DOI: 10.31276/VJSTE.2025.0089 *Mathematics and Computer Science* | *Computer Science; Computational Science*

ViR²: Beam-guided semantic and syntactic example selection for prompting Vietnamese Text-to-SQL

Dinh Minh Hoa[1, 2], Tran Thien Khai[3*]

[1]*Faculty of Information Technology, Ho Chi Minh City University of Technology, 475A Dien Bien Phu Street, Thanh My Tay Ward, Ho Chi Minh City, Vietnam*

[2]*Faculty of Information Technology, Ho Chi Minh City University of Foreign Languages - Information Technology, 828 Su Van Hanh Street, Hoa Hung Ward, Ho Chi Minh City, Vietnam*

[3]*Faculty of Information Technology, Ho Chi Minh City University of Industry and Trade, 140 Le Trong Tan Street, Tay Thanh Ward, Ho Chi Minh City, Vietnam*

Received 7 November 2025; revised 4 February 2026; accepted 26 February 2026

Abstract:

Text-to-Structured Query Language (SQL) enables natural language interfaces to databases, yet its performance in low-resource languages such as Vietnamese is limited due to scarce annotated data and translation noise. Existing few-shot prompting methods for large language models (LLMs) typically rely on semantic similarity or random sampling, which often results in redundant or syntactically misaligned in-context examples. To address this limitation, we propose Vietnamese Intelligent Retrieval and Re-ranking (ViR²), a two-stage, unsupervised example selection framework that first performs semantic retrieval and then applies beam-guided re-ranking to jointly optimise syntactic alignment and intra-set diversity. ViR² is fully language-agnostic and schema-free, making it suitable for low-resource and cross-lingual Text-to-SQL scenarios. Experiments on ViText2SQL demonstrate that ViR² achieves 23.33% Exact Match (EM) and 87.83% component-level F1, outperforming Random, Skill-KNN, ASTRES, and DICL by up to 5 EM points. Ablation studies confirm that Part of Speech (POS) -based syntactic re-ranking and beam search are the principal contributors to performance gains, while maintaining practical single query latency. By combining semantic relevance with syntactic diversity optimisation, ViR² provides a robust and extensible approach for few-shot Text-to-SQL in low-resource and multilingual environments.

Keywords: **beam-guided re-ranking, example selection, few-shot prompting, low-resource languages, Text-to-SQL.**

Classification numbers: **1.2, 1.3**

[*]*Corresponding author: Email: thientk@huit.edu.vn*

1. Introduction

Text-to-SQL translates natural language questions into executable SQL queries and is a key task in semantic parsing and database interaction. The Spider benchmark [1] remains the standard for cross-domain generalisation evaluation, with 10,181 questions and 5,693 complex SQL queries in 200 databases over 138 domains. Even state-of-the-art parsers achieved only ≈10-12% EM accuracy under schema-splitting settings, illustrating the intrinsic difficulty of Text-to-SQL on unseen schemas, where complex queries remain challenging to parse [2].

With the rise of large-scale autoregressive models such as GPT-3, few-shot prompting has revolutionised Natural Language Processing (NLP) by allowing models to achieve new tasks with only contextual examples and no updating of parameters. GPT-3 showed that very large pre-trained LMs could match fine-tuned performance on diverse tasks through prompt engineering alone [3]. In Text-to-SQL specifically, recent work has followed a chain-of-thought style prompting and structure-aware examples in improving multi-step SQL generation; example selection methods remain largely semantic-only or heuristic, without explicitly considering syntactic structure or diversity among examples.

This has a greater consequence in low-resource languages. For Vietnamese, the dataset ViText2SQL - a translation of Spider - is among the first large-scale resources enabling evaluation of Vietnamese Text-to-SQL models. Prior findings confirm that Vietnamese-specific models, such as PhoBERT, outperform multilingual counterparts when fine-tuned or prompted [4]. Since most prompting pipelines rely on semantic similarity retrieval or random sampling, they exploit little syntactic variation, resulting in suboptimal demonstration sets for complex queries or larger schemas.

To overcome these shortcomings, we propose ViR², an unsupervised retrieval strategy for few-shot Text-to-SQL prompting that is both model-agnostic and language-agnostic. The name ViR² reflects the two core stages of our approach-semantic retrieval followed by beam-guided re-ranking - designed specifically to improve example selection for Vietnamese low-resource natural language to SQL (NL2SQL) scenarios. ViR² works in two phases: a semantic retrieval phase that selects candidate examples based on embeddings, followed by a beam-guided re-ranking phase that selects a final demonstration subset based on syntactic similarity (via POS overlap) and intra-set diversity. This pipeline is implemented in ViPERSQL, a modular framework that integrates embedding modules, POS parsing (Underthesea/spaCy), various example-selection strategies including random, Skill-KNN, DICL, ASTRES, and ViR², and finally LLM prompting via configurable APIs.

Our experiments on both Vietnamese and English benchmarks, ViText2SQL and Big Bench for Large-scale database Grounded Text-to-SQL Evaluation (BIRD), respectively, show

that ViR² significantly outperforms baseline strategies regarding EM accuracy, valid SQL rate, and component-level F1 while maintaining practical latency. Extensive ablation studies confirm our suspicion that syntactic re-ranking is the leading contributor to performance gains, while diversity constraints help mitigate redundancy in the prompt examples.

By marrying semantic relevance and syntactic complementarity, ViR² advances few-shot prompting in low-resource and multilingual Text-to-SQL, thereby opening the path to more intelligent and efficient natural language interfaces to databases. Recent progress in retrieval-augmented NLP shows that search- and ranking-based strategies, such as Maximal Marginal Relevance (MMR), embedding-guided beam exploration, and RankCoT, can substantially improve context selection in complex generation tasks. Surprisingly, despite success in summarisation, question answering (QA), and code modelling, such search-based techniques have not been exploited for Text-to-SQL yet, especially in low-resource languages. Until now, NL2SQL example-selection methods have mostly fallen back on pure semantic similarity or heuristic clustering without a unified mechanism that jointly considers semantic relevance, syntactic compatibility, and intra-set diversity. In fact, this directly motivates the design of ViR², bringing a principled search-based formulation to the few-shot Text-to-SQL example selection.

The next section reviews recent prompting and example-selection strategies, which provide the context and motivation for the design of ViR².

2. Related work

This section reviews prior work on prompt-based Text-to-SQL generation and situates our method within the broader research landscape. We first review prompting strategies for large language models, with particular focus on few-shot and chain-of-thought prompting. We then examine techniques for selecting in-context examples, especially those designed to enhance semantic relevance or structural diversity. Finally, we discuss challenges unique to low-resource NL2SQL and survey search-based selection approaches, which together motivate the design of our ViR² framework.

2.1. Prompting strategies for NL2SQL

Prompting methods for NL2SQL have rapidly advanced with LLMs, commonly categorised into zero-shot, few-shot, and chain-of-thought prompting strategies.

Zero-shot prompting requires no in-context examples; the model directly generates SQL from the input question. Although convenient and broadly applicable, zero-shot methods generally perform poorly on complex or schema-wide queries. For example, the 14th Conference on Innovative Data Systems Research (CIDR 2024) report showed that naïve zero-shot

prompting achieved only ~5% SQL execution accuracy on benchmark tasks [5], underlining its limitations regarding detailed semantic understanding and schema grounding.

In contrast, few-shot prompting provides LLMs with a small set of question-SQL pairs, guiding generation with concrete examples and significantly improving performance. As summarised in recent NL2SQL surveys (2024), the choice of examples critically influences execution accuracy and generalisation across schemas [6]. In practical applications, even slight variations in example selection can lead to large performance differences, indicating that the choice of examples fundamentally determines few-shot success.

