УДК 004.822:004.65 **DOI:** 10.31673/2412-9070.2024.050247
А. Ю. БУРАЧИНСЬКИЙ, аспірант, архітектор програмних рішень;
ORCID: 0009-0003-7913-2152
А. С. ШАНТИР, канд. техн. наук,
ORCID: 0000-0002-0466-3659
Державний університет інформаційно-комунікаційних технологій, Київ

РОЗРОБКА БАЗ ЗНАНЬ З ДОПОВНЕНОЮ ГЕНЕРАЦІЄЮ З ВИКОРИСТАННЯМ БАЗОВИХ МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ

Тенденції глобального технологічного зростання, здешевлення обчислювальних потужностей та постійне збільшенням обсягів даних, якими потрібно управляти та обробляти, вимагають переосмислення та вдосконалення методів і підходів до управління цими даними. В той же час, розвиток технологій штучного інтелекту, алгоритмів машинного навчання (з англ. Machine Learning, ML) та обробки природньої мови (з англ. Natural Language Processing, NLP) дали змогу істотно спростити методи і підходи до управління даними та взаємодії з ними, а також автоматизувати процеси і рутинні задачі, які обробляють ці текстові дані.

У статті висвітлено теоретичні аспекти та принципи побудови баз знань з використанням семантичного пошуку (або пошуку подібностей) та генеративного доповнення на основі базових моделей штучного інтелекту (з англ. Foundation Models). У даному дослідженні розглянуто основні принципи алгоритму роботи пошуку подібностей, який є фундаментальною функціональною складовою частиною баз знань та його реалізація за допомогою векторних баз даних. У статті також розглянуто практичний приклад реалізації такої бази знань на основі керованого сервісу Bedrock. Приведено ситуаційні приклади для організації даних у вигляді баз знань, які можна використовувати у прикладних проєктах для автоматизації та вдосконалення процесів, які оперують з текстовими даними. Запропонований варіант реалізації бази знань можна використовувати як основу для розробки інтелектуальних чат-ботів із знанням предметної області для автоматизації систем підтримки користувача.

Ключові слова: база знань, генерація з доповненням, штучний інтелект, генеративні базові моделі штучного інтелекту, пошук подібностей, семантичний пошук, векторні бази даних.

Постановка задачі

У зв’язку з глобальним технологічним зростанням та здешевленням обчислювальних потужностей, все більш гостро постають питання ефективного пошуку та управління даними: зберігання, структурування та класифікації, швидкого пошуку із релевантними пошуковими результатами, а також автоматизація рутинних задач, які виникають у процесі використання цих даних. Це, в свою чергу, вимагає переосмислення та вдосконалення методів, підходів до пошуку та аналізу даних, а також автоматизації і спрощення процесів, які обробляють текстові дані.

Аналіз останніх досліджень і публікацій

Аналіз останніх досліджень і публікацій у галузі штучного інтелекту, яка займається обробкою природньої мови (Natural Language Processing, NLP), побудовою та тренуванням генеративних моделей штучного інтелекту, дозволили спроєктувати систему на основі баз знань у поєднанні з генеративними базовими моделями, що дозволяє автоматизувати та істотно спростити процеси, які оперують з текстовими даними. В ході дослідження, проробленого під час написання статті, проаналізовано дослідницькі роботи Lewis, Patrick, Perez, Ethan стосовно використання генерації з доповненням для завдань з обробки природньої мови, які потребують

інтенсивних знань з NLP; проаналізовано публікації Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun на предмет можливості використання мовних моделей як баз знань; опрацьовано дослідження Radford A., Narasimhan K., Salimans T. на предмет можливості покращення розуміння природньої мови генеративними моделями за допомогою попереднього тренування; проаналізовано публікації Chalmers D.J. стосовно того чи можуть мовні моделі бути «розсудливими», опрацьовано джерела та публікації стосовно пошуку подібностей та варіантів їх практичної реалізації.

