Improved Productivity with AI Models for SQL tasks: A Case Study

Thanh Vu [1], Thirunavukarasu Balasubramaniam [1]
Richi Nayak[1] , Sara Keretna [2]

[1] Queensland University of technology, Australia
[2] Telstra Group, Australia

Abstract. This study investigates the practical deployment of AI-based Text-to-SQL (T2S) models on a real-world telecommunication dataset, aiming to enhance employee productivity. Our experiment addresses the unique challenges in telecommunication datasets not explored in previous works using annotated datasets. Leveraging advanced retrieval augmented generative (RAG) models like Vanna AI and Llamaindex, we benchmark their performance on synthetic datasets such as SPIDER and BIRD with different LLM backbones and subsequently compare the best-performing model to human performance on our proprietary dataset. We propose the Productivity Gain Index (PGI) to quantify the dual aspects of productivity improvement—time efficiency and accuracy—by comparing AI performance with human analysts across various SQL tasks. Results indicate significant productivity gains, with AI-based tools demonstrating superior query processing and accuracy performance. This prominent gap signals the potential of AI-based tool applications in the actual company domain for improved productivity.

Keywords: Text-to-SQL, Large Language Models, Productivity Gain Index, Retrieval-Augmented Generation, Artificial Intelligence Evaluation

1 Introduction

Artificial Intelligence (AI) encompasses various technologies that mimic human cognitive functions, including natural language processing, machine learning, and data analysis [1]. An example of such a technology is Text-to-SQL (T2S) models that focus on translating natural language queries into SQL statements. T2S models have revolutionised database interaction by converting natural language queries into SQL statements, significantly improving query time. Traditional T2S methods can be divided into two main categories: seq2seq methods [2, 3] and non-seq2seq methods [4, 5]. Seq2seq methods, using fine-tuned transformer architectures, achieve high performance by directly translating natural language questions into SQL [3]. In contrast, non-seq2seq methods focused on training text embeddings via relation-aware self-attention mechanisms to handle complex queries and improve contextual understanding.

Despite significant advancements, the maximum accuracy obtained by traditional T2S models is around 80% on the SPIDER leaderboard[1]. Large language models (LLMs) have been incorporated to improve accuracy for their powerful linguistic capabilities. Models such as C3 [6] and DAIL-SQL [7], leveraging zero-shot learning on ChatGPT, achieved 86.6% accuracy on SPIDER, significantly outperforming traditional T2S models. Few-shot learning approaches also demonstrated high performance, with 86.75% on SPIDER and 59.6% on BIRD. BIRD's poor accuracy is due to the complexity of queries [8].

However, these existing models face multiple challenges in industrial applications. **Problem 1:** Firstly, existing T2S approaches do not sufficiently empower non-technical users to explore large datasets [9]. As depicted in Figure 1, stakeholders in a company need a "spot" insight retriever that may not have SQL knowledge. A robust and fast model with a user-friendly interface would allow them to interact with the database more effectively.

[1] https://yale-lily.github.io/spider

David C. Wyld et al. (Eds): DSML, ARIA, NLP, CSEN, BIBC, EDTECH - 2024
pp. 21-33, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142402

Moreover, new analysts require a fast T2S tool to facilitate information retrieval without consulting senior employees. Additionally, the unique nature of telecommunication databases (real-time, dynamic, and domain-specific), as illustrated in Table 1, demands the deployed T2S model to possess strong NLP capabilities to understand the context for improved information retrieval [10]. **Problem 2:** Secondly, existing techniques mostly adopted Exact Match (EM) and Execution Accuracy (EX) to evaluate the model performance. However, employee performance within the integration of advanced T2S tools must be investigated to formulate strategic decisions in AI development in the company workspace. Therefore, an evaluation metric that can quantify the impact of AI on employee performance is needed.

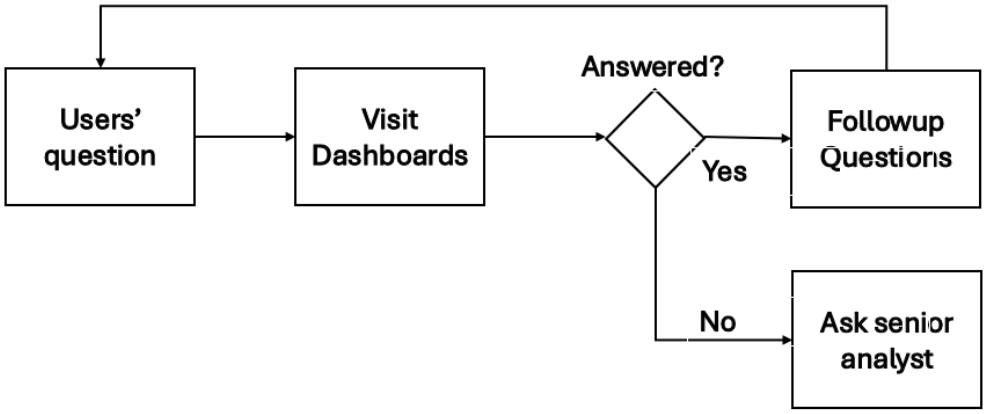


Fig. 1: A simple illustration of the business problem: Stakeholders spend more time extracting follow-up insights from the data; junior analysts must revisit dashboards or consult senior peers for information discovery in the company's data lake. Traditional T2S failed to provide this "spot" information retrieval.

