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)

Гібридне виведення знань в інформаційних системах з використанням великих мовних моделей та графів знань
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-1
Юрій БРАЖНЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Ваган ТЕРЗІЯН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Бражненку Юрію Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Гібридне виведення знань в інформаційних системах з використанням великих мовних моделей та графів знань

затверджена наказом університету від 6 квітня 20 26 р. № 269Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 19 травня 20 26 р.

3. Вихідні дані до роботи наукові статті та відкриті джерела з тематики графів знань та технологій Semantic Web (RDF, OWL, SPARQL), матеріали щодо великих мовних моделей (LLM) та їх застосування, а також методичні матеріали кафедри штучного інтелекту щодо структури та оформлення кваліфікаційних робіт

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі та методів представлення та виведення знань
2) Використання великих мовних моделей у роботі зі знаннями
3) Модель гібридного виведення знань
4) Експериментальне дослідження гібридної системи
5) Дослідження ефективності гібридної інтелектуальної системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	26.01.2026	виконано
2	Аналіз предметної галузі та літературних	27.02.2026	виконано
	джерел		
3	Дослідження графів знань та технологій	06.03.2026	виконано
	Semantic Web		
4	Аналіз LLM та їх можливостей	13.03.2026	виконано
5	Розробка моделі гібридного виведення знань	19.03.2026	виконано
6	Реалізація експериментальної частини	29.03.2026	виконано
7	Аналіз результатів та оформлення роботи	11.04.2026	виконано
8	Подання роботи та підготовка до захисту	28.04.2026	виконано
9	Захист	19.05.2026	

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Ваган ТЕРЗІЯН
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 89 с., 14 рис., 2 табл., 1 дод., 20 джерел.

ВЕЛИКІ МОВНІ МОДЕЛІ, ГІБРИДНЕ ВИВЕДЕННЯ ЗНАНЬ, ГРАФИ ЗНАНЬ, ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ, LLM, OWL, RDF, SEMANTIC WEB, SPARQL.

Об'єкт дослідження – процеси подання та виведення знань в інтелектуальних інформаційних системах.

Предмет дослідження – методи гібридного виведення знань, що поєднують графи знань, онтології Semantic Web та великі мовні моделі.

Мета роботи – дослідження та аналіз підходів гібридного виведення знань в інформаційних системах на основі поєднання графів знань і великих мовних моделей, а також оцінка їх ефективності у порівнянні з традиційними символічними та нейронними підходами.

Методи дослідження – аналіз сучасних підходів до подання та обробки знань, онтологічне моделювання, логічне виведення на основі правил, методи порівняльного аналізу, експериментальна оцінка ефективності різних підходів до виведення знань.

У результаті роботи проведено аналіз підходів до подання знань у графах знань та онтологіях Semantic Web, а також досліджено можливості використання великих мовних моделей для роботи зі знаннями. Розглянуто архітектури гібридних систем, які поєднують символічне логічне виведення та нейронні методи обробки знань. Запропоновано модель гібридного виведення знань. Проведено експериментальне порівняння трьох підходів: символічного логічного виведення на основі графів знань, нейронного підходу з використанням великих мовних моделей та гібридного підходу, який поєднує обидва методи.

ABSTRACT

Master’s thesis contains: 89 pp., 14 fig., 2 tabl., 1 ann., 20 references.

HYBRID KNOWLEDGE INFERENCE, INTELLIGENT INFORMATION SYSTEMS, KNOWLEDGE GRAPHS, LARGE LANGUAGE MODELS, LLM, OWL, RDF, SEMANTIC WEB, SPARQL.

The object of research is the processes of knowledge representation and inference in intelligent information systems.

The subject of research is hybrid knowledge inference methods that combine knowledge graphs, Semantic Web ontologies, and large language models.

The purpose of the work is to study and analyze hybrid approaches to knowledge inference in information systems based on the integration of knowledge graphs and large language models, as well as to evaluate their effectiveness in comparison with traditional symbolic and neural approaches.

Research methods**:** analysis of modern approaches to knowledge representation and processing, ontological modeling, rule-based reasoning, comparative analysis, and experimental evaluation of knowledge inference methods.

As a result of the work, approaches to knowledge representation in knowledge graphs and Semantic Web ontologies were analyzed, and the possibilities of using large language models for knowledge processing were investigated. Architectures of hybrid systems combining symbolic logical reasoning and neural methods were studied. A hybrid knowledge inference model was proposed. An experimental comparison of three approaches was conducted: symbolic reasoning based on knowledge graphs, neural reasoning using large language models, and a hybrid approach combining both methods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ABox – Assertional Box – фактичні дані онтології;

AI – Artificial Intelligence – штучний інтелект;

DL – Deep Learning – глибинне навчання;

KG – Knowledge Graph – граф знань;

LLM – Large Language Model – велика мовна модель;

ML – Machine Learning – машинне навчання;

NLP – Natural Language Processing – обробка природної мови;

OWL – Web Ontology Language – мова опису онтологій;

RDF – Resource Description Framework – модель представлення даних у вигляді трійок;

SPARQL – SPARQL Protocol and RDF Query Language – мова запитів до RDF-графів;

SWRL – Semantic Web Rule Language – мова правил для онтологій;

TBox – Terminological Box – опис структури онтології.

ВСТУП

У сучасних інформаційних системах постійно зростає обсяг даних, з якими необхідно працювати. Це створює потребу не лише у зберіганні інформації, але й у її структурованому представленні, аналізі та отриманні нових знань на основі вже наявних даних. Одним із ключових напрямів розвитку інтелектуальних інформаційних систем є використання методів представлення знань і механізмів логічного виведення.

Традиційно для представлення знань використовуються символічні моделі, зокрема онтології та графи знань. Вони дозволяють формально описувати об’єкти предметної галузі, їх властивості та взаємозв’язки між ними. Завдяки цьому стає можливим виконувати логічне виведення нових знань на основі заданих фактів і правил. Такі підходи широко застосовуються у системах підтримки прийняття рішень, експертних системах, пошукових системах та інших інтелектуальних інформаційних технологіях.

Проте символічні методи мають певні обмеження. Вони потребують чіткої формалізації знань і значних зусиль для створення та підтримки онтологій і баз знань. Крім того, такі системи погано працюють з неповними або неструктурованими даними та мають обмежену гнучкість при обробці природної мови.

Останніми роками значного розвитку набули великі мовні моделі, які здатні аналізувати природну мову, інтерпретувати запити користувачів і генерувати текстові відповіді. Такі моделі використовуються в різних інтелектуальних системах, зокрема у чат-ботах, системах пошуку інформації та автоматичному аналізі текстів. Вони можуть працювати з великими обсягами неструктурованих даних та знаходити складні закономірності у текстовій інформації.

Разом з тим, великі мовні моделі також мають певні недоліки. Однією з основних проблем є явище так званих галюцинацій, коли модель

генерує правдоподібні, але фактично неправильні або неперевірені відповіді.

У зв’язку з цим, актуальним напрямом досліджень є поєднання символічних методів представлення знань із можливостями великих мовних моделей. Такий підхід отримав назву гібридного або нейросимволічного виведення знань. Його основна ідея полягає у використанні графів знань та онтологій для формального представлення фактів і правил, а також застосуванні великих мовних моделей для обробки природної мови, інтерпретації запитів і допоміжного аналізу інформації. Поєднання цих підходів може дозволити отримати більш точні, гнучкі та інтерпретовані результати в інтелектуальних інформаційних системах.

Таким чином, дослідження методів гібридного виведення знань, що поєднують графи знань і великі мовні моделі, є актуальним науковим завданням і має важливе значення для розвитку сучасних інформаційних технологій.

Гіпотеза дослідження полягає в тому, що використання гібридного підходу до виведення знань забезпечує вищу коректність результатів та кращу інтерпретованість у порівнянні з використанням лише символічних або лише нейронних методів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати сучасні підходи до представлення знань в інформаційних системах;

– дослідити принципи побудови та використання графів знань;

– розглянути можливості використання великих мовних моделей у задачах роботи зі знаннями;

– проаналізувати основні проблеми використання великих мовних моделей, зокрема проблему галюцинацій;

– дослідити підходи до поєднання символічних і нейронних методів у задачах виведення знань;

– розробити концептуальну модель гібридного виведення знань;

– провести експериментальне порівняння символічного, нейронного та гібридного підходів;

– проаналізувати отримані результати та визначити переваги і обмеження запропонованого підходу.

Наукова новизна роботи полягає у дослідженні можливостей поєднання графів знань та великих мовних моделей для реалізації гібридного механізму виведення знань, а також у проведенні порівняльного аналізу ефективності символічного, нейронного та гібридного підходів.

Практичне значення отриманих результатів полягає у можливості використання запропонованого підходу при розробці інтелектуальних інформаційних систем, систем підтримки прийняття рішень, пошукових систем та інтелектуальних асистентів, у яких необхідне поєднання структурованих знань і обробки природної мови.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА МЕТОДІВ ПРЕДСТАВЛЕННЯ ТА ВИВЕДЕННЯ ЗНАНЬ

1.1 Аналіз предметної галузі

1.1.1 Поняття даних, інформації та знань

Основою будь-якої інформаційної системи є дані. Дані можна розглядати як набір окремих фактів, числових значень, символів або записів, що описують певні події, об'єкти або процеси. Вони можуть бути представлені у вигляді тексту, числових значень, зображень, сигналів або інших форм представлення [2], [3].

Важливо зазначити, що самі по собі дані не завжди мають зрозумілий зміст. Без відповідного контексту вони являють собою лише набір символів або значень, які не дозволяють зробити будь-які висновки.

Наприклад, окремі значення температури, часові мітки або ідентифікатори користувачів є типовими прикладами даних, які можуть зберігатися в інформаційних системах. Проте без додаткової інтерпретації такі дані не несуть значної практичної цінності.

Коли дані проходять певний процес обробки, структурування або інтерпретації, вони перетворюються на інформацію. Інформація формується тоді, коли дані отримують контекст або логічні зв’язки між окремими елементами. У цьому випадку дані стають зрозумілими та можуть бути використані для аналізу або прийняття рішень.

Наприклад, якщо значення температури поєднати з інформацією про дату та місце вимірювання, можна отримати інформацію про зміну температури в певному регіоні протягом визначеного періоду часу. Таким чином, дані набувають змісту та можуть бути використані для подальшого аналізу.

Інформація має декілька важливих характеристик. По-перше, вона повинна бути структурованою або організованою таким чином, щоб її можна було ефективно обробляти. По-друге, інформація повинна мати певний контекст, що дозволяє правильно інтерпретувати її значення. По-третє, інформація повинна бути релевантною, корисною для вирішення конкретної задачі.

Наступним рівнем у цій ієрархії є знання. Знання формуються в результаті аналізу, узагальнення та інтерпретації інформації. Вони дозволяють робити висновки, встановлювати закономірності та передбачати можливі результати певних процесів.

На відміну від інформації, знання включають не лише окремі факти, але й зв’язки між ними. Завдяки цьому з’являється можливість формувати нові твердження на основі вже наявної інформації. Саме ця властивість знань є ключовою для побудови інтелектуальних систем, здатних виконувати логічне виведення.

У науковій літературі часто використовується модель DIKW-ієрархії (Data – Information – Knowledge – Wisdom), яка описує процес перетворення даних у знання. У межах цієї концепції дані розглядаються як базовий рівень інформаційної системи, інформація формується в результаті обробки даних, а знання виникають у процесі аналізу та узагальнення отриманої інформації [2].

1.1.2 Роль знань в інтелектуальних інформаційних системах

У сучасних інтелектуальних інформаційних системах знання відіграють ключову роль. Вони використовуються для формування логічних висновків, підтримки прийняття рішень та автоматизації складних аналітичних процесів. Саме тому важливим завданням є розробка ефективних методів представлення знань, які дозволяють комп’ютерним системам працювати з інформацією на більш високому рівні [6], [7].

Знання у таких системах можуть бути представлені у вигляді різних структур, зокрема логічних правил графів знань або онтологічних моделей.

Наприклад, у експертних системах знання часто описуються за допомогою набору правил типу «якщо-то», що дозволяє системі формувати висновки на основі заданих фактів.

У свою чергу, графи знань дозволяють представляти інформацію у вигляді мережі взаємопов’язаних об’єктів, де вузли відповідають сутностям предметної галузі, а зв’язки між ними відображають певні логічні або семантичні відношення. Такий підхід значно спрощує процес інтеграції різних джерел інформації та дозволяє будувати складні системи аналізу знань.

Окрему увагу слід приділити різним типам даних, з якими працюють сучасні інформаційні системи. Зазвичай їх поділяють на три основні категорії: структуровані, напівструктуровані та неструктуровані дані.

Структуровані дані мають чітко визначену структуру та зазвичай зберігаються у реляційних базах даних у вигляді таблиць. Напівструктуровані дані мають певну організацію, але не відповідають суворій схемі. До таких даних можна віднести, наприклад, XML або JSON-документи. Неструктуровані дані включають текстові документи, зображення, відео або інші види інформації, що не мають чітко визначеної структури.

Зростання обсягів неструктурованих даних створює додаткові виклики для сучасних інформаційних систем. Традиційні методи обробки даних не завжди дозволяють ефективно працювати з такими типами інформації, що зумовлює необхідність використання спеціалізованих підходів до представлення та аналізу знань.

Одним із напрямів вирішення цієї проблеми є використання технологій Semantic Web, які дозволяють формалізувати знання та представляти їх у вигляді семантичних графів. Такі підходи дають

можливість не лише зберігати інформацію, але й виконувати логічне виведення нових фактів на основі вже наявних знань [2], [3].

Таким чином, дані, інформація та знання утворюють взаємопов’язану систему, що лежить в основі функціонування інтелектуальних інформаційних систем. Дані виступають початковим джерелом інформації, інформація формується в результаті їх обробки, а знання виникають у процесі аналізу та узагальнення отриманої інформації.

Ефективне представлення та використання знань є одним із ключових завдань сучасних інформаційних технологій. Саме тому розробка методів формалізації знань та механізмів їх автоматичного виведення є важливим напрямом досліджень у галузі штучного інтелекту. Подальший розгляд технологій представлення знань я продемонструю у наступному підрозділі, де проаналізую основні інструменти Semantic Web, зокрема RDF, OWL та SPARQL.

1.2 Основи Semantic Web і технології для представлення знань

Semantic Web є розвитком традиційної архітектури Всесвітньої павутини, основною метою якого є забезпечення машинозрозумілого представлення інформації та знань. На відміну від класичного вебу, де дані орієнтовані переважно на сприйняття людиною, Semantic Web дозволяє комп’ютерним системам не лише зберігати інформацію, але й інтерпретувати її, встановлювати зв’язки між об’єктами та виконувати логічні висновки.

Основою Semantic Web є формальні моделі представлення знань, які забезпечують структурований опис об’єктів, їх властивостей та взаємозв’язків. Такі моделі дозволяють інтегрувати інформацію з різних джерел і створювати інтелектуальні інформаційні системи. Однією з базових технологій Semantic Web є RDF – стандартна модель даних для опису ресурсів у вигляді трійок «суб’єкт – предикат – об’єкт». У такій

моделі знання представляються у формі орієнтованого графа, де вузли відповідають об’єктам або ресурсам, а ребра відображають відношення між ними.

