實驗架構

下面我幫你設計一個 **真正有研究貢獻、可以回答你們疑問、又能清楚比較傳統 vs 深度模型** 的完整實驗架構。
我會把它拆成五個階段，讓你們不會走偏、能避開「只是模型 PK」的陷阱。

這套架構你可以**直接放進期末報告方法章節**。

🌟 **整體概念：

不是在問“誰比較準？”
而是在問：
“Prompt 改變語言特徵後，不同模型**為什麼**會失效？”**

所以實驗架構一定要能回答：

1. Prompt 如何改變語言特徵？
2. 哪類模型被哪些 prompt 騙過？
3. 傳統 vs 深度的「失效模式」差在哪？
4. 有沒有防禦策略讓模型更穩？

🔥 【最推薦的實驗架構】

分成五個主要階段 ↓

---

① 資料準備（3 類文本 × 3 prompt × 2 模式）

(A) 人類文本（真實資料）

選兩種領域（建議）：

- 新聞（News Category Dataset）
- 產品評論（Amazon Reviews）

每個領域抽 2,000～3,000 篇
→ 當作「Human baseline」

(B) AI 文本（我們自己生成）

三類 Prompt（你們 PDF 裡有的很棒）

1. **指令式（Instruction）**
2. **敘述式（Narrative / Human-like）**
3. **角色式 / 情感式（Persona）**

搭配兩種生成多樣性

- **Low temperature**（0.2）→ 很 AI
- **High temperature**（0.9）→ 很人類

最終形成：

3 Prompt × 2 Temperature × 2 領域
= 12 種 AI 文風格

每格生成 200 篇 → 共 2,400 篇
非常足夠。

---

② 語言特徵分析（Prompt 影響模型的第一環）

這一步不是模型，而是**統計語言學**（你們的貢獻）。

對每種文本計算：

基本統計

- 平均句長
- 句長標準差（burstiness）
- type-token ratio（詞彙多樣性）

- Functional words 比例（I, you, like, um…）
- 標點使用分布

語言模型特徵

- GPT-2 perplexity
- perplexity variance
- 重複率（token repetition）

後續視覺化

- LSA / UMAP 降維散佈圖
- K-means clustering（看是否人類 vs AI 自然分群）

目的：找出 Prompt 真正改變了哪些語言特性。
這是你們的第一個「研究貢獻」。

---

③ 模型訓練（傳統 vs 深度模型比較）

你們會比較：

傳統模型：

- **TF-IDF + SVM**
- **Stylometry + XGBoost**

深度模型：

- **BERT / RoBERTa 微調**

Hybrid（建議，一定大加分）：

- **CLS embedding + Stylometric features → MLP**

🔧 如何公平比較？

使用：

- 同一套 **train/val/test split**
- 每個模型給 **同樣的超參數搜尋預算**（例如 20 組）
- 用 validation set 選各模型的「代表最佳模型」
- 不需保證是全宇宙最佳，只需「合理公平」

這樣就避開你們原本怕的問題：

> “如果是參數調不好，而不是模型真的比較差？”

→ 我們已經控制住「調參資源公平」。

---

④ 穩健性測試（核心實驗）

這是重頭戲。

對每種模型測試：

(1) Baseline：普通 Prompt（指令式）

看模型在最容易的狀況下表現如何。

---

(2) Prompt 偽裝測試（Hard mode）

測試：

- 敘述式（Narrative）
- 角色式（Persona）
- High temperature（0.9）
- 特別是「像大學生寫作」 prompt（最強攻擊）

看每個模型掉多少：

> AUROC_drop = (Baseline AUROC) – (Evasion AUROC)

