從鍵位到語意：推理式大型語言模型在雙語夾雜輸入法之應用
From Keystrokes to Semantics: Applying Reasoning Large Language Models to Bilingual Code-Mixed Input Method Restoration

Hsin-Jui Li[1*], Yu-Chien Chung[1*], Po-Chun Chen[2], Fang-Lan Huang[1], Hen-Hsen Huang[3] & Hsin-Hsi Chen[2,4]
[1]Taipei First Girls High School, Taiwan
[2]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[3]Institute of Information Science, Academia Sinica, Taiwan
[4]AI Research Center (AINTU), National Taiwan University, Taiwan
{lhj9833tfg, angelinachung0319}@gmail.com, flhuang@gapps.fg.tp.edu.tw
pcchen@nlg.csie.ntu.edu.tw, hhhuang@iis.sinica.edu.tw, hhchen@ntu.edu.tw

摘要

注音輸入法使用者常因忘記切換輸入法而產生鍵盤字串，例如欲輸入「火車有 delay 了」，卻得到「cji3tk u.3delayxk7」。本研究將其界定爲「高歧義符號任務」，提出 HKIF（Hierarchical Knowledge-Injected Framework）架構，透過顯性知識注入、結構化推理與語意整合三層設計進行還原。本研究以 ATOM（ASCII to Mandarin）提示詞方案實作 HKIF，比較不同推理深度與提示詞下的還原效果，並延伸至倉頡輸入法及日文、韓文語境驗證。結果顯示 HKIF 架構能有效提升還原準確度（ROUGE-L 最高達 0.74），證明分層解碼設計適用於高歧義符號任務。

Abstract

Users of the Zhuyin input method often produce keystroke strings because they forget to switch input methods; for example, intending to type "火車有 delay 了", they obtain "cji3tk u.3delayxk7". This study characterizes this problem as a *high-ambiguity symbolic task* and proposes the HKIF (Hierarchical Knowledge-Injected Framework) architecture, which performs restoration through a three-layer design of explicit knowledge injection, structured reasoning, and semantic integration. This study implements HKIF with the ATOM (ASCII to Mandarin) prompting scheme, compares restoration performance under different reasoning depths and prompts, and extends the validation to the Cangjie input method as well as Japanese and Korean contexts. Results show that the HKIF architecture effectively improves restoration accuracy (ROUGE-L up to 0.74), demonstrating that the hierarchical decoding design is suitable for high-ambiguity symbolic tasks.

關鍵字：鍵位字串還原、大型語言模型、提示工程、中英夾雜

Keywords: Keystroke String Restoration, Large Language Models, Prompt Engineering, Code-Switching

1 緒論

在日常使用電腦的過程中，使用者經常因爲忘記切換輸入法，導致輸入出現一大串無意義字串。例如，意圖輸入「火車有 delay 了」，卻在螢幕上看到「cji3tk u.3delayxk7」，如圖 1 所示。此現象源於大千式注音鍵盤佈局：37 個注音符號與 4 個聲調符號對應至 QWERTY 鍵盤 (Apple, 2024)。「火」字的注音「ㄏㄨㄛˇ」對應按鍵 c-j-i-3，在英文模式下即產生「cji3」。使用者專注於內容時，往往在輸入整段文字後才發現錯誤，只能刪除重打，嚴重影響效率。

因此，本研究希望開發一個系統框架，能自動將鍵盤字串轉換回對應的繁體中文內容，並支援現代網路語言中普遍存在的中英夾雜輸入方式。針對市面常見工具與相關研究的系統性分析顯示，現有方案均無法完整解決此任務：華碩智慧輸入法 (ASUS, 2024) 的注音與英文混合輸入模式僅能預防問題發生，對已產生的字串仍需手動刪除重打，無法事後還原；

*Equal contribution.

Proceedings of the 38th Conference on Computational Linguistics and Speech Processing (ROCLING 2026), pages 266–278
November 6-7, 2026 ©2026 Association for Computational Linguistics



圖 1: 忘記切換輸入法導致的字串輸出（鍵盤來源 Apple, 2024，其餘爲作者繪製）。

Google 輸入工具 (Google, 2024) 的核心設計爲即時輸入轉換，並非針對已產生之字串進行反向解析，且中英夾雜時傾向將所有字元視爲注音輸入，導致英文部分被錯誤轉換；王冠侖 (2024) 提出的 bopomofo2hanzi 以規則比對與機器學習進行字串還原，爲本領域先驅，但假設輸入字串皆爲注音對應字元，遇到「delay」時會將其錯誤解析爲注音序列（d= ㄎ、e= ㄍ、l= ㄠ、a= ㄇ、y= ㄗ），使還原結果偏離原意。各方法的功能對比如表 1 所示。

方法類型	中英夾雜	語意推理	事後修復
華碩智慧輸入法	✓	✓	×
Google 輸入工具	×	×	×
bopomofo2hanzi	×	✓	✓
本研究	✓	✓	✓

表 1: 既有方法的功能對比。

近年來，以 ChatGPT 爲代表的大型語言模型（LLM）展現出強大的語意理解與多語言處理能力。本研究期望結合 LLM 的語意理解與推理能力突破上述限制，特別是解決中英夾雜辨識與語境推理兩大核心問題。本研究的貢獻如下：(一) 設計並實作一套基於 LLM 的「輸入字串 → 推測注音 → 還原通順中文」處理流程，驗證此技術路徑的可行性；(二) 系統性評估推理深度對字串還原準確度的影響；(三) 解決中英夾雜與口語化用語（如 emo、XD）的辨識問題；(四) 以倉頡、韓文、日文輸入法爲延伸案例，驗證所提 HKIF 框架僅需替換鍵位對照表即可跨輸入法遷移的通用性。

2 相關研究

2.1 中文輸入法錯誤修正

過去關於中文輸入法的研究，多集中於輸入法已正確啓用的前提下，處理拼音或注音輸入過程中的錯誤，例如拼寫錯誤、同音字混淆或候選字排序問題。早期研究即以統計語言模型處理拼音到漢字的轉換與候選字排序 (Chen and Lee, 2000; Wang and Seneff, 2000)。中文拼寫校正（Chinese Spelling Check）亦有系列公開評測推動相關技術發展 (Tseng et al., 2015)，近年如 Soft-Masked BERT (Zhang et al., 2020) 等深度模型進一步提升偵錯與校正表現。隨著資料與模型規模提升，Hu et al. (2022) 提出 CSCD-IME 資料集，專注於拼音輸入法中因拼寫錯誤或同音字選擇造成的錯誤，透過錯誤資料合成與模型訓練顯著提升校正效果。然而，上述研究普遍假設輸入內容仍爲具備語言層級意義的拼音或注音序列。當使用者完全未切換輸入法時，鍵盤輸出的是一串直接對應 QWERTY 鍵盤的英文字母、符號與數字組合，單一字元可能同時對應英文字母、注音符號或聲調標記，錯誤型態與既有錯字校正問題在本質上已有明顯差異。

