УДК 004.8:004.724
DOI: https://doi.org/ 10.17721/3041-2323.2025.266-305

Ігор СІНІЦИН, д-р.тех. наук, проф.,
чл.-кор. НАН України
ORCID ID: 0000-0002-4120-0784
e-mail: ips@nas.gov.ua
Інститут програмних систем НАН України, Київ, Україна

Юлія РОГУШИНА, канд.фіз.-мат.наук, с.н.с., доц.
ORCID ID: 0000-0001-7958-2557
e-mail: ladamandraka2010@gmail.com
Інститут програмних систем НАН України, Київ, Україна

Костянтин ЮРЧЕНКО, асп., мол. наук. співроб.
ORCID ID: 0000-0003-3150-0027
e-mail:urchikak8@gmail.com
Інститут програмних систем НАН України, Київ, Україна

Юрій БОВА, асп., інж.
ORCID ID: 0009-0008-9797-5213
e-mail: bova1997@gmail.com
Інститут програмних систем НАН України, Київ, Україна

ІНТЕГРАЦІЯ СЕМАНТИЧНИХ WIKI ТЕХНОЛОГІЙ З ВЕЛИКИМИ МОВНИМИ МОДЕЛЯМИ ЯК ТЕХНОЛОГІЧНА ОСНОВА ЗДОБУТТЯ ДОСВІДУ З ПРИРОДНОМОВНИХ ДОКУМЕНТІВ

Результатами дослідження, що представлені у статті, є формалізація класу задач, ву яких застосовується здобуття досвіду з природномовних документів. На основі цього сформульовано набір вимог до технологічної платформи, що має забезпечити розв'язання задач цього класу, визначено базові функціональні модулі та послідовність обробки інформації в системі. Проаналізовано використання LLM для аналізу природномових документів, розглянуто критерії оцінювання їх ефективності та напрями вдосконалення їх роботи. Щоб обґрунтувати запропонований підхід, проаналізовано переваги інтеграції семантичних технологій (на прикладі Semantic MediaWiki) з великими мовними моделями, що виступають інструментом здобуття знань на різних етапах обробки документів. Розглянуті приклади демонструють значні

відмінності між задачами проаналізованого класу та необхідність адаптації запропонованої платформи до їх специфіки.

К л ю ч о в і с л о в а : ***великі мовні моделі, семантичні вікітехнології, здобуття досвіду з документів, природномовні документи.***

Вступ

Сучасну ера цифрової трансформації пришвидшує накопичення даних з усіх областей діяльності людина та суспільства та викликає появу великої кількості різноманітних електронних документів, які містять природномовну інформацію різного рівня структурування. Одним з головних викликів стає ефективність інтелектуального аналізу цих масивів неструктурованих даних, що містяться в електронних документах. Ціллю такого аналізу є здобуття відомостей, які потрібні користувачам для задоволення їх інформаційних потреб, та перетворення здобутої інформації у форму, що забезпечує можливість її зручного та ефективного використання. Але такий аналіз потребує засобів, які забезпечать перехід від даних до знань та нададуть можливість використовувати той досвід, що відображений в оброблених даних, доповнюючи його наявними джерелами знань та підтримуючи співпрацю з експертами.

Саме наявність таких масивів інформації та потреба в її аналізі є передумовою нашого дослідження, яке спрямоване на здобуття досвіду з документів, в яких основну частину контенту складає природномовний текст та зв’язки різного рівня формалізованості між фрагментами такого тексту, та визначає його актуальність. Інтегруючи семантичні технології, що дозволяють явно застосовувати формалізовані знання певного домену, з елементами генеративного штучного інтелекту як інструмента лінгвістичного аналізу, ми прагнемо розробити моделі та методи, які дозволяють контролювати правильність знань, що здобуваються з документів, та правил їх інтерпретації для побудови рекомендацій на основі досвіду.

Цифрова трансформація суспільства в цілому та окремих сфер його діяльності викликала появу великої кількості різноманітних електронних документів, які містять природномовну інформацію різного рівня структурування та потребують аналізу, щоб здобути

з них відомості, які потрібні користувачам, та подати ці відомості у формі, що відповідає специфічним потребам користувачів. Цей клас задач ми розглядаємо як здобуття досвіду з документів (ЗДД), в яких основну частину контенту складає природномовний текст та зв'язки різного рівня формалізованості між фрагментами такого тексту.

Прикладами задач ЗДД є:

- побудова структурованих специфікацій обраних об'єктів (інформаційних систем, організацій тощо) з метою їх атестації та сертифікації з використанням аналізу нормативних документів та стандартів;
- побудова рекомендацій щодо подальших дій на основі узагальнення прикладів досвіду, що описаний у наданих документах (звітах, результатах тестування) відповідно до правил та вимог предметної області (ПрО);
- інтелектуалізація документообігу наукових та освітніх установ, що базуються на поєднанні знань специфічних ПрО з державними та міжнародними стандартами подання інформації (такими, як принципи FAIR для відкритої науки), що забезпечує автоматизовану генерацію аналітичних звітів та довідок у заданому форматі.

У контексті семантичного аналізу природномовних текстів виникає потреба у побудові системи, яка здатна здійснювати трансформацію вхідних документів (описів прецедентів, нормативних актів, запитів користувача) у результуючі документи (структуровані відповіді, формалізовані інструкції, семантичні шаблони). Така трансформація не є тривіальною, оскільки потребує:

- виявлення релевантних фрагментів у вхідному тексті;
- побудови семантичних відповідностей між фрагментами;
- застосування продукційних правил, які моделюють логіку перетворення;
- оцінки якості та релевантності результуючого документа.

Складність таких задач пояснюється не тільки необхідністю обробки великих обсягів неструктурованої природномовної (ПМ) інформації, але й відсутністю чітко визначених вимог до результуючих документів та формальних правил їх побудови: в

більшості випадків ці знання потрібно здобувати шляхом аналізу прикладів результуючих документів (а набір таких документів зазвичай не є достатнім для знаходження однозначних рішень) та тієї нормативної документації (правил, стандартів, постанов тощо) і пертинентних джерел, які надає користувач або яка наявна у відкритому інформаційному просторі.

Методи, що застосовані в дослідженні – онтологічний аналіз, семантичне моделювання знань домену, машинне навчання, дедуктивне та традуктивне виведення, лінгвістичний аналіз із застосуванням великих мовних моделей.

Специфіка задачі здобуття досвіду з природномовних документів

Виникає питання – чому ми поєднуємо в одну групу такі досить різні задачі й чому намагаємося побудувати узагальнену платформу для їх розв'язання? Основним критерієм, що об'єднує ці приклади ЗДД, є орієнтація на результат, тобто вибір методів аналізу та процедур обробки вхідної інформації значною мірою залежать від вимог користувача щодо результуючого інформаційного об'єкта, що створюється в процесі обробки. Саме ці вимоги визначають, яку саме інформацію потрібно здобути з наявних документів.

З точки зору аналізу інформації, весь набір розглянутих вище проблем аналізу ПМ-документів може розглядатися як *задача трансформації знань*, що містяться у гетерогенному наборі природномовних документів, у *складний інформаційний об'єкт* (СІО), який складається з елементів контенту вхідних документів (безпосередньо або відповідним чином перетворених). Структура цього СІО відображена у прикладах документів, які надає користувач, та у нормативних документах ПрО, які теж надаються користувачем або доступні у Web. СІО складається з набору більш простих інформаційних об'єктів (ІО), що пов'язані різними відношеннями. Значення окремих елементів цього СІО визначаються на основі аналізу властивостей тих ІО, які описані у вхідних даних, та за тими правилами, що можуть бути здобуті з цих документів.

Після цього у багатьох випадках виникає додаткова підзадача – перетворення СІО на *результуючий документ* (РД), в якому

інформація з СІО представлена у певному формалізованому представленні відповідно до вимог ПрО (наприклад, форматування тексту за певними правилами).

Обов'язковим елементом ЗДД є наявність прикладів побудови рішення, що пов'язують РД з одним або кількома вхідними документами (ВД), що описують прецеденти, яким відповідає цей РД. ПМ-описи правил перетворення знань, що містяться у вхідних нормативних документах, та характеристики об'єктів в описах прецедентів можуть бути нечіткими та припускати неоднозначну інтерпретацію. Таким чином, для розв'язання задачі ЗДД виникає потреба як у засобах лінгвістичного аналізу (з відповідними базами знань), так і у засобах семантичного аналізу, які на основі формалізованих знань щодо ПрО генерують характеристики результуючих СІО.

Специфіка запропонованого підходу пов'язана з інтеграцією методів традуктивного навчання за прикладами («від окремого до окремого») з методами дедуктивного виведення нових знань з наявних (використання правил, що здобуваються з нормативної документації ПрО): якщо узагальнення наявних прикладів припускає генерацію кількох різних РД, то для розв'язання такої неоднозначності можуть використовуватися знання ПрО, що представлені в різних видах: явно (зовнішні бази знань, онтології, тезауруси) або неявно (нормативні ПМ-документи, інструкції, стандарти тощо). Неоднозначна інтерпретація наданих відомостей може бути викликана як неоднозначністю ПМ, так і недостатнім набором наданих даних. Обидві проблеми можуть бути вирішена залученням зовнішніх баз знань (БЗ), що пертинентні тій задачі, що цікавить користувача.

У процесі аналізу потрібно спочатку визначити загальну структуру РД – його основні елементи, їх порядок та форму представлення, та формально зафіксувати цю структуру (за допомогою шаблонів, онтологічних моделей, схем метаданих тощо), а потім для кожного елемента цієї структури визначити джерела інформації (в даних користувача) та правила їх перетворення у значення структурних складових РД.