這就是模型的「穩健性分數」。

---

(3) 未見 Prompt 測試（Unseen attack）

拿一種沒有在訓練中出現過的 Prompt
例如：「加入錯字、俚語、縮寫、emojis」。

測試：

> 哪些模型能泛化到從未看過的 Prompt？
> 哪些完全爆炸？

---

⑤ 防禦策略（真正的研究貢獻）

你們可以提出一個「我們的防禦方法」：

方案 A：對抗訓練（Adversarial Training）

訓練集加入部分偽裝 Prompt → 模型學會抵抗它。

測：

- 能不能撐住更多 unseen prompt？
- AUROC 掉得有沒有比較少？

方案 B：Hybrid Detector

RoBERTa CLS + stylometry 能不能比純 BERT 更穩？

方案 C：Prompt-invariant Learning（選做）

讓模型學會忽略 prompt_type
→ 做 multi-task（AI/Human + prompt type）

🧩 最終成果（你們報告的亮點）

你們可以得到三種研究貢獻：

🔥 貢獻 1：Prompt 如何操控 AI 文的語言特徵？

（模型外的 insight）

例如：

- 敘述式 Prompt 會提高句長變異
- Persona Prompt 會增加 functional words
- High temperature 會提高詞彙多樣性 TTR
- 這些特徵讓 AI 文更像人類

🔥 貢獻 2：不同模型遇到不同 Prompt 的「失效模式」

（真正的模型比較）

例如：

- BERT 在 high-T narrative prompt 掉最多
- SVM 在 persona prompt 掉比較少
- XGBoost 對句長變異特別敏感
- Hybrid 模型在所有 prompt 都最穩

🔥 貢獻 3：提出防 Prompt 的 Detector（你們自己做的）

例如：

- 加入 adversarial training → BERT 在 evasion prompt 掉分減少 30%
- Hybrid 模型在 unseen prompt 上也不容易爆炸

📌 最重要的：這個架構避免了「只是模型 PK」的陷阱

🎯 最直接的結論（你想要的那句話）

如果不考慮 Prompt 攻擊 → 深度學習模型（BERT/RoBERTa）通常最準。

如果考慮 Prompt 攻擊 → 混合模型（深度 + 風格特徵）最穩健。

如果資料量很小 → 傳統模型（SVM/XGBoost）反而不會太差。

下面是「真正的原因」
（你們報告裡會用到，教授會喜歡看到的那種分析）

✅ 1. 哪種模型最容易偵測出 AI 文？

情境 A：AI 文沒有刻意偽裝（普通 Prompt）

→ 深度模型（BERT、RoBERTa）= 最好

原因：

- 深度模型能抓語意一致性（AI 通常太一致）
- 能抓句法規律（AI 句子風格太穩）
- 能抓長距依存（AI 常常段落間太有邏輯）

📌 **普通生成時，深度模型很容易抓到 AI 味。**

情境 B：AI 文經過 Prompt 工程偽裝

→ 深度模型反而最容易被騙（掉分最慘）

原因：

- Prompt 可以讓語氣變得人類化：隨機、口語、錯字
- 這些變化會改動深度模型依賴的「語意 embedding 結構」

📌 **深度模型在「避檢式 Prompt」（evasion prompt）下魯棒性差。**

情境 C：只有統計風格差異（不看語意）

→ 傳統模型（SVM/XGBoost + TF-IDF）反而還撐得住一些

原因：

- 傳統模型看的是「風格統計」而不是語意
- Prompt 可以改語氣，但很難完全改掉**統計特徵**（如 burstiness、TTR）

📌 在某些 prompt 下，傳統模型掉分比 BERT 少。

⭐ 最穩健（不一定最準）的模型是：

Hybrid（深度 embedding + stylometric features）
也就是：

RoBERTa CLS +（句長變異 + TTR + POS + 標點 + PPL）→ MLP

原因：

- 深度負責抓語意（AI 過於一致）
- 統計特徵負責抓人類風格（混亂、不規律）
- Prompt 只能同時騙其中一部分，不太可能兩邊都偽裝成功

📌 **Hybrid 模型是目前對 Prompt 攻擊最穩健的偵測器。**

你們如果做這個 → 研究貢獻直接有。

🎯 所以「哪個最容易判斷 AI 文」的完整答案是：

模型種類	普通 prompt	偽裝 prompt	穩健性	為什麼
深度模型（BERT）	⭐⭐⭐⭐ 最準	⭐ 失效最嚴重	⭐⭐	深度模型看語意 → prompt 改語氣就 GG
傳統模型（SVM/XGB）	⭐⭐ 中等	⭐⭐ 還能撐	⭐⭐⭐	看統計特徵 → 比較不會被語意偽裝騙
Hybrid（深度+風格）	⭐⭐⭐⭐ 高	⭐⭐⭐ 不容易被騙	⭐⭐⭐⭐	同時抓語意 + 統計，prompt 很難一次騙兩種

