DOI: https://doi.org/10.26565/2304-6201-2025-66-03

УДК (UDC) 004.65; 004.8

Горбачова Людмила Олегівна *студентка кафедри інтелектуальних програмних систем і технологій; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: xa12850503@student.karazin.ua
https://orcid.org/0000-0002-6053-7235

Хруслов Максим Михайлович *завідувач кафедри комп'ютерних систем та робототехніки, кандидат фізико-математичних наук, старший дослідник, доцент; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: maksym.khruslov@karazin.ua
https://orcid.org/0000-0001-9639-9340

Чуб Ольга Ігорівна *доцент закладу вищої освіти кафедри комп'ютерних систем та робототехніки, кандидат економічних наук; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: o.i.chub@karazin.ua
https://orcid.org/0000-0002-1216-856X

Бережний Артем Андрійович *старший викладач закладу вищої освіти кафедри комп'ютерних систем та робототехніки, магістр; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: artem.berezhnyi@karazin.ua;
https://orcid.org/0000-0001-5407-9015

Козюберда Дмитро Олександрович *магістр кібербезпеки, факультет комп'ютерних наук, Харківськийнаціональний університет імені В. Н. Каразіна; співробітник-розробник ТОВ «ЛАДИЗАЙН»; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: koziuberda.dmytro@gmail.com;
https://orcid.org/0009-0005-3088-9685

Дослідження процедури перетворення тексту в SQL на основі large language models (LLM) шляхом міждоменного семантичного аналізу

Theme of work. Research on the Text-to-SQL conversion procedure based on Large Language Models (LLM) through Cross-Domain Semantic Analysis.
Purpose of work. To enhance the accuracy and adaptability of Text-to-SQL conversion using Large Language Models (LLM) through cross-domain semantic analysis, enabling reliable query interpretation across various domains and database structures. **Methods of research.** Comparative analysis, experimental evaluation, cross-domain semantic testing. **Results.** The research demonstrates that optimized prompt engineering and fine-tuning significantly improve the accuracy and cross-domain adaptability of Large Language Models for Text-to-SQL conversion. **Conclusions.** This study confirms that Large Language Models (LLMs) can effectively enhance the Text-to-SQL conversion process when optimized with targeted prompt engineering and fine-tuning. Cross-domain semantic analysis proved essential for enabling LLMs to handle varied database structures and domain-specific terminology, improving versatility and accuracy. The findings highlight the potential of LLMs to make SQL query generation more accessible to non-technical users, promoting broader application of AI in database management. Future work may focus on further refining these models to reduce computational costs and increase processing efficiency.

Ключові слова: *Large Language Models (LLM), Natural Language Processing (NLP), Text-to-SQL, Обробка природної мови, Глибоке навчання, Нейронні мережі, Бази даних, Штучний інтелект, Аналіз даних, Автоматизація, Інформаційні системи.*

Як цитувати: Горбачова Л. О., Хруслов М. М., Чуб О. І., Бережний А. А., Козюберда Д. О. Дослідження процедури перетворення тексту в SQL на основі large language models (LLM) шляхом міждоменного семантичного аналізу. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 66. С. 37-44. https://doi.org/10.26565/2304-6201-2025-66-03

How to quote: Horbachova L., Khruslov M., Chub O., Berezhnyi A., Koziuberda D., “Research of the procedure for converting text into sql based on large language models (LLM) through cross-domaine semantic analysis”, *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 66, pp. 37-44, 2025. https://doi.org/10.26565/2304-6201-2025-66-03[in Ukrainian]

Вступ

Large Language Models (LLM) відкривають нові можливості для автоматизованого перетворення тексту у SQL-запити, що значно спрощує роботу з базами даних, роблячи її доступнішою для нетехнічних користувачів. Автоматичний переклад запитів природною мовою на SQL розширює спектр можливостей у сфері керування даними, зокрема для таких завдань, як моментальна генерація, агрегація та фільтрація даних. Використання міждоменного семантичного аналізу дозволяє моделі розпізнавати і коректно обробляти запити незалежно від предметної області або структури бази даних, з якою вона працює. Це значно підвищує універсальність і точність моделі при взаємодії з різними типами даних і доменами.

