

ระบบแปลงภาษาธรรมชาติเป็น SQL โดยใช้ ChatGPT

A Text-to-SQL System Based on ChatGPT

ปิยะพงษ์ แดงขำ¹ และวดีนาถ วรรณสวัสดิ์กุล^{1*}

Piyapong Dangkhom¹ and Wadeenat Wannasawaskul^{1*}

¹คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเทพสตรี

¹Faculty of Industrial Technology, Thepsatri Rajabhat University

*Corresponding author: wadeenat.w@lawasri.tru.ac.th

Received: December 16, 2025

Revised: February 5, 2026

Accepted: February 12, 2026

บทคัดย่อ

งานวิจัยนี้นำเสนอการพัฒนาและประเมินระบบต้นแบบที่สามารถแปลงข้อความภาษาธรรมชาติภาษาไทยให้เป็นคำสั่ง SQL ได้อย่างแม่นยำ โดยอาศัยความสามารถของโมเดลภาษาขนาดใหญ่ (Large Language Model: LLM) ที่ออกแบบในรูปแบบ My GPTs บนแพลตฟอร์ม ChatGPT ซึ่งเชื่อมต่อกับระบบฐานข้อมูล MySQL จริงผ่าน Web API ผู้ใช้งานสามารถสอบถามข้อมูลด้วยภาษาธรรมชาติผ่านอินเทอร์เฟซแบบโต้ตอบ และระบบจะทำหน้าที่แปลงข้อความเป็นคำสั่ง SQL ส่งไปยังฐานข้อมูล และแสดงผลลัพธ์กลับมาในรูปแบบตาราง ผลการประเมินดำเนินการผ่านชุดคำถามจำนวน 30 ข้อ ผลการทดลองพบว่า ระบบมีความถูกต้องของคำสั่ง SQL เฉลี่ยร้อยละ 90.00 ความสมบูรณ์ของการทำงานร้อยละ 100.00 และเวลาเฉลี่ยในการตอบกลับ 0.0045 วินาทีต่อคำถาม สะท้อนถึงศักยภาพของระบบในการนำไปประยุกต์ใช้ในงานจริง โดยเฉพาะในบริบทที่ต้องการเข้าถึงข้อมูลจากฐานข้อมูลด้วยวิธีที่สะดวก รวดเร็ว และไม่ต้องอาศัยความรู้ด้านเทคนิค

คำสำคัญ: ภาษาธรรมชาติเป็น SQL โมเดลภาษาใหญ่ ฐานข้อมูล ChatGPT

Abstract

This research presents the development and evaluation of a prototype system capable of accurately translating Thai natural language queries into SQL statements. The system leverages a Large Language Model (LLM) implemented as a customized GPT on the ChatGPT platform and connected to a MySQL database through a Web API. Users interact with the system through a conversational interface by submitting queries in natural language. The system then translates these queries into executable SQL statements, executes them against the database, and returns the results in tabular form. The system was evaluated using a dataset of 30 Thai natural language queries. The experimental results demonstrated that the proposed system achieved an average SQL generation accuracy of 90.00%, a query execution success rate of 100.00%, and an average response time of 0.0045 seconds per query. These findings indicate the potential of the system for practical deployment, particularly in scenarios where users need efficient access to structured data without requiring specialized knowledge of database query languages.

Keywords: Text-to-SQL, Large Language Model, database, ChatGPT



บทนำ

การแปลงข้อความภาษาธรรมชาติให้เป็นคำสั่ง SQL: Text-to-SQL เป็นเทคนิคที่ช่วยลดช่องว่างระหว่างผู้ใช้งานทั่วไปกับระบบฐานข้อมูลเชิงโครงสร้าง โดยเฉพาะในยุคที่เทคโนโลยีปัญญาประดิษฐ์มีความก้าวหน้าอย่างรวดเร็ว การนำโมเดลภาษาใหญ่ (Large Language Models: LLMs) เช่น ChatGPT Bard หรือ Copilot มาใช้ในการสร้างคำสั่ง SQL ที่ถูกต้องและเข้าใจได้ง่าย ได้กลายเป็นเครื่องมือสำคัญที่ช่วยให้การเข้าถึงข้อมูลมีประสิทธิภาพมากขึ้น (Pornphol & Chittayasothorn, 2023) นอกจากนี้ยังม้งานวิจัยที่นำเสนอ ChatGPT มาประยุกต์ในการเรียนการสอนรายวิชาฐานข้อมูล โดยพบว่า ช่วยสนับสนุนและเพิ่มประสิทธิภาพอย่างมีนัยสำคัญ (Carr et al., 2023) แนวโน้มหนึ่งที่ได้รับคามนิยม คือ การใช้ prompt engineering ร่วมกับ LLM เพื่อช่วยในการพัฒนา source code และ SQL queries อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งมีประโยชน์ทั้งในด้านการเรียนรู้และการพัฒนาโปรแกรม (Srisuruk & Chittayasothorn, 2023) สอดคล้องกับงานวิจัยที่นำเสนอ ChatGPT ทำหน้าที่เป็นผู้ช่วยสอน ซึ่งช่วยลดภาระของผู้สอน และเปิดโอกาสให้ผู้เรียนได้รับข้อเสนอ

อย่างทันท่วงที (Cagliero et al., 2024) จากการเปรียบเทียบโมเดล LLM หลายตัวในงานที่เกี่ยวข้อง พบว่า โมเดลเหล่านี้มีศักยภาพสูงในการสร้างคำสั่ง SQL ที่ซับซ้อนและสอดคล้องกับโครงสร้างฐานข้อมูล โดยเฉพาะ ChatGPT และ Copilot ที่สามารถสร้างคำตอบพร้อมคำอธิบายเชิงตรรกะได้อย่างชัดเจน (Pornphol & Chittayasothorn, 2023) ซึ่งคล้ายกันกับเป้าหมายในการเพิ่มความสามารถในการเข้าถึงฐานข้อมูลสำหรับผู้ใช้งานที่ไม่ใช่สายเทคนิค (Nilsiam et al., 2025) ในขณะที่งานทบทวนวรรณกรรมโดย Ning et al. (2022) ได้เสนอภาพรวมของวิวัฒนาการในเทคโนโลยี Text-to-SQL ตั้งแต่ระบบแบบ rule-based และ template-based ไปจนถึงระบบที่ใช้ deep learning และ pre-trained models พร้อมชี้ให้เห็นถึงข้อจำกัดของชุดข้อมูลฝึกที่ยังไม่สามารถครอบคลุมความหลากหลายของภาษาและโดเมนได้อย่างทั่วถึง