Для більш детального та формалізованого опису предметної галузі використовується OWL – мова створення онтологій, яка дозволяє визначати класи, підкласи, властивості об’єктів та логічні обмеження між ними. OWL забезпечує більш високий рівень формалізації знань і підтримує механізми автоматичного логічного виведення.

Для роботи з даними, представленими у вигляді RDF-графів, використовується SPARQL – мова запитів, що дозволяє здійснювати пошук, фільтрацію та обробку семантичних даних. За принципом роботи SPARQL часто порівнюють із мовою SQL, проте вона орієнтована на роботу з графовими структурами знань.

Таким чином, поєднання технологій RDF, OWL та SPARQL створює основу для побудови інтелектуальних веб-систем, здатних ефективно інтегрувати, аналізувати та обробляти знання з різних джерел. Важливу роль у таких системах відіграють онтології, які забезпечують формальний опис предметної галузі та структуру знань.

1.3 Онтології: структура та типи знань

Онтології є фундаментальним компонентом Semantic Web, оскільки забезпечують формальний опис предметної галузі та структуру знань, що дозволяє комп’ютерним системам не лише зберігати інформацію, а й виконувати логічні висновки. В онтологіях визначаються класи, підкласи, властивості об’єктів і відносини між ними, а також логічні обмеження, що формалізують структуру предметної галузі.

Структура онтології традиційно поділяється на дві взаємопов’язані частини. Перша, термінологічна складова, або TBox, описує концептуальні

елементи предметної галузі, включаючи класи, їх підкласи та властивості, а також визначає, які взаємозв’язки між поняттями є логічно допустимими. Друга складова, ABox, містить фактичні дані про конкретні об’єкти та їхні атрибути, що дозволяє виконувати запити та робити висновки на основі реальних даних.

У контексті систем персоналізованого навчання TBox може описувати класи курсів, рівні складності, типи завдань і критерії успішності, а також відношення між курсами, наприклад, залежності «попередній курс – наступний курс». ABox у цьому випадку містить конкретні дані про студентів, їхні оцінки, пройдені курси та рівень знань у різних предметних областях. Поєднання TBox і ABox дозволяє створювати динамічні моделі знань, які використовуються для формування персоналізованих рекомендацій, автоматичного оцінювання прогресу студента та планування подальших курсів.

Правильне проектування онтологій у таких системах забезпечує ефективне виконання запитів до бази знань і створює основу для автоматичного виведення нових знань. Це дозволяє не лише зберігати інформацію про навчальний процес, а й використовувати її для побудови інтелектуальних рекомендацій, адаптації навчального матеріалу під індивідуальні потреби студента та інтеграції з нейросимволічними підходами, де великі мовні моделі допомагають інтерпретувати запити та генерувати пояснення.

Таким чином, онтології в системах персоналізованого навчання виконують ключову роль у поєднанні структурованого представлення знань із механізмами логічного виведення, створюючи фундамент для подальшої інтеграції з гібридними методами обробки знань та стають невід’ємною частиною для розробки гібридних систем [3], [4].

1.4 Логічне виведення та Rule-based reasoning

Важливим елементом систем представлення знань у Semantic Web є механізми логічного виведення, або reasoning. Вони дозволяють отримувати нові знання на основі вже наявних фактів і правил, що значно розширює можливості інтелектуальних інформаційних систем. Завдяки використанню логічного виведення система може не лише зберігати інформацію, але й аналізувати її, формувати висновки та автоматично приймати рішення у межах заданої предметної галузі [7].

У середовищі Semantic Web логічне виведення ґрунтується на використанні формальних правил і онтологічних моделей. Такі правила дозволяють встановлювати нові взаємозв’язки між об’єктами або виявляти приховані закономірності у даних. Одним із поширених підходів є rule-based reasoning, при якому нові факти виводяться на основі заданих логічних правил. У таких системах правила можуть бути представлені у вигляді конструкцій типу «якщо–то», що визначають умови формування нових тверджень.

У контексті систем персоналізованого навчання логічне виведення може використовуватися для формування рекомендацій щодо навчальної траєкторії студента. Наприклад, якщо у базі знань зазначено, що певний курс має передумову у вигляді іншого курсу, система може автоматично визначити, чи може студент перейти до наступного етапу навчання. Якщо студент ще не завершив необхідний попередній курс або має недостатній рівень знань, система може запропонувати додаткові матеріали або альтернативний навчальний шлях.

Окрім правилового виведення, важливу роль відіграє перевірка логічної узгодженості знань. Вона дозволяє виявляти суперечності у базі знань, що особливо важливо у складних інформаційних системах. У системах персоналізованого навчання така перевірка може забезпечуватикоректність структури навчальних програм, наприклад

запобігати ситуаціям, коли курс одночасно вимагає і не вимагає певної передумови.

Застосування механізмів reasoning у поєднанні з онтологіями та графами знань дозволяє створювати інтелектуальні системи, здатні автоматично аналізувати інформацію та формувати нові знання. У сучасних дослідженнях ці підходи дедалі частіше поєднуються з нейронними методами, зокрема великими мовними моделями, що дозволяє доповнити формальні механізми виведення можливостями обробки природної мови та інтерпретації запитів користувачів. Така інтеграція створює основу для побудови гібридних систем обробки знань, які поєднують точність символічних моделей із гнучкістю нейронних підходів.

1.5 Графи знань як модель представлення знань

1.5.1 Структура Knowledge Graph

Граф знань (Knowledge Graph) є однією з сучасних моделей представлення знань, що використовується в інтелектуальних інформаційних системах. Основною особливістю такої моделі є представлення інформації у вигляді графової структури, де знання описуються через сутності та зв’язки між ними. Такий підхід дозволяє формалізувати предметну область та наочно відобразити взаємозв’язки між різними об’єктами [4], [14].

Базовими елементами графа знань є вузли (nodes), зв’язки (edges) та властивості (properties). Вузли представляють об’єкти або сутності предметної галузі, наприклад навчальні курси, теми, навички або користувачів системи. Зв’язки відображають відношення між цими сутностями, наприклад залежність між навчальними темами або належність певної теми до конкретного курсу. Властивості

використовуються для опису характеристик сутностей або зв’язків, наприклад рівня складності курсу, тривалості навчання або необхідних попередніх знань.

У більшості реалізацій графи знань базуються на моделі трійок «суб’єкт – предикат – об’єкт», яка використовується у технології RDF. У цій моделі суб’єкт представляє певну сутність, предикат описує тип відношення, а об’єкт є іншою сутністю або значенням властивості. Такий підхід дозволяє представляти знання у вигляді орієнтованого графа, де кожна трійка формує окремий зв’язок між двома об’єктами.

Наприклад, у системі персоналізованого навчання граф знань може містити такі трійки:

– «Курс машинного навчання» – вимагає знання – «Лінійна алгебра»;

– «Студент» – вивчає – «Курс машинного навчання»;

– «Курс машинного навчання» – містить тему – «Нейронні мережі».

У результаті формується мережа взаємопов’язаних знань, що відображає структуру предметної галузі. Завдяки цьому система може аналізувати зв’язки між різними навчальними матеріалами та визначати оптимальні навчальні траєкторії для студентів.

Ще однією важливою характеристикою графів знань є можливість їх поступового розширення. Нові сутності та зв’язки можуть додаватися без необхідності змінювати всю структуру моделі. Це робить графи знань особливо ефективними для роботи з великими та динамічними наборами даних, де структура знань може постійно змінюватися або доповнюватися.

Таким чином, структура графа знань забезпечує гнучкий і наочний спосіб представлення знань, який дозволяє моделювати складні взаємозв’язки між об’єктами предметної галузі. Завдяки використанню графової моделі та семантичних зв’язків такі системи можуть ефективно інтегрувати інформацію з різних джерел і створювати основу для подальшого логічного аналізу та автоматичного виведення нових знань.

1.5.2 Відмінність графів знань від традиційних баз даних

Традиційні бази даних, зокрема реляційні, зберігають інформацію у вигляді таблиць із чітко визначеною структурою [4]. Кожна таблиця містить записи, що складаються з набору полів, а зв’язки між даними реалізуються через ключі та таблиці зв’язків. Такий підхід є ефективним для зберігання структурованих даних, однак він має певні обмеження при роботі зі складними мережами взаємопов’язаних знань.

На відміну від реляційних баз даних, графи знань використовують графову модель представлення даних. У такій моделі інформація представлена у вигляді вузлів і зв’язків між ними, що дозволяє природно відображати складні семантичні відношення між об’єктами. Завдяки цьому графи знань краще підходять для задач, де важливу роль відіграє аналіз взаємозв’язків між даними.

Ще однією важливою відмінністю є гнучкість структури даних. У реляційних базах даних зміна структури таблиць часто потребує модифікації всієї схеми бази. У графах знань нові типи зв’язків або сутностей можуть додаватися значно простіше, без істотної перебудови всієї моделі.

У системах персоналізованого навчання така гнучкість є особливо важливою, оскільки структура навчальних курсів, тем та навичок може постійно змінюватися. Використання графів знань дозволяє ефективно відображати залежності між навчальними матеріалами та швидко адаптувати систему до появи нових курсів або тем.

Таким чином, графи знань забезпечують більш природну модель представлення складних семантичних взаємозв’язків і мають більшу гнучкість порівняно з традиційними базами даних, що робить їх ефективним інструментом для побудови інтелектуальних інформаційних систем.

1.5.3 Rule-based reasoning над графами знань

Однією з ключових переваг використання графів знань є можливість виконання логічного виведення на основі правил. Rule-based reasoning дозволяє системі формувати нові знання на основі вже наявної інформації та визначених логічних правил [6], [14].

У такому підході знання, представлені у графі, можуть аналізуватися за допомогою спеціальних правил, які описують умови формування нових фактів. Наприклад, правило може визначати, що якщо студент успішно завершив певний курс і володіє необхідними базовими знаннями, то система може рекомендувати йому наступний, більш складний курс.

У системах, що використовують технології Semantic Web, такі правила часто реалізуються за допомогою спеціальних мов опису правил або механізмів логічного виведення. Вони дозволяють перевіряти

консистентність знань, знаходити нові зв’язки між об’єктами та формувати додаткові висновки на основі наявних даних.

У контексті персоналізованого навчання rule-based reasoning може використовуватися для автоматичного аналізу навчальних результатів студентів і формування рекомендацій щодо подальшого навчання. Наприклад, система може визначити, які курси студент повинен пройти перед тим, як перейти до складніших тем, або запропонувати додаткові матеріали у разі недостатнього рівня підготовки.

Таким чином, використання rule-based reasoning у графах знань дозволяє перетворити статичну базу даних у динамічну систему знань, здатну аналізувати інформацію та формувати нові висновки. Це є важливою складовою інтелектуальних систем, що працюють із великими обсягами структурованих знань.

1.6 Обмеження символічного підходу

Технології Semantic Web відіграють важливу роль у сучасних інтелектуальних інформаційних системах, оскільки забезпечують формальне представлення знань і створюють умови для автоматичного логічного виведення нових фактів. Використання онтологій, графів знань та механізмів reasoning дозволяє системам працювати зі структурованими знаннями та виконувати складний аналіз інформації. Завдяки цьому Semantic Web широко застосовується у різних галузях, де необхідна інтеграція знань із різних джерел та підтримка прийняття рішень [2], [3].

Однією з основних переваг технологій Semantic Web є стандартизація представлення даних. Використання таких стандартів, як RDF, OWL та SPARQL, дозволяє інтегрувати інформацію з різних інформаційних систем та забезпечувати її узгоджене використання. Це особливо важливо у складних системах, де дані можуть надходити з різних джерел і мати різну структуру.

Ще однією важливою перевагою є можливість виконання логічного виведення. Завдяки використанню онтологій і формальних правил система може автоматично формувати нові знання на основі вже наявних фактів. У системах персоналізованого навчання це дозволяє аналізувати навчальний прогрес студента, визначати його поточний рівень підготовки та формувати рекомендації щодо подальших курсів або навчальних матеріалів. Таким чином, система може не лише зберігати інформацію про навчальний процес, а й активно використовувати її для підтримки індивідуальної освітньої траєкторії.

Водночас використання технологій Semantic Web має і певні обмеження. Однією з основних проблем є складність створення та підтримки онтологій. Процес формалізації предметної галузі потребує значних зусиль і участі експертів, оскільки необхідно правильно визначити структуру знань, взаємозв’язки між об’єктами та логічні обмеження.

У великих системах це може вимагати значних ресурсів і часу. Іншим обмеженням є обчислювальна складність логічного виведення. У випадку великих графів знань процес reasoning може бути досить ресурсоємним, що ускладнює використання таких систем у режимі реального часу. Крім того, традиційні символічні моделі мають обмежену здатність працювати з неструктурованими даними, зокрема з природною мовою.

У системах персоналізованого навчання ця проблема проявляється у складності інтерпретації текстових запитів користувачів, наприклад коли студент формулює питання у довільній формі або описує власні навчальні потреби природною мовою. Традиційні онтологічні системи не завжди здатні ефективно обробляти такі запити без додаткових механізмів аналізу тексту.

З огляду на ці обмеження сучасні дослідження все частіше зосереджуються на поєднанні символічних методів представлення знань із нейронними підходами, зокрема з використанням великих мовних моделей [9], [13]. Такі моделі можуть ефективно працювати з неструктурованими текстовими даними, інтерпретувати запити користувачів і генерувати пояснення результатів аналізу. Поєднання можливостей Semantic Web і нейронних моделей створює основу для побудови гібридних систем обробки знань, які поєднують точність формальних логічних моделей із гнучкістю методів машинного навчання.

Таким чином, символічні системи є точними, але негнучкими і погано працюють із природною мовою, що відкриває інтерес до нейронних моделей та гібридних підходів.

2 ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ У РОБОТІ ЗІ ЗНАННЯМИ

2.1 Архітектура великих мовних моделей

2.1.1 Поняття великих мовних моделей

Великі мовні моделі (Large Language Models, LLM) є одним із найважливіших досягнень сучасного штучного інтелекту у сфері обробки природної мови. Вони представляють собою нейронні мережі великого масштабу, які навчаються на значних обсягах текстових даних та здатні виконувати різноманітні мовні завдання. До таких завдань належать аналіз тексту, автоматичний переклад, відповіді на запитання, узагальнення інформації та генерація зв’язних текстів [10], [11].

Основною метою великих мовних моделей є формування статистичного представлення мови на основі великої кількості прикладів. У процесі навчання модель аналізує текстові послідовності та виявляє закономірності між словами, фразами та контекстом їх використання. Завдяки цьому вона може прогнозувати наступні елементи тексту та формувати змістовні відповіді на запити користувачів.

Великі мовні моделі базуються на глибоких нейронних мережах, що містять мільйони або навіть мільярди параметрів. Кожен параметр моделі відображає певну закономірність або зв’язок між елементами мови. Чим більша кількість параметрів і навчальних даних використовується, тим точніше модель може відображати складні структури природної мови.

У сучасних інформаційних системах великі мовні моделі використовуються для обробки неструктурованих текстових даних. Вони здатні інтерпретувати запити користувачів, аналізувати текстові документи та формувати відповіді у зрозумілій для людини формі. Це робить їх важливим інструментом у системах пошуку інформації, інтелектуальних

помічниках, системах рекомендацій та освітніх платформах.