Крім того, доцільно виокремити з масиву доступних документів ті відомості, які необхідні для побудови СІО

конкретного типу, тобто необхідно проаналізувати великий обсяг інформації, щоб визначити релевантні та актуальні правила.

Виокремимо основні види інформації, що обробляються в ЗДД:

- $ID = \{id_i\}, i = \overline{1, p}$ – непорожній набір ВД;
- $RD = \{rd_i\}, i = \overline{1, k}$ – приклади РД, такий що $\forall rd_i \in RD, i = \overline{1, k},$ існує не менше одного прецеденту у ВД, $\exists id_j \in ID, j = \overline{1, k},$ таких, що $F(\{id_m \in ID, m = \overline{1, k_{rd_i}}\} = rd_i \in ID$;
- $ND = \{nd_i\}, i = \overline{0, q}$ – набір нормативних документів, інструкцій та зовнішніх баз знань, що може містити як структуровані, так і неструктуровані знання: $ND = ND_{struct} \cup ND_{non\_struct}$;
- тезаурус ЗДД $SP = \{sp_i\}, i = \overline{1, w}$ – непорожній набір семантичних властивостей ВД та РД, які використовуються для структурування контенту;
- $VSP = \{vsp_i\}, i = \overline{1, w_{sp_i}}, i = \overline{1, w}$ – непорожній набір значень, що припустимі для семантичних властивостей ВД та РД;
- онтологічна модель ЗДД, яка будується на основі SP та VSP.

На верхньому рівні абстракції всі вхідні документи можна класифікувати залежно від того, які саме відомості може здобувати з них система в процесі аналізу для генерації результуючого документа:

– описи профілів об'єктів ПрО, які явно або неявно визначають параметри об'єктів різних типів, їх відношення та припустимі значення, а також можуть характеризувати їх семантику або призначення у системі – джерело наборів властивостей;

– описи конкретних екземплярів об'єктів різних типів (ця інформація може надаватися окремо або входити до складу оцінених прецедентів) – джерело значень властивостей;

– описи оцінених прецедентів (окремих подій в ПрО, які пов'язують набори значень параметрів різних наборів об'єктів з певним рішенням, оцінкою або дією);

– описи знань щодо ПрО (нормативні документи, постанови, стандарти, інструкції) – джерела правил перетворення умов задачі на результат;

– приклади результатів (табл. 1).

Таблиця 1

Класифікація документів та їх характеристики для задачі здобуття досвіду

Клас	Кількість	Обсяг	Структура	Приклад
Профіль	Середня (10-100)	Малий	Чітка, проста	Профіль організації, підрозділу, особи
Екземпляр	Велика	Малий	Чітка, проста	Опис конкретного пристрою, системи, підрозділу
Прецедент	Дуже велика	Малий	Більш неточна, середньої складності, типова, містить екземпляри об'єктів	Звіт про подію, огляд використання системи
Норматив	Невелика	Дуже великий	Дуже складна, формалізо-вана	Стандарт, постанова, інструкція
Приклад результату	Середня (залежно від складності задачі і рівня формалізації вимог)	Малий або середній	Різної складності (залежно від задачі), формалізо-вана	Результуючий документ, що згенеровано користувачем для наданого прикладу

Продовження таблиці 1

Клас	Кількість	Обсяг	Структура	Приклад
Вимоги користувача	мала	Малий або середній	Чітка, частково формалізо-вана	Набір вимог та перелік обов'язкових та бажаних елементів результуючого документу

Результати роботи системи теж можна поділити на кілька класів:

– пошук прецедентів та аналогів;

– специфікація наявної ситуації або об'єкта – на основі інтеграції фактів та правил, що здобуті з різних документів;

– прогноз щодо оцінки певної ситуації, наслідків дій або функціювання специфікованого об'єкта – на основі аналізу прецедентів та застосування загальних правил з нормативів;

– рекомендація – набір пропозицій щодо того, як доцільно змінити властивості ситуації або об'єкта, щоб збільшити ймовірність позитивного прогнозу або зменшити ймовірність негативного.

Відповідно до такої класифікації, випливає потреба у поділі семантичних властивостей об'єктів, які використовуються у семантичній розмітці документів на такі групи:

– незмінні (значення зафіксоване та не змінюється з часом), наприклад, назва підрозділу, потужність певного пристрою або дата події;

– динамічні, незалежні від дій користувача (значення можуть змінюватися з часом, але користувач не може впливати безпосередньо на ці зміни), наприклад, погодні умови або параметри обладнання;

– динамічні залежні (користувач може явно змінювати значення своїми діями), наприклад, параметри роботи обладнання, розташування наявного персоналу, порядок виконання доступних дій.

У рекомендаціях можна включати лише поради щодо змін властивостей третього типу і за умов, що на такі зміни може впливати той користувач, який отримує рекомендації.

Доцільно описувати ці характеристики семантичних властивостей більш детально, пояснюючи, хто чи що саме може впливати на їх значення та яким чином.

З формальної точки зору, ЗВД потребує виконання таких основних етапів аналізу:

1. Визначення набору vsp_i для кожного ВД та РД;

2. Побудови функції F для перетворення довільного набору vsp_i для ВД на набір vsp_i для РД;

3. Перетворення набору vsp_i для РД на сам РД у заданому форматі.

Тому у розв'язанні задач такого типу можна виокремити наступні основні етапи:

– аналіз структури та семантики РД;

– аналіз структури та семантики прецедентів, що відповідають наданим РД;

– визначення правил і залежностей між прецедентами та РД;

– здобуття з нормативних документів ПрО правил, що дозволяють розв'язувати неоднозначності у залежностях між прецедентами та РД;

– побудова онтологічної моделі ПрО, яка містить складові СІО, відношення між ними та обмеження щодо можливих способів їх поєднання;

Якщо виконувати такий аналіз вручну, то це потребує дуже багато часу та залучення експертів, що мають відповідні знання в обраній ПрО. Крім того, динамічність інформаційного середовища може призвести до того, що згенеровані результати стануть неактуальними ще до кінця їх підготовки – наприклад, використовуються вже застарілі стандарти або документи, що втратили чинність, не враховуючи наявні нові правила ПрО.

Тому виникає потреба у створенні автоматизованих засобів обробки ПМ-документів, що можуть швидко та якісно виконувати аналіз їх контенту на рівні знань та забезпечити подання результатів аналізу у тій формі, що відповідає вимогам

користувача (тобто генерувати ПМ-документи складної структури з заданим набором елементів).

Сучасний рівень розвитку *генеративного штучного інтелекту* (ГШІ), а саме – *великих мовних моделей* (Large Learning Models, LLM) відповідає багатьом вимогам таких задач, але залишає відкритим питання достовірності отриманих результатів та можливостей пояснення шляхів їх отримання (при обробці документів великого обсягу недостатньо вказати, що для отримання результату були використані певні джерела, а замість цього потрібно явно вказувати, які саме елементи контенту були інтерпретовані для створення певних елементів результуючого документу).

У сучасному інформаційному середовищі, насиченому неструктурованими природномовними текстами, виникає потреба у побудові систем, здатних здійснювати семантичний аналіз таких даних з метою їх структуризації, інтерпретації та інтеграції у формалізовані моделі знань.

Особливої актуальності набуває ця задача у вікісередовищах, де тексти мають високий рівень варіативності, контекстної залежності та неоднозначності. Використання великих мовних моделей (LLM) відкриває нові можливості для автоматизованої обробки таких текстів, однак постає низка викликів, пов’язаних із забезпеченням достовірності, узгодженості та інтерпретованості результатів.

Однією з ключових проблем є традуктивний характер генерації знань у LLM – тобто перехід від окремих прикладів до нових окремих випадків без формального узагальнення. Це створює ризик побудови семантичних представлень, що не відповідають очікуванням користувача або нормативним вимогам. Відсутність явної формалізації знань у вхідних даних ускладнює процес перевірки результатів та потребує створення проміжних семантичних шарів, які можуть бути використані як інтерфейс між LLM та системою оцінювання.

Крім того, інструменти на основі LLM значно ефективніше обробляють англомовні тексти, ніж документи українською мовою або ж мультилінгвістичні.

Аналіз досліджень з використання LLM для аналізу документів

Сучасні досягнення в машинному навчанні, особливо глибокому навчанні, розширили можливості штучного інтелекту, забезпечивши якісне розпізнавання зображень та мовлення, обробку природномовних документів. Зростання великих даних та підвищення обчислювальної потужності завдяки графічним процесорам ще більше прискорили дослідження та застосування штучного інтелекту.

LLM – це моделі машинного навчання, які базуються на нейронних мережах та використовують сховища даних великого обсягу для завдань, що стосуються аналізу природної мови. Основою ефективного застосування LLM є критерії відбору даних, на яких модель навчається. В LLM кожне слово ПМ представлене як точка у багатовимірному просторі, якій відповідає вектор фіксованої довжини, який, в свою чергу, кодує семантичні властивості цього слова та його відношення з іншими словами. Близькі за змістом поняття відображаються близькими точками з подібними векторами, і саме ця подібність є основою для подальшого аналізу текстів. Обсяг текстів для навчання має бути достатньо великим, щоб дозволити моделі отримати достатній словниковий запас та зрозуміти значення й семантичні відношення між поняттями, яким відповідають слова та словосполучення. Але використання занадто великого обсягу інформації, яка нерелевантна задачам LLM, може призвести до зниження якості роботи моделі, особливо якщо відомості в текстах недостовірні, тенденційні, суперечливі та не актуальні.

