

Project Report: AI 的未來能力與符號理解

一、 AI 的未來能力：自動推演新文字與語言系統

能力內容與應用場景

我認為在未來 20 年內，AI 最具潛力且對人類文明有重大意義的能力是「自動推演新文字或語言系統，並能對應已有語言結構」。這項能力遠超目前的翻譯和語義理解，它要求 AI 能夠從一組完全陌生的、缺乏任何標註的符號序列中，自主推測出其內在的語法、形態學和語義結構。

該能力的最終實現，將使 AI 成為解讀任何未知符號系統的通用工具。其應用場景包括：破譯失傳的古代文字，加速人類對古文明的理解；作為星際通訊的解讀工具，分析來自外星文明的符號序列；以及協助語言學家復原瀕危或已消亡的語言結構。簡而言之，這項能力標誌著 AI 從「模仿語言」進化為「理解符號與邏輯」的里程碑。

二、 所需的成分與資源 (Ingredients)

要實現「自動推演新語言」的能力，需要多種先進技術的深度整合：

1. 資料 (Data)

實現此能力需要兩種核心數據：

首先是異構符號序列，即大量的未知文字樣本、亂序語句以及帶有詞形變化的序列，用以訓練 AI 進行非監督式語法歸納。

其次是多模態「世界觀測」數據，包含視覺、聽覺和環境互動的序列數據，這些數據模擬了 AI 對現實世界的觀測。這是為後續強化學習提供概念錨定，將抽象的符號結構（如句子的主語位置）與環境中的具體實體（如施事者 Agent）連結起來的關鍵。

2. 工具 (Tools)

核心計算工具包括：

- 自注意力結構 (Transformer): 作為序列處理的核心，用於高效捕捉符號序列中遠距離的語法依存關係，例如句子開頭的主詞如何觸發句子後段動詞的形態變化。
- 非監督形態學分析工具 (Unsupervised Morphological Analysis): 這

是處理詞形變化的關鍵前提。AI 必須能夠自主識別符號中的詞幹（核心語義）和詞綴（語法標記），才能正確地將變形後的詞彙歸類到同一潛在語法角色。

- 符號推理 (Symbolic Reasoning) 組件：負責將 AI 推導出的語法規則抽象化，並進行邏輯結構驗證。

3. 學習架構 (Learning Setup)

這項任務不能單靠單一學習範式完成，必須結合以下兩種：

- 非監督式學習 (Unsupervised Learning): 這是語法結構歸納的第一步。透過 Masked Language Modeling 等方式，在無標籤情況下，從未知數據中尋找符號序列的內在統計一致性與結構規律。
- 強化學習 (Reinforcement Learning, RL): 這是實現語義對齊的決定性步驟。AI 將被置於一個抽象世界模型環境中，透過實驗其推導出的語法結構（例如，嘗試將 Pos1 符號視為命令），並根據環境的反饋（命令是否成功執行）獲得獎勵。這是將抽象符號與具體世界概念聯繫起來的唯一途徑。

三、 涉及的機器學習類型

該能力的實現主要涉及 非監督式學習 與 強化學習 的緊密組合。

非監督學習是必要的，因為新的語言數據天然缺乏標籤，其目標訊號是符號序列之間的內在一致性和結構的可預測性。強化學習則用於實現語義對齊，即將非監督學習推導出的潛在語法角色 (Pos1, Pos2 狀態) 映射到真實世界的概念 (Agent, Action)，其回饋訊號來自於與世界模型環境的互動。

四、 第一步的「可實作模型問題 (Solvable Model Problem)」

問題設計：非監督序列中的遠距離依存語法歸納

作為邁向最終目標的第一步，我們設計了一個簡化的模型問題：讓 AI 能夠從一個包含隨機詞彙、隨機語序、主詞省略和特定詞形變化的亂序句子中，推斷出每個詞彙的潛在語法角色 (S, V, O)。

