Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Автоматизований метод побудови онтологічних баз знань
(тема)

Виконав:
студент 2 курсу, групи СШІм-20-1
Іванісенко Є.В.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник доц. Золотухін О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Іванісенку Євгенію Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизований метод побудови онтологічних баз знань

затверджена наказом університету від 8 листопада 20 21 р. № 1695Ст
2. Термін подання студентом роботи до екзаменаційної комісії _____ _____ 20 21 р.
3. Вихідні дані до роботи Теоретичні відомості про онтологію, про метод видобування знань, про підходи: засновані на лексико-синтаксичних шаблонах, на основі системи продукції та на основі статичних методів

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної області і постановка задачі
2) Принципи побудови онтологічних баз знань
3) Автоматизовані методі побудови онтологічних баз знань
4) Практичне використання результатів дослідження

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	к.т.н., доцент Золотухін О.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання на кваліфікаційну роботу	01.09.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	02.09.2021–08.09.2021	виконано
3.	Опрацювання літератури та аналіз об'єкту	09.09.2021–15.09.2021	виконано
4.	Вибір програмних засобів для розробки системи	16.09.2021–18.09.2021	виконано
5.	Розробка програмного засобу	19.09.2021–05.11.2021	виконано
6.	Аналіз отриманих результатів	06.11.2021–16.11.2021	виконано
7.	Оформлювання пояснювальної записки	17.11.2021–24.11.2021	виконано
8.	Проходження нормоконтролю	25.11.2021–28.11.2021	виконано
9.	Оформлення презентаційних матеріалів	29.11.2021–01.12.2021	виконано
10.	Попередній захист	02.12.2021	виконано
11.	Представлення кваліфікаційної роботи		

Дата видачі завдання 1 вересня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 67 с., 12 рис., 1 дод., 28 джерел.

БАЗА ЗНАНЬ, ЕЛЕМЕНТ, ЗНАННЯ, МЕТОД, ОНТОЛОГІЯ, ТЕКСТОВИЙ ДОКУМЕНТ, PROTÉGÉ.

Об'єктом дослідження є створення програмного засобу, який автоматично заповнює онтологічну базу знань результатами роботи алгоритму видобування знань.

Метою кваліфікаційної роботи є створення елементів, які дозволяють видобувати знання та автоматично будувати онтологічні бази знань.

Методами дослідження є алгоритми видобування знань та автоматичної побудови онтологій.

РЕФЕРАТ

Пояснительная записка: 67 с., 12 рис., 1 прил., 28 источников.

БАЗА ЗНАНИЙ, ЗНАНИЯ, МЕТОД, ОНТОЛОГИЯ, ТЕКСТОВЫЙ ДОКУМЕНТ, ЭЛЕМЕНТ, PROTÉGÉ.

Объектом исследования является создание программного средства, которое автоматически заполняет онтологическую базу знаний результатами работы алгоритма получения знаний.

Целью работы является создание элементов, позволяющих получать знания и автоматически строить онтологические базы знаний.

Методами исследования есть алгоритмы добычи знания и автоматического построения онтологий.

ABSTRACT

Explanatory note: 67 p., 12 fig., 1 ann., 28 sources.

ELEMENT, KNOWLEDGE, KNOWLEDGE BASE, METHOD, ONTOLOGY, PROTÉGÉ, TEXT DOCUMENTS.

The object of research is the creation of software that automatically fills the ontological knowledge base with the results of the algorithm for obtaining knowledge.

The aim of the work is to create elements that allow to acquire knowledge and automatically build ontological knowledge bases.

The research methods are algorithms for knowledge extraction and automatic construction of ontologies.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 9
1 Аналіз предметної області та постановка завдання 13
1.2 Автоматична побудова онтологій 15
1.2.1 Підхід на основі лексико-синтаксичних шаблонів 15
1.2.2 Підхід на основі системи продукції 16
1.3 Підхід на основі статистичних методів 17
1.3.1 Попередня підготовка колекції 17
1.3.2 Визначення класів онтології 18
1.3.3 Визначення відносин між класами 20
1.3.4 Оцінка якості збудованої онтології 22
1.4 Класифікація онтологій 22
1.5 Автоматичні методи побудови онтологій 26
1.6 Постановка задачі 32
2 Теоретичні дослідження 33
2.1 Система автоматичного отримання знань з тексту та її застосування 33
2.1.1 Процесор знань: функціональність лінгвістичного процесора 33
2.1.2 Процесор знань: функціональність процесора онтологій 36
2.1.3 Додатки процесора знань 40
2.2 Метод автоматизованого здобуття знань із слабоструктурованих джерел 41
2.2.1 Підходи до отримання даних та знань з Web 42
2.2.2 Метод вилучення інформації із слабоструктурованих джерел, заснований на XML 45
2.2.3 Реалізація методу вилучення 48
2.3 Метод автоматичної побудови онтології предметної галузі на основі аналізу лінгвістичних характеристик текстового корпусу 51
2.3.1 Основні засади побудови онтології 52
2.4 Класифікації онтологій 56

3 Програмна реалізація 59
Висновки 63
Перелік джерел посилання 64
Додаток А Відомість кваліфікаційної роботи 67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АПО – автоматична побудова онтології;

БЗ – база знань;

ІІС – інтелектуальна інформаційна система;

ІС – інформаційна система;

КІС – корпоративна інформаційна система;

ЛПР – лінгвістичний процесор;

МПВ – методи побудови онтологій;

ПЗ – програмне забезпечення;

ППО – принципи побудови онтологій;

ПРЗ – процесор знань;

ТС – текстові збіги.

ВСТУП

У зв'язку з постійним зростанням обсягів інформації, представленої в електронному вигляді, завдання ефективного пошуку великих колекціях текстових документів залишається однією з найактуальніших. Одним із перспективних напрямів підвищення релевантності результатів пошуку є семантичний пошук.

Семантичний пошук – вид повнотекстового автоматизованого інформаційного пошуку, який ведеться з урахуванням змісту слів та словосполучень запиту користувача та текстів інформаційних ресурсів. Основна проблема реалізації семантичного пошуку залежить від необхідності смислового аналізу текстів, тобто. отримання сенсу з тексту та його відображення в деяку формальну модель, що дозволяє визначати близькість пошукового запиту та документа.

Одним із підходів до вирішення завдання «семантизації» пошуку, який на практиці показав свою ефективність, є підхід на основі онтологій. Наприклад, запропонована модель організації пошуку інформації в мережі Internet використовує висновок на онтології для формування більш ефективних запитів і, отже, підвищення релевантності результатів.

Як правило, побудова онтології вимагає використання великих ресурсів, а також експертних знань у предметній галузі і займає суттєвий обсяг часу. Таким чином, автоматизація процесу побудови онтології є актуальним завданням.

Можлива автоматична побудова онтології з колекції текстових документів переважно на основі статистичних методів аналізу текстів природною мовою.

Варто зазначити, що зміст документів у колекції безпосередньо впливає на якість одержуваної онтології. Якщо тематично тексти документів слабко пов'язані, швидше за все, побудована онтологія виявиться невиразною, оскільки описуватиме окремі аспекти різних предметних областей, не створюючи у своїй загальної картини.

Подібна ситуація може виникнути і у разі побудови онтології на основі досить довгого документа, окремі розділи якого належать до різних предметних областей.

З іншого боку, невеликі документи зазвичай змістовно зачіпають лише деяку вузьку частину предметної області, у зв'язку з чим, мабуть, онтологія може бути неповною.

Таким чином, особливу увагу слід приділяти змісту тексту для побудови онтології.

ВСТУП

Розвиток систем електронного документообігу, що супроводжується експоненційним зростанням масивів повнотекстових документів, потребує нових засобів організації доступу та пошуку інформації, багато з яких слід віднести до розряду систем штучного інтелекту – систем обробки знань. Одним із ефективних підходів до виявлення та обробки семантики текстових документів є застосування онтологій.

У зв'язку з цим виникає необхідність розробки елементів, які дозволять отримувати знання, а також автоматично будувати онтологічні бази знань. Для вирішення цієї проблеми необхідно вивчити програмний інструментарій (Protégé – редактор моделювання онтологій), а також розробити алгоритм автоматичної побудови онтологій.

Для наочності буде локальна версія програмного забезпечення для демонстрації роботи знань та автоматичної побудови онтологій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Поняття онтології

У літературі з штучного інтелекту представлено безліч визначень поняття «онтологія». Відомо, що одним із перших у галузі інформаційних технологій це поняття почав використовувати T. Gruber, який визначив онтологію як «точну специфікацію концептуалізації» [1].

Різноманітність практичних завдань, розв'язуваних за допомогою онтологій, призводить до різних змістовних трактувань даного поняття, що вичерпує аналіз.

Під онтологією розумітимемо деяке формальне уявлення предметної області.

Визначимо онтологію як упорядковану трійку виду [2]:

$$O=<T, R, F>, \tag{2.1}$$

де T – кінцева множина термінів (концептів, понять, класів) предметної області, яку представляє онтологія O;

R – кінцева множина відносин між поняттями заданої предметної області;

F – кінцева множина функцій інтерпретації (аксіоматизація), заданих на концептах та/або відносинах онтології O.

Зауважимо, що ще одним природним обмеженням множини T (крім кінцівки) є його непустота, при цьому на множини R і F такі обмеження не накладаються. Якщо безлічі R і F є порожніми, то робота ведеться з глосарієм. Якщо R складається з єдиного відношення «is-a», а F – порожньо, то онтологія буде таксономією.

Необхідно особливо підкреслити, що процес побудови онтології є погано формалізованим і ресурсомістким завданням. Проте задля підтримки вирішення

завдань інформаційного пошуку застосовуються онтології особливого типу – лінгвістичні онтології. В онтологіях такого типу поняття суворо пов'язані з термінами предметної галузі. Прикладом лінгвістичної онтології є загальнозначуща онтологія англійської WordNet.

Надалі розглядатимемо питання автоматичної побудови лінгвістичних онтологій.

Відповідно до визначення, компоненти онтології включають класи, слоти (які являють собою атрибути класів, їх також іноді називають властивостями), факти (окремі елементи, екземпляри класів) і відносини між класами і фактами.

Як приклад розглянемо уривок тексту, що належить до предметної області «музичні інструменти»:

Музичний інструмент флейта відноситься до ряду дерев'яних духових інструментів. Флейти є цілим сімейством інструментів (блок флейта, флейта пікколо). Сучасні флейти виготовляють із дерева, металу, пластику. Професійні музиканти часто віддають перевагу інструментам виробника Yamaha.

На основі даного уривка можна виділити класи «музичний інструмент» (атрибут класу «тип інструменту»), «флейта» (з атрибутами «матеріал» та «виробник» та ставленням «is-a» з попереднім класом), «блок флейта» та «Флейта Пікколо». Останні два класи пов'язані ставленням «is-a» із класом флейт. Якби у уривку згадувалася конкретна модель флейти (наприклад, флейта Yamaha YRA-83), то вона була б екземпляр класу (у прикладі – екземпляр класу блок флейт).

Таким чином, на практиці розробка онтології включає чотири етапи:

– визначення класів онтології;
– визначення відносин та побудова ієрархії класів;
– визначення слотів та опис допустимих значень для слотів;
– заповнення значень слотів для екземплярів.

У прикладі як допустимі значення слота «тип інструменту» можуть

виступати «дерев'яні духові», «мідні духові» і т.д.

Як було зазначено, онтології застосовуються на вирішення безлічі різноманітних практичних завдань. Найбільш масштабним проектом, ядро якого становлять онтологію, став проект консорціуму W3C Semantic Web.

Основна мета проекту полягає у додаванні семантики до інформації, що міститься в Internet, з метою зробити її доступною для автоматичного сприйняття комп'ютером.

Варто зазначити, що Semantic Web, через свою масштабність і прийняті стандарти, вплинув на всі розробки, в яких використовуються онтології. В рамках проекту, зокрема, було розроблено різні мови опису онтологій (RDF та RDFS, а також стандарт OWL 2).

Відповідно до стандартів було розроблено різні редактори онтологій (такі як Protégé [4]) та процесори для їх обробки (наприклад, Jena – засіб виведення на онтології [5]).

1.2 Автоматична побудова онтологій

Методи автоматичної побудови онтологій можна умовно поділити на три основні групи залежно від галузі запозичення основного підходу: методи, засновані на підходах в галузі штучного інтелекту, статистичні методи та методи, що використовують лінгвістичні підходи. Оскільки підхід переважно заснований на теорії інформації та відноситься до статистичних, у цьому розділі розглядаються методи, що належать до двох інших груп.

1.2.1 Підхід на основі лексико-синтаксичних шаблонів

Цей підхід відноситься до групи методів автоматичного побудови онтологій, що використовують лінгвістичні засоби [3].

Прихильники підходу стверджують, що для побудови онтологій слід активно використовувати усі рівні аналізу природної мови: морфологію,

синтаксис та семантику. Таким чином, для автоматичної побудови онтології автором використовується один із методів семантичного аналізу текстів природною мовою – лексико-синтаксичні шаблони.

Як метод семантичного аналізу лексико-синтаксичні шаблони давно використовуються в комп'ютерній лінгвістиці і є характерними виразами і конструкціями певних елементів мови.

Ця методика семантичного аналізу не є спеціалізованою на певну предметну область.

