

УДК 004.421

<https://doi.org/10.31713/vt3202518>

Ляшко Д. А. [1; ORCID ID: 0009-0005-4125-7368],

аспірант,

Турбал Ю. В. [1; ORCID ID: 0000-0002-5727-5334],

д.т.н., професор,

Климюк Ю. Є. [1; ORCID ID: 0000-0003-3672-8469],

к.т.н., доцент

¹Національний університет водного господарства та природокористування, м. Рівне

HYBRID RAG-QA МЕТОД ДЛЯ ПОБУДОВИ ГРАФА ЗНАНЬ В КОНТЕКСТІ МАЛОРЕСУРСНОЇ МОВИ

У статті представлено комплексний аналіз сучасних методів побудови графів знань (Knowledge Graph Construction, KGC) з неструктурованих текстових даних, що становлять основу для широкого спектра інтелектуальних застосунків – від семантичного пошуку та систем запитань-відповідей до рекомендаційних сервісів та аналітики даних. Робота критично розглядає ключові парадигми KGC, серед яких конвеєрний підхід, спільне вилучення сутностей та відношень, генеративні Seq2Seq-моделі, фреймворки на базі багатокрокового запитання-відповіді (QA) та методи із залученням великих мовних моделей (LLM). Для кожної з парадигм окреслено їхні сильні сторони та фундаментальні обмеження: від каскадного поширення помилок і втрати структурної інформації під час лінеаризації до труднощів генерації повних шаблонів запитань і проблем галюцинацій та фактологічної непослідовності LLM.

Окрему увагу зосереджено на проблематиці побудови графів знань у малоресурсних мовах, де відсутність достатніх корпусів, анотованих даних та інструментів робить використання традиційних підходів малоефективним. У цьому контексті розглянуто сучасні стратегії: використання опорних мов, few-shot та метанавчання, промпт-орієнтовані методи для генерації синтетичних даних, а також архітектури пошуково-доповненої генерації (RAG). Наведено приклади успішного застосування цих стратегій у різних мовних середовищах і галузях, зокрема у біомедичних дослідженнях та для афро-азійських мов.

У статті запропоновано гібридну методологію RAG-QA, яка поєднує сильні сторони обох підходів: фактологічну надійність та «заземлення» інформації, що забезпечує модуль ретривера, із контрольованістю та інтерпретованістю, які надає формулювання задачі у вигляді багатокрокового QA. Такий синергетичний підхід дозволяє чітко узгоджувати побудований граф знань із заданою онтологією,



мінімізувати ризики помилок та підвищувати точність вилучення сутностей і відношень у складних умовах дефіциту даних.

Ключові слова: граф знань; побудова графа знань; вилучення відношень; малоресурсні мови; пошуково-доповнена генерація (RAG); системи запитань-відповідей (QA).

Вступ

Графи знань (Knowledge Graphs, KGs) є потужними структурами даних, що представляють факти про реальний світ у вигляді сутностей (вузлів) та відношень між ними (ребер). Вони слугують основою для широкого спектра інтелектуальних застосунків, включаючи семантичний пошук, системи запитань та відповідей, рекомендаційні системи та аналітику даних. Саме здатність представляти складні дані у вигляді мережі взаємопов'язаних об'єктів є критично важливою для побудови систем обробки природної мови. Основний виклик, що стоїть перед дослідниками та інженерами, полягає в автоматичному конструюванні (Knowledge Graph Construction, KGC) високоякісних, масштабних та достовірних графів знань з величезних обсягів переважно неструктурованих текстових даних. Цей процес є надзвичайно складним через фундаментальні проблеми, властиві природній мові: лексична та синтаксична неоднозначність, різноманітність форматів даних, а також необхідність забезпечення високої точності та повноти вилучених знань. Неточність на будь-якому етапі побудови може призвести до створення помилкових фактів у графі, що підриває його корисність для подальших застосувань.

