基於深度學習的跨多輸入法編輯器整合系統

李偉安 [1*]，葉展維 [2*]，張嘉惠 [3]
[1] 國立中央大學 資訊工程學系
E-mail: tommy920125@gmail.com
[2] 國立中央大學 資訊電機學院學士班
E-mail: jasonjasonyehyeh@gmail.com
[3] 國立中央大學 資訊工程學系
E-mail: chiahui@g.ncu.edu.tw
*這些作者對這項工作做出了相等的貢獻。

摘要

在日常鍵盤輸入中，拼寫錯誤和頻繁地切換輸入法，都會影響使用者的輸入流暢度及工作效率。因此我們運用深度學習開發了一個跨多輸入法整合系統，不僅支援多輸入法(包括注音、英文、倉頡和漢語拼音)，並透過「鍵盤輸入八鄰接 （8－adjacency）錯誤假定」之論點，解決輸入錯誤和同音字選字的問題。系統結構分為三個階段：輸入法分段、候選字轉換和候選字排序。在輸入法分段且無輸入錯誤的情況下，平均 Recall 可以達到 0.985，而個別輸入法分辨模型 F1-score 可以到 0.989。在候選字轉換且無錯誤的測試資料，注音、倉頡和漢語拼音的 Recall 表現也能達到 0.99。而在最後的 End-to-End 且無錯誤測試中，Accuracy 表現則為 0.613。

關鍵字：多語言輸入法、人機互動、深度學習

1. 緒論

無論是文章撰寫、商業信函往來或與朋友聯繫，鍵盤仍是當今人們主要的輸入方式。隨著全球化的趨勢，人們更常需要與來自不同語系的人進行交流。當鍵盤輸入參雜其他語言時，便需要切換輸入法，但使用者常因為思緒尚未切換完成，而以舊的輸入法輸入，產生錯誤的輸出。當使用者需要在不同輸入法間頻繁切換時，便容易產生錯誤。

而不同的作業系統在切換輸入法時，有著不同的操作方式，造成使用者熟悉上的負擔。如在 Windows 系統下，使用者須按 shift 鍵，來切換輸入法，而在 macOS 系統下，切換輸入法則是按下 Caps lock（中/英）。此外，輸入時也容易受到按鍵誤觸、同音字選字等問題而發生錯誤，而打斷思緒，影響工作的效率。

看似微不足道的輸入問題，卻嚴重影響使用者的輸入體驗。因此，本研究旨在建立一套新的輸入法系統，借助深度學習的技術，解決輸入法切換的問題、修正使用者的輸入錯誤並提升自動選字的準確性。第一階段使用深度學習訓練各個輸入法子模組，讓模型學習找尋輸入法的規則，並將混有不同輸入法的序列進行分段。第二階段則運用 Levenshtein distance algorithm 和 Trie 字典樹，將分段後的輸入轉換為真正的候選字。最後則考慮同音字、詞彙頻率及詞組關係，將候選字進行最佳排序。實驗結果顯示，在第一階段輸入法分段且無輸入錯誤的情況下，平均 Recall 可以達到 0.985，而個別輸入法分辨模型 F1-score 可以到 0.989。在候選字轉換且無錯誤的測試資料，注音、倉頡和漢語拼音的 Recall 表現也能達到 0.99。而在最後的 End-to-End 且無錯誤測試中，Accuracy 表現則為 0.613。

2. 相關研究

由於輸入法整合是一個相對較新且尚未被充分探討的議題，相關的研究相對匱乏，尚無研究涵蓋輸入法切換、整合及適應等問題，使得此研究題目更具突破性和挑戰性。

2.1. 華碩智慧輸入法

目前的市場中，華碩電腦(ASUS)將注音(Bopomofo)及英文(English)輸入法整合為雙語言輸入法，稱為華碩智慧輸入法。幫助經常在兩種輸入法間切換的使用者，提供了更好的體驗。然而，實際使用華碩智慧輸入法後，發現其並未針對誤按按鍵的情況進行容錯，而是直接將其判斷為錯誤的鍵盤輸入，只要不符合注音的輸入規則即視為是英文輸入。此外，華碩智慧輸入法僅支援兩種語言的轉換，假若使用者須同時使用兩種以上的輸入法，如倉頡(Cangjie)、漢語拼音(Pinyin)或日文(Japanese)時，原有的系統就要重新設計，不容易支援新的輸入法。

