

以「語言學理論」為基礎用「非機率模型」建立的數學應用問題
作答系統

A Chinese Math Word Problem Solving System Based on Linguistic
Theory and Non-statistical Approach

王文傑 Wen-jet Wang
卓騰語言科技
Droidtown Linguistic Tech. Co. Ltd.
peter.w@droidtown.co

陳加容 Chia-Jung Chen
卓騰語言科技
Droidtown Linguistic Tech. Co. Ltd.
éclair.c@droidtown.co

賴建豫 Chien-yu Lai
卓騰語言科技
Droidtown Linguistic Tech. Co. Ltd.
keith.l@droidtown.co

李家名 Chia-ming Lee
卓騰語言科技
Droidtown Linguistic Tech. Co. Ltd.
trueming.l@droidtown.co

林信宏 Hsin-hung Lin
卓騰語言科技
Droidtown Linguistic Tech. Co. Ltd.
oliver.l@droidtown.co

摘要

本論文介紹一個基於語言學知識而非機率模型或類神經網路機器學習方法來實作的「中文數學應用問題解析」系統。一方面，該系統採納功能語言學學派 (Functional Linguistics) 的方法論，保存了「句式和意義 (form and meaning)」之間的連結，將句型分類做為意

圖偵測之用；另一方面，在逐步求解過程中，則採用形式語意學 (Formal Semantics) 的真值計算方式抽取事件結構中的論元 (argument) 詞彙。最後把與計算目標相關的論元取出後，傳給相應事件的函式進行數值計算。

和統計或機器學習方法相比，本論文提出的「功能-形式混合系統」不只具有極高的應用靈活性，亦能以相當少的資料進行訓練即可取得優良的意圖分類結果。使用本研究建立的模型，對國小一年級加減法及比較類的數學應用問題答題正確率達到 99.29%。

本論文提出的中文數學應用問題解析系統除提供線上操作網頁外，亦將該系統的程式原始碼公開於 Github 專案頁面中。本論文主要貢獻如下：(1) 提出一基於語言學知識解析自然語言中數學運算概念的方法與實作的系統；(2) 以系統實作證明透過適當的中文斷詞處理及 POS/NER 標記後，句型和意義之間的發散度可迅速收斂成為人類可閱讀並理解的邏輯表徵方式；(3) 提供基於台灣小學數學課本中的數學應用問題而編寫的繁體中文測試題庫，並以 MIT 授權釋出。

Abstract

Aside from statistics based and NN machine learning based approaches, this paper presents a Chinese math word problem (CMWP) solving system that is implemented with linguistic reasons. On one hand, the system adopts the functional approach to keep the relation between form and meaning for intent detection. On the other hand, its argument extraction design follows how formal semantics calculate meanings of languages.

The proposed system shows great flexibility with minimal training data requirement. When applying the model to 1st-year elementary level CMWP, the correct rate is between 98.57% and 99.29%. This paper also presents an adjustment procedure to reveal the potentials of the system to improve edging problems.

The proposed hybrid system provides an operational webpage, its source codes are also accessible on Github.com. The main contributions of this paper are listed: (1) It implements a working system that is based on linguistic knowledge to solve CMWP. (2) The system proves that with proper Chinese word segmentation and POS/NER tagging, the divergence between form and meaning can converge to a set of human-readable regular expressions. (3) The CMWP based on Taiwan elementary math textbooks are released under MIT license on Github.com.

關鍵詞：Loki，意圖偵測，數學應用文題，語言學理論，非機率模型方法。

Keywords: Loki, Intent Detection, Math Word Problem, Linguistic Theory, Non-Statistical Approach

一、緒論

自從大數據 (Big Data) 帶動的數據驅動方式興起，許多以數據驅動的統計機率方法以及機器學習方法中，都不再以「語言學方法」做為自然語言處理方法論上的首選。然而，1950 年的語言學至 2020 年之間，語言學已有七十年的發展。即便部份議題仍在研究中，但許多在 1950 年代無法描述的語言內在規則和人類語言能力的普遍表現，在今日的語言學研究中都已得到了廣泛接受的結論。

本研究採用發展自 Chomsky 提出的句子內部結構 X-bar 的觀點 [1]，並以輕動詞 (light verb) 的內部結構 [2] 和中文動詞內部事件結構的結果貌成份 [7] 實作了中文斷詞 (Chinese Word Segmentation, CWS) 及詞性標記 (Part-of-Speech, POS tagging) 的規則與流程，以此建立斷詞與詞性標記工具 - Articut [16]。以 Articut 為基礎，再依 Sinclair & Coulthard 的語言互動模式觀察 [3]，進一步打造建立意圖模型的工具 - Loki (Linguistic Oriented Keyword Interface) [17, 18]。本論文即介紹使用「意圖模型工具 Loki」來處理中文數學應用問題的流程與方法。