1. Формалізація задачі

Для забезпечення строгості аналізу необхідно формалізувати ключові поняття. Граф знань є структурованою репрезентацією фактів про світ у вигляді графа. Формально, граф знань G визначається як орієнтований, маркований мультиграф, що представлений трійкою:

$$G=(E, R, T),$$

де

- E – скінченна множина сутностей (entities), що є вузлами графа. Сутності представляють реальні об'єкти, події або абстрактні поняття (наприклад, «Париж», «Ейфелева вежа», «Франція»).
- R – скінченна множина відношень (relations) або предикатів, що є типами ребер у графі. Відношення описують семантичні

зв'язки між сутностями (наприклад, «столиця», «знаходиться»).

- T – множина трійок (triples) або фактів, що є маркованими ребрами графа. Кожна трійка має вигляд $\langle s, p, o \rangle$, де $s \in E$ – суб'єкт (subject), $p \in R$ – предикат (predicate), а $o \in E$ – об'єкт (object). Приклад трійки:

$\langle \text{Ейфелева вежа, знаходиться, Париж} \rangle$

Важливою складовою графа знань є його схема або онтологія, яка визначає класи сутностей та типи відношень разом з їхніми властивостями. Схема забезпечує семантичну узгодженість графа і може бути описана за допомогою стандартних мов, таких як RDF Schema (RDFS) або Web Ontology Language (OWL). Сама онтологія також може бути представлена у вигляді графа і бути частиною загального KG. Таким чином, кінцевою метою будь-якого процесу побудови графа знань (KGC) є автоматичне генерування множини достовірних трійок T з неструктурованих джерел, які відповідають заздалегідь визначеній або динамічно створюваній онтології.

2. Стан сучасних досліджень

Проблеми сучасних досліджень полягають у вибіркового звітуванні про результати, де автори звертають уваги лише на ті метрики, де їхній метод показує хороші результати. Крім цього, значна кількість робіт використовує неадекватні протоколи оцінки, зокрема, симуляцію низькоресурсних умов шляхом простого зменшення вибірки з високоресурсних датасетів, що було гостро розкритиковано у роботі [1]. Це не відображає реальних лінгвістичних та доменних викликів малоресурсних мов. Також, багато досліджень не надають відкритого доступу до коду та даних, що ускладнює відтворюваність та перевірку результатів. Надзвичайно висока гетерогенність у дослідженнях не дає змоги провести адекватне кількісне порівняння показників метрик, так як навіть при використанні загальновідомих датасетів чи бенчмарків, дослідники застосовують різні стратегії семплювання. Ми дійшли до висновку, що результати досліджень інших авторів є глибоко залежними від конкретної експериментальної установки, тому оцінки метрик подаються для приблизного порівняння з деякою похибкою. Замість статистичного об'єднання, у цій роботі використовується підхід якісного синтезу. Замість того, щоб намагатися обчислити єдиний середній ефект, аналіз фокусується на



величині та послідовності повідомлених ефектів. Такий підхід дозволяє оцінити відносний вплив різних методологічних інновацій, не вдаючись до статистично необґрунтованих узагальнень. Він дає змогу зробити якісну оцінку, вказуючи, які методи, ймовірно, матимуть найбільший ефект у певних умовах.

Історично першим і найбільш інтуїтивним підходом до автоматизації KGC був конвеєрний (pipeline) підхід. Його ідея полягає у тому, щоб декомпонувати складну задачу на три етапи:

- Вилучення іменованих сутностей
- Вилучення відношень між сутностями
- Зв'язування сутностей

Незважаючи на свою логічну простоту та модульність, що дозволяє використовувати найкращі інструменти для кожної окремої підзадачі, конвеєрний підхід має фундаментальний недолік – каскадне поширення помилок (error propagation), коли похибки попередніх складових системи породжують похибки у наступних. Припущення про незалежність підзадач у конвеєрних моделях є не просто обмеженням, а фундаментальною помилкою. Класичний конвеєрний підхід базується на декомпозиції KGC на послідовні, ізольовані етапи. Ця архітектура імпліцитно припускає, що завдання NER, RE та EL можуть бути вирішені оптимально незалежно одне від одного. Насправді, ці підзадачі є глибоко взаємопов'язаними [2]. У відповідь на обмеження класичного конвеєра, дослідницька спільнота розробила кілька нових архітектурних парадигм, кожна з яких пропонує свій підхід до вирішення проблеми взаємозалежності підзадач та підвищення якості KGC. Розглянемо наступні парадигми:

Спільного вилучення – ця парадигма була розроблена як пряма відповідь на проблему каскадного поширення помилок. Її основна мета – одночасно моделювати вилучення сутностей та відношень в рамках єдиної, інтегрованої архітектури, дозволяючи моделі враховувати залежності між цими задачами. Цікавим прикладом є модель TaG (Table-to-Graph Generation), розроблена для спільного вилучення сутностей та відношень на рівні документа [3]. Ключова інновація TaG полягає в динамічній побудові латентного графа між усіма згадками сутностей у документі. Методи, які використовують механізм спільного вилучення значно зменшують каскадне поширення похибки [3], однак приводять до ускладнення архітектури та, як показали автори у своїй роботі [4], стикаються з проблемою накладання відношень (overlapping relations), коли одна й та сама пара сутностей може мати кілька різних типів зв'язків у

одному контексті. Цю проблему намагаються вирішити у роботі [5] шляхом розробки складної системи тегування сутностей, однак це обмежує гнучкість моделі і робить неможливим узагальнення на нові домени.

Генеративна (Seq to Seq) – генеративна парадигма пропонує радикально інший підхід, трансформуючи задачу KGC з традиційної задачі класифікації або тегування у задачу генерації послідовності (sequence-to-sequence, Seq2Seq)[6]. Основна ідея полягає в тому, що будь-яка структурована інформація, така як трійка графа знань, може бути «лінеаризована» – перетворена на текстовий рядок певного формату [6]. Потім потужні попередньо навчені генеративні моделі, такі як T5 або BART, навчаються генерувати цей рядок на основі вхідного тексту. Одна і та ж мовна модель виконує всі етапи (NER, RE, RL) графа знань шляхом лише зміни формату цільової послідовності [6]. Це також дозволяє обробляти складні сценарії такі як парні відношення або відношення, що перекриваються, які для моделей спільного вилучення є майже неможливими. Фундаментальний недолік генеративної парадигми полягає у втраті структурної інформації. Процес лінеаризації, по суті, є «стисненням з втратами» для графової структури. Дослідники у роботі [7] дійшли до висновку, що генеративні моделі не мають уявлення про топологію, зв'язки та структуру графа і через це, охоплення ієрархічно складної інформації є справжнім викликом.

«Питання – відповідь» (QA) – пропонує елегантний спосіб структурувати процес вилучення знань, переформулювавши його як багатокрокову задачу QA (Multi-turn Question Answering). Замість того, щоб пасивно класифікувати відношення, модель активно «допитує» текст за допомогою серії послідовних, зарання створених запитань, щоб вилучити компоненти трійок. Це робить процес вилучення ітеративним, більш керованим, інтерпретованим та дозволяє моделі сфокусуватися на конкретному аспекті інформації на кожному кроці [8]. Проте, ефективність цього підходу сильно залежить від якості та повноти розроблених шаблонів питань. Для доменів зі складною та розгалуженою онтологією створення вичерпного набору таких шаблонів може бути нетривіальним і трудомістким завданням [8]. Крім того, як і в конвеєрному підході, тут існує ризик поширення помилок між «кроками» QA: неправильна відповідь на першому кроці може призвести до некоректного формулювання наступного питання. Деякі дослідники пропонують використовувати методи навчання з підкріпленням для пом'якшення



цієї проблеми, винагороджуючи модель за довгі та правильні ланцюжки відповідей [8].

Великих мовних моделей – метод використовує zero shot/few shot навчання, які дозволяють розширити використання LLM на більш вузький домен. Прикладом структурованого підходу на базі LLM є фреймворк «Extract, Define, Canonicalize» (EDC) [9]. Він декомпозує KGC на три фази, на кожній з яких використовується LLM:

1. Extract: LLM використовується для вилучення трійок з тексту у відкритому інформаційному форматі (Open Information Extraction), без прив'язки до жорсткої схеми.
2. Define: На основі вилучених трійок та їхнього контексту, LLM генерує визначення для типів сутностей та відношень, фактично створюючи схему «на льоту».
3. Canonicalize: LLM використовується для уніфікації та злиття синонімічних сутностей та відношень (напр., «засновник» та «був заснований»), створюючи таким чином узгоджену та канонічну схему.