На основі лексико-синтаксичних шаблонів вирізняються онтологічні конструкції.

Наприклад, із пропозиції «Студент – це людина, яка навчається в університеті» [3], система виділить класи «студент», «людина» та відношення «subclass-of» між ними.

Через складність завдання оцінка результатів застосування підходу проводиться авторами опосередковано через аналіз результатів його використання в різних додатках Semantic Web.

Загалом зазначається, що лексико-синтаксичні шаблони як метод семантичного аналізу текстів природною мовою – у разі великого обсягу колекції шаблонів – є ефективним засобом для автоматичної побудови онтологій.

1.2.2 Підхід на основі системи продукції

Даний підхід відноситься до групи методів автоматичної побудови онтологій, в основі яких лежать підходи в галузі штучного інтелекту [6].

Ефективна автоматична побудова онтологій може бути заснована на можливості методів штучного інтелекту до вилучення з тексту елементів знань та їх нетривіальної переробки.

Аналіз області природно-мовної обробки тексту показує переважання використання різних правил під час вирішення завдань у аналізованої

предметної області. Даний факт, а також декларативний характер подання методів автоматичної побудови онтологій, обґрунтовує застосування системи продукції як модель уявлення знань про метод.

Для створення методів автоматичної побудови онтологій автор розробляє модель генерації системи продукцій (на основі застосування генетичного програмування), модель генерації перетворювачів (на основі генетичного та автоматного програмування), модель генерації систем логічного висновку (також на основі генетичного та автоматного програмування) та модель апарату активації продукцій (з урахуванням застосування автоматного програмування).

Таким чином, пропонується модель автоматичної побудови онтологій у вигляді системи продукцій та застосування генетичного та автоматного програмування для створення необхідних моделей.

1.3 Підхід на основі статистичних методів

Розглянуті вище підходи є єдино можливими. Пропонується підхід до вирішення проблеми автоматичної побудови онтологій, переважно заснований на статистичних методах аналізу текстів природною мовою.

Розглянемо пропонований підхід для перших двох етапів побудови онтології: виділення класів та відносин між ними.

1.3.1 Попередня підготовка колекції

Однією з особливостей роботи з текстами природною є необхідність обов'язкової попередньої обробки даних. Процес обробки може бути досить трудомістким і зазвичай складається з кількох етапів:

- приведення документів до єдиного формату;
- токенізація;
- стемінг (лематизація);
- виняток стоп-слів.

Однак не завжди є необхідність у проведенні всіх перерахованих вище етапів.

В результаті попередньої обробки кожен документ колекції характеризується вектором типів даного документа та їх частотою народження.

Раніше наголошувалося, що особливості колекції впливають на якість онтології. Для покращення одержуваної в результаті роботи системи онтології пропонується провести попередню кластеризацію документів колекції таким чином, щоб в один кластер потрапляли тематично близькі документи, а подальшу роботу проводити окремо з кожним отриманим кластером.

Варто зауважити, що будь-які спеціальні вимоги до алгоритму кластеризації відсутні. Як алгоритм кластеризації передбачається використання методу Contextual Document Clustering, що дає хороші результати на великих текстових колекціях.

1.3.2 Визначення класів онтології

На першому етапі побудови онтології потрібно виділити класи, що входять до її складу. Раніше відзначалося, що поняття лінгвістичної онтології суворо пов'язані з термінами. Таким чином, дане завдання зводиться до визначення термінів предметної області, що розглядається.

Алгоритми вилучення термінів з текстів природною мовою можна розділити на дві групи: статистичні та лінгвістичні [8].

Однак перші мають певну перевагу, оскільки їх використання не залежить від лінгвістичних особливостей конкретної мови.

Підхід до вилучення термінів є переважно статистичним. Тим не менш, передбачається, що існуючі статистичні методи можуть показати найкращі результати, якщо доповнити їх певними евристиками.

Попередньо як базові евристики пропонується використовувати наступні:

– ім'я класу містить хоча б одну іменник;

– загальновживані слова в порівнянні з термінами мають більшу з частотою народження, приблизно рівною в різних предметних областях;

– кількість інформації терміна з кількох слів більша, ніж кількість інформації окремих слів, що входять до його складу.

Опишемо пропонований підхід докладніше. На першому етапі в кожній колекції документів виділяють іменники та визначають їх частоту народження. Отже, внаслідок використання першої евристики число передбачуваних класів значно скорочується.

З другого краю етапу виділяють терміни, які з одного слова. На підставі висунутої другої евристики, порівнюються частоти різних іменників в рамках однієї колекції, також проводиться оцінка перетину різних колекцій за іменниками.

Проте статистичні дані – єдине джерело класів онтології. Термінологічні словники можуть стати джерелами знань для формування ядра онтології. У разі роботи з колекціями неспеціалізованих у конкретній галузі документів можливе використання існуючих розроблених експертами онтологій (наприклад, для англійської мови – онтології WordNet).

Нарешті, третьому етапі з урахуванням взаємної інформації може бути виділено терміни, які з кількох слів. У цьому випадку використовується третя евристика.

Для двоскладових термінів отримуємо, що взаємна інформація визначається за формулою:

$$\mathrm{mi}(x, y) = \frac{P(x,y)}{P(x)P(y)}, \tag{2.2}$$

де x_i, y являють собою окремі слова терміна,

$P(x)$ – частота народження x,

$P(x,y)$ – частота спільної зустрічальності x та y.

Виділені описаним вище чином терміни будуть класи майбутньої

онтології.

Отже, запропонований підхід можна віднести до групи статистичних методів. Передбачається, що висунуті евристики (у тому числі, що до складу імені класу має входити іменник), є досить універсальними і не обмежують застосування методу лише українськкою мовою.

1.3.3 Визначення відносин між класами

Звісно ж, що етап виділення відносин між класами створить найбільші труднощі. У зв'язку з чим спочатку має сенс говорити про автоматичну побудову не довільної прикладної онтології, а тезаурусу (таксономії з термінами).

Як базові відносини, що діють між термінами, визначимо відносини «is-a» та «synonym-of».

Для виділення відношення «is-a» можна скористатися кількісним підходом до інформації. Для цього ще раз скористаємося припущенням 3, зробленим у попередньому пункті.

Очевидно, що термін, що знаходиться на нижчому рівні ієрархії, має більшу кількість інформації, ніж узагальнюючий термін.

Так, у прикладі з розділу 1.1 кількість інформації терміна флейта пікколо буде більше, ніж кількість інформації терміна флейта. Отже, останній термін може бути надклас першого.

Однак для встановлення будь-якого відношення між термінами, знання лише про кількість інформації, яку вони містять, недостатньо.

Припустимо, що кожного отриманого першому етапі терміна обчислено відповідне йому кількість інформації. Визначення того, чи пов'язані два різні терміни з різною кількістю інформації ставленням «is-a», можна проводити двома способами.

Перший спосіб ґрунтується на припущенні, що приватні терміни містять у своєму складі слова із загальніших термінів. Наприклад, «блок флейта» і «флейта

пікколо» містять термін «флейта». З урахуванням проведеного вище аналізу за кількістю інформації цих термінів, цілком обґрунтованим виглядає припущення про встановлення відносини «is-a» між ними (а саме, «флейта пікколо is-a флейта»).

Другий спосіб полягає в понятті «контексту слова». Контекст слова може бути визначений як умовна ймовірність $P(Y|x)$, де Y – змінна величина, яка приймає значення зі словника колекції, а x – слово, що шукається.

Поняття контексту можна розширити до «контексту терміна». Тоді x буде являти собою шуканий термін, який у загальному випадку може складатися з декількох слів, а змінна величина Y прийматиме значення зі словника термінів, що зустрічаються в колекції документів.

Інакше кажучи, під контекстом терміна розумітимемо кілька слів, які зустрічаються одночасно з цим.

Якщо терміни не мають спільних слів, але збігаються контекст, і при цьому вони мають різну кількість інформації, має сенс говорити про відношення «is-a» між ними.

Повертаючись наприклад, можна припустити, що у уривку описані умови будуть виконуватися для термінів «музичний інструмент» і «флейта». Контекст їх вживання співпадатиме, тоді як кількість інформації останнього терміну виявиться вищою.

Якщо контекст слів збігається, але кількість інформації термінів приблизно однакову, то найімовірніше між термінами діє ставлення синонімії, тобто. відношення «synonym-of».

У прикладі з розділу 2.1 контекст термінів визначити досить складно через малого обсягу тексту. Тим часом можна припустити, що в предметній галузі, що розглядається, терміни «виконавець» і «музикант» виявляться синонімами, контекст яких складуть терміни «інструмент», «твір» та деякі інші.

Запропонований підхід дозволяє виділити лише базові відносини, необхідні для побудови таксономії. Проте передбачається, що можливе його

розширення виділення інших відносин.

1.3.4 Оцінка якості збудованої онтології

Поширений критерій оцінки якості онтології ґрунтується на оцінці роботи програми, що використовує онтологію. У зв'язку з чим оцінка автоматично побудованих онтологій є окремим складним завданням.

Зважаючи на існування відпрацьованих методик оцінки якості інформаційного пошуку (точність та повнота пошуку), можна оцінювати онтології за якістю роботи систем семантичного пошуку, які використовують онтології. За підсумками онтологій системи можуть значно звужувати простір пошуку з допомогою динамічного розширення запитів користувача.

У вільному доступі є безліч різних текстових колекцій. В основному, такі колекції використовуються з метою оцінки якості запропонованих підходів до вирішення завдань пошуку, кластеризації та класифікації текстів.

Має сенс проводити оцінку якості побудови онтологій на двох колекціях: спеціалізованій (наприклад, колекції MEDLINE) та колекції з неспеціалізованими документами (наприклад, Reuters–21578).

Додатковою складністю оцінки якості побудованої онтології є деякого еталона. Експертна оцінка досить трудомістка. У зв'язку з цим особливого значення набуває колекція MEDLINE, для якої існує побудований тезаурус термінів, що дозволить оцінити роботу методу на перших етапах.

1.4 Класифікація онтологій

Онтології сильно різняться за низкою параметрів, і дослідники виділяють різні підстави їх класифікації. Так Е. Хові [10] говорить, що онтології різняться залежно від набору елементів, що містяться в них, а також типів відносин, що вводяться. Він виділяє так звані «термінологічні онтології» та «справжні

онтології». Під першими Е. Хові [10] розуміє онтології, що включають сутності, явища, властивості, зв'язки предметної області та структурні відносини, що об'єднують їх. «Справжні» ж онтології включають також дефіційні відносини і відносини додаткової інформації. Поряд із цим в них входять аксіоми, що визначають взаємозалежність між відносинами та поняттями.

е. Хови вибудовує докладну класифікацію різних характеристик онтологій. Він згадує, що основними параметрами можуть бути: форма (то як формується онтологія), зміст, а також засоби використання онтології.

Можна згадати про те, що існує розбиття онтологій за кількістю та якістю понять, що включаються до них. Онтології верхньої зони зазвичай налічують приблизно 100-500 концептів. Вони включені найбільш абстрактні категорії, які мають властивістю універсальності. Вони є базовим розбиттям дійсності, що спостерігається на категорії. Зазвичай вони будуються теоретиками, філософами. Найчастіше концепти навіть лексикалізуються. Складання аксіом у цьому типі онтологій з високим рівнем узагальнення досить складно і потребує деякої уяви. Перевагою таких онтологій є можливість їх використання у багатьох сферах і навіть у багатьох мовах. Для такого роду онтологій характерний обмежений набір узагальнених відносин, які можна віднести до базових (таких як родовідові відносини, відносини частина-ціле та асоціативні відносини):

– сутність;
– явище;
– об'єкт;
– процес;
– роль.

Однак цей типовий набір може бути представлений у усіченому вигляді, наприклад, в онтології MicroKosmos, розробленої С. Ніренбургом та В. Раскіним на верхньому рівні з'являються лише три категорії понять: «об'єкт», «процес» та «роль». У цьому автори претендують універсальність цього розбиття для онтологічної семантики загалом і, отже, всім онтологій верхнього рівня.

Іншим типом є онтології середньої зони, тут елементів зазвичай більше (500 – 100000 концептів). Вони представляють світ загалом і у випадку це неаксіоматизована область. Складність у тому, що з цього виду онтологій потрібно виводити дуже багато аксіом. Зазвичай виходом є використання методів автоматизованого виведення аксіом із існуючих онтологій. Побудовою онтологій цього рівня найчастіше займаються когнітологи та лінгвісти.

Онтології нижньої зони чи звані онтології предметної області найбільш великі, зазвичай вони налічують близько 200 – 2 000 концептів. Вони описують конкретні предметні області зі специфікою. При цьому коло розв'язуваних завдань та питань, на які відповідає онтологія, обмежений обраною областю. Для цього онтологій характерно наявність відносин, специфічних для конкретної області [3]. Це високоаксіоматизована зона, тобто для неї можлива побудова великої кількості аксіом та правил. У більшості випадків цей тип онтологій будується експертами галузі знання або за їх сприяння. У зв'язку з великою специфікою кожної окремої предметної онтології її повторне використання найчастіше можливе лише рамках предметної області.