傳統的數學應用問題 (Math Word Problem, MWP) 研究都建立在語意的理解上。透過各種自然語言處理 (Natural Language Processing, NLP) 技術進行 MWP 的語意分析及資訊抽取，之後進行邏輯解析，將語意資訊輸出成可操作的數學邏輯結構或訓練成模型。最後再對應到數學公式並計算結果 [14]。由於近年來深度學習的方法和工具發展迅速，騰訊 2017 年嘗試將 End-to-end 模型 [11] 應用在 MWP 的研究中 [12]，走了不同於傳統語意理解的另一條路。本論文以解析數學題目「意圖」的方法來處理 MWP，試圖解析數學題目文字的語意後，再進行數學運算。是屬於語意理解方法的 MWP 研究。

此外，現有的 MWP 成果，不論用語意理解的方式或 End-to-end 模型，都運用了機率模型來訓練資料並預測結果。而本研究從底層的中文斷詞、詞性標記到建立意圖模型，均不使用任何機率模型來預測結果。綜合前述，1. 依循語言學的各種理論（包括 X-bar、語音、構詞、句法、等等規則）來實作 NLP 解析工具，且 2. 不使用機率模型，這是本研究的兩大特色。

以語言規則為基礎來處理 NLP 問題，不需要大量的訓練資料集。可以解決缺乏訓練資

料的問題；不以機率模型預測結果，讓 NLP 的處理結果有絕對的一致性。可解決機率模型預測結果無法解釋、不可追溯，難以除錯等等的問題。

和資訊技術相較之下，MWP 對人類而言是非常容易的問題。但 MWP 對 NLP 和 NLU 技術卻是極大挑戰的一個研究領域。目前所有的 MWP 研究與題庫都僅以國小數學題目為範圍 [12, 15] 都採用 Math23k 的資料集 [13] 做為評估和訓練的樣本。考量到 Math23k 資料集為簡體中文，且許多數學問題的描述文字和台灣的慣用句型有所出入。本研究另外參考國內國小一年級課本與習作內容，編寫了符合台灣的中文數學應用問題描述習慣和句型的題目。本研究已完成國小一年級程度的「加減法」目標函式做為評估和訓練用的樣本，並將模型及原始碼公開於 ArticutAPI Github 專案頁面中 [16]。本研究對此題庫的答題正確率達到 99.29% (commit code: 0a4057a)。

二、Articut 斷詞工具與 Loki 意圖模型工具

（一）Articut 斷詞與詞性標記工具

Articut 是一個商用中文斷詞及詞性、命名實體標記工具 [16]。Articut 依 Chomsky 對句子內部結構的觀點，定義了 X-bar 語言樹狀結構框架 [1]，再將輸入的中文句子由下而上地，透過多組中文詞組構詞原則 [2] 決定詞的邊界。以能最接近句法樹的最高點為輸出結果。

依句法樹的運作原則，一個詞彙被定位在句法樹上的某個節點時，其詞性亦固定下來。因此節點之間的分界，就是詞組的分界；而節點的位置，就標示了詞性的推算結果。此外，Articut 在設計上另外收集了一組用以表示台灣地址、台灣道路名稱、法條索引、網址以及金錢的字串模式 (string pattern) 以及可供使用者動態選用的 WikiData 詞條和政府公開資料中的景點名稱資訊做為外部字典以完成命名實體辨識的工作。

綜合以上流程，Articut 的斷詞、詞性標記以及命名實體辨識是同步完成而無法分割的。其輸出結果除了詞彙邊緣外，亦已隱含句法、句型資訊在內。

（二）Loki 設計原理

Loki [17, 18] 為 Linguistic Oriented Keyword Interface 之字首組合詞。其設計之目的在於透過句型比對，以及挑選語意計算時所需的詞彙做為參數，以便在保有「句型-語意」

的語言表現關係之餘，也能擷取出關鍵詞彙做為計算介面之所需。

Loki 意圖分析工具的架構，是依 Sinclair, J. & Coulthard, R.M. 1975 年在 *Toward an Analysis of Discourse*. [3] 書中所提，注意到的課堂上教師與學生之間的言語互動模式而設計。在該研究中提出的三個層次分別為：

效果 (Act)：例如「教學中與學生互動」、「考試時令學生安靜」...等效果。

功能 (Function)：例如「問與答的互動」、「點名與答有的互動」

實例 (Example)：例如「教師：『ㄅㄨ 四聲？』學生：『快』」

這三個層次在 David Nunan (1993). *Introducing Discourse Analysis*. Penguin Group [4]一書中，被擴充為：

場景 (Discourse) 對應效果 (Act)

對話的語境 (Context) 對應功能 (Function)