AI technologies have demonstrated significant transformative impacts on various sectors, and the widespread adoption of AI brings notable efficiency values to software organisations and individuals [11]. However, Practical applications of AI tools in real-world T2S problems remain underexplored, with few exceptions [12], who integrated LLMs with retrieval augmented generative (RAG) frameworks for health record answering to answer the problem **(1)**. RAG frameworks enhance LLMs by retrieving relevant document chunks from external knowledge bases, providing contextual understanding of database structure and schema mapping [13]. This approach reduces the rate of false predictions by clarifying ambiguities in data fields and terminologies with enriched information, thus improving SQL query predictions. Due to the high demand for RAG in the T2S field, various RAG tools are available online for reproduction to address the problem **(1)**. Waii [14] collaborated with Microsoft to deal with real-world data issues by directly using LLMs to find relevant data fields. However, the accuracy did not show statistically significant improvement. Table 2 presents the execution accuracy performance of existing models, including the baseline GPT-4 and RAG-based models, on the synthetic BIRD and SPIDER datasets. While these models show promising results for further deployment, concerns about security in current RAG implementations pose significant challenges for business applications. A comprehensive experiment on privacy risk of LLMs and RAG-based models were conducted in [15], confirming a high possibility of data leakage from their pre-training and

Aspect	SPIDER and BIRD	Telecommunication
Complex Structure	Well-structured, diverse schemas with clear relationships.	Complex, heterogeneous, various data types, lack standard schema definitions.
Domain Specific	General-purpose, varied terminology.	Domain-specific terminology.
Annotation Quality	Meticulously annotated, high quality, consistent for training	Noisy, incomplete, less annotated, and challenging for training.
Realtime	Don't include real-time queries.	Requires real-time query handling for operations and analytics.
Dynamic	Not dynamic	Dynamic data reflecting constant changes in usage, configurations, and interactions.

Table 1: Unique characteristics of real-world datasets that cannot be found in benchmarking data such as SPIDER and BIRD. The state-of-the-art AI model performance might not apply to the real-world datasets as used in this case study

fine-tuning data. These models are vulnerable to data leakage, where attackers could potentially extract sensitive information from the retrieval database, posing a serious risk for organizations handling proprietary or personally identifiable information. This vulnerability outweighs the accuracy benefits, making it essential to address security measures before considering their deployment. Conversely, Vanna AI[2] demonstrated a more promising performance by leveraging selected correct data fields with summarised database schema information. Similarly, Llamaindex[3] offers even faster query times using the vectorstore database indexing. These two techniques are feasible for integrating local LLM instances that helps to mitigate the issue of data leakage as discussed previously. To enable non-experts to use these tools effectively for information discovery, our experiment will adopt a zero-shot learning strategy, as few-shot learning requires SQL query examples to train LLMs. To quantify the impact of AI in a company context as in problem **(2)**, surveys from Weider et al. [17] and Palvalin et al. [18] confirmed that staff experience is one of the most influential factors on the productivity in an organisation. Therefore, our research classifies the complexity of query questions to validate staff experience. Babashahi et al. [1] quantified the impact of AI in various sectors, from manufacturing to services. They acknowledged the transformative role of AI on the improved human-with-AI worker ratios, but technical issues on real-world datasets and evaluation metrics were not comprehensively regarded. To address the problems (1) and (2) discussed, this study will first experiment on AI-based tools on datasets for compatibility and then compare the effec-

[2] https://vanna.ai/
[3] https://www.llamaindex.ai/

	Synthetic Datasets	
	BIRD	**SPIDER**
LLM Only		
GPT4	54.8	80.8
RAG Model		
DIN-SQL [16]	55.9	85.3
DAIL-SQL [7]	57.4	86.2

Table 2: Reported execution accuracy performance of different state-of-the-art models on synthetic datasets.

tiveness of AI-based tools with real humans via a pre-defined set of gold SQL queries. In summary, our contributions are twofold:

- A detailed deployment plan of an AI-based T2S model on a telecommunication dataset to investigate the compatibility of AI tools in real-world settings.
- **A new metric** of productivity improvement named Productivity Gain Index (PGI) to compare AI performance with human performance over a series of SQL queries in terms of time and accuracy.

This dual focus on technical performance and practical business outcomes is rare in current research, which tends to prioritise technical benchmarks on synthetic datasets over real-world data.

Our paper is structured as follows: Section 2 introduces RAG models for T2S tasks and our experiment on both synthetic and real-world datasets, including the new PGI metric for AI evaluation. Section 3 analyses their performance and practical utility. Finally, Section 5 concludes our work and proposes future considerations in metric design.