2.2 跨輸入法整合與切換輔助

部分研究關注使用者在多種輸入法間頻繁切換造成的操作負擔與錯誤風險。李倬安 et al. (2024) 提出基於深度學習的跨多輸入法編輯器整合系統，即時分析鍵盤輸入序列以判斷當下應使用之輸入法。此類研究屬於輸入當下的線上輔助機制，核心目標在於預防錯誤發生，而非在錯誤已發生後進行修復。

2.3 中英夾雜文本處理

中英夾雜（code-switching）文本已成爲常見的語言現象，相關研究廣泛探討語碼轉換在語音辨識、語言模型訓練與下游任務中的影響 (Long et al., 2021; Chiou et al., 2022)。Winata et al. (2023) 對近十年 NLP 領域的語碼轉換研究進行系統性回顧，指出現有方法多著重於語言辨識與語言模型建構，並假設輸入內容本身爲可讀的語言符號序列。在鍵盤字串情境中，模型需先判斷哪些字元應視爲英文詞彙、哪些可能對應注音輸入，該判斷需同時仰賴鍵盤對照資訊與語意推理，與既有中英夾雜文本處理的問題設定存在明顯差異。

2.4 鍵盤字串還原

在目前已知文獻中，王冠侖 (2024) 爲少數明確針對「未切換注音輸入法所產生之鍵盤字串」進行系統性研究的作品，透過規則比對與機器學習建立鍵盤輸入與注音符號的對應關係。然而該方法假設輸入內容主要爲注音對應字元，當字串中包含應保留的英文單字、品牌名稱或網路用語時，容易將英文誤判爲注音序列而錯誤轉換，且在長文本或需語境判斷的情境下仍有限制。

2.5 大型語言模型與提示工程

以 ChatGPT (OpenAI, 2022) 爲代表的 LLM 在自然語言理解與推理上取得突破性進展，並具備僅憑少量示範即可完成新任務的上下文學習能力 (Brown et al., 2020)。近年更出現在推理階段投入更多運算的推理式模型，如 OpenAI o1 (OpenAI, 2024) 與 DeepSeek-R1 (Guo et al., 2025)；相關研究亦顯示，適當擴展測試時運算（test-time compute），對困難任務的增益可勝過單純擴大模型參數 (Snell et al., 2024)，本研究對推理深度的系統性比較即立基於此脈絡。除閉源模型外，OpenAI 釋出的開放權重模型 gpt-oss 系列 (OpenAI, 2025) 使研究者能在不訓練模型的前提下測試 LLM 於特定任務的表現。模型輸出品質高度仰賴提示詞設計，形成提示工程此一研究方向 (Liu et al., 2023)。其中思維鏈（Chain-of-Thought, CoT）提示被證實能有效提升多步推理表現 (Wei et al., 2023)；即使不提供推理示範，僅加入「讓我們一步一步思考」等指令亦能觸發逐步推理，稱爲 Zero-shot CoT (Kojima et al., 2023)。後續尚有自動生成推理示範的 Auto-CoT (Zhang et al., 2022) 與整合多條推理路徑的 Self-Consistency (Wang et al., 2023)。LLM 的發展使「透過語意與上下文推理還原使用者原始輸入意圖」成爲可行，爲本研究提供關鍵技術基礎。

2.6 文本相似度評估指標

Papineni et al. (2002) 提出的 BLEU 以 n-gram 精確率爲核心並加入簡短懲罰，廣泛應用於機器翻譯。Lin (2004) 提出的 ROUGE-L 以最長公共子序列（Longest Common Subsequence, LCS）爲基礎、以召回率爲核心，僅要求詞語保持相對順序而不要求連續出現，能自然捕捉句子層面的語序資訊。由於鍵盤字串還原任務的目標在於「最大程度還原原始輸入內容」，漏掉關鍵詞將嚴重影響可讀性，本研究選用召回導向的 ROUGE-L 作爲主要評估指標。

綜合上述，既有研究多分別處理輸入錯誤修正、切換輔助、中英夾雜理解或 LLM 應用，尚缺乏方法能同時處理鍵盤層級歧義、中英夾雜判斷與長文本語意推理三項挑戰。本研究針對此缺口提出 HKIF 架構。

3 問題定義：高歧義符號任務

令輸入字串爲 $X = (x_1, x_2, \ldots, x_m)$，其中 x_i 爲一個可見字元（英文字母、數字、標點、空白等）；令標準答案（原句的繁體中文還原結果）爲 $Y = (y_1, y_2, \ldots, y_n)$，其中 y_j 爲一個中文字、標點符號或英文單字；令 LLM 在提示詞 P 下對輸入 X 的輸出爲 $Y' = (y'_1, y'_2, \ldots, y'_k)$。本研究將字串還原形式化爲條件機率最大化問題：給定鍵位字串 X，尋找最可能的原始語句

$$Y^* = \arg\max_{Y} P(Y \mid X), \tag{1}$$

並以評估函數 $L(Y, Y')$（主要採 ROUGE-L）衡量還原品質。

字串資料 X 模擬「忘記從英文輸入法切換至注音輸入法」的情境，由每一筆原句 Y 經兩個轉換函數生成：先透過 $G_{\text{zhuyin}} : Y \to Z$ 將中文轉爲注音符號序列 Z，再透過 $G_{\text{key}} : Z \to X$ 依注音鍵盤對應表映射至鍵位實際輸出的英數符號，整體記爲 $X = G_{\text{key}}(G_{\text{zhuyin}}(Y))$。英文單字、數字與標點等非中文部分保留原樣，以符合使用者在同一句中刻意輸入英文的眞實行爲。

本研究進一步將此問題界定爲「高歧義符號任務（High-Ambiguity Symbolic Task）」，指輸入序列尚未呈現意義、且單一符號可能對應多種語言單位、需依賴上下文語意判別之任務，其特徵包括：（一）符號層級歧義（單一字元多重映射）；（二）語境依賴性高；（三）無明確分詞邊界；（四）可能包含跨語言混合。爲形式化描述歧義程度，定義歧義密度函數

$$A(X) = \frac{a_1 + a_2 + a_3}{|X|}, \tag{2}$$

其中 a_1 爲鍵位多重映射比例，a_2 爲中英切換比例，a_3 爲同音字候選數量平均值。已呈現部分意義之自然文本，其歧義密度通常趨近於零；當輸入序列同時具有符號層多對一映射、語言切換不確定性與同音候選歧義時，$A(X)$ 顯著高於自然語言基準，即可界定爲高歧義符號任務。此定義使本問題與既有中文輸入法錯字修正形成明確區隔，也說明爲何此類任務需同時具備鍵位知識與語意推理能力。