對話的實際內容 (Utterance) 對應實體 (Example)

本工具在設計時，使用 NLP 領域較熟悉的詞彙，將對話分成三層，分別是：

專案名稱 (Project)：某組意圖適用的場景。例如在便利商店的場景，具備繳費意圖、購票意圖。但不具備住宿意圖。對應 Nunan 的「場景」。

意圖名稱 (Intent)：某一種意圖。例如在便利商店場中的繳費意圖。對應 Nunan 的語境。

語言表達 (Utterance)：一組可以用來表達某一場景下，某一意圖的語言表達。可以是完整的句子或是不完整的句子。對應 Nunan 的對話的實際內容。

首先，本研究利用 Loki 建立「數學應用問題」做為專案名稱 (Project name)，說明這一組意圖將適用於數學應用問題的語言場景 (Discourse)。接著建立「加減法」的意圖名稱 (Intent name)。在這意圖下，所有的句子都是為了描述「加減法」的語言表達 (Utterance)。例如「爸爸吃掉兩顆蘋果」、「姐姐弄破三張」或「哥哥又給他兩枝筆」...等。

透過 Loki 呼叫 Articut 進行斷詞、詞性標記與命名實體辨識處理後，所有的句子都將轉化為只保留標記做為句式辨識用的正規表式。以「爸爸吃掉兩顆蘋果」為例：完整的模型產生流程為：

原句	爸爸吃掉兩顆蘋果			
↓				
Articut 處理結果	<ENTITY_pronoun>	<ACTION_verb>	<ENTITY_classifier>	<ENTITY_nouny>
	爸爸	吃掉	兩顆	蘋果
	</ENTITY_pronoun>	</ACTION_verb>	</ENTITY_classifier>	</ENTITY_nouny>



Loki	<ENTITY_UserDefined>	<ACTION_verb>	<ENTITY_classifier>	<ENTITY_UserDefined>
產生模型	[^<]*?	[^<不]*?[吃掉][^<不]*?	[^<]*?	[^<]*?
	</ENTITY_UserDefined>	</ACTION_verb>	</ENTITY_classifier>	</ENTITY_UserDefined>

在 Loki 將訓練句的「爸爸吃掉兩顆蘋果」轉寫為：

<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<不]*?[吃掉][^<不]*?</ACTION_verb>)(<VerbP>[^<不]*?[吃掉][^<不]*?</VerbP>))<ENTITY_classifier>[^<]*?</ENTITY_classifier><ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>

的正規表示式同時，亦將動詞轉寫為以方括號標記的 [吃掉]，即可擴充兼容任何以「掉」為結尾的動詞，或是包含「吃」的動詞與動詞組來表示「減少」的意圖，同時把否定表述的「不」會造成的反向語意也予以排除。如此設計，便能用極少的資料，透過保留語言表達 (Utterance) 句型的方式來區分語意意圖，獲得最大的兼容性。

三、研究方法與流程

本研究將 Loki 意圖分類工具操作流程分成四部份來達成解數學應用問題的目標：

1. MWP 文字的斷詞及詞性標記預處理 (Articut)
2. 完成預處理後，建立 Loki 意圖模型 Loki_Math.atm
3. 設計數學運算的函式（本研究以「加法事件」和「減法事件」來處理加減法）
4. 使用 Loki_Math.atm 進行中文數學應用問題的解題

其中流程 1 的預處理，在 Loki 意圖分析工具中會自行於後台處理。

（一）建立 Loki 意圖模型 Loki_Math.atm

Loki 的架構分成專案名稱 (Project)、意圖名稱 (Intent) 和語言表達 (Utterance) 三層。

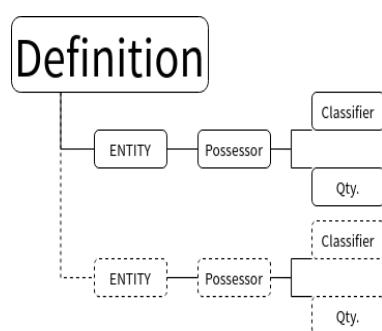
本研究建立一專案名為「國小數學」，接著透過將數學題目中的各段文字定義為三個主要意圖，分別為：

- a. 用來表示「定義語境場景 (Definition)」的意圖
- b. 用來描述「計算過程 (Calculation)」的意圖
- c. 用來說明「求解目標 (Question)」的意圖

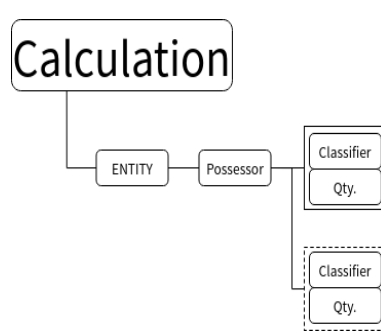
以例題：「桌上有三顆蘋果，小明吃掉一顆，現在還有幾顆蘋果？」來說明，語境場景是「桌上有三顆蘋果」，計算過程是「小明吃掉一顆」，求解目標是「還有幾顆蘋果？」