🎁 送你一句「報告版本」的總結（可以直接貼）

在無偽裝情境下，深度模型（BERT/RoBERTa）具有最佳 AI 文本偵測性能。但在 Prompt Engineering 等刻意偽裝下，深度模型容易失效；相較之下，傳統模型因依賴統計風格特徵，在部分情境下具有較高的穩健性。而結合深度語意表示與風格特徵的 Hybrid 模型同時抓取 AI 文本的一致性與統計特性，因此呈現最佳的跨 Prompt 魯棒性。

分頁 3

1) 研究目標與問題

主目標：建立並分析「AI 生成文字 vs. 人類文字」的可偵測性，並探討 **prompt 類型**、**主題與語言** 對偵測難度的影響。
研究問題：

- RQ-A：不同 **prompt 類型**（指令式 / 敘述式 / 角色扮演 / 模仿人類錯誤）是否改變可偵測性？
- RQ-B：不同 **主題領域**（評論 / 新聞 / 論述）是否影響可偵測性？
- RQ-C：**中 vs. 英** 是否有系統性差異？

2) 變項設計

- **自變項**：Prompt 類型（4）、主題（3）、語言（2）、模型來源（可選 2：如 GPT-4/Claude）
- **依變項**：被偵測為 AI 的機率（分類器輸出）、最終判斷（AI/Human）
- **控制變項**：篇幅（例如固定 120–200 詞）、主題配對、生成溫度等

3) 實驗條件與樣本量（建議）

2（語言） × 3（主題） × 4（prompt 類型） × 50 篇 = **1,200 篇 AI 文**，人類對照文同樣配對收集 **1,200 篇**（總 2,400 篇）。樣本可減半（各 25 篇）變成 1,200 篇總量以符合時間成本。

4) 文本來源

A. 「現成可用」AI vs Human 標註資料

- **HC3（Human–ChatGPT Comparison Corpus）**：同題的人類答覆與 ChatGPT 生成，含中英文版本（可直接用來做模型訓練/驗證）。ModelScope+3Hugging Face+3GitHub+3

- **Kaggle 人工標註資料集（多個）**：
  - AI vs Human Text（大規模作文匯整版）Kaggle+1
  - Human vs AI Text Classification（5k 樣本）Kaggle
  - AI vs Human Text Dataset（1k 樣本，易於快速試驗）Kaggle

B. 「純人類文本」作為對照組（多領域）

- **Yelp Open Dataset**（評論文本，英文，數百萬則可取樣）business.yelp.com+2Kaggle+2
- **Amazon Reviews**（英文長期評論資料；可選品類抽樣）jmcauley.ucsd.edu+2Kaggle+2
- **THUCNews**（中文新聞分類資料集；可取 3–5 類做主題對照）thuctc.thunlp.org+1
- （可選）**Wikipedia 結構化文本新資料源**（研究用途，適合作為「正式論述」風格的人類文本）The Verge

5) 數據蒐集與生成流程

1. **定義主題**（例如：評論、新聞、論述各 1 類；中文＋英文各一套）。

2. **人類文本抽樣**
    - 英文：Yelp（評論）、Amazon（評論/長文本）。
    - 中文：THUCNews（選 3 個類別，如社會/科技/娛樂，抽句段落或導言）。 thuctc.thunlp.org+3business.yelp.com+3Kaggle+3
3. **AI 文本生成**
    - 為每篇人類文本的「主題與大綱」設計 3 種 prompt（指令式 / 敘述式 / 角色扮演），用同一主題生成相似篇幅。
    - 控制長度（字數）、重複度（可限制「避免重複句式」的提示），並保留 `prompt_type` 欄位。
4. **品質控管**
    - 去重（MinHash / simhash），過短/過長過濾；統一字數範圍；自動記錄生成參數（溫度、top-p 等）。