Даний підхід дозволяє делегувати створення KG системи для LLM, з необхідністю лише вказати формат виводу та налаштувати сховище даних. На відміну від більш структурованих моделей, LLM не мають вбудованих механізмів для забезпечення фактологічної точності та узгодженості зі схемою. Практичні дослідження виявляють три системні проблеми: галюцинації (генерування фактів, яких немає у вхідному тексті), недостатність доменних знань (неправильна інтерпретація спеціалізованої термінології) та чутливість до шуму (вилучення нерелевантної інформації з тексту) [10]. Автори роботи [10] наголошують, що ці проблеми є іманентними властивостями сучасних LLM та наполягають на критичній необхідності механізмів валідації результатів. Крім того, вивід LLM є ймовірнісним, тобто недетермінованим, що є дуже критичним недоліком в деяких сферах, як-от медицина, атомна енергетика чи біоінформатика, де ціна похибки може бути високою.

Для об'єктивної оцінки ефективності розглянутих архітектурних парадигм, необхідно звернутися до емпіричних даних. У таблиці зведено результати продуктивності ключових моделей, що представляють різні підходи, на стандартизованих бенчмарках.

Таблиця

Порівняння міри F1 моделей KGC на академічних датасетах
(ACE05, CoNLL04)

Парадигма	Дослідження	Модель	Датасет	F1-score
Питання-відповідь	Xiaoya Li та ін. [8]	Multi-turn QA	ACE05	60.3%
			CoNLL04	69.2%
Спільне вилучення	Du Chen та ін. [5]	BERT-BiGRU-CRF	ACE05	59.9%
			CoNLL04	68.0%
	Ruoyu Zhang та ін. [3]	TaG	ACE05	58.1%
			CoNLL04	60.9%
Генеративна	Igor Melnyk та ін. [11]	Grapher (T5-base)	ACE05	57.3%
			CoNLL04	70.1%

3. Випадок малоресурсної мови

Малоресурсна мова – це та, для якої бракує цифрових ресурсів (корпусів текстів, словників, анотованих даних) та інструментів обробки, що ускладнює її використання в автоматизованих системах і штучному інтелекті. У контексті малоресурсності, домінує 4 стратегії, серед них:

- *Використання опорної мови (pivot language)* – зазвичай використовують англійську мову для перекладу з однієї мови на цільову. Дослідження [12] показує, що чим ближче опорна мова знаходиться у лексичному до цільової та вихідної, тим кращі результати.
- *Few-Shot та метанавчання* – коли навіть невеликий обсяг розмічених даних доступний, методи FSL стають привабливою альтернативою. Ці підходи, часто засновані на метанавчанні, вчать «вчитися» з малої кількості прикладів. У контексті екстракції інформації прототипні мережі є поширеним методом. Ідея полягає в тому, щоб навчити модель створювати векторний «прототип» для кожного класу відношень на основі кількох прикладів (support set), а потім класифікувати нові приклади (query set) на основі їхньої близькості до цих прототипів. Дослідження в біомедичній сфері показали [13], що використання існуючих графів знань для генерації різноманітних синтетичних прикладів може значно покращити ефективність FSL для розпізнавання іменованих сутностей. Однак, головним обмеженням FSL є те, що він



зазвичай не може обробляти відношення, які не були представлені навіть у невеликій навчальній вибірці.

