

# 具混亂資料下對 Google NotebookLM 影響及優化研究

## 一、研究目的

在資料中存在噪音或不一致性時(如：平常會議紀錄或新聞寫手的寫作風格不同等狀況)，無法確認 Google NotebookLM 的效能是否會受到影響，這些影響可能體現在生成內容的準確度、語境理解、以及結果的可靠性上。

因此研究的核心目的是了解影響程度，並提出優化方式。

## 二、研究步驟

本章內容將說明本研究的研究步驟：

### 1. 定義混亂資料

混亂資料於本研究的定義，須具備「錯誤標註的資料」、「無法歸類的或模糊的資訊」、「資料不一致或語法錯誤」、「無關或冗餘的資料」的條件，而在本研究中，使用 ChatGpt，以下方的內容為原則協助加工原始資料：

- **混亂資料的生成原則：**
  - (1) 語法錯誤引入：打亂句子結構、不正確的時態或語態、重複句子或語詞
  - (2) 模糊或含糊不清的表達：曖昧的代詞使用、無清晰主題
  - (3) 無關或無效資訊的插入：無關的插入語、噪音字符或符號
  - (4) 隨機拼湊不同領域的內容：領域混搭、隨機跳躍
  - (5) 錯誤標點與排版：標點符號錯誤使用、段落和空格錯誤

### 2. 資料來源

新聞內容較易取得外，寫手們各自風格不同也較符合平時能取得資料的一般性，將資料分成乾淨資料與混亂資料兩類，其中內容如下：

- **乾淨資料集(對照組)：**選擇兩組新聞資料作為原始資料集
- **混亂資料集(實驗組)：**對乾淨資料集使用 ChatGPT 進行加工混亂噪音

### 3. Google NotebookLM 影響評估

使用 Google NotebookLM 進行兩組資料集的生成資料後，於本研究使用自然語言處理常用的自動化評估指標(ROUGE 分數)來評估 Google NotebookLM 在原始資料與混亂資料下的表現。

## 使用 ROUGE 理由：

自然語言處理中較廣泛使用的指標為 BLEU 和 ROUGE，但該指標主要功用是使用在翻譯這類較講究精準的應用上，實際測試後確實由於用詞不相同經常趨近於 0 分，因此選擇使用時常應用於判斷摘要語意完整的 ROUGE 指標，其中包含單詞、雙詞、語句結構的評估，根據其數值可以從中獲得一些真知灼見。

## 4. 優化方法設計

本研究將會以前述方式評估優化後的結果，為保留一般性不考慮太艱深的程式操作清洗或更多工具做加工處理，以簡單為導向，本研究使用 ChatGPT 進行文句通順與補字。

## 三、研究方法

本研究使用 ROUGE 指標來評估 Google NotebookLM 針對不同資料來源後生成的結果，資料來源則附於作業格式的資料來源中，以下附上實驗程式：

```
# from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
from rouge_score import rouge_scorer

def read_text(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()
    return text

generated_text = read_text(r'C:\Users\user\OneDrive\Desktop\notebookLM作業\混亂結果.txt')
reference_text = read_text(r'C:\Users\user\OneDrive\Desktop\notebookLM作業\乾淨結果.txt')

generated_text_tokenized = generated_text.split()
reference_text_tokenized = reference_text.split()

# weights = (0.25, 0.25, 0.25, 0.25)
# smoothing_function = SmoothingFunction().method1

# bleu_score = sentence_bleu(reference_text_tokenized, generated_text_tokenized, weights=weights, smoothing_function=smoothing_function)
# print(f"BLEU Score: {bleu_score}")

scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)

scores = scorer.score(reference_text, generated_text)
print(f"ROUGE-1: {scores['rouge1']}")
print(f"ROUGE-2: {scores['rouge2']}")
print(f"ROUGE-L: {scores['rougeL']}")
```

```
PS C:\Users\user> & C:/Users/user/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/user/OneDrive/Desktop/notebookLM作業/from nltk.translate.py"
ROUGE-1: Score(precision=0.7115384615384616, recall=0.8043478260869565, fmeasure=0.7551020408163265)
ROUGE-2: Score(precision=0.5490196078431373, recall=0.6222222222222222, fmeasure=0.5833333333333334)
ROUGE-L: Score(precision=0.6538461538461539, recall=0.7391304347826086, fmeasure=0.693877551020408)
```

	Precision	recall	f-measure
ROUGE-1	0.7115	0.8043	0.7551
ROUGE-2	0.5490	0.6222	0.5833
ROUGE-L	0.6538	0.7391	0.6939