2.2. 使用者鍵盤輸入模型

在使用者的日常操作中，由於疏忽或思緒斷裂，可能會導致的鍵盤輸入錯誤。舉例而言，為了解鍵盤輸入錯誤對翻譯的影響，Belinkov & Bisk (2017)[1]介紹了一種模擬字元輸入錯誤的方法，通過刪除、替換或插入特定字元來測試語言翻譯模型對這些微小變化的表現。顯示模型會因為沒有訓練到某些特定類型的輸入錯誤，導致模型在遇到這類型的錯誤提示詞時，表現顯著下滑。

Belinkov & Bisk (2017)[1]的方法雖能模擬部分輸入錯誤情境，但無法涵蓋所有日常輸入的情況。為了更符合真實情況，Si 等人 (2023)[8]的研究提出了更具代表性的方法。通過向眾多標記者蒐集打字輸入結果，蒐集各類型的按鍵錯誤，如文字順序倒置和模糊縮寫等情境，確保資料集的多樣性，從而更全面地評估語言模型對使用者輸入錯誤的處理效能。不過，Si 等人 (2023)[8]所收集的輸入錯誤仍具有侷限性，其所邀請的標註者都來自亞洲，且皆使用拼音輸入法。受個人習慣、文化背景和心理狀態等因素影響，無法通用到其他輸入法上。因此，我們認為即便模擬使用者的輸入錯誤，存在非真實性，但若能創造出更客觀、更貼近現實的錯誤假定論點，便能夠更好的模擬使用者的輸入錯誤，並通用至不同的輸入法。

2.3. 鍵盤輸入距離和代表涵義

在 Kaluarachchi 等人 (2023)[4]的研究中，提到如何計算鍵盤上按鍵之間的距離。他們將向四個鄰近方向的按鍵距離設為 1。如 A 和 S 之間的距離為 1、而斜角 T 和 H 之間的距離為 2，並指出最長的距離是從 Q 到 P 為 9。然而，該研究僅考慮英文的輸入法，並未納入數字或周圍特殊符號的按鍵。當遇到，如注音輸入法，將數字和特殊符號按鍵也代表著某些意義的輸入法時，可能會導致計算結果的不準確或不完整。因此在錯誤規則的設計上，必須考慮到鍵盤上所有的按鍵的元素，以確保系統的全面性和完整性。

2.4. 跨多語言輸入法中句子片段切割

為了處理使用者的按鍵輸入，對各種輸入法按鍵輸入的標記顯得格外重要。我們將會探討跨輸入法時句子片段切割的標記方法，包括句子片段（SentencePiece）和子字元（Sub-Character）標記，使模型在資料集上更細緻地學習使用者的按鍵輸入與輸出預測。這些標記方法能提供更精確的分詞和更全面的語境理解，提升模型對使用者輸入的解釋和預測能力。

句子片段（SentencePiece）Kudo & Richardson (2018)[5] 是基於子詞（SubWord）Sennrich et al. (2015)[6] 的文本分割方法，通過將單詞分割成子詞來處理，能更彈性地處理不同語言和文本類型的詞彙，同時又能保留單詞的邊界資訊。而在 Si 等人 (2023)[7]的研究中，他們提到了子字元（Sub-Character）的字元標示方法，意思是指將單個中文字轉為漢語拼音的按鍵輸入的組合作為詞元（Token），當出現打錯同音字的情況時，語言模型看到的仍是同樣的序列，進一步提高語言模型對含有錯誤時中文任務的表現。

這兩種標記方法在處理使用者的按鍵輸入時，可以增加模型的理解能力，使模型能更靈活地學習和理解使用者使用不同輸入法的按鍵組合，提高模型性能和準確性。

3. 研究方法

本節將詳細介紹我們所使用的資料集及系統各個階段的設計實作細節。

3.1. QWERTY 鍵盤按鍵標示

首先，我們將各種輸入法的按鍵，利用 QWERTY 鍵盤的對應進行統一的表示。亦即使用英文輸入法的 QWERTY 鍵盤，來輸入其他輸入法的時所產生的序列。包含小寫英文字母 a~z、數字 0~9 以及大寫字母 A~Z(表示按著 Shift 鍵輸入)和特殊符號（'!', '@', '['）等等，表 1 為使用三種不同中文輸入法打出「大城市」，所需的按鍵序列。