У роботі представлено всебічний аналіз методів Text-to-SQL на основі LLM, зокрема конструювання підказок, вибору та організації прикладів, що дозволяє виявити сильні та слабкі сторони різних підходів. На основі отриманих результатів пропонується інтегроване рішення, яке підвищує ефективність і знижує фінансові витрати на реалізацію Text-to-SQL завдання при використанні LLM. Результати можуть сприяти широкому впровадженню LLM у сфері баз даних та заохочувати подальші дослідження у цьому напрямку.

Загальні підходи в перетворенні тексту в SQL на основі LLM

Text-to-SQL має на меті автоматичний переклад запитів на природній мові в SQL-запити, що полегшує взаємодію неспеціалістів з базами даних та покращує обробку даних. Це технологічне рішення відкриває нові можливості для інтелектуальних баз даних і автоматизованого аналізу. Проте, реалізація Text-to-SQL стикається з труднощами у точному розумінні природної мови та генерації коректних SQL-запитів [1].

Дослідження у цій сфері зосереджено на підходах з використанням попередньо визначених правил та моделей машинного навчання з архітектурою кодер-декодер [1].. З розвитком глибокого навчання застосовуються різні методи, такі як механізми уваги та синтаксичний аналіз, що допомагають у розв’язанні завдання Text-to-SQL. Однією з найуспішніших моделей є BERT, яка продемонструвала відмінні результати. Щоб покращити точність, були створені великі тестові набори даних, як-от WikiSQL та Spider, що дозволили досягти прогресу в дослідженнях [1].

Останніми роками LLM, як GPT-4 та LLaMA, стали новим стандартом у обробці природної мови. LLM попередньо навчаються на величезних обсягах тексту та здатні виконувати різноманітні завдання. Основним аспектом їхньої роботи є генерування слів з найвищою ймовірністю на основі вхідних даних. Для успішного виконання Text-to-SQL важливим є ефективне формування запитів, що відомо як інженерія підказок [2].

Інженерія підказок [2] класифікується на сценарії з нульовою спробою, де не надається жодного прикладу, та сценарії з кількома спробами, коли надається обмежена кількість прикладів. Ефективне навчання в контексті дає можливість LLM виявляти патерни у запитах, що підвищує їхню здатність до генерації результатів без додаткового навчання. Проте, незважаючи на успіхи, все ще існує дефіцит досліджень, що зосереджуються на контрольованому точному налаштуванні LLM для Text-to-SQL.

Таким чином, основними аспектами Text-to-SQL на основі LLM є представлення запитів, навчання в контексті та контрольоване точне налаштування. Багато досліджень фокусуються на вилученні шаблонів SQL-запитів, однак основною проблемою є те, як підштовхнути LLM до генерації коректних SQL-запитів, що потребує детального вивчення інженерії підказок [2].

Нещодавні дослідження підтверджують важливу роль включення прикладів для ефективного навчання в контексті [3, 4].

Хоча існує багато успішних моделей, більшість з них сконцентровані на OpenAI, що залишає LLM з відкритим вихідним кодом маловивченими. Це є серйозною проблемою, адже такі моделі часто мають обмеження в розумінні контексту. Важливим завданням є поліпшення їхньої продуктивності у Text-to-SQL через контрольоване налаштування.

Ефективність використання підказок також є важливим питанням, оскільки витрати на API OpenAI можуть бути високими. При цьому інженерія підказок передбачає подання запитань, вибір прикладів та організацію прикладів [5]. Відкритими залишаються питання оптимізації довжини підказок для досягнення кращих результатів.

Існує нагальна потреба в систематичному дослідженні різних репрезентацій і вивченні того, як ефективно працювати з LLM. Щодо вибору прикладів, то поширеною практикою є кодування найбільш схожих прикладів в одному представленні з цільовим запитанням [3]. Тому дуже очікуваним є систематичне дослідження з конструювання підказок, що охоплює різні LLM, представлення запитів, вибір прикладів та організації [4].

