

基於 LLaMA 3.1 和 BART 大語言模型的長文本摘要多層次演算法
Multilevel Algorithm for Long Text Summarization Based on LLaMA 3.1
and BART Large Language Models

蘇珮君

January 09, 2025

指導教授：賴振耀 博士

長文本的介紹

定義

- 一般定義：數千字以上、結構化或非結構化。
- 自然語言處理 (NLP) 領域：超過輸入限制。

摘要的問題與挑戰

- 模型的輸入限制。
- 注意力機制分散與位置偏差。

相關文獻支持

- Liu 等人 (2017)：注意力機制分散問題。
- Jiang 等人 (2023)：無法遵守壓縮比限制問題。
- Pan 和 Wu 等人 (2024)：設計提示詞以縮短文本長度。

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

- 簡介與目的

- 解決長文本摘要因注意力機制分散導致的位置偏差問題。
- 資源有限情況下，同時提升速度與準確率。

- ROUGE 評估指標介紹

- ① ROUGE-1：衡量摘要的內容覆蓋率。
- ② ROUGE-2：評估語言流暢性與結構一致性。
- ③ ROUGE-L：評估語義結構與內容順序的連貫性。

主要分析項目：壓縮率、信息保留度、語言流暢性。

- CNN/DailyMail 數據集

- 針對新聞文章進行摘要設計的資料集。
- 測試集 (Test dataset): 11490 筆資料。
- 資料集內容: 文章 (article)、摘要 (highlights)、唯一識別碼 (id)。

```
{  
  id : 062f78c2922d4050190dbba10f5d65eeff25e1ed  
  article : Bayern Munich have an interest in Chelsea defender Branislav Ivanovic ...  
  highlights : Branislav Ivanovic's contract at Chelsea expires ...  
}
```

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

固定系統 vs. 動態系統

- Ollama 平台：整合多種開源模型，簡化部署，支持本地端運行。
- LLaMa 3.1 API 的使用，主要有兩種方法：
 - 固定系統：使用 `ollama.create`
 - 動態系統：使用 `ollama.generate`
- 兩種方法在靈活性上差異的比較：
 - 固定系統：適用於**固定需求**，能預設系統提示詞（system prompt）。
 - 動態系統：**更加靈活**，接受使用者提示詞（user prompt）。

設置方式	<code>ollama.create</code> (固定系統)	<code>ollama.generate</code> (動態系統)
系統提示詞 (system prompt)	O	X
使用者提示詞 (user prompt)	O	O

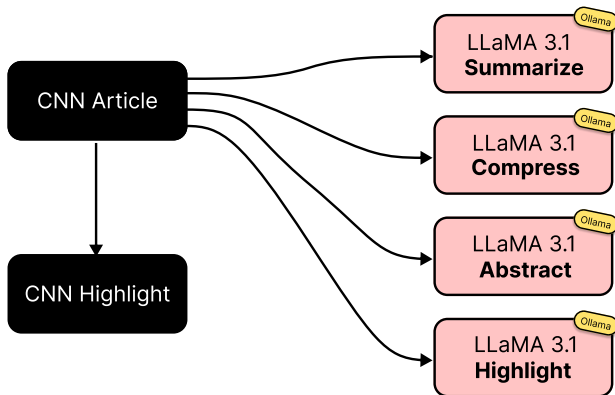
Table: 固定系統 vs. 動態系統

固定系統系統提示詞設計

- 減少文本詞彙量，保持語義和結構不變。
- 系統提示詞包含以下要求，基於 Pan 和 Wu 等人¹的研究：
 - ① You can **ONLY** remove unimportant words.
 - ② Do not reorder the original words.
 - ③ Do not change the original words.
 - ④ Do not use abbreviations or emojis.
 - ⑤ Do not add new words or symbols.

¹Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LlmLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression, 2024.

流程圖



固定系統使用者提示詞設計

- 設計四個任務指令，分別為：
 - **compress**: 壓縮文本，刪除冗餘內容。
 - **highlight**: 保留文本中關鍵資訊。
 - **summarize**: 提供文本的簡單摘要。
 - **abstract**: 生成抽象層次的總結。
- 以不改變文本核心語意的前提下進行處理。
- 目的為測試這些指令在文本壓縮的效果。




壓縮比測試結果

- 測試指標：
 - 平均 Token Ratio：壓縮後的 Token 數/原始文本數。
 - 平均 Word Ratio：壓縮後的 Word 數/原始文本數。
- 測試結果如下表所示：

Task	Test Item	Avg. Token Ratio	Avg. Word Ratio
1	Compress	0.7124	0.6956
2	Highlight	0.8111	0.8068
3	Summarize	0.2196	0.2199
4	Abstract	0.6979	0.6821

Table: 四個任務的平均壓縮比

- 動態提示詞壓縮比限制：模型未完全遵守（與 Jiang 等人²研究一致）。

²Jiang, .etc. LLMingua: Compressing prompts for accelerated inference of large language models, 2023.   