Chain-of-Thought (CoT) prompting further refines reasoning in NL2SQL by decomposing complex queries into step-by-step logic. Methods such as Divide-and-Prompt and least-to-most prompting [7] segment tasks into subtasks, enabling structured SQL generation through iterative reasoning. Experimentation on the Spider benchmark showed CoT-based prompts yield 5.2 to 6.5 point absolute improvements in execution accuracy over direct few-shot prompting, confirming the value of intermediate reasoning annotations [8]. More recent work, like ExCoT combines CoT with optimisation techniques, further enhancing execution quality and demonstrating diminishing returns for zero-shot CoT prompts, reinforcing the superior value of few-shot with reasoning paths [9]. DIN-SQL [5] follows this trend by decomposing Text-to-SQL prompting into modular steps - schema linking, Natural SQL (NatSQL) [10] transformation, and self-correction with execution feedback - achieving strong results on English benchmarks. These improvements highlight the benefits of reasoning-oriented prompts but also emphasise the importance of high-quality example selection.

2.2. Example selection techniques

Effective example selection is a critical factor in few-shot NL2SQL prompting, as the choice of in-context demonstrations largely determines SQL generation quality. The simplest approaches rely on random sampling or semantic K-Nearest Neighbors (KNN) retrieval**,** where examples are chosen based on cosine similarity in embedding space (e.g., BERT, Sentence-BERT). Although computationally efficient, semantic-only selection often retrieves topically similar yet structurally redundant examples, leading to suboptimal demonstrations for compositional SQL queries [6].

To improve semantic alignment, Skill-KNN introduces high-level “skill descriptions” describing the required SQL operations - such as joins or aggregations - and retrieves examples with similar skill profiles [11]. This reduces irrelevant matches but fails to capture syntactic patterns or token-level POS variations.

A second line of work - clustering-based retrieval, including Diversity-based In-Context Learning (DICL) and similarity-diversity sampling - aims to balance relevance and coverage by

grouping examples in embedding space. DICL explicitly re-weights clusters to favour structurally representative samples, partially mitigating redundancy [12]; however, clustering remains semantic-centric and does not model SQL compositionality directly.

To address syntactic mismatch, Z. Shen, et al. (2024) [13] propose Abstract Syntax Tree (AST)-based Reranking and Schema pruning (ASTRES), which ranks candidates via AST similarity and prunes schema elements to improve cross-schema matching. While AST-level retrieval improves structural alignment more than semantic-only methods, it still lacks a global selection mechanism that jointly optimises syntactic alignment and intra-set diversity.

Finally, several approaches are specifically tailored for batch-oriented prompting, such as Partitioning and Targeted Drilling with LLMs in Text-to-SQL (PTD-SQL) [14] or Auto-Demo-style pipelines. These methods construct drilling banks and match examples via semantic similarity across batches. Although effective for large-batch inference, they are not designed for single-query selection, and they do not incorporate explicit syntactic scoring or diversity constraints-two factors that are particularly important when K is small and schema variability is high.

Overall, existing techniques emphasise either semantic similarity, clustering-based diversity, or AST-level structural matching. However, none integrates semantic retrieval, lightweight syntactic cues (e.g., POS tags), and global diversity-aware ranking within a unified search framework, leaving a gap that motivates the design of ViR².

2.3. Limitations of existing techniques

Despite notable progress, existing example-selection methods still exhibit several limitations in the context of complex SQL generation and low-resource settings.

First, semantic-centric retrieval (KNN, clustering-based methods like DAIL-SQL [15] or similarity-diversity sampling) tends to produce example sets that are semantically close but structurally redundant, offering limited coverage over SQL operators (e.g., joins, nested queries, set operators) [16]. This becomes problematic for models that rely on few demonstrations to generalise across diverse schemas.

Second, methods that integrate structural cues-like Skill-KNN or ASTRES-eliminate some of these issues but stay incomplete: Skill-KNN relies on LLM-generated skill descriptions, disregarding syntactic cues at the POS level, while ASTRES neither models intra-set structural diversity explicitly nor performs global re-ranking over candidate subsets. Consequently, none of them truly considers the needed compositional variety for robust few-shot prompting.

While the batch-oriented prompting methods, like PTD-SQL and Auto-Demo pipelines, are efficient for large-scale inference, they are not directly applicable in single-query NL2SQL

scenarios where each decision should be optimised per question. Their reliance on semantic matching without structural selection further limits their effectiveness in low-resource settings.

Finally, none of the above methods exploit a search-based mechanism-such as beam exploration-to jointly take into account semantic relevance, syntactic compatibility, and intra-set diversity in the process of selecting K examples. The lack of unified optimisation often results in suboptimal sets of demonstrations, especially in a language like Vietnamese, where translation artifacts and schema drift further amplify syntactic mismatch.

These limitations collectively motivate the development of ViR², which combines semantic retrieval, POS-based syntactic alignment, and diversity-aware beam-guided re-ranking to provide a more comprehensive solution for low-resource NL2SQL prompting.

2.4. Low-resource NL2SQL challenges

Research on Vietnamese NL2SQL remains fundamentally constrained by the lack of native, large-scale datasets. Current efforts primarily rely on ViText2SQL, a machine-translated version of Spider, which, despite its utility, introduces substantial translation noise such as schema drift, unnatural phrase structures, and mismatches between Vietnamese expressions and database column names [1, 4]. A quantitative breakdown of these translation artefacts is provided in APPENDIX. These issues directly impair the alignment between natural-language queries and SQL semantics, making the Text-to-SQL task more fragile than in English.

Beyond ViText2SQL, several Vietnamese structured-query and structured-reasoning datasets have emerged, but none are native NL2SQL resources. Open-ViTabQA [17] contains 9,911 question-answer pairs over 329 Wikipedia tables spanning 41 domains, and is designed for table-based QA rather than SQL generation. Its questions require complex reasoning over heterogeneous table structures, including merged cells and multi-row/column alignment, and are evaluated with both answer-level F1 and the Binary Inference Fidelity (BIF) metric to capture logical consistency. Although Open-ViTabQA does not provide SQL annotations, its table-grounded reasoning closely parallels schema-grounded Text-to-SQL and highlights the need for demonstrations that are structurally diverse and robust to noisy tabular layouts-precisely the type of few-shot examples that ViR² aims to select.

Similarly, ViWiQA [18] focuses on Vietnamese open-domain QA over Wikipedia, combining hyperlink-graph-based retrieval with strong reader models on benchmarks such as UIT-ViQuAD and VIMQA. While not SQL-based, ViWiQA demonstrates that Vietnamese systems still struggle with retrieval-augmented reasoning when operating over large, noisy knowledge bases, due to sparse high-quality corpora and limited supervision. This again underscores the importance of robust example selection rather than relying solely on naive semantic similarity, especially in multilingual or translation-imperfect settings.

VIMQA [19] further stresses the complexity of Vietnamese reasoning tasks: it comprises more than 10k question-answer pairs that require multi-hop inference; the reasoning pattern resembles the compositional nature of SQL queries. Moreover, the structural variability in these datasets suggests that ViR²'s lightweight POS-based syntactic modelling and diversity-aware demonstration selection could also be beneficial for table-grounded or multi-hop QA tasks, where structural alignment and reasoning complexity strongly influence model performance through the aggregation of multiple evidence paragraphs, together with sentence-level supporting facts. Although VIMQA operates in an extractive QA paradigm, its multi-step inference model performance. A concise comparison of these Vietnamese datasets is provided in Table 1, highlighting their scale, task formulation, and structural characteristics in relation to NL2SQL.

Table 1. Overview of Vietnamese structured question answering and Text-to-SQL datasets relevant to low-resource natural language to SQL research.

Dataset	Task / Output	Notes
ViText2SQL	Text-to-SQL (SQL queries)	~7k Q-SQL pairs; 166 DBs; machine-translated from Spider; contains translation noise and schema drift.
Open-ViTabQA	Table QA	9,911 QA pairs over 329 Wikipedia tables; 41 domains; includes merged cells, multi-row alignment.
ViWiQA	Open-domain QA	Uses Vietnamese Wikipedia; emphasises retrieval-augmented QA; evaluated on UIT-ViQuAD and VIMQA.
VIMQA	Multi-hop QA	~10k multi-hop questions; requires supporting facts and multi-step inference over multiple paragraphs.

SQL: Structured Query Language; QA: question answering.