Оперативність використання підказок залишається складним і відкритим питанням. У технологіях Text-to-SQL на основі LLM ще однією важливою проблемою є ефективність. Причина полягає в тому, що більшість попередніх досліджень зосереджені на OpenAI LLM, а виклик їхніх API є дорогим, трудомістким і обмеженим у швидкості, особливо для контекстних навчальних підказок з численними прикладами. Однак попередні дослідження не можуть добре вирішити даний виклик. Зокрема, на основі інвертованої U-подібної форми точності виконання підказок щодо довжини підказок припускається, що LLM можуть мати «золоту середину» з точки зору довжини підказок, але це все ще залишається складним відкритим питанням для дослідження [6].

Базове рішення Text-to-SQL для подальшого розгляду

У 2019 році науковці з Єльського університету представили Spider – складний набір даних для семантичного аналізу тексту і перетворення його в SQL. Spider містить понад 10 тисяч запитів і 5 тисяч унікальних SQL-запитів, що охоплюють різні домени, і вимагає від моделей здатності узагальнювати нові SQL-запити та схеми баз даних.

Spider відрізняється від попередніх наборів даних тим, що останні використовували одну базу даних, у той час як Spider має кілька баз. Результати експериментів показали, що навіть найкращі моделі досягають лише 12,4% точності, що свідчить про складність завдання.

В рамках цієї роботи буде використано таблицю лідерів Spider для вибору базового рішення Text-to-SQL. Найбільш цікаві результати демонструють моделі MiniSeek та DAIL-SQL з точністю 91,2% та 86,6% відповідно. Хоча MiniSeek не має публічного доступу, DAIL-SQL доступний для подальшого дослідження [7].

DAIL-SQL підтримує інтеграцію з різними LLM та стратегіями. У нашій роботі акцент буде на систематичному оцінюванні ефективності різних стратегій розробки, включаючи LLM з відкритим вихідним кодом. Ми плануємо порівняти варіанти відповідей у сценарії «нульового пострілу» та стратегії вибору прикладів і організації в сценарії з кількома спробами. Важливими аспектами також стануть потенціал LLM з відкритим кодом та ефективність використання токенів. Зрештою, метою є знайти збалансовану стратегію, яка оптимізує продуктивність та ефективність використання токенів, а також розробити практичне рішення на базі DAIL-SQL для реальних даних, і зробити це рішення універсальним для будь-яких доменів.

Проблематика представлення питання в контексті Text-to-SQL

Розглядаючи деяке запитання q в контексті певного домену і певної бази даних D, задачею генерування запитання є збільшення можливості LLM M сформувати коректний SQL s^* наступним чином:

$$\max_{\sigma} \quad \mathbb{P}_M\left(s^* | \sigma(q, D)\right), \tag{1}$$

де функція σ (·,·) визначає представлення для питання q, з інформацією про домен і структури БД зі схеми бази даних D. Також функція може містити додаткову інформацію інструкції, імплікацію правила та зовнішній ключ [8].

Імплікація правила (RI), Інструкція (INS), та зовнішній ключ (FK) є можливими компонентами підказки. Інструкція – це опис завдання, наприклад, «Напиши SQL як відповідь на запитання». Імплікація правила¬¬ – це наказове твердження, наприклад, «Виконай SQL-запит без пояснень». Зовнішній ключ (FK) – інформація про зовнішній ключ бази даних.

Є різні варіації підходи конструювання підказок такі як: базова підказка (BS p) (Basic Prompt), підказка представлення тексту (TR p), демонстраційний запит OpenAI (OD p), підказка представлення коду (CR p), Alpaca SFT Prompt (AS p).

Навчання в контексті: вибір та організація прикладів

У Text-to-SQL питанні, маючи набір трійок $Q = \{(q_i, s_i, D_i)\}$, де q_i і s_i - питання на природній мові і відповідний йому SQL-запит до бази даних D_i, метою навчання в контексті для Text-to-SQL є збільшення ймовірності того, що LLM M згенерує правильний SQL-запит s^* на цільове питання q і базу даних D наступним чином:

$$\max_{Q',\sigma} \quad \mathbb{P}_M\big(s^* | \sigma(q, D, Q')\big),$$

$$s.t. \quad |Q'| = k \quad and \;\; Q' \subset Q, \qquad (2)$$

де функція $\sigma(\cdot,\cdot,\cdot)$ визначає представлення для цільового питання q, з інформацією зі схеми в базі даних D та k прикладів, вибраних з Q.