2 Methodology Setting

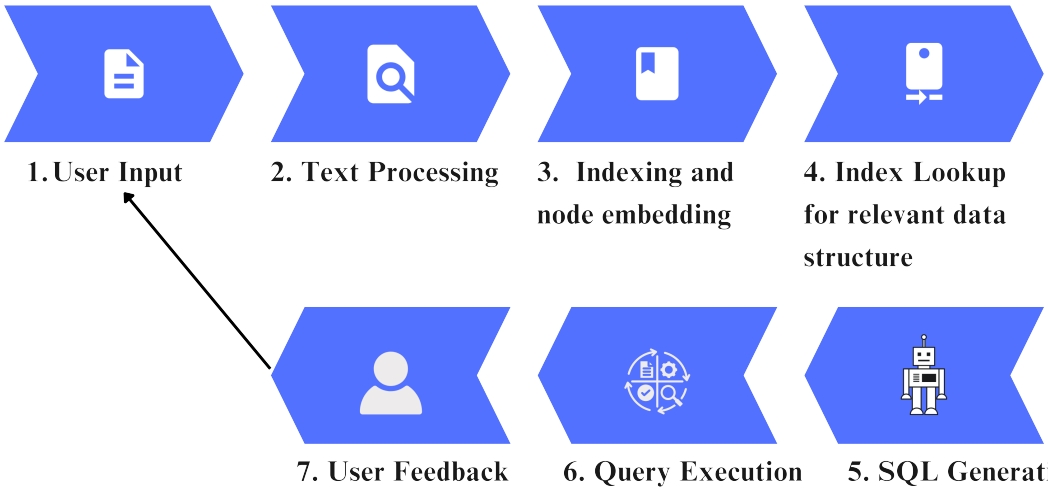
The business problem introduced in Section 1, coupled with the recent advancement of AI tools in T2S, prompted the need for robust frameworks to build LLM-powered applications. To this end, several available RAG-based options such as Vanna AI and Llamaindex can be considered for development. Figure 2 illustrates two AI frameworks to seek a fast approach to explore the datasets via dashboards and SQL queries utilising the power of LLMs. This paper conducts the experiment into three stages that correspond to each of the following research questions:

- RQ1: Which LLM-based T2S is better in accuracy and efficiency?
- RQ2: What is the difference in the performance of each framework on synthetic and real-world datasets?
- RQ3: How zero-shot learning AI tools are better than real employees in terms of accuracy and resource usage?

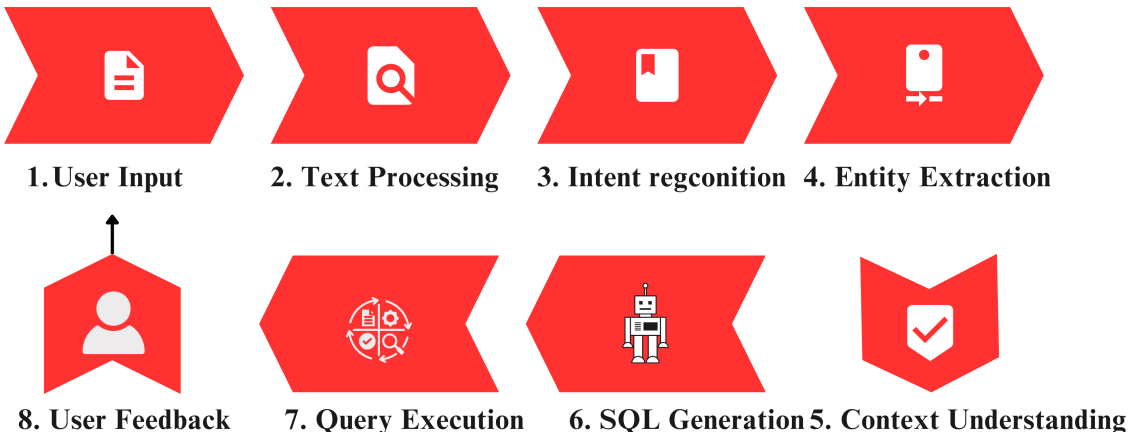
The following subsections will introduce the details of our experiment goals before discussing the evaluation scheme.

2.1 Model Description

Many T2S models show excellent accuracy on synthetic datasets, but most businesses prioritises frameworks that offer scalability and minimal security risks [1]. We evaluate security based on local data preprocessing, controlled access, and encryption. We also filter out frameworks that are difficult to integrate or not economical (pricing plans). Based on these criteria, we selected AI tools such as:



(a) A simplified illustration of Vanna AI Pipeline



(b) A simplified illustration of Llamaindex Pipeline

Fig. 2: Two RAG frameworks between Vanna AI and Llamaindex. While Llamaindex focuses on the retrieval speed with its power of indexing, Vanna AI improves the accuracy with contextual understanding and entity identification

- **Vanna AI**: Vanna AI is an advanced tool designed to simplify the process of generating SQL queries from natural language inputs. As shown in Figure 2a, the process begins with text preprocessing (tokenization, stemming, etc.). Machine learning models then recognise user intent and extract relevant entities like table and column names using named entity recognition. Advanced NLP models ensure contextual understanding and accurate SQL query generation through template-based methods or machine learning techniques. The generated SQL query is executed on the database, and the results are formatted for user-friendly output.
- **Llamaindex T2S**: Llamaindex T2S is a feature or module within the Llamaindex framework that focuses on converting natural language queries into SQL queries. Like Vanna AI, this tool allows users to interact with databases using plain English questions. From Figure 2b, Llamaindex converts documents into individual "nodes" for indexing and search, establishing relationships between nodes to provide semantic context. Moreover, it supports hybrid search by integrating SQL and vector databases and handling complex queries in structured and unstructured data.