Поряд з описаним розподілом всі онтології можуть бути поділені на глибинні та поверхневі. Поверхневі онтології будуються на поверхневій семантиці, визначають поняття через значення слів. Однак тут виникає проблема, яку кількість смислів виділятиме для кожного слова. Глибинні онтології використовують глибинну семантику.

Б. Й. Вілінга та А. Т. Шрайбер виділяють два виміри для оцінки онтологій: «обсяг і тип структури концептуалізації та предмет концептуалізації» [11]. Однак, як зазначає Н. Гуаріно [12], критерій для виділення в рамках першого виміру інформаційних, термінологічних онтологій та онтологій, що моделюють знання, не відрізняється чіткістю. Найпростіше розділити онтології за ступенем детальності, що використовується для характеристики концептуалізації. Дуже деталізована онтологія докладно спеціалізує на увазі концептуалізацію, але платою за це виявляється більш громіздка мова, який може бути складно

застосовувати на практиці. Проста ж онтологія може розвиватися з припущенням якихось прихованих умов, які маються на увазі творцями, і її можуть використовувати ті, хто вже домовився про концептуалізацію, що лежить в основі (і усвідомлює ці припущення).

Можна розрізняти «отсылочные (також звані офф-лайн) онтології» (reference ontologies) і «здійснювані (спільно використовувані, он-лайн) онтології» (implemented (shareable) ontologies). Нескладна структура, що описує, наприклад, лексикон, може міститися он-лайн, у той час як хитромудрі теорії, що визначають значення термінів з лексикону, можуть знаходитися офф-лайн.

З погляду предмета концептуалізації дослідники виділяють прикладні онтології, онтології галузі знання, загальні (родові) онтології та репрезентаційні онтології (йдеться про онтології метарівня, що включають репрезентаційні першоелементи) [12].

Онтології можуть бути розділені на одномовні та багатомовні. Вже існує ряд онтологій, орієнтованих на представлення знань кількома мовами, наприклад EuroWordNet, MikroKosmos та деякі інші. Складність створення таких онтологій зазвичай полягає в тому, що можлива наявність відмінностей у понятійних системах різних мов.

У межах роботи [13] виділяється особливий тип онтологій – лексичні (чи лінгвістичні). Відмінною властивістю таких онтологій є «фіксація в одному ресурсі (лексикалізованих) понять (слів) разом із їх мовними властивостями» [13]. Такі онтології тісно взаємопов'язані з семантикою граматичних елементів (слів, іменних груп та ін.). Основним джерелом понять у онтологіях цього є значення мовних одиниць. Їх також відрізняє своєрідний набір відносин, зазвичай властивий мовних елементів: синонімія, гіпонімія, меронімія, і навіть ряд інших. До лінгвістичних онтологій автори [13] відносять WordNet, MikroKosmos, Sensus, РуТез та інші. Коло завдань, розв'язуваних такими онтологіями, тісно взаємозалежне з обробкою природної мови.

С. А. Коваль [14] пропонує розрізняти безекземплярні та екземплярні онтології. Як відомо з назви цього типу онтологій, безекземплярні онтології відрізняються відсутність конкретних екземплярів. На нижніх рівнях ієрархії таких онтологій є не конкретні екземпляри, а поняття. Ця особливість онтології накладає деякий відбиток і на відносини, що вводяться в даній онтології.

Таким чином, існує безліч підрозділів онтологій, але ці класифікації не завжди бувають досить чіткими та послідовними.

Створювана нами онтологія належала до онтологій нижньої зони та описувала конкретну предметну підобласть. У зв'язку з цим для нашої онтології були характерні деякі відносини, які є специфічними для даної галузі, і мала можливість, що можуть використовуватися в інших областях. Вона також була безекземлярною, що наклало свій відбиток встановлення відносин між елементами. На даному етапі створення можна говорити про одномовність нашої онтології, хоча для деяких елементів вже була спроба пов'язати російські мовні вислови з англійськими.

1.5 Автоматичні методи побудови онтологій

Як вже було згадано раніше, зараз розвиток онтологій починає набувати більш масового характеру, і нині в цій галузі є низка масштабних розробок. Існує велика кількість різних списків і баз даних, але постає питання, як гарантувати їхню відповідність поточному стану речей, як бути впевненим, що вони точні і повні, а також як забезпечити достатню детальність даних. У зв'язку з тим, що світ дуже швидко змінюється, йде розвиток нових галузей, існуючі онтології потребують постійного поповнення та вдосконалення. На даному етапі з'являються ідеї використання автоматичних та напівавтоматичних методів для не лише оновлення онтологій, а й навіть для їх створення.

Існує ряд методів розширення онтологій, які є специфічними для онтологій різних зон. Для розширення верхньої, найбільш загальної зони потрібне

докладне теоретизування, після чого можна приступати до побудови понять та аксіом. Для онтологій середньої зони, що відрізняються великою кількістю понять, збір понять може виконуватися автоматично за допомогою кластеризації. У процесі обробки великої кількості інформації відбувається збір понять та розбиття їх за класами на підставі якихось загальних характеристик. Існує цілий ряд методів збільшення точності вилучення семантично пов'язаних сімей понять. За такого аналізу надалі можна також встановлювати перехресні посилання всередині онтології. Однак для онтології важливо знати не тільки те, що поняття взаємопов'язані, але і те, як саме вони взаємопов'язані. Для виявлення таких відносин між поняттями також можуть бути використані автоматичні методи перегляду та аналізу різних текстів, наприклад, як пропонують автори [15], можна витягувати цю інформацію зі словникових визначень. Це зумовлено тим, що існує обмежений набір фразових моделей, що вводять визначальне, за характером яких можна сформулювати тип зв'язків між поняттями та ввести ці дані до онтології. Таким чином, можна вибрати словникові статті, провести їх розбір, виявити семантичні тлумачення для слів та початкових конструкцій визначень. Дана ідея поповнення онтології зустрічається і в роботі, де такий підхід дозволяє виявити відносини між елементами онтології за допомогою аналізу корпусу та пошуку моделей, що відповідають будь-якому роду відносин. правил може бути зроблено на підставі Інтернету. Еге. Хови вказує на те, що збір окремих прикладів відбувається за рахунок вивчення великої бази ресурсів, проте спочатку формулюється ряд параметрів, що вказують, які саме екземпляри нам потрібні [10]. Це можна пояснити на конкретному прикладі: якщо нам потрібно зібрати назви столиць, то можна, наприклад, встановити умови пошуку з моделлю «X – столиця ...». При цьому можна використовувати не одну таку модель, а кілька. Таким чином, знаходження окремих екземплярів буде зводитись до аналізу текстів для виявлення прикладів із даними структурами. Це можна пояснити на конкретному прикладі: якщо нам потрібно зібрати назви столиць, можна, наприклад, задати

умови пошуку з моделлю «X – столиця ...». При цьому можна використати не одну таку модель, а кілька. Таким чином, знаходження окремих екземплярів буде зводитись до аналізу текстів для виявлення прикладів із даними структурами. Це можна пояснити на конкретному прикладі: якщо нам потрібно зібрати назви столиць, можна, наприклад, задати умови пошуку з моделлю «X – столиця ...». При цьому можна використовувати не одну таку модель, а кілька. Таким чином, знаходження окремих екземплярів буде зводитись до аналізу текстів для виявлення прикладів із даними структурами.

Такі шаблони можуть формуватися як вручну, так і автоматично за допомогою програм, що самонавчаються.

Поряд із пошуком окремих примірників важливим є встановлення різних зв'язків між елементами онтології, класами. Розділивши всі методи отримання відносин між елементами онтології на два класи: підходи, засновані на використанні шаблонів, і методи, що використовують кластеризацію. При використанні методів на основі шаблонів дослідники шукають мовні моделі, які вказують на тип відносин між класами. Найчастіше здійснюється пошук родовидових відносин і відносин «частина-ціле».

За наявності базового списку категорій можна збільшувати їх кількість, звертаючись до корпусу текстів. У корпусі виділяються кластери близьких за значенням елементів, потім оцінюється тіснота зв'язку між елементами, далі кожному кластеру приписується ім'я, яке асоціюється з категорією, що входить до онтології. Таким чином, поповнення онтології новими елементами відбувається автоматично і автоматично визначається місце нового елемента в ієрархії категорій.

Як згадувалося раніше, в онтології присутні не лише класи та окремі екземпляри, а й аксіоми. Вони роблять великий внесок у вдосконалення аналізу інформації, дають можливість комп'ютеру доповнити текст додатковими знаннями, які для нас здаються тривіальними. Аксіоми повідомляють комп'ютеру, наприклад, про те, що два висловлювання мають один і той же сенс

або один факт тягне за собою наявність іншого. Метод автоматичного вилучення аксіом з текстів та їх подальша класифікація та перевірка на релевантність. Як основна проблема автоматичного виявлення аксіом автори згадують побудову помилкових припущень або занадто загальних, а також проблему визначення симетричності аксіом. Вони беруть за основу гіпотезу про схожість значень при схожості контекстів, доповнюючи їх своїми обмеженнями. Тут ключовим стає визначення методів обчислення близькості контекстів та близькості понять. За твердженням авторів їх методика виявлення аксіом дозволяє видобувати релевантні аксіоми з невеликим відсотком помилок.

Найчастіше проблемою автоматичного вилучення стає велика кількість «шуму», який треба ефективно відсівати. У зв'язку з цим іноді поряд з автоматичними методами використовують подальшу обробку ручну отриманого матеріалу для отримання даних більшої точності.

Загальні вимоги до систем автоматичного вилучення даних для онтологій:

– мінімальний контроль – зведення до мінімуму чи виключення взагалі участі людини;

– універсальність - застосовність до різних джерел, всі залежності від їх розміру, галузі знання і т.д;

– точність – видобута інформація має містити якнайменше помилок.

Виконання цих вимог, можливо, дозволить побудувати ефективну систему автоматичного побудови онтологій, поки всі існуючі системи потребують доопрацюваннях і поліпшеннях чи успішно працюють лише стосовно замкнутим областям знання.

Однак ніяка онтологія не корисна в ізоляції, її перевага полягає в можливості використовуватися знову в новому оточенні. Зараз існує багато різних алгоритмів стикування кількох вже створених онтологій (їх загальний опис наводиться в [19], [20]). Ці методи використовуються для злиття онтологій та знаходження відповідностей для слів в інших мовах (в даному випадку онтологія служить міжмовною точкою відліку).

Поєднання двох онтологій здійснюється у кілька етапів. На початковій стадії відбувається знаходження сполучної ланки, на підставі якої можна зробити злиття. Потім спрацьовує алгоритм, що вирівнює, який за описом елементів знаходить їх місце в новій структурі. Потім уже йде вивіряння результатів стикування у контексті.

Як згадувалося раніше, існує цілий набір алгоритмів знаходження сполучної ланки для двох онтологій, варто розповісти про них трохи докладніше.

Виділяють кілька різних методів виявлення сполучних ланок:

– текстові збіги;

– збіги ієрархічних відносин;

– збіг форматів та даних.

Еге. Хови [19] пояснює, що під текстовими збігами мається на увазі ідентичність імен понять (тут також враховуються споріднені слова), текстових визначень (порівняння рядків, трансформація, виключення стоп-слів та ін.).

Ієрархічний збіг передбачає пошук загальних вищих понять, фільтрацію неоднозначностей, знаходження семантичної відстані, розсіювання семантичних груп (semantic group dispersal). Збіг форматів та даних спирається на внутрішньопонятійні відносини та обмеження на заповнення слотів.

Після відпрацювання алгоритмів використовується функція, яка враховує результати всіх процедур та видає загальний коефіцієнт збігу. Знаходження сполучної ланки включає також процедури валідації, в ході яких відбувається перевірка з урахуванням ієрархічних зв'язків співвідносних понять. Ця процедура намагається знайти невідповідності, порочні кола, перевірити успадкування властивостей.

Для виявлення ідентичності понять використовують спеціально створені критерії. Так, комплексний критерій, запропонований Н. Гуаріно, перевіряє подібність за кількома параметрами:

– матеріал: ідентичність матеріалу, з якого зроблено екземпляри порівнюваних понять;

– топологічний: ідентичність форми екземплярів порівнюваних понять;

– морфологічний: ті частини, з яких складаються екземпляри понять, що порівнюються;

– функціональний: використання;

– меронімічний: екземпляри понять;

– соціальний: соціальна роль [19].

Враховуються також можливі стандартні метонімічні переноси, які роблять онтологію гнучкішою та розширюють можливість знаходження близьких за змістом понять.

Згадані раніше методи є напівавтоматичними, тобто спочатку автоматично генеруються варіанти відповідностей, а потім вручну кілька етапів відбувається з'єднання онтологій. Як згадує Е. Хові [19], статистика показала, що ці процедури мають досить високий рівень точності і дають хороші результати. Так, використання таких автоматичних алгоритмів вирівнювання при побудові онтології SENSUS дало більше 90% точності.

Побічним позитивним результатом використання процедур злиття є можливість виявити помилки та опущення в онтологіях. Злиття онтологій може використовуватись також для усунення проблеми дублювання. На даному етапі розвитку відбувається так, що змагаються та взаємопов'язані колективи формують схожі моделі галузей знання, де спостерігається часткове перетинання.

За такого роду дублювання неминуче страждає і логічність. Слід визнати, деякі перетину звичайно необхідні.

Однак Е. Хові [19] зазначає, що при злитті онтологій може виникнути ряд проблем, які досить складно вирішувати автоматичними методами.