У контексті систем персоналізованого навчання великі мовні моделі можуть використовуватися для аналізу запитів студентів, пояснення навчальних матеріалів і формування рекомендацій щодо подальшого навчання. Наприклад, студент може сформулювати питання у довільній текстовій формі, а система на основі мовної моделі здатна інтерпретувати цей запит та надати відповідну інформацію або пояснення.

Таким чином, великі мовні моделі є потужним інструментом для роботи з текстовою інформацією. Вони дозволяють автоматизувати процеси аналізу природної мови та забезпечують нові можливості для створення інтелектуальних інформаційних систем.

2.1.2 Архітектура Transformer та механізм self-attention

Більшість сучасних великих мовних моделей побудовані на основі архітектури Transformer, яка була запропонована у 2017 році [11], [16]. Основною особливістю цієї архітектури є використання механізму self-attention, що дозволяє моделі ефективно аналізувати взаємозв’язки між словами у тексті.

На відміну від попередніх підходів до обробки тексту, таких як рекурентні нейронні мережі або згорткові моделі, архітектура Transformer не обробляє текст послідовно. Замість цього вона аналізує всі елементи тексту одночасно, що значно підвищує ефективність навчання та дозволяє працювати з довшими текстовими послідовностями.

Механізм self-attention є ключовим компонентом цієї архітектури. Його основна ідея полягає у тому, що кожне слово у тексті може враховувати інформацію про інші слова в межах тієї ж послідовності. Завдяки цьому модель здатна визначати, які слова є найбільш важливими для розуміння контексту.

У процесі роботи механізму self-attention кожному слову тексту призначається певний набір числових представлень, які використовуються для оцінки взаємозв’язків із іншими словами. На основі цих значень модель визначає вагу впливу кожного слова на формування кінцевого представлення тексту. Таким чином, система може зосереджувати увагу на найбільш значущих елементах контексту.

Архітектура Transformer складається з декількох послідовних шарів, кожен з яких виконує операції self-attention та подальшої обробки отриманих представлень. Завдяки великій кількості таких шарів модель поступово формує складні абстрактні представлення тексту.

Використання архітектури Transformer дозволило значно підвищити якість систем обробки природної мови. Сучасні мовні моделі здатні ефективно працювати з великими обсягами текстових даних та враховувати складні контекстуальні залежності між словами.

2.1.3 Процес навчання та обробки тексту

Навчання великих мовних моделей здійснюється на основі великих корпусів текстових даних, які можуть включати книги, наукові статті, веб-сторінки та інші текстові ресурси [10], [11]. У процесі навчання модель аналізує текстові послідовності та поступово налаштовує свої параметри таким чином, щоб максимально точно прогнозувати наступні елементи тексту.

Одним із поширених підходів до навчання мовних моделей є завдання прогнозування наступного слова або токена у текстовій послідовності. Модель отримує фрагмент тексту та намагається передбачити, який елемент має з’явитися далі. Якщо прогноз виявляється неправильним, параметри моделі коригуються таким чином, щоб зменшити помилку.

У результаті багаторазового повторення цього процесу модель поступово навчається відображати складні закономірності природної мови. Вона починає розуміти типові структури речень, взаємозв’язки між словами та різні стилі тексту.

Після завершення етапу навчання мовна модель може використовуватися для виконання різних завдань обробки тексту. Коли користувач вводить запит, система перетворює текст у числове представлення та передає його до нейронної мережі. Модель аналізує контекст запиту та генерує відповідь, формуючи послідовність слів, які найбільш ймовірно відповідають заданому контексту.

У системах персоналізованого навчання такий підхід дозволяє створювати інтелектуальні інтерфейси взаємодії з користувачем. Студенти можуть ставити запитання у природній мовній формі, отримувати пояснення складних тем або рекомендації щодо навчальних матеріалів.

Таким чином, великі мовні моделі поєднують сучасні нейронні архітектури, великі обсяги навчальних даних та ефективні алгоритми оптимізації, що дозволяє їм виконувати складні завдання аналізу та генерації текстової інформації.

2.2 Неявне знання у великих мовних моделях

2.2.1 Поняття неявного знання у великих мовних моделях

Однією з важливих особливостей великих мовних моделей є їх здатність зберігати та використовувати значні обсяги інформації, отриманої під час навчання. Часто така інформація не представлена у вигляді явних фактів або структурованих баз даних, а існує у прихованій формі всередині параметрів нейронної мережі. Саме тому у науковій літературі використовується поняття неявного знання (implicit knowledge) [12], [15].

Неявне знання у великих мовних моделях означає інформацію, яку модель засвоює під час навчання на великих текстових корпусах, але яка не представлена у вигляді окремих записів або правил. На відміну від традиційних баз знань або графів знань, де факти зберігаються у структурованій формі, мовні моделі формують статистичне представлення інформації на основі закономірностей, що зустрічаються у текстах.

У процесі навчання модель аналізує мільйони або навіть мільярди текстових прикладів. Завдяки цьому вона виявляє різні мовні закономірності, факти про світ, взаємозв’язки між поняттями та типові способи формулювання інформації. У результаті формується складна система внутрішніх представлень, яка дозволяє моделі відтворювати або використовувати ці знання під час генерації тексту.

Важливою особливістю неявного знання є те, що воно не зберігається у вигляді окремих фактів, які можна безпосередньо прочитати або витягти з моделі. Натомість знання розподіляється між великою кількістю параметрів нейронної мережі. Саме тому іноді говорять про розподілене представлення знань (distributed representation) [12].

У контексті інтелектуальних інформаційних систем така властивість мовних моделей відкриває нові можливості для роботи з текстовими даними. Модель може використовувати знання, отримані під час навчання, для інтерпретації запитів користувачів, пояснення складних понять або генерації нової інформації на основі наявного контексту.

Таким чином, неявне знання є важливою характеристикою великих мовних моделей і відрізняє їх від традиційних систем представлення знань, які базуються на явних правилах або структурованих базах даних.

2.2.2 Представлення знань у параметрах моделі

У великих мовних моделях знання не зберігаються у вигляді окремих записів або таблиць [12], [15]. Натомість вони закодовані у параметрах

нейронної мережі, які формуються під час процесу навчання. Параметри моделі представляють собою числові значення, що визначають поведінку нейронної мережі при обробці текстових даних.

У процесі навчання модель отримує текстові послідовності та намагається прогнозувати наступні слова або токени у тексті. Коли прогноз виявляється неточним, параметри моделі коригуються таким чином, щоб зменшити помилку. Поступово, у результаті великої кількості ітерацій, модель навчається відображати статистичні закономірності природної мови.

Саме у цих параметрах і формується розподілене представлення знань. Наприклад, інформація про взаємозв’язки між поняттями, типові контексти використання слів або факти про різні об’єкти може бути закодована у вагових коефіцієнтах нейронної мережі. Таке представлення не є явним і не може бути безпосередньо інтерпретоване людиною, однак воно дозволяє моделі використовувати ці знання при формуванні відповідей.

Однією з особливостей такого підходу є те, що знання розподіляється між великою кількістю параметрів моделі. Кожен окремий параметр не відповідає за конкретний факт або правило. Натомість різні параметри разом формують складну систему взаємозв’язків, що відображає структуру знань, отриманих під час навчання.

У результаті модель може використовувати ці внутрішні представлення для виконання різних мовних завдань. Наприклад, вона може визначати значення слів у контексті, встановлювати зв’язки між поняттями або генерувати пояснення певних явищ. Таким чином, знання, що зберігається у параметрах моделі, стає основою для її здатності працювати з природною мовою.

2.2.3 Приклади знань, що зберігаються у мовних моделях

Під час навчання на великих текстових корпусах мовні моделі засвоюють значну кількість фактів і закономірностей, які можуть бути використані під час генерації тексту. Ці знання можуть стосуватися різних сфер, включаючи загальні відомості про світ, наукові поняття, історичні події або взаємозв’язки між різними об’єктами.

Наприклад, мовна модель може знати, що певні поняття належать до однієї предметної галузі або що між ними існують певні логічні зв’язки. Такі знання формуються на основі статистичних закономірностей у текстах, на яких модель була навчена. Якщо у навчальних даних часто зустрічаються твердження про взаємозв’язок між певними поняттями, модель поступово засвоює ці зв’язки.

Крім того, мовні моделі можуть зберігати знання про типові структури текстів, способи пояснення різних явищ або формулювання наукових тверджень. Це дозволяє їм генерувати зв’язні та логічно структуровані відповіді на запити користувачів.

У системах персоналізованого навчання така властивість може бути використана для пояснення навчальних матеріалів або формування відповідей на питання студентів. Наприклад, студент може поставити запитання щодо певної теми, а мовна модель, використовуючи знання, отримані під час навчання, здатна сформувати пояснення або опис відповідного поняття.

Водночас важливо зазначити, що знання, збережені у мовних моделях, мають статистичний характер. Це означає, що модель формує відповіді на основі ймовірнісних закономірностей, а не на основі чітко визначених логічних правил. Саме ця особливість відрізняє мовні моделі від традиційних систем представлення знань і водночас може призводити до виникнення помилок або неточностей у згенерованій інформації.

Таким чином, великі мовні моделі здатні зберігати значні обсяги неявних знань, отриманих під час навчання на текстових даних. Ці знання дозволяють моделі інтерпретувати запити користувачів, пояснювати різні поняття та генерувати змістовні відповіді, що робить їх важливим інструментом сучасних інтелектуальних інформаційних систем.

2.3 Великі мовні моделі у задачах логічного міркування

2.3.1 Поняття логічного міркування у штучному інтелекті

Логічне міркування (reasoning) є однією з ключових задач у сфері штучного інтелекту [6], [9]. Під цим поняттям розуміють процес отримання нових знань на основі вже наявної інформації шляхом застосування певних правил або закономірностей. У традиційних системах штучного інтелекту логічне міркування зазвичай реалізується за допомогою формальних методів, таких як логічні правила, онтології або системи продукційних правил.

У символічному підході знання подаються у формалізованому вигляді, наприклад у формі логічних тверджень або триплетів у графах знань. На основі цих структур система може виконувати логічне виведення, застосовуючи правила дедукції або інші механізми формального міркування. Такий підхід забезпечує високу точність та можливість пояснення отриманих результатів, однак часто вимагає складного процесу побудови та підтримки бази знань.

На відміну від символічних систем, сучасні нейронні моделі, зокрема великі мовні моделі, реалізують інший підхід до міркування. Вони не використовують явні логічні правила, а виконують аналіз текстових даних на основі статистичних закономірностей, засвоєних під час навчання. У результаті модель може формувати відповіді, які виглядають логічними та

узгодженими, навіть якщо процес формування відповіді не базується на класичних методах формального виведення.

Таким чином, у сучасних системах штучного інтелекту можна виділити два основних підходи до логічного міркування: символічний, який базується на формальних правилах і структурах знань, та нейронний, що використовує статистичні моделі і великі обсяги навчальних даних. Кожен із цих підходів має свої переваги та обмеження, що зумовлює активний розвиток гібридних методів, які поєднують обидві парадигми.

2.3.2 Можливості великих мовних моделей у задачах міркування

Завдяки навчанню на великих текстових корпусах великі мовні моделі здатні виконувати широкий спектр завдань, пов’язаних з аналізом інформації та побудовою логічних висновків. Хоча такі моделі не використовують формальні правила логіки, вони можуть встановлювати зв’язки між різними поняттями, інтерпретувати складні запити користувачів і формувати змістовні відповіді.

Однією з важливих можливостей мовних моделей є здатність аналізувати контекст запиту. Модель враховує не лише окремі слова, але й взаємозв’язки між ними, що дозволяє правильно інтерпретувати зміст тексту. Завдяки цьому великі мовні моделі можуть відповідати на складні запитання, виконувати узагальнення інформації або пояснювати певні явища [10], [13].

Наприклад, у системах персоналізованого навчання мовна модель може аналізувати запит студента та надавати відповідні пояснення або рекомендації. Якщо студент ставить питання щодо певної теми або просить пояснити складне поняття, модель здатна сформувати відповідь на основі знань, отриманих під час навчання. У деяких випадках вона може також поєднувати інформацію з різних джерел та формувати більш повні пояснення.

Крім того, великі мовні моделі можуть виконувати завдання, пов’язані з аналізом логічних залежностей між різними твердженнями. Наприклад, модель може визначити, чи випливає певне твердження з наданої інформації, або встановити можливі причини певного явища. Хоча такі операції не є формальним логічним виведенням у класичному розумінні, вони демонструють здатність мовних моделей до складних когнітивних операцій.

Разом із тим важливо враховувати, що результати міркування мовних моделей мають ймовірнісний характер. Це означає, що модель формує відповіді на основі статистичних закономірностей, а не суворих логічних правил. Саме тому у деяких випадках отримані результати можуть бути неточними або суперечливими.

2.3.3 Методи покращення міркування у великих мовних моделях

У сучасних дослідженнях значна увага приділяється методам покращення здатності великих мовних моделей до логічного міркування. Одним із найбільш поширених підходів є використання спеціальних стратегій формулювання запитів, які допомагають моделі більш послідовно аналізувати інформацію.

Однією з таких стратегій є підхід, відомий як поетапне міркування. У цьому випадку модель стимулюється формувати відповідь не одразу, а через послідовність проміжних кроків. Такий підхід дозволяє моделі більш чітко структурувати процес аналізу інформації та формувати більш обґрунтовані висновки.

Іншим важливим методом є використання додаткового контексту. Якщо модель отримує більше інформації про предметну область або конкретну задачу, вона може формувати більш точні та узгоджені відповіді. У практичних системах це може реалізовуватися шляхом

підключення зовнішніх джерел знань, наприклад баз даних або графів знань.

Особливо перспективним напрямом є поєднання великих мовних моделей із структурованими системами представлення знань. У такому підході мовна модель використовується для інтерпретації запитів користувача та формування пояснень, тоді як формальне логічне виведення виконується за графами знань або правил логіки. Це дозволяє поєднати гнучкість нейронних моделей символічних систем [6], [13].

У системах персоналізованого навчання такий гібридний підхід може використовуватися для побудови інтелектуальних навчальних платформ. Наприклад, мовна модель може інтерпретувати запит студента та перетворювати його у формальний запит до графа знань, після чого результати логічного виведення можуть бути представлені у вигляді зрозумілого пояснення.

Таким чином, великі мовні моделі демонструють значний потенціал у задачах логічного міркування, однак їх ефективність значно підвищується при інтеграції з формальними системами представлення знань. Саме ця ідея лежить в основі сучасних досліджень у сфері гібридних інтелектуальних систем.

2.4 Галюцинації у великих мовних моделях

2.4.1 Поняття галюцинацій у мовних моделях

Однією з відомих проблем сучасних великих мовних моделей є явище, яке в науковій літературі називають галюцинаціями (hallucinations). Під цим терміном розуміють ситуації, коли модель генерує інформацію, що виглядає логічною та правдоподібною, але фактично є неправильною або не відповідає дійсності [10], [15].

Галюцинації можуть проявлятися у різних формах. Наприклад, модель може створювати неіснуючі факти, вигадані посилання на джерела, неправильні дати або неточні твердження щодо певних подій чи об’єктів. Особливість таких помилок полягає в тому, що сформульована відповідь часто виглядає переконливо і має граматично правильну структуру, що може ускладнювати виявлення помилки [18].