У цій статті ми розглянемо бази знань (з англ. Knowledge Bases) у поєднанні із генеративними моделями штучного інтелекту як ефективний спосіб пошуку та генерації релевантних результатів у великих об'ємах даних, структурування даних та забезпечення спільного доступу до цих даних авторизованим особам. Новизна запропонованого у цій статті підходу полягає у використанні баз знань саме у поєднанні з розмовними генеративними моделями штучного інтелекту, що дозволяє реалізувати більш інтелектуальні системи пошуку з генеративним доповненням на основі базових моделей штучного інтелекту, автоматизувати і спростити процеси, які обробляють текстові дані.

Результати досліджень

Оскільки фундаментальною функціональною складовою частиною баз знань є семантичний пошук (з англ. Semantic Search) або пошук подібностей (з англ. Similarity Search), у цій частині статті пропонується розглянути теоретичні засади та варіант практичної реалізації семантичного пошуку на основі векторних баз даних.

У цій частині статті розглянемо, що представляє собою семантичний пошук, та в чому його відмінність від широковідомого повно-текстового пошуку. Семантичний пошук – це тип пошуку, який використовує штучний інтелект та алгоритми машинного навчання з метою зрозуміти наміри користувача, знайти певну інформацію та контекст запиту користувача замість простого пошуку за точним або частковим співпадінням текстових фраз. Наприклад, якщо ми шукаємо "Який преміальний автомобіль кращий?", семантичний пошук врахував би, що ми шукаємо порівняння преміальних брендів автомобілів, їх переваг та особливостей, а не просто сторінки, які включають ці слова. В той час як традиційний повно-текстовий пошук здійснює простий пошук повних або часткових співпадінь слів або фраз, ніж аналізує семантику та контекст. Якщо б ми ввели той самий запит ("Який преміальний автомобіль кращий?") в систему повно-текстового пошуку, вона б у переважній більшості випадків просто знайшла б сторінки, які включають ці слова або фрази і зовсім не обов'язково дають відповідь на наше запитання.

У контексті семантичного пошуку або пошуку подібностей кожна фраза або концепція представляється як вектор у багатовимірному просторі. У практичних реалізаціях семантичного пошуку кількість вимірів може варіювати від щонайменше декількох сотень до кількох тисяч. Якщо спробувати уявити простір, наприклад, з 1500 вимірами, спрощена візуальна ілюстрація концептів (термінів) у тривимірному просторі мала б вигляд, представлений на рис.1.