本研究利用參考國小一年級數學課本及習作題目，另行編寫的應用問題題目 [16] 做為建立意圖模型使用。將資料集裡，所有用以表示定義語境場景的句子，例如「小明有兩枝鉛筆」或「桌上有一顆蘋果」，輸入定義語境場景 (Definition) 中建立相關意圖 (Intent) 模型；再將資料集裡用來表示計算過程的句子，例如「爸爸再給她兩瓶」或「姐姐折斷了一支」...等句子，輸入計算過程 (Calculation) 的意圖中；最後將說明求解目標的句子，例如「請問池子裡有幾隻烏龜」或「妹妹剩下多少塊蛋糕」輸入求解目標 (Question) 中建立 Loki_Math.atm 模型。有了 Loki 意圖模型，符合描述場景句型的文字，就會被 Loki 分類為 Definition 意圖。以此類推，「媽媽再給他三個」就會被分類為 Calculation 意圖，而「有幾個蘋果」則是 Question 意圖。

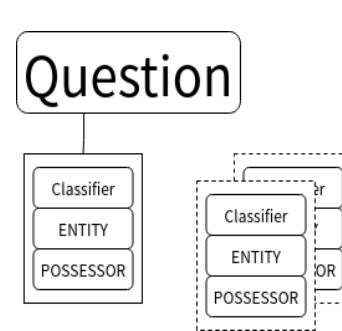
題目文字轉成 Definition, Calculation, Question 三種意圖的資料結構如下圖一、二、三：



圖一、Definition 意圖資料結構



圖二、Calculation 意圖資料結構



圖三、Question 意圖資料結構

在定義語境場景 (Definition) 的意圖中，依句型取出其實體名稱 (Entity)、物體持有人 (Possessor)、分類/量詞 (Classifier) 以及其數量 (Quantity)。以若無，則儲存空字串。例題中的「蘋果」是 Entity，「桌上有」是 Possessor，「顆」是 Classifier，而「一」是 Quantity。若在同一題目中有組定義，例如「姐姐有三張黑紙，妹妹有五張白紙」，則儲存多組定義。

在計算過程 (Calculation) 的意圖中，依句型取出其實體名稱 (Entity)、物體持有人 (Possessor)、分類/量詞 (Classifier) 以及其數量 (Quantity)。若在同一題目中有組計算過程，例如「姐姐早上搞丟三枝筆，下午去買了五枝」，則儲存多組過程以記錄發生順序。

在求解目標 (Question) 的意圖中，依句型取出其實體名稱 (Entity)、物體持有人 (Possessor)、分類/量詞 (Classifier) 以及其數量 (Quantity)。若在同一題目中有組計算

過程，例如「姐姐總共吃了幾塊蛋糕，妹妹剩下多少蛋糕」，則儲存多組求解目標以待稍後依序求解時使用。

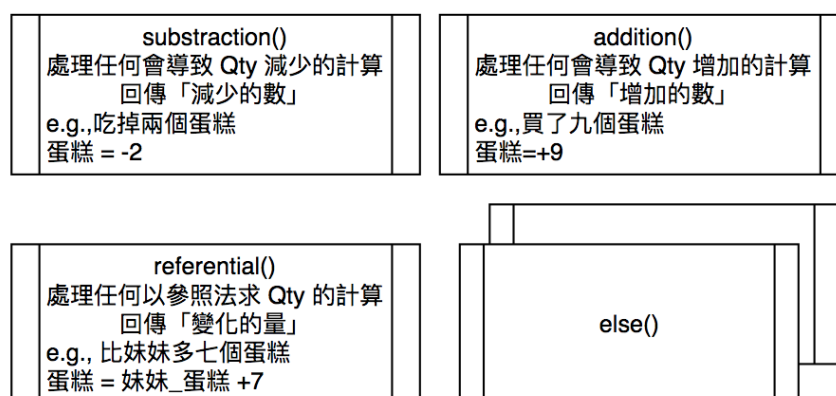
模型建立後，Loki 系統會將產生的 `Loki_Math.atm` 儲存於雲端，同時產生用以表示每一個「句型 - 意義」連結關係以及自動取出該句型中可供計算的「詞彙元素 (e.g., 兩顆，蘋果)」的程式碼。