Причина цього явища пов’язана з принципом роботи мовних моделей. Основною задачею таких моделей є прогнозування наступного слова або токена у текстовій послідовності. Модель формує відповідь, обираючи найбільш імовірні варіанти слів на основі статистичних закономірностей, засвоєних під час навчання. У результаті модель може генерувати текст, який виглядає логічно узгодженим, але не обов’язково відповідає фактичній інформації.

Важливо відрізняти галюцинації від звичайних помилок у міркуванні. Якщо помилка міркування пов’язана з неправильним логічним висновком на основі наданих даних, то галюцинація виникає у випадках, коли модель створює нову інформацію, яка не базується на достовірних джерелах. У деяких ситуаціях модель може навіть формувати детальні пояснення або аргументацію для інформації, яка фактично є вигаданою.

Таким чином, галюцинації є характерною особливістю великих мовних моделей і пов’язані з їх статистичною природою. Незважаючи на високий рівень розвитку таких моделей, проблема генерації неточної інформації залишається одним із ключових викликів у сфері їх практичного використання.

2.4.2 Причини виникнення галюцинацій

Існує кілька основних факторів, які спричиняють виникнення галюцинацій у великих мовних моделях. Перш за все, це пов’язано з тим, що такі моделі будуються на основі статистичних методів обробки тексту.

Під час навчання модель виявляє закономірності між словами та фразами у великих текстових корпусах, однак вона не має прямого механізму перевірки достовірності інформації [18].

Однією з причин виникнення галюцинацій є відсутність доступу до структурованих джерел знань під час генерації відповіді. Модель формує текст, спираючись лише на внутрішні параметри, які містять узагальнені статистичні закономірності. Якщо інформація у навчальних даних була неповною або суперечливою, модель може відтворювати неточні твердження.

Іншою важливою причиною є неоднозначність запитів користувачів. Якщо запит сформульований нечітко або містить недостатньо контексту, модель може інтерпретувати його різними способами. У таких випадках вона генерує відповідь, яка здається найбільш імовірною з точки зору статистики текстів, але не обов’язково є правильною [15].

Також галюцинації можуть виникати у випадках, коли модель намагається заповнити відсутню інформацію. Якщо у процесі генерації відповіді модель стикається з браком даних, вона може створювати нові твердження на основі загальних мовних шаблонів. У результаті формується текст, який виглядає змістовним, але не має фактичного підтвердження.

Крім того, на появу галюцинацій впливає складність запиту. Чим складнішим є питання або задача, тим вищою є ймовірність того, що модель сформує неточний або суперечливий висновок. Це особливо актуально для задач, пов’язаних з логічним міркуванням або аналізом складних взаємозв’язків між різними поняттями.

Додатково на появу галюцинацій впливає відсутність механізмів перевірки фактів у моделі. Тому перспективним напрямом є поєднання мовних моделей із зовнішніми джерелами знань для підвищення достовірності відповідей.

2.4.3 Вплив галюцинацій на інтелектуальні системи

Проблема галюцинацій має суттєвий вплив на використання великих мовних моделей у практичних інтелектуальних системах. У багатьох сферах, таких як освіта, медицина або системи підтримки прийняття рішень, точність інформації є критично важливою. Наявність навіть незначних помилок може призводити до неправильних висновків або рекомендацій [17].

Наприклад, у системах персоналізованого навчання мовна модель може використовуватися для пояснення навчальних матеріалів або відповіді на запитання студентів. Якщо модель генерує неточну інформацію, це може призвести до формування неправильного розуміння навчальної теми. У результаті користувач отримує відповідь, яка виглядає переконливо, але фактично містить помилки.

Ще однією проблемою є складність автоматичного виявлення галюцинацій. Оскільки згенерований текст зазвичай має граматично правильну та логічно послідовну структуру, користувач може не одразу помітити неточність. Це створює ризик використання недостовірної інформації у різних інформаційних системах.

З огляду на ці обмеження, у сучасних дослідженнях активно розглядаються методи зменшення впливу галюцинацій. Одним із перспективних підходів є використання структурованих джерел знань, таких як графи знань або бази знань. У такому випадку мовна модель може використовуватися для інтерпретації запитів користувача та формування текстових пояснень, тоді як перевірка фактів і логічне виведення здійснюються за допомогою формальних методів.

Таким чином, галюцинації залишаються важливим викликом для використання великих мовних моделей у складних інтелектуальних системах. Водночас поєднання мовних моделей із структурованими

системами представлення знань відкриває можливості для створення більш надійних та пояснюваних інтелектуальних рішень.

2.5 Основні відмінності графів знань та великих мовних моделей

Графи знань і великі мовні моделі представляють різні підходи до представлення та використання знань у системах штучного інтелекту. У графах знань інформація зберігається структуровано у вигляді триплетів (суб’єкт–предикат–об’єкт), що дозволяє здійснювати точне логічне виведення та прозоро відстежувати джерела знань. Великі мовні моделі, навпаки, зберігають знання неявно в параметрах нейронної мережі, що забезпечує гнучкість у роботі з природною мовою, але не гарантує формальної точності результатів і пояснюваності [4], [12], [13].

Ще однією важливою відмінністю є механізм отримання нових знань. Графи знань оперують формальними правилами та логічними висновками, тоді як великі мовні моделі генерують відповіді статистично на основі контексту навчальних даних. Через це вони можуть давати непередбачувані або неточні результати, відомі як галюцинації, але водночас здатні формувати розгорнуті текстові пояснення.

Ураховуючи ці характеристики, оптимальним підходом є поєднання обох технологій: точності та структурованості графів знань із гнучкістю та здатністю до природномовного висновку великих мовних моделей. Тому для подальшого дослідження буде використано інтеграцію обох технологій з метою порівняльного аналізу їхніх можливостей у виведенні знань.

3 МОДЕЛЬ ГІБРИДНОГО ВИВЕДЕННЯ ЗНАНЬ

3.1 Мотивація використання гібридного підходу

У попередньому розділі було проаналізовано особливості використання графів знань та великих мовних моделей у задачах представлення та виведення знань. Проведений аналіз показав, що ці підходи використовують різні принципи роботи з інформацією та мають власні переваги й обмеження при застосуванні в інтелектуальних системах.

Графи знань забезпечують формальне та структуроване представлення інформації, що дозволяє виконувати точні запити до баз знань і застосовувати логічні методи виведення. Великі мовні моделі, у свою чергу, демонструють високу ефективність у роботі з природною мовою, інтерпретації текстових запитів та генерації змістовних відповідей. Водночас використання кожного з цих підходів окремо має певні обмеження, що ускладнює їх застосування як універсального інструменту для роботи зі знаннями.

У зв’язку з цим у сучасних дослідженнях активно розвивається напрям neuro-symbolic систем, що поєднують символічні методи представлення знань із нейронними моделями обробки інформації. Основна ідея такого підходу полягає у використанні структурованих джерел знань для забезпечення точності та логічної узгодженості результатів, а також застосуванні нейронних моделей для обробки природної мови та взаємодії з користувачем.

У межах гібридних систем графи знань можуть використовуватися як формальна база знань, що містить структуровану інформацію про об’єкти та їхні зв’язки. Великі мовні моделі при цьому виконують роль інтерфейсу інтерпретації запитів користувача та формування зрозумілих текстових відповідей. Така взаємодія дозволяє поєднати точність логічного виведення з гнучкістю обробки природної мови.

У даній роботі пропонується модель гібридної системи виведення знань, що поєднує використання графа знань та великої мовної моделі. Передбачається, що мовна модель буде використовуватися для інтерпретації запиту користувача та його перетворення у формальний запит до графа знань. Після виконання логічного виведення отримані результати використовуються для формування зрозумілої текстової відповіді.

Таким чином, використання гібридного підходу дозволяє поєднати можливості символічних та нейронних методів обробки знань, що створює основу для побудови більш ефективних інтелектуальних систем. У наступних підрозділах буде розглянуто основні підходи до інтеграції великих мовних моделей та графів знань, а також запропоновано архітектуру гібридної системи, що використовується у даному дослідженні [9], [13].

3.2 Підходи до інтеграції великих мовних моделей та графів знань

У сучасних дослідженнях у сфері штучного інтелекту активно розглядаються підходи до інтеграції великих мовних моделей та графів знань. Поєднання цих технологій дозволяє об’єднати структуроване представлення інформації з можливостями обробки природної мови, що відкриває нові можливості для побудови інтелектуальних інформаційних систем [10], [13].

Одним із найбільш поширених підходів є використання великої мовної моделі як інструменту інтерпретації запитів користувача. У такій архітектурі користувач формулює запит природною мовою, після чого мовна модель аналізує його зміст і перетворює у формальний запит до графа знань. Отриманий запит може бути представлений у вигляді SPARQL-запиту, який виконується над графом знань для отримання

необхідної інформації. Результати запиту повертаються системі та можуть бути використані для формування відповіді користувачу.

Інший підхід передбачає використання графа знань як джерела контекстної інформації для мовної моделі. У цьому випадку перед генерацією відповіді система спочатку здійснює пошук релевантних фактів у графі знань, після чого отримані дані передаються мовній моделі як додатковий контекст. Завдяки цьому модель формує відповіді, спираючись не лише на власні параметри, але й на структуровані знання з зовнішнього джерела [19]. Такий підхід дозволяє зменшити ймовірність генерації неточних або вигаданих тверджень.

Ще один підхід полягає у поєднанні графів знань із методами логічного виведення та подальшому використанні мовної моделі для пояснення отриманих результатів. У такій системі граф знань виконує функцію формальної бази знань, у якій зберігаються факти та правила. На основі цих даних здійснюється логічне виведення нових знань або визначення відповідей на запити. Після цього мовна модель використовується для формування текстового пояснення результатів виведення, що робить взаємодію із системою більш зрозумілою для користувача.

Зазначені підходи демонструють, що інтеграція великих мовних моделей та графів знань може реалізовуватися різними способами залежно від цілей системи та особливостей задачі [9]. У даній роботі буде використано підхід, у якому мовна модель застосовується для інтерпретації запиту користувача та формування зрозумілої відповіді, тоді як граф знань використовується для зберігання структурованої інформації та виконання логічного виведення. У наступному підрозділі буде розглянуто запропоновану архітектуру гібридної системи, що поєднує зазначені компоненти [17].

3.3 Запропонована архітектура гібридної системи виведення знань

У даній роботі пропонується архітектура гібридної системи виведення знань, що поєднує використання графа знань та великої мовної моделі [13]. Основною метою такої системи є поєднання структурованого представлення знань із можливостями обробки природної мови, що дозволяє забезпечити як точність логічного виведення, так і зручність взаємодії з користувачем.

Запропонована система використовує граф знань як основну базу для зберігання фактів про предметну область. У графі знань інформація представлена у вигляді зв’язків між об’єктами, що дозволяє формалізувати знання та виконувати запити до бази знань. Для отримання нових фактів або визначення відповідей на запити можуть застосовуватися правила логічного виведення.

Великі мовні моделі у цій архітектурі виконують роль інтерфейсу взаємодії з користувачем. Вони використовуються для аналізу запитів, сформульованих природною мовою, та їх перетворення у формалізований вигляд, придатний для виконання у графі знань. Крім того, мовна модель може використовуватися для формування зрозумілих текстових пояснень отриманих результатів.

У межах даної роботи гібридна система розглядається на прикладі задачі персоналізованого вибору навчальних курсів. У такій системі граф знань використовується для представлення інформації про курси, їх тематику, необхідні навички, рівень складності та можливі навчальні цілі користувача. Зв’язки між цими об’єктами дозволяють формалізувати знання про освітні ресурси та виконувати пошук курсів, що відповідають заданим критеріям.

Архітектура системи передбачає декілька основних етапів обробки запиту користувача. На першому етапі користувач формулює запит природною мовою, щодо пошуку навчальних курсів, які відповідають його

інтересам або поточному рівню знань. Далі мовна модель аналізує зміст запиту та визначає ключові параметри, що характеризують потреби користувача. На основі цієї інформації формується формалізований запит до графа знань.

Після цього система виконує запит до графа знань, у межах якого здійснюється пошук відповідних об’єктів та застосування правил логічного виведення. У результаті обробки визначається набір навчальних курсів, що відповідають заданим умовам або рекомендаціям системи.

На завершальному етапі отримані результати передаються мовній моделі, яка формує зрозумілу текстову відповідь для користувача. Така відповідь може містити рекомендації щодо вибору курсів, пояснення причин їх вибору або додаткову інформацію про навчальні програми.

Таким чином, запропонована архітектура поєднує можливості структурованого представлення знань, логічного виведення та обробки природної мови. Використання графа знань забезпечує точність та формальну узгодженість отриманих результатів, тоді як застосування великої мовної моделі дозволяє забезпечити зручну взаємодію із системою та формування зрозумілих текстових пояснень.

Ця модель гібридної системи виведення знань буде використана у подальшому дослідженні для проведення експериментального порівняння символічного, нейронного та гібридного підходів до обробки знань. Результати цього порівняння дозволять оцінити ефективність використання гібридної архітектури у задачах підтримки прийняття рішень.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ГІБРИДНОЇ СИСТЕМИ

4.1 Постановка задачі експерименту

Метою даного розділу є практичне дослідження ефективності різних підходів до виведення знань в інформаційних системах на прикладі задачі персоналізованого навчання. В якості предметної галузі обрано задачу рекомендації навчальних курсів на основі наявних знань, навичок та цілей користувача [6].

Персоналізоване навчання є актуальною задачею сучасних інформаційних систем, оскільки дозволяє формувати індивідуальні освітні траєкторії з урахуванням попереднього досвіду користувача, його інтересів та вимог до професійних компетенцій. Для формування рекомендацій необхідно враховувати структуровані знання про навчальні курси, їх взаємозв’язки, а також вміти інтерпретувати запити користувачів, сформульовані природною мовою.

У межах експерименту буде реалізовано три підходи до розв’язання задачі рекомендації курсів:

Символічний підхід, що базується на використанні графів знань та логічного виведення для визначення можливих навчальних траєкторій.

Підхід на основі великих мовних моделей (LLM), який використовує можливості нейронних мереж для аналізу текстових запитів користувача та формування рекомендацій.

Гібридний підхід, що поєднує використання графів знань та великих мовних моделей з метою підвищення точності та обґрунтованості отриманих результатів.

В рамках дослідження буде сформовано набір навчальних курсів, описано їх взаємозв’язки у вигляді графа знань, а також підготовлено набір запитів користувачів, які відображають типові сценарії вибору освітньої траєкторії. Для кожного підходу буде отримано результати рекомендацій,

які у наступному розділі будуть порівняні за визначеними критеріями якості.

Таким чином, експеримент дозволить оцінити переваги та обмеження символічного, нейромережевого та гібридного підходів у задачах інтелектуального аналізу знань.

4.2 Формалізація предметної галузі та підготовка даних

Для проведення експериментального дослідження сформовано формалізовану модель предметної галузі, що описує процес рекомендації навчальних курсів залежно від наявних компетенцій користувача. Формалізація предметної галузі дозволяє представити знання у структурованому вигляді та забезпечує можливість їх подальшого використання в межах символічного, нейромережевого та гібридного підходів. Представлення знань у формальному вигляді створює основу для логічного виведення нових фактів, виконання запитів до графа знань та інтеграції методів обробки природної мови [2], [3].