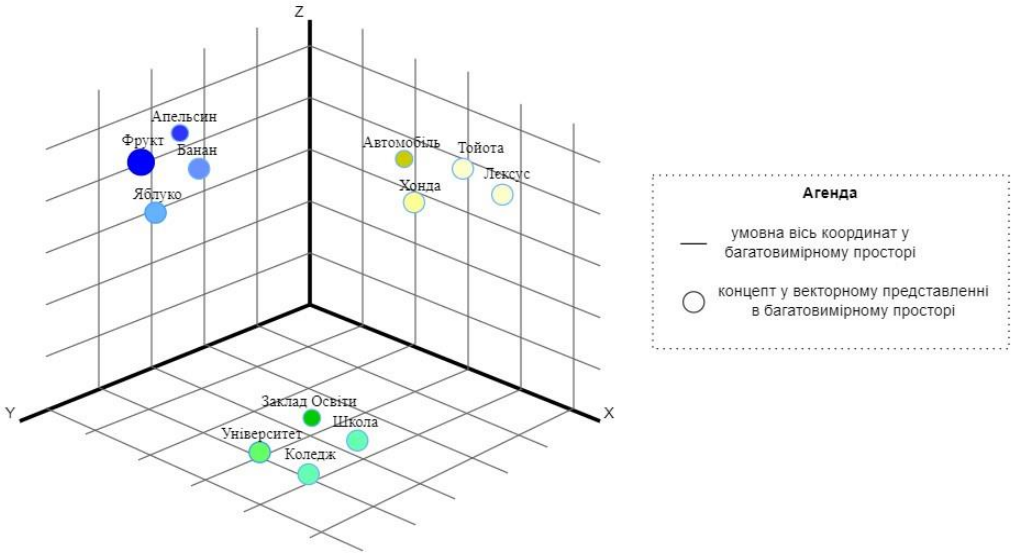

Рис. 1. Концепти у векторному представленні в багатовимірному просторі

Таким чином, для реалізації семантичного пошуку або пошуку подібностей концепти (слова і фрази) конвертуються у семантичні вектори за допомогою відповідної моделі трансформації. Наприклад, Embeddings API від OpenAI або бібліотека SentenceTransformers описують алгоритм та методику трансформації термінів у векторні представлення. Мета такої конвертації даних — щоб семантично подібні елементи (або терміни) представлялись у вигляді близьких один до одного векторів у багатовимірному просторі.

Для практичної реалізації семантичного пошуку використовуються векторні бази даних, найбільш розповсюдженими з яких є pgvector від Postgres, LlamaIndex від Meta, Chroma, Milvus та інші. Векторні бази даних найчастіше використовують алгоритми пошуку наближено найближчих сусідів (ННС, англ. Approximate Nearest Neighbor, ANN) [1][2], що дає можливість здійснювати пошук у базі даних, знаходячи найближчі сусідні записи до терміну, який потрібно знайти. За замовчуванням у векторних базах даних зазвичай використовується точний пошук найближчих сусідів (англ. Exact Nearest Neighbor Search, ENN), який базується на поелементному порівнянні векторів і є точнішим за наближений пошук найближчих сусідів. Але у зв'язку із переважаючою продуктивністю та прийнятними з точки зору релевантності пошуковими результатами, зазвичай перевага віддається наближеному пошуку найближчих сусідів. При використанні наближеного пошуку найближчих сусідів релевантність пошукових результатів та час пошуку, як правило, є компромісним і може налаштовуватися та отпимізуватися шляхом побудови відповідних індексів. Варто відзначити, що розмір індексів та час перебудови індексу при вставці нових векторів безпосередньо впливає на точність і швидкість пошуку, а також вартість хмарної інфраструктури.

Розглянемо основи алгоритму семантичного пошуку, який використовується векторними базами даних. Векторні бази даних використовують два основні алгоритми пошуку:

1. Вимірювання Евклідової дистанції (з англ. Euclidean (or L2) distance) між векторами у багатовимірному просторі. Евклідова відстань є різницею координат двох точок в Евклідовому просторі. Графічна ілюстрація даного підходу представлена на рис. 2.

Рис. 2. Евклідова дистанція між векторами

Евклідова відстань є фундаментальним математичним інструментом для вимірювання відстаней у Евклідових просторах та має широке застосування у сфері комп'ютерних наук, статистиці та машинному навчанні у зв'язку з простотою обчислення та інтуїтивно зрозумілою інтерпретацією.

2. Косинусова схожість (англ. Cosine Similarity) – це метрика, яка показує, на скільки схожі вектори, які порівнюються, у багатовимірному просторі. З математичної точки зору косинусову схожість з деяким наближенням можна розглядати як кут між двома векторами у багатовимірному просторі (рис. 3). Використання косинусової схожості допомагає виявити семантичні зв'язки між словами або фразами, які можуть не бути явно вираженими в тексті. Це особливо корисно для обробки природної мови (з англ. NLP), де важливо визначити схожі-

сть концептів (або термінів), а не лише точні збіги слів.