Taken together, low-resource Vietnamese NLP faces three compounding issues: (i) scarcity of native annotated NL2SQL datasets, forcing reliance on machine-translated corpora with inherent inconsistencies; (ii) weak cross-domain generalisation, due to limited training resources and translation artifacts that disrupt schema alignment; and (iii) structural and reasoning complexity, evidenced by Vietnamese table QA and multi-hop QA benchmarks, which expose the need for syntactic cues and example-set diversity. These observations align with the design philosophy of ViR², reinforcing its emphasis on semantic-syntactic complementarity and diversity-aware selection as key ingredients for robust question-to-query modelling in low-resource environments. In the following section, we introduce ViR², which operationalises these insights through an unsupervised two-stage framework that integrates semantic retrieval with lightweight syntactic cues and diversity-aware example set construction.

2.5. Search- and ranking-based selection in prompting

Different approaches, including search- and ranking-based methods, have been investigated to improve in-context example selection by balancing semantic relevance and

diversity. Techniques such as MMR have been used to reduce redundancy while keeping relevance, and ranking-aware selection has proven effective in retrieval-augmented generation and summarisation [20]. Recent work including PromptRefine [21] shows that embedding-based retrieval along with diversity-aware filtering can select few-shot informative demonstrations effectively, in particular for low-resource languages where high-quality examples are few. PromptRefine utilises example banks of related languages and employs similarity-guided ranking for the refinement of example sets. It presents how cross-lingual retrieval and selection can significantly impact few-shot performance. Meanwhile, RankCoT [22] incorporates ranking into chain-of-thought prompting for the sake of improving multi-step reasoning. Nonetheless, these are effective, without an existing effort in NL2SQL-mainly for low-resource languages-using beam-guided re-ranking which jointly considers semantic relevance, syntactic similarity and intra-set diversity. This gap has directly motivated us to design ViR², which will leverage semantic retrieval followed by beam-guided syntactic-diversity re-ranking, filling the mentioned limit and setting up the stage for the method that is discussed in the next section.

3. ViR²: Two-stage unsupervised example selection

Motivated by the limitations discussed above, we propose ViR² for in-context example selection in NL2SQL, an unsupervised and language-agnostic framework. Unlike existing methods that rely on semantic similarity or heuristic clustering, ViR² combines semantic retrieval with beam-guided re-ranking to jointly optimise semantic relevance, syntactic alignment, and intra-set diversity. Our method consists of two stages. First, semantic retrieval selects the top-M candidate examples most relevant to the input question. Second, beam-guided search explores and scores multiple example subsets to select the top-K demonstrations for prompting a large language model. This two-stage design enables ViR² to produce structurally diverse and contextually aligned prompts without requiring any labelled data or schema annotations. It is particularly suitable for low-resource and cross-lingual NL2SQL scenarios.

3.1. Problem definition

In few-shot NL2SQL prompting, the model receives a natural language question q and a pool of M candidate examples:

$$\mathcal{C} = \{C_1, C_2, \dots, C_M\} \quad (1)$$

where each candidate C_i consists of a question-SQL pair. The objective is to select a subset of K examples:

$$\mathcal{E}^* = \{E_1, E_2, \dots, E_K\} \subseteq \mathcal{C} \quad (2)$$

that best guides a LLM to generate the correct SQL query for q. We formalise example selection as an unsupervised optimisation problem:

$$\mathcal{E}^* = \text{argmax } S(\mathcal{E}, q); \mathcal{E} \subset \mathcal{C}; |\mathcal{E}| = K \quad (3)$$

where argmax $S(\mathcal{E}, q)$ is a scoring function that evaluates each candidate set based on three complementary criteria:

(1) Semantic relevance: The selected examples should be topically related to the input question.

(2) Syntactic alignment: The examples should exhibit surface-level structural similarity, measured through token-level POS-tag overlap between the question and candidate examples. This lightweight syntactic signal effectively complements semantic cues, consistent with cognitive evidence on the interplay between parsing and meaning construction [23].

(3) Intra-set diversity: The example set should cover a variety of SQL operations to avoid redundant prompts.

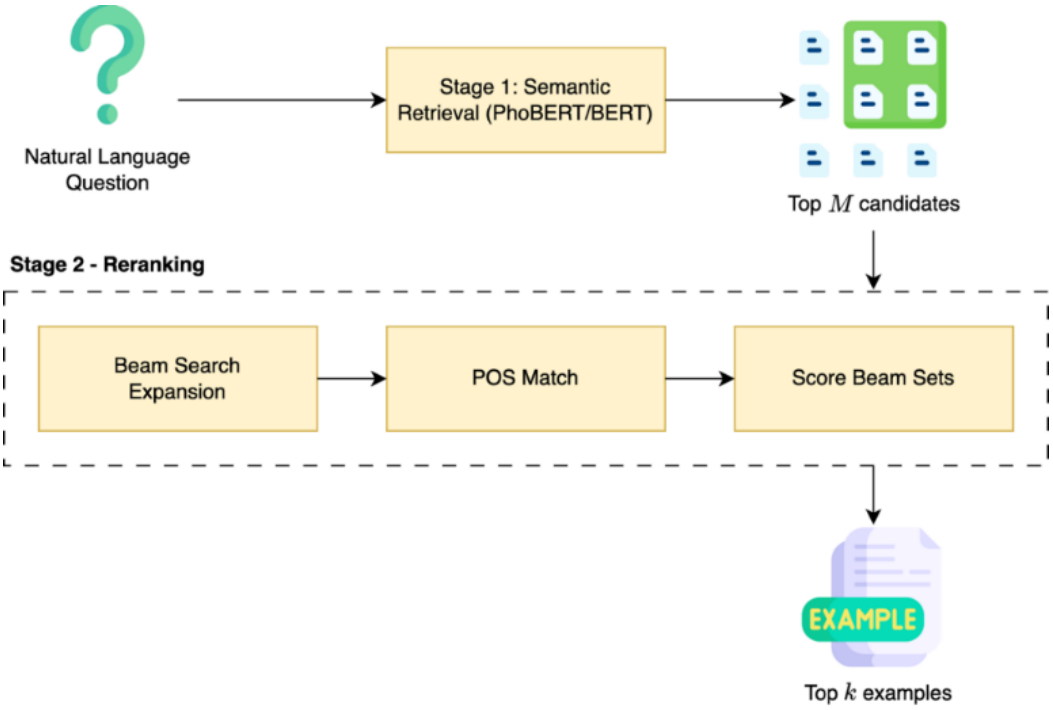

Fig. 1. Vietnamese Intelligent Retrieval and Re-ranking two-stage example selection framework.

This problem formulation is illustrated in Fig. 1, which presents the overall ViR² pipeline, and Fig. 2, which details the beam-guided re-ranking process used to explore and score candidate example sets. By combining semantic retrieval and beam-based search with syntactic and diversity scoring, ViR² efficiently identifies the most informative in-context examples for NL2SQL prompting without requiring any labelled data, schema annotations, or execution feedback.

3.2. Stage 1 - Semantic retrieval

The first stage of ViR² focuses on efficiently narrowing the candidate pool by retrieving the most semantically relevant examples for a given Vietnamese question. Each question and candidate example is encoded using PhoBERT [24], which has demonstrated strong performance in capturing both semantic and syntactic features for Vietnamese. Semantic similarity between the input question q and each candidate C_i is computed via cosine similarity:

$$\mathrm{sim}(\mathrm{q}, \mathrm{C_i}) = \frac{\mathrm{h_q} \cdot \mathrm{h_i}}{\|\mathrm{h_q}\| \cdot \|\mathrm{h_i}\|} \quad (4)$$

where h_q and h_i are the PhoBERT embeddings of the input question and the candidate example.

The top-M examples with the highest similarity scores are retained as the retrieval pool:

$$\mathcal{C}_{\text{top-M}} = \mathrm{TopM}\big(\mathcal{C}, \mathrm{sim}(\mathrm{q}, \mathrm{C_i})\big) \quad (5)$$

which is then passed to Stage 2 - Beam-guided re-ranking for structural and diversity-aware selection (see Fig. 1).