Концептуальна модель предметної галузі реалізована у вигляді онтології, що описує ключові сутності освітнього процесу та взаємозв’язки між ними. Основними класами онтології є Course, Skill та Student. Клас Course представляє навчальні дисципліни, які формують визначений набір компетенцій. Клас Skill описує окремі знання, вміння або компетентності, необхідні для успішного проходження курсів. Клас Student представляє користувача системи, для якого формується рекомендація навчальної траєкторії.

Зв’язки між сутностями предметної галузі визначено за допомогою об’єктних властивостей. Властивість hasSkill використовується для опису наявних компетенцій студента, властивість requiresSkill визначає перелік навичок, необхідних для проходження конкретного курсу, а властивість completedCourse дозволяє зафіксувати курси, які студент уже завершив.

Використання зазначених властивостей дозволяє формалізувати prerequisite залежності між навчальними дисциплінами у вигляді вимог до попередніх компетенцій. Таким чином, кожен курс характеризується множиною необхідних знань, які повинні бути сформовані до початку його вивчення.

У межах експерименту сформовано обмежений набір курсів у галузі інформаційних технологій, орієнтований на формування компетенцій у сфері аналізу даних та штучного інтелекту. До моделі включено базові курси з програмування та математичної підготовки, а також спеціалізовані курси, що охоплюють методи машинного навчання, аналізу даних та обробки природної мови. Між курсами визначено логічні залежності, що відображають послідовність їх вивчення. Наприклад, вивчення методів машинного навчання передбачає наявність знань з програмування, статистики та лінійної алгебри, тоді як більш спеціалізовані дисципліни, зокрема глибинне навчання, обробка природної мови та комп’ютерний зір, базуються на попередньому опануванні методів машинного навчання.

Кожному курсу поставлено у відповідність множину навичок, які формуються після його завершення. У моделі використано набір компетенцій, що охоплює базові навички програмування, володіння мовою Python, знання математичних методів, методів аналізу даних, машинного навчання та сучасних підходів штучного інтелекту. Така структура дозволяє відобразити логічну послідовність формування знань та забезпечує можливість автоматичного визначення доступності курсів залежно від рівня підготовки студента.

Для перевірки роботи моделі сформовано тестові профілі студентів, що містять інформацію про вже завершені навчальні курси та наявні компетенції. Наявність декількох профілів дозволяє змоделювати різні сценарії освітньої підготовки та перевірити здатність системи коректно визначати перелік доступних курсів відповідно до prerequisite залежностей. Використання формалізованого опису студентів забезпечує можливість

застосування логічного виведення для визначення відповідності між профілем користувача та вимогами навчальних дисциплін.

Отримана формалізація забезпечує можливість застосування логічних правил для виведення нових знань, виконання запитів до графа знань за допомогою мови SPARQL, а також використання текстових описів курсів у межах нейромережевого підходу на основі великих мовних моделей. Це створює єдину концептуальну основу для реалізації символічного, нейромережевого та гібридного підходів у межах експериментального дослідження.

4.3 Символічний підхід до рекомендації навчальних курсів на основі графа знань

4.3.1 Розробка онтології предметної галузі

У межах символічного підходу було розроблено онтологію предметної галузі персоналізованого навчання, яка забезпечує формалізоване представлення знань про навчальні курси, компетенції та взаємозв’язки між ними. Основною метою побудови онтології є створення структурованої моделі навчальної галузі, що дозволяє виконувати автоматичне логічне виведення рекомендацій курсів відповідно до наявних знань і навичок студента.

Онтологія реалізована з використанням мови OWL (Web Ontology Language), яка є стандартом представлення знань у семантичних системах. Використання OWL дозволяє формально описати класи, властивості та обмеження предметної галузі, а також застосовувати механізми логічного виведення (reasoning) для отримання нових фактів на основі заданих залежностей. Завдяки цьому рекомендації формуються не на основі статистичних методів або емпіричних припущень, а шляхом формального аналізу структури знань та логічних правил.

У процесі моделювання предметної галузі було визначено основні класи, що описують ключові сутності системи:

– Course – описує курси, які можуть бути рекомендовані студенту;

– Skill – описує компетенції та знання, які формуються в результаті проходження курсів;

– Student – описує користувача рекомендаційної системи;

– DifficultyLevel – визначає рівень складності навчального курсу;

– Domain – описує предметну область курсу.

Особливістю розробленої онтології є використання компетентнісного підходу до опису предметної галузі. Замість створення великої кількості тематичних підкласів курсу (наприклад ProgrammingCourse або MathCourse), предметна область описується через компетенції, які формуються або необхідні для проходження курсу. Такий підхід дозволяє забезпечити більшу гнучкість для моєї моделі.

Клас Skill має ієрархічну структуру, що дозволяє групувати компетенції за напрямами підготовки. Зокрема, визначено такі підкласи: ProgrammingSkill, MathSkill, DataSkill, AISkill.

Ієрархія класів дозволяє узагальнювати знання студента та враховувати належність компетенцій до відповідних галузей під час формування рекомендацій. Структура класів онтології представлена на рисунку 4.1.

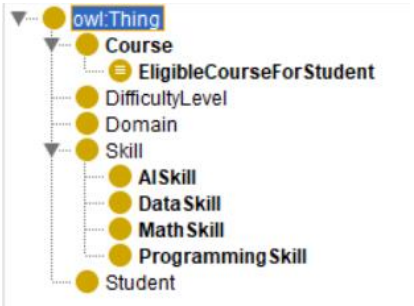

Рисунок 4.1 – Ієрархія класів онтології

Клас Course має підклас EligibleCourseForStudent, який використовується для представлення результатів логічного виведення. До цього класу автоматично відносяться курси, які задовольняють умови рекомендації відповідно до наявних компетенцій студента. Таким чином, підклас EligibleCourseForStudent не описує тип курсу, а визначає його статус у процесі формування рекомендацій.

Використання такого підходу дозволяє чітко відокремити структуру навчальної галузі від результатів логічного аналізу, що підвищує інтерпретованість моделі та спрощує процес її подальшого розширення.

Для опису взаємозв’язків між сутностями в онтології визначено об’єктні властивості, які формалізують залежності між курсами, компетенціями та студентами:

– requiresSkill – визначає компетенції, необхідні для проходження курсу;

– providesSkill – визначає компетенції, які формуються після завершення курсу;

– hasSkill – описує компетенції, якими володіє студент;

– completedCourse – визначає курси, які вже пройдені студентом;

– hasPrerequisite – визначає залежності між курсами;

– belongsToDomain – визначає предметну область курсу;

– hasDifficulty – визначає рівень складності курсу;

– eligibleForCourse – визначає курси, які можуть бути рекомендовані студенту.

Особливу роль у моделі відіграють властивості requiresSkill та providesSkill, оскільки саме через них встановлюється зв’язок між навчальними курсами та компетенціями. Властивість requiresSkill визначає знання, які необхідні для успішного проходження курсу, тоді як providesSkill описує результати навчання у вигляді набутих компетенцій. Такий підхід дозволяє формувати рекомендації на основі фактичних знань студента, незалежно від того, яким способом ці знання були отримані.

Властивості hasSkill та completedCourse використовуються для формування профілю студента, який відображає його поточний рівень підготовки. Властивість hasPrerequisite дозволяє встановити структурні залежності між курсами, що визначають логічну послідовність навчання.

Сукупність об’єктних властивостей онтології представлена на рисунку 4.2.

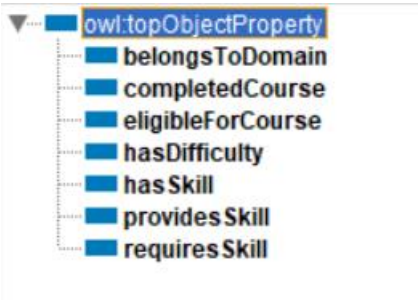

Рисунок 4.2 – Об’єктні властивості онтології

Завдяки використанню онтології знання про навчальні курси представлені у формалізованому вигляді, що дозволяє автоматично визначати нові зв’язки між сутностями та формувати рекомендації курсів без необхідності ручного аналізу залежностей між компетенціями. Використання формальної моделі знань забезпечує інтерпретованість результатів та дозволяє пояснити логіку формування рекомендацій, що є важливою перевагою символічного підходу порівняно з методами, заснованими виключно на статистичних залежностях.

4.3.2 Формування графа знань

На основі розробленої онтології було сформовано граф знань, який містить екземпляри навчальних курсів, компетенцій та зв’язків між ними. Граф знань є структурованим представленням інформації про предметну

область та дозволяє описати залежності між навчальними елементами у вигляді орієнтованого графа.

Кожен курс у графі знань пов’язаний із компетенціями, які формуються під час його проходження, за допомогою властивості providesSkill. Наприклад, курс PythonBasicsCourse формує компетенцію PythonSkill, а курс LinearAlgebraCourse формує компетенцію LinearAlgebraSkill. Таким чином визначається результат навчання для кожного курсу.

Залежності між курсами визначаються за допомогою властивості hasPrerequisite, яка дозволяє описати послідовність вивчення дисциплін. Наприклад, курс MachineLearningCourse має передумови у вигляді курсів PythonBasicsCourse, StatisticsCourse та LinearAlgebraCourse.

На рисунку 4.3 показано граф залежностей для курсами моєї онтології.

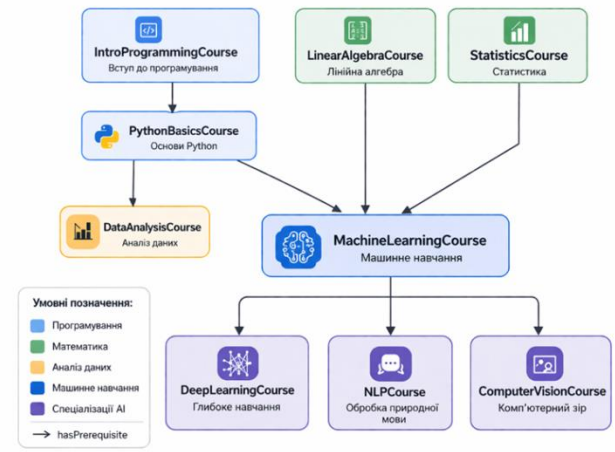

Рисунок 4.3 – Граф залежностей між курсами

Додатково використовується властивість requiresSkill, яка дозволяє узагальнити prerequisite залежності у вигляді вимог до компетенцій. Такий підхід дозволяє визначати доступність курсу на основі знань студента незалежно від того, яким саме шляхом ці знання були отримані.

Кожен курс також характеризується рівнем складності та належністю до певної предметної галузі. Таким чином формується граф знань, який описує структуру навчальної галузі та може бути використаний для автоматичного формування рекомендацій.

4.3.3 Формування профілів студентів

Для перевірки роботи розробленої моделі було сформовано профілі студентів із різним рівнем підготовки. Кожен студент описується множиною завершених курсів та набутих компетенцій, що дозволяє змоделювати різні сценарії навчання. Профіль студента формується за допомогою властивостей completedCourse та hasSkill.

Student1 характеризується базовим рівнем підготовки та має лише одну компетенцію ProgrammingFundamentalsSkill, що відповідає початковому рівню знань у сфері програмування.

Student2 має ширший набір компетенцій, що включає ProgrammingFundamentalsSkill, PythonSkill та StatisticsSkill. Такий набір знань дозволяє студенту переходити до курсів у сфері аналізу даних.

Student3 має розширений набір компетенцій, що включає ProgrammingFundamentalsSkill, PythonSkill, StatisticsSkill, LinearAlgebraSkill та MachineLearningSkill.

Приклад опису студента наведено на рисунку 4.4.

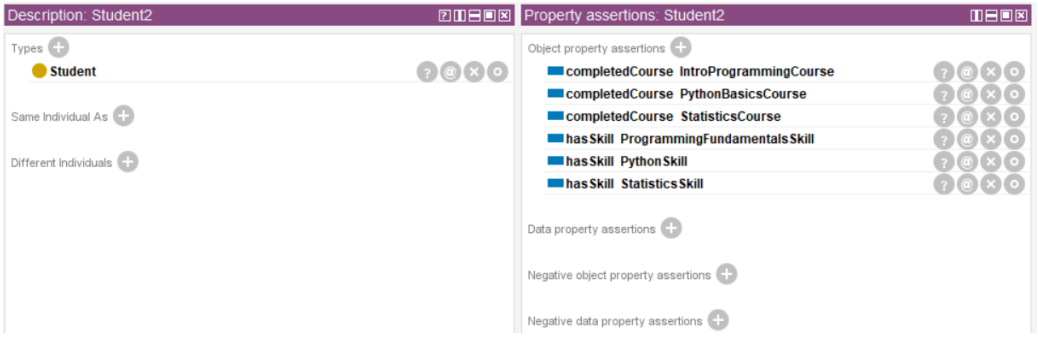

Рисунок 4.4 – Приклад опису студента у Protege

Використання кількох профілів студентів дозволяє продемонструвати, що система здатна формувати різні рекомендації залежно від рівня підготовки користувача.

4.3.4 Логічне виведення рекомендацій за допомогою SWRL

Для автоматизації процесу визначення доступних курсів було використано правила SWRL (Semantic Web Rule Language) [7]. Використання SWRL дозволяє формалізувати логіку рекомендацій у вигляді правил, які можуть бути використані reasoner для отримання нових фактів [20].

Основна ідея правила полягає у визначенні курсів, які студент може пройти на основі наявних компетенцій. Якщо студент має всі необхідні компетенції, визначені для курсу, то такий курс може бути рекомендований студенту.

Логіка рекомендації базується на перевірці відповідності між множиною компетенцій студента та множиною компетенцій, необхідних для проходження курсу. Приклад правил наведено на рисунку 4.5.

	Name	Rule
☑	S1	Student(?s) ^ completedCourse(?s, ?c) ^ providesSkill(?c, ?sk) -> hasSkill(?s, ?sk)
☑	S2	Student(?s) ^ hasSkill(?s, ?sk) ^ requiresSkill(?c, ?sk) -> eligibleForCourse(?s, ?c)

Рисунок 4.5 – SWRL правила

У загальному вигляді правило визначає, що якщо студент володіє всіма необхідними компетенціями курсу, то між студентом і курсом встановлюється зв’язок eligibleForCourse.

Використання SWRL дозволяє автоматично визначати нові зв’язки між студентами та курсами без необхідності ручного аналізу залежностей між компетенціями.

Таким чином, SWRL виконує функцію механізму логічного виведення, який формує множину потенційно доступних курсів для кожного студента.

4.3.5 Формування рекомендацій за допомогою SPARQL та результати виконання запиту

Для отримання списку рекомендованих курсів використано запити SPARQL, які дозволяють виконувати вибірку даних із графа знань відповідно до заданих умов [2], [3].

SPARQL-запит визначає курси, які відповідають поточному рівню підготовки студента та дозволяють розширити його компетенції. Логіка запиту базується на трьох основних умовах:

– студент має всі компетенції, необхідні для проходження курсу;
– студент ще не проходив відповідний курс;
– курс формує нову компетенцію, якою студент ще не володіє.