原文	**大城市**
注音輸入法	284t/6g4
倉頡輸入法	k gihs ylb
漢語拼音輸入法	da cheng shi

表 1. 在各輸入法欲輸入範例句時之按鍵序列

3.2. 鍵盤輸入八鄰接錯誤假定

了更符合我們的任務需求並更全面地理解鍵盤布局的特性，我們提出了「鍵盤輸入八鄰接（8-adjacency）錯誤假定」之論點，如表 2 所示。我們將鍵盤上的按鍵分為多個叢集(cluster)，每個叢集代表著一組彼此相鄰八個的按鍵。基於此劃分我們可以建立一套新的輸入錯誤規則，用於模擬、檢測和修正使用者的輸入錯誤。透過比較使用者實際的輸入按鍵和預期的按鍵叢集，將能夠及時發現錯誤，並提供反向的錯誤修正。

T	Y	U
G	**H**	J
B	N	M

表 2. 以鍵盤中字母 H 為中央位置的叢集

3.3. 資料集

由於過去對於輸入法及按鍵錯誤的研究不足，缺乏現成的資料集可供使用。因此，本研究使用了訓練大型語言模型時的資料集 CC100[2] 和 Multi-News[3]作為中文和英文原始資料，大約 71 MB 的文字資料，並自行開發按鍵轉換器，將純文字資料轉換成各種輸入法的按鍵表示。採用如此大量的文字資料是考慮到輸入法應用的場景廣泛，須盡可能涵蓋各種詞彙，以減少模型的偏見與誤差。

在這兩份資料集中，原先參雜許多標點符號、特殊符號或非該語言的字符。我們首先進行資料清洗，確保訓練資料僅涵蓋特定語言後，再利用自行撰寫的候選字轉換器，將純文字轉換為對應的輸入法序列，分別為注音(Bopomofo)、英文(English)、倉頡(Cangjie)和漢語拼音(Pinyin)。接著再透過隨機(插入、刪除、交換或替代按鍵)或依八鄰接錯誤假定，在按鍵序列中加入錯誤，以模擬使用者的輸入錯誤，如圖 1。

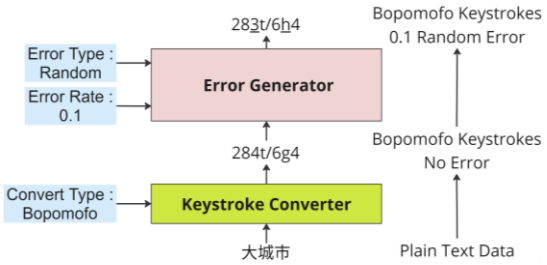

圖 1. 訓練資料生成流程

接著，針對不同階段的資料格式進行準備。如針對系統第一階段中，輸入法分辨器的訓練，將屬於該輸入的按鍵序列標記為正，而不屬於該輸入法的序列標記為負。而針對輸入法分段的測試資料，則隨機將兩種不同輸入法的按鍵序列連接在一起，模擬混合輸入的情況。

3.4. 系統架構

為了構建一個能夠支援多種語言的輸入法系統，且滿足輸入法的按鍵輸入映射的特性，系統總共有三個階段(phase)，分別為輸入法分段、候選字轉換和候選字排序圖 2，來處理從使用者按鍵輸入到候選字顯示的整個過程。這三階段流程同時也考量了系統的可管理性和擴充性，以方便未來支援新的輸入法。