4 HKIF 架構與 ATOM 實作

4.1 HKIF：分層知識注入框架

在鍵盤字串還原任務中，鍵位對照屬於可完全形式化之規則映射，而同音字選擇與中英混合判斷則屬語意推理問題。據此，本研究提出 HKIF（Hierarchical Knowledge-Injected Framework）架構，將解碼流程分爲三層（圖 2）：

第一層：顯性知識注入層（Explicit Knowledge Injection Layer），提供鍵位對照表與聲調處理規則，完成符號層級之形式化轉換，此層之輸出唯一且可預期。

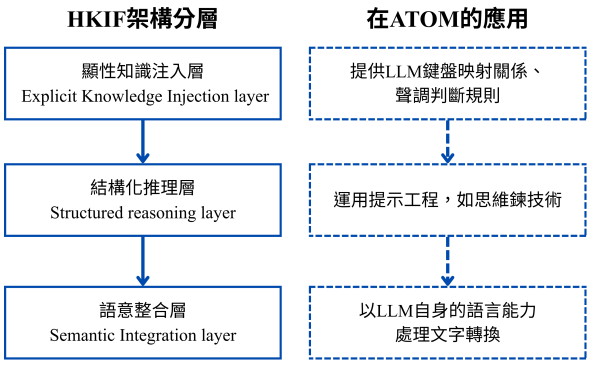

圖 2: HKIF 架構與 ATOM 實作。

聲調	一聲（陰平）	二聲	三聲	四聲	輕聲
符號	（無）	ˊ	ˇ	ˋ	·
對應字元	空白鍵	6	3	4	7

表 2: 大千式注音鍵盤之聲調對應。

第二層：結構化推理層（Structured Reasoning Layer），運用提示工程技巧引導 LLM 依明確步驟產出中間表示，提高文字轉換的效率及可追溯性。

第三層：語意整合層（Semantic Integration Layer），利用 LLM 進行語境推理與最終語句生成，處理同音字選擇、中英夾雜辨識與句子流暢化等語意層級問題。

此設計可降低語言模型於低層級映射任務上的負擔，使其專注於語意判斷與語境整合，提升整體穩定性與可解釋性。其中「顯性知識注入」指在模型推理前，將任務所需之結構性規則明確提供於輸入中，使模型可利用外部知識而非依賴其參數化知識：鍵位對照規則屬高度結構化知識，若未明確提供，模型僅能依語意機率猜測，易產生錯誤。

4.2 ATOM 提示詞方案

本研究提出通用提示詞方案 ATOM（ASCII to Mandarin）作爲 HKIF 架構之具體實作，核心想法是把「字串解碼」拆成模型可依序執行的步驟，並把最容易出錯的規則寫死在提示詞中。整體結構包含五個部分：(1) 任務說明：清楚描述輸入是「注音鍵位字串」，輸出要還原成正體中文；(2) 鍵位對照表：提供「英數符號 → 注音符號／聲調」的對應表，避免模型憑印象猜鍵位；(3) 操作步驟：要求模型先把字串轉回注音，再依語意轉成正體中文；(4) 規則補充：包含「英文單字不轉換」、「常見網路用語判別」、「未標聲調視爲一聲」、「標點與空白處理」等；(5) 輸出格式：指定最後一行輸出 `Answer: ...`，方便後續程式自動擷取。在推理提示上採用「請一步步思考」的 Zero-shot CoT 風格 (Kojima et al., 2023)，讓模型先完成中間的對照與拆解，再生成最終答案。ATOM 提示詞全文如附錄圖 5 所示。

方法選擇上，本研究採用提示工程而非微調（fine-tuning）：微調需額外資料蒐集、運算資源與訓練時間，且模型版本迭代快速時投入成本不一定划算；提示工程可在不改動模型參數的前提下，以低成本調整輸入格式與解題步驟，快速建立可重用的解碼流程 (Liu et al., 2023)。

爲評估各提示元素的貢獻，本研究設計多個消融變體，透過移除或替換特定元素，觀察其對整體性能的影響：ATOM-NoTable 移除鍵位對照表，其餘與完整版相同；ATOM-ShortTask 將任務說明簡化爲一句話，刪除情境鋪陳；ATOM-NoSteps 保留鍵位對照表但移除操作步驟；ATOM-NoTableNoSteps 則同時移除鍵位對照表與操作步驟，僅保留最基本的任務描述與輸出格式。

4.3 跨輸入法之延伸設計

爲驗證 HKIF 框架的可移植性，本研究延伸至倉頡、韓文與日文輸入法，僅替換顯性知識注入層之鍵位知識並依各輸入法的結構特性調整流程分工。

倉頡輸入法：倉頡以字根分解爲基礎，每個中文字對應至多五個字根碼、以空格分隔，字根碼與字元的對應關係相對明確。本研究設計混合式流程：(1) 英文識別階段（LLM）：判斷輸入字串中哪些片段屬於應保留的英文單字並暫時移出；(2) 倉頡碼解析階段（規則式）：以字根對照表系統性判斷各碼串所代表的字；(3) 語意整合階段（LLM）：將還原的中文片段與英文內容重新組合，依上下文調整輸出。此設計在可規則化的部分以確定性判斷取代 LLM 推理，降低模型的歧義處理負擔。

韓文輸入法：韓文透過標準兩發式（2-beolsik）鍵盤將初聲、中聲、終聲直接對應至鍵位，對應關係高度規則。流程爲：(1) 鍵位碼拆解：依兩發式規則拆解字串，處理雙子音（Shift ＋子音鍵位）與複合母音、雙終聲等複合結構；(2) 語意與字串識別：由模型依上下文識別應保留的英文單字並還原韓文文本；(3) 語法修正：確保韓英混寫句符合自然語法。

日文輸入法：日文採羅馬字輸入，對應關係明確且選字較單一。流程爲：(1) 羅馬字轉假名：依標準平文式或訓令式規則轉換爲假名，處理促音（下一字子音重複）、長音符與拗音等特殊情況；(2) 假名轉漢字：依上下文語意轉換爲漢字、平假名或片假名混寫形式；(3) 語法修正。韓文與日文輸入法之特殊鍵位規則

整理於附錄 B。

5 實驗設定

5.1 資料集建構

爲模擬「使用者忘記切換輸入法」的眞實情境，本研究建立成對的標準答案（原始句子 Y）與對應鍵位字串（模型輸入 X），整體流程如圖 3 所示，單筆資料之生成與評估流程示意見附錄 C。本研究共建構五個資料集：(1) School（中英構成）：以問卷形式於校內蒐集日常語句，共 159 句；(2) PTT-with English（中英構成）：從批踢踢實業坊（PTT）資料檔中隨機抽樣 600 句含中英混雜字元的句子；(3) PTT-without English（中文構成）：從 PTT 資料檔中隨機抽樣 600 句僅含中文字元的句子；(4) Korean-with English：從 HiKE 韓英雙語語音資料集 (Paik et al., 2026) 隨機抽樣 100 筆韓英雙語句子；(5) Japanese-without English：取自常用日語 1000 句合集之第 101–200 句 (上川日本語, 2025)。