Такий підхід дозволяє уникнути рекомендації курсів, які не додають нових знань, або тих, які студент вже завершив. На рисунку 4.6 показано результат виконання SPARQL запиту.

student	course
Student1	DatabasesCourse
Student1	PythonBasicsCourse
Student1	StatisticsCourse
Student1	LinearAlgebraCourse
Student2	DatabasesCourse
Student2	DataAnalysisCourse
Student2	LinearAlgebraCourse
Student3	DatabasesCourse
Student3	NLPCourse
Student3	DataAnalysisCourse
Student3	ComputerVisionCourse
Student3	DeepLearningCourse

Рисунок 4.6 – Результати SPARQL запиту

Результати виконання SPARQL-запиту демонструють, що система формує множину курсів для студента залежно від компетенцій.

Для студента початкового рівня (Student1) система рекомендує базові курси, які формують фундаментальні знання у сфері програмування, математики та аналізу даних.

Для студента середнього рівня (Student2) система рекомендує курси, що дозволяють поглибити знання у сфері аналізу даних та підготуватися до вивчення методів машинного навчання.

Для студента просунутого рівня (Student3) система рекомендує спеціалізовані курси у сфері штучного інтелекту, такі як DeepLearningCourse, ComputerVisionCourse та NLPCourse.

Отримані результати демонструють, що рекомендації формуються з урахуванням поточного рівня знань користувача та спрямовані на поступове розширення його компетенцій.

4.3.6 Висновки до роботи символічної моделі

У результаті було розроблено символічну модель рекомендаційної системи, що базується на використанні онтології, логічних правил SWRL та запитів SPARQL. Побудована онтологія дозволяє формалізувати структуру навчальної галузі та представити знання у вигляді графа знань.

Використання SWRL дозволило автоматизувати процес визначення доступних курсів на основі компетенцій студента, а застосування SPARQL забезпечило можливість отримання персоналізованих рекомендацій.

Отримані результати демонструють, що запропонований підхід дозволяє враховувати рівень підготовки користувача та формувати множину релевантних курсів, що забезпечує можливість побудови індивідуальної навчальної траєкторії.

Таким чином, символічний підхід забезпечує інтерпретованість рекомендацій, прозорість логіки прийняття рішень та можливість подальшого розширення моделі шляхом додавання нових курсів.

4.4 Розробка моделі рекомендацій на основі великої мовної моделі

У межах дослідження, поряд із символічним підходом, було реалізовано модель рекомендацій навчальних курсів із використанням великої мовної моделі (Large Language Model, LLM). Метою застосування LLM є дослідження можливостей сучасних нейромережевих методів обробки природної мови для формування персоналізованих освітніх рекомендацій, а також проведення порівняльного аналізу ефективності нейромережевого та символічного підходів.

Використання великих мовних моделей у задачах рекомендацій є перспективним напрямом досліджень, оскільки такі моделі здатні враховувати семантичні зв’язки між компетенціями та навчальними темами, узагальнювати інформацію та формувати логічно узгоджені висновки на основі текстового опису предметної галузі. На відміну від символічних методів, які ґрунтуються на чітко визначених правилах логічного виведення, LLM використовують статистичні закономірності, отримані під час навчання на великих корпусах текстових даних. Це дозволяє моделі відтворювати експертні міркування та формувати рекомендації навіть у випадках неповноти або нечіткості вхідних даних.

У роботі мовна модель використовується як окремий інтелектуальний компонент системи рекомендацій, який формує перелік потенційно релевантних навчальних тем на основі опису наявних компетенцій студента у семантичному форматі JSON. Отримані результати використовуються для подальшого порівняння з результатами символічного логічного виведення, а також для побудови гібридної моделі рекомендацій, що поєднує переваги обох підходів.

Для реалізації нейромережевого підходу було використано локальне середовище виконання LLM Ollama, яке забезпечує можливість запуску відкритих мовних моделей на персональному комп’ютері без необхідності використання зовнішніх API сервісів. Такий підхід дозволяє забезпечити

відтворюваність експериментів, контроль параметрів генерації та незалежність від обмежень сторонніх платформ. Як базову модель було використано сучасну відкриту LLM Mistral, яка демонструє високі показники якості у задачах семантичного аналізу тексту та генерації рекомендацій.

Застосування локальної моделі дозволяє отримувати результати у контрольованому середовищі, що є важливою вимогою для наукових досліджень, оскільки забезпечує повторюваність експерименту та можливість подальшого аналізу результатів.

4.4.1 Підготовка даних для використання у великій мовній моделі

Для забезпечення узгодженості нейромережевого підходу із символічною моделлю предметної галузі було виконано підготовку структурованого набору даних у JSON-форматі, який зручно використовувати для генерації запитів до LLM.

Структуровані дані базуються на онтологічних залежностях між курсами та компетенціями. Зокрема, були використані такі об’єктні властивості:

– requiresSkill – визначає компетенції, необхідні для проходження курсу;

– providesSkill – визначає компетенції, які формуються після завершення курсу.

На основі цих властивостей було сформовано файл courses.json, що містить опис навчальних курсів та залежностей між ними у вигляді prerequisite-зв’язків. Кожен курс представлений у вигляді об’єкта з полями: перелік необхідних компетенцій та перелік компетенцій, що формуються після завершення курсу. Приклад реалізації файлу courses.json на рисунку 4.7.

Рисунок 4.7 – Приклад опису навчальних курсів

Такий підхід дозволяє представити онтологічні знання у форматі, який є зрозумілим для мовної моделі, зберігаючи при цьому логіку предметної галузі.

Окремо було сформовано файл students_profiles.json, який містить інформацію про наявні компетенції студентів. Для кожного студента визначено перелік компетенцій, отриманих у результаті проходження попередніх курсів. Приклад реалізації файлу students_profiles.json показано на рисунку 4.8.

Рисунок 4.8 – Приклад опису студентів

Використання JSON як проміжного представлення знань дозволяє відокремити етап побудови онтології від етапу застосування мовної моделі, що підвищує модульність системи та спрощує її подальше розширення.

Таким чином, підготовлений набір даних забезпечує відповідність між символічною моделлю предметної галузі та нейромережевим підходом, що є необхідною умовою для коректного порівняння результатів.

4.4.2 Формування запитів до LLM

Наступним етапом було формування текстових запитів до LLM, які відображають структуру компетенцій та курсових залежностей у вигляді природної мови. Запит містить:

– роль моделі як системи рекомендацій;
– перелік курсів у семантичному описі (skills, prerequisites);
– перелік компетенцій студента;

– інструкцію щодо формату відповіді у JSON-форматі, що включає список рекомендованих навчальних тем та коротке пояснення.

Реалізація текстового запиту показана на рисунку 4.9.

```
You are an expert educational recommendation system.

Your task is to recommend suitable learning topics for a student.

IMPORTANT RULES:
- Recommend ONLY based on provided skills
- Respect prerequisite relationships
- Avoid recommending already known skills
- DO NOT use exact course names
- DO NOT repeat items
- DO NOT include explanations inside the list
- Use natural language descriptions of learning topics

Courses (abstract descriptions, not names):
<list of courses with "requires skills" and "provides skills">

Student skills:
<list of student's current skills>

Return STRICTLY in the following format:

Recommended learning topics:
1. <short topic>
2. <short topic>
3. <short topic>

Explanation:
<2-3 sentences explaining why>
```

Рисунок 4.9 – Текстовий запит до AI моделі

Такий підхід дозволяє моделі аналізувати залежності між компетенціями та курсами у текстовому вигляді, що відповідає принципам функціонування великих мовних моделей.

Формування запиту здійснюється автоматично на основі JSON-файлів, що дозволяє масштабувати систему для більшої кількості курсів або студентів без необхідності ручного редагування тексту.

Використання структурованого prompt забезпечує відтворюваність результатів та дозволяє контролювати формат відповіді моделі.

4.4.3 Реалізація програмного модуля взаємодії з LLM

Для автоматизації процесу отримання рекомендацій було розроблено програмний модуль мовою Python, який забезпечує взаємодію з локальною мовною моделлю через інтерфейс Ollama.

Лістинг 4.1 – Програмний код реалізації модулю Ollama

```
import json
import ollama
import pandas as pd

courses = json.load(open("courses.json"))
students = json.load(open("student_profiles.json"))

def format_courses():
    text = ""
    for c in courses:
        text += f"""
Learning opportunity:
- requires skills: {", ".join(c["requires"])}
- provides skills: {", ".join(c["provides"])}
"""
    return text

results = []

for s in students:
    prompt = f"""
You are an expert educational recommendation system.

Rules:
- Recommend ONLY based on student's current skills
- Respect prerequisites
- Avoid recommending known skills
- Use natural language
- DO NOT repeat topics

Courses:
{format_courses()}
```

Продовження лістингу 4.1

```
Student skills:
{s["skills"]}

Return STRICTLY in JSON format:

{{
      "RecommendedTopics":     ["<topic1>",     "<topic2>",
"<topic3>"],
  "Explanation": "<short 2-3 sentence explanation>"
}}
"""
    response = ollama.chat(
        model='mistral',
        messages=[{"role": "user", "content": prompt}]
    )

    output = response['message']['content'].strip()

    try:
        data = json.loads(output)
    except:
          data = {"RecommendedTopics": [], "Explanation":
output}

    results.append({
        "student": s["student"],
        "topics": data["RecommendedTopics"],
        "explanation": data["Explanation"]
    })

df = pd.DataFrame(results)
df.to_csv("llm_results.csv", index=False)
```

Розроблений модуль виконує такі функції:

– зчитування даних про курси та студентів із JSON-файлів;

– автоматичне формування запиту до LLM усемантичному форматі;

– передача запиту та отримання відповіді у JSON-форматі;

– збереження результатів у структурованому CSV-файлі.

Автоматизація процесу дозволяє виконувати багаторазові експерименти та отримувати результати для різних наборів даних, що є важливим для проведення подальшого порівняльного аналізу.

Використання мови Python зумовлене її широким застосуванням у задачах обробки даних та інтеграції з інструментами штучного інтелекту.

Отримані результати зберігаються у окремому файлі, що дозволяє виконувати подальший аналіз точності рекомендацій та порівняння з результатами символічного підходу.

4.4.4 Результати формування рекомендацій за допомогою LLM

У результаті виконання моделі було отримано перелік рекомендованих навчальних тем для кожного студента відповідно до наявних компетенцій, представлених у семантичному форматі. Аналіз результатів показав, що велика мовна модель здатна враховувати залежності між компетенціями та формувати логічно узгоджені траєкторії навчання у більшості випадків.

Для студентів із базовим рівнем підготовки модель пропонує фундаментальні теми, що сприяють формуванню ключових компетенцій у програмуванні, статистиці та роботі з даними. Студенти зі середнім рівнем підготовки отримують рекомендації щодо більш комплексних тем, включаючи аналіз даних, лінійну алгебру та основи машинного навчання.

Для студентів із високим рівнем підготовки модель пропонує спеціалізовані напрямки сучасного штучного інтелекту, такі як глибоке навчання, обробка природної мови та комп’ютерний зір. Водночас було зафіксовано, що LLM іноді формує рекомендації без повного врахування всіх prerequisite-залежностей, що пояснюється статистичним характером моделі та роботою на основі закономірностей текстових даних.

Приклад результату виведення представлено на рисунку 4.10.

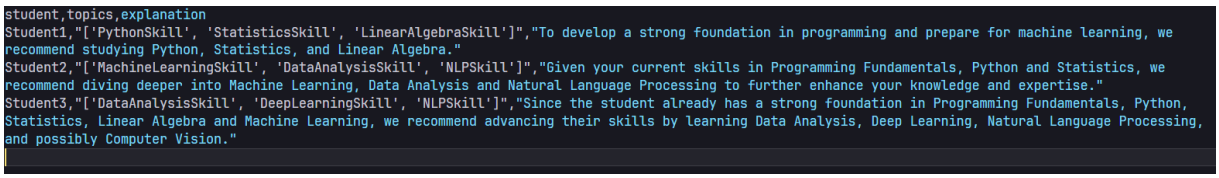

Рисунок 4.10 – Результат виконання, файл llm_result.csv

Отримані результати демонструють здатність LLM відтворювати загальну логіку побудови навчальних траєкторій, проте можуть містити незначні неточності у випадках складних залежностей між компетенціями.

4.4.5 Висновки до роботи LLM

Розроблена модель рекомендацій на основі великої мовної моделі забезпечила альтернативний підхід до формування персоналізованих освітніх траєкторій, що доповнює традиційні символічні методи. Використання даних, підготовлених на основі онтології та представлених у семантичному форматі, гарантує узгодженість нейромережевого підходу із структурованою моделлю предметної галузі.

Проведені експерименти показали, що LLM здатна формувати осмислені рекомендації навчальних тем на основі опису компетенцій, однак не гарантує повного дотримання формальних обмежень предметної галузі та prerequisite-зв’язків.

Особливості роботи моделі свідчать про доцільність використання гібридного підходу, що поєднує формальну точність онтологічного reasoning та гнучкість нейромережевих методів. Це забезпечує ефективне поєднання семантичної узгодженості та адаптивності рекомендацій, що стане предметом наступного розділу, у якому буде розглянуто інтеграцію символічної та нейромережевої моделей та визначено метрики оцінювання якості рекомендацій.

4.5 Розробка гібридної моделі рекомендацій на основі інтеграції LLM та онтологічного підходу

У межах дослідження було розроблено гібридну модель рекомендацій навчальних курсів, яка поєднує можливості великої мовної моделі (LLM) із формальним логічним виведенням на основі онтології OWL. Метою створення гібридної моделі є підвищення якості рекомендацій шляхом об’єднання семантичної гнучкості нейромережевого підходу та строгості символічного reasoning.

Застосування лише одного з підходів має певні обмеження. Нейромережеві моделі здатні ефективно виявляти приховані семантичні залежності між компетенціями та навчальними темами, однак не гарантують повного дотримання формальних prerequisite-зв’язків. У свою чергу, онтологічні моделі забезпечують логічну коректність рекомендацій, проте можуть демонструвати обмежену здатність до узагальнення та врахування контексту. Поєднання цих підходів дозволяє компенсувати недоліки кожного з них та сформувати більш узгоджену систему підтримки прийняття рішень у задачі формування індивідуальних освітніх траєкторій.

Гібридна модель базується на послідовному використанні двох типів знань: символічних знань, представлених у вигляді онтології, та статистичних знань, отриманих за допомогою LLM. Онтологія описує формальні залежності між навчальними курсами та компетенціями, включаючи prerequisite-зв’язки, тоді як мовна модель формує семантичні рекомендації на основі текстового опису навичок студента. Інтеграція результатів виконується шляхом комбінування оцінок релевантності курсів, отриманих з обох компонентів.

Таким чином, запропонований підхід відповідає концепції neuro-symbolic AI, у якій поєднуються методи символьного представлення знань та нейромережеві алгоритми обробки природної мови.

4.5.1 Архітектура гібридної системи рекомендацій

Архітектура гібридної моделі рекомендацій розроблена з урахуванням необхідності поєднання двох різних підходів до представлення та обробки знань – символічного та нейромережевого. Такий підхід дозволяє враховувати як формальні залежності між навчальними курсами, так і семантичний контекст компетенцій студента. У результаті формується більш гнучка та інтелектуальна система підтримки прийняття рішень щодо побудови індивідуальної освітньої траєкторії.