By design, this semantic retrieval step is lightweight and unsupervised, requiring no schema annotations or SQL execution feedback. While ViR² is developed primarily for Vietnamese NL2SQL, its adaptability to English is demonstrated later in Section 5 through ablation studies.

3.3. Stage 2 - Beam-guided re-ranking

After Stage 1 produces a semantically relevant retrieval pool $\mathcal{C}_{\text{top-}M}$, Stage 2 performs beam-guided re-ranking to select the final top-K examples. Instead of relying on greedy selection, beam search systematically explores multiple candidate subsets to balance semantic relevance with structural compatibility. In our framework, this structural signal is operationalised through lightweight token-level POS-tag overlap, which provides robust syntactic cues during example set construction. Such joint semantic-syntactic guidance is essential, as LLMs often fail to make consistent use of long prompts when relevant demonstrations are buried or structurally mismatched [24].

At each step, beam search expands partial sets by adding one new candidate example and evaluates each beam using a composite scoring function:

$$S(\mathcal{E}, q) = \lambda_{sem} \cdot Sem(\mathcal{E}, q) + \lambda_{syn} \cdot Syn(\mathcal{E}, q) + \lambda_{div} \cdot Div(\mathcal{E}) \quad (6)$$

where $Sem(\mathcal{E}, q)$ means cosine similarity between q and the examples in $\mathcal{E}$; $Syn(\mathcal{E}, q)$: means syntactic alignment measured via POS-tag overlap. $Div(\mathcal{E})$ means intra-set diversity, measured

by 1- average pairwise cosine similarity among selected examples; λ_{sem}, $\lambda_{syn}, \lambda_{div}$ are non-negative weights balancing the three components.

In practice, we fix $\lambda_{\text{sem}} = 1$ and $\lambda_{\text{syn}} = 1$, and tune λ_{div} using a small grid search over $\{0.1, 0.3, 0.5\}$ on a held-out portion of the ViText2SQL development set, selecting $\lambda_{\text{div}} = 0.3$ as the best trade-off between syntactic alignment and diversity. This value is then kept fixed for all experiments in Sections 4-5.

For a question q and a candidate example C_i, we compute:

$$\text{POS\_matching}(\text{q}, \text{C}_\text{i}) = \frac{1}{|\text{q}|}\sum_{\text{t} \in \text{q}} 1[\text{POS}(\text{t}) \in \text{POS}(\text{C}_\text{i})] \quad (7)$$

where $POS(t)$ is the POS tag of token t and $POS(C_i)$ denotes the multiset of POS tags appearing in C_i. This normalised overlap score provides a lightweight approximation of syntactic similarity that is robust for short NL2SQL queries. Higher values indicate stronger structural correspondence.

Beam search proceeds layer by layer (see Fig. 2):

- Initialisation: Each beam starts with a single example from $\mathcal{C}_{top-M}$.

- Expansion: Beams are expanded by adding the next candidate example.

- Scoring & Pruning: Each beam $\mathcal{E}$ is scored using $S(\mathcal{E}, q)$; only the top-B beams are kept.

- Termination: After K steps, the best-scored beam $\mathcal{E}^*$ is returned as the final top-K example set.

Formally, Stage 2 optimises the same objective defined in Eq. (3), but the search is constrained to the semantically retrieved pool $\mathcal{C}_{\text{top-}M}$. Beam search incrementally expands partial sets and prunes low-scoring beams based on the composite score $S(\mathcal{E}, q)$, efficiently yielding the final top-K examples.

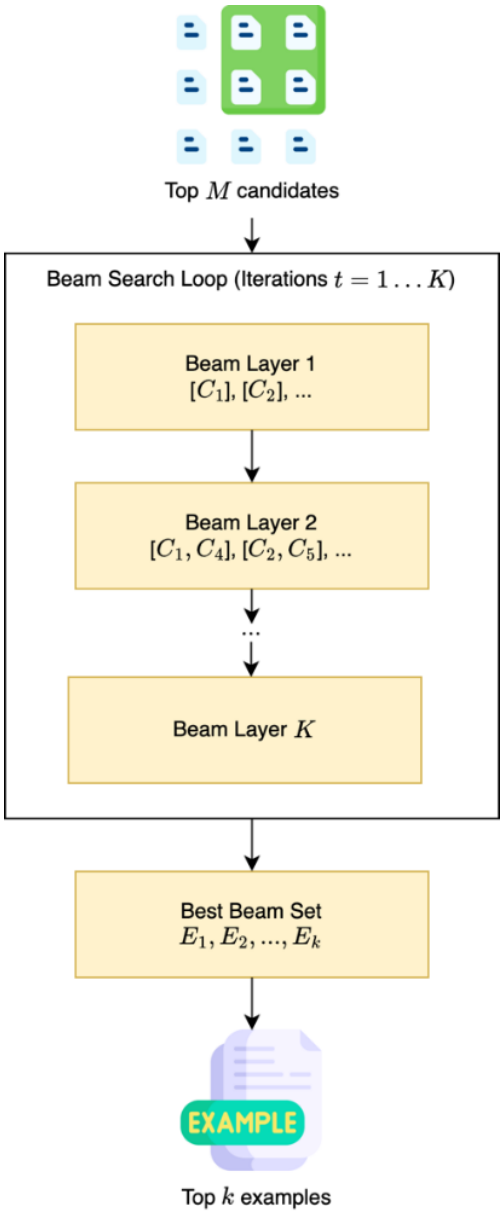

Fig. 2. Beam-guided expand-score-prune loop for final example-set selection in Vietnamese Intelligent Retrieval and Re-ranking.

3.4. Implementation notes

ViR² is implemented as a modular and lightweight example selection framework, designed to integrate seamlessly with NL2SQL pipelines such as ViPERSQL. The framework exposes a unified interface:

selected_examples = select(question=q, K=K)

which encapsulates both Stage 1: Semantic retrieval and Stage 2: Beam-guided re-ranking. This design allows the system to remain agnostic to the underlying LLM and easily compatible with different prompting strategies, including standard few-shot and CoT prompting.

From an engineering perspective, ViR² has several notable characteristics:

Unsupervised and schema-free: The method does not require any labelled training data, database schema annotations, or SQL execution feedback. This makes it particularly suitable for low-resource settings, where annotated NL2SQL datasets are scarce.

Modular architecture: Embedding models and syntactic parsers are implemented as pluggable components. In our default Vietnamese configuration, we use PhoBERT for semantic encoding and a POS-tagger to obtain the token-level POS tags required for syntactic overlap scoring in Stage 2. These modules can be seamlessly replaced with English or multilingual counterparts, enabling cross-lingual and ablation studies without modifying the core ViR² algorithm (see Section 5).

Computational efficiency: By first narrowing candidates via semantic retrieval and then applying beam-guided pruning, the method avoids the combinatorial explosion of $\binom{M}{K}$ subsets while still achieving high structural coverage.

Extensibility: The framework can integrate additional scoring components (e.g., schema-linking confidence) or alternative search heuristics without altering its core interface.

Combined with the two-stage pipeline illustrated in Fig. 1 and the beam search mechanism shown in Fig. 2, ViR² provides a practical and scalable solution for unsupervised example selection in few-shot NL2SQL prompting. Its design enables straightforward adaptation to new languages and domains, which is further validated in our experiments and ablation studies in Section 5.

Open-source availability: The full implementation of ViR² and all experiment scripts are publicly released on GitHub[†].

4. Experimental setup

We investigated the effectiveness of ViR² for low-resource NL2SQL prompting through experiments conducted in the Vietnamese setting, where labelled datasets are limited, and translation artefacts affect SQL generation. We first evaluate ViR² on the ViText2SQL benchmark in its original target language. Then, we extend experiments to English datasets, including BIRD, to demonstrate cross-lingual generalisation. In this section, the overall experimental setup is summarised, covering datasets, models, prompting strategies, and evaluation metrics.

