DOI 10.70286/ISU-15.04.2026.007

ПЕРСПЕКТИВНІ ПІДХОДИ ДО ПІДВИЩЕННЯ ТОЧНОСТІ МОДЕЛЕЙ СІМЕЙСТВА T5 ДЛЯ ПЕРЕТВОРЕННЯ ПРИРОДНОЇ МОВИ В SQL-ЗАПИТИ

Борисюк Володимир Миколайович
аспірант
Козловський Андрій Володимирович
к.т.н., доцент
Кафедра комп'ютерних наук
Вінницький національний технічний університет, Україна

Вступ. Задача автоматичного перетворення запитів, сформульованих природною мовою, у структуровані SQL-запити (технологія Text-to-SQL) є однією з фундаментальних проблем обробки природної мови (Natural Language Processing, NLP), що має високу практичну цінність для створення інтуїтивних інтерфейсів доступу до баз даних [1]. Модель T5 (Text-to-Text Transfer Transformer) є одним із найбільш універсальних рішень для цієї задачі завдяки єдиній парадигмі кодер-декодер, у якій усі завдання NLP формулюються як перетворення тексту в текст [2]. Незважаючи на значний прогрес, досягнутий спеціалізованими системами на основі T5, такими як PICARD [3], Graphix-T5 [4] та RESDSQL [5], залишаються відкритими питання оптимального використання архітектурних переваг T5 та впливу інструкційного навчання на генерацію структурованого коду.

Flan-T5 — це варіант моделі T5, що пройшов додаткове інструкційне навчання на понад 1800 задачах [6]. У попередньому дослідженні авторів було проведено перше систематичне порівняння ванільних моделей T5 та Flan-T5 у задачі Text-to-SQL на наборі даних Spider в умовах обмеженого тренувального бюджету (10 епох). Було виявлено ефект «податку на вирівнювання» (alignment tax): Flan-T5-Large показав лише 12,3 % точності повного збігу (Exact Match, EM) порівняно з 35,4 % для T5-Large, а частка синтаксично некоректних SQL-запитів зросла з 41,8 % до 76,2 %. Це свідчить про те, що загальне інструкційне навчання на природномовних задачах погіршує здатність моделі до генерації структурованого коду.

Метою цієї роботи є аналіз сучасних тенденцій у сфері Text-to-SQL та обґрунтування перспективних підходів до підвищення точності моделей сімейства T5, які можуть перевершити існуючі результати. Зокрема, розглядаються три напрямки: (1) застосування навчання з підкріпленням (Reinforcement Learning, RL) до архітектури кодер-декодер; (2) комбінація обмеженого декодування з навчанням з підкріпленням; (3) гібридне використання різних варіантів T5 для різних підзадач.

Аналіз сучасного стану. Сучасні дослідження у сфері Text-to-SQL демонструють значний зсув у бік застосування навчання з підкріпленням.

Зокрема, у 2025 році було запропоновано низку методів на основі алгоритму Group Relative Policy Optimization (GRPO) [7]: Reasoning-SQL [8] досягає 85 % точності виконання (Execution Accuracy, EX) на наборі даних Spider з моделлю Qwen2.5-Coder-14B, CSC-SQL [9] використовує коригувальну самоузгодженість через GRPO, а Graph-Reward-SQL [10] пропонує графову модель нагороди без виконання запитів. Однак усі ці роботи базуються виключно на декодерних архітектурах (Qwen, DeepSeek-Coder, LLaMA), тоді як потенціал кодер-декодерної архітектури T5 у поєднанні з RL залишається невивченим.

Єдиною роботою, що досліджує RL для T5 у задачі Text-to-SQL, є робота Berdnyk та Collery [11], де застосовано алгоритми REINFORCE та RELAX до T5-Small та T5-Base, отримавши покращення на +6,6 та +2,0 відсоткових пункти (в.п.) EM відповідно. Однак ця робота не досліджує GRPO, не використовує T5-Large та не поєднує RL з обмеженим декодуванням.