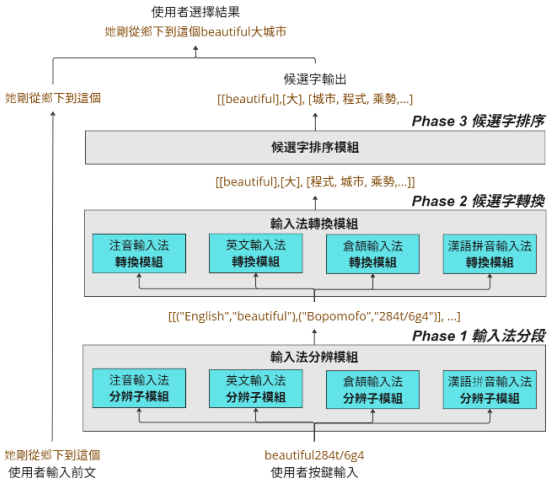

圖 2. 系統架構圖

3.5. Phase 1 – 輸入法分段

此階段希望從使用者打出參雜多種輸入法的序列中，找到輸入法的分段點。我們針對各別輸入法，訓練了 4 個獨立的深度學習分辨模型，專注分辨於該序列是否為特定輸入法。

從生成的按鍵資料集中，隨機抽取了 480K 筆作為訓練資料，以四種輸入法為基準，各自占有總訓練資料筆數的 25%，並根據欲訓練的模型，將輸入法序列標記為 1 或 0。假設當前為訓練注音輸入法模型，則注音的序列標籤為 1，其餘序列（倉頡、拼音、英文）皆為 0，以此類推。此外，為了讓模型在含有錯誤時也具有通用性，也在 25%的訓練資料中加入 10%錯誤。所謂 10%錯誤，即單筆測試資料中，每一個鍵盤輸入按鍵都有 1/10 的機率會發生替換、刪除或交換（各 1/3 的機率）。

接著我們根據先前 QWERTY 鍵盤按鍵標示的按鍵並增加<PAD>、<SOS>、<EOS>和<UNK>等特殊詞元作為特徵，共 100 種，並以 One-Hot Encoding 表示。由於每筆訓練資料的按鍵從長度不等，我們利用<PAD>統一將其填充到長度 30，以便訓練。個別輸入法分辨的深度學習模型，使用四層全連接的神經網路模型，維度從 3000 維逐漸降至 2 維，即為分類的類別數。層與層之間都有 Rectified Linear Unit (ReLU) 激勵函數和 50%的正則化避免過擬合。此設計是因為輸入法都具有明確可循的輸入規則，僅需少量模型參數即可滿足要求，與此同時符合系統輕量化的設計原則。

接著將四種輸入法的分辨模型運用在輸入法分段中，當使用者輸入含有多於一種輸入法時，分段演算法會從索引位置 1 開始持續到字串最後，對分段點前後的字串進行判斷，若前後序列判斷出的輸入法不相同，則將其視為一組可能的分段點，輸入到下一階段。

3.6. Phase 2 – 候選字轉換

此階段負責將鍵盤按鍵輸入轉換為真正的候選字，我們針對個別輸入法字典內的字依輸入時的按鍵，建構字典樹（Trie）圖 3，並以深度優先搜尋結合 Levenshtein distance (LD)演算法，計算輸入和字典內詞彙的按鍵距離，以找到距離最近的候選字群。即便使用者不小心輸入 1~2 個錯誤，系統仍能自動找到最可能的候選字，對輸入錯誤進行容錯。

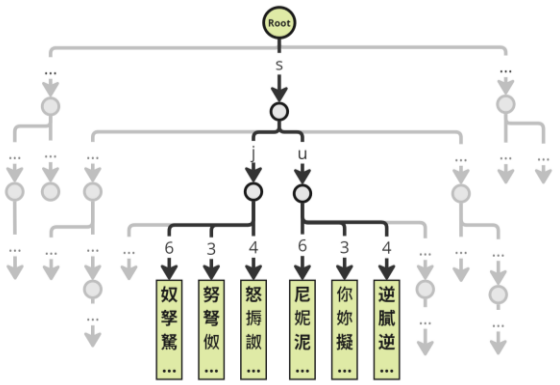

圖 3. 注音輸入法之字典樹

在原先 LD 演算法中，兩個字串間距離定義為將一個字串 A 更改為另一個字串 B 所需的最小編輯（插入、刪除或替換）次數。如字串 su3rl4 和 su3cl3，兩者的距離為 2。距離越小，代表輸入與詞典內的候選字越相似。我們針對 LD 演算法再進行改良，若在相鄰位置上發生按鍵互換時（cadr/card），距離設為 1，而非原先 LD 演算法的 2，將其視為使用者輸入順序上的小錯誤，不將其與其他距離為 2 錯誤的列為同等關係。