(二) 設計「加法事件」和「減法事件」的函式

基於意圖模型，我們有了可運算的詞彙元素 (e.g., 蘋果)、計量單位 (e.g., 顆) 和數字 (e.g., 二)。但欲處理「加法」和「減法」兩種計算方式，則需要定義「加法」和「減法」兩種事件的函式，以便處理當句子裡出現諸如「弟弟有 X 顆橘子，再給他 Y 顆」的 $X+Y$ 以及「哥哥有 X 枝鉛筆，借給弟弟 Y 支」時的 $X-Y$ 的「加法事件」和「減法事件」的需求。

不同的數學計算，例如加、減、乘、除、集合運算等等，需要定義不同的事件計算函式。不同的 MWP 研究對數學題目（事件函式）的分類略有不同。本論文以加法和減法為例進行說明。

在定義事件函式這個步驟中，本研究將運算的事件函式定義於事件池 (Event Pool) 中，參見圖四。各函式內有「動詞、句型組合」和「數學運算子及運算公式」的對照規則。計算意圖的文字，經過配合事件的比對，即產生該計算意圖文字所應進行的數學運算。



圖四、事件池 (Event Poll)

從「文字」到「數學邏輯」的轉換工作，本論文和現有研究 [19] 最大的不同有兩點。一是本研究不以「資料對齊算式」以求其數學意義的方式進行訓練，而是以「資料比對

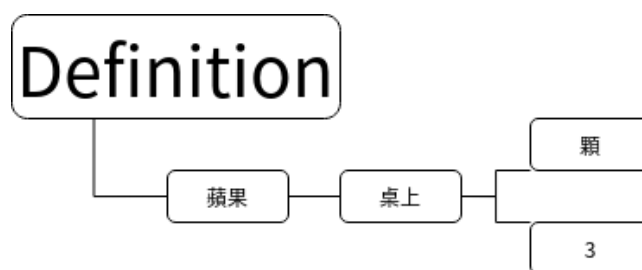
句型」以解析其數學意義的方式進行訓練；二是本研究不需依每一個動詞都重新設計一次事件框架。例如「給予」事件的框架是加法、「贈予」事件的框架也是加法、「購買」事件的框架也是加法...等。本研究直接將「加法事件」設計為一個獨立的函式，透過 Loki 將事件轉譯為正規表示式時，即能表現「在某些句型下，正規表示式可透過 (贈予|給予|購買) 的標記方式來說明這些句子的語意都是表示加法事件」的功能。如此一來，便能依不同應用的需求，只需設計最小需求的函式，置於事件池 (event pool) 中，供後續使用即可。而不需要一開始就依不同的動詞事件設計不同的框架。

(三) 使用 Loki_Math.atm 的中文數學應用問題解題系統

本論文是少數提供系統展示的 MWP 研究 [18]。本研究的系統流程，在中文數學應用問題載入後，系統先將題目中的句子一次一句送出至雲端的 Loki_Math 專案，Loki_Math 專案將句子經 Articut 處理後含有 POS/NER 標記的結果字串，再轉譯為可表示意圖的句型字串。經比對過該專案下所有的意圖內包含的句型字串後，將比對成功的意圖和句型一次回傳。

取得回傳的意圖和句型後，即能依本機程式中的句型正規表示式取出該句型中可做為語意計算論元的詞彙元素，依不同的句型表達的語意，區分事件是「加法事件」或「減法事件」，分別呼叫事件池中的函式，並將取出的論元輸入進行計算，即能求解。

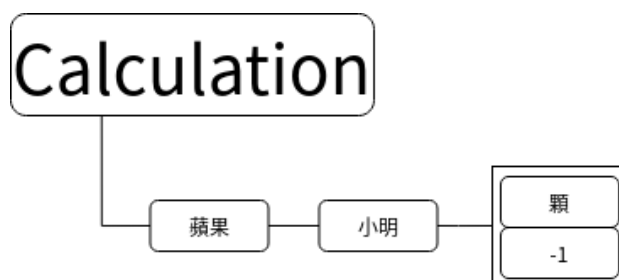
以「桌上有三顆蘋果，小明吃掉一顆，還剩下幾顆蘋果」的題目為例。依前述流程，系統將依次送出「桌上有三顆蘋果」、「小明吃掉一顆」和「還剩下幾顆蘋果」三個句子。在雲端經比對 Loki_Math.atm 專案後，「桌上有三顆蘋果」將回傳「定義語境場景 (Definition)」的意圖，以及句型中可供計算的詞彙單位為「桌上 (Possessor)」、「蘋果 (Entity)」和「三 (Quantity) 顆 (Classifier)」等三個論元。如下圖五所示：