[†]https://github.com/hoadm-net/ViPERSQL

4.1. Datasets

Research in Vietnamese NL2SQL suffers from the severe lack of data, where no large-scale native datasets like Spider or BIRD exist in English. We present results focusing on ViText2SQL, supplemented with BIRD for some cross-lingual experiments.

ViText2SQL is a Vietnamese Text-to-SQL dataset translated from Spider, which involves around 7k question-SQL pairs across 166 databases. It remains the only publicly available Vietnamese NL2SQL benchmark of reasonable quality. The dataset retains Spider's cross-domain query diversity while introducing an additional layer of translation noise: schema mismatches and lexical artifacts. In this sense, it forms an ideal benchmark for testing the performance of ViR² under low-resource conditions. However, since it relies on machine translation, ViText2SQL does not actually represent the native linguistic patterns in Vietnamese. We will later discuss it in more detail in our cross-lingual and translation-robustness analysis, Section 5.4.

We adopt the BIRD (dev) split for cross-lingual validation. Since BIRD does not offer a public test set, the dev split is a standard choice for evaluation. In order to avoid overfitting to high-resource English benchmarks and also maintain consistency with our low-resource setting, we evaluate ViR² on a reproducible subset of 500 queries, sampled using a fixed random seed (seed=42) and uniform sampling per database in order to maintain schema diversity. This controlled sampling strategy ensures that our cross-lingual evaluation reflects structural diversity rather than dataset scale, thereby enabling clearer attribution of performance differences to example-selection behaviour instead of resource imbalance.

To put the Vietnamese resource landscape into perspective, we also draw on three structured-reasoning datasets-Open-ViTabQA, ViWiQA, and VIMQA-though these are not directly used for evaluation. Open-ViTabQA points out, for instance, the challenges of table-grounded reasoning with, among others, merged cells and multi-row alignment. ViWiQA points to retrieval-augmented limitations for Vietnamese open-domain QA, while VIMQA illustrates the difficulty of multi-hop compositional reasoning. Taken together, these limitations highlight the lack of native NL2SQL resources and motivate our experiments to focus on the closest available benchmark, ViText2SQL.

A summary of both datasets is provided in Table 2, and the SQL query type distribution of ViText2SQL is analysed in Table 3 to highlight the skew toward complex queries.

Table 2. Summary of datasets.

Dataset	Language	Origin	#Questions	#Databases
ViText2SQL	Vietnamese	Spider (MT)	7,000	166
BIRD (dev)	English	Native	10,000+	600+

Table 3. Structured Query Language query type distribution in ViText2SQL.

SQL type	Description	Count	Percentage (%)
SET_OP	UNION/INTERSECT/EXCEPT	419	6.1
SUBQUERY	Contains subqueries	764	11.2
JOIN	Multi-table join operations	2,582	37.8
GROUP	Includes GROUP BY	666	9.7
AGG	Aggregation (COUNT, SUM, AVG, etc.)	774	11.3
ORDER	Includes ORDER BY	527	7.7
SELECT_WHERE	Simple SELECT with WHERE	860	12.6
SELECT_ONLY	Simple SELECT without conditions	239	3.5

SQL: Structured Query Language.

The ViText2SQL dataset is heavily skewed toward JOIN queries, with relatively few simple SELECT statements. This pattern underlines the need for diverse example selection to cover a range of SQL structures, as detailed in Table 3.

Overall, ViText2SQL is small, translation-based, and dominated by complex queries, which makes it ideal for evaluating unsupervised example selection in low-resource scenarios. In contrast, BIRD (dev) provides a native English benchmark that we sample partially to demonstrate the cross-lingual compatibility of ViR² with an eye toward full high-resource optimisation.

4.2. Language models and tools

Following the focus of ViR² on Vietnamese NL2SQL, our default configuration uses PhoBERT-base-v2 for semantic embeddings and Underthesea for POS tagging. PhoBERT has been shown to effectively capture Vietnamese lexical and syntactic properties. Underthesea tokenises and brings in POS annotations that are consistent between colloquial and formal Vietnamese. Importantly, neither module requires additional fine-tuning since ViR² operates in a purely unsupervised setting.

To investigate cross-lingual generalisation, we adapt ViR² to English by replacing PhoBERT with BERT-base-uncased for semantic encoding and Underthesea with the lightweight

spaCy en_core_web_sm model for POS tagging. We deliberately use the small spaCy model for two reasons: (i) it reduces latency by an order of magnitude during beam-guided re-ranking, and (ii) pilot experiments with medium and large variants show negligible performance differences (<0.3 EM) for our syntactic-overlap scorer, which relies only on coarse token-level POS categories rather than fine-grained dependency structures. This observation aligns with prior findings that in-context retrieval benefits more from high-quality embeddings than from heavy syntactic models.

Table 4 summarises the embedding and parsing tools used in our Vietnamese and English setups.

Table 4. Embedding models and Part of Speech parsers.

Language	Embedding model	POS parser	Usage scope
Vietnamese	PhoBERT-base-v2 (VinAI)	Underthesea	Default ViR² configuration
English	BERT-base-uncased	spaCy (en_core_web_sm)	Cross-lingual ablation

POS: Part of Speech; ViR²: Vietnamese Intelligent Retrieval and Re-ranking.

For Stage 1 (semantic retrieval), sentence-level embeddings are extracted using the corresponding BERT/PhoBERT model, and cosine similarity is applied to rank candidate examples. For Stage 2, POS tags from the associated parser contribute to syntactic-alignment scoring in the composite function $S(\mathcal{E}, q)$ (Eq. 6). This modular architecture allows ViR² to switch between languages without retraining and without any schema-dependent supervision, making it suitable for low-resource NL2SQL scenarios and cross-lingual evaluation.

For SQL generation, we use a single LLM configuration across all experiments. Unless otherwise stated, ViR² generates SQL with the GPT-4.1-mini model (OpenAI, May-2025 snapshot), accessed via the ViPERSQL prompting interface with temperature = 0.1, top-p = 1.0, and a maximum output length of 2,000 tokens. Chain-of-thought decoding is disabled unless explicitly evaluated (e.g., zero-shot CoT in Section 5). Although ViPERSQL supports alternative backends such as GPT-4.1-Turbo and Claude 3.5-Sonnet, these models are not used in the results reported in this article.

4.3. Prompting strategies

We evaluate ViR² under three prompting paradigms for Text-to-SQL: zero-shot, few-shot, and CoT. These settings are chosen to assess both baseline performance and the impact of example selection on low-resource NL2SQL. All few-shot experiments use the same candidate pool size $M = 50$, top-$K = 3$ examples for prompting, and a beam size of 5 in Stage 2 of ViR², unless otherwise stated.

Zero-shot prompting: The LLM generates SQL queries without any in-context examples. This serves as a lower-bound baseline, generally effective only for frequent or simple query types.

Few-shot prompting: The LLM is provided with K question-SQL example pairs as in-context demonstrations. We evaluate four example selection strategies under identical M and K configurations:

- Random: Uniformly sampled examples.

- Skill-KNN [11]: Retrieval based on skill-oriented descriptions.

- DICL [12]: Clustering-based selection for structural diversity.

- ViR² (ours): Two-stage selection combining semantic retrieval and beam-guided syntactic-diversity re-ranking.

By using identical M and K across methods, we ensure fair comparison and isolate the impact of example selection.

CoT prompting: We apply CoT prompting to encourage step-by-step reasoning in Text-to-SQL generation. CoT is tested in both zero-shot and few-shot settings, allowing us to evaluate whether structurally diverse examples selected by ViR² further enhance reasoning performance.

To ensure reproducibility, we summarise all LLM-related hyperparameters used in our experiments-including model backend, decoding configuration, and maximum output length-in Table 5 below. These parameters remain fixed across all prompting strategies and languages unless otherwise specified.

Table 5. Large language model hyperparameters for reproducibility.

Parameter	Value	Notes
LLM backend	GPT-4.1-mini	Used for all SQL generation experiments
Temperature	0.1	Stable decoding for deterministic SQL structure
Top-p	1.0	Default greedy-sampling hybrid
Max output length	2,000 tokens	Ensures full SQL generation without truncation
CoT mode	Disabled unless specified	Enabled only in CoT experiments
Random seed	42	Controls reproducibility of few-shot sampling

LLM: large language model; SQL: Structured Query Language; CoT: Chain-of-Thought.