在大千式注音鍵盤中，二、三、四聲與輕聲分別對應「6」「3」「4」「7」，一聲對應空白鍵（表 2）；由於一般注音標示通常不顯示一聲符號，模型在切分注音時容易斷錯位置。本研究在資料建構時保留必要的空白資訊，並在提示詞規則中明確宣告「若未出現聲調符號，視爲一聲」等處理方式。此外，注音轉換工具（曉聲通注音字典 ToneOZ, 2024）可能自動分詞、換行或插入額外空格，本研究以程式將注音結果重新串接成單行字串再套用鍵位映射，並保留英文、數字、標點的原始順序，使輸入更貼近「使用者一次打出的一整串字串」。

5.2 模型與生成參數

模型選用考量爲：具備優異推理能力、對注音有一定認識、能處理語意還原與中英夾雜情境，且爲開放權重模型，可本地部署以保障個資安全。各候選模型之評估整理於附錄 D 表 11。經性能與成本評估，本研究採用 gpt-oss-120b (OpenAI, 2025)：OpenAI 於 2025 年 8 月發布之開放權重推理模型，採 Apache 2.0 授權，爲 MoE（Mixture of Experts）架構，總參數量約 117B，每次推論激活約 5.1B 參數，可在單張 80GB GPU 上運行 (Raschka, 2025)。實驗透過 Cerebras Inference 與 NVIDIA NIM 兩個 API 平台呼叫此模型；在相同模型與相同解碼參數下，不同平台的推論結果視爲等價，同時使用兩者僅爲因應平台流量限制。該模型支援 high、medium、low 三種推理深度設定，本研究據此系統性分析推理強度對還原任務的影響。爲降低隨機取樣造成的輸出波動，各呼叫盡量固定關鍵生成參數：temperature 設爲 0、top-p 設爲 1、max tokens 設爲模型允許之最大輸出上限，並完整記錄模型版本與參數設定以利重現。

5.3 對照基準

本研究採用多種常見基準進行比較：(1) Direct Prompt：只提供任務說明與輸出格式，不含 CoT 指令；(2) Zero-shot CoT：在 Direct Prompt 基礎上加入「請一步步思考」(Kojima et al., 2023)；(3) Few-shot Prompt：提供數筆輸入字串與理想輸出作爲示範 (Liu et al., 2023)；(4) bopomofo2hanzi：先前研究訓練之機器學習模型，非 LLM 方法 (王冠侖, 2024)。四者分別代表「無提示」、「零示範推理」、「少量示範學習」及「傳統非 LLM 方法」，提供完整的對照視角。各基準提示詞之全文、示範例與消融變體之組成差異見附錄 E。

5.4 執行流程

本研究以程式串接 LLM API，對每筆測試資料執行：讀入待還原字串 X；對同一筆 X 分別套用不同提示詞設定（ATOM、各消融變體、各基準）呼叫模型取得 Y'；以固定格式存檔（模型名稱、提示詞名稱、輸出全文、擷取 `Answer:` 後的字串）；以 ROUGE-L 計算 Y' 與 Y 的相似度並彙整平均分數；最後進行錯誤案例之質性分析。

6 實驗結果與討論

6.1 與常見基準之比較

表 3 爲 ATOM 與三種提示基準在三個注音資料集上的 ROUGE-L 分數，圖 4 爲 School 資料集之視覺化比較。ATOM 在所有資料集與推理深度設定下均大幅領先：平均而言，ATOM（high）達 0.6186，而 Direct Prompt、Zero-shot CoT、Few-shot Prompt 僅約 0.07–0.17。

Direct Prompt 的分數極低（約 0.08–0.10），證實 LLM 在不具備外部知識的情況下，無法僅憑黑箱式的預測解碼非直觀的鍵位映射；即便加入「一步步思考」的 Zero-shot CoT，提升效果亦有限，說明本任務的難點不只在於邏輯推理，也包含「符號與語言間的映射」。Few-shot Prompt 的表現介於基準方法與 ATOM 之間：範例示範能幫助模型學習任務格式，但在缺乏明確對照表與操作指令下，模型容易產生錯誤類比（例如將英文單字誤認作注音編碼），在 PTT 等口語化且混雜英文的資料集上表現不佳。

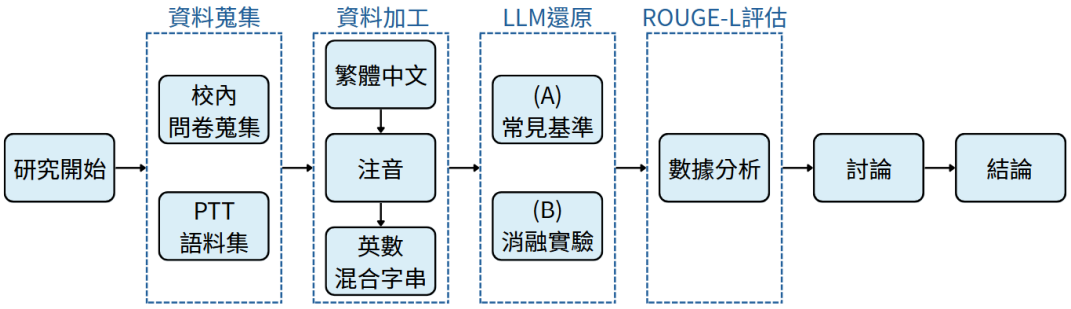

圖 3: 實驗流程圖。

資料集	推理深度	ATOM	Direct	ZS-CoT	Few-shot
School	high	**0.7424**	0.1179	0.1173	0.1578
	medium	0.6840	0.1264	0.1171	0.2913
	low	0.5791	0.1076	0.1222	0.2547
PTT-with English	high	**0.6218**	0.0898	0.0779	0.0473
	medium	0.5800	0.1415	0.1425	0.1434
	low	0.4515	0.1607	0.1500	0.1291
PTT-without English	high	**0.4917**	0.0271	0.0145	0.0204
	medium	0.4840	0.0442	0.0425	0.0847
	low	0.3379	0.0542	0.0500	0.0630
Average	high	**0.6186**	0.0783	0.0699	0.0752
	medium	0.5827	0.1040	0.1007	0.1731
	low	0.4562	0.1075	0.1074	0.1489