6) 特徵工程與偵測器

- `TF-IDF` + `Logistic Regression / Linear SVM`，輕巧可解釋（看重要 n-grams、標點密度）。
- **輔助統計特徵**（作為可視化與 ablation）：
    - 詞彙多樣性（TTR）、平均句長、標點密度、重複 n-gram 比例
    - 可讀性（英：Flesch–Kincaid）、情感分數（VADER，英）
    - **Perplexity**（以公開 LM 計算，如 GPT-2/roberta LM head 作近似）：AI 文通常更「可預測」。
- **現成資料做 baseline**：用 HC3 先訓練一個判別器，再看對 Yelp/Amazon/THUCNews 的外部泛化表現（能不能不降太多）。Hugging Face

7) 評估與統計分析

- **指標**：Accuracy / Precision / Recall / F1 / ROC-AUC；混淆矩陣。
- **假設檢定**：
    - RQ-A（prompt 類型）：各類型的「被判 AI 機率」做 **單因子 ANOVA**（或 Kruskal–Wallis）＋ Tukey HSD 事後比較。
    - RQ-B（主題）：主題 × 類型做 **雙因子 ANOVA**（觀察交互作用）。
    - RQ-C（語言）：語言 × 類型的二因子 ANOVA；或把語言當共變項做 ANCOVA。

Proposal_1

文字探勘期末專案 - 第 12 組

B13705003 謝欣佑、B13705007 林采穎、B13705027 黃柏琁、B13705055 陳沐頤、B13608027 吳昕醍

1. 研究背景與目的

近年生成式 AI（如 ChatGPT、Gemini）能生成與人類極為相似的文本，造成教育、媒體與研究倫理上的挑戰，現有 AI 文本偵測研究多著重於模型分類精度，卻忽略了「**Prompt Engineering 對生成風格與可偵測性的影響**」。因此本研究想探討「**不同 prompt 類型是否會改變 AI 生成文字被偵測為 AI 的可能性**」，並分析生成文字與真實人類文本在語言特徵上的差異。

2. 研究設計概述

資料來源	內容	目的
現成標註資料（如 HC3、Kaggle AI vs Human）	含 AI 與人類文字對照	用來訓練基準分類模型（AI 偵測器）
純人類文本（西元2000年以前）	真實世界人類撰寫文本	作為外部人類對照，測試模型泛化能力
自建資料（使用ChatGPT等 AI 工具以不同 Prompt 生成的 AI 文）	由研究者設計多類 prompt 生成	作為實驗組，用來測試 Prompt 類型對偵測難易度的影響

3. Prompt 類型設計

類型	範例 Prompt	預期語氣／風格特徵
A. 指令式	「請以正式語氣撰寫一篇三段式短文，主題為環保的重要性。」	條理分明、邏輯清晰、句式規律
B. 敘述式	「談談你對環保的想法，可以用日常語氣。」	自然、口語化、較隨意
C. 設定角色式	「你是一位詩人，用感性語氣描述環保的重要。」	風格化、具情感色彩

4. 研究假設

編號	假設內容	理論依據
H1	使用**自然語氣**（如模仿學生作文）的 prompt，生成文字較難被模型辨識為 AI 產出。	因為語法錯誤、人性化的不一致會增加「人類特徵」。
H2	使用**明確結構**的 prompt（例如「請用三段式論述」），生成文字更容易被辨識為 AI 文字。	因為機器產生的句子更具邏輯性與結構性。
H3	改變**指令長度**與**明確度**會影響生成文字的可偵測性。	Prompt 複雜度會改變語言模型的生成風格與統計特徵。

Proposal_2

文字探勘期末專案 - 第 12 組

Prompt 工程對 AI 文本偵測的影響：傳統與深度學習模型的比較分析

B13705003 謝欣佑、B13705007 林采穎、B13705027 黃柏琁、B13705055 陳沐頤、B13608027 吳昕醍

1. 專案背景與研究問題

近年生成式 AI（如 ChatGPT、Gemini）文本的普及，對教育與研究倫理帶來挑戰，但現有 AI 文本偵測研究大多注重模型分類的精度，卻忽略了提示工程（Prompt Engineering）對文本生成風格與模型可偵測性的影響，因此我們想探討以下兩個研究問題：

i. Prompt 類型是否會系統性的改變 AI 文本的語言特徵，進而影響其被模型偵測的難易度？
ii. 不同原理的偵測模型在面對刻意設計的 Prompt 文本時，其性能的穩健性差異為何？