ในด้านของเครื่องมือเพื่อรองรับการใช้งานจริง งานวิจัยของ Deshpande et al. (2022) ได้นำเสนอแพลตฟอร์ม Querylizer ซึ่งเป็นระบบเชิงโต้ตอบที่ผสานความสามารถของ NLP เข้ากับกระบวนการออกแบบฐานข้อมูล ช่วยให้ผู้ใช้สามารถแปลงคำถามจากภาษาธรรมชาติเป็น SQL ได้

โดยไม่ต้องมีความรู้ด้านเทคนิคอีกทั้งยังมีการแสดงผล schema แบบ visual ที่ช่วยให้เข้าใจโครงสร้างข้อมูลได้ชัดเจนยิ่งขึ้น การปรับปรุงความแม่นยำของการเชื่อมโยง schema ก็เป็นอีกหัวข้อที่ได้รับความสนใจ โดย Lan and Li (2024) ได้เสนอโมเดล PS-SQL ที่ใช้ schema-linking แบบ phrase-based ร่วมกับ pre-trained language models เพื่อจับคู่คำถามกับ schema ได้อย่างแม่นยำและมีประสิทธิภาพมากขึ้น โดยเฉพาะในฐานข้อมูลที่มีหลายตารางและความซับซ้อนเชิงโครงสร้าง ในขณะเดียวกัน Shen et al. (2023) ได้เสนอแนวทางเพิ่มความทนทานให้กับโมเดล Text-to-SQL ด้วยเทคนิค Sequential Feature Augmentation: SFAM ซึ่งออกแบบให้โมเดลสามารถจัดการกับคำถามที่มีรูปแบบหรือลำดับคำที่เปลี่ยนแปลงไปได้ โดยไม่ลดทอนความแม่นยำของผลลัพธ์ นอกจากนี้ ยังมีงานที่ศึกษาศักยภาพของ LLM ในฐานะผู้ช่วยการเขียน SQL โดยเฉพาะในโจทย์ที่มีความซับซ้อนหลายระดับ พบว่าโมเดลอย่าง ChatGPT และ Bard สามารถสร้างคำสั่ง SQL ได้อย่างแม่นยำ และรองรับการใช้งานในเชิงการเรียนรู้และสื่อสารกับฐานข้อมูลได้เป็นอย่างดี (Pornphol & Chittayasothorn, 2024) ส่วนมิติของการขยายข้อมูลฝึกได้ปรากฏในงานวิจัยของ Liu et al. (2025) ได้เสนอ REFORMER ซึ่งเป็นเฟรมเวิร์กที่ใช้ ChatGPT สังเคราะห์ชุดข้อมูลใหม่ผ่านกระบวนการ retrieve-and-edit และตรวจสอบความสอดคล้องของคำถามกับ SQL ด้วย cycle consistency แนวทางนี้ไม่เพียงช่วยให้โมเดลมีความสามารถในการ generalize ไปยังโดเมนใหม่ ๆ แต่ยังช่วยลดต้นทุนในการเตรียมชุดข้อมูลฝึกได้อย่างมีนัยสำคัญ

อย่างไรก็ตาม งานวิจัยที่ผ่านมาแม้จะแสดงให้เห็นถึงศักยภาพของ LLM ในการแปลงข้อความธรรมชาติเป็นคำสั่ง SQL ได้อย่างแม่นยำ แต่ยังคงมีข้อจำกัดในด้านการนำไปใช้งานจริง โดยเฉพาะการที่ผู้ใช้จำเป็นต้องเรียกใช้งานโมเดลและรันคำสั่ง SQL บนเครื่องเซิร์ฟเวอร์ภายนอกด้วยตนเอง ซึ่งเป็นอุปสรรคต่อการเข้าถึงและใช้งานสำหรับผู้ทั่วไป งานวิจัยนี้จึงเสนอแนวทางใหม่โดยใช้ ChatGPT ร่วมกับความสามารถของ CustomGPT เพื่อเชื่อมต่อกับฐานข้อมูลโดยตรง ทำให้สามารถสร้างและรันคำสั่ง SQL

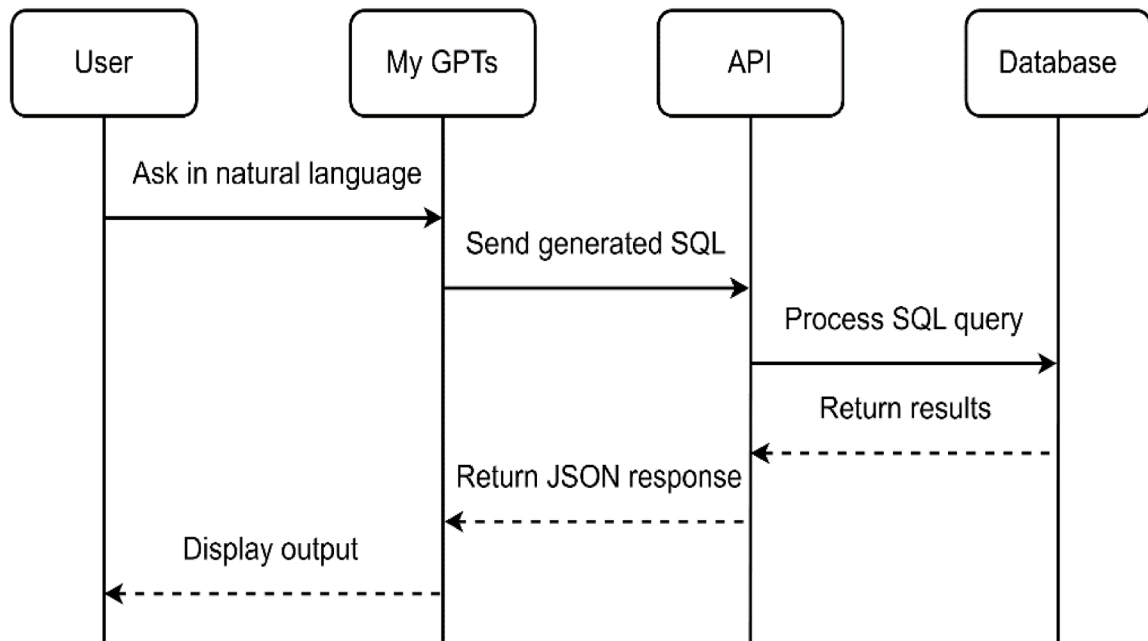
พร้อมแสดงผลลัพธ์ได้ภายในอินเทอร์เฟซเดียวกันแบบโต้ตอบได้ (interactive) ซึ่งช่วยลดความซับซ้อนของกระบวนการ ทำให้ระบบสามารถนำไปใช้งานจริงได้ง่ายและเข้าถึงผู้ใช้ปลายทางได้อย่างแท้จริง