При розгляданні DAIL-SQL буде робитися акцент на міждоменному Text-to-SQL, що означає, що цільова база даних D не належить до числа баз даних D, згаданих у Q, тобто, $D \notin \{D_i | (q_i, s_i, D_i) \in Q\}$. Контекстне навчання для Text-to-SQL передбачає вибір найбільш релевантних прикладів Q' і прийняття рішення про те, як переформовувати інформацію з цих вибраних прикладів у підказку.

Тобто це є дві окремі підзадачі: відбір прикладів та організація прикладів.

Вибір прикладів.

1) Випадковий вибір – це стратегія, що передбачає випадковий вибір k прикладів з доступних кандидатів [9].

2) Вибір подібних питань за маскою (Masked Question Similarity Selection(MQS)). Для міждоменного Text-to-SQL, MQS видаляє специфічно-доменну інформацію, змінюючи назви таблиць, стовпців і т.д. на лексеми-маски, а потім обчислює подібність їх вбудовування за алгоритмом kNN [10].

3) Вибір подібності питань (Question Similarity Selection, QTS). QTS вибирає число k прикладів з найбільш релевантними запитаннями, схожими по схемі. Далі він застосовує евклідову відстань до кожної пари приклад-ціль. Нарешті, алгоритм kNN використовується для вибору k прикладів з Q, які найбільш точно відповідають первинному питанню q [10].

4) Відбір за схожістю запитів (Query Similarity Selection (QRS)). QRS передбачає вибір k прикладів, схожих на цільовий SQL-запит s^*. QRS також генерує SQL-запит s^' з використанням цільового запитання q та бази даних D, де цей згенерований s^' можна розглядати як наближення до s^*. Далі запити кодуються у двійкові дискретні синтаксичні вектори відповідно до їх ключових слів. Після цього обираються k прикладів, враховуючи як схожість з наближеним запитом s', так і відмінності між обраними прикладами [9].

Стратегії, що вказані вище, концентруються на виборі прикладів на основі цільового запитання, однак, враховуючи дослідження [9] контекстне навчання являє собою навчання за аналогією. У випадку Text-to-SQL основною ціллю є формування запитів SQL на основі питання природньою мовою, відображення запитань у SQL-запити є набором навчання для LLM, тож варто враховувати як самі запитання, так і відповіді.

Організація прикладів має важливу роль у визначенні, яку саме інформацію з поданих прикладів буде сформовано у підказку. Існують два види організації: організація повної інформації та організація на основі SQL.

Повно-інформаційна організація (Full-Information Organization ¬¬– FI o) структурує приклади в представленні як цільове запитання, але відмінність закладається в тому, що замість лексеми «SELECT», наприклад, в кінці, конкретні приклади мають сформовані SQL-запити [9].

Організація, що використовує тільки SQL (SQL-Only Organization – SO o) включає SQL-запити обраних прикладів з префіксною інструкцією у підказці. Така організація має на меті збільшити кількість прикладів з мінімальною довжиною токенів [11]. Однак вона виключає інформацію про зв'язок між природнім запитанням та відповідними SQL-запитом, проте, як зазначалось раніше, такий зв'язок може бути корисним.

Підсумовуючи, Full-Information Organization відображає цілісну інформацію про приклади, тоді як SQL-Only Organization зберігає лише SQL-запити для додавання більшої кількості прикладів. В контексті дослідження важливо зрозуміти, чи існує вигідний компроміс між кількістю і якістю в організації прикладів, що може бути додатково корисним для основної задачі Text-to-SQL.

$$\max_{Q',\sigma} \quad \mathbb{P}_M\big(s^*|\sigma(q,D,Q')\big),$$

$$s.t. \quad |Q'| = k \quad and \;\; Q' \subset Q, \qquad (2)$$

де функція $\sigma(\cdot,\cdot,\cdot)$ визначає представлення для цільового питання q, з інформацією зі схеми в базі даних D та k прикладів, вибраних з Q.