Ці системи розробляються для обробки інформації та прийняття рішень або прогнозів на основі наданих їм даних, тому якість їх роботи залежить не тільки від алгоритмів обробки, але й від обсягу та якості наданих для їх навчання даних.

У розвитку LLM можна виокремити кілька поколінь (Wu et al., 2024), які різняться продуктивністю та глибиною аналізу. Розробкою LLM займаються в різних країнах світу та великих корпораціях (таких, як Microsoft, OpenAI та Google), причому багато моделей надаються з відкритим кодом.

Зараз розробляються LLM четвертого покоління, для яких характерно:

– покращенні можливості генерації текстів для створення більш точних відповідей на питання користувачів;

– виявлення складних семантичних відношень між словами та покращене розуміння контексту;

– більш досконале налаштування персональних параметрів роботи.

Для оцінки ефективності та якості мовних моделей використовуються різні метрики (Chang et al., 2024) (як для загальних задач обробки текстів, так і для спеціалізованих задач, таких як створення програмного забезпечення, побудова чат-ботів, математичні задачі та логічне виведення, Human-in-the-loop (можливість контролю користувачем) тощо). Найбільш вживаними з загальних метрик є:

– *точність* (Accuracy) – частка правильних відповідей моделі на тестовому наборі даних: Exact match, Quasi-exact match, F1 score, ROUGE score (Lin, 2004);

– *калібрування* (Calibrations) – можливість додаткових налаштувань: Expected calibration error, Area under the curve (Geifman, & El-Yaniv, 2017);

– *чесність* (Fairness) – відсутність неправдивих, вигаданих тверджень та посилань: Demographic parity difference, Equalized odds difference (Hardt, Price, & Srebro, 2016);

– *робастість* (Robustness) – стійкість до малих змін параметрів об'єкта: Attack success rate, Performance drop rate (Zhu et al., 2023).

Крім того, важливими параметрами є час виконання – наскільки швидко модель обробляє запити, та обсяг обчислювальних ресурсів, що необхідні для функціювання LLM.

Слід зазначити, що серед найбільш успішних LLM важко надати перевагу якійсь одній розробці за всіма цими параметрами, тому що кожна нова версія цих систем пропонує певні переваги перед іншими. При цьому програмні реалізації різняться вимогами до обладнання, швидкодією, налаштованістю на певні природні мови. Крім того, деякі LLM більш орієнтовані на певні функції (такі як переклад, генерація схем, зображень,

програмного коду тощо), і це робить вибір LLM для конкретного завдання складною багатокритеріальною задачею, яка потребує як теоретичного аналізу параметрів, так і практичної перевірки.

Донавчання LLM – сукупність методів адаптації попередньо навчених базових моделей до конкретних доменів, задач або поведінкових цілей. Потреба в ньому виникає тоді, коли конкретна задача потребує додаткових знань, специфічної термінології або використання недостатньо повно вивченої ПМ. Вибір методу залежить від: доступних даних, обчислювального бюджету, обмежень приватності, вимог до розгортання (латентність, пам'ять) і критеріїв вирівнювання/безпеки. Виокремлюють наступні основні підходи у донавчанні:

- *Продовжене вивчення ПрО (Domain Adaptive Pretraining)* здійснюється на великих корпусах нерозмічених текстів конкретної ПрО для зменшення розриву між загальною предтренованою моделлю і розподілом ПМ цільової області (наприклад, медичні статті, юридичні документи, наукові публікації) та підвищення продуктивності, але може викликати «відкат» загальних властивостей моделі (catastrophic drift) та потребує значних ресурсів та якісної фільтрації корпусу текстів (Gururangan et al., 2020);
- *Супервізоване донавчання з орієнтацією на задачу (Task specific supervised fine tuning)* спрямоване на оптимізацію ваг моделі на парах розмічених даних «вхід-вихід», якщо доступні якісні мітки для цільової задачі і є можливість оновлювати більшість параметрів моделі, але потребує багато пам'яті й обчислень (Howard, & Ruder, 2018);
- *Донавчання з інструкцією (Instruction tuning)* здійснюється на наборах «інструкція-приклад відповіді» для підвищення здатності моделі виконувати ПМ-інструкції користувачів, але якість результатів залежить від різноманітності та якості інструкційних пар (Wei et al., 2021);
- *RLHF (Reinforcement Learning from Human Feedback) та оптимізація на основі преференцій* дозволяє вирівняти модель під людські оцінки корисності/безпеки й зменшує небажані результати, але потребує багатоступеневої обробки, яка включає збір людських преференцій, які порівнюють пари відповідей;

навчання моделі на цих преференціях; та оптимізацію основної моделі для максимізації винагороди (Hatgis-Kessell et al., 2025);

- *PEFT* (*Parameter-efficient fine-tuning*) залишає основну модель більшою мірою фіксованою, а для адаптації додаються або оновлюються невеликі підмножини параметрів, такі як Prefix/Prompt tuning – оптимізація контекстних векторів або префіксів замість ваг моделі та LoRA – низькорангова адаптація шляхом підставлення тренованих матриць в оновлення ваг;
- *Гібридні конвеєри*, що поєднують супервізоване інструкційне донавчання (SFT), RLHF для остаточного вирівнювання та RAG;), найчастіше застосовують для побудови чатботів та фактологічно чутливих систем
- *Продовжувальне навчання (Continual learning)* підтримує оновлення моделі на основі потоку нових даних, але вимагає складних механізмів контролю забування та збереження прикладів зразків.

Вибір між донавчанням відкритої моделі через API й донавчанням локальної копії моделі визначається двома групами факторів: вимогами щодо приватності, регуляції й контролю даних; та техніко-організаційними обмеженнями, такими як обчислювальні ресурси, затрати, можливість збереження інформації. Для багатьох ситуацій розроблено типові підходи, такі як SFT, instruction tuning, RLHF, PEFT, RAG, domain pretraining і приклади їх впровадження.

Деякі задачі ЗДД, для яких доцільно застосовувати LLM, потребують високого рівня захищеності та конфіденційності, і тому документи, що аналізуються, не можна надсилати до доступних через Web LLM. Але для цього можуть бути використані локальні версії LLM, що запобігає витоку даних у відкритий доступ (Слюсар, 2024), хоча треба враховувати, що такі LLM мають значно меншу продуктивність та повертають значно менш надійні результати з великою кількістю «галюцинацій», тобто якість аналізу інформації значно нижча (хоча це значно залежить від розміру моделі та її налаштованості на певну задачу та ПМ).

Коли обирають відкриті моделі LLM (через постачальника API – OpenAI, Azure OpenAI, Anthropic та ін.), то для донавчання

використовують такі сценарії, як supervised fine tuning (на невеликому наборі розмічених пар), вирівнювання (alignment) для зміни стилю або політики відгуку, функціональна інтеграція і спеціальні API (наприклад, fine tuning або демонстрації через API). Обмеження та ризики використання відкритих моделей – це висока вартість та відсутність приватності й доступу до внутрішніх ваг моделі.

LLM розгортається локально (self hosted) або в приватному хмарному середовищі, якщо потрібно зберігати чутливі дані локально, гарантувати відповідність законодавству та захисту персональних даних (наприклад медичних); якщо потрібно підтримувати багато різних адаптацій для кожного завдання; потрібне навчання під власну політику відповідей та внутрішні експертні преференції (наприклад, для фінансової установи); поєднання LLM із локальною базою знань (векторним сховищем) для фактологічності й миттєвого оновлення (наприклад, науково-дослідна установа інтегрує LLM із локальним репозитарієм публікацій з внутрішніх джерел). Якщо дані підлягають обмеженням, то локальне донавчання – майже обов'язкове.

Таким чином, аналіз досліджень та прикладів впровадження показує, що відкриті API моделі зручні для швидких, маловитратних адаптацій, задач з нестрогою приватністю та для прототипування, а локальні моделі виправдані, коли потрібен повний контроль над даними, складні доменні адаптації або застосування в регульованих ПрО. Донавчання локальних LLM застосовують для адаптації моделі до вузького предметного домену; персоналізації поведінки в межах організації; підвищення факт-чекінгу та обґрунтованості відповідей через локальні знання (RAG); забезпечення вимог приватності й регуляторної відповідності. Початкові параметри для локального донавчання визначаються такими основними факторами, як розмір моделі (від 7B до 70B+ параметрів), доступна обчислювальна інфраструктура (кількість GPU та їх пам'ять), обсяг і якість корпусу ПрО. Метрики, за якими оцінюється ефективність локального донавчання , поділяють на: автоматичні (task oriented та retrieval oriented); експертні оцінки корисності,

правдивості, стилю, безпечності; порівняння з результатами без донавчання (Ablation studies) та моніторинг після розгортання.

Retrieval Augmented Generation (RAG) – це парадигма поєднання зовнішніх систем інформаційного пошуку з мовними моделями, в якій генерація тексту LLM орієнтована на контекст, який здобувається з індексованого корпусу документів під час запиту (inference time). Мета RAG – компенсувати обмеженість параметричної пам'яті LLM (статичні знання, «knowledge cutoff») і таким чином підвищити точність, фактологічність та оновлюваність відповідей, зменшуючи частоту «галюцинацій» (вигаданих фактичних тверджень) в генерації тексту. У класичній реалізації RAG містить три компоненти: формувальник запиту (retriever), корпус документів з механізмом ранжування та генератор (conditional LM), який отримує відповіді на запит (retrieved passages) і з них генерує результат (Lewis et al., 2020).