วัตถุประสงค์การวิจัย

พัฒนาระบบที่ผู้ใช้สามารถส่งคำถามในรูปแบบภาษาธรรมชาติและได้รับคำตอบกลับจากฐานข้อมูลได้อย่างเรียลไทม์ และสร้างระบบที่ช่วยลดความซับซ้อนของกระบวนการสืบค้นข้อมูลในฐานข้อมูล ทำให้ระบบสามารถนำไปใช้งานจริงได้ง่ายและเข้าถึงผู้ใช้ปลายทางได้อย่างแท้จริง โดยไม่ต้องอาศัยความรู้ด้านเทคนิค

วิธีดำเนินการวิจัย

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบต้นแบบที่สามารถแปลงข้อความภาษาธรรมชาติ (Natural Language) ให้เป็นคำสั่ง SQL และดำเนินการประมวลผลกับฐานข้อมูลจริงผ่านอินเทอร์เฟซแบบโต้ตอบ (interactive interface) โดยใช้ความสามารถของโมเดลภาษาขนาดใหญ่ (Large Language Model: LLM) ที่พัฒนาในรูปแบบ Custom GPT ร่วมกับ Web API ที่เชื่อมต่อฐานข้อมูล MySQL ซึ่งช่วยให้ผู้ใช้สามารถส่งคำถามในรูปแบบภาษาธรรมชาติและได้รับคำตอบกลับจากฐานข้อมูลได้อย่างเรียลไทม์ ภาพรวมของการทำงานแสดงไว้ในภาพ 1 ซึ่งเป็นแผนภาพลำดับ (sequence diagram) ประกอบด้วย 4 องค์ประกอบหลัก ได้แก่ (1) ผู้ใช้งาน (2) โมเดลปัญญาประดิษฐ์แบบกำหนดเอง (My GPTs) (3) เว็บ API และ (4) ระบบฐานข้อมูล โดยเริ่มจากผู้ใช้ส่งคำถามเข้าสู่ระบบ My GPTs ที่ทำหน้าที่แปลงคำถามเป็นคำสั่ง SQL จากนั้น API จะรับคำสั่งดังกล่าวเพื่อส่งไปยังฐานข้อมูล และเมื่อฐานข้อมูลประมวลผลเสร็จ ระบบจะส่งผลลัพธ์กลับในรูปแบบ JSON ให้ GPT แสดงผลตอบกลับต่อผู้ใช้ภายในอินเทอร์เฟซเดียวกันอย่างครบวงจร



ภาพ 1 ขั้นตอนการทำงานของระบบ

1. โมเดลปัญญาประดิษฐ์แบบกำหนดเอง

ผู้วิจัยได้ประยุกต์ใช้แพลตฟอร์ม Custom GPT ของ ChatGPT (OpenAI, 2022) ภายใต้ชื่อ My GPTs (OpenAI, 2023) โดยมีวัตถุประสงค์เพื่อรองรับการรับคำถามในรูปแบบภาษาธรรมชาติ และสามารถแปลงข้อความดังกล่าวให้เป็นคำสั่ง SQL ที่สอดคล้องกับโครงสร้างฐานข้อมูลที่ใช้งานได้อย่างถูกต้องและมีประสิทธิภาพ กระบวนการสร้าง My GPTs ประกอบด้วยขั้นตอนสำคัญ ได้แก่ การกำหนด schema ของระบบ API ผ่านมาตรฐาน OpenAPI Specification (เวอร์ชัน 3.1.0) ซึ่งใช้กำหนดจุดเชื่อมต่อระหว่างโมเดล LLM และระบบฐานข้อมูลที่อยู่ภายนอก งานวิจัยนี้ประยุกต์ใช้เทคนิคการแปลงข้อความภาษาธรรมชาติเป็นคำสั่ง SQL: Text-to-SQL โดยใช้โมเดลภาษาขนาดใหญ่ในการสร้างคำสั่ง SQL จากคำถามภาษาไทย ระบบสามารถรันคำสั่ง SQL ที่ได้กับฐานข้อมูล MySSQL จริง ผ่าน Web API และแสดงผลต่อผู้ใช้งาน โดยในงานวิจัยนี้ API ดังกล่าวมีเส้น

ทาง (endpoint) เป็น /sqlv1.php และถูกโฮสต์ไว้บนเซิร์ฟเวอร์ที่ <https://api.theepsatri.com> โดยรับคำสั่ง SQL ผ่าน method แบบ POST และส่งค่ากลับในรูปแบบ JSON ตัวอย่างโครงสร้าง endpoint แสดงได้ดังภาพ 2

เพื่อให้โมเดลสามารถเข้าใจโครงสร้างของฐานข้อมูลและสร้างคำสั่ง SQL ได้อย่างถูกต้อง ผู้วิจัยได้จัดเตรียมข้อมูล DDL: Data Definition Language ของฐานข้อมูล ClassicModels ซึ่งประกอบด้วยคำสั่ง CREATE TABLE ของตารางทั้งหมด รวมถึงคีย์หลัก (primary keys) คีย์นอก (foreign keys) และประเภทข้อมูลของแต่ละฟิลด์ แล้วฝังเข้าไปในส่วน Instruction ของ GPT ซึ่งทำหน้าที่เป็นระบบความรู้พื้นฐาน (system context) สำหรับโมเดลการฝัง DDL นี้ช่วยให้ GPT เข้าใจ schema จริงของฐานข้อมูลและสามารถทำการ mapping คำถามกับตารางหรือฟิลด์ที่เกี่ยวข้องได้โดยตรงโดยไม่ต้องฝึกโมเดลใหม่