При розгляданні DAIL-SQL буде робитися акцент на міждоменному Text-to-SQL, що означає, що цільова база даних D не належить до числа баз даних D, згаданих у Q, тобто, $D \notin \{D_i|(q_i,s_i,D_i) \in Q\}$. Контекстне навчання для Text-to-SQL передбачає вибір найбільш релевантних прикладів Q' і прийняття рішення про те, як переформовувати інформацію з цих вибраних прикладів у підказку.

Тобто це є дві окремі підзадачі: відбір прикладів та організація прикладів.

Вибір прикладів.

1) Випадковий вибір – це стратегія, що передбачає випадковий вибір k прикладів з доступних кандидатів [9].

2) Вибір подібних питань за маскою (Masked Question Similarity Selection(MQS)). Для міждоменного Text-to-SQL, MQS видаляє специфічно-доменну інформацію, змінюючи назви таблиць, стовпців і т.д. на лексеми-маски, а потім обчислює подібність їх вбудовування за алгоритмом kNN [10].

3) Вибір подібності питань (Question Similarity Selection, QTS). QTS вибирає число k прикладів з найбільш релевантними запитаннями, схожими по схемі. Далі він застосовує евклідову відстань до кожної пари приклад-ціль. Нарешті, алгоритм kNN використовується для вибору k прикладів з Q, які найбільш точно відповідають первинному питанню q [10].

4) Відбір за схожістю запитів (Query Similarity Selection (QRS)). QRS передбачає вибір k прикладів, схожих на цільовий SQL-запит s^*. QRS також генерує SQL-запит s^' з використанням цільового запитання q та бази даних D, де цей згенерований s^' можна розглядати як наближення до s^*. Далі запити кодуються у двійкові дискретні синтаксичні вектори відповідно до їх ключових слів. Після цього обираються k прикладів, враховуючи як схожість з наближеним запитом s', так і відмінності між обраними прикладами [9].

Стратегії, що вказані вище, концентруються на виборі прикладів на основі цільового запитання, однак, враховуючи дослідження [9] контекстне навчання являє собою навчання за аналогією. У випадку Text-to-SQL основною ціллю є формування запитів SQL на основі питання природньою мовою, відображення запитань у SQL-запити є набором навчання для LLM, тож варто враховувати як самі запитання, так і відповіді.

Організація прикладів має важливу роль у визначенні, яку саме інформацію з поданих прикладів буде сформовано у підказку. Існують два види організації: організація повної інформації та організація на основі SQL.

Повно-інформаційна організація (Full-Information Organization ¬¬– FI o) структурує приклади в представленні як цільове запитання, але відмінність закладається в тому, що замість лексеми «SELECT», наприклад, в кінці, конкретні приклади мають сформовані SQL-запити [9].

Організація, що використовує тільки SQL (SQL-Only Organization – SO o) включає SQL-запити обраних прикладів з префіксною інструкцією у підказці. Така організація має на меті збільшити кількість прикладів з мінімальною довжиною токенів [11]. Однак вона виключає інформацію про зв'язок між природнім запитанням та відповідними SQL-запитом, проте, як зазначалось раніше, такий зв'язок може бути корисним.

Підсумовуючи, Full-Information Organization відображає цілісну інформацію про приклади, тоді як SQL-Only Organization зберігає лише SQL-запити для додавання більшої кількості прикладів. В контексті дослідження важливо зрозуміти, чи існує вигідний компроміс між кількістю і якістю в організації прикладів, що може бути додатково корисним для основної задачі Text-to-SQL.

Доопрацювання DAIL-SQL

Варіантом вирішення проблем з відбором та організацією прикладів ми розглядаємо метод Text-to-SQL – DAIL-SQL (це гнучке рішення на основі LLM, яке можна розширювати та інтегрувати з іншими компонентами).

Нульовий постріл (zero-shot) — це підхід у машинному навчанні та обробці природної мови, коли модель виконує завдання без жодних прикладів або попереднього навчання на аналогічних завданнях. У цьому випадку модель намагається генерувати відповіді на основі загальних знань, закладених у її архітектуру під час попереднього тренування. При нульовому пострілі модель працює тільки на основі загальних знань, закладених під час її тренування, без прямого контексту чи зразків для поточного запиту. Це означає, що вона має розуміти завдання "з нуля" і формулювати SQL-запити, виходячи лише з розуміння мови, структури даних, а також синтаксису SQL.

