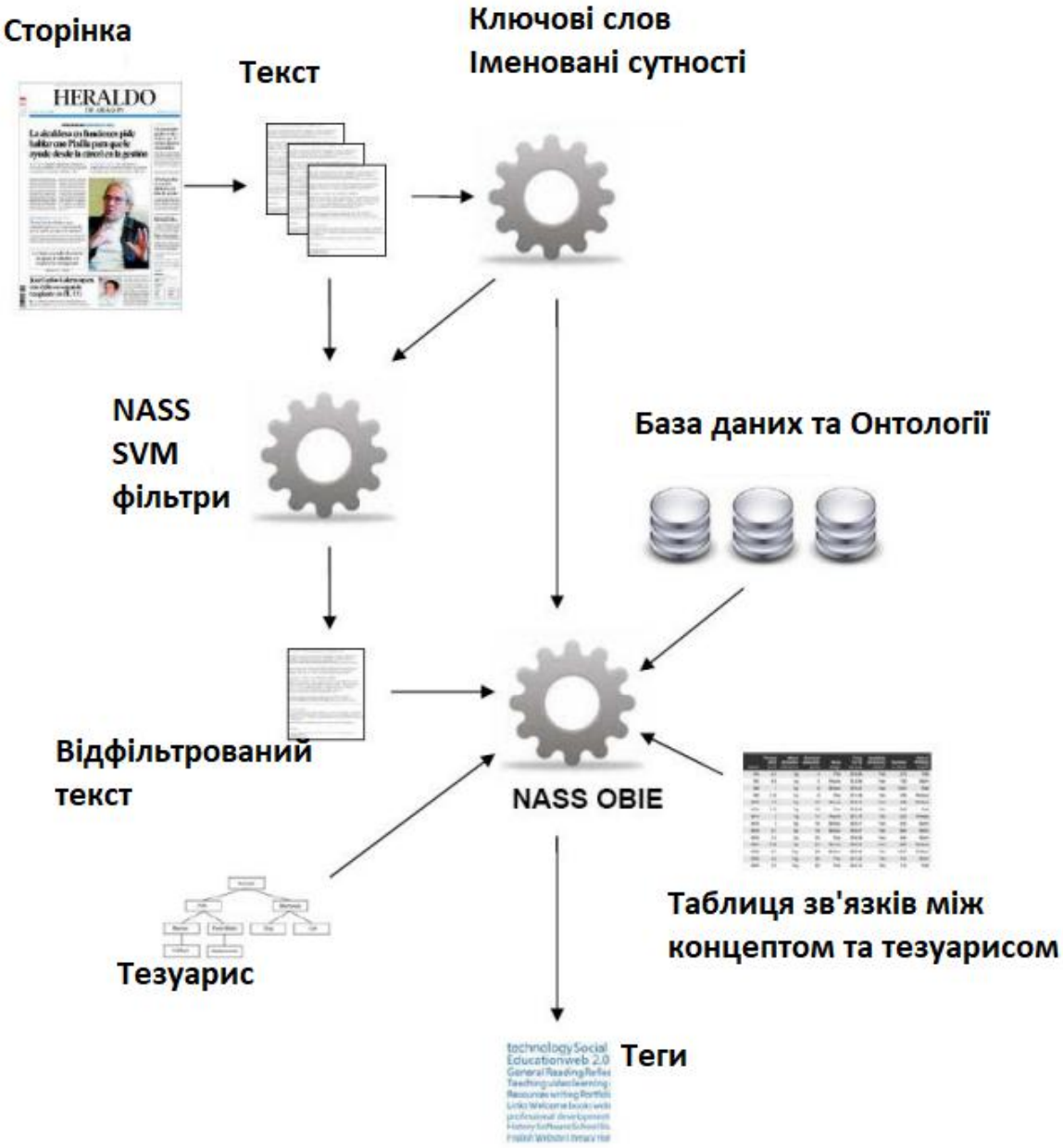

Рис. 3.2 – Архітектура NASS

Спочатку потрібно отримати основні ключові слова зі статті, використовуючи методи видобутку тексту. У той же час, використовуючи обробку натуральної мови (NLP), система отримує інший тип ключових слів, що називаються іменованими сутностями [3]. По-друге, NASS застосовує класифікацію тексту "Support Vector Machines", щоб відфільтрувати лише найрелевантніші статті, що належать до певної теми. Для тексту вибрана ктаегоризація SVM, оскільки це потужний та надійний інструмент. Але існує

одна проблема, як тільки застосували SVM до справжніх газетних статей: він сильно залежить від даних, що використовуються в навчанні. Хоча це дуже добре працює з текстами, що стосуються дуже загальних тем, це не той випадок, коли ключові слова змінюються в текстах, наприклад, коли ми говоримо про спорт.

Результати SVM були покращені за допомогою методів онтологічної інженерії: NASS використовує ключові слова та сутності відфільтрованих текстів для аналізу онтології щодо теми, яку досліджують. Потім NASS використовує відповіді на ці запити, щоб збільшити ймовірність отримання правильного елемента тезауруса в кожному тексті та оновлює оцінки в таблиці. Нарешті, система переглядає цю таблицю, вибирає терміни з оцінкою, що перевищує заданий поріг, а потім позначає текст цими тегами.

3.2.2 Експериментальна оцінка

Під час експерименту було взято 1755 статей, позначених термінами тезуарису, вручну призначеними відділом документації іспанської компанії Grupo Heraldo. Кількість статей обмежується тим фактом, що ми використовували іспанську футбольну онтологію, яка діє лише протягом одного сезону, оскільки щороку команди, гравці та тренери можуть змінюватися. Застосовуючи NASS, досягли не лише понад 95% відкликання та точності, але розширили та виправили маркування зроблені людиною. Операція заснована на проведенні належної сукупності онтологій.

3.3 Висновки

В даному розділі були розглянуті приклади використання семантики і формалізованих знань для аналізу інформації на успішних проектах в сферах медицини та журналістики.

Одним з них iASiS, що спеціалізується на обробці медичних даних. Зважаючи на перечень особливостей, що притаманні саме медичній інформації,

наприклад, неточності в опису інформації, її деталізація або різне трактування одного поняття, були показані проблеми та методи їхнього вирішення в даному проекті. Також до особливостей проекту можна віднести можливість аналізу даних з різних носіїв інформації, наприклад тестову у вільних джерелах або в форматі фотографії.

Інша система NASS, система класифікації тексту в сфері журналістики. Були розглянуті переваги використання такої системи порівнюючи зі звичайною роботою людини. Розглянуті загальна архітектура системи, її робота, а також досліджений приклад роботи NASS на існуючому наборі спортивних статей.

4 РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ІАД ІЗ ЗАСТОСУВАННЯМ СЕМАНТИКИ

Поява великих та розподілених даних RDF у хмарі зв’язаних відкритих даних вимагає підходів до вилучення корисних знань за допомогою таких методів як кластеризація. Ми розглянемо декілька різних підходів вилучення даних за допомогою мови запитів SPARQL, в кінці запропонуємо модифікації цієї мови та розширення, що дозволять зменшити роботу на етапі виконання кластеризації. Основна програма буде включати в себе метод кластеризації K-means, для аналізу належності вхідного об'єкту відносно існуючих кластерів, які сформуються під час цієї кластеризації.

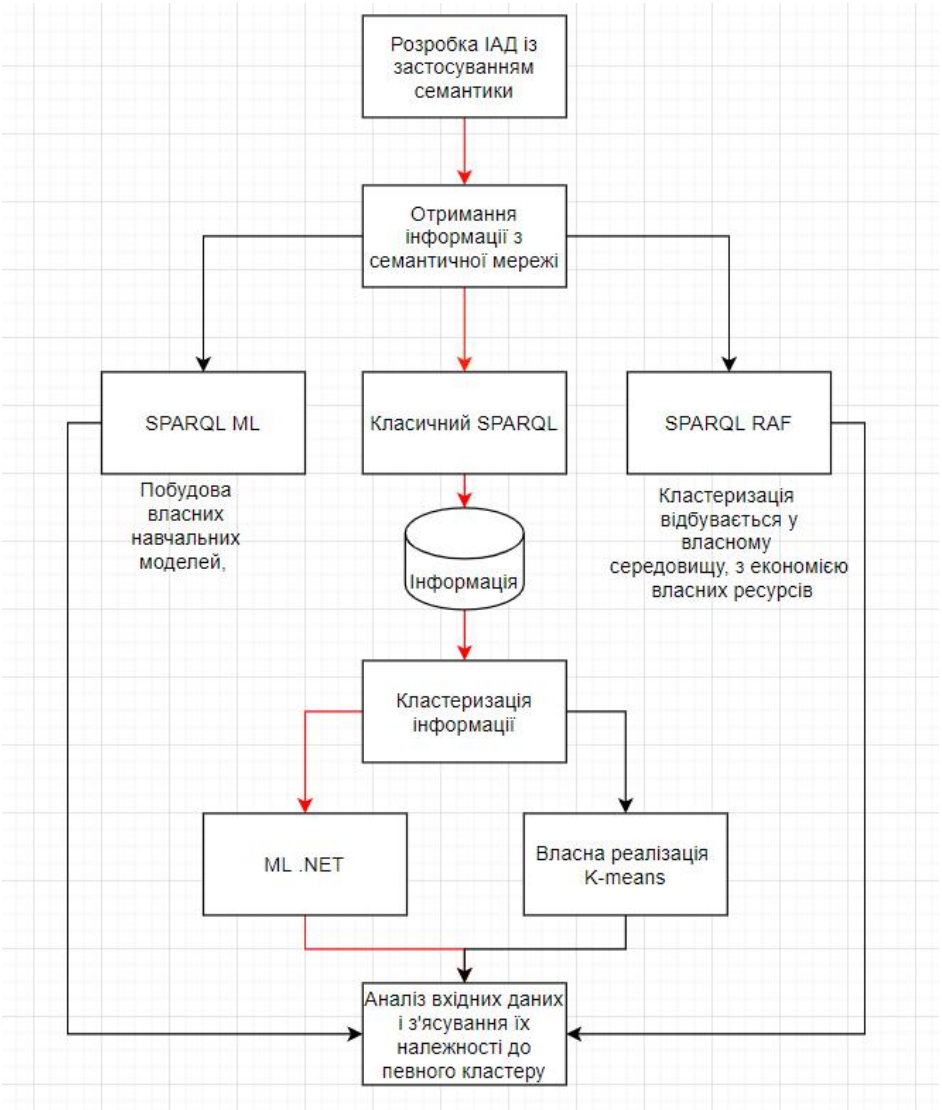

Рис. 4.1 – Можливі шляхи реалізації застосунку

Перед початком розробки та огляду шляхів реалізації нашого застосунку, пропонуємо ознайомитися з Рис.4.1, він дозволить в загальному зрозуміти потенційні способи вирішення нашої задачі. Червоним кольором позначений основний шлях, тобто той, якому присвячена більша частина роботи, по причині його зручності у застосуванні та сучасності.