We adopted several prominent backbone LLMs for SQL generation in each of these tools, such as GPT3.5 [4] and Ollama [5] on synthetic datasets. However, for the deployment on real-world datasets, we preferred to use a local GPT-like instances named **"GPT3.5-turbo-16k-tpu"** to avoid potential data leakage.

2.2 Data Collection

In this paper, AI tools will be benchmarked on two synthetic datasets to validate their efficiency before the real-world application:

SPIDER [19] is a large-scale dataset with 138 domains designed to test model generalisation capabilities from database schemas. It is a standard benchmark for evaluating cross-domain and multi-table Text-to-SQL (T2S) tasks. The dataset includes a training set with 7,000 samples, a development set with 1,034 samples, and a test set with 2,147 samples. For feasible implementation, we adopted the development set only.

BIRD [20] contains adversarial examples to test robustness, featuring over 12,751 unique question-SQL pairs across 95 big databases and more than 37 professional domains, such as blockchain, hockey, healthcare, and education. Our study used a lite version called BIRD-MiniDev, which contains 500 high-quality text-to-SQL pairs from the original benchmark dataset. This helps to facilitate efficient and cost-effective development cycles during our testing phase.

Besides BIRD and SPIDER, this study uses real-world telecommunications data, covering customer interactions, service usage, network configurations, and billing information. The dataset consists of 25 tables with 1,624 columns and 31,125 rows, lacking foreign and primary keys, which adds complexity to retrieval tasks. Given the proprietary nature of the data, strict anonymisation procedures are followed to ensure the removal of sensitive information, complying with data privacy regulations. For testing purposes, we asked analyst experts to perform 15 gold SQL queries of varying difficulty levels. These queries were designed to extract desired information based on the query types discussed next.

2.3 SQL Query Types

To comprehensively evaluate the performance and robustness of the proposed tools on an unannotated telecommunication dataset, we designed a list of 15 questions categorised into two levels of difficulty defined by our experts - *Easy (one clause or one table) and Difficult (multiple clauses or tables)* - across five distinct types of SQL tasks. These tasks are defined to correspond to our need for information extraction, including: *aggregation and join (AJ), filtering and grouping (FG), recursive queries (RQ), conditional aggregation sub-queries (CQ),* and *trend analysis* (TA). Each category is crafted to assess the models' capabilities in handling various intricate SQL constructs and real-world data scenarios, as demonstrated in Figure 1. Specifically, AJ addresses complex database structures through join commands; RQ navigates dynamic changes in the dataset; FG isolates subsets of data to address domain-specific terminology; CQ identifies detailed information within the data pool; and TA investigates real-time changes in the database.

2.4 Evaluation Metrics

To quantify the impact of AI on employee performance, this paper introduces a new index called **Productivity Gain Index (PGI)**. PGI is designed to capture the dual aspects of productivity improvement, time efficiency and accuracy, using AI for human performance. By evaluating the performance changes of an employee before (B) and after (A) receiving

[4] https://chat.openai.com/
[5] https://ollama.com/

assistance from an AI tool, the index provides a holistic view of cumulative productivity gain on time and accuracy using integrals across given N tasks as follows:

$$\begin{aligned} \text{PGI} = \alpha \underbrace{\left(\int_1^N \frac{\sum_{i=1}^{N} (B_i^{\text{acc}} - A_i^{\text{acc}}) \mu_i}{\sum_{i=1}^{N} A_i^{\text{acc}} \mu_i} \, dN \right)}_{\text{Cumulative Accuracy Gain (CAG)}} \\ + (1 - \alpha) \underbrace{\left(\int_1^N \frac{\sum_{i=1}^{N} (B_i^{\text{ti}} - A_i^{\text{ti}}) \mu_i}{\sum_{i=1}^{N} A_i^{\text{ti}} \mu_i} \, dN \right)}_{\text{Cumulative Time Gain (CTG)}}, \end{aligned} \tag{1}$$

where N is the total number of test questions in our experiment. Some examples of our test questions are shown in Table 5. The full list is available via Github[6]. These questions are annotated by our experts and categorised into two difficulty levels: easy and difficult. α is the priority weight that measures the importance of two components (time and accuracy). We value them equally in this paper, hence $\alpha = 0.5$. $B_i^{\text{acc}} and A_i^{\text{acc}}$ are execution accuracy measured for a task before and after the help of AI, respectively. They are recorded as binary values (1 for True and 0 for False). $B_i^{\text{ti}} and A_i^{\text{ti}}$ measure the duration of time (in seconds) taken to write a query by an employee before and after AI tool, respectively. μ_i is the difficulty weight where $\mu_i^{\text{easy}} = 1$ and $\mu_i^{\text{hard}} = 2$;

Each component of Equation 1 initially quantifies the improvements in accuracy and time efficiency separately. Subsequently, integrals compute the cumulative linear gains over the specified tasks. The final index is derived by aggregating these components, providing a comprehensive measure of the balanced productivity improvements attributable to the AI tools. In addition to PGI, **Execution Accuracy (EX)** and **Valid Efficiency Score (VES)** [21] are also used to test the accuracy and efficiency of AI tools on synthetic datasets, respectively. EX compares the accuracy of the SQL execution output, while VES aims to utilise other components such as computation resources, time taken per query, and accuracy to validate the model's efficiency.