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

多層次摘要框架

- 目的：通過分層摘要濃縮訊息來減少位置偏差問題。
- 主要步驟包括：
 - ① 文本段落的拆分；
 - ② 摘要生成；
 - ③ 摘要合併。
- 最終摘要長度控制：
 - 根據 CNN/DailyMail 資料集中的 highlights 來設置目標長度範圍。
 - 設定最終摘要的目標長度在 highlights 長度的 90% 至 120% 之間。

模型與分割工具

- 模型選擇：
 - 選用 BART 作為基本摘要模型。
 - 使用預訓練摘要模型 bart-large-cnn。
- 段落分割工具：
 - 使用 spaCy 進行段落分割。
 - 基於句子結束符號判斷邊界。
- Token 長度控制：
 - 限制最大 Token 長度為 512，對應 BART 模型的上限一半。
 - 縮短輸入文本的長度，降低位置偏差的影響。

摘要長度控制與評估方式

- 段落摘要長度設定：
 - 長度設定在 230 至 256 Tokens 之間。
 - 230 Tokens 為目標長度的 90%，避免過度壓縮。
 - 256 Tokens 為目標長度的 100%，確保信息的完整性。

layer	
original	$[T_1^{(1)}], [T_2^{(1)}], [T_3^{(1)}], [T_4^{(1)}], \dots, [T_{m_1}^{(1)}]$
summary 1	$[S_1^{(1)}], [S_2^{(1)}], [S_3^{(1)}], [S_4^{(1)}], \dots, [S_{m_1}^{(1)}]$
combine 1	$[C_1^{(1)}], [C_2^{(1)}], [C_3^{(1)}], \dots, [C_{m_2}^{(1)}]$
summary 2	$[S_1^{(2)}], [S_2^{(2)}], [S_3^{(2)}], \dots, [S_{m_2}^{(2)}]$
combine 2	$[C_1^{(2)}], [C_2^{(2)}], \dots, [C_{m_3}^{(2)}]$
\vdots	\vdots
summary n	$[S_1^{(n)}], [S_2^{(n)}]$
combine n	$[C_1^{(n)}]$

Table: 多層次摘要框架流程圖

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

單一模型摘要結果

- 實驗模型：
 - 使用模型 LLaMA 3.1 和 BART 分別進行文本摘要。
 - LLaMA 3.1：使用固定系統提示詞和 summarize 指令。
 - BART：使用多層次摘要算法。
- 結果分析：
 - LLaMA 3.1 摘要長度不足，導致分數較低。
 - BART 摘要質量較高，但計算資源消耗較大。

Model		R-1	R-2	R-L
1.	LLaMA 3.1	0.3350	0.1280	0.2054
2.	BART + 多層次摘要	0.4754	0.2334	0.4150

Table: 單一模型摘要性能評估

實驗一：文本壓縮 (1/2)

- 設計背景
 - 目的: 減少冗餘信息，降低計算負擔。
 - 使用 Ollama API 創建 **compressor** 模型，負責壓縮任務。
- 系統提示詞設計
 - ① You can ONLY remove unimportant words.
 - ② Do not reorder the original words.
 - ③ Do not change the original words.
 - ④ Do not use abbreviations or emojis.
 - ⑤ Do not add new words or symbols.
 - ⑥ **Provide ONLY the result text, without any explanations or additional information.**
- 使用者提示詞設計
 - 四項指令任務：compress、highlight、summarize、abstract。
 - 用於評估模型在不同指令下的效果。

實驗一：文本壓縮 (2/2)

- 壓縮模型的效果評估

task		R-1	R-2	R-L
I.	compress	0.8271	0.6775	0.8237
II.	highlight	0.8906	0.7501	0.8900
III.	summarize	0.3415	0.2241	0.2803
IV.	abstract	0.8117	0.7228	0.8031

Table: 壓縮任務與原始資料 article 的 ROUGE 比較

Scott Dann was a fraction offside when he set up Glenn Murray for Palace's first goal (2), but it would be harsh to put too much blame on assistant John Brooks. He was spot on with two equally tight calls in the same move — before Dann got the ball (1) and for Murray's finish (3). The speed of it, plus two players blocking his view, made it unbelievably hard for him to get all three right.