圖五、Definition 架構儲存實際資料示意圖

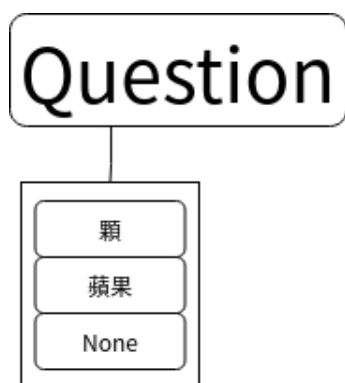
第二句「小明吃掉一顆」將回傳「計算過程 (Calculation)」的意圖，故句型中可供計算

的詞彙單位為「小明 (Possessor)」、「蘋果 (Entity)」、「一 (Quantity) 顆 (Classifier)」等論元，因「吃掉」為「減法事件」，故「一」加上負號成為“-1”，儲存如下圖六所示：

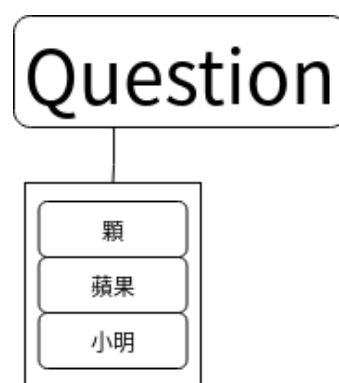


圖六、Calculation 架構儲存實際資料示意圖

最後一句「還剩下幾顆蘋果」將從雲端的 Loki_Math.atm 專案得到「求解目標 (Question)」的意圖回傳。詢問的目標是「(幾) 顆」，詢問的實體則是「蘋果」。因句子中沒有提到持有者，故將該欄位留空。如下圖七所示：



圖七、Question 架構儲存實際資料示意圖



圖八、Question 架構儲存實際資料示意圖

依求解指示，要詢問的實體是「蘋果」，故從「定義語境場景」開始進行計算，得到「蘋果 = 3 顆」的初始定義。接著進入計算過程，得到「蘋果 -1」的計算過程。因沒有其它的計算過程，故得到 $3-1=2$ ，還剩下 2 顆蘋果的最終答案。

若題目不是問「還剩下幾顆蘋果」，而是問「小明總共吃掉幾顆蘋果」，則「求解目標 (Question)」會是如圖八的結果：

則計算過程中，因為在「定義語境場景 (Definition)」的資料中不存在「小明持有蘋果」的記錄，因此會略過定義語境場景，直接進入「計算過程 (Calculation)」的資料中，取得「小明持有蘋果數為 -1」的記錄。再因為「吃掉」是一「減法事件」，因此取得的數值要再加上負號，得到“-(-1)”為“1”。故小明總共吃掉「1 顆蘋果」的最終答案。

四、資料及與實驗結果

基於三項考量，本研究沒有採用常見的 Math23k 資料集：

1. 範圍差異：Math23k 的程度包含國小全年級，超過本研究設定的國小一年級目標。
2. 解答標準：Math23k 的題目是中國普通話的譯文，內容有部份詞彙和台灣使用的國語不同。
3. 打字錯誤：Math23k 的題目中有部份句子內容繕打錯誤。

為避免前述因素干擾，本研究參考台灣的國小一年級數學課本和習作題目，另行編輯 140 題摻雜以「加減法」計算數量和透過「比大小」推估數量兩種意圖的題目。

在實驗過程中，本研究的事件池中只定義了「加法事件」(例如「媽媽又買給他兩枝」)、「減法事件」(例如「早上吃掉一個」) 以及「比大小事件」(例如「蘋果比水梨多兩顆」) 共 3 種事件。

在 140 題個題目中，共有 464 個句子，從中取出 99 句做為訓練 `Loki_math.atm` 模型之用。解答正確的標準有三項：

1. 數值部份需與正確答案一致，例如，若正確解答是「50 箱蘋果」，則計算結果的「50」必須與正確答案一致。
2. 單位部份需與正確答案一致。例如，若正確解答是「50 箱蘋果」，則計算結果的「箱」必須與正確答案一致。
3. 實體部份需正正確答案一致，例如，若正確解答是「50 箱蘋果」，則計算結果的「蘋果」必須與正確答案一致。

只要三個標準中未達到其中一個，就當做沒有正確理解題目語意，視為錯誤。本研究在 140 題中文數學應用問題所使用的 464 個句子中，使用了其中的 99 個句子，即能答對其中的 138 題。

透過逐步反查，可得知模型在遇到「小美有 33 張貼紙，媽媽又給她 5 張，小美現在有幾張？」的題目時，因缺乏「小美現在有幾張」的句型訓練，而只取用了「現在有幾張」的句型，由於題目文字中出現兩個人物 (小美和媽媽)，而「現在有幾張」的句型只能處理題目文字中只有一個人物的題型。因此回覆的答案就錯了。