2.5 Experiment Design

Our experimental design is structured in three phases. Each phase is designed to answer specific research questions and involves different datasets and evaluation criteria. The goal is to comprehensively evaluate the performance of various T2S tools and determine the most effective model for our specific use case.

Phase 1: Benchmarking on Synthetic Datasets using Vanna AI and Llamaindex - Discussion of security concern. In the first phase, we compare state-of-the-art RAG-based AI tools on synthetic datasets such as SPIDER and BIRD. The primary objective of this phase is to answer the research question RQ1. Each model is run in AzureML Studio and assessed based on its accuracy in generating correct SQL queries (SQL queries that extract correct results) and its computational cost. The use of synthetic datasets allows us to establish a baseline performance for each model in a controlled environment. Although there have been

[6] https://github.com/thanh31596/

Phase 2: Application on the Real-world telecommunication Dataset. The evaluation criteria for this phase remain consistent with Phase 1, focusing on accuracy and security. Before deployment, the telecommunication dataset must be converted from Excel to PostgreSQL format, as Vanna AI only accepts SQL-like databases. The DDL documentation will only be used during the training phase, and no further information will be given. For a fair evaluation, we randomly selected 15 pairs of queries and questions from the SPIDER and BIRD datasets. We tested the model's performance in real-world settings using a 10-fold cross-validation approach. The model's results are then compared with the analysts' answers based on two criteria: the time to complete the queries and the accuracy of the generated SQL statements using t-test evaluation in Equation 2 [22].

$$t = \frac{\bar{X}}{s_d/\sqrt{N}}, \tag{2}$$

Here, $\bar{X}$ is the mean difference between paired samples from different data sources and s_d is the standard deviation of the differences

To operate this test evaluation, we compute the average time per query in each fold, and accuracy is calculated using the EX metric.

Phase 3: Final Evaluation and Human Comparison. In the final phase, illustrated in Figure 3, we choose the best-performing model (Vanna AI-GPT3.5) based on the first two phases and apply it to our dataset with a set of 15 pre-defined questions. This phase is designed to simulate real-world usage and assess the model's practical utility. To benchmark the model's performance, we collect answers from our analysts—who are asked to solve the same set of SQL queries categorised by difficulty levels. The selected AI tool is provided with a description of the datasets and telecommunication terminologies; this is expected to improve its accuracy on the model and be able to simulate a senior data analyst. PGI is used in this phase to give a holistic view of the impact of the deployment of AI. Phase 3 is conducted using an AWS EC2 [7] instance to ensure compatibility with our organisational dataset location, which is stored in an S3 bucket with enhanced security measures. This evaluation helps us understand the model's efficiency and accuracy in a practical setting and its potential to improve productivity within our organisation.

3 Results

The result section aims to comprehensively evaluate the collected models to answer the three research questions.

3.1 RQ1: Model performance over synthetic datasets

The results of our experiments are divided into two primary categories: accuracy and security. Each category highlights the performance of the evaluated models—Llamaindex with GPT-3.5, Vanna AI, on both synthetic datasets and real-world telecommunication datasets.

Results shown in Table 3, Vanna AI with GPT-3.5 demonstrated the highest execution accuracy on both the SPIDER and BIRD datasets. It can be explained that Vanna AI's retrieval mechanism allows it to dynamically fetch relevant schema and historical queries to obtain strong contextual information for enhanced SQL generation. Llamaindex generated

[7] https://aws.amazon.com/ec2/

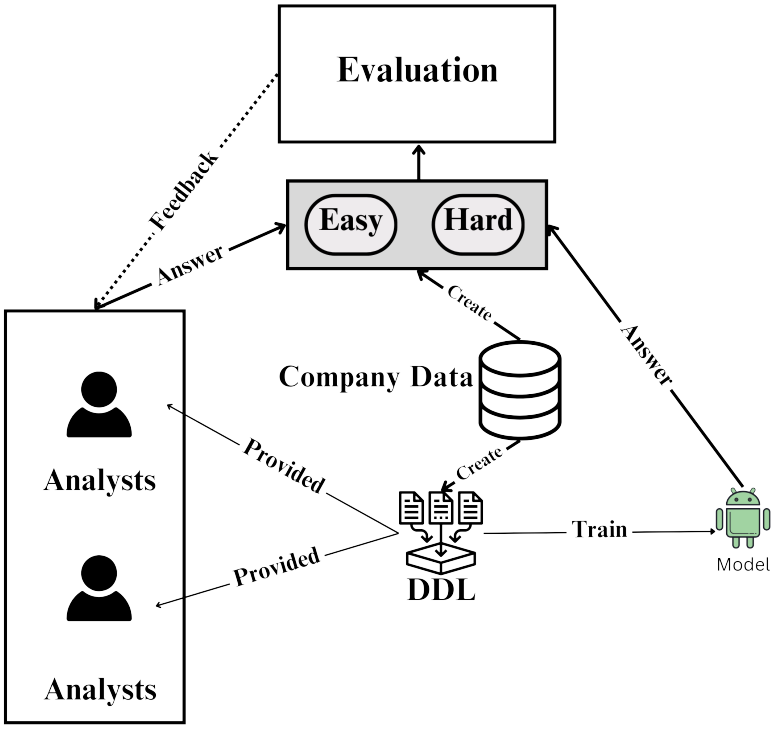


Fig. 3: Testing Scenario in a real-world company setting. A group of analysts is asked to complete the same SQL queries for benchmarking with AI tools with varying difficulty levels.