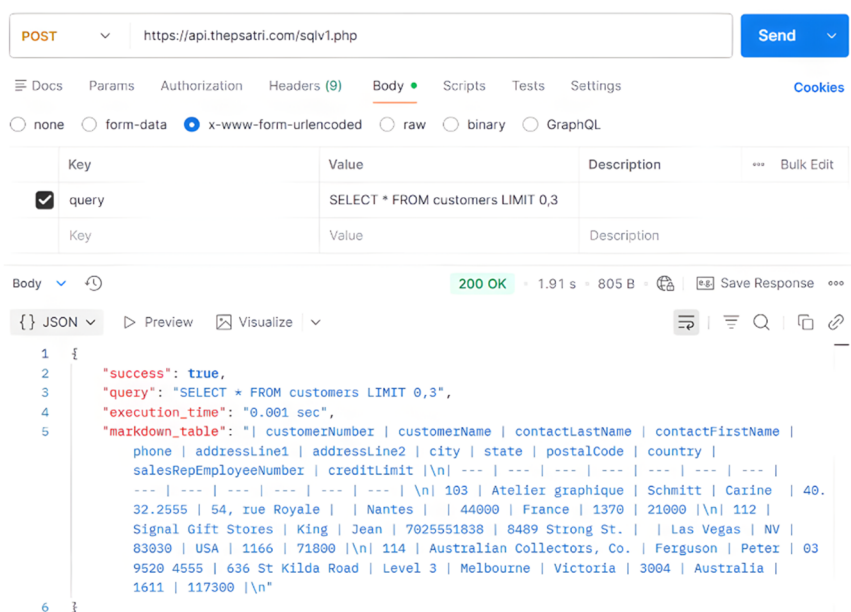
```
{
  "openapi": "3.1.0",
  "info": {
    "title": "MySQL Query API",
    "version": "1.0.0",
    "description": "API สำหรับรัน SQL Query บน MySQL ที่ thepsatri.com"
  },
  "servers": [
    {
      "url": "https://api.thepsatri.com"
    }
  ],
  "paths": {
    "/sqlv1.php": {
      "post": {
        "summary": "Execute SQL Query",
        "operationId": "executeQuery",
        "requestBody": {
          "required": true,
          "content": {
            "application/x-www-form-urlencoded": {
              "schema": {
                "type": "object",
                "properties": {
                  "query": {
                    "type": "string",
                    "description": "SQL Query ที่ต้องการรัน (เฉพาะ SELECT เท่านั้น)"
                  }
                }
              },
              "required": [
                "query"
              ]
            }
          }
        }
      }
    }
  }
}
```

ภาพ 2 ตัวอย่างโครงสร้าง endpoint

2. เว็บ API

เพื่อลดความซับซ้อนในการเข้าถึงและดำเนินการรันคำสั่ง SQL ผู้วิจัยได้พัฒนาเว็บ API โดยใช้ภาษา PHP เพื่อเชื่อมโยงกับระบบฐานข้อมูล MySQL ซึ่งทำหน้าที่เป็นกลไกกลางสำหรับรับคำสั่ง SQL ที่แปลงมาจากข้อความภาษาธรรมชาติ และทำการประมวลผลกับฐานข้อมูลจริง พร้อมจัดส่งผลลัพธ์กลับในรูปแบบ JSON อย่างเป็นระบบ โครงสร้างของ API รองรับคำขอแบบ POST ซึ่งกำหนดค่าพารามิเตอร์ query เป็นข้อมูลหลักที่ต้องรับเข้าจากฝั่งผู้เรียกใช้งาน โมเดล LLM จะทำหน้าที่แปลงคำถาม

จากผู้ใช้ให้เป็น SQL string แล้วส่งผ่านพารามิเตอร์นี้ไปยัง API ที่กำหนดไว้ที่ sqlv1.php ผลลัพธ์ที่ได้กลับมามีอยู่ในรูปแบบ JSON ที่ประกอบด้วย (1) success แสดงสถานะความสำเร็จของการรันคำสั่ง (2) query แสดงคำสั่ง SQL ที่ถูกสร้างจริง (3) execution_time แสดงระยะเวลาที่ใช้ในการประมวลผลคำสั่ง และ (4) markdown_table แสดงผลลัพธ์ในรูปแบบตารางที่สามารถนำไปแสดงผลบนหน้าต่างตอบกลับของ GPT ได้อย่างชัดเจน ผลการเรียกใช้ API แสดงดังภาพ 3



ภาพ 3 การตอบกลับของ API ในรูปแบบ JSON

3. ฐานข้อมูล

ผู้วิจัยได้เลือกใช้ฐานข้อมูลเชิงสัมพันธ์ (relational database) ซึ่งเป็นฐานข้อมูลตัวอย่างจากเว็บไซต์ MySQLTutorial.org (MySQLTutorial.org, 2025) ที่ใช้ในงานสอนและฝึกฝน MySQL ฐานข้อมูล ClassicModels นี้จำลองสถานการณ์ของธุรกิจค้าปลีกสินค้าจำลองรถยนต์ โดยมีข้อมูลจำลองที่สอดคล้องกับระบบงานธุรกิจจริง ประกอบด้วยข้อมูลลูกค้า (customers) สินค้า (products) กลุ่มสินค้า (productlines) การสั่งซื้อ (orders) รายการสินค้าในแต่ละคำสั่งซื้อ (order line items) การชำระเงิน (payments) พนักงาน (employees) และสำนักงานสาขา (offices) ซึ่งมีการเชื่อมโยงกันด้วยคีย์หลัก (primary Key) และคีย์นอก (foreign Key) แผนภาพความสัมพันธ์ระหว่างตาราง (Entity-Relationship Diagram: ERD) แสดงได้ดังภาพ 4

ฐานข้อมูลมีความสมบูรณ์ในด้านโครงสร้างฐานข้อมูลเชิงสัมพันธ์และมีความซับซ้อนระดับปานกลางเหมาะสำหรับการประเมินประสิทธิภาพของระบบที่เกี่ยวข้องกับการแปลงข้อความธรรมชาติเป็นคำสั่ง SQL เนื่องจากสามารถรองรับการสอบถามข้อมูลทั้งแบบพื้นฐานและ