4.1 Sparql – отримання даних

Мова запитів SPARQL - одне з доповнень до набору інструментів Semantic Web. Він забезпечує потужні засоби для вилучення інформації з великих наборів даних RDF. В даному розділі ми коротко опишем можливості SPARQL

SPARQL - це мова запитів RDF, тобто семантична мова запитів для баз даних, здатна отримувати дані, маніпулювати ними та зберігати їх у форматі Resource Description Framework (RDF). Він був розроблений робочою групою з питань доступу до даних RDF (DAWG) Консорціуму Всесвітньої Мережі, визнаний однією з ключових технологій семантичної мережі. 15 січня 2008 року W3C визнав SPARQL 1.0 офіційною рекомендованим та SPARQL 1.1 у березні 2013 р.

SPARQL дозволяє запиту складатися з потрійних шаблонів, сполучників, від’єднань та необов’язкових шаблонів. Існують реалізації для багатьох мов програмування. Також існують інструменти, які дозволяють підключатись та напівавтоматично створювати запит SPARQL для кінцевої точки SPARQL, наприклад ViziQuer. Крім того, існують інструменти для перекладу запитів SPARQL на інші мови запитів, наприклад, на SQL та на XQuery.

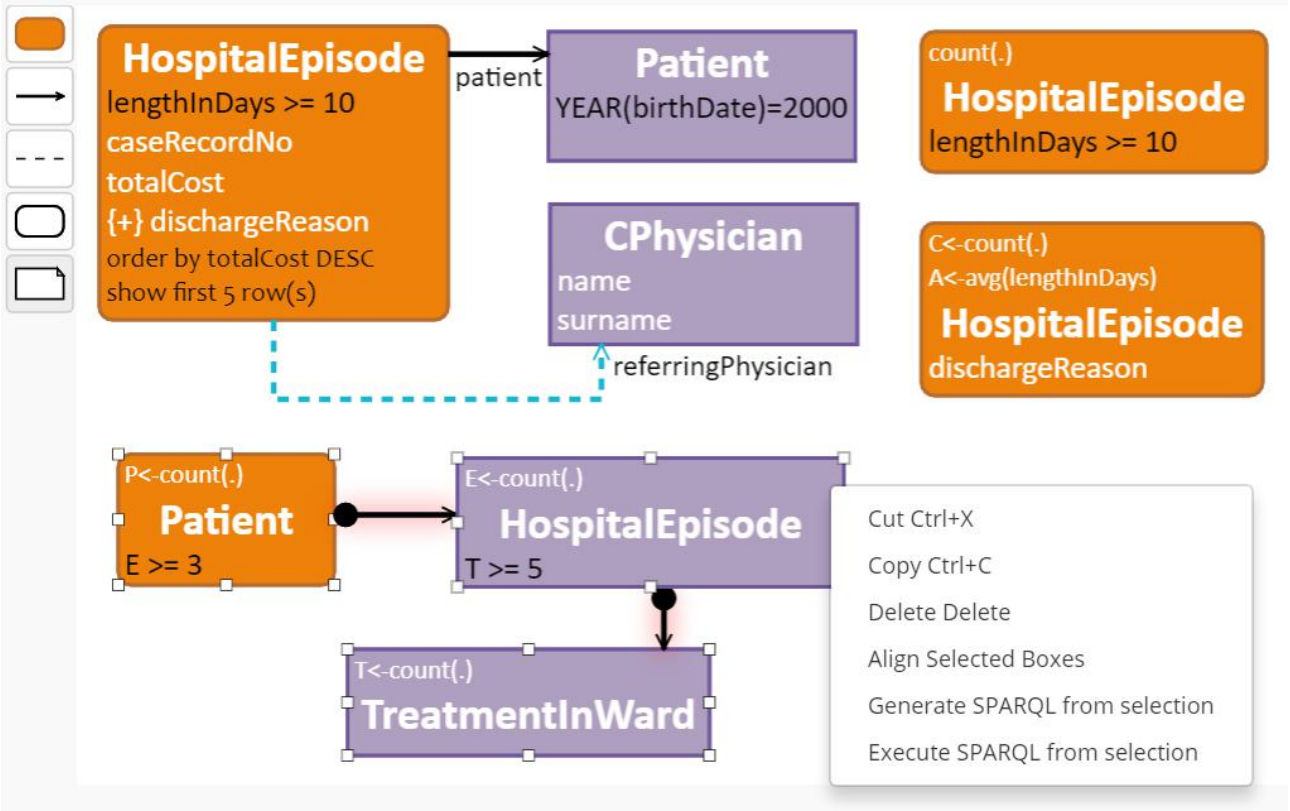

Рис. 4.2 – Інструмент ViziQuer

В роботі ми будемо використовувати розширення, dotnetrdf, для мови C#, що дозволить інтегрувати необхідні запити в код нашої програми. Якщо уникати використання такого розширення буде необхідність в створенні проміжного сервісу для отримання даних через SPARQL, в такому разі потрібно буде використовувати локльну базу даних, що дозволить розділити інформацію з нашою програмою, як альтернативу можна запропонувати передачу даних через JSON. Але такі варіанти недоречні, бо вони не можуть прискорити розробку, порівнюючи з нашим варіантом.

dotNetRDF - це бібліотека, написана на C #, розроблена для забезпечення простого, але потужного API для роботи з даними Resource Description Framework (RDF). Бібліотека пропонує велику різноманітність класів для виконання всіх загальних завдань, починаючи від читання та запису даних RDF до створення запитів. Бібліотека спроектована таким чином, щоб вона була розширюваною і дозволяла користувачам додавати підтримку додаткових

функцій за потреби. Бібліотека працює в основному на рівні Triples, Graphs і Triple Stores.

Перед виконанням наступних кроків пов'язаних з методами кластеризації потрібно обрати, яку інформацію ми будемо використовувати під час аналізу. Був обраний граф з інформацієбю про квіти та їхні характеристики. Характеристики, які будуть нас цікавити:

- Довжина зовнішньої частки оцвітини
- Ширина зовшіньої частки оцвітини
- Довжина внутрішньої частки оцвітини
- Ширина внутрішньої частки оцвітини
- Вид квітки

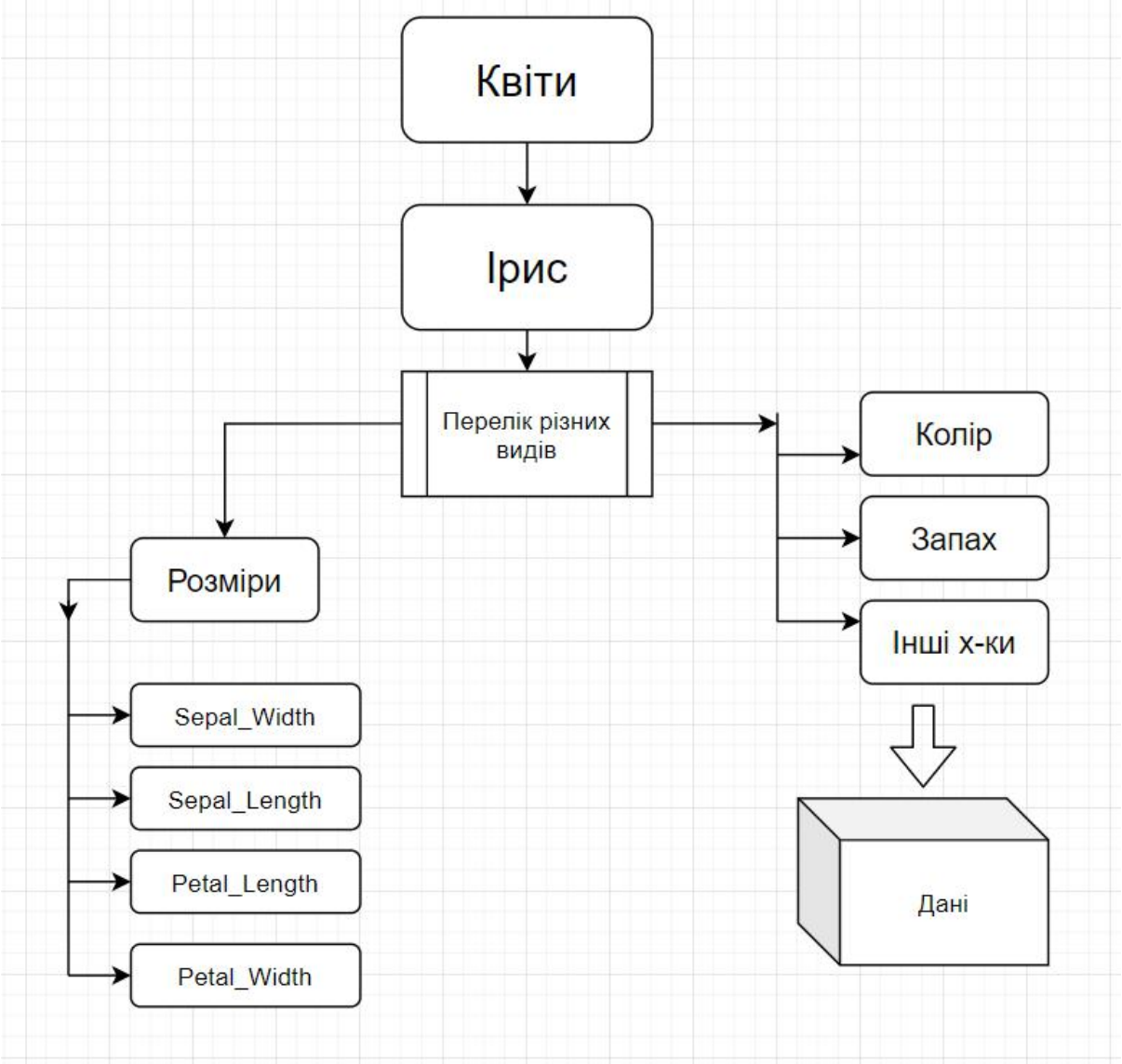

Рис. 4.3 – Схематичне зображення можливих зв’язків в нашому датасеті

В Додатку А для Flowers.rq можна переглянути приклад запиту для отримання інформації, що нас цікавить, а саме: Species, Sepal_Width, Sepal_Length, Petal_Length, Petal_Width.