並提出核心假設如下：

i. 模仿自然語氣（如敍述式 Prompt）的 AI 文本將最難被偵測。
ii. 深度學習模型（BERT/LoRA）對抗 Prompt 類型變化的穩健性將優於傳統模型 （XGBoost / SVM）。

2. 研究方法

我們預計採取「資料生成」、「模型優化訓練」與「穩健性實驗」三階段進行研究。

2.1 資料來源與實驗變因（Prompt 類型）

資料類型	目的
基準標註資料	訓練與優化兩種基準偵測器
自建實驗組資料	核心實驗，測試偵測難易度
純人類文本	語言特徵對照組

Prompt 類型	預期風格
A. 指令式	條理分明、易偵測
B. 敍述式	自然、口語化、難偵測
C. 設定角色式	風格化、具情感色彩

2.2 實驗步驟與模型訓練

步驟	內容與核心技術
i. 準備與特徵工程	使用 TF-IDF 進行傳統模型特徵工程，並準備 BERT Tokenization。
ii. 傳統模型訓練	訓練 XGBoost/SVM，使用 Grid Search 進行參數優化，獲得最佳性能。
iii. 深度模型訓練	使用 LoRA 微調 BERT/RoBERTa（learning_rate 等參數）獲得最佳性能。
iv. 核心穩健性測試	將三類 Prompt 文本輸入兩種最佳模型，記錄其「被偵測為 AI」的準確率。

3. 預期成果展現

i. 兩種模型（傳統與深度學習）的訓練與參數優化結果。
ii. 繪製長條圖，量化比較兩模型對三類 Prompt 文本的偵測準確率，驗證模型穩健性。
iii. 基於 LSA 降維與聚類分析（K-means）的語言特徵視覺化圖表。

完整報告書草稿

📚 文字探勘期末專案報告書草稿

🏆 專案名稱：Prompt 工程對 AI 文本偵測穩健性的影響：傳統與深度學習模型的比較分析

1. 研究背景與目的 (Introduction)

1.1 研究背景

生成式 AI (如 ChatGPT) 的普及對文本偵測技術構成巨大挑戰[1]。現有研究多專注於提升單一模型的分類準確度[2]，卻常忽略 **Prompt Engineering**[3] 對生成文本風格與可偵測性帶來的系統性影響。特別地，不同 AI 文本偵測器（傳統與深度學習）在面對刻意設計的 Prompt 時，其**性能的穩定性（穩健性）**差異，是當前研究的關鍵盲點。

1.2 專案目的

本專案的核心目的，是通過**系統化的模型訓練與比較**，探討不同分類模型對 Prompt 類型變化的敏感度。具體目標如下：

1. **訓練與優化：** 訓練並優化兩種原理截然不同的分類模型（傳統的 XGBoost/SVM[4]與現代的 BERT/LoRA[5]），使其達到各自的最佳基準性能。
2. **穩健性比較：** 比較這兩種模型在偵測由三類 Prompt 類型（指令式、敘述式、設定角色式）生成的 AI 文本時，其**偵測準確率的衰退程度**。
3. **變因分析：** 驗證 Prompt 類型對生成文本的語言特徵與模型偵測難易度的影響。

2. 研究假設 (Hypotheses)

編號	假設內容	理論依據	專案關聯性
H1	敘述式 Prompt (B) 生成的文字最難被模型辨識為 AI 產出。	模仿人類不一致性，增加「人類特徵」[6]。	測量**絕對偵測難度**。
H2	指令式 Prompt (A) 生成的文字最容易被模型辨識為 AI 文字。	機器產生的句子更具邏輯性與結構性[7]。	測量**絕對偵測難度**。
H3	**深度學習模型** (BERT/LoRA) 對抗 Prompt 類型變化的穩健性將優於**傳統機器學習模型** (XGBoost/SVM)。	深度學習模型能捕捉更複雜的語義特徵，泛化能力較強。	測量**模型比較穩定性**。