Тому доцільно поєднати використання LLM із семантичною розміткою документів термінами ПрО: це дозволить виокремити змістовні елементи документів для подальшої обробки, а експертам ПрО спростить процес контролю процедури створення результуючих документів.

Треба зауважити, що через неоднозначність природної мови, інтерпретування формулювань з документів може потребувати узгодження, додаткової перевірки та консультацій з експертами ПрО. Тому обробка інформації в такій системі має складатися з кількох етапів, з можливістю повернення до попередніх етапів для внесення уточнень. Для того, щоб забезпечити гнучкість та адаптивність такої системи, доцільно розділити її функції на окремі модулі, а зв'язок між цими модулями організувати за принципами сервіс-орієнтованого програмування: модулі обмінюються даними відповідно до чітко визначених правил та протоколів, а зміни в одному з модулів не потребують переналаштування інших модулів (потреба у змінах може бути викликана вимогами до масштабування, безпеки, якості та швидкості обчислень, змінами умов використання програмного забезпечення тощо).

Безпосереднє використання LLM для лінгвістичного аналізу вхідних даних ЗДД, без формального визначення тих елементів,

які потрібні для побудови результату, не надасть якісні результати, що придатні для подальшого аналізу та верифікації. Користувач не отримає повну інформацію про те, які саме відомості з вхідних документів були використані для генерації результуючого CIO. Це пов'язано не з обмеженими властивостями LLM, а з відсутністю формалізованого набору понять ПрО, в яких LLM має сформулювати ці правила.

Огляд досліджень з використання LLM для семантичної розмітки

Зараз проблема поєднання великих мовних моделей (LLM) із семантичною розміткою досить слабо відображена у наукових публікаціях, але деякі роботи присвячені окремим аспектам цій задачі. Наприклад, в (Musumeci et al., 2024) автори досліджують проблему автоматизації створення напівструктурованих документів у сфері публічного адміністрування, де шаблони не завжди охоплюють усю варіативність структури. Система складається з набору агентів, кожен з яких відповідає за окрему фазу генерації документа. Рольова інструкція для LLM розробляється як частина промпту та застосовує семантичні відомості про документи для побудови розмітки. Система адаптує шаблони до реальних кейсів, зменшуючи потребу в ручному втручанні.

У статті (Misback, Tatlock, & Tanimoto, 2024) розглядається задача семантичної розмітки документів, які змінюються в процесі редагування. Система автоматично оновлює розмітку при зміні тексту, використовуючи LLM для розпізнавання семантичних зв'язків без явного тегування. У роботі (Wu et al., 2023) розглядається автоматизація генерації оглядів літератури за допомогою LLM, яка анотує статті певної тематики за запропонованою схемою. Це дозволяє подолати обмеження людської обробки великих обсягів наукової інформації. LLM інтегрує дані з понад 1000 статей. Якість автоматичних оглядів не поступається ручним, а рівень галюцинацій знижено до менш ніж 0.5% з 95% довірою. Інше дослідження подібної спрямованості (Haryanto, 2024) описує відкритий інструмент для автоматизованої побудови літературного огляду, що використовує LLM для витягування релевантної інформації та оцінки її

відповідності дослідницькому запиту. Таким чином, проведене дослідження показує, що, незважаючи на наявність досліджень у близьких напрямах (семантичний парсинг, лексико-обмежене декодування, імпорт даних Wikidata тощо), в науковій літературі не зафіксоване явно описані методи та моделі застосування LLM для генерації семантичної розмітки.

Базові вимоги до технологічної платформи здобуття досвіду з документів

Відповідно до призначення та специфіки задачі, технологічна платформа ЗДД має підтримувати наступні функції:

– накопичення та збереження різних типів документів (ВД, РД, нормативних документів тощо), як природномовних, так і мультимедійних;

– створення та збереження схеми бази знань ПрО (онтологічної моделі або іншого виду формального подання знань), з підтримкою її інтеграції як із зовнішніми онтологіями та тезаурусами, так і з накопиченим контентом;

– структурування контенту та генерація семантичної розмітки документів з використанням схеми бази знань ПрО, збереження цих метаданих, а також забезпечення навігації у контенті на основі цієї розмітки;

– засоби пошуку та формалізації правил, що дозволяють пов'язувати структурні елементи РД з інформацією, яка отримується із структурованих ВД;

– засоби пояснення побудови результуючих СІО, які забезпечують доступ як до використаних джерел інформації (конкретних ВД та їх структурних елементів, що виокремлюються засобами семантичної розмітки), так і правил, що були застосовані для їх перетворення (це можуть бути правила, здобуті з нормативних документів, або результати традуктивного виведення на основі аналізу класифікованих ВД);

– засоби надання користувачам результуючої інформації, що забезпечують як форму представлення РД, так і відбір інформації, що релевантна потребам конкретних користувачів.

Вибір моделей, методів та конкретної програмної реалізації кожної з цих функцій залежить від специфіки ПрО та інформаційних потреб користувача, але цей набір дозволяє

визначити основні вимоги до уніфікованої технологічної платформи, яка може адаптуватися до конкретних задач. Потрібно підкреслити, що всі визначені функції пов'язані зі створенням, поданням та аналізом знань, і тому виникає необхідність в уніфікованій системі менеджменту знань для обміну інформацією між окремими модулями.

У спрощеному вигляді прикладна задача ЗДД може бути описана наступним чином: задачі ЗДД передбачають певний порядок етапів створення та застосування знань, але, якщо змінюються середовище або потреби користувачів, то ці етапи обробки можуть повторюватися.

Попереднім етапом аналізу документів ЗДД є створення *тезаурусу* – набору семантичних властивостей, що можуть бути використані для семантичної розмітки. Тезаурус будується ітеративно, за допомогою таких засобів, як імпорт знань з зовнішніх онтологій (за їх наявності), лінгвістичного аналізу стандартів і нормативних документів ПрО та типових прикладів ВД, а також шляхом консультацій з експертами.

Після цього такий тезаурус використовується як основа для семантичної розмітки ВД. Важливо звернути увагу, що в семантичній розмітці потрібно виокремлювати лише ті елементи контенту, що можуть бути корисними для генерації РД, а не всі існуючі структурні та змістовні елементи контенту (повний лінгвістичний аналіз дозволяє побудувати досить складну для обробки онтологічну структуру навіть за одним параграфом ПМ-тексту, але такий рівень деталізації не потрібний для більшості конкретних задач). Якщо в процесі розв'язання задачі тезаурус буде доповнено або змінено, створення семантичної розмітки ВД виконується повторно, але лише для нових або змінених елементів тезаурусу. Такий підхід спрямований на зменшення обчислень та має забезпечити значно більшу швидкість аналізу документів.

На наступному етапі потрібно визначити семантичні зв'язки понять тезаурусу з елементами контенту документів: фрагменти тексту мають бути перетворені на значення відповідних семантичних властивостей документів та визначити відношення між ними. Для такого лінгвістичного аналізу доцільно

застосовувати LLM, яка отримує на вхід цей тезаурус та ті документи, що потребують розмітки.

На наступному етапі LLM обробляє вже семантично структуровані документи, будуючи з них структуру РД, а потім заповнюючи цей шаблон значеннями. Ці етапи теж доцільно контролювати – явно співставляти шаблон, що створює LLM, з наявними зразками, а згенеровані значення – з вимогами користувача. Для навчання системи потрібно залучати експертів ПрО, які можуть вказати на помилки або недоліки у згенерованому РД, і після цього потрібно побудувати промпт, що запитує у LLM джерела та правила, за якими були отримані неправильні результати, використовуючи для цього поняття та відношення з тезаурусу. Проаналізувавши відповіді, можна вдосконалити роботу системи, крім того, це дозволить вирішити проблему довіри до результатів, які створює генеративний штучний інтелект.

На різних етапах можна використовувати як одну LLM, так і різні. Це залежить від складності аналізу, вимог щодо безпеки та захисту даних, орієнтації на обробку певної природної мови чи на окремий клас задач.

Постановка задачі

Пропонується виокремити у задачі ЗДД перетворення набору ПМ-документів на CIO кілька окремих етапів та фіксувати проміжні результати кожного з них за допомогою формалізованої семантичної розмітки. Залежно від вимог ПрО щодо виразної потужності для цього можуть застосовуватися різноманітні синтаксичні представлення – від Semantic MediaWiki до OWL та RDF. Крім того, в багатьох ПрО існують загальноприйняті стандарти подання метаданих та інструменти, які реалізують ці стандарти, які теж можуть бути використані для структурування знань.

Наявність проміжного формалізованого відображення семантики документів дозволить перевіряти, чи правильно LLM інтерпретує знання в природномовних документах – з якими поняттями пов’язує фрагменти тексту, які значення обирає та як саме визначає відношення між ними.

Результати

Технологічна платформа «Лінза»

Розглянемо приклад інструменту для розв'язання задач ЗДД. Технологічна платформа «Лінза» розробляється для виконання задач ЗДД різного рівня складності й призначена для інтелектуальної обробки неструктурованих документів, який дозволяю перетворювати різнорідні набори ПМ-документів складної структури на результуючі СІО відповідно до розгалуженого набору вимог користувача (Сініцин та ін., 2025) Цей функціонал досягається шляхом поєднання семантичних технологій, які дозволяють системі розуміти зміст інформації, з великими мовними моделями, які забезпечують лінгвістичний аналіз контенту документів та генерацію природномовних текстів. Основне призначення системи – трансформувати величезні обсяги різнорідних, часто неструктурованих документів – від стандартів та описів до сирих даних – на інтегровану базу знань, що дозволяє генерувати чіткі, структуровані та зрозумілі звіти та рекомендації.