Show 10 entries Search:

	Species	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
41	virginica	6	3	4.8	1.8
42	virginica	6	3	4.8	1.8
43	versicolor	5.4	3	4.5	1.5
44	versicolor	5.4	3	4.5	1.5
45	versicolor	5.1	2.5	3	1.1
46	versicolor	5.1	2.5	3	1.1
47	setosa	5.4	3.4	1.7	0.2
48	virginica	5.7	2.5	5	2
49	virginica	5.7	2.5	5	2
50	virginica	5.7	2.5	5	2

Showing 41 to 50 of 236 entries Previous 1 … 4 5 6 … 24 Next

Рис. 4.4 – Візуалізація результату виконаного запиту з додатку А

Даний набір даних був обраний не без причин. Оскільки набір «Іриси Фішера» достатньо популярний для задач інтелектуальної обробки даних, більшість дослідників з легкістю зрозуміє подану інформацію, тобто ми уникнемо проблеми входження та розуміння представленої інформації та зосередимся на розробці застосунку. Також цей набір даних можна зустріти в багатьох роботах, наприклад в роботі «Optimal design of fuzzy classification systems» [27] або «Simulation of Back Propagation Neural Network for Iris Flower Classification» [23]. Під час опису наших результатів роботи програми ми запропонуємо ознайомитися з іншим набором даних.

Варто додати, що інформація, що приходить через SPARQL може бути надлишковою, в такому разі можливе використання програмних методів на стороні C# без особливих змін коду SPARQL, хоча в такому випадку і будуть використані надлишкові ресурси пам’яті для зберігання та під час обробки.

На рисунку 4.3 показана візуалізація отриманих даних, для полегшення їхнього читання людиною, насправді для машинної обробки це буде надлишковою дією, тому проміжні результати можуть зберігатися в базі даних, в пам’яті програми, в залежності від їхнього розміру, адже при великих масивах це буде недоцільно, або в нашому випадку у форматі Comma-Separated Values.

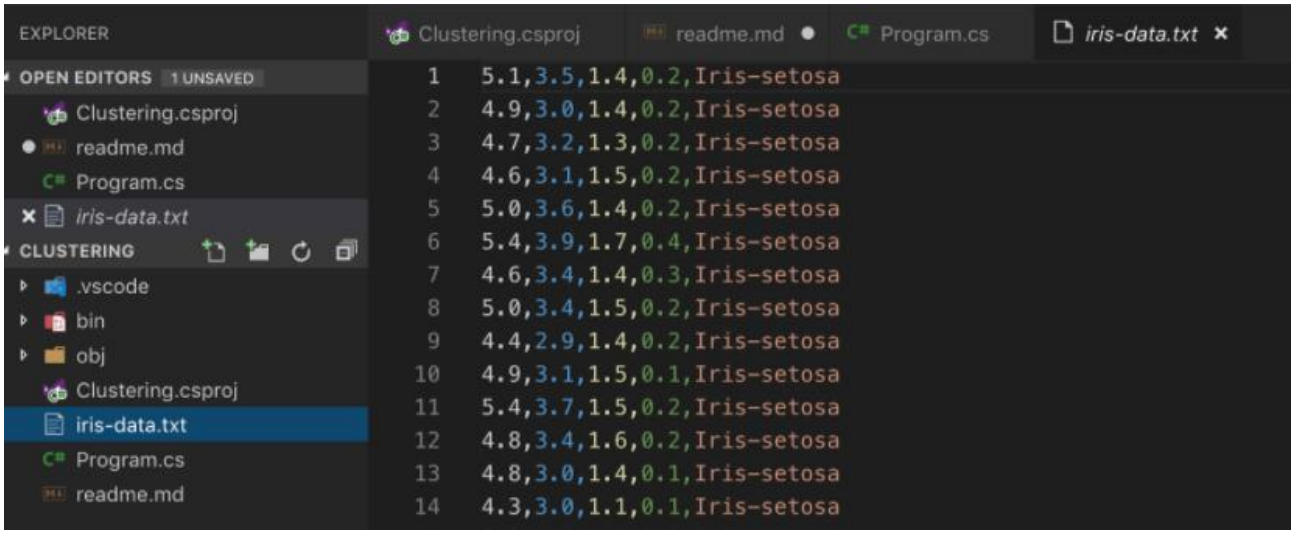

Рис. 4.5 – приклад зберігання інформації в контексті нашої реалізації

4.2 Реалізація методу кластеризації K-Means

Як було описано в попердньому розділі ми будемо реалізовувати програму кластеризації на мові програмування C#. Оскільки Microsoft пропонує існуючий інструмент на основі машинного навчання – ML.NET, то ми розглянемо реалізацію за допомогою цього інструменту та власноруч.

Головний функціонал, що повинен бути розроблений у нашій програмі – це можливість автоматичної ідентифікації квітки за вказаними зарактеристиками.

4.2.1 Реалізація на ML.NET

ML.NET - це безкоштовна програмна бібліотека машинного навчання для мов програмування C # та F #. Вона також підтримує моделі Python, коли вони використовуються разом з NimbusML. ML.NET включає трансформацію для розробки особливостей, таких як створення n-грамів, бінарну класифікацію, багатокласову класифікацію та завдання регресії. У нових випусках бібліотеки були додані додаткові можливості, такі як системи виявлення аномалій та рекомендації, інші підходи, такі як глибоке навчання, будуть включені в наступні версії.

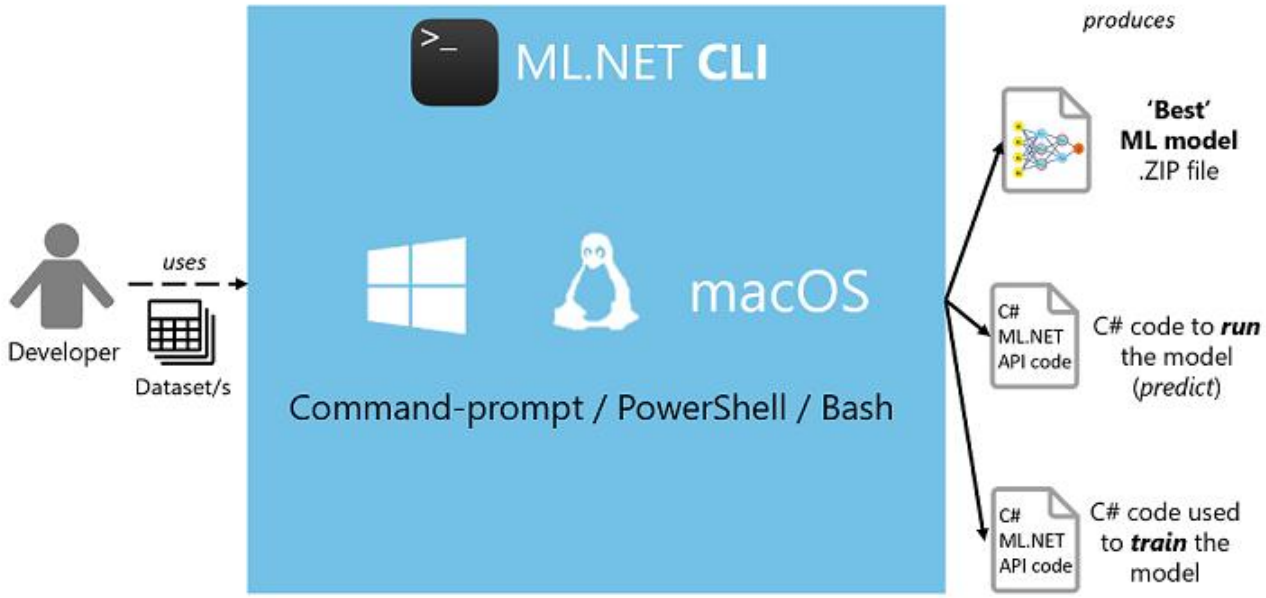

Рис. 4.6 – Принцип роботи ML.NET

Дана реалізація передбачає відсутність будь-яких маркованих даних. Натомість всі ярдики будуть ігноруватися, модель будде самостійно з’ясовувати до якого виду належить квітка.

Модель зможе переглядати шаблони розмірів характеристик зовнішньої та внутрішньої оцвітини та спробує згрупувати всі квіти на три окремих

скупчення, які повинні відповідати трьом типам квітів. Такий тип навчання зветься «навчання без учителя».

В Додатку А Program.cs клас IrisData включає інформацію з характеристиками однієї квітки ірису. Кожне поле класу має атрибути для орієнтації в файлі CSV. Також тут міститься клас для передбачення - ClusterPrediction.

Наступним кроком буде проведення навчання та побудови її моделі, що показано в тому ж файлі. Цей крок проводиться після зчитування інформації. Моделі машинного навчання в ML.NET побудовані на основі конвеєр, в послідовності: завантаження даних, перетворення та навчання.

Наш конвеєр буде складатися з двох компонентів:

- Зв'язування - поєднує всі стовпці вхідних даних в один стовпець, який називається Особливості. Це необхідний крок, оскільки ML.NET може навчатись лише на одній вхідній колонці.
- KMeans - навчальна кластеризація, яка спробує знайти 3 різні кластери в наборі даних.

Коли конвеєр повністю зібраний ми можемо розпочати навчання моделі і встановлення даних для наступного пошуку, до якого з знайдених кластерів відноситься вхідна інформація.