這個簡化問題模擬了最終目標中最困難的結構歸納和形態學分析階段。輸入是索引化的隨機詞彙序列（長度為 2 或 3），輸出是每個詞彙的真實潛在語法角

色 (S, V, 0)。

模型與方法： Simple Transformer Tagger

我們選擇 PyTorch 框架實現的簡化 Transformer Encoder 作為模型。理由是 Transformer 的自注意力機制能有效地捕捉我們數據中設定的遠距離依存關係（例如，某個主詞 S 觸發動詞 V 的形態變形，即使它們相隔很遠）。

實作與結果

我們生成了**三組獨立的 CSV 數據集**，每組數據具有完全不同的隨機語法規則（例如 VOS、OSV 等）和隨機詞彙。訓練結果顯示，模型在每組數據上都能達到 97% 以上的詞彙角色標註準確率。

範例分析：

在一個以 VOS 為規則且主詞 'lof' 觸發動詞變形 (V + 'ish') 的實驗中：

- 輸入句子： zorish ie zul
- 模型預測： V, S, 0

即使 V 詞 (zorish) 帶有不影響核心語義的詞綴，且 S 和 V 的語序是隨機的， Transformer 仍能準確標註。

我們確保了每次的生成數據都是隨機的語言邏輯排列與隨機的語言辭彙生成，確保模型的準確率不是因為某種特別的數據，而是對於普遍性的隨機語言都能生效。

討論

這個簡化問題證實了： Transformer 模型具備強大的泛化能力，能夠有效地歸納任意隨機語法結構。

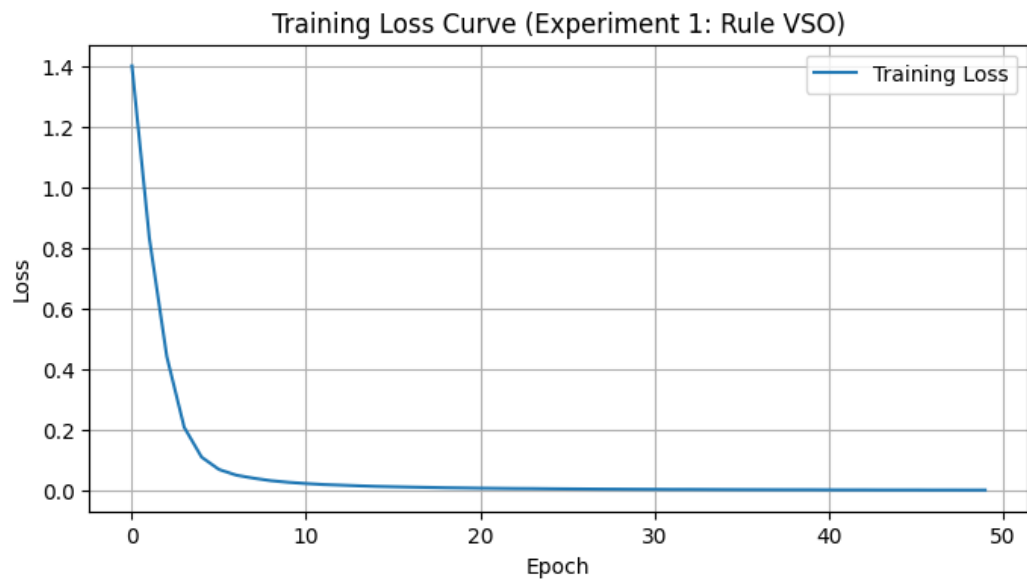
但它也揭示了最大的困難點：「符號到概念的語義鴻溝」。模型雖然學會了將 zorish 標註為 PosV，但它不知道 PosV 在現實世界中代表「動作」。這最終確認了我們設計的學習架構的必要性：結構歸納 (Transformer) 必須結合概念錨定 (RL)，才能使 AI 真正從符號處理者進化為符號理解者。