fewer output tokens for the same LLM backbone and exhibited lower resource usage and query time than Llama Index. This efficiency was consistent across both SPIDER and BIRD datasets, indicating that Llamaindex can deliver decent accuracy while being more resource-efficient. In general, GPT3.5 has faster training with lower running time compared to Ollama, regardless of RAG frameworks. Additionally, despite Llamaindex offering higher customizability, it cannot adopt our local LLM instance to secure data protection. **Based on this first experiment, we decided that Vanna AI - GPT3.5 is the optimal option for the next deployment**.

3.2 RQ2: Differences on synthetic and real-world datasets

Investigating the performance differences of each model on synthetic and real datasets allows us to determine the true capability of unannotated datasets and the robustness of collected models in practical contexts. Figure 4a displays the average accuracy and time complexity of 10-fold validation from 15 pre-defined questions using two backbone LLMs for the Vanna AI framework. Figure 4 shows that both models experienced a notable decline in accuracy when transitioning from synthetic to real-world data. Specifically, Vanna AI -Ollama saw a 12% reduction in accuracy, while it was about 27% in Vanna AI-GPT3.5 from SPIDER-dev to real-world datasets. Similar trends were observed in the BIRD-mini dev set. Notably, neither tool was augmented with external knowledge, such as database summaries or field definitions. Furthermore, the complexity of the real-world dataset also resulted in increased resource demands, as witnessed in the Timelines (shown in orange colour in Figure 4a). The negative t-test values shown in Figure 4b indicate that Vanna AI (GPT3.5) and Vanna AI (Ollama) performed significantly better on the

	BIRD-minidev			SPIDER-DEV		
	EX	VES	Time Avg	EX	VES	Time Avg
Vanna - GPT3.5	**0.64**	0.47	9.2±4.7	**0.76**	0.72	46.7±8.6
Vanna - Ollama	0.51	0.33	13.2±5	0.69	0.51	71.4±11.2
Llamaindex-GPT	0.62	**0.51**	**7.1±2.8**	**0.76**	**0.82**	**42.6±1.8**
Llamaindex-Ollama	0.41	0.42	9.7±3.3	0.53	0.61	60.7±4.3

Table 3: Comparison of Vanna AI and Llamaindex on synthetic datasets

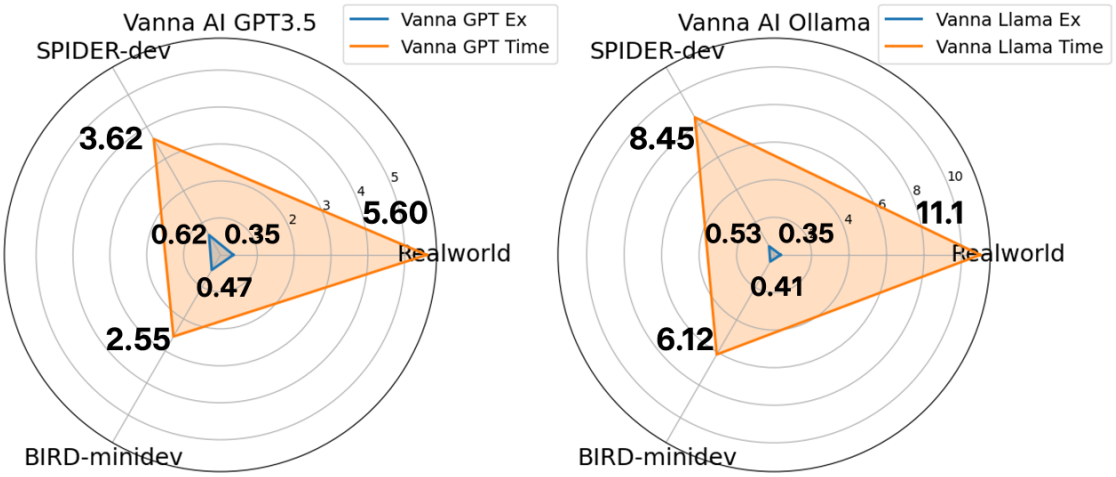
synthetic datasets (SPIDER and BIRD) compared to the real-world telecommunication dataset. This suggests that the models achieve higher accuracy and consistency in controlled, well-annotated environments, highlighting the challenges posed by the complexities and noisiness of real-world data. From this analysis, we confirm that Vanna AI is a better option for deployment to compare with human in the next experiment.