Після виконання програми ми отримуємо наступний результат:

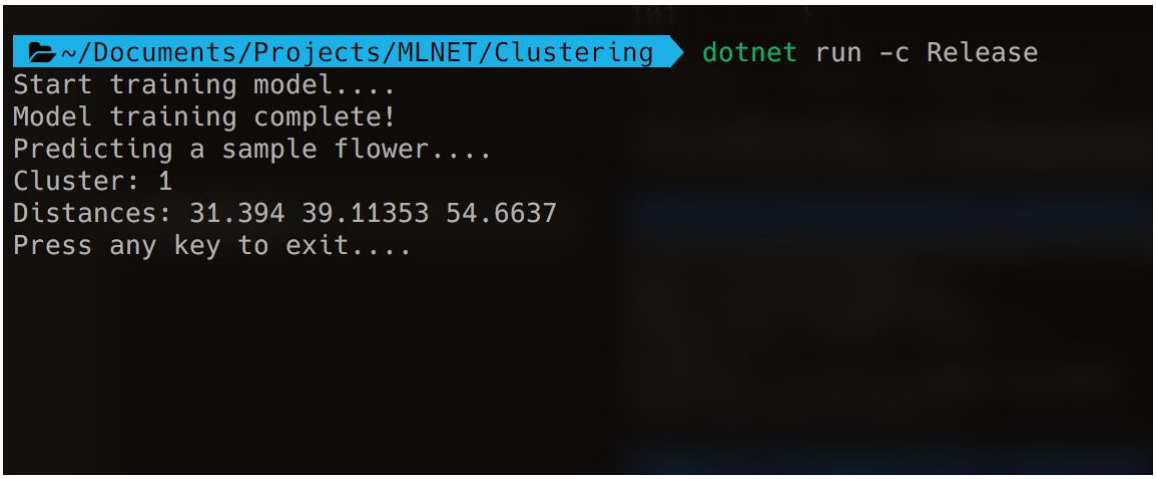

Рис. 4.7 – Результат роботи нашої програми

Для кожного кластера модель обчислює центральну точку або центроїд. Її можна уявити, як ідеальну квітку цього конкретного виду. Зазначені відстані - це те, наскільки далеко задана пробна квітка знаходиться від трьох скупчень центральних квітів. Ми отримали відстані 31.39, 39.11 та 54.66, це означає, що наша пробна квітка найближча до центру кластеру номер 1, тобто ірис setosa.

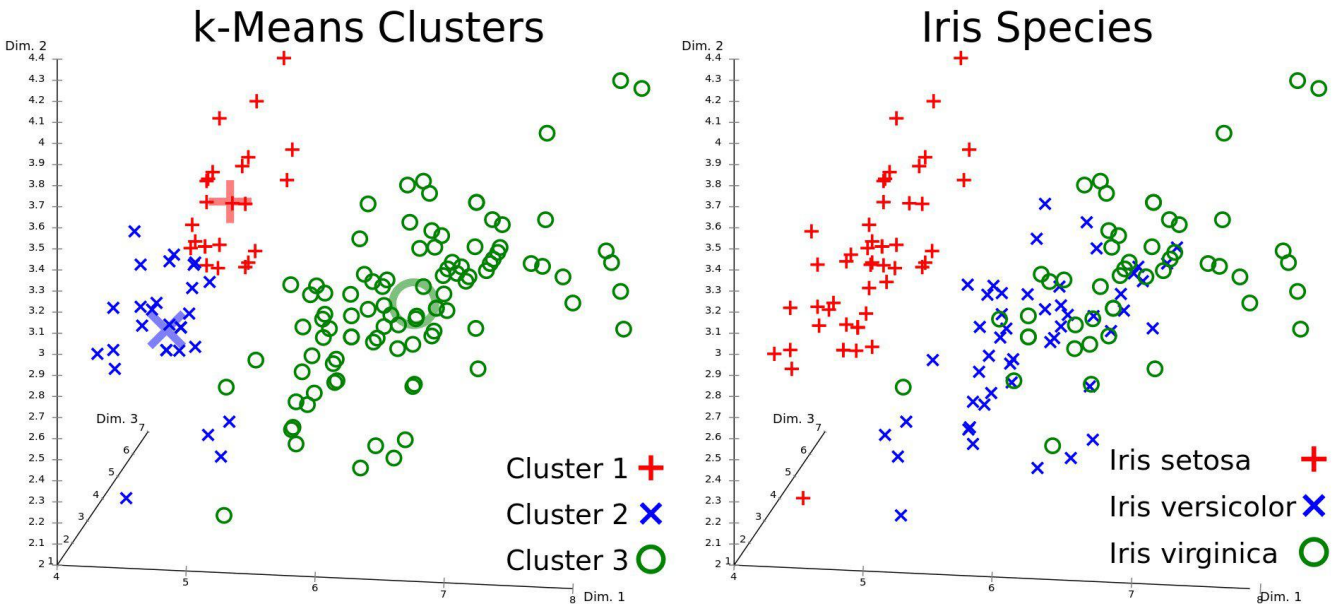

Рис. 4.8 – Розподілення квітів після використання K-Means та початкові дані

Модель виділила три окремі скупчення квітів: зелений, синій та червоний маркери на графіку. Великим плюсом, хрестом і колом помічені три центри кожного скупчення.

Отже, це був приклад навчання без учителя. Під час тренувань модель не мала інформації у формі мічених об’єктів, вона цілком самостійно розібралася з трьома видами квітів. Модель не уявляє, який тип нашої тестової квітки, але вона може визначати, що вона знаходиться до кластера 1.

Для полегшення роботи користувача можна представити нашу програму у веб форматі:

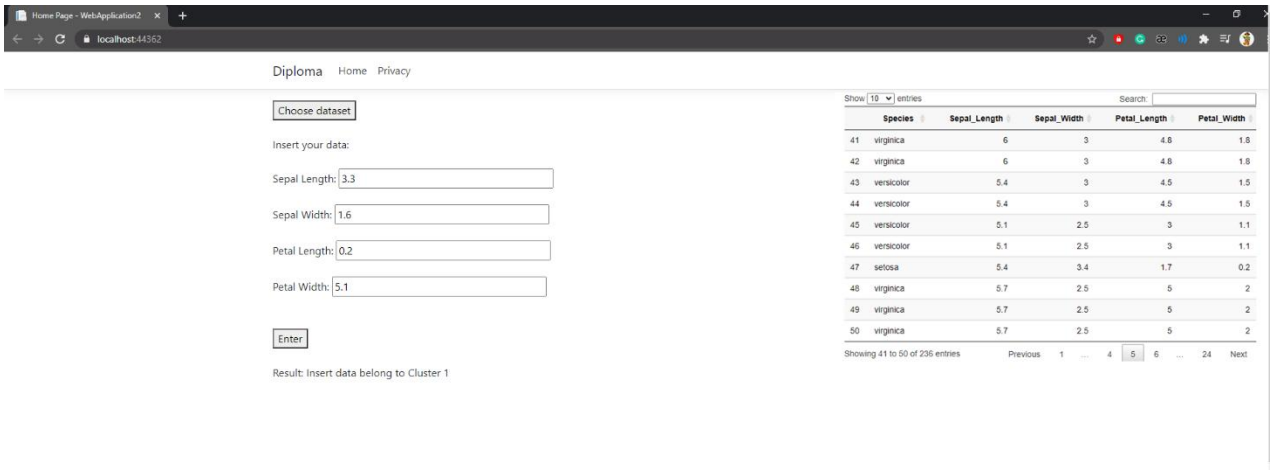

Рис. 4.9 – Приклад можливой реалізації у формі веб-сторінки

Для перевірки точності результатів ми спробуємо провести перевірку на більш реальних даних. Для тестування ми обрали набор даних захворювання на рак молочної залози. Ці структуровані дані представлені у відкритому доступі, а також до них можна доступитися за допомогою SPARQL.

Початковий набір представлений великою кількістю характеристик:

- radius;
- texture**;**
- perimeter;
- area;
- smoothness;
- compactness;
- concavity;
- concave points;
- symmetry;
- fractal dimension.

Для нашого дослідження ми обмежимо трьома основними характеристиками: radius, area, smoothness, оскільки в описі датасету було вказано про важливість цієї інформації [15].

В нашому випадку буде представлено три кластери, для нас вони будуть означати ступінь та наслідки хвороби за цими трьома характеристиками. Сюди входять: фатальний наслідок, тяжкий вплив на здоров'я та мінімальний.

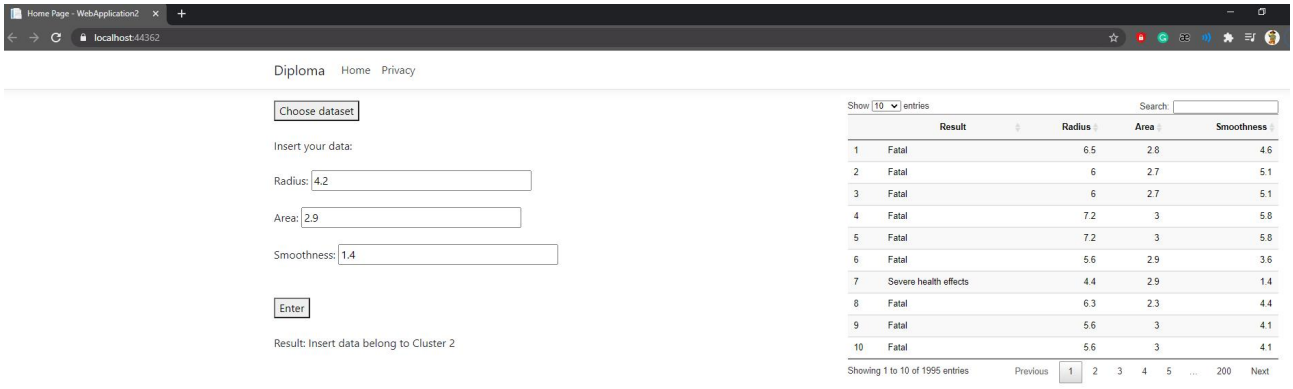

Рис. 4.10 – Результати другого тестового набору

З результатів виконання програми можна зробити висновок, що програма змогла побудувати моделі та класифікувала об'єкт з вхідними параметрами як частина другого кластеру, тобто значний вплив на здоров'я людини. Потенційно цієї інформацією можуть скористуватися медичні працівники та скорегувати або вибрати певну стратегію лікування, щоб зберегти здоров'я пацієнтки.

4.2.2 Реалізація без сторонніх інструментів

Оскільки ML.NET відноситься до сучасних інструментів, може виникнути проблема для відтворення нашої програми в середовищі з старішим C#/.NET. Як альтернативу можна запропонувати, власну реалізацію.