4.4. Evaluation metrics

Our evaluation is based on the metrics adopted in the original ViText2SQL benchmark, focusing on SQL generation quality rather than execution accuracy due to the lack of public databases. We report three complementary metrics, including EM, Valid SQL Ratio, and Component-level F1, summarised in Table 6. In this regard, EM measures the percentage of the predicted SQL queries exactly matching the reference SQL string after canonical formatting. This metric is strict and sensitive to table alias variations, ordering of conditions, and formatting choices. Valid SQL Ratio measures the proportion of syntactically valid SQL queries that can be parsed by a SQL parser. Reflecting structural correctness, it fails to capture semantic faithfulness.

Table 6. Evaluation metrics.

Metric	Description	Purpose
EM	% of predicted SQL queries identical to gold SQL	Measures full query generation accuracy
Valid SQL ratio	% of syntactically valid and parsable SQL queries	Evaluates structural correctness
F1-score	F1 for major clauses (SELECT, WHERE, GROUP BY, etc.)	Captures partial correctness and coverage

EM: Exact Match; SQL: Structured Query Language.

For instance, Component-level F1 - which has now been better defined - is a more fine-grained assessment by calculating precision and recall over major SQL clauses. According to the ViText2SQL specification, every SQL query is decomposed into clause-level components, such as SELECT, WHERE, GROUP BY, ORDER BY, aggregation functions, JOIN conditions, etc., and F1 is computed against predicted components to gold components. This captures partial correctness even when the full SQL string does not exactly match and thus is more robust to aliasing and formatting differences in low-resource settings.

In addition, because ViR² introduces a two-stage retrieval and re-ranking pipeline, we also measure end-to-end latency (semantic retrieval + re-ranking + LLM inference) to reflect the practical feasibility of each example-selection strategy. Latency is averaged over all queries and reported in seconds.

Together, these metrics jointly evaluate accuracy, structure, and practical usability, providing a fair and reproducible comparison between ViR² and all baselines.

5. Results and analysis

We now report the experimental results of ViR² on both Vietnamese and English NL2SQL benchmarks. This section evaluates the framework across the prompting strategies and example selection methods introduced in Section 4, focusing on low-resource settings and cross-lingual generalisation. First, we report the overall accuracy and validity of SQL generation on ViText2SQL and then compare the baselines on BIRD to show the language-agnostic adaptability. Additional analyses, including ablation studies and latency measurements, provide further insight into the contribution of semantic retrieval, syntactic alignment, and diversity-aware re-ranking.

5.1. Prompting strategy comparison (ViText2SQL)

We start by running several baseline prompting paradigms on ViText2SQL: zero-shot, random few-shot, and CoT prompting. In the few-shot setting, examples are randomly sampled, while CoT is run both without in-context examples (zero-shot CoT) and with random few-shot demonstrations. For completeness, we also include a fine-tuned baseline from the original study about ViText2SQL [4]. The results in terms of EM, Component-Level F1, and latency are in Table 7.

Table 7. Prompting strategy performance on ViText2SQL.

Prompting strategy	EM (%)	Component-level F1 (%)	Latency (s)
Baseline	53.3	74.36	n/a
Zero-Shot	10.12	81.50	2.425
Few-shot (Random)	18.33	86.04	3.312
CoT (w/o examples)	6.67	81.63	11.562
CoT (with random examples)	18.21	83.74	11.512

EM: Exact Match; CoT: Chain-of-Thought.

Compared with the fine-tuned baseline, GPT-based prompting exhibits a different performance profile, achieving high component-level F1 but relatively low EM. This gap arises because LLMs often generate semantically correct queries that differ from the reference SQL in table or column aliasing, which reduces exact string matching accuracy. While the zero-shot prompting only achieves 10.12% EM, even random few-shot examples almost double the EM and elevate F1 to 86.04%, which indicates the minimal context helps the model align better with schema structure.

CoT reasoning gives a mixed result. Zero-Shot CoT reaches similar F1 to the standard zero-shot but drops in EM, underlining that reasoning itself cannot solve the schema grounding

problem. Furthermore, this setting is not applicable in real life due to its inference latency of 11.562 s/query. When CoT is combined with randomly sampled demonstrations ("CoT with random examples"), EM substantially improves (from 6.67 to 18.21%), but F1 only increases modestly. Again, it confirms that CoT benefits from contextual cues, yet random demonstrations still lack syntactic alignment for robust schema grounding. These observations again point out that semantically rich queries, which are syntactically unaligned, remain a bottleneck; hence, structurally diverse and syntactically compatible few-shot examples are needed to help improve both EM and F1 results when the resource is low.

5.2. Example selection performance

We next evaluate the effect of different example selection strategies for few-shot prompting on ViText2SQL, using a candidate pool size of $M = 50$ and $K = 3$ in-context examples. Four representative baselines are considered-Random, Skill-KNN, ASTRES, and DICL-alongside ViR² (ours), which combines semantic retrieval with beam-guided syntactic-diversity re-ranking.

Figure 3 shows the EM results for all strategies. EM is the most informative metric for this comparison because it directly reflects the model's ability to generate canonical SQL queries consistent with reference outputs.

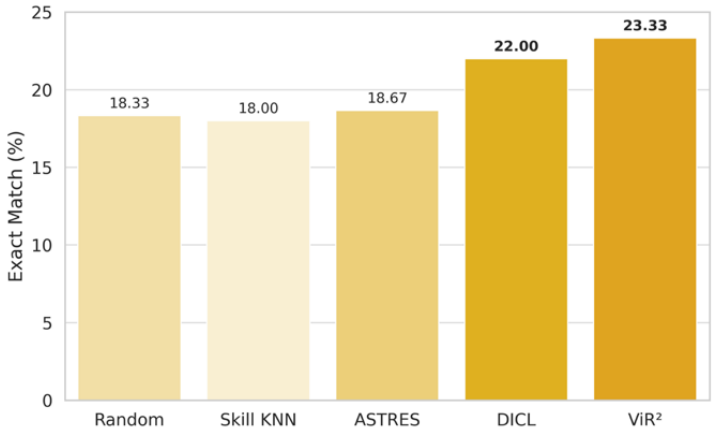

Fig. 3. Exact Match (%) of example selection strategies on ViText2SQL.

From this figure, several observations emerge: First, the highest EM is 23.33% by ViR², which clearly outperforms all baselines, confirming that semantic relevance, syntactic alignment, and intra-set diversity should be combined for effective few-shot prompting in low-resource NL2SQL. DICL has a rank of second with 22.00%, benefiting from structural clustering but still without global beam-guided optimisation. ASTRES improves over random and Skill-KNN slightly due to its AST-level alignment but lacks diversity-aware re-ranking, thus allowing it only a small EM gain.

In contrast, ViR² consistently finds example sets that are semantically relevant yet structurally complementary, thereby yielding higher EM without prohibitive latency. These results demonstrate that the quality and structural diversity of in-context examples are more critical than quantity, which validates the design choices of ViR² for low-resource Text-to-SQL.

5.3. Ablation study

To analyse the contribution of each component in more detail, we perform an ablation study by removing modules one at a time from the full system. In particular, we assess three ablated variants: removing the POS-based syntactic alignment ("No POS"), removing diversity-aware scoring ("No Diversity"), and replacing beam search with greedy selection ("No Beam"). All experiments are run under identical settings on ViText2SQL with $\mathrm{M} = 50, \mathrm{K} = 3$.

Table 8 shows the ablation results where removal of each module decreases the scores of EM and F1.

Table 8. Ablation study on ViR² components (ViText2SQL).

Model	Exact Match	F1
ViR²	23.33	87.83
ViR² No POS	21.67	86.70
ViR² No Diversity	23.33	87.25
ViR² No Beam	22.67	87.66

The results show that the full ViR² model achieves the highest EM and F1, confirming the benefit of integrating all components:

- Removing POS-level matching leads to a clear drop in both EM and F1, indicating its importance for aligning examples with the target query syntactically.