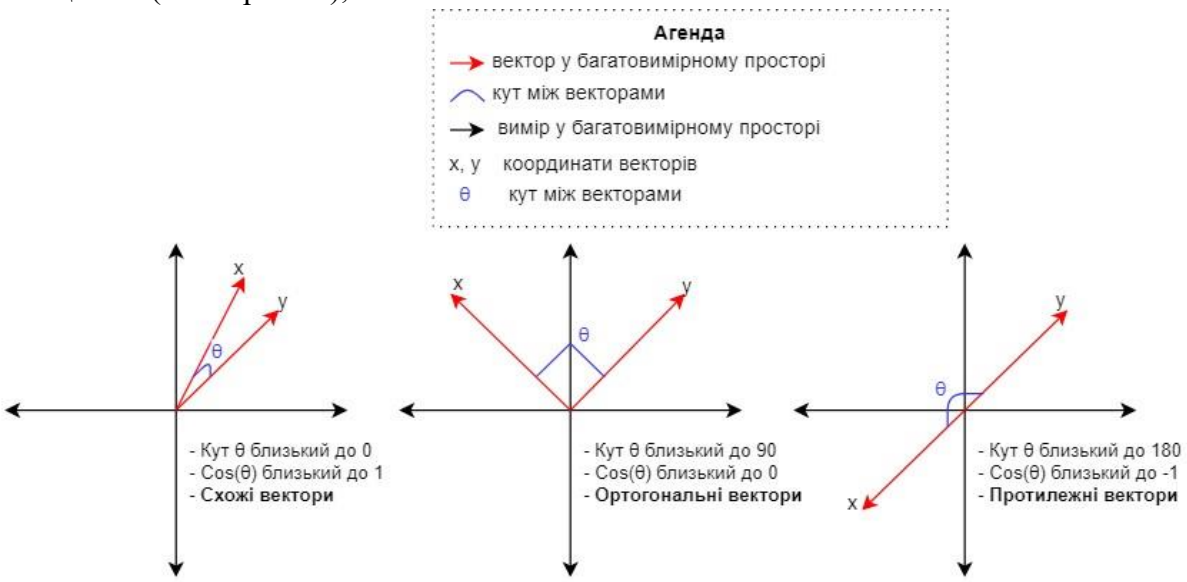

Рис. 3. Косинусова схожість між векторами

У семантичному пошуку косинусова схожість дозволяє здійснювати порівняння між запитом користувача і набором концептів, що зберігаються в базі даних. Результати пошуку потім ранжуються за мірою схожості, що допомагає користувачеві знайти найбільш релевантну інформацію. Як правило, при ранжуванні результатів по мірі подібності застосовується граничний допустимий коефіцієнт подібності для фільтрації нерелевантних результатів. Косинусова схожість ігнорує довжину тексту, оскільки фокусується на напрямку вектора, а не на його абсолютній величині. Це забезпечує ефективність та високу релевантність результатів вибірки навіть при порівнянні текстів різної довжини.

Для семантичного пошуку рекомендованою до використання є косинусова схожість, адже цей алгоритм забезпечує значно кращу продуктивність із прийнятною релевантністю пошукових результатів.

Для оптимізації продуктивності пошуку створюються та налаштовуються індекси, які надзвичайно пришвидшують семантичний пошук на великих об'ємах даних. Як правило, індекс являє собою компроміс між розміром (від кількох сотень мегабайт до кількох гігабайт), швидкістю перебудови при вставці нових даних, швидкісного пошуку та вартістю сервісу у хмарному середовищі.

Ілюстрація алгоритму роботи семантичного пошуку (пошуку подібностей) з використанням векторних баз даних приведена на блок-схемі (рис. 4).

Рис. 4. Алгоритм роботи семантичного пошуку

Алгоритм роботи семантичного пошуку на рис.4 дещо спрощений і відображає основні логічні кроки та принципи, абстрагуючись від деталей та засобів реалізації конкретних кроків алгоритму.

Приклад побудови бази знань з генеративним доповненням з використанням базових моделей штучного інтелекту. У цій частині статті розглянемо теоретичні аспекти та практичний приклад побудови бази знань із застосуванням семантичного пошуку та алгоритму вибірки з

доповненням (з англ. Retrieval Augmented Generation, RAG) за допомогою генеративних агентів на основі попередньо тренованих базових моделей штучного інтелекту.

База знань – це з деяким наближенням база даних, призначена для управління даними та інформацією, тобто збору, зберігання, структурування та класифікації, пошуку та видаванням релевантних результатів. З появою та розвитком галузі штучного інтелекту, яка займається обробкою природньої мови NLP, а також завдяки появі та вдосконаленню мовних генеративних моделей, стало можливим поєднання традиційних підходів пошуку і зберігання даних із новітніми генеративними методами для автоматизації рутинних задач та оптимізації роботи з текстовими даними.