Спершу опишемо сам алгоритм K-means. Центральним поняттям алгоритму k-means є центроїд. У кластеризації даних центроїд набору даних - це той об'єкт, який є найбільш репрезентативним для конкретної групи. Ідею найкраще пояснити на прикладі. Припустимо, у нас є три кортежи з висотою і вагою:

[a] (61.0, 100.0)
[b] (64.0, 150.0)
[c] (70.0, 140.0)

Одним із підходів визначення найбільш репрезентативного з них є обчислення математичного середнього кортежу, а потім в якості центроїда вибирається кортеж, який є найближчим до цього середнього кортежу. Отже, у цьому випадку середній кортеж:

[m] = ((61.0 + 64.0 + 70.0) / 3, (100.0 + 150.0 + 140.0) / 3) = (65.0, 130.0)

А тепер, потрібно визначити який із трьох кортежів є найближчим до (65,0, 130,0). Найпоширенішим підходом, який ми будемо використовувати, є використання евклідової дистанції. На словах, евклідова відстань між двома кортежами - це квадратний корінь із суми квадратних різниць між кожним компонентом кортежів. Знову ж таки, приклад - найкращий спосіб пояснити. Евклідова відстань між кортежем (61,0, 100,0) і середнім кортежем (65,0, 130,0) становить:

dist(m,a) = sqrt((65.0 - 61.0)^2 + (130.0 - 100.0)^2) = 30.27
dist(m,b) = sqrt((65.0 - 64.0)^2 + (130.0 - 150.0)^2) = 20.02
dist(m,c) = sqrt((65.0 - 70.0)^2 + (130.0 - 140.0)^2) = 11.18

Оскільки найменша з трьох відстаней - це відстань між середнім [m] та [c], центроїд трьох кортежів - кортеж [c]. Приклад псевдокоду алгоритму знаходиться в Додатку А KmeansPsevdo.txt

Повну реалізацію даного методу можна розглянути в Додатку А KMeansProgram.cs. Також варто відмітити, що реалізація методу могла б бути і на інших мовах, наприклад Python, в якійсь мірі така реалізація була б простішою через наявність необхідних бібліотек методів, але оскільки ми розглядали попередню реалізацію в контексті мови програмування C#, тому було доцільно показати приклад реалізації саме на ній.

4.3 Альтернативні розширення SPARQL

Під час проведення дослідження, ми виявили наявність розробок в області розширення SPARQL, які б могли допомогти нам з раннім аналізом або відсіюванням даних, якраз та робота яку ми виконуємо на етапі реалізації продукту на мові C#. Оскільки такі підходи можуть представляти цінність під час реалізації продукту, ми вирішили за потрібне описати їхню роботу.

4.3.1 SPARQL-ML

SPARQL-ML (SPARQL для машинного навчання) - це розширення SPARQL, яке розширює мову запитів семантичної мережі інструментами виявлення знань. Це розширення додає нові елементи синтаксису та семантику до офіційної граматики SPARQL. Коротко, SPARQL-ML полегшує виконання наступних двох завдань у будь-якому наборі даних семантичної мережі: тренування, вивчення, формування моделі на основі навчальних даних за допомогою нового висловлення CREATE MINING MODEL та застосувати модель для прогнозування за допомогою двох нових функцій властивостей. Модель будується на кроці CREATE MINING MODEL.

SPARQL-ML існує як розширення до ARQ - механізм запитів SPARQL для Jena. Поточна версія SPARQL-ML підтримує, але не обмежується Proximity3 та Weka4 як модулі інтелектуального аналізу даних.

SPARQL-ML дозволяє формувати класифікатор (модель) з будь-якими навчальними даними Семантичної Мережі, використовуючи нову інструкцію CREATE MINING MODEL. Обраний синтаксис був зроблений на основі розширення Microsoft (DMX), яке є розширенням SQL для створення та роботи з моделями інтелектуального аналізу даних у Microsoft SQL Server Analysis Services (SSAS).

Розширена граматика SPARQL представлена на рис. 4.9, а в Додатку А SparqlMlSample наведено конкретний приклад запиту. Цей підхід додає символ CreateQuery до офіційного правила граматики SPARQL Query. Структура CreateQuery нагадує структуру SelectQuery, але має повністю різну семантику:

[1]	*Query*	::=	Prologue(SelectQuery | ConstructQuery | DescribeQuery | AskQuery | CreateQuery)
[2]	*CreateQuery*	::=	CREATE MINING MODEL' SourceSelector '{' Var 'RESOURCE' 'TARGET' (Var ('RESOURCE' | 'DISCRETE' | 'CONTINUOUS') 'PREDICT'?)+ '}' DatasetClause* WhereClause SolutionModifier UsingClause
[1.2]	*UsingClause*	::=	'USING' SourceSelector BrackettedExpression

Рис. 4.11 – Розширення додані в SPARQL-ML

CreateQuery додає нові ключові слова CREATE MINING MODEL до граматики з подальшим SourceSelector для визначення імені моделі тренування. У CreateQuery перелічені змінні (атрибути) для навчання моделі. Кожна змінна вказується зі своїм типом вмісту, який на даний момент є одним із наступного: RESOURCE - змінна містить ресурс RDF (IRI або порожній вузол), DISCRETE - змінна містить дискретне або номінальне значення літералу, CONTINUOUS - змінна містить безперервне літеральне значення, і PREDICT — повідомляє алгоритму навчання, що цю функцію слід передбачити. Перший атрибут додатково вказується за допомогою ключового слова TARGET для позначення ресурсу, для якого слід передбачити функцію. UseClause розширення, яке додає

нове ключове слово USING, за яким слідує SourceSelector - визначає назву та параметри навчального алгоритму.

4.3.2 SPARQL RAF

Більшість існуючих підходів до кластеризації даних RDF передбачає прямий доступ до даних, однак часто не бажано передавати весь набір даних на локальне середовище, оскільки: це вимагає високих витрат швидкості інтернету, навіть якщо набір даних буде переданим, він може не поміститися в пам'яті. Тому, цікавим питанням є кластеризація даних RDF без прямого доступу до даних, наприклад алгоритм отримує лише результати кластеру або інші статистичні дані. Для вирішення цього питання можна розглянути RAF - remote access framework, його структура зображена на рис. 4.10, його можна застосовувати з нашим завданням кластеризації. Цей фреймворк розширює інтерфейс SPARQL з доступом до запису на попередньо виділений простір сховища даних, тобто пісочниці. Кожен клієнт може зареєструвати сеанс, який отримує доступ до запису набору тимчасових графіків. Потім клієнти

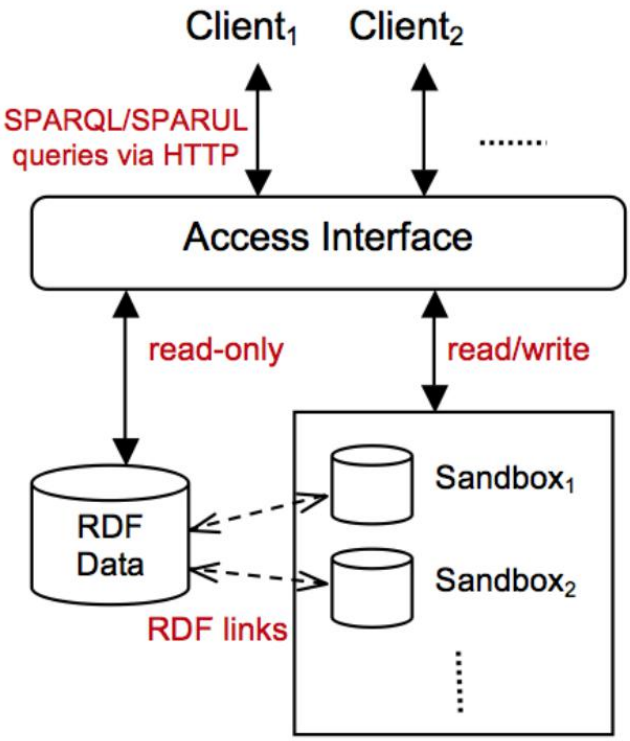

Рис. 4.12 – Схематичне зображення роботи RAF

можуть задавати запити на оновлення SPARQL до інтерфейсу доступу та встановлювати корисні посилання RDF на інші дані RDF для аналізу.

4.4 Висновки

В цьому розділі ми розібрали реалізацію прототипу програми, яка дозволить кластеризувати вхідну інформацію з семантичного вебу.

В першій частині ми сфокусувалися на описі SPARQL та як ми зможемо отримати навчальну інформацію для наступних кроків кластеризації та аналізу вхідних об’єктів. Також в останній частині ми повернулися до мови SPARQL, щоб запропонувати альтернативні шляхи вирішення проблеми отримання інформації. Наприклад SPARQL RAF, фреймворк на основі SPARQL, дозволить розробнику економити пам’ять на етапі отримання інформації, замість отримання надлишкової інформації для своєї програми. SPARQL ML – дозволить розробнику створити навчальну модель перед виконанням основної програми.

Наступним кроком став розгляд підходів кластеризації. Оскільки основна програма розглядалася в контексті мови C#, як основний інструмент був запропонований ML.NET – це розширення від Microsoft, яке в змозі аналізувати вхідну інформацію та проводити, так зване, «навчання без вчителя». Такий підхід дозволив нам отримати три кластери та ідентифікувати їхні центроїди, тобто найбільш репрезентативні точки відповідних кластерів. Далі ми розібрали як за допомогою алгоритму K-means можна знайти до якого кластеру належить вхідна інформація. Також ми розглянули випадок, коли ми не можемо застосовувати даний інструмент і запропонували вирішення простими програмними методами.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

У даному розділі описано економічне обґрунтування реалізації стартап проекту на тему “Використання семантики і формалізованих знань в інтелектуальній обробці даних”. Описана технологія буде реалізована у вигляді веб додатку, тому нею можна буде користуватися у будь-який зручний час. Процес включає:

- Розробка застосунку згідно обраної теми;
- Побудова стратегії конкурентного продукту, що зможе протистояти реаліям ринку.

Метою розділу є поліпшення наших знань в побудові стартапу, навчитися вирішувати реальні проблеми та навчитися правильно оцінювати ринок, для успішного випуску продукту.