Таблиця 1. Порівняння підходів на основі RL для Text-to-SQL

Система	Backbone	EX (%)	Архіт.	RL метод
Reasoning-SQL [8]	Qwen-14B	~85	Dec	GRPO
CSC-SQL [9]	Qwen-3B	65,3	Dec	GRPO
Graph-Reward [10]	DSCoder-7B	~82	Dec	GRPO + граф
Berdnyk et al. [11]	T5-Base	+2,0	Enc-Dec	REINFORCE
T5-GRPO (пропон.)	T5-Large	?	Enc-Dec	GRPO

Як показано в таблиці 1, усі сучасні RL-підходи базуються на декодерних (Dec) архітектурах. Застосування GRPO до кодер-декодерної архітектури T5 є прогалиною, яку пропонується заповнити.

Запропоновані підходи. На основі проведеного аналізу пропонуються три перспективні підходи, кожен з яких адресує конкретну прогалину в літературі.

Підхід 1: GRPO для кодер-декодерної архітектури T5. Кодер-декодерна архітектура T5 створює унікальну можливість для RL-оптимізації: кодер фіксує контекстуалізоване представлення схеми бази даних та запиту природною мовою, тоді як декодер генерує SQL-запит авторегресивно. Пропонується застосувати GRPO з вибірковим оновленням параметрів: під час RL-оптимізації параметри кодера заморожуються, а оновлюються лише параметри декодера. Це зменшує кількість оптимізованих параметрів удвічі та стабілізує навчання, оскільки представлення запиту, засвоєне під час попереднього контрольованого донавчання (Supervised Fine-tuning, SFT), зберігається незмінним.

Функцію нагороди пропонується визначити як композитний сигнал з чотирьох компонентів: точності виконання (бінарний сигнал за результатом виконання SQL у базі даних), синтаксичної коректності (перевірка граматики SQL), точності прив'язки схеми (порівняння використаних таблиць і стовпців з еталонними) та структурної подібності (косинусна подібність SQL-скелетів). Такий багатовимірний сигнал вирішує проблему розрідженості нагороди, характерну для бінарної метрики точності виконання, та надає моделі детальніший зворотний зв'язок на кожному етапі генерації SQL.

Підхід 2: комбінація PICARD та GRPO. Метод PICARD [3] обмежує декодування моделі T5 лише синтаксично коректними SQL-токенами під час променевого пошуку (beam search), зменшуючи частку помилок виконання з 12 % до 2 %. Пропонується інтегрувати PICARD безпосередньо в процес генерації кандидатів GRPO: на кожному кроці RL модель генерує G = 8 кандидатів SQL-запитів, кожен з яких проходить через фільтр PICARD. Таким чином, усі кандидати гарантовано синтаксично коректні, а RL-оптимізація фокусується виключно на семантичній правильності запиту. Це усуває марну витрату обчислювальних ресурсів на оцінювання кандидатів, які навіть не можуть бути виконані в базі даних.

Підхід 3: гібридний пайплайн Dual-T5. Попередні експерименти авторів показали, що Flan-T5 погано генерує SQL (12,3 % EM проти 35,4 % для ванільного T5-Large), але зберігає сильні здатності до розуміння інструкцій. Пропонується використати цю асиметрію: Flan-T5-Base застосовується для першого етапу — прив'язки схеми (schema linking), де його інструкційні здатності є перевагою, а ванільний T5-Large — для другого етапу — генерації SQL-запиту на основі відфільтрованої схеми. Загальний обсяг параметрів гібридної системи (250 + 770 = 1020 млн) менший за T5-3B (3000 млн), що створює потенціал для досягнення конкурентних результатів з меншими обчислювальними витратами.

Обґрунтування очікуваних результатів. Ефективність запропонованих підходів обґрунтовується результатами суміжних досліджень. У таблиці 2 наведено порівняння прогнозованих результатів із відомими бейзлайнами.

