

# Solve Minesweeper with LLM

## Proposal

Team ID : 11

### Problem Statement and Task Definition

Large Language Model(LLM)有強大的語言理解能力，並且可以完成許多任務包括文本生成、文章重點整理、情感分析、程式碼生成等。但在邏輯推理上存在一些限制，我們寫藉由踩地雷遊戲來測試 LLM 的邏輯推理能力。

### Description of the challenges

1. 地圖要轉換成 LLM 能看得懂的文字格式
2. 單一 LLM 可能沒有很好的效果，會需要多個合作
3. 因為可能要使用 LLM 的 API，若訓練過程太複雜，會讓成本過高

### Input/Output Behavior with Concrete Examples

#### Input

將踩地雷的遊戲地圖轉換成用字元表示

對於未知的位置標示為 '.'

對於已知的位置，以數字字元標示旁邊八個位置的地雷數量

## Output

回答一個座標，代表選擇的位置

範例輸入：

.....

.....

..1111...

..1001...

..2011...

..111....

.....

.....

.....

範例輸出：(3, 2)

## Related works

### 1. Neural Network Learner for Minesweeper

這篇論文提到踩地雷是一種 Constraint satisfaction problem，並且使用 CNN 進行訓練，與現有的 CSP solver 進行比較。現實生活中也有許多 Constraint satisfaction problem，如計畫安排、時間規劃等。希望透過這份報告，測試 LLM 在這些任務上的表現。

### 2. Assessing Logical Puzzle Solving in Large Language Models: Insights from a Minesweeper Case Study

這篇論文使用 LLM 進行踩地雷遊戲，測試 LLM 讀地圖、解地圖的能力。使用了 few-shot、CoT 等方式，但並沒有嘗試 fine-tune。我們可以參考他們的實作方式，再進行 fine-tune，或是其他 prompt 的方式。

### 3. Tree of Thoughts: Deliberate Problem Solving with Large Language Models

這篇論文使用 Tree of Thoughts，是 Chain of Thoughts 的變體，實驗中發現 ToT 在 Game of 24 的表現會比 CoT 及其他方式還要好。若時間及成本允許，我們的報告中也可以往這個方向嘗試。

# Methodology

## 1. In-context learning

我們希望設計不同的 prompt，並且嘗試 zero-shot、one-shot、few-shot 等方式，讓 LLM 能夠更好的理解地圖，並得到更高的正確率。

## 2. fine tune

我們會提供地圖以及當前的最佳解，並且進行 fine-tune，與其他方式進行正確率的比較。

## 3. chain of thought

將 LLM 產生的回覆作為輸入，請 LLM 再次確認答案是否合理。可以分別進行有 fine-tune 及沒有 fine-tune 的模型進行測試。

# Evaluation Metrics

## 1. 操作合法性

紀錄不合法操作的次數，但因為不同地圖和操作順序所需的合法操作次數不同，所以以錯誤操作跟合法操作的比例來評估這項的分數，代表平均要錯幾次才能做出合法操作。

## 2. 與最佳解的差距

用來評估模型是否能選擇當下爆炸機率最低的位置，目前想法是計算每次操作選擇的位置的存活機率和當下最高存活機率的比值再取幾何平均。

## 3. 成功率

將地圖分類成不同難度，對於同一個難度的地圖計算成功率。

# Baselines

我們為這次的 project 訂定了三個 baseline：

1. LLM 能夠讀懂地圖，並且比較不同的 prompt 對 LLM 理解地圖能力的影響。
2. LLM 能夠理解踩地雷遊戲的規則，並正確進行遊戲。
3. LLM 能夠成功解出踩地雷地圖。

ChatGPT 測試：<https://reurl.cc/Vz619A>

# Time schedule

由上而下逐個任務進行製作

## 地圖生成(Dataset)

製作地圖生成程式

測試現有踩地雷 Solver/製作踩地雷 Solver

地圖相關數據統計(難易度、炸彈密度等)

## Part1 : In-context learning

測試 LLM 對地圖的理解

測試 LLM 對踩地雷規則的理解

測試 LLM 在踩地雷遊戲的勝率

## Part2 : fine-tune

測試 LLM 對地圖的理解

測試 LLM 對踩地雷規則的理解

測試 LLM 在踩地雷遊戲的勝率

## Part3 : Chain of Thought

測試 LLM 對地圖的理解

測試 LLM 對踩地雷規則的理解

測試 LLM 在踩地雷遊戲的勝率

## Discussion

<https://hackmd.io/@nycu-ai/B1RygE4gR>

<https://hackmd.io/@nycu-ai/rJiPj44g0>

<https://hackmd.io/@nycu-ai/BylC9wqe0>

## Repo

<https://github.com/yyh-603/LLM-minesweeper>