- *Промпт-орієнтовані методи* – поява великих мовних моделей кардинально змінила ландшафт. LLMs продемонстрували здатність виконувати завдання в режимі zero-shot та few-shot через «навчання в контексті» (in-context learning), коли кілька прикладів надаються безпосередньо в запиті (промпті). Це усуває необхідність у складному fine-tuning. Роботи, такі як REPAL 7, пішли ще далі, використовуючи LLMs не як кінцевий екстрактор, а як інструмент для генерації високоякісних синтетичних навчальних даних. У цьому фреймворку LLM отримує лише текстове визначення відношення (наприклад, «відношення 'столиця' пов'язує країну з її столицею») і генерує позитивні та негативні приклади. Потім ці синтетичні дані використовуються для навчання набагато меншої, ефективнішої моделі (наприклад, на базі RoBERTa), яка виконує кінцеву екстракцію. Цей підхід вирішує дві проблеми: високу вартість використання LLM API для обробки великих обсягів тексту та їхню схильність до «галюцинацій». Схожий підхід був застосований для уйгурської мови, де промпти використовувалися для генерації синтетичних даних для навчання екстрактора триплетів [14].
- *Пошуково-доповнена генерація (RAG)* – архітектури RAG є кульмінацією сучасних тенденцій і поєднують сильні сторони генеративних моделей та інформаційного пошуку. Система RAG складається з двох ключових компонентів:
 - **Ретривер (Retriever):** модуль, що відповідає за пошук релевантних фрагментів тексту з великого зовнішнього корпусу (наприклад, Вікіпедії) на основі вхідного запиту.
 - **Генератор (Generator):** велика мовна модель, яка отримує як вхідний запит, так і знайдені ретривером тексти, і генерує кінцеву відповідь.

У контексті KGC для малоресурсних мов RAG є надзвичайно потужною парадигмою. Робота mRAKL [15] демонструє її ефективність для амхарської та тигринья. Замість того, щоб покладатися на обмежені внутрішні знання LLM про ці мови, система спочатку знаходить релевантні статті з Вікіпедії відповідною мовою, а потім передає їх генератору для вилучення факту. Ключовим аспектом є якість ретривера. Дослідження [15] показує, що

створення диференційованого, наскрізно навчуваного ретривера, який оптимізується разом з генератором під кінцеву задачу, значно покращує продуктивність порівняно з використанням фіксованих ретриверів, таких як BM25.

4. Пропонована методологія

Жоден з існуючих підходів не є достатньо надійним сам по собі. Крос-мовне проєктування залежить від паралельних даних, zero-shot LLM є ненадійними, а стандартний RAG не забезпечує належного контролю над схемою вилучення. Оптимальне рішення лежить у синергетичному поєднанні сильних сторін RAG та QA-парадигми.

Аргументація Вибору:

- *RAG для Заземлення та Фактологічності:* компонент RAG, як у mRAKL [15], вирішує ключову проблему LLM – галюцинації. Замість того, щоб покладатися на потенційно нерелевантні або невірні параметричні знання моделі, система змушена базувати свої висновки на фрагментах тексту з реального, хоч і невеликого, корпусу на цільовій мові. Це значно підвищує фактологічну точність вилучених трійок.
- *QA для Контролю над Схемою та Інтерпретованістю:* Формулювання задачі у вигляді багатокрокового QA вирішує проблему безконтрольності та непередбачуваності LLM [8]. Замість загального промпу «вилуч всі трійки», система ставить конкретні, керовані схемою питання (напр., «Яка столиця у [країна]?»). Це дозволяє:
 1. Контролювати Схему: явно вказувати, які саме типи сутностей та відношень потрібно вилучати, забезпечуючи відповідність кінцевого графа заданій онтології.
 2. Декомпонувати Складність: розбивати складне завдання вилучення на послідовність простіших, керованих кроків.
 3. Моделювати Залежності: використовувати відповіді, отримані на попередніх кроках (наприклад, вилучені сутності), для динамічного формування нових, більш конкретних запитань.

Схема пропонованого Конвеєра (Hybrid RAG-QA Pipeline) полягає у наступному. Пропонований конвеєр поєднує ці два підходи в ітеративному циклі для побудови графа знань.