表 1.混亂資料與乾淨資料的 ROUGE 結果

參考表 1 確認比對混亂資料與乾淨資料的產出結果評估如下：

1. ROUGE-1 的準確度顯示了在單詞上的匹配程度高，說明了混亂文本與乾淨文本相似，沒有使用到太多無用或無關的單詞；召回率也高，代表混亂文本其實有生成大多數乾淨文

本的單詞，只是有些冗余、不準確的單詞。

2. ROUGE-2 的準確度顯示了雙詞上的匹配程度數值顯示準確度和召回率都相對較低，因為混亂資料中語法錯誤、冗語或不合邏輯的連接詞，導致雙詞之間的匹配度較低。
3. ROUGE-L 是要尋找最長共同子字串的匹配程度，這個數值結果顯示混亂文本和乾淨文本在語法結構上有尚可的相似度。

為了產出更好的生成結果，將資料集置入 ChatGPT 請他協助補齊與使文句通順後，產出的結果如下圖：

```
PS C:\Users\user> & C:/Users/user/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/user/OneDrive/Desktop/notebookLM作業/from nltk.translate.py"
ROUGE-1: Score(precision=0.8648648648648649, recall=0.6956521739130435, fmeasure=0.7710843373493975)
ROUGE-2: Score(precision=0.5555555555555556, recall=0.4444444444444444, fmeasure=0.49382716049382713)
ROUGE-L: Score(precision=0.8378378378378378, recall=0.6739130434782609, fmeasure=0.746987951807229)
```

	Precision	recall	f-measure
ROUGE-1	0.8649	0.6957	0.7711
ROUGE-2	0.5556	0.4444	0.4938
ROUGE-L	0.8378	0.6739	0.7470

表 2. 優化資料與乾淨資料的 ROUGE 結果

參考表 2 確認比對優化資料與乾淨資料的產出結果評估為：

1. 與混亂文本相比，優化後生成文本在單詞的精確度上有顯著提升，從 0.7115 提升到 0.8649，優化文本中的單詞與參考文本更一致，且冗餘或不準確的單詞減少；雖然召回率略微下降，但精確度提升使得 F-measure 基本持平，顯示文本的品質有提高。
2. 雙詞的精確度輕微提高（由 0.5490 提升至 0.5556），但是召回率有所下降。這表明優化後的文本在某些語句結構的雙詞匹配上有所改善，但同時也可能造成某些詞組的遺漏。這裡的 F-measure 稍有下降，顯示出雙詞層面表現與混亂資料持平。
3. 語法結構方面的匹配提升顯著，精確度也有顯著提高，從 0.6538 增加至 0.8378，顯示出優化後的文本在語法結構層面與參考文本的相似度大幅提高。這意味著經過優化的文本在語法流暢性和結構合理性方面有了顯著改善。

## 四、結論

混亂資料顯著影響 Google NotebookLM 的生成結果。在混亂資料下，Google NotebookLM 產生的文本雖然在單詞匹配程度上表現不錯，顯示未使用過多無關或無效的單詞，但由於混亂資料中存在語法錯誤、冗語或不合邏輯的連接詞，導致雙詞匹配度顯著下降。儘管語法結構保持一定程度的相似度，但整體而言，混亂資料影響了 Google NotebookLM 生成內容的準確性與語境理解。

為解決此問題，本研究嘗試利用 ChatGPT 優化混亂資料，將混亂文本變得更加通順與補足缺字。實驗結果顯示，經過 ChatGPT 優化後的單詞精確度顯著提升，代表優化後的文本與乾淨文本更為一致，冗餘或不準確的單詞大幅減少。語法結構方面的匹配度亦有顯著改善，顯示優化後的文本在語法流暢性與結構合理性上取得了顯著進展。

然而，值得注意的是，優化後的文本在單詞精確度和語法結構上有明顯提升，雙詞的召回率仍有所下降，顯示當初在加工混亂或優化語句結構的同時，某些詞組可能被遺漏掉了。

不過整體而言，利用 ChatGPT 等工具對要放入 Google NotebookLM 的混亂資料進行初步處理能有效提升生成結果。

## 未來應用或研究方向

本研究僅使新聞資料做為資料來源，未來應用在不同場景討論(如：論壇資料、爬蟲資料、會議資料、論文研究等等)，無論是使用不同的語言、檢測工具、清洗資料工具，探討其更優化的方式，於未來的使用上意義重大。