附圖：

--- 載入 [實驗 1] ---
檔案: unknown_language_data_Exp1.csv
真實語法規則: VSO
詞形變化: 'ede' 觸發 V + 'ay'

--- 模型訓練開始 (Transformer Tagger) ---
Epoch [10/50], Loss: 0.0274
Epoch [20/50], Loss: 0.0090
Epoch [30/50], Loss: 0.0045
Epoch [40/50], Loss: 0.0029
Epoch [50/50], Loss: 0.0019

--- 訓練結果 ---
訓練準確率 (Token Accuracy): 1.0000
驗證準確率 (Token Accuracy): 1.0000



--- 預測範例分析 ---
真實語法規則: VSO
詞形變化: 'ede' 觸發 V + 'ay'

輸入句子: pukay ede ufu
真實標籤: ['V', 'S', 'O']
模型預測: ['V', 'S', 'O']
* 包含遠距離依存的詞形變化 (V 詞尾有變形)

輸入句子: tev ur
真實標籤: ['V', 'O']
模型預測: ['V', 'O']

輸入句子: tev vis ur
真實標籤: ['V', 'S', 'O']
模型預測: ['V', 'S', 'O']

--- 載入 [實驗 2] ---

檔案: unknown_language_data_Exp2.csv

真實語法規則: VOS

詞形變化: 'igi' 觸發 V + 'ay'

--- 模型訓練開始 (Transformer Tagger) ---

Epoch [10/50], Loss: 0.0189

Epoch [20/50], Loss: 0.0063

Epoch [30/50], Loss: 0.0033

Epoch [40/50], Loss: 0.0021

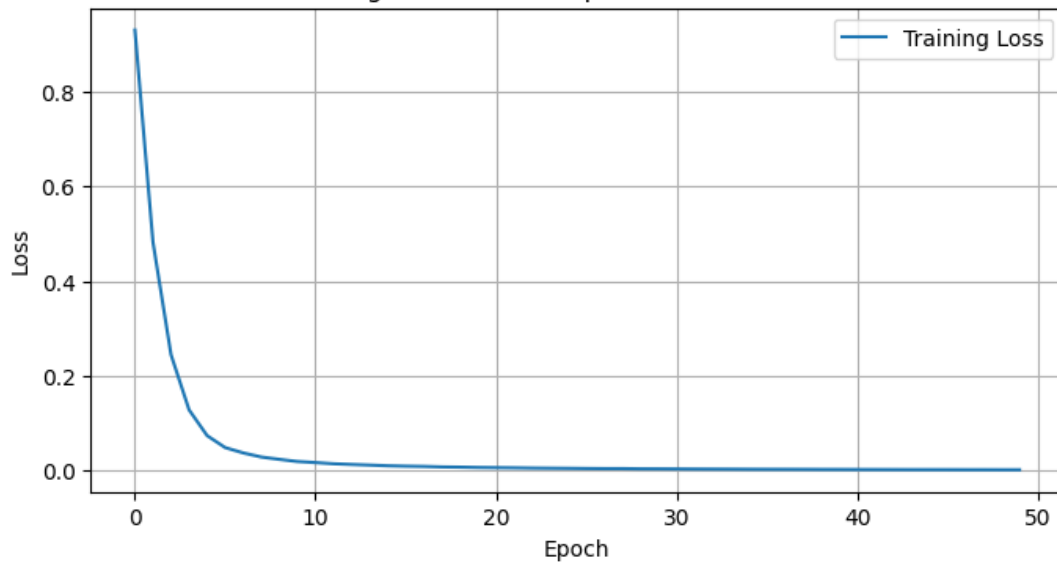
Epoch [50/50], Loss: 0.0014

--- 訓練結果 ---

訓練準確率 (Token Accuracy): 1.0000

驗證準確率 (Token Accuracy): 1.0000

Training Loss Curve (Experiment 2: Rule VOS)