3.7. Phase 3 - 候選字排序

前階段已經轉換成可直接顯示在使用者介面中的候選字，此階段的目標，是在將候選字排序，減少使用者選字的時間。多個候選字間，若存在的

詞組關係則優先排序，讓使用者想要的字詞出現在最上方。若存在同音詞組的情況，以注音輸入法(Bopomofo)為例（如表 3），則直接根據詞組出現頻率統計來做排序。

同音詞鍵盤輸入	同音詞-1	同音詞-2
au6xj4 (ㄇㄧˊㄌㄨˋ)	迷路	麋鹿
aj42u4 (ㄇㄨˋ ㄉㄧˋ)	目的	墓地
u.6m6 (ㄧㄡˊ ㄩˊ)	由於	魷魚

表 3. 注音輸入法(Bopomofo)同音詞組範例

4. 實驗

為了驗證本研究所提出的系統能有效解決當前輸入法面臨的諸多問題，我們採取一系列系統化的驗證方法。首先，針對系統設計的每階段（phase）進行詳細的性能分析，確保各階段的任務目標是具有意義的，並採用多種模型進行比較，以驗證不同方法的優劣。最後，我們模擬使用者的真實使用情境，進行 End-to-End 的性能分析，以全面評估系統的實際應用效果。

4.1. Phase 1 – 輸入法分段

針對以下三種不同的模型設計方式進行比較，分別為運用 Term Frequency-Inverse Document Frequency（TF-IDF）Vectorizer 作為 Encode 方式在 Support Vector Machine（SVM）以及 Deep Neural Network（DNN）訓練模型下的表現，和直接採取 One-Hot Encoding 的 DNN 模型下的表現。

我們以 480K 筆資料進行訓練，針對 2 種不同的錯誤比例，各 5K 筆進行測試，以評估不同模型在有錯誤的資料下表現是否良好。分別為無輸入錯誤（如圖 4）、0.1 錯誤（如圖 5）和整體巨集 F1-Score（如圖 6）。在三種模型的表現中，可以看到採取 One-Hot Encoding 的 DNN 訓練模型下的表現較為優秀，甚至在 0.1 Error 的測試資料中，遠遠勝過其他模型的表現。此外，也可以發現判別拼音和英文輸入法會是相對較難的任務。

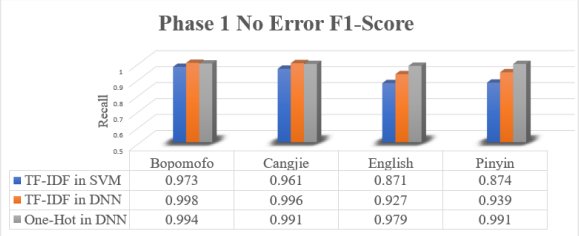

圖 4. Phase 1 No Error F1-Score Performance

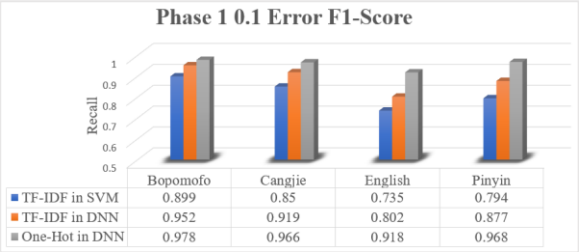

圖 5. Phase 1 0.1 Error F1-Score Performance

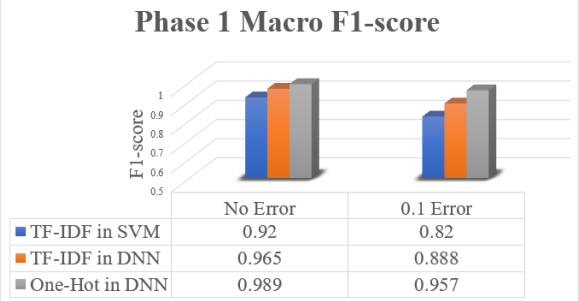

圖 6. Phase 1 Macro F1-Score Performance

接著我們準備了 3 種不同的錯誤比例，各 2K 筆混雜了任意兩種輸入法的測試資料。尋找輸入法的分段點，並標記前後兩種輸入法，採用 3 種衡量標準。假設一組測試資料中，預測出 15 種不同的分段組合，而其涵蓋正確的分段組合。則此測資的 Precision 則為 1/15，Recall 則為 1，針對 2K 筆的測試資料的 Precision、Recall 取平均後再計算 F1-Score，表現如表 4。由於目標是盡可能抓到正確的分段點，因此 Recall 的表現是此階段較在意的指標，其餘的錯誤組合則依靠後續候選字轉換，逐步消除。