1. Вхідні дані:
 - Неструктурований текстовий корпус D на малоресурсній мові.
 - Цільова схема (онтологія) S , що визначає типи сутностей (E та відношень R).
2. Індксація Корпусу:
 - Створення пошукового індексу (наприклад, на основі BM25 або векторних ембедингів) для корпусу D для забезпечення швидкого пошуку релевантних фрагментів.
3. Генератор Питань:
 - На основі схеми S генерується початковий набір питань для вилучення сутностей першого рівня (напр., для типу сутності Країна генерується питання «Які країни згадуються в тексті?»).
4. Ітеративний Цикл Вилучення:
 - Для кожного питання Q з набору:
 - а. Крок Пошуку (Retrieval): Питання Q використовується для пошуку k найбільш релевантних текстових фрагментів (контекстів) $C=\{c_1, c_2, \dots, c_k\}$ з індексованого корпусу D .
 - б. Формування Контекстуального Промпту: Створюється промпт для LLM, що включає оригінальне питання та знайдені контексти: *Питання: [Q] \n Контекст: [C] \n Відповідь:.*
 - в. Генерація Відповіді (Generation): LLM генерує відповідь A , яка є сутністю або списком сутностей, що відповідають на питання Q .
 - г. Валідація та Збереження: Вилучені сутності валідуються (наприклад, шляхом перевірки їхньої присутності у знайденому контексті C для зменшення ризику галюцинацій) та додаються до проміжного графа знань KG_t .
 - е. Генерація Наступного Питання: Для кожної нової вилученої сутності e та на основі схеми відношень S , генерується новий набір питань для вилучення пов'язаних сутностей (напр., якщо e – це Країна, генерується питання «Яка столиця у [e]?»). Ці нові питання додаються до черги для обробки.

5. Вихід:

- Побудований граф знань KG, що відповідає заданій схемі S і заземлений на текстовому корпусі D.

Деталі для відтворюваності

Гіперпараметри:

- **Ретривер (fine-tuning):**
 - Оптимізатор: AdamW
 - Швидкість навчання (learning rate): $2 \cdot 10^{-5}$
 - Розмір батча: 64
- **Генератор (fine-tuning з LoRA):**
 - Оптимізатор: AdamW
 - Швидкість навчання: $5 \cdot 10^{-5}$
 - Ранг LoRA (r): 8 або 16
 - Альфа LoRA (α): 32
 - Розмір батча: 4–8 (залежить від розміру моделі та пам'яті GPU)

Ініціалізація:

- **Ретривер:** sentence-transformers/paraphrase-multilingual-mpnet-base-v2
- **Генератор:** google/mt5-large або meta-llama/Llama-3-8B-Instruct (з подальшим fine-tuning на доменну область задачі).

Навчання такого пайплайну є досяжним в академічних умовах. Орієнтовні вимоги: один GPU з об'ємом пам'яті 24 GB (напр., NVIDIA RTX 3090/4090) або 40–80 GB (напр., NVIDIA A100) для більш великих моделей. Орієнтовний час навчання: 24–72 години, залежно від обсягу даних для fine-tuning. Оцінку слід проводити на відкладеному тестовому наборі триплетів. Основною метрикою має бути **Точно Відповідність (Exact Match)** згенерованої сутності-хвоста з еталонною. Додатково слід вимірювати **Hits@3** та **Hits@5** для оцінки здатності моделі включати правильну відповідь до своїх найкращих прогнозів.

Висновки

Проведений аналіз продемонстрував, що еволюція методів побудови графів знань є постійним пошуком балансу між строгістю, гнучкістю та надійністю. Цей шлях пролягає від жорстких, але схильних до каскадних помилок конвеєрних систем, через більш інтегровані моделі спільного вилучення, до універсальних, але структурно-наївних генеративних архітектур та, нарешті, до



надзвичайно гнучких, але фактологічно-ненадійних підходів на базі LLM. Кожна парадигма має свої фундаментальні обмеження, що впливають безпосередньо з її архітектурних припущень. Запропонована в цій статті гібридна методологія RAG-QA є прагматичним синтезом, розробленим спеціально для вирішення проблеми KGC в умовах екстремального дефіциту даних, характерного для малоресурсних мов. Вона використовує сильні сторони сучасних підходів — гнучкість LLM та керованість QA-фреймворки. Також можна зробити висновок, що сучасні дослідження мають опиратись на спільні фреймворки до тестування та більш відкритий дизайн експериментів, що дозволить порівнювати представлені моделі на основі метрик.