3.3 RQ3: Human Vs AI - Who is better?

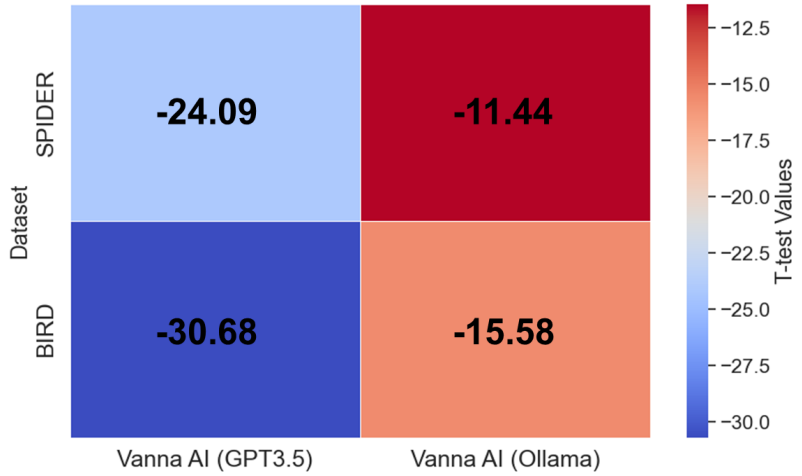
Based on the results shown in Table 4, it is evident that within the enriched information from data description and field definitions, the LLM component in RAG frameworks enhanced the accuracy of T2S model (0.35 without external knowledge and 0.8 with external knowledge). Table 4 confirms that Vanna AI significantly outperforms human analysts regarding time efficiency and accuracy across various SQL tasks. The AI's ability to handle complex queries, such as recursive and conditional queries, is evident as it performs these tasks with greater speed and fewer errors than analysts, who often fail to complete them. Particularly in trend analysis, Vanna AI successfully executes tasks that human analysts struggle with, underscoring its capability to manage intricate data scenarios. Vanna AI achieves 60% accuracy on difficult tasks, while human analysts average only 10% accuracy on these same tasks. On the productivity aspect, $\text{PGI}_{Analyst1}$ is about 10.28, implying that the productivity of accuracy and time retrieval has increased approximately tenfold when AI tools are used. $\text{CAG}_{Analyst1}$ is 7, lower than PGI, revealing that AI tools impact on time more than accuracy. A similar observation in Analyst 2 with $\text{PGI}_{Analyst2}$ of 9.12, higher than its accuracy gain of 5.5. Indeed, the integration of AI not only completed a variety of tasks more accurately but also in much less time, highlighting the effectiveness of AI in enhancing work efficiency and performance.

Query Types	**AJ**				**FG**					**RQ**		**CQ**		**TA**		**PGI**
Levels	easy	easy	easy	difficult	easy	easy	easy	easy	easy	difficult	easy	easy	difficult	difficult	difficult	
Vanna AI	11	4	2	4	2	2	6	X	2	X	14	8	X	14	12	
Analyst 1	60	60	120	X	180	180	300	240	300	X	480	1200	X	X	X	**10.28**
Analyst 2	55	156	91	297	130	135	114	150	297	X	X	309	X	X	X	**9.12**

Table 4: Performance comparison of Vanna AI and Analysts across different types of SQL tasks in time values (seconds), X represents wrong results.



(a) A comparison of Vanna AI GPT3.5 and Vanna AI Ollama on real-world dataset and synthetic datasets



(b) t-test values of synthetic datasets to real-world datasets with the degree of freedom to be 28

Fig. 4: Performance of Vanna AI with different backbone LLMs on synthetic and real-world datasets. A slight decline in accuracy on real-world datasets indicates the higher complexity of unannotated databases

4 Discussion

The result from this paper holds significant promise for non-technical users who often struggle with complex database queries. By allowing natural language interaction with databases, these models empower users without SQL expertise to extract valuable insights directly from large datasets. This ease of access means that stakeholders, such as managers, marketing professionals, or customer service representatives, can independently retrieve data-driven insights, speeding up decision-making processes and reducing reliance on technical staff. Furthermore, by lowering the technical barrier, organisations can democratise data access, enabling a wider range of employees to contribute to data analysis and fostering a more data-informed culture across the company. However, existing RAG models for text-to-SQL downstream tasks are limited in business application due to their low effort in improving security concerns. To the best of knowledge, only Vanna AI adopted data encryption in a secure on-premises databases, allowing the higher protection

against unauthorised breaches. Additionally, it employs the differential privacy technique to minimise the potential for individual information to be exposed. Therefore, it is suitable for deployment in a business setting.

The experiment of this paper also witness a significant difference in application on synthetic datasets and real-world datasets. Unlike synthetic datasets, which are often well-structured and annotated, real-world datasets are complex, heterogeneous, and frequently contain noise, missing values, and inconsistencies. This lack of standardisation poses challenges for AI models that were trained in controlled environments. Additionally, real-world datasets are often dynamic and evolve over time, requiring models to adapt to changing schemas and data structures, unlike the static nature of synthetic data. The presence of domain-specific terminology and relationships further complicates the model's ability to generate accurate SQL queries without extensive fine-tuning. These limitations highlight the difficulty of generalising RAG-based models for text-to-SQL tasks in different domains due to the unpredictable nature of real-world applications.