Система «Лінза» використовує поєднання технологій Semantic MediaWiki та формалізації знань на основі онтологічних моделей з інструментами генеративного штучного інтелекту для аналізу природномовних документів. Ця вікіплатформа платформа робить знання, які LLM здобувають з документів, наочними, зрозумілими та доступними для редагування. Така інтеграція дозволяє об'єднати потужність LLM з гарантованістю та пояснюваністю результатів семантичного аналізу.

«Лінза» пропонує наступні функціональні можливості:

– автоматизація рутини з обробки та структурування первинних даних;

– генерація рекомендацій на основі зразків та аналізу розмічених документів;

– наявність пояснень запропонованих рекомендацій та багатоступеневий контроль: на відміну від «чорного ящика» LLM, система дозволяє оцінювати проміжні результати інтерпретації даних через семантичну розмітку;

– динамічність адаптації при зміні вхідних даних та модифікації потреб користувачів.

Тезаурус – це словник базових термінів та визначень, які дозволяють системі однозначно трактувати інформацію. Онтологічна модель ПрО – це повний, структурований набір понять та взаємозв'язків у обраній сфері. Потрібно зауважити, що ця онтологія може інтегруватися з онтологічною моделлю «Лінза», але ці різні онтології: модель «Лінза» є відносно статичною, тоді як онтлогічні моделі ПрО є специфічними для кожної задачі і можуть значно різнитися за структурою. Це забезпечує формалізацію знань, що є основою для застосування методів машинного навчання, ефективного керування знаннями та, що найважливіше, надання чітких та обґрунтованих пояснень до всіх рекомендацій.

Відповідно до функцій з перетворення документів на СІО, потрібно виокремити функціональні модулі «Лінза» і визначити вимоги до них, і тільки після цього переходити до вибору технологій їх реалізації та засобів взаємодії.

Основні функціональні модулі «Лінза» представлені на рис. 1.

Рис. 1. *Функціональні модулі технологічного середовища «Лінза»*

До складу основних функціональних модулів технологічного середовища «Лінза» входять (див. рис. 1):

1. Модуль генерації тезаурусу ПрО.
2. Модуль збереження документів (репозиторій).
3. Модуль семантичної розмітки документів.
4. Модуль генерації правил побудови РД.
5. Модуль генерації пояснень результатів.
6. Модуль взаємодії з користувачами (чатбот).

Крім того, «Лінза» забезпечує операційне середовище, яке координує всі процеси обробки документів від збору та аналізу до перетворення та перевірки даних та підтримує безпечний і надійний обмін даними між іншими модулями системи, які забезпечують завантаження та попередню обробку документів, їх семантичне анотування, моделювання знань, генерацію відповідей за допомогою LLM та інтеграцію інформації, а також вирішує питання доступу авторизованих користувачів до системи. Чутливі дані користувачів залишаються під контролем і не передаються до зовнішніх LLM, що є неприпустимим з погляду інформаційної безпеки. Також система захищена за допомогою ролей доступу, повного логування всіх дій, захищеного HTTPS з'єднання, що унеможливлює несанкціоновані зміни або видалення даних.

«Лінза» застосовує концепцію «human-in-the-loop», що забезпечує повну прозорість та контроль. Людина-експерт може контролювати та валідувати кожен етап обробки та отримані результати, отримуючи доступ до всієї потрібної інформації, але не може бачити документи інших користувачів. Це не лише підвищує довіру до автоматично створених документів, а й гарантує їхню фактичну правдивість, що є життєво необхідним для прийняття відповідальних рішень.

Модуль генерації тезаурусу ПрО забезпечує засоби для побудови узгодженого словника, який дозволяє визначати відношення між базовими класами ПрО, їх екземплярами та властивостями цих екземплярів. Наявність такого формалізованого опису, який базується на онтологічній моделі ПрО, дозволяє всім учасникам розробки однозначно інтерпретувати семантику даних та обмінюватися інформацією.

Залежно від конкретної задачі, тезаурус може генеруватися безпосередньо за релевантною онтологією, формуватися на основі аналізу документів (з використанням LLM або інших аналітичних інструментів) або вручну створюватися експертами. Найбільш ефективним є поєднання всіх трьох підходів та ітеративне поповнення тезаурусу в процесі більш глибокого дослідження ПрО на основі отриманого досвіду та зворотного зв'язку з користувачами. Результатом роботи модуля є впорядкований набір понять (слів та словосполучень), які можуть застосовуватися в якості тегів семантичної розмітки та як складові схеми метаданих документів. Модуль дозволяє автоматизовано виокремити множину семантичних властивостей із природномовного тексту (наприклад, опису прецеденту, нормативного документа), які можуть бути використані для побудови онтологічних моделей, семантичної розмітки або генерації промптів для LLM. Семантична властивість у цьому контексті – це формалізований фрагмент знання, що описує відношення між сутностями, їхні характеристики або контекстуальні залежності.

Репозиторій – модуль, що забезпечує збереження відомостей про вхідні документи на всіх етапах їх семантичного структурування, аналізу та переробки на елементи знань для результуючого СІО. Передбачається, що основним елементом фіксації результатів обробки буде семантична розмітка документів, що подібна до Semantic MediaWiki. Залежно від специфіки задачі, передбачається створення й збереження інших типів метаданих, що характеризують контент документів. Значеннями властивостей можуть бути як дані (текст, число, інші формати констант), так і посилання на інші об'єкти. Для того, щоб відображати більш складні концепції ПрО, доцільно, крім конструкцій типу [[властивість::значення]], які підтримуються Semantic MediaWiki, застосовувати конструкції довільної арності: [[властивість::значення1+значення2+...значення-n]]. Подібний синтаксис дозволить використовувати (за потреби) середовище Semantic MediaWiki для пошуку та навігації у семантично розмічених документах, хоча значна частина отриманих знань може бути втрачена, якщо не буде зафіксована окремо: для кожної

семантичної властивості з арністю більше 1 потрібно явно описати семантику та тип кожного значення.

Модуль семантичної розмітки документів перетворює ПМ-документи на структуровані, пов'язуючи елементи контенту з поняттями тезаурусу. Для цього доцільно застосовувати LLM, які виокремлюють такі фрагменти. Такі документи можуть бути розмішені у семантизованому вікіресурсі, що надає можливості пошуку, навігації та інтеграції даних. Це спрощує для експертів операції х оцінювання якості розмітки та внесення вдосконалень у роботу LLM.

Модуль генерації правил побудови РД шукає семантичні зв'язки між семантично розміченими елементами ВД та РД, здобуваючи для цього знання з наданої документації та формалізованих джерел знань ПрО. Для цього теж застосовується LLM, а за потребою – більш потужні спеціалізовані інструменти аналізу даних та машинного навчання.

Модуль генерації пояснень результатів надає користувачам доступ до використаних правил та явно показує, яка саме інформація з ВД буда використана для створення окремих елементів РД.

Чатбот надає природномовний інтерфейс та забезпечує персоналізовану взаємодію користувача з системою.

LLM – основний засіб лінгвістичного аналізу, що застосовується в «Лінза». Передбачається, що для різних модулів може застосовуватися «команда» LLM, що різняться функціями, потужністю та захищеністю. Необхідно брати до уваги, що у багатьох задачах ЗДД на різних етапах аналізу виникає потреба в обробці захищених даних (секретна інформація, персональні дані тощо), і для цього можуть бути використані локальні LLM. У виборі ефективної моделі потрібно брати до уваги не тільки її потужність, обсяг та орієнтацію на обробку певної природної мови, але й обсяг контексту, який LLM може обробляти. Для задач «Лінза» останній параметр може бути найбільш критичним.

Тому передбачається наступне рішення – спочатку за допомогою значно більш потужних відкритих LLM, що опрацьовують «навчальні» приклади даних, створюються правила семантичної розмітки, які дозволяють пов'язати

природномовні фрагменти вхідних документів з семантичними властивостями (а надалі – і зі структурними елементами СІО). Такий підхід базується на припущенні, що вхідні документи, які потребуватимуть захищеної обробки, мають подібну структуру та використовують значно обмежену підмножину природної мови.

Відкриті LLM дозволяють спочатку побудувати тезаурус ПрО, а потім для цього тезаурусу визначити набір правил розпізнавання у тексті значень цих семантичних властивостей. Наприклад, для семантиної властивості [[Переміщення::хто+звідки+куди]] будується таблиця, яку надалі потрібно розширити з урахуванням всіх припустимих конструкцій ПМ. У спрощеному вигляді такі правила мають наступну форму, що представлена у табл. 2.

Таблиця 2

Формалізація правил розпізнавання значень семантичних властивостей на основі тезаурусу предметної області

Переміщення	Хто	Звідки	Куди
Їхати, йти, пересуватися, змінити розташування	Об’єкт «Людина», Об’єкт «Група»	Координати, місце, назва	Координати, місце, назва

Наступний крок – локальна LLM отримує ці правила та припустиму кількість документів, що потребують захисту. Після виконання семантичної розмітки її результати зберігаються у репозиторій – і розмічений документ, і окремо – його семантичні метадані.

Наступний етап – аналіз прикладів СІО та визначення його основних структурних елементів. Залежно від вимог задачі щодо захищеності прикладів, цей аналіз можуть виконувати:

– відкрита LLM (найбільш ефективно);
– локальна LLM (найбільш захищено);
– експерт ПрО (найбільш якісно).

На практиці доцільно знаходити поєднання цих підходів.