表一、實驗數據表及實驗程式 Github commit code

	第一次實驗	第二次實驗
--	-------	-------

訓練資料佔全部資料比例 (訓練句數/全部句數)	99 / 464 = 21.33%	100 / 464 = 21.55%
模型正確率 (答對題數/全部題數)	138 / 140 = 98.57%	139 / 140 = 99.29%
ArticutAPI [16] commit code	46edb4f	0a4057a

在新增了「小美現在有幾張」的句型做為訓練材料後，訓練句型的數量從 99 提升到 100 句。而模型的答對題數則從 138 提升到 140 題。兩次實驗也凸顯出透過訓練「句型 - 意義」再取出計算論元的方式，可以輕易溯及錯誤發生原因，並加以調整模型以便容納邊緣問題的特性。實驗結果數據如表一所示。

五、相關研究

Mandal 在 2017 的回顧研究中 [14]，將 MWP 的 NLP 研究分成初期階段 (Early Stage, 1964–2006) 和新世代 (New Era, 2007–2017) 兩個階段。2007 之後的研究因為各種 NLP 和機器學習技術在語意理解上有長足的進步，使得 MWP 可處理的題型範圍擴大，方法上也更多，因此稱為新世代。在 Mandal 的回顧中，所有研究都包括四個共同的步驟：

1. 原文 NLP 前處理：對題目原文進行句法結構及語意的 NLP 處理，抽取所需的語意資訊。
2. 數學邏輯解析：將前處理過的語意資訊，輸出成用數學邏輯形式描述（或理解）的資料結構或建立類似的模型。
3. 產生對應計算方程式：將數學邏輯解析的結果轉成數學等號方程式。
4. 依答案計算結果：有了可計算結果的方程式，經過分析所問的答案後（原文前處理階段），計算結果。

然而，自 2017 至今，因為深度學習演算法蓬勃發展，有研究將端對端 (End-to-end) 模型 [11] 應用於數學應用問題。端對端 (End-to-end) 模型跳脫了上述的四個步驟，直接設計一個序列對序列 (Sequence-to-sequence/End-to-end) 模型 [12]，準備好所有「題目文字」和「對應的數學算式解答」的訓練資料，使用深度學習訓練出解題模型。這樣的方法完全不處理語意的問題，任何語言只要準備好訓練資料，即可直接訓練。騰訊在 2017 就使用 LSTM 架構訓練了中文的數學題庫 Math23k [13] 的 MWP 模型，此模型可達到 58% 的解題正確率 [12]。

由於使用端對端 (End-to-end) 模型跟之前的語意理解的方法不同，因此可以算是初期階段 (Early Stage) 和新世代 (New Era) 之後的第三個階段。

除了這三個發展階段外，從 MWP 難易度來看，目前所有研究都是以國小 (elementary school) 應用數學題目為目標。從語言資料的角度來看，跟其他 NLP 研究一樣，所有 MWP 的研究都是從英文開始。中文的 MWP 則是參考英文的研究成果後，再進行調整而來。Mandal 的研究回顧 [14] 中所整理的研究，都是解英文的 MWP 研究。在針對中文 MWP 的成果上，近期的成果有許聞廉教授的研究 [19]，中研院 CKIP 團隊的研究 [10]、台大 MiuLab 的研究 [15]，以及上述的騰訊研究團隊的成果 [12]。

許聞廉教授進行過一個「小學數學深度理解解題系統 Demo」計畫 [19] 是以各種「動詞語意事件框架 (Frames)」為語意解析的核心。流程上從 MWP「問題句」開始，拉起每個陳述句的關係。再透過各個陳述句對應的框架來建立起整個數學的解題邏輯。本研究則是直接以自然語言本身做為後設語言 (meta language)，因此不是以框架做為模型目標。在實作解析時，成本較低。

表二是在 Mandal 整理的 MWP 四個步驟下，本論文的研究方法，和中研院 CKIP 團隊在 2015 年提出的研究 [10] 的比較。

表二、研究步驟比較

MWP 研究步驟 [14]	本論文研究流程	CKIP 團隊研究流程 [10]
1. 原文 NLP 前處理	Articut 斷詞、詞性標記及命名實體標記	Language Analysis module: Sinica Treebank E-HowNet
2. 數學邏輯解析	Loki 意圖模型 (Definition, Calculation, Question Structures)	Problem Resolution module: Logic Form Structure (LF)
3. 產生對應計算方程式	事件池 (Event Pool)	Problem Resolution module: Inference Engine (IE)
4. 依答案計算結果	依 Question Structure 分析所需結果並計算	Explanation Generation module (Question & Answer Mapping)