Так експерти в різних галузях можуть відсилати до того самого поняття і розуміти його по-різному. Проблеми виникають і тоді, коли те саме слово використовується для позначення різних понять у кожному окремому полі.

Вирішенням такої проблеми може бути тісніша комунікація укладачів

онтологій, а також використання ширших онтологій, що застосовуються до різних галузей знання.

1.6 Постановка задачі

Для досягнення мети кваліфікаційної роботи необхідно вирішити такі завдання:

– провести аналіз предметної галузі;

– провести дослідження теоретичної основи розробки програмного докладання;

– розробити алгоритм для вилучення;

– розробити алгоритм, що дозволяє виробляти автоматичну побудову онтологій;

– розробити програмний продукт, який дозволяє робити висновок із онтології.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Система автоматичного отримання знань з тексту та її застосування

Надії на можливість створення баз знань (БЗ) для реальних додатків тільки на узагальненні досвіду експертів предметних областей не виправдалися. Що ж, по-перше, є найнадійнішим і найефективнішим джерелом таких знань, і, по-друге, яким чином вони можуть бути вилучені з цього джерела і потім формалізовані? Можна легко відповісти на перше запитання: таким є Текст, який розуміється в широкому значенні цього слова як безліч текстових документів (книг, статей, патентів, звітів тощо), представлених в електронній формі. Більше того, оскільки мова в кінцевому рахунку йде про розробку інтелектуальних інформаційних систем (ІІС), що передбачає наявність у них природно-мовного інтерфейсу користувача, процедур смислової обробки документів і т.д. то необхідні для цього лінгвістичні знання також можуть і повинні бути вилучені з тексту, до речі, з яким (умовно) цим системам доведеться працювати надалі. Таким чином, базовими посилками в наших дослідженнях та розробках у розрізі першого питання є такі дві:

– текст – це ідеальна природно-інтелектуальна модель уявлення знань;

– у тексті «є все».

Зрозуміло, що друга посилка дуже категорична, але зі зростанням об'єму тексту вона все більше має місце.

2.1.1 Процесор знань: функціональність лінгвістичного процесора

Процесор знань (ПРЗ) має таку структурно-функціональну схему (рисунок 2.1).

Розглянемо докладніше основні модулі ПРЗ, що, власне, і буде відповіддю другого з поставлених питань. Текст практично в будь-якому форматі (DOC,

PDF, RTF, HTML, XML, TXT та ін.) надходить на вхід лінгвістичного процесора (ЛПР), який виконує його преформатування, лексичний (розпізнавання меж слів та речень), лексико-граматичний, синтаксичний та семантичний аналіз.

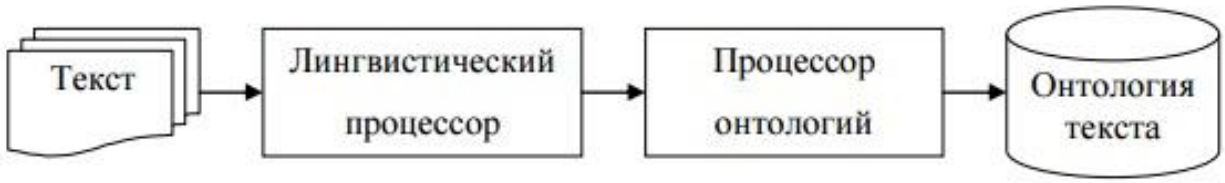

Рисунок 2.1 – Структурно-функціональна схема процесора знань

Преформатор наводить документи, представлені у різних форматах, до певного уніфікованого формату. Отже, на виході преформатора ЛПР має структурований текст, у якому виділено розділи, абзаци та заголовки, а також відфільтровано таку допоміжну інформацію, як тексти кнопок, скрипти тощо. Далі ЛПР здійснює лексико-граматичний та синтаксичний аналіз тексту. У першому випадку ЛПР визначає лексико-граматичні класи всіх слів, тоді як у другому – будує синтаксичне дерево кожної його фрази. Вирішенню цих завдань у науковій літературі присвячено чимало робіт, але досі актуальними залишаються питання оптимізації алгоритмів за часом і точності обробки саме довільних текстів, причому великих (промислових) обсягів. У нашому випадку ефективне вирішення цих проблем досягнуто за рахунок розробки та комбінування різних типів алгоритмів (імовірнісно-статистичних та евристичних) та певного розвитку, та використання теорії формальних мов та теорії кінцевих автоматів. У результаті, наприклад, точність лексико-граматичного аналізу у нас склала в середньому 97%, при цьому точність розпізнавання іменних груп – 90% (при ідеальному лексико-граматичному аналізі – 98,5%), а повнота – 98%, що на сьогодні час є найкращими характеристиками автоматичного аналізу довільних текстів. При цьому

тимчасові показники ЛПР виявилися досить високими і перевершують, наприклад, аналогічні такого відомого виробника промислових лінгвістичних ресурсів, як при цьому точність розпізнавання іменних груп – 90% (при ідеальному лексико-граматичному аналізі – 98,5%), а повнота – 98%, що нині є найкращими характеристиками автоматичного аналізу довільних текстів. При цьому тимчасові показники ЛПР виявилися досить високими і перевершують, наприклад, аналогічні такого відомого виробника промислових лінгвістичних ресурсів, як при цьому точність розпізнавання іменних груп – 90% (при ідеальному лексико-граматичному аналізі – 98,5%), а повнота – 98%, що нині є найкращими характеристиками автоматичного аналізу довільних текстів. При цьому тимчасові показники ЛПР виявилися досить високими і перевершують, наприклад, аналогічні такого відомого виробника промислових лінгвістичних ресурсів, якConnexor [20]. Наступним етапом роботи ЛПР є семантичний аналіз тексту. Його основне завдання – розпізнавання концептів та семантичних відносин між ними, типу С – А – О (суб'єкт – акція – об'єкт) та відносин типу причина – наслідок між самими САО-трійками (повними та неповними) [21]. Зауважимо, що насправді ЛПР у нас розпізнає так зване розширене САО-відношення [22]:

– суб'єкт С, який виконує акцію А над об'єктом;

– акція А;

– об'єкт;

– непрямий об'єкт (наприклад: lamp is placed on the table; the light at the top is dim);

– атрибут-прикметник (наприклад: the invention is efficient; the water becomes hot);

– атрибут-прийменник (наприклад: the lamp is placed on the table; the light at the top is dim);

– атрибут-прислівник (наприклад: process is slowly modified; driver must not turn the steering wheel in such a manner).

Зазначені додаткові компоненти є важливими семантичними характеристиками САО-трійки елементів. Наприклад, останній компонент (атрибут-говір) характеризує особливості виконання (протікання) акції А.

Робота ЛПР забезпечується «розвиненою» лінгвістичною базою знань (ЛБЗ), основними показниками якої є:

– близько 100 типів лінгвістичних ресурсів: словники, граматики, корпуси текстів, алгоритми, класифікатори властивостей мови тощо;

– базовий корпус текстів обсягом понад 100 млн. слів;

– близько 60 різних словників (базовий словник, словник ідіом, словник абревіатур, словник стоп-слів, словники синонімів тощо);

– базовий словник англійської, близько 200 тис. слів;

– імовірнісно-складова та заснована на правилах граматики англійської мови;

– близько 20 тис. евристичних правил (патернів) аналізу тексту різних рівнях глибини мови тощо.

Таким чином, на виході ЛПР ми фактично маємо лінгвістичний індекс тексту, який далі може бути використаний при вирішенні різних прикладних завдань.

2.1.2 Процесор знань: функціональність процесора онтологій

У разі йдеться про інженерії знань, й у завдання ми орієнтуємося передусім такі класичні елементи знань, як об'єкти (класи об'єктів), факти й правила, що відображають закономірності зовнішнього світу. Неважко помітити, що всі ці елементи ми вже маємо у складі лінгвістичного індексу тексту відповідно: концепти, САО-трійки та причинно-наслідкові відносини між САО-трійками.Інша річ, що це лише частина знань, що витягуються; існує, наприклад, ще багато важливих відносин між концептами та їх властивостями, які мають

бути розпізнані ПРЗ. Саме це завдання вирішує разом процесор онтологій (ПРО). Визнаючи факт існування різних визначень поняття «онтологія», ми, як і в [23], розглядаємо її як одну з найбільш ефективних моделей уявлення знань про світ у вигляді словника термінів (концептів) і логічних виразів, які описують значення цих термінів і їхні стосунки. Онтологія дуже «природна», а значить, добре зрозуміла користувачеві, компактна, легко редагована і візуалізується, гнучка по відношенню до конкретного додатка. У нашому випадку ПРО для кожного отриманого з тексту за допомогою ЛПР концепту C_i будує онтологію типу:

а) структура C_i (головне/атрибут);

б) властивості C_i:

1) C_i є (це) …;

2) C_i складається з …;

3) C_i має параметри …;

4) C_i має переваги …;

5) C_i має недоліки ...;

в) семантичне визначення C_i (семантичний тэг);

г) відносини C_i:

1) ієрархічні;

2) синонімічні;

3) функціональні:

– типу суб'єкт – акція – об'єкт;

– причинно-наслідкові.

САО-відносини тут – це всі ті САО-трійки, у яких Сі перебуває у функції суб'єкта чи об'єкта, а причинно-наслідкові – всі ті відносини типу причина-слідство, елементом яких (у складі будь-якої САО-трійки й у будь-якій функції) є Сі.

Робота процесора онтологій підтримується відповідної БЗ, основними компонентами якої є, по-перше, безлічі правил, що формалізують поведінку

експерта при ручній обробці документів та узагальнюють його у вигляді так званих патернів (пошукових шаблонів), що оперують, зокрема, вихідними даними ЛПР. Наприклад, один із патернів розпізнавання причинно-наслідкових відносин між фактами має вигляд:

$$NG1 <cause> NG2 + TO + VB + NG3, \quad (3.1)$$

що означає таке: якщо у реченні є іменна група *NG1*, а далі дієслово «cause | s | d», за ним – іменна група *NG2*, дієслово *VB* у формі інфінітива та іменна група *NG3*, то два факти, що мають тут місце і виражені відповідно за допомогою *NG1* та *NG2 + TO + VB + NG3*, знаходяться у причинно-наслідковому відношенні, причому перший факт є причиною, а другий – наслідком (наприклад: pulsating voltage causas electrode to produce the corona effect in the chamber.). З перелічених раніше відносин Сі найбільш складними для розпізнавання виявляються ієрархічні та синонімічні відносини. Тому, по-друге, БЗ ПРО включає універсальну лексичну базу даних ModWN типу WordNet [24] (останній є частиною її), яка являє собою певний «зріз» онтології зовнішнього світу і дозволяє «занурювати» в неї знання, що витягуються з тексту, що особливо важливо при вирішенні інформаційних завдань, особливо – інноваційних завдань, завдань інформаційного пошуку і т.п. Сама по собі WordNet, як показало тестування, є малоефективною для вирішення прикладних завдань: концепти, що використовуються в ній, є, як правило, занадто загальними поняттями і глибина відносин у них дуже мала. Розширення WordNet, що є трудомісткою процедурою, зажадало розробки засобів її автоматизації. З цією метою було реалізовано два функціональні модулі Autology і Autosyn. Перший на основі системи патернів розпізнає ієрархічні відносини між концептами тексту, не звертаючись до ModWN, тобто. до «зовнішнім» знанням, а потім, порівнюючи за певним алгоритмом дерева ієрархічних відносин тексту з ModWN, здійснює поповнення останньої (з можливим зверненням до експерта). При цьому чим

більше вершин має кожне таке дерево, тим, очевидно, вища ймовірність, що його місце ModWN буде визначено автоматично. В цілому необхідно прагнути до того, щоб ModWN включала, як мінімум, всі ті концепти, які присутні в базовому словнику ЛБЗ, у тому числі і ідіоматичні концепти. Це забезпечить конкретно розроблюваної ИИС властивість гарантованої інформаційної реакції, а то й точної стосовно запиту користувача, хоча хоча б має щодо нього певне ставлення. вище ймовірність, що його місце ModWN буде визначено автоматично. В цілому необхідно прагнути до того, щоб ModWN включала, як мінімум, всі ті концепти, які присутні в базовому словнику ЛБЗ, у тому числі і ідіоматичні концепти. Це забезпечить конкретно розроблюваної ИИС властивість гарантованої інформаційної реакції, а то й точної стосовно запиту користувача, хоча хоча б має щодо нього певне ставлення. вище ймовірність, що його місце ModWN буде визначено автоматично. В цілому необхідно прагнути до того, щоб ModWN включала, як мінімум, всі ті концепти, які присутні в базовому словнику ЛБЗ, у тому числі і ідіоматичні концепти. Це забезпечить конкретно розроблюваної ИИС властивість гарантованої інформаційної реакції, а то й точної стосовно запиту користувача, хоча хоча б має щодо нього певне ставлення.