แบบที่มีหลายตารางเชื่อมโยงกันผ่านความสัมพันธ์แบบ Foreign Key

4. การประเมินประสิทธิภาพของระบบ

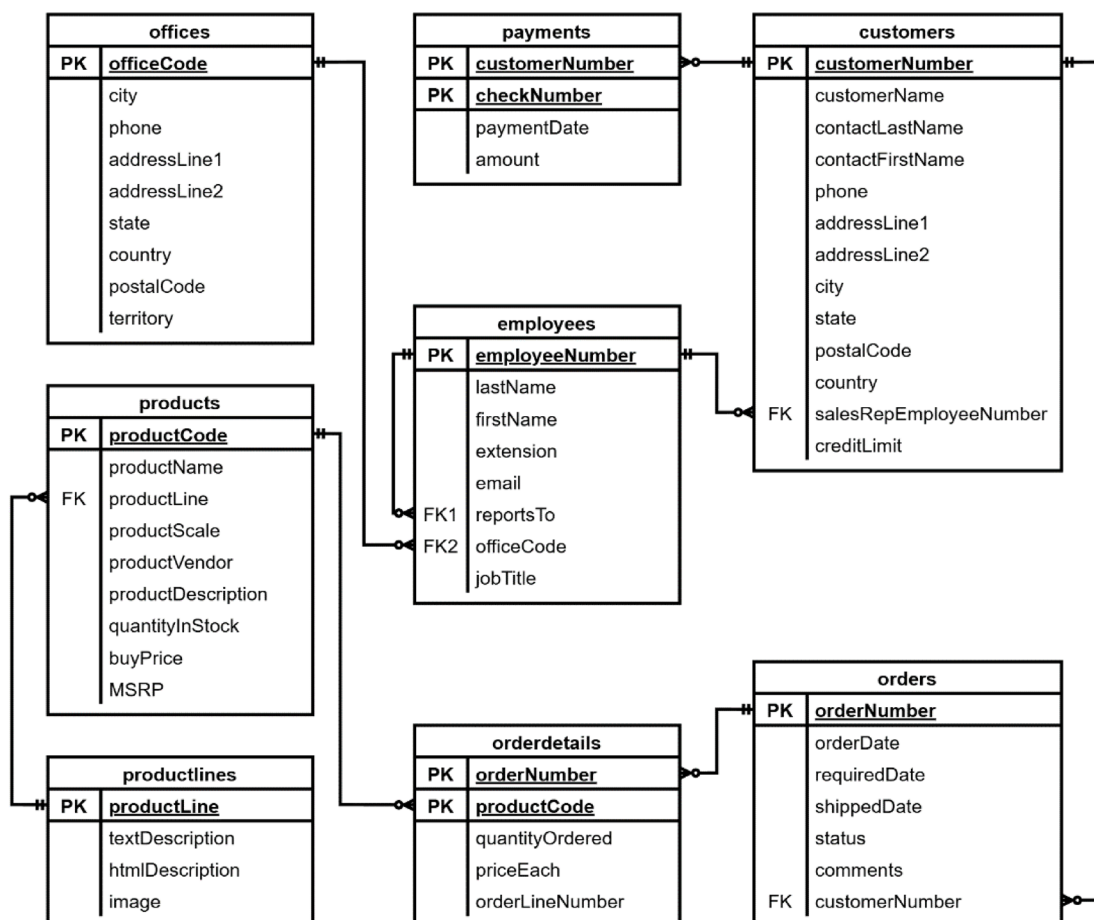
เพื่อประเมินความสามารถของระบบที่พัฒนาขึ้น ผู้วิจัยได้ออกแบบกระบวนการวัดผลอย่างเป็นระบบ โดยมุ่งเน้นการตรวจสอบความถูกต้องของคำสั่ง SQL ที่ได้จากการแปลงข้อความภาษาธรรมชาติ รวมถึงประสิทธิภาพการทำงานของระบบแบบครบวงจร (end-to-end) ตั้งแต่การรับคำถามจนถึงการแสดงผลลัพธ์ในการประเมินผลครั้งนี้ ผู้วิจัยได้จัดทำชุดคำถามภาษาธรรมชาติจำนวนทั้งสิ้น 30 ข้อ ครอบคลุมหลากหลายโครงสร้างของคำสั่ง SQL เช่น การเลือกข้อมูลแบบมีเงื่อนไข การจัดกลุ่ม การเชื่อมโยงหลายตาราง และการเรียงลำดับ โดยคำถามทั้งหมดถูกนำเข้าสู่ระบบผ่านอินเทอร์เฟซของโมเดล My GPTs เพื่อแปลงเป็นคำสั่ง SQL และส่งไปยัง Web API เพื่อดำเนินการกับฐานข้อมูลจริง โดยผลลัพธ์ที่ได้จากระบบจะถูกตรวจสอบจากผู้เชี่ยวชาญด้านฐานข้อมูล เพื่อพิจารณาความถูกต้องของผลลัพธ์ที่ได้จากฐานข้อมูล

การวัดผลจะครอบคลุมทั้งด้านความถูกต้อง (accuracy) ความสมบูรณ์ของกระบวนการทำงาน (functional success) และประสิทธิภาพการตอบสนองของระบบ (response time) โดยผลการประเมินจะถูกนำเสนอในรูปแบบเชิงสถิติคือ ร้อยละของความถูกต้อง ค่าความสำเร็จในการดำเนินการ และค่าเฉลี่ยเวลาการตอบกลับ เพื่อใช้เป็นเกณฑ์ในการพิจารณาความพร้อมของระบบสำหรับการนำไปใช้งานจริง

ผลการวิจัย

ระบบ My GPTs ที่พัฒนาขึ้นในงานวิจัยนี้ถูกออกแบบให้สามารถใช้งานผ่านแพลตฟอร์ม ChatGPT ของ OpenAI ผู้ใช้ทั่วไปสามารถเข้าถึงระบบ My GPTs ได้โดย

การเข้าสู่ระบบผ่านเว็บไซต์ของ OpenAI แล้วเลือกโมเดล My GPTs จากรายชื่อ GPT ที่เปิดให้ใช้งานสาธารณะหรือผ่านลิงก์เฉพาะที่ผู้พัฒนาจัดเตรียมไว้ โดยในการประเมินประสิทธิภาพของระบบ LLM สำหรับการแปลงข้อความภาษาไทยเป็นคำสั่ง SQL (Text-to-SQL) ผู้วิจัยได้ออกแบบคำถามที่มีลักษณะการดึงข้อมูลจากเพียงหนึ่งตาราง โดยไม่มีการซ้อนคำสั่งหรือความสัมพันธ์กับตารางอื่น โดยตั้งคำถามว่า “ให้แสดงข้อมูลสำนักงาน (officeCode city state และ country) ที่อยู่ในประเทศ USA France และ Australia” ระบบสามารถเข้าใจคำถามภาษาไทย และแปลงเป็นคำสั่ง SQL ได้อย่างถูกต้อง ดังแสดงในภาพ 5 ซึ่งจะพบว่า ในกรณีที่เป็น คำถามพื้นฐานจาก 1 ตาราง และไม่มีการซ้อนเงื่อนไขซับซ้อน ระบบสามารถตอบกลับด้วย SQL ที่แม่นยำ