CKIP 團隊在 NLP 前處理階段，使用了中研院進行多年的中文句結構樹 (Sinica Treebank) 以及廣義知網 (E-HowNet) 來進行 MWP 題目的語意解析。中文句結構樹包含了 61,087 個中文樹圖及 361,834 個詞 [5, 6]。而廣義知網包含了九萬多個知識條目 [8, 9]。相對於本研究使用的 Articut 斷詞及詞性標記及 Loki 意圖工具，兩者皆以語言學規則建立，僅使用一個 Chomsky 提出的句法樹圖 X-bar，且不需維護龐大的詞典資源。

在「數學邏輯解析」和「產生對應計算方程式」兩個階段，CKIP 的研究均使用機率來

預測最可能的對應結果。本研究在所有階段都不使用任何機率模型，其優點是一致性高且可明確知道不能處理的邊緣問題在哪裡，亦可以持續針對未能處理的 MWP 邊緣問題逐步調整。缺點則和其它基於統計機率或機器學習方法一樣，不易在未知題目中，估計尚需處理的範圍和所需時間。

六、結論

本論文提出了一個基於「語言學知識」而「非機率模型」方法來實作的「中文數學應用問題解析」系統。方法上本研究將數學題目的意圖分成「環境背景 / Definition」、「數學計算 / Calculation」以及「問題 / Question」三種意圖。透過 Loki 建立意圖偵測模型。再搭配事件池，對計算意圖的句子進行數學運算邏輯解析。最後透過問題句子的分析，進行所需的計算並取得答案。

本研究使用台灣小學一年級的題目語料庫，並以加法和減法為範圍實作了本方法的理論，本方法對國小一年級加減法及比較類的數學應用問題答題正確率達到 99.29%。

本論文提出的中文數學應用問題解析系統除提供線上操作網頁外，亦將該系統的程式原始碼公開於 ArticutAPI Github (commit code: 0a4057a) 專案頁面中 [16]。

參考文獻

- [1] Chomsky, N. (1968). Remarks on Nominalization: Linguistics Club, Indiana University.
- [2] Chomsky, N. (1995). The Minimalist Program: Cambridge: MIT.
- [3] John McHardy Sinclair, M. C. (1975). Towards an Analysis of Discourse: Oxford Univ Pr.
- [4] Nunan, D. (1993). Introducing Discourse Analysis: Penguin Group.
- [5] Feng-Yi Chen, P.-F. T., Keh-Jiann Chen, Chu-Ren Hunag. (1999). 中文句結構樹資料庫 (Sinica Treebank) 的構建. IJCLCLP.
- [6] Chen Keh-Jiann, Y.-M. H. (2004). Chinese Treebanks and Grammar Extraction. IJCNLP.
- [7] 王文傑. (2008). 結局與結果：中文的兩種動詞後結果貌成份研究. (碩士). 國立交通大學,

- [8] Wei-Te Chen, S.-C. L., Shu-Ling Huang, You-Shan Chung, Keh-Jiann Chen. (2010). E-HowNet and Automatic Construction of a Lexical Ontology. COLING.
- [9] Shu-Ling Huang, K.-J. C. (2013). Semantic Analysis and Contextual Harmony of Durations. Journal of Chinese Linguistics, 41.
- [10] Yi-Chung Lin, C.-C. L., Kuang-Yi Hsu, Chien-Tsung Huang, Shen-Yun Miao, Wei-Yun Ma, Lun-Wei Ku, Churn-Jung Liao, Keh-Yih Su. (2015). Designing a Tag-Based Statistical Math Word Problem Solver with Reasoning and Explanation. Paper presented at the Computational Linguistics and Chinese language Processing.
- [11] Ian Goodfellow, Y. B., Aaron Courville. (2016). Deep Learning: MIT Press.
- [12] Yan Wang, X. L., Shuming Shi. (2017). Deep Neural Solver for Math Word Problems. Paper presented at the Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark.
- [13] Math 23k Dataset. (2019). Retrieved from https://github.com/ShichaoSun/math_seq2tree
- [14] Sourav Mandal, S. N. (2019). Solving Arithmetic Mathematical Word Problems: A Review and Recent Advancements: Springer Nature Singapore Pte Ltd.
- [15] Ting-Rui Chiang, Y.-N. C. (2019). Semantically-Aligned Equation Generation for Solving and Reasoning Math Word Problems. Paper presented at the Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2019), Minneapolis.
- [16] 卓騰語言科技. (2020). ArticutAPI. Retrieved from <https://github.com/Droidtown/ArticutAPI>
- [17] 卓騰語言科技. (2020). LokiAPI. Retrieved from <https://api.droidtown.co/document/#loki>
- [18] 卓騰語言科技. (2020). Loki 算數學. Retrieved from <https://api.droidtown.co/#lokiMath>
- [19] 許聞廉. (2020). 小學數學深度理解解題系統 Demo. Retrieved from <http://iasl.iis.sinica.edu.tw/hsu/zh/小學數學深度理解解題系統 demo/>