Таблиця 5.1 - суть ідеї стартапу

Зміст ідеї	Напрямки застосування	Переваги для користувача
Веб додаток, що зможе допомгти під час дослідницької діяльності у класифікації об’єктів	Підвищення швидкості виконання роботи під час пошуку інформації	Відпадає необхідність у використанні додаткового пз, достатньо онайомитися з простим інтерфейсом
	Полегшення роботи науковців	Полегшення роботи з інформацією

Звідси можемо зробити висновок, що застосунок буде використаним для автоматизації роботи дослідників в процесах, коли відбувається класифікація об’єктів.

Таблиця 5.2 - Нейтральні, слабкі та сильні характеристики ідеї застосунку

№ п\п	Техніко-економічні характеристики	Потенційні конкуренти				W слабка сторона	N нейтральна сторона	S сильна сторона
		Мій проект	К-т 1	К-т 2	К-т 3			
1.	Форма виконання	Веб-сервіс, додаток	Додаток	Веб-сервіс	Веб-сервіс			+
2.	Собівартість	Низька	Висока	Середня	Середня			+
3.	Наявність адміністратора для налаштування	Не треба	Не треба	Потрібно	Потрібно			+
4.	Наявність інтернету	Треба	Треба	Треба	Не треба		+	
5.	Крос-платформеність	Так	Так	Ні	Ні	+		

Побудова таблиці сильних та нейтральних характеристик та властивостей ідеї потенційного товару являється початком для розробки нашого плану успішної конкурентоспроможності. Сильною стороною даного проекту є форма виконання у веб-сервісу та локального додатку, а також використання .NET Core, що забезпечать кросплатформеність.

До негативних сторін можна віднести необхідність використання інтернету, але така поведінка передбачена темою проекту і бажанням уникнути попереднього завантаження інформації на локальну машину.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Таблиця 5.3 - технологічний аудит проекту

№ п\п	Ідея проекту	Технології	Наявність технології	Доступність технології
1	Створення веб-сервісу	.NET Core, SPARQL	Наявна	Безкоштовна, доступна
		Python, SPARQL	Наявна	Безкоштовна, доступна
		SPARQL ML	Наявна	Частково доступна
Для створення веб-сервісу обрані технології (.NET Core, SPARQL), які є безкоштовними, доступними та добре дослідженими потенційними розробниками.				

Ми навели три шляхи рішення нашої програми, але основним було обрано перший - .NET Core, SPARQL, оскільки не потребує додаткових витрат.

5.3 Аналіз ринкових можливостей

Перед розробкою та початком продаж надзвичайно важливо розглянути розміри ринку та можливості потенційних конкурентів. Оскільки таке дослідження допоможе сформувати об’єктивну картину ринку, це поліпшить початкові позиції застосунку у порівнянні з існуючими конкурентами, а також допоможе заощадити ресурси компанії.

Проводимо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку:

Таблиця 5.4 - Базова характеристика ринку стартапу

№ п\п	Показники ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум. од	14000 грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (характер)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Конфіденційність оброблюваних даних
6	Середня норма рентабельності в галузі (або по ринку), %	R = (3000000 * 100) / (1000000 * 12) = 25%

З результатів бачимо, що середня норма рентабельності в галузі менша, ніж банківський відсоток на вкладення. Це означає, що доцільно влкасти кршти в даний проект, оскільки для нього відсутні обмеження для входу на ринок і нестандартних вимог до стандартизації та сертифікації, бо він буде виконаний у вигляді веб сервісу.

Наступним кроком буде оцінка потенційних груп клієнтів, їхні характеристики та формування орієнтовних переліків вимог до товару для кожної групи

Таблиця 5.5 - Характеристика потенційних клієнтів

№ п\п	Формуюча ринок потреба	Цільова аудиторія	Відмінності у поведінці різних цільових груп	Вимоги споживачів до продукту
1	Система сервісів, що спрощує процес аналізу даних	Дослід-ницькі центри різної спрямованості, такі як медецина, економіка та ін.	Цільова група займається отриманням інформації	Зручний інструмент для полегшення роботи

Визначено характеристики стартап-проекту: основну потребу, що формує ринок – веб сервіс; наведено основні цільові сегменти ринку - дослідницькі центри різного спрямування; відмінності у поведінці різних потенційних цільових груп клієнтів – отримання інформації; затверджено основні вимоги до споживачів – сервіс повинен залишатися зручним у використанні.

Таблиця 5.6 - Аналіз загроз

№ п\п	Фактор	Зміст загрози	Можлива реакція компанії
1	Динаміка ринку	Уповільнення росту ринку	Співпраця з іншими компаніями
			Розширення на інші ринки
2	Конкуренція	Вихід на ринок великої компанії	Акцентувати увагу на надійність нашої системи
3	Технології	Поява нових технологій	Програма передбачає розширення, потрібно розробити новий функціонал
4	Держава	Збільшення податків	Оптимізація діяльності для скорочення витрат

Основними проблемами та загрозами стартапу можуть бути: динаміка ринку, поява нового конкуренту, впровадження нових технологій та зміна політики держави. Для кожної з наведених вище проблем ми пропонуємо власні рішення.

Таблиця 5.7 - Фактори можливостей

№ п\п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання на ринку покупців	Поява великої кількості бажаючих використовувати подібні продукти	Проведення рекламної компанії для заохочення нових користувачів
2	Зниження ефективності конкурента 1	Конкурент 1 втрачає робото спроможність	Переконання покупців в ефективності нашого веб-сервісу
3	Зменшення витрат на технічну підтримку	Збільшення продуктивності роботи штату компанії за рахунок підвищення їхнього професійного рівня	Підвищувати рівень кваліфікації своїх співробітників

Також ми з’ясували можливі фактори під час зміни ринку та поведінки конкурента: зростання покупців на ринку, зниження ефективності конкурентів, зменшення витрат на технічну підтримку.

Далі проведем аналіз пропозицій:

Таблиця 5.8 - Ступеневий аналіз конкуренції на ринку

Назва характеристики	Особливість конкурентного середовища	В чому проявляється характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип	Досконале	Існує 3 фірми - конкуренти	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
Рівень конкурентної боротьби	Міжнародне	Два зарубіжних конкуренти	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
Галузева ознака	Фокус на галузі	Конкуренти спеціалізуються в одній галузі	Створити основу ПЗ таким чином, щоб можна було легко його переробити для використання у інших галузях

Вид товарів	Товарно-видове	Однаковий вид товарів (ПЗ) і послуг (сфера медицини)	Створити ПЗ, враховуючи недоліки конкурентів
Характер конкурентних переваг	Нецінове	Вдосконалення технології створення ПЗ, для зменшення його собівартості	Використання менш дорогих технологій, більш ефективних методологій
Інтенсивність	Немарочне	Бренди відсутні	-

Був проведений аналіз конкурентних ступенів на ринку. Змогли визначити, що конкуренція знаходиться на міжнародному рівні і конкуренти мають сильні позиції, це потрібну буде врахувати при формуванні наших цін. Також з'ясували, що конкуренти спеціалізуються на конкретних галузях, це можна зробити особливістю нашого продукту.

Далі розробляється перелік факторів конкурентоспроможності для ринку на основі аналізу складових моделі 5 сил М. Портера.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Фактор сили постачальників	Фактор сили споживачів	Фактор загроз з боку товарів-замінників
Висновки	Існує 3 конкуренти на ринку. Найбільш схожим являється конкурент 1, має кросплатформеність та відсутність адміністрування	Так, можливості для входу на ринок є, бо наше рішення спрощує та пришвидшує роботу спеціаліста	Постачальники відсутні	Важливим для користувача є зручність у користуванні	Товаризамінники можуть використати більш дешеву технологію створення додатку та зменшити собівартість товару

Таким чином, з огляду на ситуацію з конкурентами, можна з упевненістю сказати, що проект має можливість працювати на ринку, тому що серед цих конкурентів немає тих, хто міг би спростити роботу як наше, бо розроблене рішення спрощує і прискорює роботу фахівця. Ми можемо визначити основні сильні сторони продукту, які допоможуть стати конкурентоспроможними на ринку, - надійність, зручність у використанні, простота використання і безпеку.

На основі аналізу висновків, наведених вище, визначається та обґрунтовується перелік факторів конкурентоспроможності

Таблиця 5.10 - Обгрунтування факторів конкурентоспроможності

№ п\п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Реалізація як локального додатку та веб сайту	Дозволить використовувати службу незалежно від роботи веб-додатку
2	Простота користувацького інтерфейсу	Доступ відбувається через веб-сайт або додаток на пк зі схожим інтерфейсом

Основні фактори конкурентоспроможності, які будуть представлені на ринку: реалізація на двох платформах, тому що розробка користувальницького інтерфейсу не припускає зайвих складнощів, тому основна частина програми залишається незмінною, а також простота призначеного для користувача інтерфейсу полегшить освоєння програми.

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту.

Таблиця 5.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п\п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів конкурентів у порівнянні з нашим						
			-3	-2	-1	0	+1	+2	+3

1	Реалізація як локального додатку та веб сайту	20			+				
2	Простота користувацького інтерфейсу	15	+						

Був проведений порівняльний аналіз сильних сторін проекту конкуруючих продуктів та нашої компанії. Найвищі бали набираються за такі фактори конкурентоспроможності - наявність двох типів додатків, як на веб-служби, так і локальної служби

.

Таблиця 5.12 - SWOT-аналіз стартапу

S	Простий користувацький інтерфейс, надійність, безпечність	Необхідність доступу до інтернету	W
O	Падіння конкурентоспроможності конкурента, успішна реклама з акцентом на надійність	Збільшення конкуренції, зміна пріоритетів користувачів, податки	T

Таблиця SWOT допомогла нам накопичити всі плюси і мінуси стартапу. Позитив: інтерфейс, безпека та надійність. Слабкі сторони: для роботи запитів SPARQL потрібен доступ до Інтернету. З можливостей: реклама та помилки конкурентів. Загрози: перерозподіл ринку за конкурентами, скорочення ринку.

На основі SWOT-аналізу складаються альтернативи ринкового впровадження стартап-проекту.

Таблиця 5.13 - Альтернативи ринкового впровадження стартап-проекту

№ п\п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створення ПЗ	80%	6 місяців

	використовуючи машинне навчання		
2	Створення ПЗ використовуючи власні методи	30%	12 місяців

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими.