Наступний етап є найбільш складним – потрібно визначити джерела інформації, з яких здобуваються відомості щодо значень

структурних елементів СІО (це можуть бути значення семантичних властивостей вхідних документів, які відповідають певним вимогам) та алгоритми, за якими визначаються ці результуючі значення. Це можуть бути найпростіші обчислювальні операції (наприклад, «кількість витрачених елементів за добу» визначається як сума значень «кількість витрачених елементів» з усіх документів, де значення «дата» відповідає вимозі результату. Але для багатьох задач такого простого аналізу та пошуку подібних документів (за обраним набором властивостей) недостатньо, а надана множина прикладів результуючих СІО припускає різні інтерпретації таких правил, і тоді здобувати правила перетворень потрібно або з нормативних документів ПрО, або безпосередньо від експертів.

Екосистема технологічної платформи «Лінза»

Концепція екосистеми є потужним інструментом формалізації базових зв'язків (інформаційних та матеріальних) між суб'єктами певної складної системи, тими ресурсами, які вони сумісно використовують у своїй діяльності, та результатами цієї діяльності, якими вони можуть обмінюватися. Цей підхід дозволяє більш точно моделювати системи, в яких суб'єкти спільно використовують певні ресурси та конкурують за доступ до цих ресурсів, а результати діяльності одних суб'єктів можуть розглядатися як ресурси для інших.

Екосистема здобуття досвіду з документів

Одним із наслідків такого підходу, що розділяє задачу ЗДД на набір відносно незалежних підзадач, є необхідність інтеграції цих модулів та діяльності груп їх розробників. Це потребує чіткого та однозначного розуміння базової термінології, формалізації функцій окремих модулів та ролей всіх учасників проєкту: результати роботи одних модулів є вхідними даними для інших, але їх функції можуть змінюватися, розширюватися або передаватися іншим модулям, тобто потрібно формалізувати саму екосистему розробки. Такий підхід до створення програмного забезпечення зараз використовують для створення складних інформаційних систем та для розробки різноманітних застосунків на одній програмній платформі.

У сучасному науковому дискурсі термін «екосистема» досить широко використовується не лише в прямому значенні – як опис спільного проживання певної сукупності живих організмів (екологічних об'єктів) у спільному середовищі, але й як метафора для моделювання різних типів систем, для яких характерний фіксований набір суб'єктів, об'єктів і видів взаємодії між ними під час спільної праці (наприклад, освітній процес, програмне забезпечення тощо) (Geary et al., 2020). Кожен суб'єкт екосистеми має власний набір інтересів і цілей, реалізація яких пов'язана з діяльністю інших суб'єктів (учасників) і з екосистемою в цілому.

Екосистеми програмного забезпечення (Software Ecosystems – SECO) використовуються для моделювання процесу проектування та програмування складних інформаційних систем, що містять гетерогенні компоненти, які розробляються як внутрішніми, так і зовнішніми учасниками, але використовують спільну програмну платформу. Концепція SECO дозволяють формалізувати дії розробників та користувачів програмного забезпечення (Manikas, & Hansen, 2013). На верхньому рівні абстракції такі екосистеми описують взаємини між програмами та сервісами, що виникають в процесі їх створення та використання. Існуючі SECO значно різняться між собою за рівнем складності, технологічними платформами і операційними системами, користувачами, ПрО застосування тощо.

У 2003 році в SECO визначалося як набір програмних продуктів, що розроблені разом в одному технологічному середовищі. Надалі це визначення доповнювалося такими елементами, як спільний ринок програмних інструментів, сервісів та застосунків та засоби взаємодії між розробниками для обміну інформацією, ресурсами та артефактами. Деякі більш специфічні визначення SECO враховують такі аспекти, як програмна інженерія, а також технічні, соціальні та економічні відношення між суб'єтами екосистеми. Крім того, відкритим залишається питання щодо того, до яких компонентів екосистеми – біотичних чи абіотичних – відносити елементи генеративного штучного інтелекту. З одного боку, вони є продуктами діяльності та ресурсами, а з іншого – самі генерують програмний код та інші ресурси.

Таким чином, SECO – це система, що описує взаємодію непорожньої множини учасників на спільній технологічній платформі, в результаті якої виникає набір програмних рішень та сервісів. Доцільно формально фіксувати актуальні вимоги до системи в цілому та до всіх її суб'єктів та об'єктів на основі засобів подання знань, що забезпечують однозначну інтерпретацію. Таким засобом є онтологічний аналіз, який зараз широко застосовується для формалізації розподілених знань у різних ПрО (Gruber, n.d.).

Онтологічна модель SECO дозволяє точно та однозначно описати всі основні компоненти конкретної екосистеми, а також визначити пріоритети у зусиллях з розробки окремих модулів, розподілити задачі та визначити план робіт.

Онтологічна модель екосистеми «Лінза» розрізняє біотичні (суб'єкти) (рис. 2) та абіотичні (об'єкти) компоненти (рис. 3) та визначає змістовні зв'язки між екземплярами відповідних класів за допомогою об'єктних властивостей та властивостей даних (рис. 4).

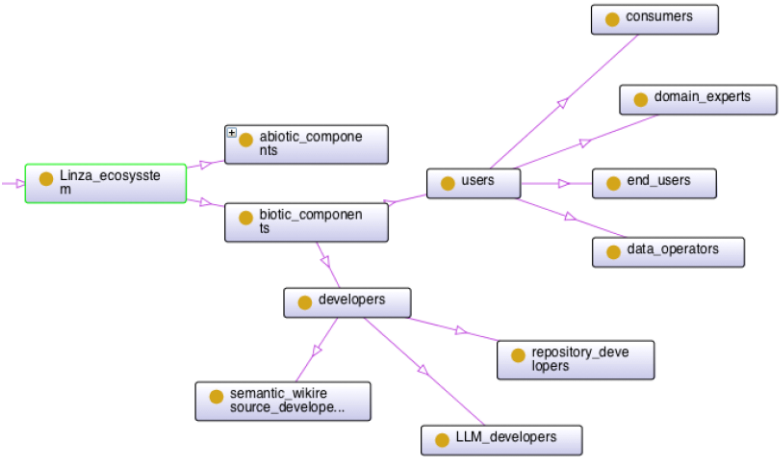

Рис. 2. *Біотичні компоненти онтологічної моделі екосистеми «Лінза»*

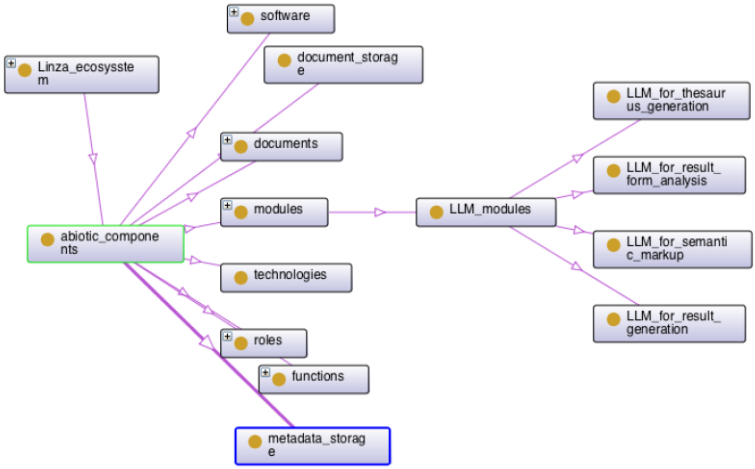

Рис. 3. *Абіотичні компоненти онтологічної моделі* екосистеми «Лінза»

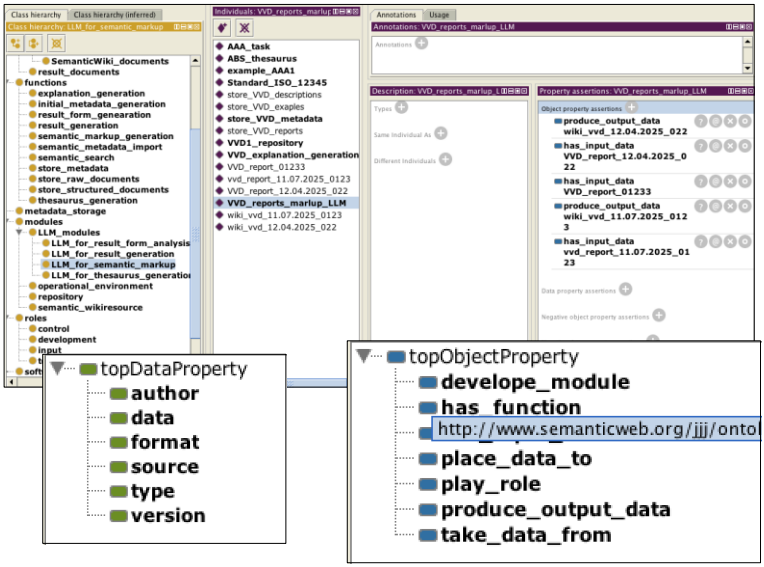

Рис. 4. *Опис екземпляра класу «LLM for semantic markup» онтологічної моделі екосистеми «Лінза» та її властивості*

Це дозволяє перетворити узагальнений опис багаторівневої архітектури «Лінза» (див. рис. 4) на чіткий план її розробки та експлуатації, чітко визначити функції окремих модулів та ролі розробників і користувачів.

Приклади застосування «Лінза» для розв'язання прикладних задач здобуття досвіду з документів

Розглянемо кілька прикладів адаптації технологічної платформи «Лінза» для розв'язання прикладних задач ЗДД різного рівня складності.