	No Error	**0.05 Error**	**0.1 Error**
Recall	0.985	0.966	0.951
Precision	0.074	0.073	0.072
F1-Score	0.137	0.135	0.134

表 4. Phase 1 Separator Performance

4.2. Phase 2 – 候選字轉換

此階段為衡量候選字轉換演算法是否能將鍵盤輸入轉換為正確的候選字。我們準備了 3 種不同的錯誤比例，各 5K 筆的測試資料，表現如圖 7。可以看到四種輸入法的 Recall 表現，即使在 0.1 錯誤的測試資料中，表現依舊可以高於 0.74。

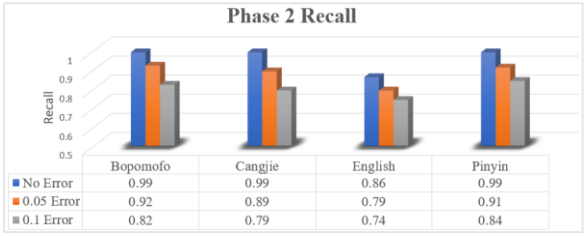

圖 7. Phase 2 Recall Performance

4.3. Phase 3 – 候選字排序

在此階段主要希望能幫助使用者根據優先性排序候選字，並組合詞組。但是因為同音詞組的情況較為少見，如表 3。且候選字排序對於實際使用的打字速度提升難以直接測定。因此此階段的表現，直接和 End-to-End 一起考慮。

4.4. End-to-End 表現

系統最終是希望使用者能不必切換輸入法，並在列出候選字選項時，有著使用者想要的單字或詞組。我們準備了 2K 筆有著兩種輸入法的無錯誤測試資料，從第一階段一直執行到第三階段。將原先混雜兩種輸入法的序列，轉換出多個候選字。而衡量標準則為，只要有一個字沒有成功出現在最終的候選字中，即為錯誤。假設測試輸入為「weather 5p cl3 」但系統將其轉換為[[“wealth”], [“真”, …], [“好”, …]]，即使僅一組候選字未轉換成功，仍依舊當作錯誤。此衡量標準設定較為嚴格，最終 Accuracy 表現為 0.613，未來仍舊有改進空間。

5. 結論

本研究通過輸入法分段、候選字轉換和候選字排序三個階段，系統能準確識別輸入法的轉換並提供錯誤修正，改善使用者的打字體驗。在輸入法分段且無輸入錯誤的情況下，平均 Recall 可以達到 0.985，而個別輸入法分辨模型 F1-score 可以達到 0.989。在候選字轉換且無錯誤的測試資料，注音、倉頡和漢語拼音的 Recall 表現也能達到 0.99。而在最後的 End-to-End 且無錯誤測試中，Accuracy 表現則為 0.613。我們提出的「鍵盤輸入八鄰接 (8 – adjacency) 錯誤假定」理論，為往後的研究提供了新的參考。未來將著重於利用不同技術及演算法，進一步提升系統性能。我們也期望能將此系統整合至作業系統的原生輸入法框架中，並基於用戶反饋持續改進，推動多輸入法整合系統的發展。

6. 誌謝

本研究承蒙國家科學及技術委員會 - 大專學生研究計畫補助，特此致謝。研究計畫編號 113-2813-C-008-024-E。

7. 參考文獻

[1] Belinkov, Y., & Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. arXiv preprint arXiv:1711.02173.

[2] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

[3] EMNLP 2011 Sixth Workshop on Statistical Machine Translation： https://statmt.org/wmt11/ translation-task.html

[4] Kaluarachchi, N., Kandanaarachchi, S., Moore, K., & Arakala, A. (2023). Deft: A new distance-based feature set for keystroke dynamics. In 2023 international conference of the biometrics special interest group (biosig) (pp. 1–6).

[5] Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language indepen-dent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

[6] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[7] Si, C., Zhang, Z., Chen, Y., Qi, F., Wang, X., Liu, Z., ... Sun, M. (2023). Sub-character tokenization for chinese pretrained language models. Transactions of the Association for Computational Linguistics, 11, 469–487.

[8] Si, C., Zhang, Z., Chen, Y., Wang, X., Liu, Z., & Sun, M. (2023). Readin: A chi- nese multi-task benchmark with realistic and diverse input noises. arXiv preprint arXiv:2302.07324.