5.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.14 - Вибір цільових груп потенційних користувачів

№ п\п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	Лабораторії	Можливість автоматичної категоризації, що зменшує роботу людей	Великий	Існує 3 конкуренти, які надають схожі, але менш швидкі рішення.	Швидкодія, зручний користува-цький інтерфейс, точність класифікації
2	Статистичні центри	Можливість автоматичної категоризації, що зменшує роботу людей	Великий		Швидкодія, зручний користува-цький інтерфейс, точність класифікації
Які цільові групи обрано: лабораторії та статистичні центри					

Результати розгляду цільових груп потенційних споживачів стартап-проекту, наведені в таблиці 5.14, дозволяють визначити, які переваги продукту

можна використовувати для виходу в цей сегмент ринку, а також доцільно витрачати ресурси на вплив на ту чи іншу групу. .

За результатами аналізу потенційних груп (сегментів) споживачів були обрані дві основні цільові групи, яким буде запропоновано проект до використання. Для ефективного впровадження продукту в обраних групах необхідно розробити стратегію охоплення ринку.

Таблиця 5.15 - Визначення базової стратегії розвитку

№ п\п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення ПЗ використовуючи машинне навчання	Ринкове позиціонування	Швидкодія, простота у користуванні, безпечність, надійність	Диференціація

Було обрано таку альтернативу розвитку проекту: створення ПЗ використовуючи машинне навчання

Таблиця 5.16 - Визначення базової стратегії конкурентної поведінки

№ п\п	«Першопроходець» на ринку	Агресивний пошук нових споживачів	Копіювання основних характеристик в конкурент	Стратегія конкурентної поведінки
1	Ні	Так	Буде, а саме: основною задачею є розробка ПЗ з використанням машинного навчання (конкуренти 1, 2, 3), простий інтерфейс користувача (конкурент 2)	Зайняття конкурентної ніші

.

Отже, було визначено базову стратегію конкурентної поведінки - зайняття конкурентної ніші.

Далі визначається стратегія позиціонування проекту, яка допоможе користувачам ідентифікувати програмний продукт

Таблиця 5.17 - Визначення стратегії позиціонування

№ п\п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Простота інтерфейсу, швидкодія, точність результатів	Диференціація	Простота користувацького інтерфейсу дозволить отримувати необхідні дані і відслідковувати події в режимі реального часу	Швидкодія, безпека, простота, точність результатів

Таким чином, було визначено стратегію позиціонування, а саме визначено основні вимоги до продукту цільової аудиторії: простота інтерфейсу, швидкість, надійність, безпеку; основна стратегія розвитку: диференціація; Ключові конкурентні позиції стартап-проекту: простота призначеного для користувача інтерфейсу дозволить отримувати необхідні дані і відстежувати події в режимі реального часу, а також буде безпечним і надійним. Також сформована комплексна позиція проекту: швидкість, безпека, простота.

5.5 Розробка мартекингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 5.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18 - Визначення ключових переваг концепції потенційного товару

№ п\п	Потреба	Вигода, яку пропонує продукт	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкодія	ПЗ працює досить швидко, результат можна отримати до 10 мс	Перевага у швидкості
2	Простота користувацького інтерфейсу	Простота роботи додатку	Користувачі мають зручний інтерфейс для взаємодії з додатком

Таким чином, ми бачимо, що проект має велику кількість переваг перед конкурентами, які повністю відповідають потребам цільової аудиторії. Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього в табл. 5.18 необхідно узагальнити результати попереднього аналізу конкурентоспроможності товару.

Далі в таблиці 5.19 проілюстровано трирівневу маркетингову модель товару: зазначено ідею товару та / або послуги, її фізичні компоненти та особливості процесу її надання.

Таблиця 5.19 - Трирівнева модель товару

Рівні товару	Сутність та складові

1. Товар за задумом	Додаток дозволяє проводити аналізи на основі структурованої інформації		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Зручність та простота користувацького інтерфейсу	Нм	Технологічна
	2. Швидкість роботи	Нм	Технологічна
	3. Безпека згідно до світових стандартів	Нм	Технологічна
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє		
	Моя компанія: “Future methods”		
3. Товар із підкріплюнням	1-місячна пробна безкоштовна версія		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: патент			

Було описано три рівні моделі товару, з чого можна зробити висновок, що основні властивості товару у реальному виконанні є нематеріальними та технологічними. Також було надано сутність та складові товару у задумці та товару з підкріпленням.

Наступним кроком є визначення цінових меж (табл. 5.20). Аналіз проводиться експертним методом.

Таблиця 5.20 - Визначення меж встановлення ціни

№ п\п	Рівень цін на	Рівень цін на	Рівень доходів	Верхня та

	товари-замінники, грн	товари-аналоги, грн	цільової групи споживачів, грн	нижня межі встановлення ціни на товар/послугу, грн
1	31000	42000	250000	25000-39000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 5.21).

Таблиця 5.21 - Формування системи збуту

№ п\п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Купують підписку та роблять щорічні внески для подовження ліценції	Продаж	0(напряму), 1(через одного посередника)	Власна та через посередників

Отже, було сформовано систему збуту у вигляді щорічної підписки (ліцензії). Збут буде проводитися власними силами та через посередників, напряму та через одного посередника у вигляді продажу товару.

Далі розробляється концепція маркетингових комунікацій (табл. 5.22).

Таблиця 5.22 - Концепція маркетингових комунікацій

№ п\п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Використання за допомогою сайту	Інтернет	Швидкодія, простота у	Показати переваги	Демо-ролик із

			використан ні, безпека	сервісу, у тому числі і перед конкурента-ми	використан ня

Отже, в Таблиці 5.22 наведено концепцію маркетингових комунікацій, було визначено, що придбання ліцензії на користування буде здійснюватись в мережі Інтернет, необхідним буде щомісячне її продовження, користування сервісом можливе у хмарі або ж на власних серверах

5.6 Висновки

У даному розділі ми провели дослідження основних аспектів входу нашого продукту на ринок аналогічних товарів. Поріг входження на ринок не є дуже високим, хоч конкуренція знаходиться на міжнародному рівні Для успішної реалізації були обрані C#/.NET та ML.NET.

Для успішного виконання проекту необхідно реалізувати сервіс із використанням машинного навчання для кластеризації і подальшої класифікації інформації. В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

Під час аналізу ми розрахували головні фінансово-економічні параметри проекту, а також дослідили потенційні ризики проекту. Провели аналіз можливостей проекту в разі змін успішності конкурентів на ринку та, що готові протиставити в таких випадках. В наслідок отриманих результатів можна зробити висновок, що імплементація буде конкурентоспроможною.

Було обрано цільові групи потенційних споживачів, визначено базову стратегію розвитку та стратегії конкурентної поведінки та позиціонування; розроблено маркетингову програму.

Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

6 ВИСНОВКИ

В результаті виконання даної роботи були розглянуті семантичні технології та методи їхньої обробки. Семантичні технології не являються новітніми і вже мають свою реалізацію у формі семантичного вебу та мови запитів до них, але наша робота мала на меті вирішити проблему обробки цих структурованих даних, що було показано в практичній частині. Продуктом виконання цієї роботи є програмна реалізація додатку, що класифікує вхідну інформацію на основі попередньої, що була взята з семантичної павутини.

Для розуміння теми семантики та формалізованих знань, спершу були розглянуті теми семантичної мережі, web mining та отримання інформації через структуру веб сторінок. Також описали стандартний процес виявлення знань та поглибили знання про web mining семантичної мережі через наведений приклад з готелями та ресторанами, які відвідують люди.

Далі ми розглянули тему семантичного data mining. Дослідили роль та значення онтологій відносно цієї теми, а саме дізналися, про сприяння під час подолання семантичного розриву, подову формальної інструкції під час роботи процесу та створення попередніх знань і обмежень. Розібрали приклад системи рекомендації на основі онтології. Привели достатньо велику інформацію про досвід іноземних колег у вирішенні цих проблем. Також дослідили продуктивність сучасних підходів та альтернативні підходи видобутку даних, спираючись на існуючі дослідження та експерименти.

Також важливою частиною роботи став розділ з існуючими та успішними проектами, які спираються на семантику та формалізовані знання для подальшого аналізу інформації. Серед таких, була описана система iASiS, система розроблена Європейськими університетами, для аналізу медичних даних, наприклад аналіз інформації про хворобу, співставлення інформації синонімічної інформації, але з потенційно різним описом. Особливістю можна назвати, що проект може зчитувати інформацію з різноманітних джерел. Інша система NAAS, покликана допомогти у вирішенні проблеми пошуку інформації

для журналістів. В обох випадках розробники та користувачі мали позитивний досвід використання цих систем.

У результаті виконання даної роботи є такі речі:

- Розроблена програма для локального середовища та її альтернатива у форматі веб-сайту на ASP.NET Core, покликана використовуватися багатьма користувачами як публічний сервіс. Програма може отримувати інформацію за допомогою класичного SPARQL та кластеризувати дану навчальну інформацію для подальшого визначення належності вхідної інформації.
- Інструкція з використання альтернативних версій та розширень для SPARQL, які можуть допомогти розробникам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Stumme G. Semantic Web Mining State of the art and future directions / G. Stumme, A. Hotho., 2005. – 21 с.

2. Vidal M. Semantic Data Integration of Big Biomedical Data for Supporting of Personalised Medicine / Vidal Maria-Esther. – 31 с.

3. Ristoski P. Semantic Web in data mining and knowledge discovery: A comprehensive survey / Ristoski Petar – Mannheim, Germany, 2015.

4. S. Acharyya, J. Ghosh, Context-sensitive modeling of web-surfing behaviour using concept trees, in: Proceedings of the WebKDD Workshop on Web Mining and Web Usage Analysis, 2003

5. NASS: News annotation semantic system / Luis Angel – Boca Raton, FL, USA, 2011. – 18 с.

6. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, V. Zacharias, Kaon - towards a large scale semantic Web, in: K. Bauknecht, A. Min Tjoa, G. Quirchmayr (Eds.), E-Commerce and Web Technologies, Third International Conference Proceedings, EC-Web 2002, vol. 2455 of LNCS, Springer, Berlin, 2002.

7. Qi L. Clustering Remote RDF Data Using SPARQL Update Queries / Qi Letao, 2013. – 8 с.

8. L. Tang and H. Liu, “Community detection and mining in social media,” Synthesis Lectures on Data Mining and Knowledge Discovery, 2010.

9. X. Wang, L. Tang, H. Liu, and L. Wang, “Learning with multi-resolution overlapping communities,” Knowledge and Information Systems, 2012.