Слід відмітити, що «Лінза» орієнтована на результат в тому розумінні, що основою для перетворення вхідної інформації є в першу чергу не явні описи користувача або формальні правила (з інструкцій та нормативів), а аналіз прикладів. Таким чином, йдеться не про дедуктивний аналіз, а про традуктивний (від окремого до окремого). Саме це може спричинити побудову результуючих документів, що не задовольняють вимогам користувача (традуктивна інтерпретація може бути неоднозначною), і тому виникає потреба в ітеративному уточненні побудованих результуючих документів та донавчання. Це призводить до того, що просте використання навіть потужних LLM не є достатнім, тому що виникає проблема саме з оцінювання отриманих результатів. Тому потрібно створювати проміжні формалізовані представлення семантики, які розпізнає LLM з ПМ-документів.

Сама переробка інформації має виконуватися у кілька етапів, і ці етапи можуть повторюватися за необхідністю.

Залежно від задачі, деякі етапи обробки можуть не використовуватися (наприклад, побудова тезаурусу для формалізованої області знань) або, навпаки, повторюватися чи виконуватися різними засобами залежно від складності вхідних та вихідних даних.

Інтелектуальний помічник аспіранта

Задачею є вибір персоніфікованого набору навчальних об'єктів (лекцій, підручників, дистанційних курсів), що відповідають програмі аспірантури, але розширюють її відповідно до обраної теми досліджень, враховують наукові інтереси самого аспіранта та його наукового керівника та

використовують додаткові відомості про здатність аспіранта щодо сприйняття інформації (знання іноземних мов, мотивацію, попередні результати навчання).

Основна проблема полягає в тому, що у наукових дослідженнях досить складно формалізувати актуальну сферу інтересів – її модель є високо спеціалізованою і не може застосовувати без змін вже існуючу онтологічну модель. Але задача спрощується через наступні фактори: практично всі документи є відкритими (крім невеликої підмножини персональних даних, які не є обов'язковими для побудови рекомендацій); відносно невеликий обсяг документів, що обробляються, та кількість користувачів системи; наявність стандартів для профілювання студентів та викладачів; нескладні умови щодо РД.

Вхідні документи – профілі аспірантів, профілі наукових керівників, описи навчальних курсів, неструктуровані описи навчальних ресурсів, прецеденти (інформація про аспірантів, що навчалися раніше, та отримані ними результати); нормативні документи щодо процесу навчання в аспірантурі та вимог до аспірантів. Результуючі документи – впорядковані набори навчальних об'єктів.

Модуль генерації тезаурусу ПрО для цієї задачі є досить складним, він обробляє інформацію про стандарти профілювання здобувачів освіти та про метадані навчальних об'єктів, а також застосовує елементи онтології ПрО та документи вебсайту організації.

Модуль збереження документів (репозиторій) не потребує складних процедур обмеження доступу до документів і може бути реалізований на основі семантичного розширення вікіресурсу. Більшість документів не потребує розмежування початкового вигляду та семантичної розмітки, і тому вся інформація може бути представлена наборами вікісторінок.

Модуль семантичної розмітки документів забезпечує семантичну розмітку для інформації про навчальні об'єкти. Передбачається, що інформація про користувачів відразу вноситься з використанням розмітки.

Модуль генерації правил побудови РД застосовує правила з нормативних документів (наприклад, використання нових ресурсів), а LLM використовує онтологію ПрО для встановлення міри семантичної близькості між цими ресурсами та темою дисертації.

Модуль генерації пояснень результатів надає користувачам значення параметрів та набори ключових слів, за якими здійснюється співставлення навчальних ресурсів з потребами користувача.

Чатбот надає пояснення про процес навчання, знаходить релевантні нормативні документи, а також пропонує інформацію про подібні прецеденти за схемою «аспірант, який мав набір значень параметрів, схожих на ваші, та раніше використав ресурс X, вдало захистив дисертацію». Це досить простий приклад застосування «Лінза», який не тільки надає корисний практичний результат, але й дозволяє порівняти якість використання різних відкритих LLM на різних етапах.

Побудова профілю захищеності інформаційної системи

Цей приклад використання «Лінза» дозволяє за набором прикладів пар «Інформація про користувача – РД» та великої кількості актуальних стандартів створювати документ, що використовується для сертифікації інформаційних систем. Цей приклад значно складніший за попередній через великий обсяг нормативних документів та необхідність побудови РД зі складною формалізованою структурою. Але нормативні документи можуть аналізуватися відкритими LLM (це пришвидшує обробку та підвищує її якість), а правила побудови РД є досить чітко формалізованими і описані на обмеженій підмножині природної мови. Тому з великою ймовірністю у донавчанні LLM не виникне потреба.

Значна складність полягає у видобутті тезаурусу – повного набору параметрів, що впливають на склад РД. Система відношень між ними може виявитися занадто складною для того, щоб відобразити її виразними засобами Semantic MediaWiki, і це джерело інформації буде лише однією зі складових репозиторію. Тому доцільно передбачити більш складну схему опису метаданих та засоби її збереження. Крім того, правила

трансформації ВД у РД можуть бути теж досить складними й не зводитися до набору продукцій. Тому формат їх збереження та засоби представлення користувачам потребують додаткового аналізу.

Пошук та аналіз прецедентів у накопиченому досвіді Військових сил України

У процесі бойових дій військові підрозділи накопичують цінний критично важливий досвід, але він зберігається неструктуровано, тому його складно оперативно перетворювати на чіткі, перевірені рекомендації для інших підрозділів, що сприяють ефективності дій на полі бою. Ця інформація від бойових підрозділів потребує семантичної інтерпретації та уніфікації з нормативними документами, щоб забезпечити інформаційно-аналітичну підтримку процесу виокремлення, формалізації та узагальнення актуальних зміни у кращих практиках прийняття рішень оперативно-тактичного та оперативно-стратегічного рівнів у бойових умовах для їх впровадження у підготовки військ. Така задача ЗДД є найбільш складною з усіх проаналізованих прикладів. Потрібно аналізувати значно більшу кількість ВД (доповідей про реальні події, проаналізовані уроки) з урахуванням їх актуальності, а також відомості про супротивника, директиви Генштабу ЗСУ, бойові статути, стандарти НАТО тощо. Ці документи подані за певними правилами, але підготовлені значно менш формалізовано, можуть містити нечіткі, неповні та некоректні дані (помилково написані слова, різні види скорочень та абревіатур, неоднозначні скорочення тощо). Крім того, ця інформація є закритою, і тому для її аналізу можуть використовуватися лише локальні LLM, що знижує якість цього аналізу. Тезаурус ПрО може будуватися як за цими документами, так і на основі нормативних документів (зокрема показники, що відображають коди спроможностей, та встановлені вимоги до їх реалізації в оборонному плануванні НАТО, характеристики техніки та підрозділів). Також виникає потреба у застосуванні зовнішніх джерел знань (наприклад, про географічні об’єкти, погодні умови). Ситуації та відношення, що відображаються у доповідях, мають досить складну структуру, яка потребує

значних зусиль для співставлення та визначення семантичної подібності. У створення рекомендацій та у пошуку прецедентів потрібно враховувати персональні характеристики того користувача, якому надається відповідь. На різних етапах виникає набір додаткових вимог до LLM, і тому доцільно використовувати донавчання різних локальних LLM відповідно до задач окремих модулів. Необхідно враховувати, що неякісні результати такої системи можуть призвести до критичних наслідків.

Дискусія і висновки

Використання LLM із застосуванням семантичних технологій та менеджменту знань дозволяє підвищити достовірність результатів аналізу та надає понятєвий апарат для пояснення шляхів їх отримання та вдосконалення (на основі елементів онтологічної моделі задачі). Така інтеграція двох суттєво різних напрямів штучного інтелекту забезпечує застосування верифікованих структур знань для генерації складних інформаційних об'єктів. Внаслідок цього користувач отримує пояснення шляхів побудови результатів, які явно відстежують та вказують, які формалізовані елементи контенту та логічні правила були використані, що є важливою складовою довіри до системи (особливо у критичних предметних областях).

У статті запропонована інтеграція генеративного штучного інтелекту з сучасними семантичними технологіями, що забезпечує поєднання потужності LLM з гарантовністю та пояснюваністю результатів у традиційних системах менеджменту знань. Програмна реалізація – інтелектуальна платформа «Лінза», що забезпечує інструменти здобуття та впровадження досвіду, яких зафіксовано у природномовних документах – демонструє переваги та виклики запропонованого підходу.

Основні елементи ШІ, які використовуються у різних модулях системи «Лінза»:

– онтологічне моделювання знань предметної області – основа терміносистеми (тезаурусу) для аналізу змісту документів, інжинірингу LLM-запитів та формулювання рекомендацій;

– семантична розмітка контенту поняттями тезаурусу для побудови база знань на основі документів: спеціалізована LLM

на основі лінгвістичного аналізу природномовних документів пов'язує їх фрагменти з поняттями тезаурусу та перетворює на інтегровану базу знань;

– семантичні вікітехнології – основа семантичного пошуку та верифікації семантичної розмітки;

– інтелектуальний помічник користувача (чат-бот на базі LLM та побудованої бази знань) забезпечує природномовний діалог для пошуку прецедентів, виявлення закономірностей, а також є інструментом уточнення реальних потреб користувачів та вдосконалення тезаурусу;

– алгоритми машинного навчання для побудови дерев рішень і розпізнавання ситуацій на основі навчальних вибірок, класифікації досвіду, пошуку причинно-наслідкових зв'язків та формування послідовностей успішних дій;

– підтримка цілісного та спадкоємного експертно-аналітичного моніторингу практик та пропозицій для виявлення протиріч та взаємовпливів, ризиків та потенційних загроз ефективному використанню.