--- 預測範例分析 ---

真實語法規則: VOS

詞形變化: 'igi' 觸發 V + 'ay'

輸入句子: um ge va

真實標籤: ['V', 'O', 'S']

模型預測: ['V', 'O', 'S']

輸入句子: um vo va

真實標籤: ['V', 'O', 'S']

模型預測: ['V', 'O', 'S']

輸入句子: zo map va

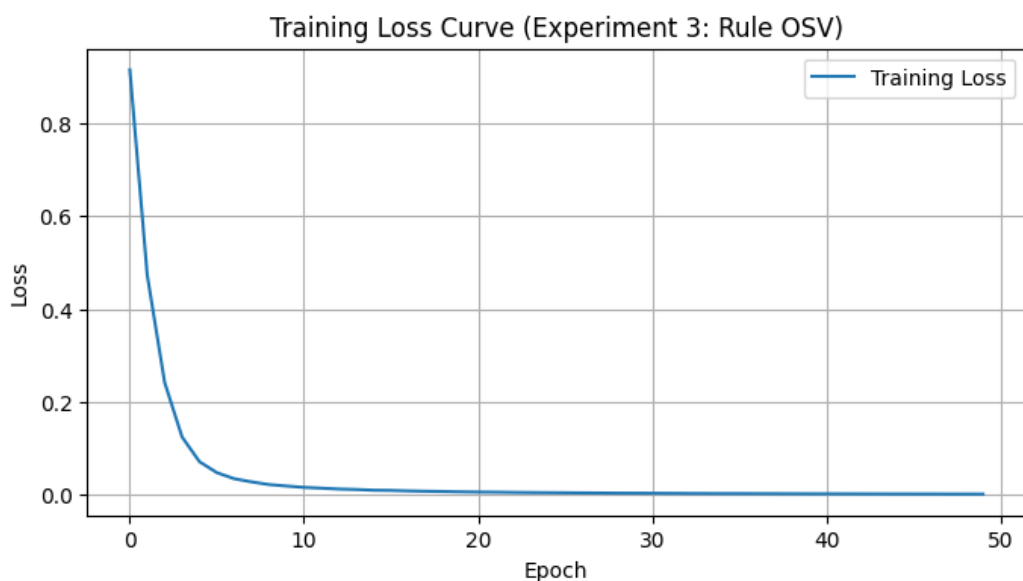
真實標籤: ['V', 'O', 'S']

模型預測: ['V', 'O', 'S']

```
--- 載入 [實驗 3] ---  
檔案: unknown_language_data_Exp3.csv  
真實語法規則: OSV  
詞形變化: 'ut' 觸發 V + 'u1'
```

```
--- 模型訓練開始 (Transformer Tagger) ---  
Epoch [10/50], Loss: 0.0187  
Epoch [20/50], Loss: 0.0062  
Epoch [30/50], Loss: 0.0033  
Epoch [40/50], Loss: 0.0020  
Epoch [50/50], Loss: 0.0014
```

```
--- 訓練結果 ---  
訓練準確率 (Token Accuracy): 1.0000  
驗證準確率 (Token Accuracy): 1.0000
```



```
--- 預測範例分析 ---  
真實語法規則: OSV  
詞形變化: 'ut' 觸發 V + 'u1'
```

```
-----  
輸入句子: dan ut kaul  
真實標籤: ['O', 'S', 'V']  
模型預測: ['O', 'S', 'V']  
* 包含遠距離依存的詞形變化 (V 詞尾有變形)
```

```
-----  
輸入句子: fil ut kaul  
真實標籤: ['O', 'S', 'V']  
模型預測: ['O', 'S', 'V']  
* 包含遠距離依存的詞形變化 (V 詞尾有變形)
```

```
-----  
輸入句子: dan ono ka  
真實標籤: ['O', 'S', 'V']  
模型預測: ['O', 'S', 'V']  
-----
```

程式碼與使用方法:附在 [github](#) 其他檔案中