Другий модуль, використовуючи так званий семантичний контекст розпізнає синонімічні відносини між концептами тексту. Як засвідчили дослідження, розробки промислових ИИС не можна орієнтуватися лише у вже існуючі списки, словники тощо. синонімів, а необхідно вміти розпізнавати реальні контекстні синонімічні відносини, які у цьому документі, і, можливо, лише у ньому, тобто. мова йде про «обчислювану» синонімію, розпізнавання якої і виконується модулем Autosyn. Основна ідея алгоритму зазначеного модуля у тому, що синонімічні об'єкти мають близькі семантичні контексти. Як останні у разі виступають часткові онтології концептів, тобто. певні властивості (атрибути, параметри) і відносини (наприклад, які акції виконує об'єкт, виражений цим концептом, та які акції можуть бути виконані над ним). При

цьому при порівнянні онтологій концептів враховується частотний розподіл семантичного контексту.

Таким чином, представлений процесор знань забезпечує побудову онтології тексту та можливість «занурення» її в онтологію зовнішнього світу на рівні ієрархічних та синонімічних відносин між концептами.

2.1.3 Додатки процесора знань

Що стосується його додатків, то зупинимося докладніше лише на одному з найбільш важливих завдань семантичного пошуку в повнотекстових базах даних. Саме тут ПРЗ дозволяє досить повно виявити свої можливості, забезпечуючи реалізацію найефективнішої, на наш погляд, стратегії вирішення завдання відповідно до схеми:

$$NLI + IR + IE + TS \qquad (3.2)$$

тобто. ефективний пошуковий механізм (пошукова система, машина) повинен забезпечувати користувачеві природньо-мовний-інтерфейс (Natural Language Interface), використовувати при цьому стратегії інформаційного пошуку (Information Retrieval), вилучення інформації (Information Extraction) та автоматичного реферування (Text Summarization). У разі семантичний індекс запиту користувача будується з розрахунку пошуку знань, що стосуються концептів, їх властивостей і атрибутів, і навіть відносин між концептами.

Використовуючи можливості ЛПР, можна автоматично виділити ключові слова в будь-якому із запитів з перерахованих вище типів і потім здійснити по них «швидкий» пошук саме тих документів, в яких, ймовірно, будуть знайдені відповідні запиту користувача знання (це якраз і є IR – компонент загальної стратегії пошуку, оскільки він справді найбільше забезпечує функції пошуку документів, а чи не знань).

2.2 Метод автоматизованого здобуття знань із слабоструктурованих джерел

Вирішення перелічених завдань можливе шляхом створення автоматичних та автоматизованих систем збирання, фільтрації та аналізу інформації, так званих своєрідних «інтелектуальних посередників» між користувачем або КІС організації та Інтернетом. Подібна система виконує всю «чорнову» роботу зі збирання та селекції інформації з Мережі та створює документальну базу даних у певній предметній галузі. Завантаження інформації до бази даних супроводжується її класифікацією та частковою структуризацією. Для подальшої інформаційно-аналітичної роботи кінцевому користувачеві КІС надаються ефективні засоби навігації та пошуку інформації у створеній документальній базі даних.

Для більш ефективного маніпулювання Web даними, деякі дослідники звернулися до ідей, взятих з області баз даних. Традиційні бази даних працюють із регулярною структурованою інформацією і тому технології баз даних не можна безпосередньо застосувати до Web. Тим не менш, в Web доступно багато інформації, яка має деяку певну структуру (наприклад, електронний каталог товарів, новини електронних ЗМІ і т.д.) [25], і можливість використання структурованих запитів для таких ресурсів могла б бути дуже корисною. Ці міркування стимулюють дослідження у сфері організації доступу до такої інформації, що зазвичай називається слабоструктурированной.

Традиційний підхід полягає у вилученні даних з Web та відображенні отриманих даних у деякий проміжний формат, а потім – у виконанні структурованих запитів вже до цих даних. Підхід реалізується на основі механізму посередників (wrappers) – програм, які ідентифікують шукану інформацію у вихідному документі та відображають її в деякий проміжний формат, наприклад, XML (eXtensible Markup Language, мову розмітки, що

розширюється).

Розробка та підтримка посередників вручну дуже трудомісткий процес (зокрема, через різнорідність та нерегулярність оброблюваної інформації). Тому різними колективами фахівців багатьох країн ведеться робота над засобами автоматизації цього завдання.

Для цієї мети запропоновано низку мов опису програм-посередників, які дозволяють значно знизити обсяг програмного коду, що розробляється вручну.

Подальша автоматизація ведеться у напрямі розробки алгоритмів, які автоматично генерують програми-посередники. Для цього проводиться аналіз набору документів, з якого належить витягувати інформацію, і, у ряді випадків, готуються навчальні приклади. Завдання вилучення є складним, оскільки потрібно витягти як вид схеми даних, а й пов'язану з ним семантичну інформацію (наприклад, тип конкретного елемента даних) [25]. Досягнення повної автоматизації у випадку малоймовірно, і може йтися створення автоматизованих систем вилучення інформації з Інтернету.

Одна з основних труднощів, з якою стикаються розробники подібних систем, – це нерегулярність інформації, що обробляється. Зокрема, для автоматичної генерації видобувного інформацію посередника потрібно підготувати набір документів, у тому числі посередник і витягуватиме інформацію. У цьому потрібно, щоб набір був щодо регулярним, тобто. сторінки в цьому наборі повинні мати схожу структуру тих розділів (інформаційних блоків), яка містить інформацію, що витягується. Досі існування цього набору передбачалося апріорі, питання його створення не розглядалися, що створювало певні труднощі при практичному використанні подібних систем, тому в цій роботі було проведено попередній аналіз деяких сайтів новин, який дозволив уточнити алгоритми вилучення.

2.2.1 Підходи до отримання даних та знань з Web

Активні дослідження в галузі роботи із слабоструктурованою інформацією призвели до появи великої кількості альтернативних інструментів, які використовуються для створення програм-посередників [26]. Пропоновані підходи у дослідженні проблеми вилучення даних з Web використовують методи, запозичені з таких областей як: обробка даних природною мовою, мови та граматики, машинне самонавчання, інформаційний пошук, бази даних та онтологія. В результаті, вони надають різні можливості, тому строгих критеріїв для їх порівняння поки не вироблено.

Основним завданням вилучення даних із Web є: отримання певних фрагментів інформації (поля) із зазначених HTML документів у вказаний час.

Вона близька до завдання автоматичної кластеризації і полягає у пошуку розкладання безлічі HTML-документів $D\{d1,...,dn\}$ на класи $C_1,...,C_k$, які містять документи зі схожою структурою.

Завдання відображення прикладних об'єктів точки багатовимірного простору полягає у визначенні базису ознак $\{ei\}$, що формують багатовимірний простір, і методу розкладання документа по цьому базису (тобто обчислення координат $\{wi\}$).

Для визначення координат документа $\{wi\}$ у просторі базисних ознак {ei} використовуються різні підходи. Зокрема, авторами роботи [24] пропонується використовувати підхід, популярний при обчисленні ваг термів у пошукових системах, що використовують векторну модель подання документів. При цьому: $wi=tfi/(logN/k_i)$, де tfi – це частота народження i-го ознаки, k_i – кількість документів, в яких він зустрічається, а N – загальна кількість документів, що розглядаються. Для оцінки якості кластеризації пропонується використовувати ентропійний захід. На підставі обчислення загальної ентропії та ентропії кластера автори роботи [25] запроваджують критерій «хорошого» кластера, який може бути оброблений автоматично. Потім запроваджується ще один захід, що відображає можливість побудови багаторівневої ієрархії кластерів.

Існує ціла низка інших підходів до формалізації завдання вилучення

слабоструктурованої інформації з Інтернет-джерел. Вони використовують логіка ранжированных чи неранжованих дерев першого чи вищого порядку (зазвичай, другого порядку) [25], і навіть логіка на графах [26]. Ці методи відносяться до перспективного напряму сучасної математичної логіки, який активно розвивається. Ефективна реалізація таких методів вимагає розробки нових способів подання та зберігання документів у вигляді деревоподібних (а у загальному випадку – графових) структур, а також мов запитів до них, що потребує великих зусиль на етапі розробки. З погляду простоти реалізації було вирішено використовувати стандартні засоби подання документів та мови запитів.

Для систематизації існуючих підходів можна використовувати класифікацію, запропоновану в роботі, що виділяє такі групи методів та засобів їх реалізації: декларативні мови; використання HTML структури; системи, що здійснюють обробку даних природною мовою; системи з урахуванням індуктивних логічних міркувань; засоби моделювання; засоби, що ґрунтуються на онтології. Ця класифікація не сувора і деякі інструменти можуть ставитись одразу до кількох груп.

Виходячи з виконаного аналізу існуючих методів вилучення, можна зробити висновок, що сторінки HTML навряд чи можуть бути оброблені за допомогою звичайного граматичного аналізу, зважаючи на ряд причин. Основні з них полягають у тому, що розробники HTML-сторінок користуються необмеженою свободою за браку засобів синтаксичного контролю. Наслідком є різноманітність способів структурування даних у сторінці, отже два логічно однорідних елемента даних може бути форматовані зовсім по-різному, і навіть містити помилки, тобто. частини коду можуть не цілком відповідати граматиці HTML (відсутність тегів, що закривають, є поширеним прикладом). Тому ми насамперед орієнтувалися на підходи, які використовують інформацію про HTML-розмітку.

2.2.2 Метод вилучення інформації із слабоструктурованих джерел, заснований на XML

З урахуванням аналізу існуючих підходів та методів вилучення інформації з Інтернет-джерел та проведених експериментальних досліджень низки новинних сайтів, було обрано підхід, заснований на інформації про HTML-розмітку, який використовує як цю інформацію дерево, що відображає структуру Web-сторінки. Це дозволяє використовувати переваги ієрархічної структури тегів.

Для того, щоб мати можливість характеризувати структурні властивості документа, ми розглядаємо його у вигляді дерева розбору згідно з стандартною об'єктною моделлю подання документів DOM (Document Object Model - DOM).

Коренем DOM-дерева для HTML-сторінки є тег «html». Внутрішні вузли дерева відповідають іншим, які у документі HTML-тегам, дуги між якими характеризують вкладеність їх використання. Листя дерева може бути не лише тегами, а й представляти текстові літерали.

Далі виникають дві проблеми, які потрібно вирішити:

– відсутня стандартна мова навігації по HTML;

– відсутня будь-яка стандартна структура HTML-документів.

Тому передбачається, що значно простіше витягувати інформацію з HTML-документів, якщо попередньо привести цей документ до певної відомої нам структурі, і після цього витягувати інформацію з неї [24]. Такою структурою буде XML-подання документа, до якого буде перетворюватись вихідний HTML-документ.

Необхідно також відзначити, що HTML є основним, але не єдиним форматом представлення даних у мережі Інтернет. Тому пропонується наступний підхід: якщо джерелом є HTML-сторінка, вона перетворюється на формат XML, якщо це файл XML (наприклад, RSS), то відповідно перетворення опускається; потім з XML-документа витягуються необхідні дані і

перетворюються на певний формат (їм може бути XML). На рисунку 2.2 дано схематичне представлення запропонованого методу отримання інформації, заснованого на XML.

Застосування цього методу дозволяє вирішити такі задачи:

– забезпечення доступу до джерела даних;

– отримання інформації з HTML-сторінки;

– структурування інформації (редагування неправильно сформованого HTML);

– збереження інформації у форматі XML;

– опитування XML-документа для отримання потрібних користувачеві даних.

Ключовим моментом у запропонованому методі є перетворення HTML-документа, що має слабку структуру, у формат XML. Додаткова складність на цьому етапі виникає через те, що багато сторінок Web-сайтів мають дефектний формат, хоча цілий ряд браузерів, допускають такий формат і можуть його обробити.

Для того, щоб отримати доступ до конкретної частини XML-документа, можливе використання спеціальної мови адресації частин XML-документа – Xpath . Запит на XPath не обмежується обробкою елементів. Можна вказати різні види вузлів, включаючи атрибути, текст, команди обробки та коментарі, або залишити його селектором для будь-якого типу вершини.

За допомогою Xpath надзвичайно спрощується завдання пошуку та опис типів оброблюваних вершин у документі. Істотно скорочується обсяг створюваної програми, оскільки залишається лише обійти дерево, щоб перевірити різні частини. Оскільки Xpath, по суті, представляє стандарт, він може застосовуватися в різних модулях.