- Disabling the diversity module maintains EM but slightly reduces F1, showing that structural variety improves clause-level correctness.

- Replacing beam search with greedy top-K selection causes moderate degradation in both metrics, confirming that beam exploration helps avoid suboptimal example sets.

Overall, each module contributes to the final performance, and the full combination provides the most consistent improvements, validating ViR²'s design as a modular yet synergistic framework for low-resource NL2SQL.

5.4. Cross-lingual generalisation and GPT-translated data analysis

To assess the language-agnostic capability of ViR², we perform two complementary cross-lingual evaluations on the BIRD development set: (i) applying ViR² directly to English queries,

and (ii) translating BIRD queries into Vietnamese to simulate a low-resource scenario where English NL2SQL datasets are adapted into Vietnamese. As dataset characteristics have already been described in Section 4.1, we focus here solely on how ViR² behaves under cross-lingual and translation-induced variation.

We evaluate three settings:

(1) ViR² (En):

a) English BIRD dev queries processed using BERT-base-uncased for semantic retrieval and spaCy (en_core_web_sm) for POS tagging.

b) This setting isolates ViR²'s cross-lingual generalisation without involving translation.

(2) ViR²-Random (Vi):

a) Vietnamese queries produced by GPT-4.1-mini translation.

b) In this ablation variant, Stage-1 semantic retrieval is preserved, but Stage-2 syntactic re-ranking is disabled: the final K demonstrations are sampled uniformly at random from the retrieved candidate pool $C_{\text{top-}M}$, without applying POS-based alignment or diversity scoring.

c) This baseline isolates the contribution of ViR²'s re-ranking module and shows how translated NL2SQL data behaves without structured example selection.

(3) ViR² (Vi):

a) The same GPT-translated Vietnamese queries evaluated with the full ViR² pipeline-PhoBERT embeddings + Underthesea POS tags.

b) This setting tests whether ViR² can mitigate the instability introduced by machine translation.

Translation process: We translate each English query into Vietnamese using GPT-4.1-mini with a constrained format preserving SQL-linked entities whenever possible. Although high-quality, the translated queries frequently exhibit schema drift, unnatural phrasing, or inconsistent terminology relative to column names-issues consistent with those previously observed in ViText2SQL. A detailed quantitative audit of these translation artefacts is provided in APPENDIX. This analysis allows us to specifically attribute performance drops to translation-induced variability rather than dataset-specific factors already covered earlier.

Table 9 reports the EM and Component-F1 results for all three cross-lingual configurations.

Table 9. Cross-lingual and GPT-translated performance on Big Bench for Large-scale database Grounded Text-to-SQL Evaluation (Exact Match and F1 in %).

Model	Exact Match	F1
ViR² (En)	1.78	79.25
ViR² Random (Vi)	4.32	79.43
ViR² (Vi)	1.26	78.57

ViR²: Vietnamese Intelligent Retrieval and Re-ranking.

These results yield three main observations: First, ViR² generalises to English with 1.78% EM and 79.25% F1 under the setting of ViR² (En). Although low in absolute EM-consistent with the difficulty of BIRD-this confirms that ViR²'s semantic-syntactic scoring framework transfers across languages with minimal modifications. Second, GPT-translated Vietnamese queries perform poorly under random example selection with 4.32% EM despite a comparable component-level F1 to English. This indicates that machine translation introduces structural inconsistencies that disproportionately affect EM. Third, applying ViR² on translated Vietnamese queries results in an EM drop to 1.26%, which suggests that translation artifacts hinder both interpretation and example-selection quality. However, the relatively stable F1 score across all settings indicates that ViR² maintains clause-level correctness even when translation noise disrupts global SQL structure. These findings together highlight that: (i) ViR² is language-agnostic; (ii) translation-based data augmentation is not reliable for NL2SQL; and (iii) high-quality, native Vietnamese resources continue to remain indispensable for robust Text-to-SQL research.

6. Discussion

Our experiments provide important insights into the effectiveness and generalisability of ViR². First, the results on ViText2SQL indicate that ViR² outperforms both the baseline prompting strategies and the existing example selection methods. This convincingly demonstrates that the combination of semantic retrieval, POS-based syntactic alignment, diversity scoring, and beam-guided re-ranking forms an effective approach to few-shot Text-to-SQL in low-resource settings. These ablation studies further indicate that each component contributes to overall performance and that syntactic alignment and beam exploration are the most important factors for improving EM accuracy.

Another exciting insight comes from the cross-lingual experiments: ViR² can be easily adapted to English, for which only PhoBERT and Underthesea are replaced with BERT and spaCy, respectively, proving the language-agnostic capabilities of the framework. Simultaneously, experiments on GPT-translated BIRD queries show that automatic translation

introduces schema mismatch, semantic drift, and unnatural phrasing, significantly lowering the scores of EM despite the relatively stable component-level F1 scores. The results demonstrate that machine translation cannot replace high-quality native NL2SQL datasets in low-resource settings.

Despite these promising findings, several limitations still remain: ViR² relies on pre-trained embeddings and POS parsers that can inherit domain or language biases. Furthermore, the present framework does not make use of schema execution feedback nor database content; as a result, it cannot disambiguate queries or resolve aliasing automatically. Overall, these observations indicate that careful in-context example selection and language-sensitive prompt engineering are crucial for building reliable Text-to-SQL systems in low-resource and multilingual settings.

7. Conclusions and future work

This article introduced ViR², a two-stage unsupervised framework for example selection in few-shot Text-to-SQL, tailored for low-resource and multilingual scenarios. ViR² first performs semantic retrieval to select candidate examples and then applies beam-guided re-ranking that considers both POS-based syntactic alignment and structural diversity. Furthermore, ViR² is integrated into our unified Text-to-SQL framework, ViPERSQL, which provides a modular and extensible platform that supports multiple retrieval strategies, embedding backbones, and prompting configurations, thereby facilitating reproducible research and practical deployment across languages. Extensive experiments in ViText2SQL prove that ViR² consistently achieves higher EM and component-level F1 than baseline prompting and existing example selection methods. Ablation studies confirm the contribution of each module to performance, and cross-lingual evaluations demonstrate that ViR² generalises to English. Furthermore, experiments with GPT-translated Vietnamese queries show that automatic translations introduce schema and semantic inconsistencies, which again highlights the importance of high-quality contextual examples when resources are limited.

Despite these modest cross-lingual scores, ViR² remains practically valuable because it operates without supervision, schema annotations, or language-specific resources and can therefore function reliably even with noisy or translated data. Its ability to consistently improve example selection under such constraints makes it directly applicable to real-world multilingual NLIDB systems where high-quality native Text-to-SQL data are often absent.

Future work will focus on three directions.

First, we aim to incorporate schema-aware scoring and execution feedback to further improve query disambiguation and handle aliasing automatically.

Second, we plan to explore adaptive beam and diversity control, allowing the framework to dynamically adjust to different database schemas and query complexity.

Finally, lightweight multilingual adaptation and embedding alignment will be investigated to enhance cross-lingual performance without requiring high-quality parallel datasets.

By combining language-agnostic design, structural reasoning, and unsupervised example selection, ViR² provides a practical and extensible approach for robust Text-to-SQL in low-resource and multilingual environments, forming a solid foundation for future research in multilingual natural language interfaces to databases.

APPENDIX: Quantitative Audit of Translation Noise in ViText2SQL

To assess how translation artefacts may affect Text-to-SQL performance, we conducted a lenient-criteria audit of 700 randomly sampled questions (≈10% of the ViText2SQL training split). Only severe and unambiguous issues were counted-minor synonym choices or stylistic differences were not considered errors.

Issue type	Rate	Description
Schema Drift (SCH)	13.9%	Question uses natural Vietnamese terms that do not align with mixed-language schema names.
Entity Mismatch (ENT)	5.3%	EN-VI inconsistency in literal values (titles, room names, product names).
Structural Mismatch (STR)	5.0%	Question phrasing underspecifies or misleads the SQL structure.
Lexical Issues (LEX)	0.4%	Minor typos or awkward phrasing caused by translation.
No Issues (OK)	81.3%	Translation is good and semantically faithful.