ภาพ 4 แผนภาพ ER ของฐานข้อมูล

ให้แสดงข้อมูลสำนักงาน (officeCode, city, state, country) ที่
อยู่ในประเทศ USA, France , Australia

Talked to api.theptsatri.com ▾

officeCode	city	state	country
1	San Francisco	CA	USA
2	Boston	MA	USA
3	NYC	NY	USA
4	Paris		France
6	Sydney		Australia

```
SELECT officeCode, city, state, country FROM offices WHERE country IN ('USA',  
'France', 'Australia')
```

(Execution time: 0.04 sec)

ภาพ 5 การแสดงผลจากคำสั่ง SQL

เพื่อประเมินศักยภาพของระบบ LLM ในการสร้างคำสั่ง SQL ที่มีความซับซ้อนยิ่งขึ้น ผู้วิจัยได้ออกแบบคำถามที่ต้องใช้ การซ้อนคำสั่งย่อย (subquery) เพื่อคำนวณค่ากลางจากข้อมูลภายในตารางเดียวกัน แล้วใช้ค่านั้นเป็นเงื่อนไขสำหรับการกรองข้อมูลเพิ่มเติม ตัวอย่างคำถามที่ใช้ในการทดสอบ คือ “ให้แสดงข้อมูลลูกค้าที่ประกอบไปด้วย customerName contactFirstName และ phone ที่อยู่ในประเทศเดียวกับลูกค้าที่ชื่อ Vida Sport, Ltd.” ดังแสดงในภาพ 6

ผลการทดสอบสามารถสังเกตได้ว่า LLM มีความสามารถในการจัดโครงสร้างคำสั่ง SQL ที่มีคำสั่งย่อยได้อย่างถูกต้อง โดยสามารถระบุลำดับการประมวลผลของ subquery ได้อย่างเหมาะสม ซึ่งถือเป็นความสามารถระดับสูงในการเข้าใจคำถามที่มีบริบทแฝงซึ่งต้องตีความจากข้อมูลภายในฐานข้อมูลเอง

นอกจากนี้ยังทำการตรวจสอบศักยภาพของระบบ LLM ในการสร้างคำสั่ง SQL ที่ต้องอาศัยความเข้าใจในความสัมพันธ์ระหว่างหลายตาราง ผู้วิจัยได้ออกแบบคำถาม

ที่ต้องใช้การเชื่อมโยงข้อมูลระหว่างตาราง customers และ employees ผ่านความสัมพันธ์ของคีย์นอก ได้แก่ salesRepEmployeeNumber โจทย์คำถามที่ใช้ในการทดสอบ คือ “ให้แสดงข้อมูลลูกค้าที่มีข้อมูลดังนี้ customerName contactLastName และแสดงรายละเอียดของพนักงานที่เป็นคนดูแลลูกค้ารายนั้น ๆ (lastName) เรียงลำดับตามรหัสลูกค้าจากน้อยไปมาก แสดง 5 ลำดับแรก” ดังแสดงในภาพ 7

ผลลัพธ์ที่ได้แสดงข้อมูลของลูกค้ารายแรกจำนวน 5 ราย พร้อมชื่อของพนักงานที่ดูแลลูกค้าแต่ละรายอย่างครบถ้วน โดยระบบสามารถเข้าใจได้ว่าเงื่อนไข salesRepEmployeeNumber = employeeNumber เป็นความสัมพันธ์ระหว่างตารางที่ต้องใช้ในการเชื่อมโยงข้อมูลในด้านการวิเคราะห์คำสั่ง SQL ผิดพลาด ผู้วิจัยได้ตั้งคำถามที่มีโครงสร้างง่าย โดยต้องการดึงข้อมูลพนักงานจากตาราง employees ที่มีตำแหน่งงาน (jobTitle) เป็น “Sales Manager” ดังตัวอย่างคำถาม “ให้แสดงข้อมูลพนักงาน (employees) ดังนี้ firstName lastName และ

email โดยเลือกเฉพาะ jobTitle เป็น Sales Manager” ดังแสดงในภาพ 8 แม้ว่า รูปแบบของคำสั่ง SQL จะถูกต้องตามไวยากรณ์ แต่ผลลัพธ์ที่ได้คือ “ไม่มีข้อมูลที่ตรงกับคำค้นในตำแหน่งงาน “Sales Manager” ซึ่งระบบควรมีการตรวจสอบค่าที่ใช้เปรียบเทียบในคำสั่ง SQL กับค่าที่มีอยู่จริงในฐานข้อมูลก่อน หรือมีฟังก์ชันช่วยแนะนำค่าที่ใกล้เคียงที่สุด เช่น ใช้ LIKE IN หรือแสดงค่าที่พบมากที่สุดในการคอลัมน์นั้นเพื่อช่วยผู้ใช้ในการตัดสินใจ

เพื่อประเมินประสิทธิภาพของระบบที่พัฒนาขึ้นในการแปลงข้อความภาษาธรรมชาติให้เป็นคำสั่ง SQL อย่างถูกต้องและสามารถประมวลผลกับฐานข้อมูลจริงได้อย่างสมบูรณ์ ผู้วิจัยได้ดำเนินการทดสอบระบบโดยใช้ชุดคำถามภาษาธรรมชาติทั้งหมดจำนวน 100 คำถาม ซึ่งครอบคลุมลักษณะคำถามหลากหลายรูปแบบ ทั้งการดึงข้อมูลพื้นฐาน การใช้คำสั่งซ้อน การเชื่อมโยงหลายตาราง การจัดกลุ่มข้อมูล ตลอดจนคำถามที่อาจไม่ตรงกับข้อมูลที่มีอยู่จริง คำถามทั้งหมดถูกนำเข้าสู่ระบบผ่านอินเทอร์เฟซของโมเดล LLM ที่พัฒนาในรูปแบบ My GPTs ซึ่งจะทำการ