Завдяки семантичним Wiki робота LLM перестає бути «чорним ящиком»: користувач може контролювати хід міркувань системи та оцінювати коректність інтерпретації контенту документів.

Для зниження ризику «галюцинацій» (помилкова генерація внаслідок недостатнього навчання або перенавченості нейронної моделі щодо певного аспекту) та підвищення точності, ШІ інтегровано з семантичними базами знань і онтологіями. Використовується підхід Retrieval-Augmented Generation (швидке доповнення нейронної моделі додатковими знаннями), що гарантує достовірність прозорість та актуальність рішень.

Список використаних джерел

Сініцин, І. П., Рогушина, Ю. В., & Юрченко, К. Ю. (2025). Інтеграція великих мовних моделей із засобами семантичної обробки як інструмент цифровізації знань. *Проблеми програмування*, (2), 63–76. https://pp.isofts.kiev.ua/index.php/ojs1/article/download/838/889.

Слюсар, В. (2024). Локальні великі мовні моделі для обробки конфіденційної інформації. *Озброєння та Військова Техніка, 4*(44), 79–91. https://doi.org/10.34169/2414-0651.2024.4(44).79-91.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Li, K., ... Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology, 15*(3), 1–45. https://doi.org/10.1145/3641289.

Geary, W. L., Styan, M. C. J., Loo, H. H. T., Thompson, D. G., Davies, P. D. A., & Whitehouse, C. D. (2020). A guide to ecosystem models and their environmental applications. *Nature Ecology & Evolution, 4*(11), 1459–1471.

Geifman, Y., & El-Yaniv, R. (2017). Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems, 30*.

Gruber, T. R. (n.d.). What is an ontology? Retrieved from http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv*. https://arxiv.org/pdf/2004.10964.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems, 29*.

Haryanto, C. Y. (2024). A framework for legal information retrieval using semantic search and fine-tuned large language models. *International Journal of Computer Science and Network Security (IJCSNS), 24*(4), 161–168.

Hatgis-Kessell, S., Knox, W. B., Booth, S., Niekum, S., & Stone, P. (2025). Influencing humans to conform to preference models for RLHF. *arXiv*. https://doi.org/10.48550/arXiv.2501.06416.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv*. https://doi.org/10.48550/arXiv.1801.06146.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., He, H., Chen, Y., Li, A., & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*.
https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Association for Computational Linguistics. https://aclanthology.org/W04-1013.

Manikas, K., & Hansen, K. M. (2013). Software ecosystems. *Journal of Systems and Software, 86*(5), 1294–1306.

Misback, E., Tatlock, Z., & Tanimoto, S. L. (2024). Magic markup: Maintaining document-external markup with an LLM. In *Companion proceedings of the 8th international conference on the art, science, and engineering of programming* (pp. 22–35).

Musumeci, P., D'Agata, P., D'Angelo, S., & Scardapane, S. (2024). LLM based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. *arXiv*.

Wei, J., Bosma, M., Zhao, V. Y., Xu, D., Schuurmans, D., Gelbart, M., Guu, K., Davies, A., Salakhutdinov, R., Le, Q. V., Chi, E. H., Dean, J., & Raffel, C. (2021). Finetuned language models are zero-shot learners. *arXiv*. https://doi.org/10.48550/arXiv.2110.08207.

Wu, S., Ma, X., Luo, D., Li, L., Shi, X., Chang, X., Cui, H., Tang, B., Wu, Y., Liu, Y., & Gong, J. (2023). A survey on large language model for recommendation. *arXiv*. https://doi.org/10.48550/arXiv.2311.13969.

Wu, X., Wu, S.-H., Wu, J., Feng, L., & Tan, K. C. (2024). Evolutionary computation in the era of large language models: Survey and roadmap. *arXiv*. https://doi.org/10.48550/arXiv.2401.10034

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Gong, N. Z., & Zhang, Y. (2023). PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv*. https://doi.org/10.48550/arXiv.2306.04528.

References

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Li, K., ... Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology, 15*(3), 1–45. https://doi.org/10.1145/3641289.

Geary, W. L., Styan, M. C. J., Loo, H. H. T., Thompson, D. G., Davies, P. D. A., & Whitehouse, C. D. (2020). A guide to ecosystem models and their environmental applications. *Nature Ecology & Evolution, 4*(11), 1459–1471.

Geifman, Y., & El-Yaniv, R. (2017). Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems, 30*.

Gruber, T. R. (n.d.). What is an ontology? Retrieved from http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv*. https://arxiv.org/pdf/2004.10964.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems, 29*.

Haryanto, C. Y. (2024). A framework for legal information retrieval using semantic search and fine-tuned large language models. *International Journal of Computer Science and Network Security (IJCSNS), 24*(4), 161–168.

Hatgis-Kessell, S., Knox, W. B., Booth, S., Niekum, S., & Stone, P. (2025). Influencing humans to conform to preference models for RLHF. *arXiv*. https://doi.org/10.48550/arXiv.2501.06416.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv*. https://doi.org/10.48550/arXiv.1801.06146.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., He, H., Chen, Y., Li, A., & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*.
https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). Association for Computational Linguistics. https://aclanthology.org/W04-1013.

Manikas, K., & Hansen, K. M. (2013). Software ecosystems. *Journal of Systems and Software, 86*(5), 1294–1306.

Misback, E., Tatlock, Z., & Tanimoto, S. L. (2024). Magic markup: Maintaining document-external markup with an LLM. In *Companion proceedings of the 8th international conference on the art, science, and engineering of programming* (pp. 22–35).

Musumeci, P., D'Agata, P., D'Angelo, S., & Scardapane, S. (2024). LLM based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. *arXiv*.

Sinitsyn, I. P., Rohushyna, Yu. V., & Yurchenko, K. Yu. (2025). Integration of large language models with semantic processing tools as a knowledge digitalization instrument. *Problemy Programuvannya*, (2), 63–76. https://pp.isofts.kiev.ua/index.php/ojs1/article/download/838/889 [in Ukrainian].

Slyusar, V. (2024). Local large language models for confidential information processing. *Ozbroiennia ta Viiskova Tekhnika, 4*(44), 79–91. https://doi.org/10.34169/2414-0651.2024.4(44).79-91 [in Ukrainian].

Wei, J., Bosma, M., Zhao, V. Y., Xu, D., Schuurmans, D., Gelbart, M., Guu, K., Davies, A., Salakhutdinov, R., Le, Q. V., Chi, E. H., Dean, J., & Raffel, C. (2021). Finetuned language models are zero-shot learners. *arXiv*. https://doi.org/10.48550/arXiv.2110.08207.

Wu, S., Ma, X., Luo, D., Li, L., Shi, X., Chang, X., Cui, H., Tang, B., Wu, Y., Liu, Y., & Gong, J. (2023). A survey on large language model for recommendation. *arXiv*. https://doi.org/10.48550/arXiv.2311.13969.

Wu, X., Wu, S.-H., Wu, J., Feng, L., & Tan, K. C. (2024). Evolutionary computation in the era of large language models: Survey and roadmap. *arXiv*. https://doi.org/10.48550/arXiv.2401.10034

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Gong, N. Z., & Zhang, Y. (2023). PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv*. https://doi.org/10.48550/arXiv.2306.04528.

Отримано редакцією журналу / Received: 27.09.25
Прорецензовано / Revised: 30.09.25
Схвалено до друку / Accepted: 01.10.25

Igor SINITSYN, D. Sc. (Tech.), Prof.; Corr. Memb. of the NAS of Ukraine
ORCID ID: 0000-0002-4120-0784
e-mail: ips@nas.gov.ua
Inst. of Software Systems of the NAS of Ukraine, Kyiv, Ukraine

Julia ROGUSHINA, Ph.D. (Phys. & Math.), Senior Res., Assoc. Prof.
ORCID ID: 0000-0001-7958-2557
e-mail: ladamandraka2010@gmail.com
Inst. of Software Systems of the NAS of Ukraine, Kyiv, Ukraine

Kostiantyn YURCHENKO, Postgrad. Stud., Junior Res. Fellow
ORCID ID: 0000-0003-3150-0027
e-mail: urchikak8@gmail.com
Inst. of Software Systems of the NAS of Ukraine, Kyiv, Ukraine

Yurii BOVA, Postgrad. Stud., Eng.
ORCID ID: 0009-0008-9797-5213
e-mail: bova1997@gmail.com
Inst. of Software Systems of the NAS of Ukraine, Kyiv, Ukraine

INTEGRATION OF SEMANTIC WIKI TECHNOLOGIES WITH LARGE LANGUAGE MODELS AS A TECHNOLOGICAL FOUNDATION FOR EXPERIENCE ACQUISITION FROM NATURAL LANGUAGE DOCUMENTS

The results of the research presented in the article include the formalization of a class of tasks involving the extraction of domain-specific knowledge from natural language documents, as well as the formulation of a set of requirements for a technological platform capable of solving tasks of this class. This analysis supports identification of the basic functional modules of technological platform and the sequence of information processing in it. The use of LLMs for the analysis of natural language documents has been examined, criteria for evaluating their effectiveness and directions for improving their performance have been considered. To justify the proposed approach, we consider the advantages of integrating semantic technologies (on the example of Semantic MediaWiki) with LLMs used act as tools for knowledge acquisition at different stages of document processing. The considered practical examples demonstrate significant differences between tasks of the analyzed class and the necessity of adapting the proposed platform to the specificity of the tasks.

K e y w o r d s: ***Large Language Models, semantic wiki technologies, document knowledge acquisition, natural language documents.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.