Для максимізації продуктивності LLM в сценарії «нульового пострілу» є навчання в контексті, як альтернативний варіант є контрольоване доопрацювання (supervised fine-tuning), що є менш дослідженим на сьогодні. Для порівняння використовується точність збігу (EM) і точність виконання (EX). Точність збігу – збіг ключових слів SQL між прогнозованим SQL-запитом і базовою істиною. Точність виконання – це порівняння результатів виконання прогнозованого SQL-запиту з базовим SQL на тестових екземплярах бази даних [12].

Для всіх методів використовується однакова максимальна довжина питання, тобто 4096 для OpenAI LLM і 2048 для LLM з відкритим кодом. 200 токенів для генерації.

Висновки

На основі проведених експериментів можна зробити деякі емпіричні висновки та рекомендації:

- Для представлення запитань рекомендовано користуватися підказками представлення коду та демонстраційного запиту OpenAI, але інша інформація як імплікація правил та зовнішній ключ, може бути дуже корисною.
- Для вибору прикладу важлива схожість між питанням на природній мові та SQL-запитом. Ці два фактори разом є хорошим показником для розробки ефективної стратегії відбору.
- Якщо прийнята LLM є досить потужною, як GPT-4, наприклад, то представлення їм пар запитань і SQL-запитів є раціональним вибором. В іншому випадку краще представити їм повні інформаційні приклади.
- Наявність більшої кількості параметрів у LLM з відкритим вихідним кодом покращує Text-to-SQL завдання. Крім того, контрольоване доопрацювання є необхідним.

Також, у ході роботи було досліджено декілька стратегій для нульового та кількох пострілів, оцінено різні способи представлення питань, відбору та організації прикладів для LLM. Було виявлено, що використання DAIL-SQL у поєднанні з GPT-4 у сценарії з кількома пострілами дозволяє досягти найвищої точності з розглянутих, та забезпечує економічне використання токенів.

У межах базової моделі DAIL-SQL було проведено тестування стратегій використання й налаштування, і варто зазначити, що для максимізації продуктивності рекомендується сконцентруватись на питанні оптимізації підказок та відборі подібних прикладів запитань. Це дозволяє балансувати вартість і точність виконання SQL-запитів.

У результаті треба відзначити, що організація DAIL є більш економічною (вартість токенів), ніж повноінформаційний підхід, точність виконання при цьому висока (83.5% з GPT-4). Це доводить основне твердження, що представлення питань із включенням SQL у вигляді зовнішніх ключів є оптимальним як у точності, так і в економії ресурсів.

У процесі дослідження показано, що економічне та ефективне використання токенів є основною метрикою для реальних задач Text-to-SQL, враховуючи кошторис обчислень на OpenAI платформах.

СПИСОК ЛІТЕРАТУРИ

1. Katsogiannis-Meimarakis G., Koutrika G. Survey on Deep Learning Approaches for Text-to-SQL. VLDB. 2023. 32, 4. P. 905–936. URL: https://doi.org/10.1007/s00778-022-00776-8 Дата звернення: 21.08.2024.
2. A Comprehensive Evaluation of ChatGPT's Zero-Shot Text-to-SQL Capability / A. Liu et al. CoRR abs/2303. 2023. P. 13547. URL: DOI:10.1007/s00778-022-00776-8
3. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation / D. Gao et al. Proceedings of the VLDB Endowment. 2024. Vol. 17, no. 5. P. 132–1145. URL: DOI:10.14778/3641204.3641221 Дата звернення: 13.09.2024.
4. RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL / H. Li et al. *37th AAAI Conference on Artificial Intelligence*, 2023. P. 13067–13075 URL: https://doi.org/10.48550/arXiv.2302.05965
5. C3: Zero-shot Text-to-SQL with ChatGPT / X. Dong et al. 2023 URL: https://doi.org/10.48550/arXiv.2307.07306
6. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task / T. Yu et al. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. Stroudsburg, PA, USA, 2018. URL: https://yale-lily.github.io/spider Дата звернення: 10.08.2024.
7. Stanford Alpaca: An Instruction-following LLaMA model / R. Taori et al. URL: https://en.wikipedia.org/wiki/Llama_(language_model)
8. Enhancing Few-shot Text-to-SQL Capabilities of Large Language Models: A Study on Prompt Design Strategies / L. Nan et al. CoRR abs/2305.12586. 2023. URL: https://doi.org/10.48550/arXiv.2305.12586
9. What Makes Good In-Context Examples for GPT-3? / J. Liu et al. In Pro- ceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, 2022. P. 100–114. https://doi.org/10.18653/v1/2022.deelio-1.10
10. A Case-Based Reasoning Framework for Adaptive Prompting in Cross-Domain Text-to-SQL / C. Guo et al. CoRR abs/2304.13301. 2023. https://doi.org/10.48550/arXiv.2304.13301
11. Zhong R., Yu T., Klein D. Semantic Evaluation for Text-to-SQL with Distilled Test Suites. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. P. 396–411. https://doi.org/10.18653/v1/2020.emnlp-main.29