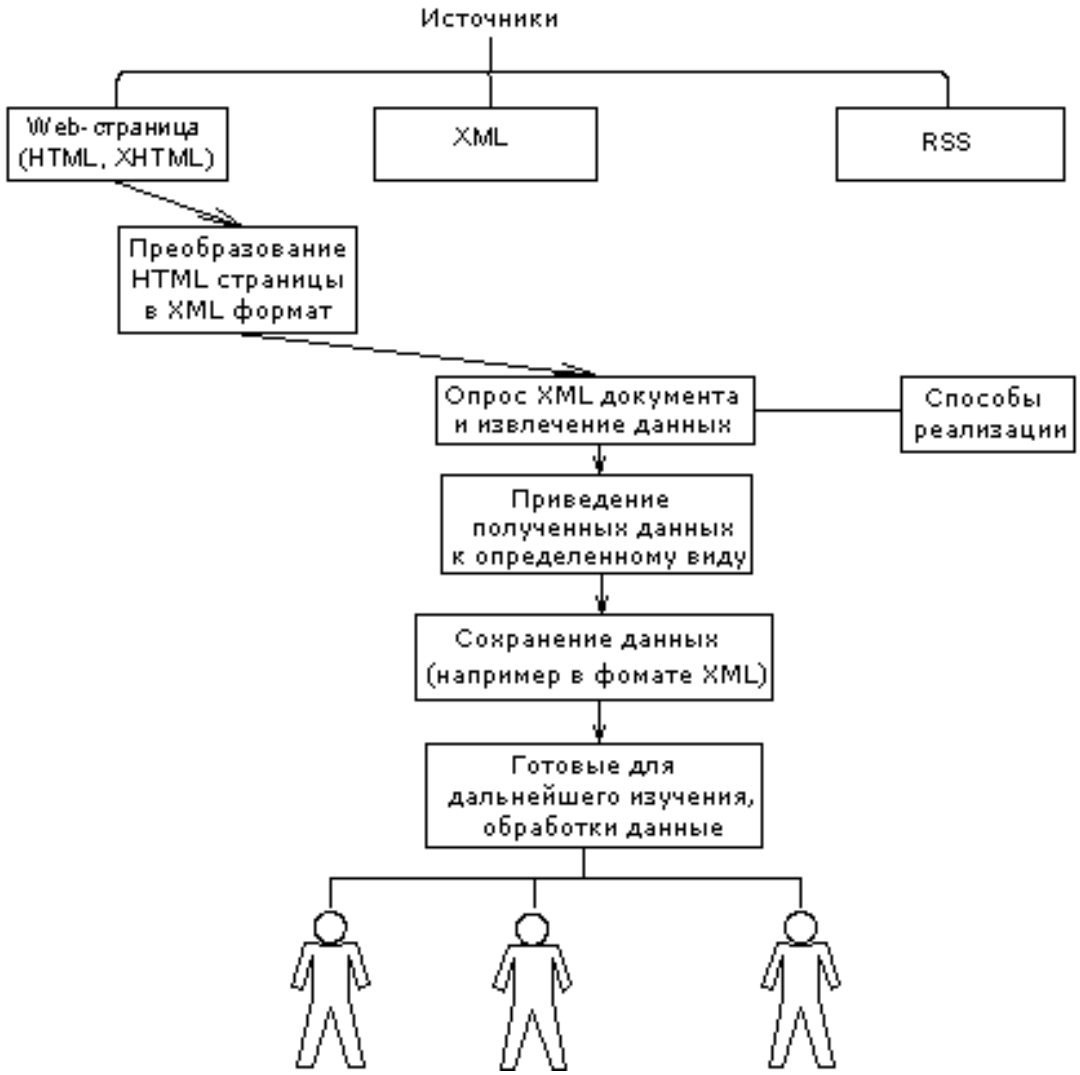

Рисунок 2.2 – Схематичне уявлення методу вилучення

Важливим етапом вилучення є однозначна ідентифікація потрібної інформації у слабоструктурованому джерелі та включення її до результуючого документа (наприклад, XML-документ, структура якого суворо визначена). Зазвичай при цьому використовують інформацію про абсолютний шлях до потрібних елементів або прив'язку до їх змісту за допомогою правил мовою регулярних виразів, а також комбінацію цих способів. Обидва підходи можуть бути успішними для деяких видів джерел, але мало підходять для збору інформації новин, т.к. неможливо прив'язатися до контексту новини, а структура блоків новин на сайті також піддається змінам (змінюється кількість статей новин, деякі блоки в якийсь момент можуть бути порожніми і т.д.). При цьому

комбінація підходів також буде неефективною. Тому в запропонованому методі застосовувалися складніші правила ідентифікації, що використовують, поряд з традиційними способами, прив'язку до типів елементів, які визначаються їх атрибутами і враховують їх розмітку. З цих правил будуються фільтри, які пов'язуються з конкретним джерелом.

2.2.3 Реалізація методу вилучення

Один із варіантів реалізації автоматизованої системи вилучення інформації полягає в написанні окремих посередників для кожного джерела інформації. Саме такий варіант був спочатку реалізований для збирання новин з різнорідних джерел в інформаційній системі «Огляд ЗМІ». При такому способі реалізації для кожного джерела новин доводиться створювати окремий скрипт-посередник, який забезпечує збір новин.

Як подальший розвиток системи вилучення було запропоновано створення універсальної програми (скрипта-посередника), який міг би обробляти будь-яке джерело новин.

Для вирішення цього завдання передбачається таке:

Розділити систему на настроювальну та виконувану частини. При цьому створюється окремий XML-файл, що описує як структуру даних новин, так і схему їх верстки, характерні для кожного джерела (заголовок новини, анотація, дата, час та ін) і єдина програма-скрипт, що використовує при зборі новин даний описувач.

Створити візуальний редактор, який дозволить спеціалісту без спеціальної підготовки (не програмісту), відзначити на сайті необхідний йому блок інформації та реквізити, що містяться у цьому блоці. Основним результатом роботи цього візуального редактора є керування настроювальним файлом.

Можна виділити основні етапи налаштування системи вилучення на новий Інтернет-джерело:

– у редакторі після завантаження HTML-сторінки джерела новин користувач виділяє новинні повідомлення та визначає їх реквізити: заголовок, анотацію, дату, час;

– в результаті розмітки формується шлях до кожного з реквізитів новини, а також складений мовою обробки XML-даних – XPath. При цьому результат розмітки візуально можна спостерігати в редакторі (змінюється колір фону області сайту, що містить повідомлення новин);

– дані заносяться до настроювального файлу. Користувач може перевірити правильність налаштувань;

– створюється (автоматично після виконання користувачем відповідної команди у редакторі) універсальний скрипт, який на основі настроювального файлу збере новини та завантажить їх у вказане місце.

Таким чином, створюється один файл XML, який описує налаштування кожного джерела новин. При зміні верстки будь-якого джерела користувачу не важко швидко внести в нього зміни.

Нижче наведено функціональну схему системи вилучення даних (рисунок 2.3).

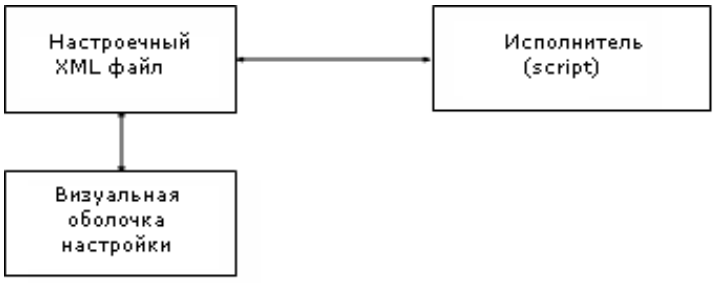

Рисунок 2.3 – Функціональна схема системи вилучення даних

Необхідно додати, що даний підхід, на наш погляд, є універсальним рішенням завдання вилучення інформації зі слабоструктурованих джерел і надалі зможе обробляти будь-яке джерело, в якому людина при початковому

налаштуванні зможе виділити блоки і деякий набір обов'язкових і необов'язкових реквізитів.

Передбачається, що таким же способом можна збирати будь-яку інформацію з WWW-ресурсів, змінюватиметься XML-файл та його описувач (DTD або XSD).

Запропонований підхід був реалізований у прототипі нової версії раніше згаданої інформаційної системи «Огляд ЗМІ», який включає до свого складу засоби отримання даних та знань із структурно різнорідних джерел, створені на основі викладеного підходу. У прототипі реалізовано кілька алгоритмів вилучення, вибір яких здійснюється користувачем, виходячи з того, наскільки строго структурований інформаційний ресурс. Є можливість вибрати суворий варіант вилучення, прив'язавши реквізити, що витягуються, до певного місця його положення на сторінці, так і зробити правила вилучення нечіткими, щоб забезпечити вилучення потрібної інформації в разі невеликих змін у верстці сторінок.

Система виконує автоматичне регулярне зчитування новин з конкретних Інтернет-джерел та розміщення цих новин у локальному сховищі інформації. Як локальне сховище інформації може використовуватися файлова система або база даних. У разі використання бази даних як локального сховища інформації, система виконує систематизацію новин та надає користувачам засоби для контролю та корекції проведеної систематизації, а також засоби формування оглядів новинних повідомлень.

Дані проектованої системи є XML-картки, що містять основні реквізити документа, документи з текстом повідомлення, службові файли описувача та журналу подій.

Перед початком роботи система має бути налаштована адміністратором. Адміністратор за допомогою підсистеми візуальної розмітки здійснює налаштування на конкретні джерела інформації. Внаслідок дій адміністратора відбувається створення XML-файлу описувача.

Підсистема первинної обробки наводить отриману інформацію до формату, необхідного для завантаження до бази. У процесі обробки формується XML-файл, що містить інформацію про реквізити новини, та файл із текстом новини у форматі html або doc. Якщо текст новини представлений у вигляді архіву (zip, rar, arj, tar), файл попередньо розархівується і перевіряється на відповідність допустимим форматам: html або doc. Для забезпечення безпеки, ця підсистема має можливість пересилання даних через FTP-сервер із незахищеної мережі до захищеної. Далі слідує завантаження даних у базу. На цьому закінчується первинна обробка та управління передається ІПС «ODB-text».

Процес збору даних здійснюється модулем збору даних в автономному режимі за встановленим розкладом. Для цього використовується описник, створений адміністратором на етапі налаштування. Дані про лічені документи та результат виконання процесу фіксуються в журналі подій. В основу алгоритму роботи модуля збору даних покладено дворівневу схему доступу до даних.

При розробці прототипу системи використовувалися мови Jscript для візуальної розмітки джерела та створення описувача, а також Perl для виконання автоматичного вилучення даних та їх первинної обробки.

2.3 Метод автоматичної побудови онтології предметної галузі на основі аналізу лінгвістичних характеристик текстового корпусу

Автоматичне вилучення знань із монологічних текстів з метою побудови онтології передбачає як виявлення термінів, а й вилучення знання термінах. Це означає, що з опису семантичної структури термінології необхідно розпізнати у тексті як терміни, і семантичні відносини між термінами. Але перш ніж розглядати типи семантичних відносин необхідно вибрати підхід, на основі якого слід виконувати аналіз пропозиції монологічного тексту.

2.3.1 Основні засади побудови онтології

Розглядаємо підхід до побудови онтології на основі тексту, запропонований Buitelaar та колегами в 2005 р., відомий як Layer Cake. Весь процес побудови онтології розбитий кілька незалежних етапів, кожному з яких вирішується одне певне завдання, результати якої, своєю чергою, служать вихідними даними завдання наступного, зазвичай, складнішого рівня. Відповідно до цього підходу можна виділити таку послідовність дій:

– вилучення з тексту термінів-кандидатів;

– розбиття термінів на групи (кластеризація);

– присвоєння узагальнюючого поняття-концепту кожній групі;

– визначення відносин між концептами;

– формування правил виведення (розширення концептів).

Для початкових етапів численні статистичні методи дозволяють отримувати дуже якісні результати (до 90% точності, порівняно з результатами роботи експертів).

Однак для побудови таксономій понять та виявлення не таксономічних відносин необхідна розробка лінгвістичних методів. У лінгвістичних шаблонів, запропонованих щеM. Hearst [16]. Подальше вдосконалення такого підходу залежить від наявності загальної теорії опису та аналізу конкретної мови. У цьому автор поділяє ідею Золотовий Р. А. у тому, що синтаксичний лад промови (тексту) організується регулярними комбінаціями «елементарних» одиниць, у тому числі будуються й інші складніші (мовні чи текстові) конструкції. Як така одиниця висунуто поняття синтаксеми. Синтаксемою по Золотовій Г.А. називається мінімальна далі неподільна семантико-синтаксична одиниця російської мови, що виступає одночасно і як носій елементарного сенсу, і як конструктивний компонент з функціональністю, необхідною та достатньою для побудови складніших синтаксичних конструкцій [17].

Ознаками синтаксеми є:

– категоріальне значення слова, від якого вона утворена;

– певна (конкретна) морфологічна форма;

– функціональність, яка з двох перших ознак, як здатність реалізовуватися у певних позиціях (промови, тесту),тобто. як можлива роль у побудові комунікативної одиниці. Золотова Г.А. розрізняє три такі функціональні ролі:

– можливість самостійного незалежного існування синтаксеми;

– використання синтаксеми як компонент пропозиції;

– використання синтаксеми як компонент словосполучення (або поєднання слів), так зване прислівне вживання.

Функціональні властивості синтаксем служать основою для розмежування трьох їх можливих типів, званих відповідно, вільними – А (або володіють повним функціональним репертуаром), зумовленими – Б (здатними виконувати функції 2, 3, вкрай зрідка – 1), пов'язаними – В (виступають тільки в ролі 3). Вочевидь, що функціональні властивості синтаксем мають транзитивний характер.

Розглядаючи текст як динамічну систему, ми маємо можливість описати весь процес побудови онтології як цілеспрямовану операційну діяльність у межах цієї системи, організовану на вирішення завдань змістовного наповнення елементів онтології.