1. Shuyan Zhou et al. Towards Zero-resource Cross-lingual Entity Linking. *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Pp. 243–252. URL: <https://aclanthology.org/D19-6127/> (дата звернення: 10.04.2026).
2. Hongbin Ye та ін. Generative Knowledge Graph Construction: A Review. 2022. DOI:10.48550/arXiv.2210.12714.
3. Ruoyu Zhang et al. A Novel Table-to-Graph Generation Approach. *ACL Anthology 2023*.
4. Zhe Yang et al. Learning to Leverage High-Order Medical Knowledge Graph for Joint Entity and Relation Extraction. *Findings of the Association for Computational Linguistics: ACL 2023*. ResearchGate 2023.
5. Du Chen et al. Joint Entity-Relation Extraction for Knowledge Graph. *Applied Sciences*. ResearchGate 2025. Vol. 15(13). P. 7611.
6. Hui Chen et al. Knowledge Graph Construction: A Review. *ACL Anthology 2022*.
7. Wei Guo et al. CodeKGC: Code Language Model for Generative Knowledge Graph Construction. *ACM Transactions on Asian and Low-Resource Language Information*. 2024. DOI:10.1145/3641850.
8. Fan Yin et al. Entity-Relation Extraction as Multi-Turn Question Answering. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. URL: <https://aclanthology.org/P19-1129/> (дата звернення: 10.04.2026).
9. Bowen Zhang, Harold Soh. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024. Pp. 9820–9836.
10. Knowledge Graphs and LLMs - tackling limitations with synergy. DSS Blog. URL: <https://roundtable.datascience.salon/knowledge-graphs-and-llms-tackling-limitations-with-synerg> (дата звернення: 10.04.2026).
11. Igor Melnyk, Pierre Dognin Payel Das. Grapher: Multi-Stage Knowledge Graph Construction. *Workshop on Deep Generative Models and Downstream Application*. 2021.
12. Jiaying Gong, Hoda Eldardiry. Prompt-based Zero-shot Relation Extraction with Semantic Knowledge Augmentation. *Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. 2024.
13. Peng Chen et al. Few-shot biomedical named entity recognition via knowledge-guided instance generation and prompt contrastive learning. *Bioinformatics*. Oxford Academic, 2023.
14. Auiguli Halike та ін. Zero-Shot Relation Triple Extraction with Prompts for Low-Resource. *Appl. Sci.MDPI*. 2023.
15. Hellina Hailu Nigatu, Min Li, Maartje Ter Hoeve, Saloni Potdar, Sarah Chasins. mRAKL: Multilingual Retrieval-Augmented. *Findings of the Association for Computational*

Linguistics: ACL 2025. Pp. 13072–13089.

REFERENCES:

1. Shuyan Zhou et al. Towards Zero-resource Cross-lingual Entity Linking. *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Pp. 243–252. URL: <https://aclanthology.org/D19-6127/> (data zvernennia: 10.04.2026).
 2. Hongbin Ye та ін. Generative Knowledge Graph Construction: A Review. 2022. DOI:10.48550/arXiv.2210.12714.
 3. Ruoyu Zhang et al. A Novel Table-to-Graph Generation Approach. *ACL Anthology 2023*.
 4. Zhe Yang et al. Learning to Leverage High-Order Medical Knowledge Graph for Joint Entity and Relation Extraction. *Findings of the Association for Computational Linguistics: ACL 2023*. ResearchGate 2023.
 5. Du Chen et al. Joint Entity–Relation Extraction for Knowledge Graph. *Applied Sciences*. ResearchGate 2025. Vol. 15(13). P. 7611.
 6. Hui Chen et al. Knowledge Graph Construction: A Review. *ACL Anthology 2022*.
 7. Wei Guo et al. CodeKGC: Code Language Model for Generative Knowledge Graph Construction. *ACM Transactions on Asian and Low-Resource Language Information*. 2024. DOI:10.1145/3641850.
 8. Fan Yin et al. Entity-Relation Extraction as Multi-Turn Question Answering. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. URL: <https://aclanthology.org/P19-1129/> (data zvernennia: 10.04.2026).
 9. Bowen Zhang, Harold Soh. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024. Pp. 9820–9836.
 10. Knowledge Graphs and LLMs - tackling limitations with synergy. DSS Blog. URL: <https://roundtable.datasalonsalon.com/knowledge-graphs-and-llms-tackling-limitations-with-synerg> (data zvernennia: 10.04.2026).
 11. Igor Melnyk, Pierre Dognin Payel Das. Grapher: Multi-Stage Knowledge Graph Construction. *Workshop on Deep Generative Models and Downstream Application*. 2021.
 12. Jiaying Gong, Hoda Eldardiry. Prompt-based Zero-shot Relation Extraction with Semantic Knowledge Augmentation. *Joint International Conference on Computational Linguistics, Language Resources and Evaluation*. 2024.
 13. Peng Chen et al. Few-shot biomedical named entity recognition via knowledge-guided instance generation and prompt contrastive learning. *Bioinformatics*. Oxford Academic, 2023.
 14. Auiguli Halike та ін. Zero-Shot Relation Triple Extraction with Prompts for Low-Resource. *Appl. Sci.MDPI*. 2023.
 15. Hellina Hailu Nigatu, Min Li, Maartje Ter Hoeve, Saloni Potdar, Sarah Chasins. mRAKL: Multilingual Retrieval-Augmented. *Findings of the Association for Computational Linguistics: ACL 2025*. Pp. 13072–13089.
-



Liashko D. A. [1; ORCID ID: 0009-0005-4125-7368],

Post-graduate Student,

Turbal Y. V. [1; ORCID ID: 0000-0002-5727-5334],

Doctor of Engineering, Professor,

Klumiuk Yu. Ye. [1; ORCID ID: 0000-0003-3672-8469],

Candidate of Engineering (Ph.D.), Associate Professor

¹National University of Water and Environmental Engineering, Rivne

HYBRID RAG-QA METHOD FOR BUILDING A KNOWLEDGE GRAPH IN THE CONTEXT OF A LOW-RESOURCE LANGUAGE

The article presents a comprehensive analysis of modern methods for Knowledge Graph Construction (KGC) from unstructured textual data, which form the basis for a wide range of intelligent applications – from semantic search and question-answering systems to recommendation services and data analytics. The study critically examines the key KGC paradigms, including the pipeline approach, joint entity–relation extraction, generative Seq2Seq models, multi-turn question-answering (QA) frameworks, and methods leveraging large language models (LLMs). For each paradigm, the strengths and fundamental limitations are outlined: from cascading error propagation and loss of structural information during linearization to the challenges of generating complete question templates and the issues of hallucinations and factual inconsistency in LLMs.

Particular attention is devoted to the problem of constructing knowledge graphs in low-resource languages, where the lack of sufficient corpora, annotated data, and processing tools makes traditional approaches inefficient. In this context, modern strategies are reviewed, including pivot-language approaches, few-shot and meta-learning, prompt-oriented methods for generating synthetic data, as well as retrieval-augmented generation (RAG) architectures. Examples of successful applications of these strategies in different linguistic environments and domains are provided, particularly in biomedical research and for Afro-Asiatic languages.

The paper proposes a hybrid RAG-QA methodology that combines the strengths of both approaches: factual reliability and “grounding” ensured by the retriever module, with controllability and interpretability enabled by formulating the task as multi-turn QA. This synergistic approach allows for precise alignment of the constructed knowledge graph with a predefined ontology, minimizes error risks, and improves the accuracy of entity and relation extraction in challenging low-resource settings.

Keywords: knowledge graph; knowledge graph construction; relation extraction; low-resource languages; retrieval-augmented generation (RAG); question-answering systems (QA).

Отримано: 17 червня 2025 року
Прорецензовано: 02 вересня 2025 року
Прийнято до друку: 25 вересня 2025 року