H4	不同 Prompt 類型生成的文本，其語言特徵在 LSA 空間中可被清晰區隔。	Prompt 複雜度會改變語言模型的生成風格與統計特徵[8]。	測量**特徵空間可區隔性**。

3. 研究設計與資料 (Methodology)

3.1 資料來源與類型

資料來源	內容	目的
基準標註資料	含 AI 與人類文字對照[9]	**訓練與優化**兩種基準分類模型。
自建實驗組資料	由研究者設計多類 Prompt[10]生成的 AI 文[11]	**核心實驗變因**，測試 Prompt 類型對偵測難易度的影響。
純人類文本 (西元 2000 年以前)	真實世界人類撰寫文本[12]	作為**外部人類對照組**，用於語言特徵分析。

3.2 Prompt 類型設計 (實驗變因)

類型	範例 Prompt	預期語氣/風格特徵
A. 指令式	「請以正式語氣撰寫一篇三段式短文，主題為環保的重要性。」	條理分明、邏輯清晰、句式規律[13]
B. 敘述式	「談談你對環保的想法，可以用日常語氣」	自然、口語化、較隨意[14]
C. 設定角色式	「你是一位詩人，用感性語氣描述環保的重要。」	風格化、具情感色彩[15]

4. 實驗步驟與流程 (Experimental Procedure)

步驟 1：資料準備與特徵工程

1. **文本清洗：** 對所有資料集進行通用文本清理。

2. **實驗組資料生成：** 使用 ChatGPT 等工具，依據 A、B、C 三類 Prompt 生成足夠數量的實驗文本。
3. **傳統模型特徵：** 對基準標註資料和純人類文本進行分詞，並計算 **TF-IDF 向量**。
4. **深度學習特徵：** 對基準標註資料進行 **Tokenization**，準備模型訓練輸入。

步驟 2：基準模型訓練與參數優化

目標： 找出兩種模型的**最佳參數組合** P_{trad} 和 P_{deep}。

1. **傳統模型優化 (XGBoost/SVM)：** 使用基準標註資料，執行 **Grid Search/Random Search** 搭配交叉驗證，調整模型參數（如：max_depth、learning_rate），並記錄最佳基準準確率 $\text{Acc}_{\text{trad, base}}$。
2. **深度學習模型優化 (BERT/LoRA)：** 使用基準標註資料，進行 Fine-tuning。調整 learning_rate 和 epochs，並記錄最佳基準準確率 $\text{Acc}_{\text{deep, base}}$。

步驟 3：核心實驗：穩健性測試

目標： 比較兩模型對 A, B, C 文本的偵測能力。

1. **測試執行：** 將**自建實驗組資料**（A, B, C 三類文本）輸入 P_{trad} 和 P_{deep} 兩種最佳模型。
2. **數據記錄：** 記錄每種模型對每一類 Prompt 文本**「判斷為 AI 產出」的準確率** $\text{Acc}_{\text{trad, type}}$ 和 $\text{Acc}_{\text{deep, type}}$。

步驟 4：結果分析與特徵量化

1. **穩健性分析 (H3 驗證)：** 計算兩種模型對每類文本的**性能衰退幅度** ($\text{Acc}_{\text{base}} - \text{Acc}_{\text{type}}$)。比較衰退幅度，驗證 H3 假設。
2. **語言特徵分析 (H4 驗證)：** 對所有文本（A, B, C, 純人類）的 TF-IDF 向量應用 **Latent Semantic Analysis (LSA)** 進行降維，並執行 **K-Means 聚類**，觀察文本在語義空間的分佈。

5. 預期分析與結果呈現 (Expected Analysis and Presentation)

5.1 模型穩健性與可偵測性分析 (H1, H2, H3 驗證)

- **數據呈現：** 繪製**分組條形圖**，比較 XGBoost/SVM 與 BERT/LoRA 在偵測 A、B、C 三類文本時的準確率。
- **分析重點：**
    - **H1/H2：** 確定 B 類文本是否為兩種模型的**共同盲點**（難度最高），A 類文本是否最容易偵測。
    - **H3 (核心)：** 比較 $\text{Acc}_{\text{trad, type}}$ 與 $\text{Acc}_{\text{deep, type}}$ 的差異。如果深度學習模型的準確率在所有類型文本上都高於傳統模型，則驗證 H3（BERT/LoRA 穩健性較強）。