Розглянемо практичний приклад реалізації бази знань у поєднанні з генеритивними базо-

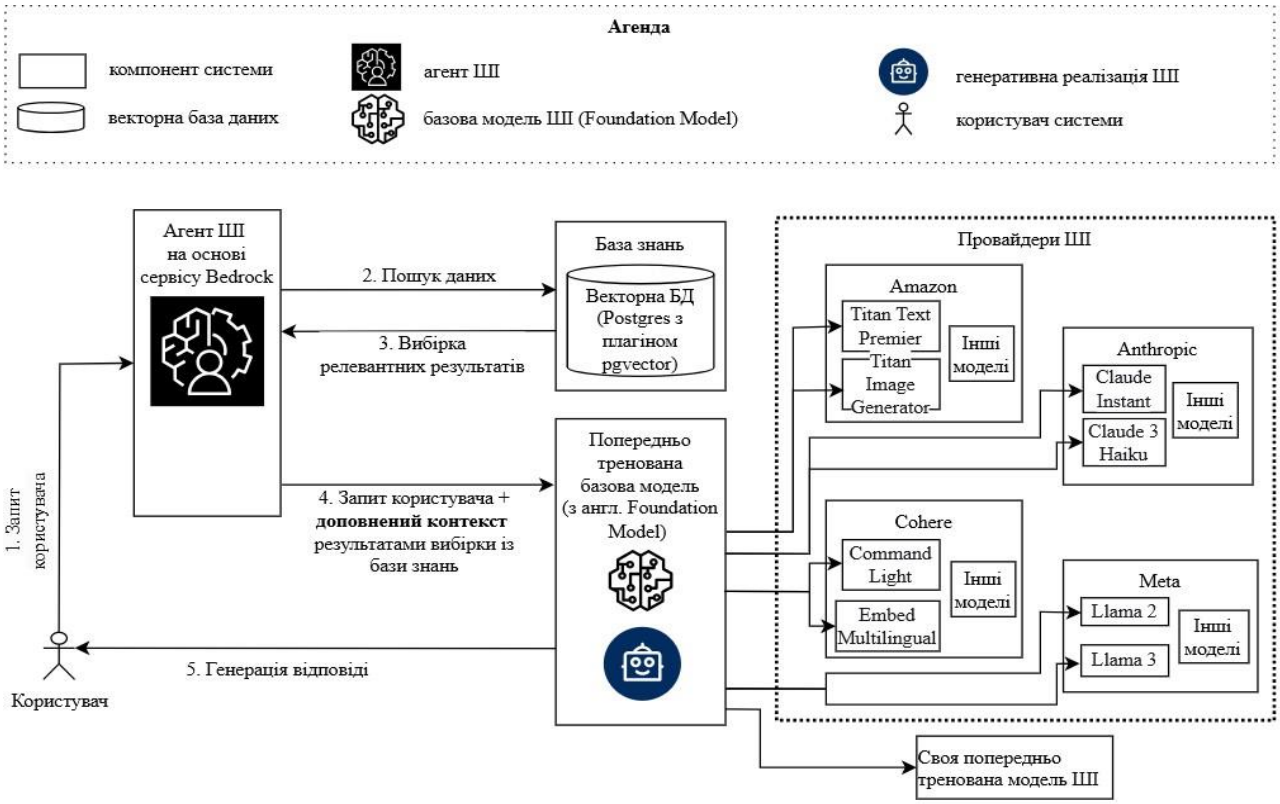

Рис. 5. База знань у поєднанні з генеративними базовими моделями ШІ

вими моделями штучного інтелекту на основі управляємого сервісу Amazon Bedrock (рис. 5).

У запропонованому варіанті реалізації бази знань з генеративним доповненням з використанням генеративних базових моделей штучного інтелекту алгоритм роботи системи наступний:

1. Користувач надсилає запит.