表 3: ATOM 與常見基準之 ROUGE-L 分數（gpt-oss-120b，三種推理深度）。

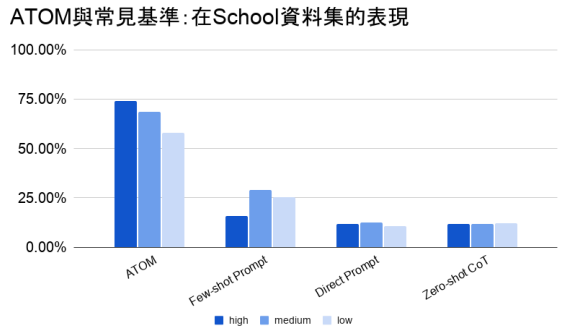

圖 4: ATOM 與常見基準在 School 資料集的表現。

6.2 與前人研究之比較

在從 PTT-without English 隨機抽取的 100 句上，ATOM（high）達 ROUGE-L 0.4953、（medium）0.4977，遠高於 bopomofo2hanzi 的 0.1328（表 4）。相較於規則比對方法，HKIF 在處理如「火車有 delay 了」這類中英夾雜語句時表現出強大的穩定性：由於 LLM 本身具備中英雙語的語意空間，遇到非注音編碼規律的英文字串（如 delay）時，語意整合層能依上下文判定該部分無需轉換，克服了傳統方法將英文單字強行解碼爲注音的困境。

推理深度	ATOM	bopomofo2hanzi
high	0.4953	
medium	**0.4977**	0.1328
low	0.3514	

表 4: ATOM 與 bopomofo2hanzi 於 PTT-without English 隨機抽取 100 句之比較。

6.3 消融實驗

表 5 拆解了 ATOM 成功的核心因素。首先，移除鍵位對照表（ATOM-NoTable）時分數出現斷崖式下跌（平均由 0.6186 跌至 0.0645），證明現有 LLM 的訓練資料並未包含足夠的「注音鍵盤佈局」與「QWERTY 字元」關聯知識，提供對照表是達成還原任務的必要條件，此結果直接驗證 HKIF 顯性知識注入層之設計。其次，移除操作步驟（ATOM-NoSteps）後，模型在不同推理深度間表現極不穩定（如 PTT-with English 由 high 的 0.6263 降至 low 的 0.2362），顯示「先轉注音再轉中文」的兩段式策略能防止模型跳過中間邏輯，強制產生可解釋之中間表示是提升解碼穩定性的關鍵設計。簡化任務說明（ATOM-ShortTask）的影

資料集	推理深度	ATOM	-NoTable	-ShortTask	-NoTableNoSteps	-NoSteps
School	high	**0.7424**	0.0861	0.6232	0.0554	0.4236
	medium	0.6840	0.2340	0.6293	0.1960	0.6015
	low	0.5791	0.1976	0.4855	0.1517	0.2333
PTT-with English	high	0.6218	0.0900	0.5549	0.0698	**0.6263**
	medium	0.5800	0.1338	0.5388	0.1361	0.4874
	low	0.4515	0.1766	0.4260	0.1505	0.2362
PTT-without English	high	0.4917	0.0174	0.4953	0.0211	**0.5334**
	medium	0.4840	0.0527	0.4832	0.0545	0.4326
	low	0.3379	0.0352	0.3324	0.0404	0.1414
Average	high	**0.6186**	0.0645	0.5578	0.0488	0.5278
	medium	0.5827	0.1402	0.5504	0.1289	0.5072
	low	0.4562	0.1365	0.4146	0.1142	0.2036

表 5: 消融實驗之 ROUGE-L 分數（gpt-oss-120b，三種推理深度）。

			常見基準			消融變體			
資料集	推理深度	ATOM	Direct	ZS-CoT	Few-shot	-NoTable	-ShortTask	-NoTable -NoSteps	-NoSteps
Korean-with English	high	0.4862	0.4699	0.4223	0.4093	0.4900	0.4735	0.4533	0.5551
	medium	**0.7998**	0.6914	0.3323	0.4009	0.8050	0.7694	0.7078	0.4932
	low	0.3827	0.2388	0.2466	0.2457	0.4114	0.3437	0.3214	0.2128
Japanese-without English	high	0.7371	0.6572	0.6706	0.8575	0.7843	0.8037	0.8363	0.6189
	medium	**0.9065**	0.6594	0.6907	0.8715	0.9030	0.9078	0.8731	0.7082
	low	0.8597	0.7114	0.6845	0.8404	0.8652	0.8909	0.8747	0.6656

表 6: 韓文與日文輸入法延伸實驗之 ROUGE-L 分數：ATOM 與常見基準及消融變體之比較（粗體爲 ATOM 各資料集最佳結果）。

響則相對較小。

6.4 推理深度的影響

觀察 gpt-oss-120b 在 low、medium、high 三種推理深度下的表現可發現：(1) 深度推理的優勢：在 high 設定下 ATOM 於 School 資料集達 0.7424，顯示當模型被賦予更多運算資源進行自我修正與邏輯檢核時，能更精準處理「同音異字」與「聲調缺失」導致的歧義；(2) 過度思考的影響：用詞常規的資料，分數會因過度思考而降低，出現 medium 高於 high 的現象，此與 LLM 研究中的 overthinking 問題一致 (Sui et al., 2025; Aggarwal et al., 2025)，過深的推理反而引入不必要的修改。

6.5 案例分析

本研究對三個注音資料集進行錯誤案例的質性分析，常見錯誤型態整理如表 7。PTT-without English 中有不少句子含專有名詞，例如「鶯谷園聽說不錯」中的地名「鶯谷園」較不常見，模型未考慮其爲名稱之可能性而選字錯誤。PTT-with English 中呈現不少新造詞及特殊輸入習慣，例如「肥宅喜歡玩 pokemongo 嗎」中的「pokemongo」爲遊戲 Pokemon Go 的變形輸入（忽略大小寫與空格），模型不易判斷此字串的類別。School 資料中英文多爲簡短單字，例如「不要 emo 啦」中的「emo」長度短且屬新興用語，模型可能將其辨識爲單一中文字的注音。

錯詞種類	例詞	常見誤判原因
專有名詞	鶯谷園	非常見名詞可能出現選字、分詞錯誤
特殊縮寫	pokemongo	不易辨識非正式的組合縮寫
新興短詞	emo	短詞可能被辨識爲一字的注音

表 7: 錯誤案例分析。

推理深度	PTT-without English	PTT-with English	School
high	0.8468	0.7141	0.7289
medium	0.8451	0.7122	0.7337
low	**0.8768**	**0.7338**	**0.7534**

表 8: 倉頡輸入法延伸實驗之 ROUGE-L 分數。

6.6 框架延伸性：倉頡、韓文、日文

表 8 與表 6 爲延伸實驗結果。倉頡還原在 PTT-without English 上達 ROUGE-L 0.8768、韓英資料集達 0.7998、日文資料集達