(~ 77 words)

Summarize

Abstract

Compress

Highlight

Scott Dann set up Glenn Murray's goal but was just offside himself. Assistant John Brooks had a tough call with multiple factors making it difficult for him to get it right.

(~ 31 words)

Scott Dann was offside when he set up Glenn Murray for Palace's first goal but would be harsh to put blame on assistant John Brooks. He was spot on with two tight calls before Dann got the ball and for Murray's finish. The speed of it made it hard for him to get all three right.

(~ 56 words)

Scott Dann was offside when he set up Glenn Murray but would be harsh to put blame on assistant John Brooks. He was spot on with two tight calls before Dann got the ball and for Murray's finish. The speed of it plus players blocking his view made it hard for him to get all three right.

(~ 57 words)

Scott Dann was a offside when he set up Glenn Murray for Palace's first goal but it would be harsh to put too much blame on assistant John Brooks. He was spot with two equally tight calls in the same move — before Dann got the ball and for Murray's finish The speed of it plus two players blocking his view made it hard for him to get right

(~ 69 words)

實驗一：文本壓縮 (2/2)

- 壓縮模型的效果評估

task		R-1	R-2	R-L
I.	compress	0.8271	0.6775	0.8237
II.	highlight	0.8906	0.7501	0.8900
III.	summarize	0.3415	0.2241	0.2803
IV.	abstract	0.8117	0.7228	0.8031

Table: 壓縮任務與原始資料 article 的 ROUGE 比較

- 實驗結果結論

- highlight 和 abstract：在壓縮率和信息保留度間取得平衡。
- summarize：可能由於文本過於簡化，ROUGE 分數偏低。

- 壓縮結果將作為後續多層次摘要的輸入。

實驗二：BART 多層次摘要 (1/2)

- 實驗背景：
 - 基於實驗一的壓縮文本，作為本階段摘要的輸入。
- 實驗目標：
 - 分析與單層摘要方法的效率和準確性差異。
 - 檢驗多層次摘要在結構連貫性上的表現。
- 實驗設計：
 - 壓縮文本逐層進行摘要，每層輸出的摘要用於下一層輸入，逐步提煉內容。
 - 每層摘要以固定的 token 限制進行，確保處理效率與摘要質量。
 - 引入 4 種不同的指令任務，分析不同指令在多層次摘要中的優勢。

實驗二：BART 多層次摘要 (2/2)

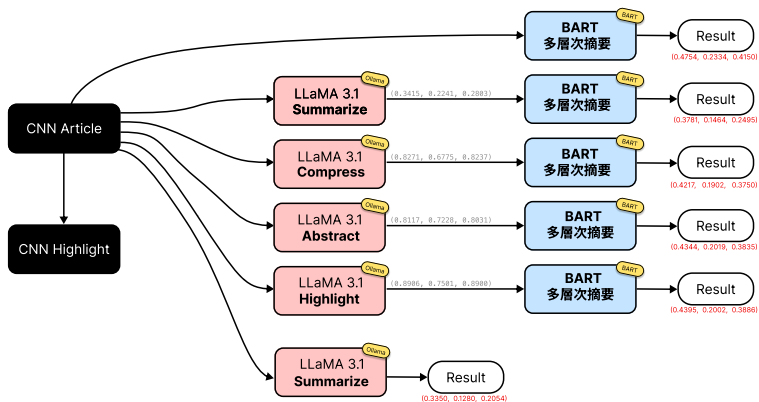
task		R-1	R-2	R-L
V.	compress+bart	0.4217	0.1902	0.3750
VI.	highlight+bart	0.4395	0.2002	0.3886
VII.	summarize+bart	0.3781	0.1464	0.2495
VIII.	abstract+bart	0.4344	0.2019	0.3835

Table: 摘要結果與 CNN/DailyMail highlights 的 ROUGE 比較

● 分析與觀察：

- highlight+bart (task VI.) 分數表現最佳。
- summarize+bart (task VII.) 在壓縮後分數較低，但此階段有所改善。
- 方法優勢：有效縮短文本處理時間，減少訊息流失。