แปลงข้อความเป็นคำสั่ง SQL และส่งผ่าน Web API ไปยังฐานข้อมูลเพื่อดึงผลลัพธ์กลับมาแสดงต่อผู้ใช้ โดยผลลัพธ์แต่ละชุดได้รับการตรวจสอบจากผู้เชี่ยวชาญด้านฐานข้อมูลเพื่อประเมินความถูกต้องของคำสั่ง SQL และความสมบูรณ์ของกระบวนการทำงาน ผลการประเมินครอบคลุม 3 ด้านหลัก ได้แก่ ความถูกต้องของคำสั่ง SQL (accuracy) ความสมบูรณ์ของการทำงาน (functional success) และเวลาเฉลี่ยในการตอบกลับ (response time) รายละเอียดผลการทดสอบจำแนกตามลักษณะคำถามแสดงไว้ในตาราง 1

การประเมินระบบแปลงภาษาธรรมชาติเป็นคำสั่ง SQL จากชุดคำถามจำนวน 100 ข้อ พบว่าระบบสามารถสร้างคำสั่ง SQL ได้อย่างถูกต้องในอัตราร้อยละ 92 และดำเนินการทำงานได้ครบถ้วนในทุกกรณี (ร้อยละ 100) โดยมีเวลาเฉลี่ยในการตอบกลับเพียง 0.0048 วินาทีต่อคำถาม แสดงให้เห็นถึงความสามารถของระบบทั้งในด้านความแม่นยำ ความเสถียร และประสิทธิภาพที่เหมาะสมต่อการนำไปประยุกต์ใช้ในระบบการจัดการฐานข้อมูลที่ขับเคลื่อนด้วยปัญญาประดิษฐ์

ให้แสดงข้อมูลลูกค้าที่ประกอบไปด้วย customerName, contactFirstName และ phone ที่อยู่ประเทศเดียวกับลูกค้าที่ชื่อ Vida Sport, Ltd

Talked to api.theptsatri.com ▾

customerName	contactFirstName	phone
BG&E Collectables	Ed	+41 26 425 50 01
Vida Sport, Ltd	Mihael	0897-034555
Precious Collectables	Braun	0452-076555

```
SELECT customerName, contactFirstName, phone FROM customers WHERE country =  
(SELECT country FROM customers WHERE customerName = 'Vida Sport, Ltd')  
(Execution time: 0.0044 sec)
```

ภาพ 6 ผลลัพธ์จากโครงสร้าง SQL ที่มีคำสั่งย่อย

ให้แสดงข้อมูลลูกค้าซึ่งให้มีข้อมูลดังนี้ customerName, contactLastName และแสดงรายละเอียดของพนักงานที่เป็นคนดูแลลูกค้ารายนั้นๆ (lastName) เรียงลำดับตามรหัสลูกค้าจากน้อยไปมาก แสดง 5 ลำดับแรก

Talked to api.thepsatri.com ▾

customerName	contactLastName	salesRepLastName
Atelier graphique	Schmitt	Hernandez
Signal Gift Stores	King	Thompson
Australian Collectors, Co.	Ferguson	Fixter
La Rochelle Gifts	Labrune	Hernandez
Baane Mini Imports	Bergulfsen	Jones

```
SELECT c.customerName, c.contactLastName, e.lastName AS salesRepLastName FROM
customers c JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber ORDER
BY c.customerNumber ASC LIMIT 5
(Execution time: 0.0017 sec)
```

ภาพ 7 ผลลัพธ์จากโครงสร้าง SQL ที่มีการรวมตาราง

ให้แสดงข้อมูลพนักงาน (Employees) ดังนี้ firstName, lastName และ email โดยเลือกเฉพาะ jobTitle เป็น Sales Manager

Talked to api.thepsatri.com ▾

ไม่มีข้อมูลที่ตรงกับคำค้นหาในตำแหน่งงาน "Sales Manager"

```
SELECT firstName, lastName, email FROM employees WHERE jobTitle = 'Sales Manager'
(Execution time: 0.0022 sec)
```

ภาพ 8 ผลลัพธ์การดำเนินงานที่ผิดพลาด

ตาราง 1

ผลการทดสอบจำแนกตามลักษณะคำถาม

Question Type	No. of Questions	Accuracy (%)	Functional Success (%)	Response Time (S)
1. Basic Selection	20	100	100	0.0042
2. Nested Queries	20	100	100	0.0050
3. Join Queries	20	90	100	0.0047
4. Group & Aggregate	20	85	100	0.0059
5. No Match/Value Error	20	85	100	0.0040
Total	100	92	100	0.0048

อภิปรายและสรุปผลการวิจัย

ผลการประเมินระบบ Text-to-SQL ด้วย ChatGPT แสดงให้เห็นถึงศักยภาพในการแปลงข้อความภาษาธรรมชาติให้เป็นคำสั่ง SQL ได้อย่างแม่นยำ และสามารถดำเนินการกระบวนการสืบค้นข้อมูลแบบครบวงจร (end-to-end) ได้อย่างสมบูรณ์ โดยเฉพาะอย่างยิ่งในด้านความถูกต้องของคำสั่งและความเร็วในการประมวลผลที่อยู่ในระดับเหมาะสมกับการใช้งานจริง กลุ่มคำถามประเภท Group & Aggregate และ No Match มีความถูกต้องต่ำกว่ากลุ่มอื่น เนื่องจากคำถามต้องอาศัยการตีความเงื่อนไขเชิงสรุปความคลาดเคลื่อนส่วนใหญ่เกิดจากการเลือกโครงสร้าง SQL ที่ไม่ตรงกับเจตนาของคำถาม แม้ว่าคำสั่ง SQL ที่สร้างขึ้นสามารถรันได้สำเร็จ ผลการวิจัยครั้งนี้สอดคล้องกับงานของ Pornphol and Chittayasothorn (2024) ซึ่งยืนยันว่า LLM เช่น ChatGPT สามารถสร้างคำสั่ง SQL ที่ถูกต้องทั้งในรูปแบบพื้นฐานและซับซ้อน แม้จะยังมีข้อจำกัดในกรณีคำถามที่เฉพาะเจาะจงมาก นอกจากนี้ผลการทดลองยังมีความสอดคล้องกับแนวทางของ Liu et al. (2025) และ Shen et al. (2023) ซึ่งชี้ให้เห็นว่า ความแข็งแกร่งของระบบ Text-to-SQL ไม่ได้ขึ้นอยู่กับโมเดลเท่านั้น แต่ยังรวมถึงการจัดการข้อมูลเสริม (data augmentation) ด้วย นอกจากนี้ ยังแสดงให้เห็นว่า ChatGPT สามารถสร้างคำสั่ง SQL ได้อย่างแม่นยำในระดับโครงสร้างสอดคล้องกับงานของ Cagliero et al. (2024) แม้ในกรณีคำสั่งที่