10. S. Datta, C. Giannella, and H. Kargupta, “K-Means Clustering over a Large, Dynamic Network,” in Proceedings of 2006 SIAM Conference on Data Mining, April 2006.

11. C. Chen, Information Visualisation and Virtual Environments, Springer, London, 1999.

12. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, Advances in knowledge disc overy and data mining, in: American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

13. O. Peña, U. Aguilera, D. López-de Ipiña, Linked open data visualization revisited: A survey, Semant. Web J.

14. Z. Syed, T. Finin, V. Mulwad, A. Joshi, Exploiting a web of semantic data for interpreting tables, in: Proceedings of the Second Web Science Conference, 2010.

15. G. Solskinnsbakk, J.A. Gulla, Semantic annotation from social data, in: Proceedings of the Fourth International Workshop on Social Data on the Web Workshop, 2011.

16. E. Muoz, A. Hogan, A. Mileo, Triplifying wikipedia's tables, in: LD4IE@ ISWC'13, 2013, 1–1 c.

17. G.K.D. de Vries, S. de Rooij, A fast and simple graph kernel for rdf, in: Proceedings of the Second International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, 2013.

18. A. Schulz, C. Guckelsberger, F. Janssen, Semantic abstraction for generalization of tweet classification.

19. C. Antunes, Onto4ar: a framework for mining association rules, in: Workshop on Constraint-Based Mining and Learning, CMILE-ECML/PKDD 2007, 2007, 37–48 c.

20. C. Diamantini, D. Potena, Semantic annotation and services for kdd tools sharing and reuse, in: ICDM Workshops, 2008, pp. 761–770.

21. N. Lavrač, P.K. Novak, Relational and Semantic Data Mining for Biomedical Research, 2012.

22. L. Cao. Data science: challenges and directions. Commun. ACM, 60(8):59–68, 2017.

23. P. Colombo and E. Ferrari. Privacy aware access control for big data: A research roadmap. Big Data Research, 2(4):145–154, 2015.

24. O. Hartig, M. Vidal, and J. Freytag. Federated semantic data management (dagstuhl seminar 17262). Dagstuhl Reports, 7(6):135–167, 2017.

25. E. Kamateri, E. Kalampokis, E. Tambouris, and K. A. Tarabanis. The linked medical data access control framework. Journal of Biomedical Informatics, 50:213–225, 2014.

26. P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: linked data quality assessment and fusion. In Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, March 30, 2012, pages 116–123, 2012.

27. Boettiger C. A tidyverse lover's intro to RDF [Електронний ресурс] / Carl Boettiger. – 2020. – Режим доступу до ресурсу: https://cran.r-project.org/web/packages/rdflib/vignettes/rdf_intro.html.

28. McCaffrey J. K-Means Data Clustering Using C# [Електронний ресурс] / James McCaffrey. – 2013. – Режим доступу до ресурсу: https://visualstudiomagazine.com/articles/2013/12/01/k-means-data-clustering-using-c.aspx.

29. Farragher M. Easy K-Means Clustering with C# and ML.NET [Електронний ресурс] / Mark Farragher. – 2019. – Режим доступу до ресурсу: https://medium.com/machinelearningadvantage/easy-k-means-clustering-with-c-and-ml-net-7b154ccd219e.

30. Rehman H. Sentiment Analysis In ASP.NET Core Using ML.Net [Електронний ресурс] / Habib Rehman. – 2019. – Режим доступу до ресурсу: https://www.c-sharpcorner.com/article/getting-started-with-sentiment-analysis-with-ml-net/.

31. Real-Time Sentiment Analysis with C# [Електронний ресурс] – Режим доступу до ресурсу: https://www.luisquintanilla.me/2018/01/18/real-time-sentiment-analysis-csharp/.

ДОДАТОК А

Flowers.rq

```
SELECT  ?Species ?Sepal_Length ?Sepal_Width ?Petal_Length ?Petal_Width
WHERE {
 ?s <iris:Species>  ?Species .
 ?s <iris:Sepal.Width>  ?Sepal_Width .
 ?s <iris:Sepal.Length>  ?Sepal_Length .
 ?s <iris:Petal.Length>  ?Petal_Length .
 ?s <iris:Petal.Width>  ?Petal_Width
}
```

Program.cs

```
/// <summary>
/// Клас передачі даних, який містить одну квітку ірису.
/// </summary>
public class IrisData
{
    [LoadColumn(0)] public float SepalLength;
    [LoadColumn(1)] public float SepalWidth;
    [LoadColumn(2)] public float PetalLength;
    [LoadColumn(3)] public float PetalWidth;
    [LoadColumn(4)] public string Label;  // ігнорується під час навчання
}

/// <summary>
/// Клас передбачення, який містить єдине кластерне передбачення.
/// </summary>
public class ClusterPrediction
{
    [ColumnName("PredictedLabel")] public uint PredictedClusterId;
    [ColumnName("Score")] public float[] Distances;
}

/// <summary>
/// Головний клас
/// </summary>
class Program
{
    /// <summary>
    /// Вхідна точка
    /// </summary>
    /// <param name="args"The command line arguments></param>
    static void Main(string[] args)
    {
        var mlContext = new MLContext();

        // зчитування інформації з тектсового файлу
        var trainingData = mlContext.Data.ReadFromTextFile<IrisData>(
            path: "iris-data.txt",
```

```
            hasHeader: false,
            separatorChar: ',');


        // створити навчальний конвеєр
        // крок 1: об'єднати вхідні функції в один стовпець
        var pipeline = mlContext.Transforms.Concatenate(
            "Features",
            "SepalLength",
            "SepalWidth",
            "PetalLength",
            "PetalWidth")

       // крок 2: використання алгоритму k-means
       // припустимо, що існує 3 типи квітів
       .Append(mlContext.Clustering.Trainers.KMeans(
        "Features",
        clustersCount: 3));

        // навчання моделі на файлі даних
        Console.WriteLine("Start training model....");
        var model = pipeline.Fit(trainingData);
        Console.WriteLine("Model training complete!");

        // Передбачення вхідної квітки на характеристиках
        Console.WriteLine("Predicting a sample flower....");
        var prediction = model.CreatePredictionEngine<IrisData,
        ClusterPrediction>(mlContext).Predict(
            new IrisData()
            {
                SepalLength = 3.3f,
                SepalWidth = 1.6f,
                PetalLength = 0.2f,
                PetalWidth = 5.1f,
            });

        Console.WriteLine($"Cluster: {prediction.PredictedClusterId}");
        Console.WriteLine($"Distances: {string.Join(" ",
        prediction.Distances)}");
    }
}
```

KmeansPsevdo.txt

```
assign each tuple to a randomly selected cluster
compute the centroid for each cluster
loop until no improvement or until maxCount
  assign each tuple to best cluster
   (the cluster with closest centroid to tuple)
  update each cluster centroid
   (based on new cluster assignments)
end loop
return clustering
```

KMeansProgram.cs

```
using System;
namespace ClusteringKMeans
{
  class ClusteringKMeansProgram
  {
    static void Main(string[] args)
    {
      try
      {
        Console.WriteLine("\nBegin outlier data detection demo\n");
        Console.WriteLine("Loading all (height-weight) data into memory");
        string[] attributes = new string[] { "Height", "Weight" };
        double[][] rawData = new double[20][];
        Console.WriteLine("\nRaw data:\n");
        ShowMatrix(rawData, rawData.Length, true);
        int numAttributes = attributes.Length;
        int numClusters = 3;
        int maxCount = 30;
        Console.WriteLine("\nk = " + numClusters + " and maxCount = " +
maxCount);
        int[] clustering = Cluster(rawData, numClusters, numAttributes,
maxCount);
        Console.WriteLine("\nClustering complete");
        Console.WriteLine("\nClustering in internal format: \n");
        ShowVector(clustering, true);
        Console.WriteLine("\nClustered data:");
        ShowClustering(rawData, numClusters, clustering, true);
        double[] outlier = Outlier(rawData, clustering, numClusters, 0);
        Console.WriteLine("Outlier for cluster 0 is:");
        ShowVector(outlier, true);
        Console.WriteLine("\nEnd demo\n");
      }
      catch (Exception ex)
      {
        Console.WriteLine(ex.Message);
      }
    } // Main
    // 14 short static method definitions here
  }
}

static void UpdateMeans(double[][] rawData, int[] clustering,
  double[][] means)
{
  int numClusters = means.Length;
  for (int k = 0; k < means.Length; ++k)
    for (int j = 0; j < means[k].Length; ++j)
      means[k][j] = 0.0;
```

```
  int[] clusterCounts = new int[numClusters];
  for (int i = 0; i < rawData.Length; ++i)
  {
    int cluster = clustering[i];
    ++clusterCounts[cluster];
    for (int j = 0; j < rawData[i].Length; ++j)
      means[cluster][j] += rawData[i][j];
  }
  for (int k = 0; k < means.Length; ++k)
    for (int j = 0; j < means[k].Length; ++j)
      means[k][j] /= clusterCounts[k]; // danger
  return;
}

static int[] Cluster(double[][] rawData, int numClusters,
  int numAttributes,  int maxCount)
{
  bool changed = true;
  int ct = 0;
  int numTuples = rawData.Length;
  int[] clustering = InitClustering(numTuples, numClusters, 0);
  double[][] means = Allocate(numClusters, numAttributes);
  double[][] centroids = Allocate(numClusters, numAttributes);
  UpdateMeans(rawData, clustering, means);
  UpdateCentroids(rawData, clustering, means, centroids);
  while (changed == true && ct < maxCount)
  {
    ++ct;
    changed = Assign(rawData, clustering, centroids);
    UpdateMeans(rawData, clustering, means);
    UpdateCentroids(rawData, clustering, means, centroids);
  }
  return clustering;
}
```

SparqlMlSample

```
CREATE MINING MODEL <http://www.example.org/projectSuccess >
{
      ?project RESOURCE TARGET
      ?success DISCRETE PREDICT {'YES', 'NO'}
      ?member RESOURCE
      ?class RESOURCE
}
WHERE
{
      # SPARQL Basic Graph Pattern (BGP) matching part (lines 9-11)
      ?project ex: isSuccess ? success .
```

```
      ? project ex : hasTeam ? member .
      ? member rdf : type ? class .
} USING <http://kdl.cs.umass.edu/ proximit
```