流程圖



公開平台比較

- Papers with Code 平台
 - 匯總機器學習領域的最新研究成果。
 - 提供不同資料集的基準比較，評估任務效能。
 - 針對 CNN/Daily Mail 資料集，列出基於 ROUGE 指標的摘要模型結果。
- 表格標示說明
 - **紅色字體**: 單純 BART 模型下多層次摘要演算法的分數。
 - **黑色粗體字**: 本研究的最佳方法，**highlight** 指令的多層次摘要演算法分數。

Rank	Model	ROUGE-1
1.	Scrambled code + broken (alter)	48.18
2.	BART 多層次摘要	47.54
3.	PEGASUS + SummaReranker	47.16
⋮	⋮	⋮
8.	MatchSum (BERT-base)	44.22
9.	LLaMA 3.1 + BART 多層次摘要	43.95
10.	BertSumExt	43.85
⋮	⋮	⋮

Table: ROUGE-1 in Papers with Code

Rank	Model	ROUGE-2
1.	BART 多層次摘要	23.34
2.	PEGASUS + SummaReranker	22.55
3.	Fourier Transformer	21.55
⋮	⋮	⋮
12.	BERTSUM+Transformer	20.24
13.	LLaMA 3.1 + BART 多層次摘要	20.02
13.	Scrambled code + broken (alter)	19.84
⋮	⋮	⋮

Table: ROUGE-2 in Papers with Code

Rank	Model	ROUGE-1
1.	Scrambled code + broken (alter)	45.35
2.	PEGASUS + SummaReranker	43.87
3.	HAT-BART	41.52
4.	BART 多層次摘要	41.50
5.	GLM-XXLarge	41.4
⋮	⋮	⋮
13.	BERTSUM+Transformer	39.63
14.	LLaMA 3.1 + BART 多層次摘要	38.86
15.	Selector+Pointer Generator	38.79
⋮	⋮	⋮

Table: ROUGE-L in Papers with Code

● 測試結果

- 在 ROUGE 三項指標上的穩定性能和競爭力表現。
- 有效性與實用性: 精簡計算資源需求並保持摘要水準。

Contents

1. 介紹

2. LLaMa 3.1 文本壓縮

2.1 固定系統 vs. 動態系統

2.2 提示詞設計

3. Bart 多層次摘要演算法

3.1 多層次摘要框架

3.2 參數選取

4. 實驗方法與結果

4.1 單獨模型實驗的觀察

4.2 實驗一：文本壓縮

4.3 實驗二：BART 多層次摘要

4.4 Papers with Code 網站的比較結果

5. 結論與未來展望

結論與未來展望

● 結論

① 避免注意力機制分散與硬體資源消耗問題

- LLaMA3.1 壓縮長文本，減少冗餘內容。
- BART 多層次摘要，逐步提取語義連貫且準確的摘要。

② 實驗結果

- 在 CNN/Daily Mail 資料集上，於公開平台上取得良好結果。
- 平衡效能與資源效率，展現穩定性與競爭力。

③ 實際應用上的靈活性

- 適合計算資源有限的環境。
- 可根據不同需求調整壓縮與摘要的層次。

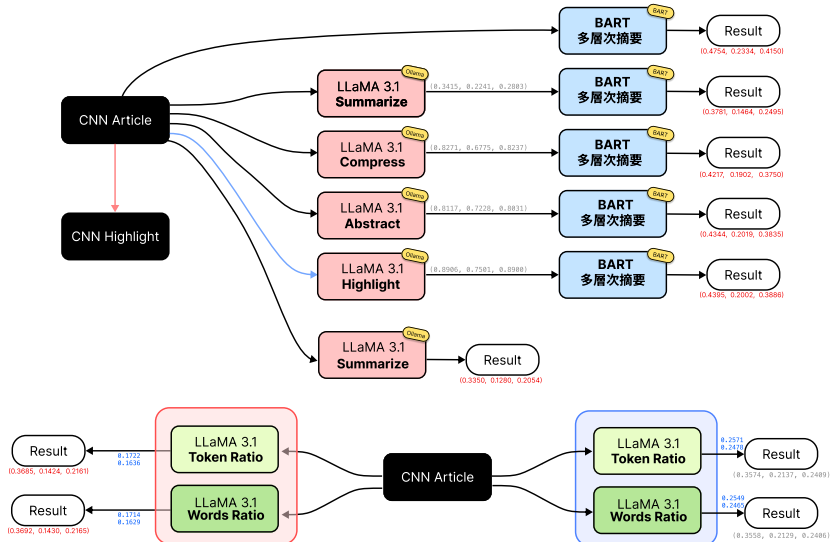
● 未來展望

- 觀察在其他數據集上的表現，驗證是否完全避免位置偏差問題。
- 持續優化模型效能，提升摘要質量及多樣性。
- 拓展模型的應用範圍：如多語言摘要、領域專業文檔摘要等。