Математично онтологія визначається так:

$$O := (C, \leq c, R\sigma r, \leq r, A\sigma a, T), \tag{3.1}$$

де *C, R, A, T* – є незв'язаними множинами, чиї елементи називаються ідентифікаторами концептів, відносин, атрибутів та типів даних відповідно;

$\leq c$ – напівпов'язана таксономія (semi-upperlattice) концептів із загальним елементом найвищого рівня rootC;

функція $\sigma R: R \rightarrow C+$ звана ознакою відношення (relation signature);

$\leq R$ *on* R, ієрархія відносин де, $r1 \leq R\ r2$ має на увазі $=|\sigma R(r1)| = |\sigma R(r2)|$ *i* $\pi i(\sigma R(r1)) \leq c\ \pi i(\sigma R(r2))$ для кожного $\leq i \leq = |\sigma R(r1)|$;

функція $\sigma A: A \rightarrow C \times T$ звана ознакою атрибуту (attribute signature);

безліч типів даних (Datatypes) T , таких як рядки, цілі числа і т.д.

Для забезпечення мовної компетентності, достатньої для побудови та самонавчання онтології предметної області на базі тексту, ІС сама повинна володіти знаннями відповідного порядку, загальними (мовними) та спеціальними (що стосуються даного ПЗ). Така ІС повинна, по суті, поєднувати в собі дві «вбудовані» онтології: онтологію мови (російської) та базову (стартову) онтологію ПЗ. Численні електронні словники-тезауруси (MRD – machine readable dictionary), з лексикографічною розміткою (які як WordNet та її російськомовні аналоги RussNet, RuNet та інших.) можна як елементи З і ≤с необхідної лінгвістичної онтології. Парсери мови (наприклад, проект АОТ) можуть бути успішно використані як визначення ролей концептів, так визначення їх атрибутів. Те, чого дійсно не вистачає для повноцінної інтелектуальної системи, так це систематизованого репертуару лексико-синтаксичних одиниць мови, що несуть у собі семантику, що однозначно трактується, і одночасно виконують роль «елементарних одиниць складання» висловлювань (текстів). Перетин такого репертуару «архе-функцій», несомих синтаксичною формою і репертуару слів-носіїв категоріальних референцій, дає проекцію ролі, що однозначно трактується (функції) виконуваної даним поняттям. Подібна систематизуюча праця, як було сказано вище, виконана значною мірою Золотовою Г.А. і чекає на гідну технічну реалізацію.

Пропонований підхід автоматизованої побудови онтології ПЗ поєднує в собі швидкість статистичних методів та точність лінгвістичного підходу з позицією синтаксем (або лінгвістичних шаблонів).

Процес побудови онтології складається з кількох послідовних етапів.

Етап 1. Розмітка корпусу синтаксичними тегами (наприклад,

інформацією про частину мови, рід числа і т.д.).

Етап 2. Визначення характерних, тобто. найчастіше які у корпусі лінгвістичних структур і формування з їхньої основі лінгвістичних шаблонів.

Етап 3. Пошук фрагментів тексту, які відповідають даним шаблонам.

Етап 4. Нормалізація та додавання до базової онтології нових концептів, виявлених на попередньому етапі.

На третьому етапі необхідно залучення експерта предметної галузі з метою оцінки релевантності знайдених потенційних концептів.

Проілюструємо сказане прикладом аналізу наступного тексту: «Після аварійного опромінення великої потужності розвиток поразок у часі проходить через чітко визначені стадії. Тривалість та час настання стадій залежить від дози. Малі дози не дають ефектів, що спостерігаються. Типовий розвиток подій після опромінення всього тіла від джерела проникаючої радіації включає початкову продромальну стадію з такими симптомами, як нудота, блювання, втома і, можливо, лихоманка та діарея, після чого слідує латентний період різної тривалості.»

Модуль лінгвістичної онтології ІВ розпізнає такі властивості концептів: квалітатив (якість) суб'єкта та його об'єкт (рисунок 2.4). У російській це може виражатися такими формулами:

– допоміжними дієсловами {відрізнятися, висловлюватись, виявлятися, характеризуватись, володіти} + Т.п. ознакового слова (передуючий компонент);

– за допомогою родового відмінка, коли попередній компонент визначений поєднанням {прикметник + іменник};

– в експліцитній формі конструкцією типу «об'єкт може мати такі {характеристики, властивості, ознаки}».

– за допомогою пов'язаної синтаксеми з родовим відмінком, що позначає об'єкт при віддієслівному іменнику.

Виявлення синтаксичних характеристик виконується парсером

російської (АОТ-парсер), здатним розпізнавати граматичні форми слів, що зустрічаються. Для правильної роботи визначається порядок застосування (пріоритет) правил.

Визначаються характеристики концепту «опромінення», тобто. система застосовує описані правила лише до пропозицій, у яких виявить будь-яку словоформу «опромінення».

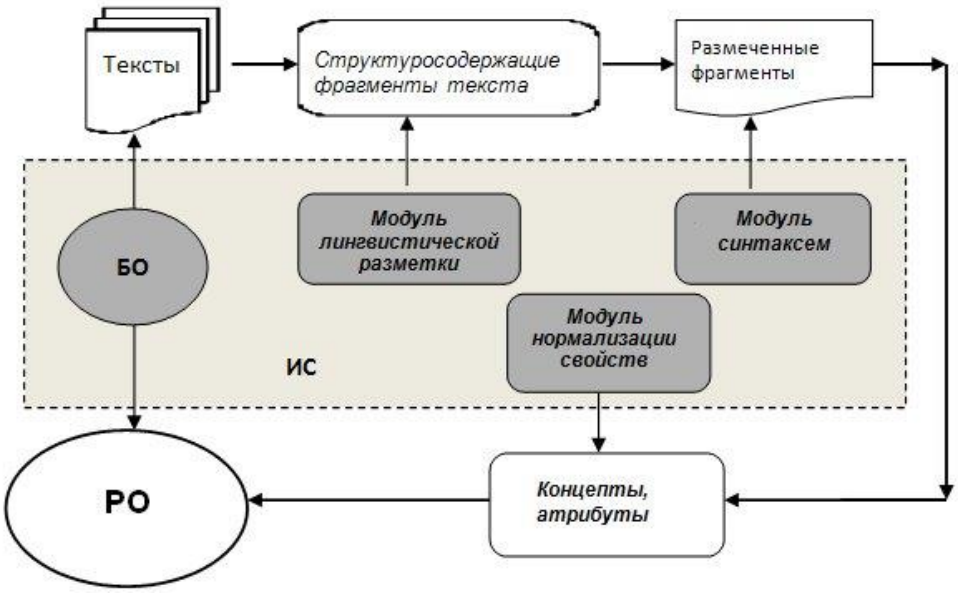

Рисунок 2.4 – Фреймворк ІС на основі синтаксем

Таким чином, маємо можливість виявити в тексті та витлумачити, наприклад, такі властивості терміна: «…опромінення великої потужності» – Правило 2; «потужність» – атрибут концепту «опромінення»; «…опромінення всього тіла» – правило 4; предикат «опромінювати» та його об'єкт «тіло»;

Перевагою запропонованого методу і те, що «точками входу» для аналізу тексту можуть і слова і власне синтаксеми.

2.4 Класифікації онтологій

У проектуванні онтологій умовно можна виділити два напрямки, які до

певного часу розвивалися окремо. Перше пов'язані з поданням онтології як формальної системи, заснованої на математично точних аксіомах. Другий напрямок розвивався в рамках комп'ютерної лінгвістики та когнітивної науки. Там онтологія розумілася, як система абстрактних понять, що існують лише у свідомості людини, яка може бути виражена природною мовою (або якоюсь іншою системою символів). При цьому зазвичай не робиться припущень про точність чи несуперечність такої системи.

Таким чином, існує два альтернативні підходи до створення та дослідження онтологій. Перший (формальний) заснований на логіці (предикатів першого порядку, дескриптивної, модальної тощо). Другий (лінгвістичний) заснований на вивченні природної мови (зокрема, семантики) та побудові онтологій на великих текстових масивах, так званих корпусах.

Нині ці підходи тісно взаємодіють. Йде пошук зв'язків, що дозволяють комбінувати відповідні методи. Тому іноді буває складно відокремити лексичні онтології з елементами формальних аксіоматик від логічних систем із включеннями лінгвістичних знань.

Незалежно від різних підходів можна виділити 3 основні принципи класифікації онтологій:

– за ступенем формальності
– за наповненням, вмістом
– за метою створення

В рамках цієї класифікації за метою створення виділяють 4 рівні: Онтології уявлення, онтології верхнього рівня, онтології предметних областей та прикладні онтології.

Мета створення онтології уявлення - описати область уявлення знань, створити мову специфікації інших онтологій нижчих рівнів. Приклад: опис понять мови OWL засобами RDF/RDFS.

Призначення онтології верхнього рівня у створенні єдиної “правильної онтології”, яка фіксує знання загальні всім предметних областей і

багаторазовому використанні даної онтології. Існує кілька серйозних проектів: Sumo, Sowa's Ontology, Cyc. Але загалом спроби створити онтологію верхнього рівня на всі випадки життя поки що не призвели до очікуваних результатів. Багато онтології верхнього рівня схожі одна на одну. Вони містять одні й самі концепти: Сутність, Явище, Процес, Об'єкт, Роль тощо.

Призначення онтологій предметних областей схоже на призначення онтологій верхнього рівня, але область інтересу обмежена предметною областю (авіація, медицина, культура). Приклади: Авіаонтологія, CIDOC CRM, UMLS.

Призначення Прикладні онтології цих онтологій у тому, щоб описати концептуальну модель конкретної задачі чи додатку. Вони містять найспецифічнішу інформацію. Приклад проектів: TOVE, Plinius.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

У ході виконання кваліфікаційної роботи було створено елементи, які дозволяють отримувати знання з тексту. Як приклад було обрано текстовий документ, у якому йдеться про опис велозапчастин. Для наочності було запропоновано програмний засіб, написаний на Java, з дружнім інтерфейсом, який не ускладнить користувача виконувати поставлені завдання. Запускаючи програму, можна побачити вікно, в якому є поле для вказівки адреси файлу, а також кнопки: «Завантажити» та «Добути знання» (рисунок 3.1).

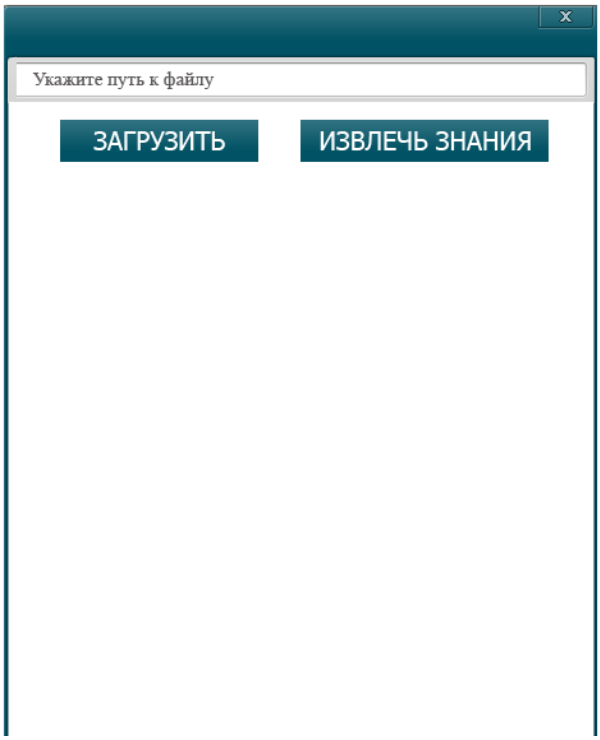

Рисунок 3.1 – Головне вікно програми

У відповідному полі вказується шлях до файлу, потім натисніть кнопку «Завантажити». (рисунок 3.2)

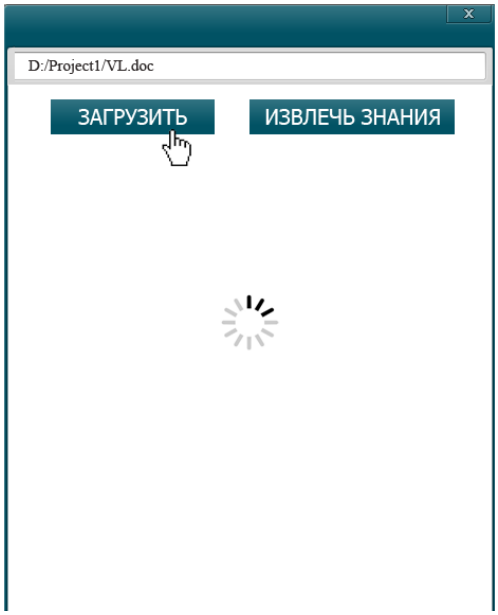

Рисунок 3.2 – Завантаження документа

Після того, як програма завантажила документ, висвічується додаткове вікно із повідомленням про успішне завантаження (рисунок 3.3). Операція займає близько 2-6 секунд, залежно від розміру файлу.