ซับซ้อนและผลการทดลองยังสอดคล้องกับงานวิจัยของ Nilsiam et al. (2025) ที่เน้นการนำเทคนิค Text-to-SQL ที่ใช้ LLM ไปเชื่อมกับฐานข้อมูลจริง อย่างไรก็ตาม จุดเด่นของระบบที่นำเสนอครั้งนี้ คือการมุ่งเน้นไปที่การนำไปใช้งานจริงผ่าน API ที่เชื่อมกับฐานข้อมูลจริง โดยไม่ต้องพึ่งพาการฝึกฝนโมเดลเพิ่มเติมหรือโครงสร้างที่ซับซ้อนมากนัก

จากผลการประเมินพบว่า ระบบที่พัฒนาขึ้นสามารถตอบสนองคำถามที่มีโครงสร้างหลากหลายได้อย่างมีประสิทธิภาพ โดยใช้เวลาประมวลผลเฉลี่ยเพียง 0.0048 วินาที และสามารถดำเนินการได้อย่างสมบูรณ์โดยไม่เกิดข้อผิดพลาด แสดงให้เห็นถึงความพร้อมของระบบสำหรับการใช้งานในบริบทจริง เช่น การช่วยเหลือนักเรียน นักศึกษา หรือผู้ใช้ทั่วไปที่ไม่มีความรู้ SQL มากนัก

ในภาพรวม งานวิจัยนี้สนับสนุนแนวคิดที่ว่าแม้ระบบ LLM ขั้นสูงจะมีบทบาทสำคัญในด้านความแม่นยำและการเรียนรู้ข้ามโดเมน แต่ระบบที่ออกแบบโดยเน้นความเข้าใจบริบทของผู้ใช้และการเชื่อมต่อกับฐานข้อมูลจริง (MySQL-GPT) ก็สามารถให้ผลลัพธ์ที่มีประสิทธิภาพและใช้งานได้จริงเช่นกัน ทั้งนี้ ข้อเสนอแนะในอนาคตควรพิจารณาการเพิ่มกลไกตรวจสอบความสอดคล้องกับ schema และเสนอการแก้ไขอัตโนมัติเมื่อเกิดข้อผิดพลาดเพื่อเพิ่มความยืดหยุ่นและความแม่นยำของระบบให้มากยิ่งขึ้น



References

- Cagliero, L., Farinetti, L., Fior, J., & Manenti, A. I. (2024). ChatGPT, be my teaching assistant! automatic correction of SQL exercises. In L. O’Conner (Ed.), *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 81–86). IEEE.
- Carr, N., Shawon, F. R., & Jamil, H. M. (2023). An experiment on leveraging ChatGPT for online teaching and assessment of database students. In M. Kuo & R. Sinha (Eds.), *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)* (pp. 1–8). IEEE.
- Deshpande, A., Mane, P., Kothari, D., Salvi, A., & Kolhe, V. (2022). Querylizer: An interactive platform for database design and text to SQL conversion. In S. A. Thorat, A. Adamuthe & D. Gupta (Eds.), *Proceedings of the International Conference on Advanced Technology (ICONAT)* (pp. 1–6). IEEE.
- Lan, Z., & Li, S. (2024). PS-SQL: Phrase-based schema-linking with pre-trained language models for Text-to-SQL parsing. In E. X. Wang & R. Minasian (Eds.), *Proceedings of the 6th International Conference on Natural Language Processing (ICNLP)* (pp. 31–34). IEEE.
- Liu, S., Almohaimeed, S., & Wang, L. (2025). *REFORMER: A ChatGPT-driven data synthesis framework elevating Text-to-SQL models*. arXiv:2504.04363v1. <https://doi.org/10.48550/arXiv.2504.04363>
- MySQLTutorial.org. (2025). *MySQL sample database*. <https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/>
- Nilsiam, Y., Khammaw, P., & Nuchitprasitchai, S. (2025). OpenAI-powered NLP for SQL generation: Enhancing database access for non-technical users [In Thai]. In K. Chaiwong (Ed.), *The 10th International Conference on Digital Arts, Media and Technology (DAMT) and 8th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (NCON)* (pp. 423–428). IEEE.
- Ning, Z., Zhang, D., Zhang, L., Yu, H., & Wan, F. (2022). Review of question answering technology based on Text-to-SQL. *Proceedings of the IEEE International Conference on Computer Science Education and Informatization (CSEI)* (pp. 167–172). IEEE.
- OpenAI. (2022). *Introducing ChatGPT*. <https://openai.com/index/chatgpt>
- OpenAI. (2023). *Creating a GPT*. <https://help.openai.com/en/articles/8554397-creating-a-gpt>

- Pornphol, P., & Chittayasothorn, S. (2023). Verification of relational database languages codes generated by ChatGPT. In Y. Watanobe, G. Mogos & M. Rahman (Eds.), *Proceedings of the 4th Asia Service Science and Software Engineering Conference (ASSE)* (pp. 17–22). ACM.
- Pornphol, P., & Chittayasothorn, S. (2024). Using LLM artificial intelligence systems as complex SQL programming assistants. In S. Yamaguchi & N. Funabiki (Eds.), *Proceedings of the 12th International Conference on Information and Education Technology (ICIET)* (pp. 187–190). IEEE.
- Shen, H., Sun, G., Wang, Y., Shen, R., Li, Y., & Zhang, P. (2023). Sequential feature augmentation for robust Text-to-SQL. In L. Trinh (Ed.), *Proceedings of the International Conference on Algorithms, Computing and Data Processing (ACDP)* (pp. 217–221). IEEE.
- Srisuruk, T., & Chittayasothorn, S. (2023). An approach for rapid source code development based on ChatGPT and prompt engineering. *IEEE Access*, 12, 53074–53087.
<https://doi.org/10.1109/ACCESS.2024.3385682>