Horbachova Liudmyla Olehivna
student of the Department of Intellectual Software Systems and Technologies; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022
e-mail: xa12850503@student.karazin.ua
https://orcid.org/0000-0002-6053-7235

Khruslov Maksym Mikhailovich
Head of the Department of Computer Systems and Robotics, Candidate of Physical and Mathematical Sciences, Senior Researcher, Associate Professor; V.N. Karazin Kharkiv National University Karazin, Svobody Square, 4, Kharkiv-22, Ukraine, 61022
e-mail: maksym.khruslov@karazin.ua
https://orcid.org/0000-0001-9639-9340

Chub Olga Igorivna
Associate Professor of the Department of Computer Systems and Robotics, PhD in Economic;
V.N.Karazin Kharkiv National University
4 Svobody Square, Kharkiv-22, Ukraine, 61022
e-mail: o.i.chub@karazin.ua
https://orcid.org/0000-0002-1216-856X

Berezhnyi Artem Andriyovych
senior lecturer of the higher education institution of the Department of Computer Systems and Robotics, Master; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022
e-mail: artem.berezhnyi@karazin.ua;
https://orcid.org/0000-0001-5407-9015

Koziuberda Dmytro Oleksandrovych
Master of Cybersecurity, Faculty of Computer Science, V. N. Karazin Kharkiv National University; development employee of LADYZAYN LLC.; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022
e-mail: koziuberda.dmytro@gmail.com;
https://orcid.org/0009-0005-3088-9685

Research of the procedure for converting text into sql based on large language models (LLM) through cross-domaine semantic analysis

Theme of work. Research on the Text-to-SQL conversion procedure based on Large Language Models (LLM) through Cross-Domain Semantic Analysis. **Purpose of work.** To enhance the accuracy and adaptability of Text-to-SQL conversion using Large Language Models (LLM) through cross-domain semantic analysis, enabling reliable query interpretation across various domains and database structures. **Methods of research.** Comparative analysis, experimental evaluation, cross-domain semantic testing. **Results.** The research demonstrates that optimized prompt engineering and fine-tuning significantly improve the accuracy and cross-domain adaptability of Large Language Models for Text-to-SQL conversion. **Conclusions.** This study confirms that Large Language Models (LLMs) can effectively enhance the Text-to-SQL conversion process when optimized with targeted prompt engineering and fine-tuning. Cross-domain semantic analysis proved essential for enabling LLMs to handle varied database structures and domain-specific terminology, improving versatility and accuracy. The findings highlight the potential of LLMs to make SQL query generation more accessible to non-technical users, promoting broader application of AI in database management. Future work may focus on further refining these models to reduce computational costs and increase processing efficiency.

Keywords: *Large Language Models (LLM), Natural Language Processing (NLP), Text-to-SQL, Natural Language Processing, Deep Learning, Neural Networks, Databases, Artificial Intelligence, Data Analysis, Automation, Information Systems.*

Надійшла у першій редакції 05.04.2025, в останній - 14.05.2025.
The first version has been received on 05.04.2025, the final version - on 14.05.2025.