0.9065，均優於注音情境下的最佳結果。我們推測原因爲：當字根對應關係已由規則明確處理，LLM 得以將推理資源集中於其眞正擅長的語意判斷與中英夾雜識別，兩者分工反而帶來更高的整體準確率。值得注意的是，韓文與日文情境下各消融變體與 ATOM 完整版之差距明顯縮小，甚至互有高低，顯示當鍵位映射已由規則層處理時，提示詞中的對照表與操作步驟不再是效能瓶頸，此結果與 HKIF「可形式化部分交由規則、語意部分交由模型」的分工設計預期一致。

綜合注音與倉頡、韓文、日文幾組實驗，HKIF 架構的貢獻可總結爲兩點：其一，提出「顯性知識注入＋結構化推理」的通用設計原則，使解碼流程得以依不同輸入法的結構特性彈性調整，而無需重新訓練模型；其二，確立 LLM 在此類高歧義符號任務中的最適定位爲需要語意推理之處（雙語夾雜識別、同音字選擇、語意整合），而可形式化之低階映射則交由確定性規則層處理。單純依賴 LLM 進行符號還原表現不穩定，單純規則映射亦無法處理語境歧義；HKIF 透過分層設計將可確定部分與需語意推理部分分離，使模型專注於其優勢領域。

6.7 研究限制

本研究的鍵位字串係由原句模擬生成，與眞實使用者的錯誤分佈（如漏鍵、多鍵）可能存在落差，且各資料集規模有限。受限於運算資源與 API 成本，實驗僅以 gpt-oss-120b 進行，HKIF 在其他模型上的表現仍待驗證。此外，ROUGE-L 爲自動化指標，無法完全反映語意正確性，未來若資源允許，可輔以人工評估。再者，各實驗設定皆採 temperature 爲 0 之貪婪解碼（greedy decoding），且僅執行一次。

7 結論

本研究將鍵盤字串還原界定爲「高歧義符號任務」，提出 HKIF（Hierarchical Knowledge-Injected Framework）架構並以 ATOM（ASCII to Mandarin）提示詞方案實作，透過顯性知識注入、結構化推理與語意整合三層設計完成還原。實驗結果顯示：（一）推理式 LLM 在適當提示工程引導下，能以高準確率（最高 ROUGE-L 達 0.74）將誤打的鍵盤字串還原爲通順的正體中文；（二）成功還原依賴顯性知識（鍵位對照表）與結構化推理步驟的結合，僅靠模型內建知識或範例學習均不足；（三）LLM 的跨語言語意理解能準確區分注音編碼與原始英文內容，突破傳統規則模型處理現代網路用語的瓶頸；（四）推理深度對還原品質有決定性影響，惟亦觀察到過度思考現象；（五）HKIF 僅需替換鍵位對照表並調整流程分工，即可移植至倉頡、韓文、日文輸入法情境。

未來本架構可整合至輸入法事後修復工具、通訊軟體機器人或瀏覽器擴充功能，並依相同原則延伸至其他輸入法與語言，建立跨輸入法、跨語言的通用字串還原方案。我們亦已初步嘗試以 HKIF 架構產生之資料微調模型：微調後模型已能正確解碼輸入語句，惟回應多爲對語句內容的答覆而非還原結果（如輸入「rup wu0 j03h0 t gk6ak7?」應輸出「今天晚餐吃什麼?」，卻得到晚餐推薦），完整達成任務仍需更長時間的訓練，此爲後續工作方向。

致謝

本研究承教育部高等教育深耕計畫之特色領域研究中心計畫（114L9009、115L900901）支持，特此致謝。另感謝臺北市立第一女子高級中學同學填寫問卷，協助 School 資料集之語句蒐集。

References

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. OptimalThinkingBench: Evaluating over and underthinking in LLMs. *arXiv preprint arXiv:2508.13141*.

Apple. 2024. 在 Mac 上使用繁體注音輸入中文. Apple 支援（台灣）. Accessed: 2026-09-27.

ASUS. 2024. 華碩智慧輸入法. Accessed: 2026-09-27.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Zheng Chen and Kai-Fu Lee. 2000. A new statistical approach to Chinese Pinyin input. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 241–247, Hong Kong. Association for Computational Linguistics.

Chung-Pu Chiou, Hou-An Lin, and Chia-Ping Chen. 2022. Mandarin-English code-switching speech recognition system for specific domain. In *Proceedings of the 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022)*, pages 200–204, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Google. 2024. Google 輸入工具. Accessed: 2026-09-27.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yong Hu, Fandong Meng, and Jie Zhou. 2022. CSCD-IME: Correcting spelling errors generated by Pinyin IME. *arXiv preprint arXiv:2211.08788v1*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yanhua Long, Shuang Wei, Jie Lian, and Yijie Li. 2021. Pronunciation augmentation for Mandarin-English code-switching speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, 2021(1):34.

OpenAI. 2022. Introducing ChatGPT. Accessed: 2026-09-27.

OpenAI. 2024. Learning to reason with LLMs. Accessed: 2026-09-27.

OpenAI. 2025. Introducing gpt-oss. Accessed: 2026-09-27.

Gio Paik, Yongbeom Kim, Soungmin Lee, Sangmin Ahn, and Chan Woo Kim. 2026. HiKE: Hierarchical evaluation framework for Korean-English code-switching speech recognition. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 673–681, Rabat, Morocco. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sebastian Raschka. 2025. From GPT-2 to gpt-oss: Analyzing the architectural advances. Accessed: 2026-09-27.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

ToneOZ. 2024. 注音國語字典：曉聲通. Accessed: 2026-09-27.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37, Beijing, China. Association for Computational Linguistics.

Chao Wang and Stephanie Seneff. 2000. Improved tone recognition by normalizing for coarticulation and intonation effects. In *6th International Conference on Spoken Language Processing (ICSLP 2000)*, pages vol. 2, 83–86.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 882–890, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

上川日本語. 2025. 常用日語 1000 句合集 02（第 101–200 句）. YouTube. Accessed: 2026-09-27.

李倬安, 葉展維, and 張嘉惠. 2024. 基於深度學習的跨多輸入法編輯器整合系統. In *TAAI 2024* 台灣人工智慧年會論文集.

王冠侖. 2024. 修正未切換注音輸入法產生之字元. 2024 年臺灣國際科學展覽會優勝作品專輯.