5 Conclusion

The transition from synthetic to real-world datasets is a pivotal evaluation of this study, highlighting applicability in practical settings beyond controlled environments, particularly in Telecommunication domain. To quantify the effectiveness of AI models in business contexts, we proposed PGI as a metric of real-world productivity improvement. This metric is particularly valuable because it goes beyond traditional performance measures, which typically focus on either time or accuracy but not both in concert. By capturing the holistic improvements AI tools can offer, PGI enables organisations to make more informed decisions about integrating such technologies into their workflows. Our result analysis indicates that AI can act as a senior analyst to provide fast and accurate insight from a large-scale and complex database for junior peers from a high cumulative gain.

In our future works, we will incorporate the financial aspects, such as the salary of employees, subscription pricing of AI tools, etc., into the proposed productivity gain index to speculate the corporate savings that companies might perceive within the usage of AI.

Type	Question
AG	What are the related X count, average Y value, and Z value count for the top 50 critical A?
FG	Find the X and Y from the Z table where the status is 'active' and A is 'investment'
RQ	What is the hierarchical breakdown of X and the count of Y associated with each Z at each level?
CQ	What are the details of X, including their status, Y, Z, along with the count of active and retired A, ordered by the count of active A and then by the count of retired A?
TA	What are the monthly trends in the active and inactive status of X for each Y?

Table 5: Example of test questions in our experiment for SQL retrieval.

References

1. L. Babashahi, C. E. Barbosa, Y. Lima, A. Lyra, H. Salazar, M. Argôlo, M. A. d. Almeida, and J. M. d. Souza, "Ai in the workplace: A systematic review of skill transformation in the industry," *Administrative Sciences*, vol. 14, no. 6, p. 127, 2024.
2. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

3. J. Li, B. Hui, R. Cheng, B. Qin, C. Ma, N. Huo, F. Huang, W. Du, L. Si, and Y. Li, "Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 076–13 084.
4. R. Cao, L. Chen, J. Li, H. Zhang, H. Xu, W. Zhang, and K. Yu, "A heterogeneous graph to abstract syntax tree framework for text-to-sql," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
5. D. Ma, X. Chen, R. Cao, Z. Chen, L. Chen, and K. Yu, "Relation-aware graph transformer for sql-to-text generation," *Applied Sciences*, vol. 12, no. 1, p. 369, 2021.
6. X. Dong, C. Zhang, Y. Ge, Y. Mao, Y. Gao, J. Lin, D. Lou *et al.*, "C3: Zero-shot text-to-sql with chatgpt," *arXiv preprint arXiv:2307.07306*, 2023.
7. D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou, "Text-to-sql empowered by large language models: A benchmark evaluation," *Proceedings of the VLDB Endowment*, vol. 17, no. 5, pp. 1132–1145, 2024.
8. B. Wang, C. Ren, J. Yang, X. Liang, J. Bai, L. Chai, Z. Yan, Q.-W. Zhang, D. Yin, X. Sun *et al.*, "Mac-sql: A multi-agent collaborative framework for text-to-sql," *arXiv preprint arXiv:2312.11242*, 2024.
9. F. Brad, R. Iacob, I. Hosu, S. Ruseti, and T. Rebedea, "A syntax-guided neural model for natural language interfaces to databases," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 229–233.
10. A. J. Ko, R. DeLine, and G. Venolia, "Information needs in collocated software development teams," in *29th International Conference on Software Engineering (ICSE'07)*. IEEE, 2007, pp. 344–353.
11. R. Kumar, V. Naveen, P. K. Illa, S. Pachar, and P. Patil, "The current state of software engineering employing methods derived from artificial intelligence and outstanding challenges," in *2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP)*. IEEE, 2023, pp. 105–108.
12. A. Ziletti and L. D'Ambrosi, "Retrieval augmented text-to-sql generation for epidemiological question answering using electronic health records," *arXiv preprint arXiv:2403.09226*, 2024.
13. Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.
14. A. Floratou, F. Psallidas, F. Zhao, S. Deep, G. Hagleither, W. Tan, J. Cahoon, R. Alotaibi, J. Henkel, A. Singla, A. Van Grootel, B. Chow, K. Deng, K. Lin, M. Campos, V. Emani, V. Pandit, V. Shnayder, W. Wang, and C. Curino, "Nl2sql is a solved problem... not!" in *CIDR*, 2024.
15. S. Zeng, J. Zhang, P. He, Y. Xing, Y. Liu, H. Xu, J. Ren, S. Wang, D. Yin, Y. Chang *et al.*, "The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag)," *arXiv preprint arXiv:2402.16893*, 2024.
16. M. Pourreza and D. Rafiei, "Din-sql: Decomposed in-context learning of text-to-sql with self-correction," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
17. D. Y. Weider, D. P. Smith, and S. T. Huang, "Software productivity measurements," *AT&T Technical Journal*, vol. 69, no. 3, pp. 110–120, 1990.
18. M. Palvalin, A. Lönnqvist, and M. Vuolle, "Analysing the impacts of ict on knowledge work productivity," *Journal of Knowledge Management*, vol. 17, no. 4, pp. 545–557, 2013.
19. T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
20. Y. Zhang, X. Wang, C. Zhu, Y. Zhu, and J. Cheng, "Bird: Benchmarks for implicit and reversed semantic parsing in text-to-sql," in *Proceedings of the International Conference on Data Engineering (ICDE)*, 2019.
21. J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, B. Wang, B. Qin, R. Geng, N. Huo *et al.*, "Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
22. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2012.