2. Система за допомогою семантичного пошуку здійснює пошук у векторній базі даних та повертає найбільш релевантні результати. Знайдені результати можуть передаватися з атрибутом джерело даних (source) для мінімізації ймовірності «галюцинацій» генеративної моделі штучного інтелекту.

3. Агент штучного інтелекту на основі сервісу Bedrock створює контекст, використовуючи знайдені результати та ініціює запит до попередньо тренованої базової моделі (з англ. Foundation Model) штучного інтелекту.

4. Запит до генеративної базової моделі або виклик стороннього провайдера ШІ через прикладний програмний інтерфейс (з англ. API від Application Programming Interface). В залежності від конфігурації та функціональних вимог до системи, можуть використовуватись як базові моделі, так і запити до сторонніх провайдерів штучного інтелекту (таких як OpenAI, Anthropic, AzureAI, Meta), використовуючи їх прикладні програмні інтерфейси.

5. Генеративна реалізація штучного інтелекту, використовуючи згенеровані базовою моделлю або провайдером ШІ результати, генерує відповідь.

Розглянемо, що представляє собою генеративна базова модель штучного інтелекту. Генеративна базова модель – це попередньо тренована на великих текстових (або графічних) даних модель, оптимізована під певні задачі (найбільш поширеними задачами є генерація тексту або зображень). Базові моделі штучного інтелекту першопочатково тренуються на великих об'ємах даних, а потім гранулярно налаштовуються за допомогою таких методів машинного навчання, як наприклад, навчання з підкріпленням (з англ. Reinforcement Learning) та навчання з підкріпленням зі зворотним зв'язком (з англ. Reinforcement Learning with Human Feedback), які полягають у ранжуванні результатів алгоритму машинного навчання та повторних ітерацій з метою отримання покращених результатів.

Ситуаційні приклади для організації даних у вигляді баз знань. Нижче приведені деякі реальні приклади, в яких можна автоматизувати рутинні операції, вдосконалити і спростити управління даними за допомогою організації даних у вигляді баз знань:

• Системи підтримки користувача. Бази знань можуть використовуватися для зберігання статей, частих запитань (FAQ) та інструкцій. Таким чином, користувачі можуть швидко знаходити рішення своїх (часто однотипних) проблем без необхідності залучення фахівців служби підтримки. А доповнення бази знань генеративними агентами можна використовувати для реалізації чат-ботів служби підтримки.

• Корпоративні бази знань. У комерційних та виробничих компаніях бази знань можуть використовуватися для зберігання та структурування даних стосовно виробничих процесів, корпоративних практик, навчальних практик, взаємного обміну досвіду співробітників, що сприяє оптимізації процесів і підвищенню ефективності.

• Системи управління проєктами та розробка програмного забезпечення. Бази знань дозволяють командам зберігати, структурувати, здійснювати спільний доступ до даних, а також автоматизувати деякі рутинні задачі, такі як генерація сценаріїв використання, тестових сценаріїв та супровідної проєктної документації.

• Освітні та навчальні платформи. Університети, заклади освіти, онлайн-курси можуть використовувати бази знань для зберігання, структурування, швидкого пошуку та спільного доступу до навчальних матеріалів, методичних рекомендацій та матеріалів для самопідготовки студентів.

• Медицина. Цифровізована інформація про симптоми, діагнози хвороб та методи лікування вимірюється терабайтами даних. Застосування баз знань у медицині або медицинському програмному забезпеченні дозволятиме лікарям набагато швидше отримувати потрібну інформацію, точніше ідентифікувати потенційні діагнози та надавати пацієнтам точніші та своєчасні рекомендації стосовно лікування.