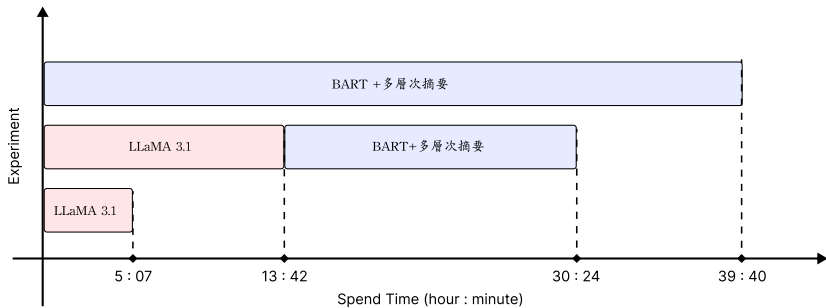
Thank you!

Back Up Pages

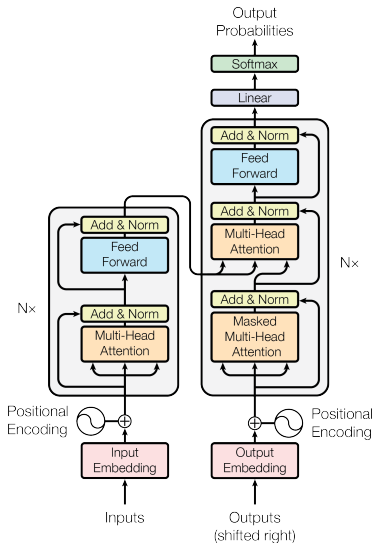
流程圖



花費時間圖

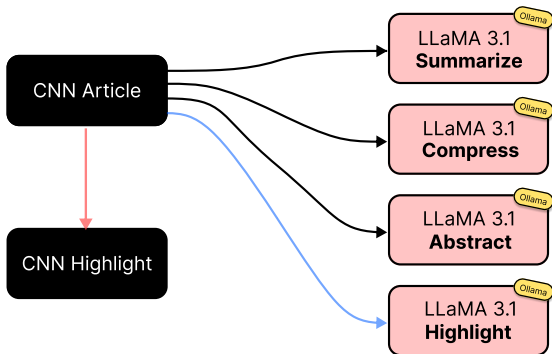


Transformer 架構圖



動態系統提示詞

- 目標壓縮比
 - 使用原始資料 (摘要/原始文本) 和 highlight 來進行測試。



動態系統提示詞

User Prompt:

- ① Please summarize the following text to approximately **{ratio}** of the original length.
- ② The output should be a direct, concise summary without any extra commentary or introductory text.
- ③ original text: {...}

動態系統提示詞

- 目標壓縮比
 - 使用原始資料 (摘要/原始文本) 和 **highlight** 來進行測試。
- 結果顯示，語言模型未能有效遵守設定的壓縮比，特別是在較大壓縮比設定下，生成的文本往往偏離目標值。

task	test item	target average ratio	results average ratio	R-1	R-2	R-L
A-1	token ratio	0.1003	0.1636	0.3685	0.1424	0.2161
A-2	words ratio	0.1048	0.1714	0.3692	0.1430	0.2165
B-1	token ratio	0.8111	0.2478	0.3574	0.2137	0.2409
B-2	words ratio	0.8068	0.2549	0.3558	0.2129	0.2406

Table: task A 與 task B 壓縮比測試平均結果

Ollama 官網：LLaMA 3.1 介紹


[Discord](#)
[GitHub](#)
[Models](#)

[Sign in](#)
[Download](#)

llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

[tools](#)
[8b](#)
[70b](#)
[405b](#)

[↓ 17.1M Pulls](#)
[🕒 Updated 5 weeks ago](#)

8b	93 Tags	ollama run llama3.1	
Updated 5 weeks ago		46e0c10c039e • 4.9GB	
model	arch llama • parameters 8.83B • quantization Q4_K_M		4.9GB
params	{ "stop": ["< start_header_id >", "< end_header_id >", "< end_text >"] }		96B
template	{ { - if or .System .Tools } }< start_header_id >system< end_header_id >{ { - if or .System .Tools } }< end_text >		1.5kB
license	LLAMA 3.1 COMMUNITY LICENSE AGREEMENT Llama 3.1 Version Re...		12kB