Запропонована архітектура базується на багатокомпонентній структурі, у межах якої кожен модуль виконує окрему функцію в загальному процесі формування рекомендацій. Основним джерелом формальних знань виступає онтологія предметної галузі, що містить опис навчальних курсів, компетенцій та зв’язків між ними. Зокрема, в онтології визначені відношення requiresSkill та providesSkill, які відображають необхідні та набуті компетенції відповідно. Завдяки використанню онтології забезпечується формалізація prerequisite-зв’язків і можливість логічного виведення нових знань.

На основі наявної інформації про студента, зокрема переліку вже пройдених курсів, застосовується механізм логічного виведення з використанням правил SWRL. Це дозволяє автоматично визначити множину компетенцій, які студент уже сформував у процесі навчання. Таким чином, формується структуроване представлення поточного рівня підготовки користувача системи.

Після визначення наявних компетенцій виконується SPARQL-запит до онтології, який відбирає курси відповідно до формальних обмежень предметної галузі. До таких обмежень належать:

– наявність необхідних prerequisite-компетенцій;

– відсутність курсу у переліку вже пройдених;

– можливість отримання нових знань і навичок.

Паралельно із символічною обробкою даних у системі функціонує модуль на основі великої мовної моделі. LLM аналізує текстове представлення компетенцій студента та визначає навчальні теми, які є семантично релевантними для подальшого розвитку професійних навичок. Крім того, мовна модель генерує текстові пояснення рекомендацій, що підвищує прозорість та інтерпретованість роботи системи.

Загальна схема взаємодії компонентів гібридної моделі рекомендацій наведена на рисунку 4.11.

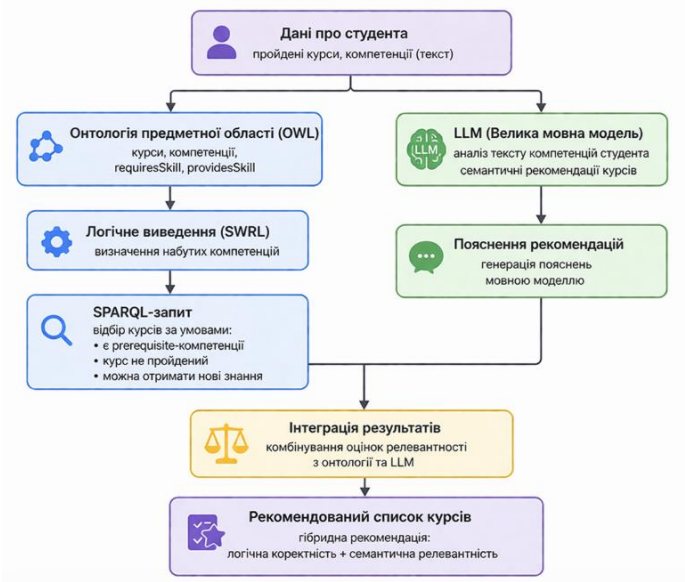

Рисунок 4.11 – Архітектура гібридної системи рекомендацій

Після отримання результатів від обох підсистем виконується їх інтеграція в межах єдиного модуля оцінювання. На цьому етапі поєднуються логічні результати, отримані з онтології, та семантичні оцінки релевантності, сформовані мовною моделлю. Об’єднання результатів дозволяє сформувати узгоджену оцінку доцільності вивчення кожного курсу.

Заключним етапом роботи системи є формування підсумкового списку рекомендованих курсів. Отримані рекомендації поєднують логічну

коректність prerequisite-зв’язків із контекстною релевантністю, що забезпечує більш точний та персоналізований підбір навчальних матеріалів для студента. Такий підхід відповідає концепції neuro-symbolic AI та демонструє ефективність інтеграції різних типів знань у задачах освітньої аналітики.

4.5.2 Інтеграція результатів логічного виведення та LLM

Інтеграція результатів виконується шляхом зіставлення навчальних тем, запропонованих мовною моделлю, із компетенціями, що формуються після проходження курсів, описаних в онтології. Для цього використовується семантичне порівняння текстових описів компетенцій та тем навчання.

Кожен курс отримує оцінку релевантності, яка формується на основі двох складових: логічної відповідності вимогам предметної галузі та семантичної подібності до рекомендацій LLM. Логічна відповідність визначається результатами виконання SPARQL-запиту та приймає бінарне значення. Семантична подібність визначається на основі аналізу текстових представлень компетенцій та рекомендованих тем.

Підсумкова оцінка курсу визначається за формулою 4.1:

$$Score_{final} = 0.7 * Score_{ont} + 0.3 * Score_{LLM} \quad (4.1)$$

де $Score_{ont}$ – відповідність формальним prerequisite-залежностям;

$Score_{LLM}$ – характеризує ступінь семантичної відповідності рекомендаціям мовної моделі.

Використання зваженої функції дозволяє контролювати вплив кожного з компонентів системи. Більша вага символічної складової забезпечує логічну коректність рекомендацій, тоді як внесок LLM дозволяє

враховувати контекстні особливості предметної галузі та формувати більш гнучкі освітні траєкторії.

Такий підхід забезпечує узгодженість результатів та зменшує вплив можливих помилок, пов’язаних із статистичним характером мовної моделі.

4.5.3 Реалізація програмного модуля гібридної рекомендаційної системи

Для реалізації гібридного підходу було розроблено програмний модуль мовою Python, який забезпечує інтеграцію результатів SPARQL-запитів та рекомендацій LLM. Програмна реалізація передбачає послідовне виконання кількох етапів: зчитування результатів логічного виведення, імпорт результатів роботи мовної моделі, обчислення підсумкових оцінок релевантності та формування узагальненого набору рекомендацій.

Модуль використовує табличне представлення даних у форматі CSV, що дозволяє забезпечити сумісність результатів різних етапів обробки. Результати SPARQL-запитів містять перелік курсів, дозволених відповідно до логічних обмежень онтології. Результати роботи LLM містять перелік рекомендованих навчальних тем та текстові пояснення.

У процесі інтеграції кожен курс перевіряється на відповідність рекомендаціям LLM шляхом аналізу переліку компетенцій, що формуються після його проходження. Якщо компетенції курсу відповідають рекомендованим темам, курс отримує додатковий коефіцієнт релевантності. Після обчислення підсумкової оцінки результати впорядковуються за спаданням значення Score_final.

Лістинг 4.2 демонструє приклад програмної реалізації.

Лістинг 4.2 – Фрагмент програмного коду гібридної моделі

```
import re
import ast
```

Продовження лістингу 4.2

```
import pandas as pd

def llm_similarity(topics, course):
"""
Обчислення семантичної відповідності курсу рекомендаціям
LLM.
topics – список тем, запропонованих мовною моделлю
course – назва курсу з онтології
Повертає значення у діапазоні 0..1
"""

# Виділення слів з назви курсу (без слова Course)
course_words = re.findall(r'[A-Z][a-z]*', course)
course_words = [w.lower() for w in course_words if
w.lower() != "course"]

if not course_words:
return 0.0

matches = sum(
1 for w in course_words
if any(w in topic.lower() for topic in topics)
)
return matches / len(course_words)


ALPHA = 0.7 # вага логічного виведення (онтологія)
BETA = 0.3 # вага рекомендацій LLM

llm_results = pd.read_csv("llm_results.csv")
hybrid_results = []

for row in llm_results.itertuples():
student = row.student
```

Продовження лістингу 4.2

```
topics = ast.literal_eval(row.topics)

ontology_courses = ontology_recommendations.get(student,
[])
scored_courses = []

for course in ontology_courses:
ontology_score = 1.0
llm_score = llm_similarity(topics, course)

final_score = ALPHA * ontology_score + BETA * llm_score

if final_score >= 0.75:
scored_courses.append((course, round(final_score, 2)))

scored_courses.sort(key=lambda x: x[1], reverse=True)

hybrid_results.append({
"student": student,
"recommended_courses": scored_courses
})

rows = []

for entry in hybrid_results:
    student = entry["student"]
    for c in entry["recommended_courses"]:
        rows.append({
            "student": student,
            "course": c["course"],
            "final_score": c["final_score"],
            "llm_score": c["llm_score"],
            "explanation": c["explanation"]
        })

df = pd.DataFrame(rows)
```

Запропонована реалізація демонструє можливість інтеграції символічного та нейромережевого підходів у межах єдиного програмного середовища. Використання Python забезпечує гнучкість обробки даних та можливість масштабування системи для більшої кількості навчальних курсів.

4.5.4 Аналіз результатів роботи гібридної моделі

Результати застосування гібридної моделі демонструють підвищення узгодженості рекомендацій порівняно з використанням лише одного з підходів. Курси, що не відповідають prerequisite-залежностям, автоматично виключаються на етапі логічного виведення, що гарантує формальну коректність результатів.

У той же час використання мовної моделі дозволяє враховувати ширший контекст предметної галузі та пропонувати альтернативні напрями розвитку компетенцій, які можуть не бути явно представлені у структурі онтології. Це дозволяє формувати більш гнучкі та адаптивні освітні траєкторії.

Порівняння результатів показало, що гібридна модель зменшує кількість нерелевантних рекомендацій, які можуть виникати при використанні лише LLM, а також доповнює символічну модель новими семантичними зв’язками між компетенціями. Отримані результати свідчать про ефективність комбінування методів різної природи у задачах рекомендацій.

4.5.5 Висновки щодо гібридної моделі рекомендацій

Запропонована гібридна модель демонструє можливість ефективного поєднання нейромережевих методів обробки природної мови та символічних підходів до представлення знань. Інтеграція LLM та

онтологічного reasoning дозволяє підвищити якість рекомендацій за рахунок одночасного врахування формальних обмежень предметної галузі та семантичного контексту навчальних компетенцій.

Розроблена модель забезпечує відтворюваність експерименту, модульність архітектури та можливість подальшого розширення бази знань. Отримані результати підтверджують доцільність застосування neuro-symbolic підходів у задачах формування персоналізованих освітніх траєкторій.

Застосування гібридної моделі дозволяє підвищити узгодженість рекомендацій, зменшити вплив помилок генеративних моделей та забезпечити логічну обґрунтованість результатів. Це свідчить про перспективність використання інтегрованих підходів штучного інтелекту у системах підтримки прийняття рішень у сфері освіти.

5 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ГІБРИДНОЇ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

5.1 Методика оцінювання якості роботи моделей

У попередніх розділах було розглянуто теоретичні засади побудови гібридної інтелектуальної системи, яка поєднує можливості великих мовних моделей та онтологічного представлення знань. Основною метою даного розділу є експериментальна перевірка ефективності запропонованого підходу та підтвердження доцільності використання гібридної архітектури для задач підтримки прийняття рішень [6], [10].

Для оцінювання якості роботи моделей було сформовано тестовий набір запитів, який відображає типові сценарії використання інтелектуальної системи. Кожен запит передбачає отримання рекомендації або висновку на основі заданих параметрів предметної галузі. Оцінювання результатів проводилося для трьох варіантів реалізації системи: онтологічної моделі, LLM моделі, гібридної моделі, що поєднує обидва підходи.

Якість відповідей оцінювалася за чотирма ключовими критеріями:

– Accuracy характеризує правильність отриманої рекомендації відносно очікуваного результату;

– Logical consistency визначає ступінь логічної узгодженості відповіді та відсутність суперечностей між окремими твердженнями;

– Completeness відображає повноту сформованої рекомендації, тобто наскільки відповідь охоплює всі необхідні аспекти запиту;

– Explainability характеризує рівень інтерпретованості результату та можливість пояснення логіки отримання рекомендації.

Для кожного критерію було обрано шкалу оцінювання, що дозволяє провести кількісне порівняння моделей. Accuracy оцінюється як бінарна величина, де значення 1 відповідає коректній рекомендації. Logical

consistency та completeness визначаються у вигляді нормованих значень у діапазоні від 0 до 1. Explainability оцінюється за експертною шкалою від 1 до 5.

Використання декількох критеріїв дозволяє комплексно оцінити ефективність моделей та визначити сильні і слабкі сторони кожного підходу.

5.2 Результати оцінювання моделей

Результати обчислення середніх значень метрик наведено у таблиці 5.1, яка відображає узагальнені показники якості роботи трьох досліджуваних підходів: онтологічної моделі, великої мовної моделі та гібридної інтелектуальної системи.

Таблиця 5.1 – Порівняння моделей за основними метриками

method	accuracy	logical_consistency	completeness	explainability
ontology	0,8	0,87369192	0,69092012	3,204001751
llm	0,7	0,663998836	0,835765392	4,102495009
hybrid	0,933333333	0,926825489	0,904663482	4,497540658

Аналіз отриманих даних свідчить про наявність помітних відмінностей у поведінці моделей. Онтологічний підхід характеризується високими значеннями logical consistency, що пояснюється використанням формалізованої структури знань та механізмів логічного виводу. Завдяки чітко визначеним відношенням між концептами забезпечується узгодженість результатів та відсутність суперечностей у сформованих рекомендаціях. Водночас показник completeness є відносно нижчим, що пов’язано з обмеженістю бази знань, яка формується вручну та не завжди охоплює всі можливі варіанти запитів.

Велика мовна модель демонструє протилежну тенденцію. Значення completeness є вищим, оскільки модель здатна використовувати узагальнені знання, отримані під час навчання на великих масивах текстових даних. Це дозволяє формувати більш розгорнуті рекомендації, однак у деяких випадках спостерігається зниження logical consistency через появу непослідовних або недостатньо обґрунтованих тверджень. Подібні ситуації пов’язані з відсутністю формального механізму перевірки знань, що є характерною проблемою великих мовних моделей.

Гібридна модель поєднує переваги обох підходів і демонструє найвищі значення accuracy та logical consistency. Підвищення точності рекомендацій досягається завдяки використанню онтологічної перевірки результатів генерації, що дозволяє зменшити кількість некоректних відповідей. Одночасно забезпечується достатній рівень completeness, оскільки мовна модель дозволяє розширювати відповіді з урахуванням контексту запиту.

Візуальне порівняння отриманих результатів наведено на рисунку 5.1.

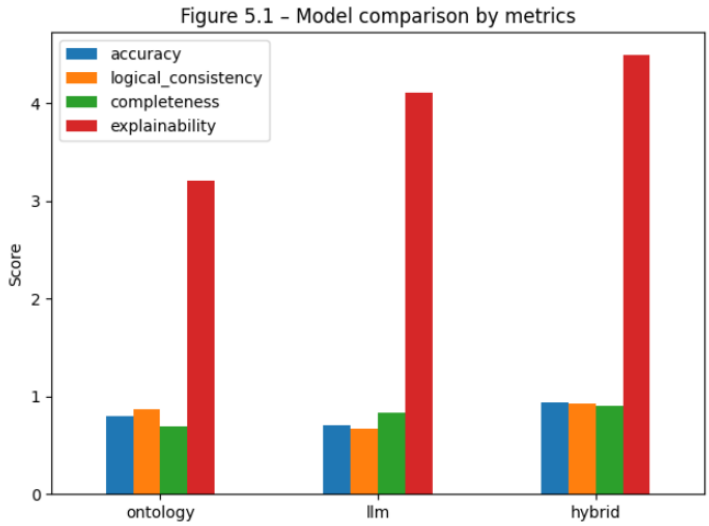

Рисунок 5.1 – Порівняння моделей за основними метриками

З рисунка видно, що гібридна модель демонструє найбільш збалансовані значення за всіма критеріями. Найбільш суттєве покращення спостерігається для показника logical consistency, що підтверджує доцільність використання онтологічного reasoning для перевірки відповідей великої мовної моделі. Водночас показник completeness також залишається на високому рівні, що свідчить про ефективність використання генеративних можливостей нейромережевого підходу.