Висновки

У зв'язку з глобальним технологічним зростанням та збільшенням обсягів даних, все більш актуальними постають питання ефективного пошуку, аналізу та управління цими даними. У даній статті висвітлено теоретичні аспекти та принципи побудови баз знань з використанням генеративного доповнення на основі мовних базових моделей штучного інтелекту. Розглянуто практичний приклад реалізації такої бази знань на основі керованого сервісу Bedrock від провайдера хмарних сервісів AWS. Результати дослідження, отримані під час написання цієї статті, дозволяють будувати бази знань для вдосконалення пошуку, аналізу та управління великими об'ємами даних, а також можуть застосовуватись у прикладних проєктах для автоматизації процесів обробки текстових даних. Ситуаційні приклади для організації даних у вигляді баз знань описують сценарії та ситуації для організації даних у вигляді баз знань.

Список літератури

1. Roie Schwaber-Cohen. What is a Vector Database & How Does it Work. Pinecone. Cited on 18 November 2023. URL: https://www.pinecone.io/learn/vector-database

2. Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman; Küttler, Heinrich. Retrieval-augmented generation for knowledge-intensive NLP tasks Advances in Neural Information Processing Systems 33. 2020. URL:

https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5Abstract.html

3. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. Improving Language Understanding by Generative Pre-training. OpenAI. 2018. P. 2-6 URL: https://hayate-lab.com/wp-content/uploads/2023/05/43372bfa750340059ad87ac8e538c53b.pdf

4. Brown, T. B., et al. Language Models are Few-Shot Learners. 2020. P. 7-12, 26-37. URL: https://arxiv.org/pdf/2005.14165

5. Dodge, J., et al. Fine-Tuning Language Models from Human Preferences. 2020. P. 2-4, 11-12. URL: https://arxiv.org/pdf/1909.08593

6. Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases, 2021. URL: https://arxiv.org/pdf/2106.09231

7. Chalmers, D.J. Could a large language model be conscious? 2023. URL: https://arxiv.org/pdf/2303.07103

8. Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In Proceedings of EACL, 2014. URL: https://aclanthology.org/E14-1049.pdf

9. Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In Proceedings of ICML, 2012. P. 6-7. URL: https://icml.cc/2012/papers/855.pdf

10. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. P. 2-8. URL: https://arxiv.org/pdf/1301.3781

11. Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of ICML, 2008. URL: https://ronan.collobert.com/pub/matos/2008_nlp_icml.pdf

12. Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In Proc. of ACL, 2010. URL: https://aclanthology.org/P10-1040.pdf

A. Burachynskyi, A. Shantyr

DEVELOPMENT OF KNOWLEDGE BASES WITH AUGMENTED GENERATION USING BASIC MODELS OF ARTIFICIAL INTELLIGENCE

Trends in global technology growth, computing power costs cheapening and continuous increase in data volumes that need to be managed and processed, necessitate cardinal reconsideration and improvement of methods and approaches to data governance and management. At same time, recent achievements in artificial intelligence (further abbreviated as AI) field, machine learning (abbreviated as ML) technologies, and natural language processing (abbreviated as NLP) algorithms have signifycantly simplified methods and approaches to data management and interaction, as well as allow to substantially automate processes and routine tasks, which handle this textual data. The article descrybes theoretical aspects and principles of building knowledge bases using semantic search (or similarity search) and generative augmented generation (abbreviated as RAG), based on foundation models of artificial intelligence (frequently referred to as Foundation Models). This study examines the fundamental principles of similarity search algorithm as a fundamental functional component of knowledge bases, and its implementation using vector databases. The article also discusses a practical example of implementing such knowledge bases based on managed Bedrock service. Situational examples provided in the article for organizing data in form of knowledge bases, can be used in practical projects for automating and enhancing processes which operate with the textual data. The implementation of the knowledge base proposed in the article, can be used as a foundation for developing intelligent chatbots familiar with the subject domain for automating user support systems. Knowledge bases, integrated with retrieval augmented generation also can be used for scientific researches and variety of scientific tasks, like data analysis and classification, automation of literature review, question-answering systems, data extraction from unstructured sources. Knowledge bases can be used as collaborative research platforms.

Keywords: Knowledge bases, retrieval augmented generation, artificial intelligence, generative foundation models of artificial intelligence, natural language processing, similarity search, semantic search, vector databases.