A ATOM 提示詞全文

圖 5 爲 ATOM 提示詞之完整內容，包含任務說明、鍵位對照表、轉換步驟、輸出格式要求與輸出範例五個部分；實際使用時，{} 處代入待轉換之鍵位字串。

【任務說明】
使用者原本想用注音輸入法打中文，但忘記切換輸入法，
導致在英文輸入模式下按照注音鍵位習慣打字，
產生了一串英文字母和符號。
但使用者的輸入可能是中英夾雜的內容，
其中英文單字是刻意保留的，不需要轉換。您需要將下方「以下是待轉換字串」中的內容進行處理：
識別哪些是要轉成中文的部分，哪些是要保留的英文，
然後根據注音鍵盤配置將需要轉換的部分還原成繁體中文。
請參考下方的「英文鍵盤轉注音對應表」進行轉換。

【英文鍵盤轉注音對應表】
以下是標準英文鍵盤按鍵與注音符號的對應關係：
{{"a": "ㄇ", "b": "ㄖ", "c": "ㄏ", "d": "ㄎ", "e": "ㄍ", "f": "ㄑ", "g": "ㄕ", "h": "ㄘ", "i": "ㄛ", "j": "ㄨ", "k": "ㄜ", "l": "ㄠ", "m": "ㄩ", "n": "ㄙ", "o": "ㄟ", "p": "ㄣ", "q": "ㄆ", "r": "ㄐ", "s": "ㄋ", "t": "ㄔ", "u": "ㄧ", "v": "ㄒ", "w": "ㄊ", "x": "ㄌ", "y": "ㄗ", "z": "ㄈ", "1": "ㄅ", "2": "ㄉ", "3": "ˇ"(注音三聲), "4": "ˋ"(注音四聲), "5": "ㄓ", "6": "ˊ"(注音二聲), "7": "˙"(注音輕聲), "8": "ㄚ", "9": "ㄞ", "0": "ㄢ", "-": "ㄦ", ";": "ㄤ", ",": "ㄝ", ".": "ㄡ", "/": "ㄥ", "[": "[", "]": "]", "=": "=", " ": "(注音一聲)"}}

【轉換步驟】
請根據以下步驟執行：

(一) 請一步步的思考，
參照以下步驟來將拼字轉換成繁體中文字。請換成漢字，
而不是注音。有可能夾雜英文單字、英文姓名等，
請區分清楚。

1. 判別是否有英文單字，如有，
接下來不將單字轉換成中文。(請注意網路常見用語)
2. 將符號換成注音，拼成一句。
3. 將注音依可能語意換成繁體中文。
4. 若一個字沒有音調符號，視為注音一聲。

(二) 輸出格式要求：
- 請在思考過程結束後，在最後一行輸出最終答案
- 格式：Answer: [將此處替換為轉換後的繁體中文結果]
- 注意：請用實際轉換結果替換方括號內容，
不要保留方括號

【輸出範例】
如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎

以下是待轉換字串：
{}

【注意】請針對上方「以下是待轉換字串」的內容，根據「轉換步驟」一步步寫出思考過程，
最後根據輸出格式要求寫出答案。

圖 5: ATOM 提示詞完整內容。

B 韓文與日文輸入法之鍵位規則

韓文屬於音節文字，由初聲（子音）、中聲（母音）及終聲（收音）組合而成；透過標準兩發式鍵盤輸入時，一般字母直接對應單一鍵位，特殊結構之處理如表 9 所示。日文句子多含平假名、片假名以及漢字，平假名與片假名依照羅馬字輸入法的字根對應原則轉換成英文字串，特殊情況之處理如表 10 所示。本研究於延伸實驗之鍵位碼拆解與羅馬字轉假名階段，即依據此二表之規則進行處理。

C 資料建構與評估流程示意

資料集建構之整體流程包含：蒐集多樣化的原始語句、將原句轉換爲注音序列並映射至鍵位字串、以及處理聲調與選字規則以符合眞實輸入行爲。圖 6 爲單筆資料由原句 Y 生成鍵位字串 X 之流程，以「火車」爲例，先轉爲注音「ㄏㄨㄛˇㄔㄜ」，再映射爲鍵位輸出「cji3tk」。評估時，將待還原字串 X 輸入模型取得輸出 Y'，與標準答案 Y 計算 ROUGE-L 分數：若模型將「cji3tk」正確還原爲「火車」，該筆分數即爲 1.0，流程與實例如圖 7 所示。

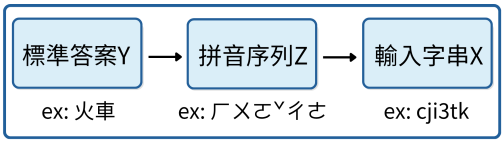

圖 6: 資料字串生成流程。

D 候選模型之評估與選擇

表 11 爲本研究於模型選用階段，對各候選大型語言模型與 API 服務之評估，評估面向包

特殊情況	字母組合	輸出方式	範例
雙子音（緊音）	ㄲ、ㄸ、ㅃ、ㅆ、ㅉ	Shift ＋子音鍵位	Q → ㄲ
複合母音	ㅘ、ㅝ、ㅚ、ㅟ 等	雙母音鍵位組合	h + k → ㅘ
複合收音（雙終聲）	ㄳ、ㄵ、ㄶ、ㄺ 等	雙子音鍵位組合	r + s → ㄳ

表 9: 韓文輸入法之特殊鍵位規則。

特殊情況	假名	輸出	範例／說明
促音	っ／ッ	下一字子音重複	きって → kitte
長音符	ー	-	コーヒー → ko-hi-
拗音	しゃ／じゅ	sha ／ ju	兩字合併爲一組
撥音	ん	n	後接母音時仍爲 n

表 10: 日文輸入法之特殊鍵位規則。

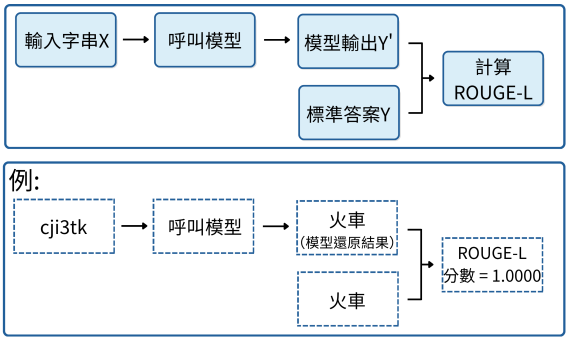

圖 7: ROUGE-L 評估流程與實例。

括：模型是否具備注音相關知識（了解注音）、是否提供免費 API 方案、以及免費方案是否無流量限制。gpt-oss-120b 爲三項條件均符合之選項，且其推理能力與整體效能領先多數開放權重模型，並可本地部署以保障個資安全，故爲本研究最終選用之模型。

大型語言模型	了解注音	免費 API 方案	免費無限制
Groq	×	✓	×
Mistral AI	×	✓	×
Claude	✓	×	–
gemini-2.5-flash	✓	✓	×
gpt-oss-120b	✓	✓	✓