Таким чином, результати дослідження підтверджують, що поєднання символічного та нейромережевого підходів дозволяє підвищити загальну якість роботи інтелектуальної системи. Подальший аналіз спрямований на оцінювання ступеня покращення характеристик гібридної моделі відносно базових підходів.

5.3 Аналіз покращення якості гібридної моделі

Для визначення ефективності запропонованої архітектури було виконано обчислення відносного покращення значень метрик гібридної моделі порівняно з онтологічною моделлю та великою мовною моделлю. Результати розрахунків наведено у таблиці 5.2.

Таблиця 5.2 – Відносне покращення метрик гібридної моделі, %

Metrics	Improvement vs llm (%)	Improvement vs ontology (%)
accuracy	33,33333333	16,66666667
Logical consistency	39,58239676	6,081499391
completeness	8,243711706	30,93604539
explainability	9,629399861	40,37260298

Отримані значення свідчать про суттєве зростання accuracy порівняно з використанням лише великої мовної моделі. Це підтверджує

гіпотезу про те, що використання формалізованого представлення знань дозволяє зменшити кількість помилкових рекомендацій. Особливо помітним є покращення показника logical consistency, що пояснюється застосуванням механізму логічного виводу, який забезпечує узгодженість сформованих тверджень.

Позитивна динаміка показника completeness демонструє, що гібридна модель не втрачає здатності формувати змістовні та інформативні відповіді. Це досягається за рахунок використання мовної моделі як джерела контекстуальної інформації, що доповнює структуровані знання предметної галузі.

Окрему увагу було приділено аналізу показника explainability, який характеризує рівень зрозумілості результатів роботи системи. Відповідні результати наведено на рисунку 5.2.

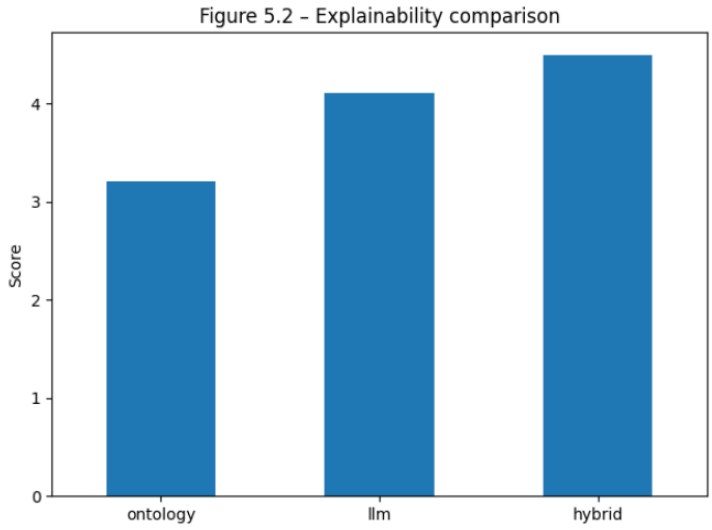

Рисунок 5.2 – Порівняння моделей за показником explainability

Як видно з рисунка, гібридна модель демонструє найвищий рівень explainability. Це пояснюється можливістю використання онтологічних зв’язків для обґрунтування рекомендацій, що дозволяє відстежити логіку формування результату. Наявність пояснень є важливим фактором для

практичного застосування інтелектуальних систем, особливо у задачах підтримки прийняття рішень, де користувач повинен розуміти причини отримання певної рекомендації.

Отримані результати свідчать про те, що гібридний підхід дозволяє досягти компромісу між точністю, повнотою та інтерпретованістю результатів, що є важливою вимогою до сучасних інтелектуальних систем [9], [13].

5.4 Аналіз типових помилок моделей

У процесі експериментального дослідження було виявлено характерні типи помилок, притаманні кожному з досліджуваних підходів. Онтологічна модель у деяких випадках формує неповні рекомендації через відсутність необхідних знань у базі. Це обмеження пов’язане з необхідністю попереднього формування структури предметної галузі, що потребує значних витрат часу та ресурсів. Крім того, розширення онтології потребує залучення експертів, що ускладнює масштабування системи.

Велика мовна модель характеризується високим рівнем узагальнення, проте іноді генерує правдоподібні, але некоректні твердження. Такі помилки виникають через відсутність механізму формальної перевірки знань та можуть призводити до появи логічних суперечностей у відповідях.

Гібридна модель дозволяє зменшити кількість подібних помилок завдяки використанню онтологічного reasoning як механізму валідації результатів генерації. Використання структурованих знань дозволяє відфільтрувати некоректні твердження та забезпечити узгодженість рекомендацій.

Загальне співвідношення результатів роботи моделей за всіма метриками наведено на рисунку 5.3.

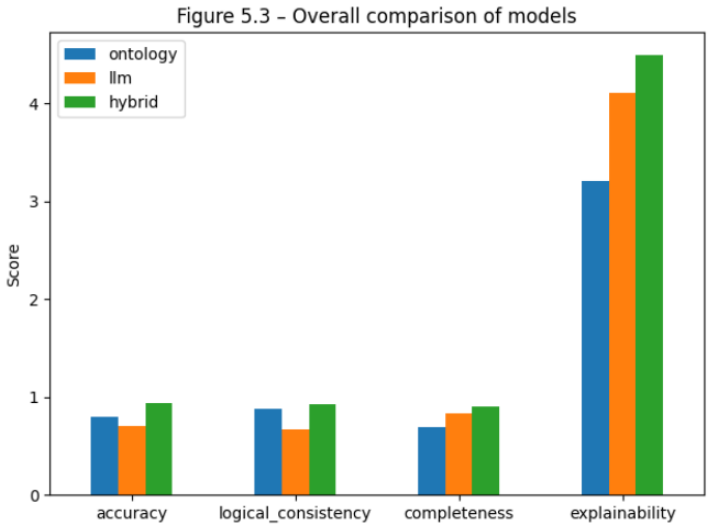

Рисунок 5.3 – Загальне порівняння моделей

Представлені результати демонструють, що гібридний підхід забезпечує найбільш стабільні показники якості серед досліджуваних моделей. Особливо важливим є зменшення кількості логічних суперечностей, що є однією з ключових проблем використання великих мовних моделей у задачах підтримки прийняття рішень.

Таким чином, результати експериментального дослідження підтверджують доцільність використання гібридних інтелектуальних систем, які поєднують переваги символічного та нейромережевого підходів, що дозволяє підвищити точність, узгодженість та інтерпретованість отриманих рекомендацій.

ВИСНОВКИ

У процесі виконання даної кваліфікаційної роботи було досліджено проблему підвищення якості інтелектуальних рекомендацій шляхом поєднання методів обчислювального інтелекту та символічного представлення знань. Проведено аналіз сучасних підходів до побудови інтелектуальних систем, зокрема великих мовних моделей, онтологічних систем та графа знань, визначено їх переваги, обмеження та особливості застосування у задачах підтримки прийняття рішень.

У роботі розглянуто принципи функціонування великих мовних моделей, які забезпечують здатність до генерації узагальнених відповідей на основі попереднього навчання на великих обсягах текстових даних. Разом з тим встановлено, що використання лише генеративного підходу може призводити до появи логічних суперечностей та так званих галюцинацій, що знижує надійність отриманих результатів. Проведений аналіз показав, що символічні підходи, зокрема онтології та графи знань, забезпечують високий рівень формальної узгодженості знань, однак характеризуються обмеженою повнотою через залежність від явно визначеної структури предметної галузі.

У результаті виконання роботи було розроблено методологію побудови гібридної інтелектуальної системи, що поєднує можливості великих мовних моделей та онтологічного логічного виводу. Запропонований підхід дозволяє використовувати переваги генеративних моделей для формування повних рекомендацій та одночасно застосовувати формальні механізми перевірки знань для підвищення логічної узгодженості результатів.

Було спроєктовано архітектуру гібридної моделі, яка включає компонент інтерпретації запиту користувача на основі мовної моделі, модуль представлення знань у вигляді онтології та механізм логічного виводу, що забезпечує перевірку коректності сформованих рекомендацій.

Реалізація взаємодії між компонентами дозволила створити єдиний механізм оцінювання релевантності результатів із використанням комбінованої функції оцінювання.

Для перевірки ефективності запропонованого підходу було проведено експериментальне дослідження із використанням набору тестових запитів. Оцінювання якості роботи моделей здійснювалося за метриками accuracy, logical consistency, completeness та explainability, що дозволяють комплексно оцінити як коректність результатів, так і їх інтерпретованість.

Результати експерименту показали, що онтологічна модель демонструє високий рівень логічної узгодженості, однак характеризується нижчими показниками повноти відповідей. Велика мовна модель забезпечує високу повноту рекомендацій, проте в окремих випадках допускає логічні неточності. Запропонована гібридна модель показала найбільш збалансовані результати за всіма критеріями оцінювання, зокрема досягнуто підвищення показників accuracy та logical consistency порівняно з окремим використанням базових підходів.

Отримані результати підтверджують ефективність використання нейросимволічного підходу для підвищення якості інтелектуальних систем. Поєднання генеративних можливостей великих мовних моделей із формальним представленням знань дозволяє зменшити кількість помилок, підвищити надійність рекомендацій та забезпечити кращу інтерпретованість результатів.

Практична цінність роботи полягає у можливості застосування розробленої моделі у задачах підтримки прийняття рішень, рекомендаційних системах, інтелектуальних навчальних середовищах та системах управління знаннями. Запропонований підхід може бути використаний як основа для подальшого розвитку інтелектуальних систем, що потребують високого рівня пояснюваності та логічної узгодженості результатів.

Програмну реалізацію розробленої моделі виконано мовою Python із використанням сучасних бібліотек для обробки природної мови та роботи з графами знань. Проведене дослідження підтвердило доцільність використання гібридних моделей у задачах інтелектуального аналізу знань та визначило перспективи подальшого розвитку, зокрема розширення онтології, збільшення обсягу навчальних даних та вдосконалення механізмів інтеграції нейромережевих і символічних методів.

У результаті виконання роботи поставлену задачу вирішено, досягнуто мету дослідження та визначено напрями подальших наукових досліджень у сфері neuro-symbolic artificial intelligence.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Terziyan V., Vitko O., Terziyan O. Semantic AI for future industries: bridging explainability and integration in black box models. *Procedia computer science*. 2026. Vol. 277. P. 2235–2246. URL: https://doi.org/10.1016/j.procs.2026.02.261 (date of access: 30.04.2026).

2. Heath T., Bizer C. Linked data: evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*. 2011. Vol. 1, no. 1. P. 1–136. URL: https://doi.org/10.2200/s00334ed1v01y201102wbe001 (date of access: 26.01.2026).

3. Hitzler P., Krotzsch M., Rudolph S. Foundations of semantic web technologies. Chapman and Hall/CRC, 2009. URL: https://doi.org/10.1201/9781420090512 (date of access: 28.01.2026).

4. Knowledge graphs / A. Hogan et al. *Synthesis lectures on data, semantics, and knowledge*. 2021. Vol. 12, no. 2. P. 1–257. URL: https://doi.org/10.2200/s01125ed1v01y202109dsk022 (date of access: 29.01.2026).

5. Freebase / K. Bollacker et al. *The 2008 ACM SIGMOD international conference*, Vancouver, Canada, 9–12 June 2008. New York, New York, USA, 2008. URL: https://doi.org/10.1145/1376616.1376746 (date of access: 01.02.2026).

6. Neuro-Symbolic approaches in artificial intelligence / P. Hitzler et al. *National science review*. 2022. URL: https://doi.org/10.1093/nsr/nwac035 (date of access: 05.02.2026).

7. Garcez A. S. D., Gabbay D. M., Lamb L. C. Neural-Symbolic cognitive reasoning. Springer, 2008. 198 p.

8. Marcus G. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. 2020. URL: https://arxiv.org/abs/2002.06177 (date of access: 11.02.2026).

9. Wang W., Yang Y., Wu F. Towards data-and knowledge-driven AI: a survey on neuro-symbolic computing. *IEEE transactions on pattern analysis and machine intelligence*. 2024. P. 1–22. URL: https://doi.org/10.1109/tpami.2024.3483273 (date of access: 18.02.2026).

10. DeLong L. N., Mir R. F., Fleuriot J. D. Neurosymbolic AI for reasoning over knowledge graphs: a survey. *IEEE transactions on neural networks and learning systems*. 2024. P. 1–21. URL: https://doi.org/10.1109/tnnls.2024.3420218 (date of access: 25.02.2026).

11. Wan Z., Liu C., Yang H. Towards Cognitive AI Systems: A Survey and Prospective on Neuro-Symbolic AI. 2024. URL: https://arxiv.org/abs/2401.01040 (date of access: 03.03.2026).

12. Neuro-Symbolic AI: foundations and frontiers / S. R. Nalawade et al. *Journal of intelligent decision technologies and applications*. 2025. Vol. 2, no. 3. P. 1–7. URL: https://doi.org/10.46610/joidta.2025.v02i03.001 (date of access: 07.03.2026).

13. Ben Chaabene N. E. H., Hammami H. Neuro-symbolic synergy in education: a survey of LLM-knowledge graph integration for explainable reasoning and emotion-aware student support. *Smart learning environments*. 2026. Vol. 13, no. 1. URL: https://doi.org/10.1186/s40561-025-00423-z (date of access: 12.03.2026).

14. Tiddi I., Schlobach S. Knowledge graphs as tools for explainable machine learning: a survey. *Artificial intelligence*. 2022. Vol. 302. P. 103627. URL: https://doi.org/10.1016/j.artint.2021.103627 (date of access: 21.03.2026).

15. Awolesi Abolanle Ogunboyo. Neuro-Symbolic generative AI for explainable reasoning. *International journal of science and research archive*. 2025. Vol. 16, no. 1. P. 121–125. URL: https://doi.org/10.30574/ijsra.2025.16.1.2019 (date of access: 30.03.2026).

16. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. *Advances in Neural*

Information Processing Systems. 2017. Vol. 30. P. 5998–6008. URL: https://arxiv.org/abs/1706.03762 (date of access: 28.01.2026).

17. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 9459–9474. URL: https://arxiv.org/abs/2005.11401 (date of access: 15.02.2026).

18. Survey of hallucination in natural language generation / Z. Ji et al. *ACM computing surveys*. 2022. URL: https://doi.org/10.1145/3571730 (date of access: 20.02.2026).

19. Unifying large language models and knowledge graphs: a roadmap / S. Pan et al. *IEEE transactions on knowledge and data engineering*. 2024. P. 1–20. URL: https://doi.org/10.1109/tkde.2024.3352100 (date of access: 05.03.2026).

20. SWRL: A semantic web rule language combining OWL and ruleml / I. Horrocks et al. *W3C Member Submission*. URL: https://www.w3.org/Submission/SWRL/ (date of access: 18.03.2026).