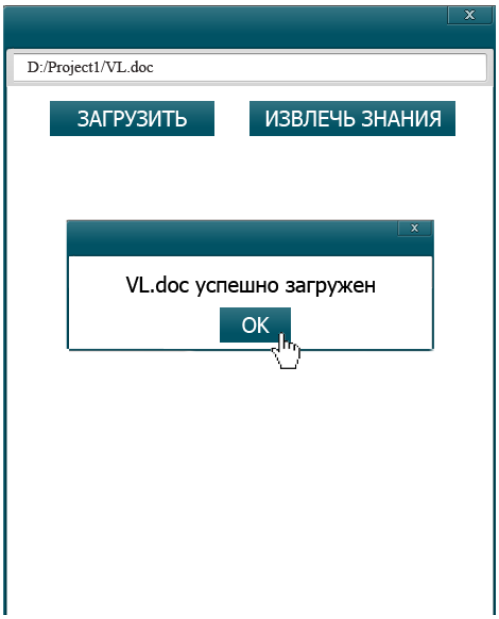

Рисунок 3.3 – Повідомлення про завантаження

Натиснувши кнопку «Здобути знання», програма завантажить додаткове вікно з текстом із вже завантаженого документа, де будуть виділені супер-клас, класи та підкласи для майбутньої автоматично побудованої онтології (рисунок 3.4).

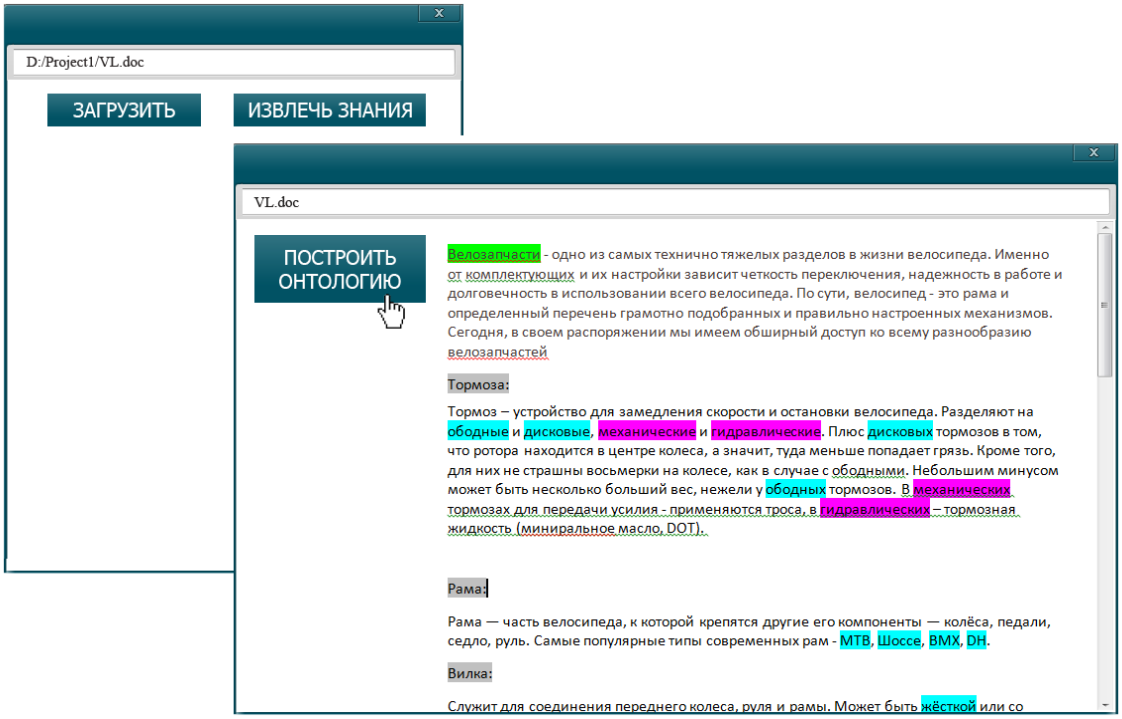

Рисунок 3.4 – Текст із файлу

У тексті різними кольорами виділяються супер-клас (зелений), його класи (сірий), а також підкласи (бірюзовий). Рожевим кольором виділення ті слова, які відносяться безпосередньо до підкласів велозапчастин. Вилучення автоматично збудованої онтології проводилося за допомогою редактора моделювання онтологій Protégé 2000 (рисунок 3.5).

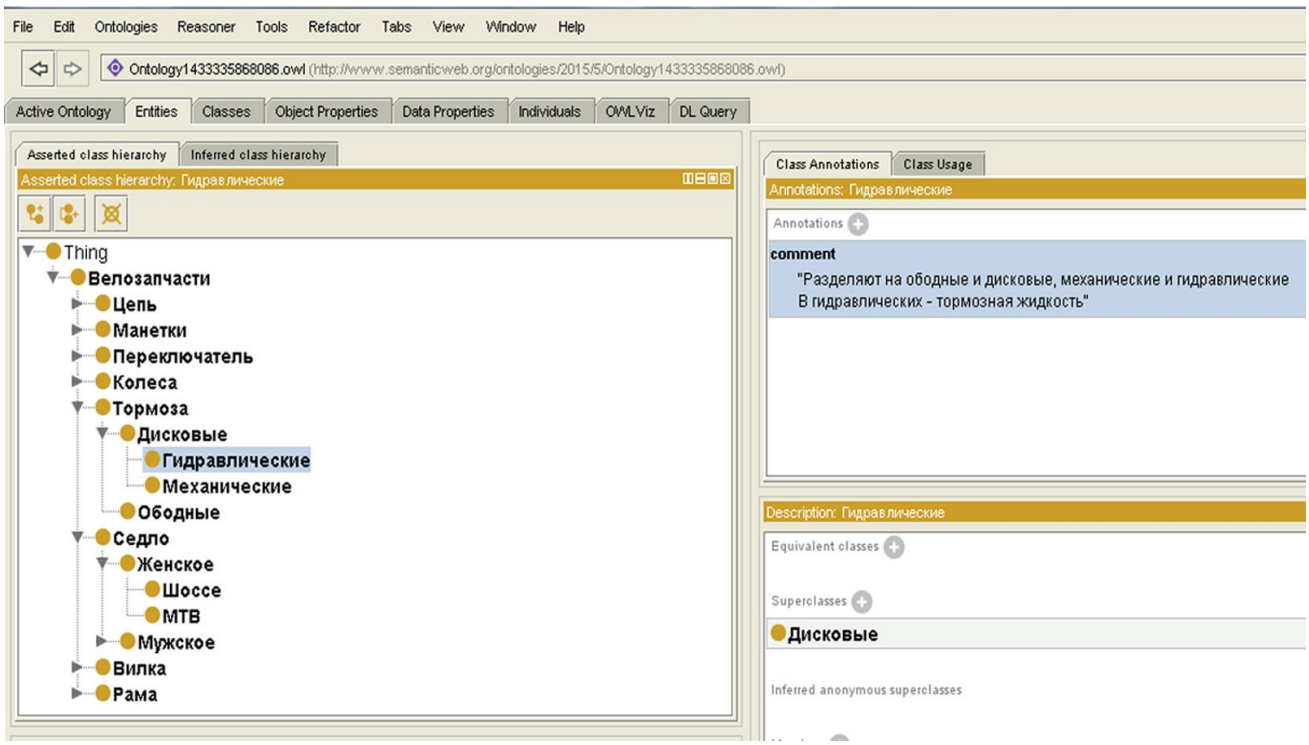

Рисунок 3.5 – Автоматично побудована онтологія

І так є супер-клас «Велозапчастини», його класи: «Ланцюг», «Манетки», «Перемикач», «Колеса», «Тормоза», «Сідло», «Вилка» та «Рама». Натиснувши кнопку «Побудувати онтологію», у середовищі Protégé 2000 можна побачити автоматично побудовану онтологію.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено аналіз предметної галузі, дослідження теоретичної основи для розробки програмного додатку, а також розроблено алгоритм, який дозволяє отримувати знання, а також виробляти автоматичну побудову онтологій.

У процесі теоретичних досліджень було встановлено, що запропонований алгоритм пріоритетним, т.к. в ньому є такі переваги: висока точність, простота і в реалізації, а також є досить універсальним і не обмежує застосування методу лише російською мовою.

Як редактор для перегляду онтологій було обрано вільно розповсюджуване середовище Protégé 2000, яке дозволяє в ієрархічному та графічному вигляді переглядати автоматично створену онтологію.

Було реалізовано програмний продукт, який дозволяє робити висновок із онтології.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Brickley D., Guha R.V. RDF 1.0: RDF Schema. URL:http://www.w3.org/TR/rdf-schema/ (дата звернення: 15.04.2014).

2. Гаврилова Т. А., Хорошевський В. Ф. Основи знань інтелектуальних систем: навч. Для ВНЗ. СПб.: Пітер, 2000. 384 с.

3. Захарова І. В. Математична модель семантичного пошуку з використанням онтологічного підходу: Автореф. дис. канд. фіз.-мат. наук. Челябінськ, 2009. 20 с.

4. Клєщов А. С., Артем'єва І. Л. Математичні моделі онтологій предметних областей. Частина 1. Існуючі підходи до визначення поняття «онтологія». Науково-технічна інформація, серія 2. «Інформаційні процеси та системи». М.: ВІНІТІ, 2001. № 2. С. 20-27.

5. Найханова Л. Ст. Методи і моделі автоматичного побудови онтологій на основі генетичного і автоматного програмування: Автореф.дис.докт. тех. наук. Красноярськ, 2008. 36 с.

6. Рабчевський Е. А. Автоматичне побудова онтологій. *Науково-технічні відомості Санкт-Петербурзького державного політехнічного університету*. СПб.: Видавництво Політехнічного Університету, 2007. № 52-2. С. 22–26.

7. Рабчевський Е. А. Автоматичне побудова онтологій на о знову лексико-синтаксичних шаблонів для інформаційного пошуку. Праці 11-ї Всеросійської наукової конференції «Електронні бібліотеки: перспективні методи та технології, електронні колекції» – RCDL'2009. Петрозаводськ, 2009. С. 69–77.

8. Росїїва О. І., Загорулька Ю. А.Організація ефективного пошуку на основі онтологій. Праці міжнародного семінару Діалог'2001 з комп'ютерної лінгвістики та її додатків. Аксакове, 2001. Т. 2. С. 333–342.

9. Carroll J. J., Dickinson I., Dollin C., Reynolds D., Seaborne A., та

Wilkinson, K. Jena: здійснюючи Semantic Web recommendations. Proc. of the 13th Int. World Wide Web Conf. на Alternative Track Papers & Posters. USA, 2004. P. 74–83.

10. Ding L., Kolari P., Ding Z., Avancha S., Finin T., Joshi A. За допомогою ontologies в Semantic Web: a survey. *Ontologies: handbook of principles, concepts and applications in information systems*. Springer US, 2006. P. 79–114.

11. Gennari JH, Musen MA, Fergerson RW, Grossolo WE, Crubézy M., Eriksson H., Noy NF, Tu SW Розвиток захисту: природа для пізнання базованих систем розвитку. *International Journal of Human-Computer Studies*. 2003. Vol. 58. №1. P. 89–123.

12. Gruber T.R. Translation approach до портативної онтології specification. *Knowledge Acquisition*.1993. Vol. 5. № 1. С. 199–220.

13. Pantel P., Lin D. Статистичний корпус-базований терм extractor. Proc. of Canadian Conf. on AI. 2001. С. 36–46.

14. Syafrullah M., Salim N. Improving Term Extraction За допомогою Particle Swarm Optimization Techniques. *Journal of Computer Science*. 2010.Vol. 6. № 3. С. 323–329.

15. Hovy E. A Standard for Large Ontologies. URL: http://www.isi.edu/nsf/papers/hovy2.htm (дата звернення 10.10.2021).

16. Hovy E. Презентації Ontologies: lecture 1 , lecture 2 Issues of Content, lecture 3 Methods for Automated Ontology Building. Information Sciences Institute University of Southern California., з XX літньої школи ім. В.Матезіуса з лінгвістики та семіотики (7-12 березня 2005 р., Карлів університет).

17. Nirenburg S., Raskin V. Ontological Semantics. Cambridge, MA, 2004.

Добров Б.В., Лукашевич Н.В. Вторинне використання лінгвістичних онтологій: зміна у структурі концептуалізації. URL: http://www.rcdl2006.uniyar.ac.ru/papers /paper_78_v1.pdf (дата звернення 12.10.2021).

18. Коваль С.А. Безекземплярні та екземплярні онтології. Матеріали XXXVI. філолог. конф. 12 березня 2007 р. у Санкт-Петербурзькому

університеті (у пресі). також матеріали на URL: http://skowal.narod.ru/research/ontology2007 (дата звернення 15.10.2021).

19. Voutilainen A. Natural Language Components for Smart Applications. Elsnews, winter. 2003/4. № 12.4. С. 11–14.

20. Batchilo L.S., Sovpel I.V., Tsourikov V.M. Document semantic analysis. December, 2000. С. 12–15.

21. Batchilo L.S. Natural language processing and query driven information retrieval: US Patent Appl. №20020010574.

22. Гаврилова Т.А. Використання онтологій у системах управління знаннями / Праці Міжнар. конгресу «Штучний інтелект у XXI столітті». Т. 1. Дивноморське (Росія). 2001. С. 21–32.

23. Голубєв С.А. Досвід впровадження та використання інформаційно-пошукової системи ODB-Text у Раді Федерації Федеральних Зборів РФ. Збірник наукових праць. ФГУП НДІ «Схід», МИРЕА. 2001. С. 58–61.

24. Laender A.H.F., Ribeiro-Neto B.A. A brief survey of web data extraction tools. ACM SIGMOD Record 31 (2). 2002. С. 84–93.

25. Неселян І. Виявлення структурної подоби HTML-документів. СПДУ. 2002. С. 38–54