Таблиця 2. Прогнозоване порівняння підходів на Spider (dev set)

Система	Backbone	EM (%)	EX (%)	Статус
T5-Large SFT *	T5-Large	35,4	40,0	отримано
T5-3B + PICARD [3]	T5-3B	71,9	75,1	літерат.
RESDSQL-Base [5]	T5-Base	71,7	77,9	літерат.
T5-Large + GRPO	T5-Large	60–68	70–78	прогноз
PICARD + GRPO	T5-Large	68–75	76–83	прогноз
Dual-T5	Flan+T5-L	65–72	75–82	прогноз

* — результати попередніх експериментів авторів, GPU NVIDIA Tesla T4, 10 епох навчання.

Прогноз для T5-Large + GRPO базується на результатах Berdnyk та ін. [11], які отримали +6,6 в.п. для T5-Small з RL, та на тому, що більша модель (T5-Large) має більший потенціал для покращення. Прогноз для PICARD + GRPO враховує, що усунення синтаксичних помилок (з 40 % до 2 %) дозволяє RL фокусуватися виключно на семантиці, що має дати додатковий приріст. Якщо T5-Large + PICARD + GRPO досягне ≥ 76 % EX, це перевищить результат T5-3B + PICARD (75,1 %) з моделлю у 4 рази меншою за кількістю параметрів.

Висновки. У роботі обґрунтовано три перспективні підходи до підвищення точності моделей сімейства T5 у задачі перетворення природної мови в SQL-запити.

Першою ключовою прогалиною є відсутність досліджень алгоритму GRPO для кодер-декодерної архітектури T5, тоді як усі сучасні роботи з RL для Text-to-SQL обмежені декодерними моделями. Другою прогалиною є відсутність комбінації обмеженого декодування (PICARD) з навчанням з підкріпленням, що дозволило б одночасно усунути синтаксичні помилки та покращити семантичну точність. Третьою прогалиною є невикористання асиметрії між Flan-T5 (сильне розуміння) та ванільним T5 (сильна генерація коду) для побудови гібридного пайплайну.

Попередні експерименти авторів продемонстрували, що T5-Large досягає 40,0 % EX на наборі даних Spider за 10 епох навчання. Реалізація запропонованих підходів має потенціал підвищити цей показник до 76–83 %, що дозволить T5-Large (770 млн параметрів) конкурувати з моделями T5-3B (3 000 млн параметрів), забезпечуючи значне зменшення обчислювальних витрат. Подальші дослідження будуть спрямовані на експериментальну верифікацію запропонованих підходів.

Список використаних джерел

1. Yu, T., Zhang, R., Yang, K. et al. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and Text-to-SQL task. In Empirical Methods in Natural Language Processing (EMNLP).
2. Raffel, C., Shazeer, N., Roberts, A. et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485–5551.
3. Scholak, T., Schucher, N., Bahdanau, D. (2021). PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In EMNLP, pp. 9895–9901.
4. Li, J., Hui, B., Cheng, R. et al. (2023). Graphix-T5: Mixing pre-trained transformers with graph-aware layers for Text-to-SQL parsing. In AAAI, Vol. 37, pp. 13076–13084.
5. Li, H., Zhang, J., Li, C., Chen, H. (2023). RESDSQL: Decoupling schema linking and skeleton parsing for Text-to-SQL. In AAAI, Vol. 37, pp. 13067–13075.
6. Chung, H. W., Hou, L., Longpre, S. et al. (2024). Scaling instruction-finetuned language models. The Journal of Machine Learning Research, 25(1).
7. Shao, Z., Wang, P., Zhu, Q. et al. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300.
8. Pourreza, M. et al. (2025). Reasoning-SQL: Reinforcement learning with SQL tailored partial rewards for reasoning-enhanced Text-to-SQL. In Conference on Language Modeling (COLM).
9. Sheng, L., Xu, S. (2025). CSC-SQL: Corrective self-consistency in Text-to-SQL via reinforcement learning. arXiv preprint.
10. Weng, H. et al. (2025). Graph-Reward-SQL: Execution-free reinforcement learning for Text-to-SQL via graph matching. In Findings of EMNLP.
11. Berdnyk, O., Collery, M. (2025). Fine-tuning text-to-SQL models with reinforcement-learning training objectives. Neurocomputing, 625, 129441.