表 11: 候選模型之評估與選擇。

E 基準與消融提示詞

E.1 基準提示詞全文

Direct Prompt 僅包含任務說明與輸出格式要求，全文如下：

【任務說明】
您需要將下方「以下是待轉換字串」中的內容轉成中英夾雜的句子。

輸出格式要求：
- 在最後一行輸出最終答案
- 格式：Answer: [將此處替換爲轉換後的繁體中文結果]
- 注意：請用實際轉換結果替換方括號內容，不要保留方括號

【輸出範例】
如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎

以下是待轉換字串：
{}

Zero-shot CoT 在 Direct Prompt 之後加入下列指令：

【注意】請針對上方「以下是待轉換字串」的內容，一步步思考，最後根據輸出格式要求寫出答案。

Few-shot Prompt 則在 Direct Prompt 的「以下是待轉換字串」之前插入【範例】區塊，提供三筆輸入與理想輸出之示範（表 12），讓模型模仿輸出格式與解碼方式。圖 8 爲 Zero-shot CoT 之運作示意；圖 9 爲兩種基準提示詞之結構示意。

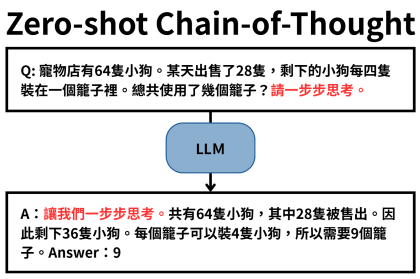

圖 8: Zero-shot CoT 示意圖（內容參考 Kojima et al., 2023，圖爲作者繪製）。

編號	輸入字串	理想輸出
1	`ji3rup wu0 vup fu/61j4cl3`	我今天心情不好
2	`Ashelycjo4x96au/6wu0 2k7rm4cjo4a87？`	Ashley 會來明天的聚會嗎？
3	`ji32k75;4cl4u3286l4ao3b4tokeng;4vu04xk7。`	我的帳號已達到每日 token 上限了。

表 12: Few-shot Prompt 之三筆示範例。

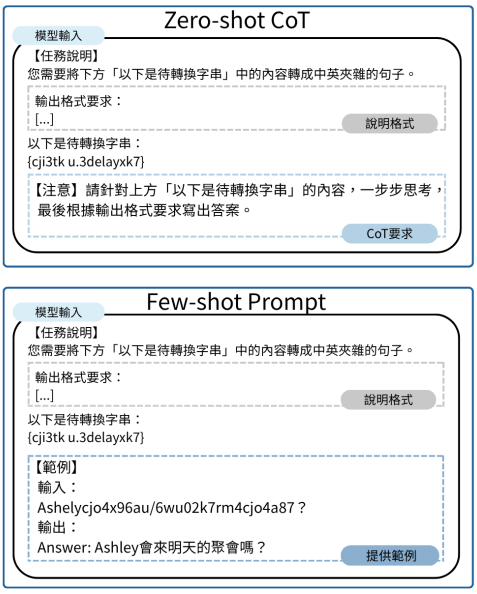

圖 9: Zero-shot CoT 與 Few-shot Prompt 基準提示詞之結構示意。

E.2 消融變體提示詞全文

各消融變體均以 ATOM 完整提示詞（附錄圖 5）爲基礎修改，以下爲各變體全文；其中【英文鍵盤轉注音對應表】之對應內容與 ATOM 相同，此處以「（同 ATOM）」表示，從略。

ATOM-NoTable（移除鍵位對照表）：

【任務說明】
使用者原本想用注音輸入法打中文，但忘記切換輸入法，導致在英文輸入模式下按照注音鍵位習慣打字，產生了一串英文字母和符號。但使用者的輸入可能是中英夾雜的內容，其中英文單字是刻意保留的，不需要轉換。您需要將下方「以下是待轉換字串」中的內容進行處理：識別哪些是要轉成中文的部分，哪些是要保留的英文，然後根據注音鍵盤配置將需要轉換的部分還原成繁體中文。

【轉換步驟】
請根據以下步驟執行：

（一）請一步步的思考，參照以下步驟來將拼字轉換成繁體中文字。請換成漢字，而不是注音。有可能夾雜英文單字、英文姓名等，請區分清楚。
1. 判別是否有英文單字，如有，接下來不將單字轉換成中文。（請注意網路常見用語）
2. 將符號換成注音，拼成一句。
3. 將注音依可能語意換成繁體中文。
4. 若一個字沒有音調符號，視爲注音一聲。

（二）輸出格式要求：
- 請在思考過程結束後，在最後一行輸出最終答案
- 格式：Answer: [將此處替換爲轉換後的繁體中文結果]
- 注意：請用實際轉換結果替換方括號內容，不要保留方括號

【輸出範例】
如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎

以下是待轉換字串：
{}

【注意】請針對上方「以下是待轉換字串」的內容，根據「轉換步驟」一步步寫出思考過程，最後根據輸出格式要求寫出答案。

ATOM-ShortTask（任務說明簡化爲一句話）：

【任務說明】
您需要將下方「以下是待轉換字串」中的內容轉成中英夾雜的句子。

【英文鍵盤轉注音對應表】
以下是標準英文鍵盤按鍵與注音符號的對應關係：（同 ATOM）

【轉換步驟】（同 ATOM-NoTable 之轉換步驟與輸出格式要求）

【輸出範例】
如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎

以下是待轉換字串：
{}

【注意】請針對上方「以下是待轉換字串」的內容，根據「轉換步驟」一步步寫出思考過程，最後根據輸出格式要求寫出答案。

ATOM-NoSteps（保留對照表、移除轉換步驟）：

【任務說明】
（同 ATOM-NoTable 之任務說明，末尾加上「請參考下方的『英文鍵盤轉注音對應表』進行轉換。」）

【英文鍵盤轉注音對應表】
以下是標準英文鍵盤按鍵與注音符號的對應關係：（同 ATOM）

輸出格式要求：
- 在最後一行輸出最終答案
- 格式：Answer: [將此處替換爲轉換後的繁體中文結果]
- 注意：請用實際轉換結果替換方括號內容，不要保留方括號

【輸出範例】

如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎
以下是待轉換字串：
{}
【注意】請針對上方「以下是待轉換字串」的內容，進行轉換，最後根據輸出格式要求寫出答案。

ATOM-NoTableNoSteps（移除對照表與轉換步驟）：

【任務說明】
（同 ATOM-NoTable 之任務說明）
輸出格式要求：
- 在最後一行輸出最終答案
- 格式：Answer: [將此處替換爲轉換後的繁體中文結果]
- 注意：請用實際轉換結果替換方括號內容，不要保留方括號
【輸出範例】
如果轉換結果是「你好嗎」，應輸出：
Answer: 你好嗎
以下是待轉換字串：
{}