These findings show that ViText2SQL remains broadly usable but contains non-trivial translation artefacts (18.7%) that can affect schema alignment and structural interpretation. This supports our argument that native datasets are preferable, and that ViR²'s syntactic-aware example selection helps mitigate such inconsistencies.

Representative examples from the audit

Question: "Tìm những người bạn có giới tính nữ của Alice."

SQL: ... WHERE t2.tên = "Alice" AND t1.giới_tính = "female"

Issue: "những người bạn" (colloquial) vs schema column "bạn_bè".

→ Natural phrasing mismatches schema terminology.

Question: "Cho biết trang trí trong phòng 'Ẩn dật và thách thức'"

SQL: WHERE tên_phòng = "Recluse and defiance"

Issue: Entity translated inconsistently (VI → EN).

These examples demonstrate how translation noise appears in practice: schema vocabulary drift, EN-VI entity inconsistency, and discrepancies between question phrasing and SQL complexity.

Implication for ViR²

Because many errors involve lexical variation but preserved syntactic structure, ViR²'s POS-pattern matching and diversity-aware re-ranking help reduce the impact of these inconsistencies-reinforcing the motivation for our design.

CRediT author statement

Dinh Minh Hoa: Conceptualisation, Methodology, Investigation, Writing original draft; Tran Khai Thien: Supervision, Methodology Guidance, Writing - Reviewing & Editing.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

[1] T. Yu, R. Zhang, K.C. Yang, et al. (2018), "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and Text-to-SQL task", *Proceedings of The 2018 Conference on Empirical Methods in Natural Language Processing*, pp.3911-3921, DOI: 10.18653/v1/D18-1425.

[2] J. Yi, G. Chen, X. Zhou (2025), "Decoupling SQL query hardness parsing for Text-to-SQL", *Neurocomputing*, **621**, 13pp, DOI: 10.1016/j.neucom.2024.129293.

[3] T.B. Brown, B. Mann, N. Ryder, et al. (2020), "Language models are few-shot learners", *Advances in Neural Information Processing Systems*, pp.1877-1901.

[4] A.T. Nguyen, M.H. Dao, D.Q. Nguyen (2020), "A pilot study of Text-to-SQL semantic parsing for Vietnamese", *Findings of The Association for Computational Linguistics: EMNLP 2020*, pp.4079-4085, DOI: 10.18653/v1/2020.findings-emnlp.364.

[5] M. Pourreza, D. Rafiei (2023), "DIN-SQL: Decomposed in-context learning of Text-to-SQL with self-correction", *Advances in Neural Information Processing Systems*, pp.36339-36348.

[6] Z. Hong, Z. Yuan, Q. Zhang, et al. (2025), "Next-generation database interfaces: A survey of LLM-based Text-to-SQL", *IEEE Transactions on Knowledge and Data Engineering*, **37(12)**, pp.7328-7345, DOI: 10.1109/TKDE.2025.3609486.

[7] D. Zhou, N. Schärli, L. Hou, et al. (2023), "Least-to-most prompting enables complex reasoning in large language models", *International Conference on Learning Representations 2023*, 61pp.

[8] Z. Tan, X. Liu, Q. Shu, et al. (2024), "Enhancing Text-to-SQL capabilities of large language models through tailored promptings", *Proceedings of The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pp.6091-6109.

[9] B. Zhai, C. Xu, Y. He, et al. (2025), "Optimizing reasoning for Text-to-SQL with execution feedback", *Findings of The Association for Computational Linguistics: ACL 2025*, pp.19206-19218, DOI: 10.18653/v1/2025.findings-acl.982.

[10] Y. Gan, X. Chen, J. Xie, et al. (2021), "Natural SQL: Making SQL easier to infer from natural language specifications", *Findings of The Association for Computational Linguistics: EMNLP 2021*, pp.2030-2042, DOI: 10.18653/v1/2021.findings-emnlp.174.

[11] S. An, B. Zhou, Z. Lin, et al. (2023), "Skill-based few-shot selection for in-context learning", *Proceedings of The 2023 Conference on Empirical Methods in Natural Language Processing*, pp.13472-13492, DOI: 10.18653/v1/2023.emnlp-main.831.

[12] J. Kapuriya, M. Kaushik, D. Ganguly, et al. (2025), "Exploring the role of diversity in example selection for in-context learning", *Proceedings of The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.2962-2966, DOI: 10.1145/3726302.3730194.

[13] Z. Shen, P. Vougiouklis, C. Diao, et al. (2024), "Improving retrieval-augmented Text-to-SQL with AST-based ranking and schema pruning", *Proceedings of the 2024 Conference on*

Empirical Methods in Natural Language Processing, pp.7865-7879, DOI: 10.18653/v1/2024.emnlp-main.449.

[14] R. Luo, L. Wang, B. Lin, et al. (2024), "PTD-SQL: Partitioning and targeted drilling with LLMs in Text-to-SQL", *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp.3767-3799, DOI: 10.18653/v1/2024.emnlp-main.221.

[15] D. Gao, H. Wang, Y. Li, et al. (2024), "Text-to-SQL empowered by large language models: A benchmark evaluation", *Proceedings of The VLDB Endowment*, **17(5)**, pp.1132-1145, DOI: 10.14778/3641204.3641221.

[16] C. Mai, R. Tal, T. Mohamed (2024), "Learning metadata-agnostic representations for Text-to-SQL in-context example selection", *NeurIPS 2024 Third Table Representation Learning Workshop*, 23pp.

[17] D.H. Dao, N.T.K. Huynh, K.Q. Tran, et al. (2025), "Open-ViTabQA: A novel benchmark for Vietnamese question answering on open domain wikipedia table", *Knowledge-Based Systems*, **330(A)**, DOI: 10.1016/j.knosys.2025.114391.

[18] D.H. Nguyen, N.K. Le, L.M. Nguyen (2023), "ViWiQA: Efficient end-to-end Vietnamese Wikipedia-based open-domain question-answering systems for single-hop and multi-hop questions", *Information Processing & Management*, **60(6)**, DOI: 10.1016/j.ipm.2023.103514.

[19] N.K. Le, D.H. Nguyen, T. Le, et al. (2023), "VIMQA: A Vietnamese dataset for advanced reasoning and explainable multi-hop question answering", *Proceedings of The Thirteenth Language Resources and Evaluation Conference*, pp.6521-6529, DOI: 10.1016/j.ipm.2023.103514.

[20] X. Wang, Z. Chen, T. Xu, et al. (2024), "In-context former: Lightning-fast compressing context for large language model", *Findings of The Association for Computational Linguistics: EMNLP 2024*, pp.2445-2460, DOI: 10.18653/v1/2024.findings-emnlp.138.

[21] S.S. Ghosal, S. Pal, K. Mukherjee (2025), "PromptRefine: Enhancing few-shot performance on low-resource indic languages with example selection from related example banks", *Proceedings of The 2025 Conference of The Nations of The Americas Chapter of The Association for Computational Linguistics: Human Language Technologies*, **1**, pp.351-365, DOI: 10.18653/v1/2025.naacl-long.17.

[22] M. Wu, Z. Liu, Y. Yan, et al. (2025), "RankCoT: Refining Knowledge for Retrieval-Augmented Generation through Ranking Chain-of-Thoughts", *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, **1**, pp.12857-12874, DOI: 10.18653/v1/2025.acl-long.629.

[23] Y. Zhang, M. Taft, J. Tang, et al. (2024), "Neural correlates of semantic-driven syntactic parsing in sentence comprehension", *NeuroImage*, **289**, 13pp, DOI: 10.1016/j.neuroimage.2024.120543.

[24] D.Q. Nguyen, A.T. Nguyen (2020), "PhoBERT: Pre-trained language models for Vietnamese", *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp.1037-1042, DOI: 10.18653/v1/2020.findings-emnlp.92.