5.2 語言特徵與風格分析 (H4 驗證)

- **分析方法：** **LSA/K-Means 聚類**。
- **結果呈現：** 繪製 LSA 降維後的**二維散點圖**。圖中將以顏色或形狀標註四類文本（A, B, C, 純人類）。
- **分析重點：** 驗證 H4，即不同 Prompt 類型生成的文本在語義空間上是否能被 LSA 清晰區隔。同時，觀察 B 類文本的聚類是否與純人類文本的聚類空間最接近，以此佐證 H1（B 類難以偵測）的語言學依據。

分頁 5

**方法 1：對抗資料增強（Adversarial Training）

→ 最簡單、最有效、你們一定做得出來。**

核心概念

如果攻擊者用「避檢 Prompt」生成 AI 文，那我們就讓「偵測器」在訓練時也看過這些文，學會它們的模式。

這就像：

> 「我訓練你辨識作弊手法 → 所以就算有人變招，你也能擋住。」

實作方式（非常容易）

1. 生成兩種 AI 文：
    - **normal-prompts**（一般語氣）
    - **attack-prompts**（避檢、口語、錯字、burstiness、亂語氣）

合併成訓練集：

```
train = human_text + normal_AI + attack_prompt_AI
test  = human_text + unseen_attack_AI
```

2.
3. 訓練你們自己的 detector（例如 RoBERTa）
4. 要做的測試：
    - baseline (only normal data)
    - adversarial training 後比較 AUROC 差異
    - 是否能抵擋 unseen prompt

研究貢獻寫法

> 我們提出並實作了一套「對抗式資料增強（Adversarial Data Augmentation）」策略，使模型能抵禦由 prompt engineering 所產生的偽裝 AI 文本攻擊。

👉 **這是一個正式的研究 innovation**，而且你們完全做得到。

**方法 2：混合偵測器（Hybrid Detector）

→ 將「風格特徵」 + 「語義 embedding」結合。**

這招可以「破解避檢 Prompt」。

為什麼它有效？

因為 Prompt Engineering 常做以下事情：

- 加錯字
- 加同義詞
- 簡化語法
- 讓語句看起來比較像人、較不一致

→ 這些都會破壞語言模型特有的「統計特徵」。

傳統 detector（RO-BERTa-only）可能會被騙，但：
風格特徵（stylometry）不會。

你們可以加入的特徵（你們能做）

✔ 字母多樣性、詞彙多樣性（TTR）
✔ 平均句長、變異數
✔ 標點比例
✔ POS-character n-gram
✔ 重複率（burstiness）
✔ GPT-2 perplexity
✔ emoji、縮寫、口語詞比例

混合方式：

將這些特徵 **concat** 到 Transformer 的 CLS 向量後面：

```
[CLS_embedding | stylometry_features] → MLP → output
```

這個模型的名字你們可以叫：

> **Hybrid Anti-Prompt Detector (HAPD)**
> **or**
> **Mixed Stylometric–Semantic Detector**

這聽起來就很像一個新方法。

**方法 3：Prompt-invariant training（讓模型學「不管 prompt 怎麼變，AI 文特徵本質不變」）

→ 進階，但你們也能做到簡易版。**

做法（簡易版）

你們在訓練資料標註：

```
text | label(human/AI) | prompt_type
```

讓模型學習 **忽略 prompt_type 的差異**。

可以做到兩種：

A. prompt-type 作為「nuisance factor」

模型學習：

> 就算 prompt 不同，我仍要專注在「AI vs Human 本質差異」。

方法：
在訓練時加入 **prompt embedding**，讓模型學習「扣掉 prompt noise」。

B. prompt-type 作為多任務學習（multi-task）

模型輸出：

- 任務 1：AI vs Human
- 任務 2：prompt type（normal / attack / casual / slang）

這會強迫模型學到：

- prompt 的語風變化規律
- AI 文的深層特徵

這是學術圈已有的方法，你們做簡化版就能寫成研究貢獻。