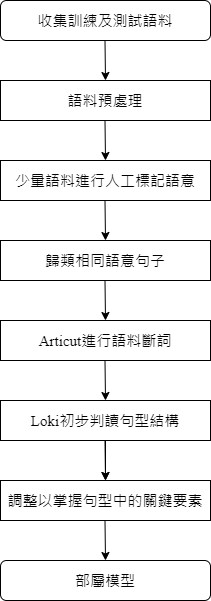
Articut/Loki 的運作細節：以全稱語意為例

1. 建立句型模型流程:



1. 收集訓練及測試語料

以下用全稱語意 (程式中註記為universal\_use) 為例說明運用 Articut/Loki 處理中文 wh 詞詮釋的流程。首先，我們收集中研院平衡語料庫中 (https://asbc.iis.sinica.edu.tw/) 所有包含「誰」的語料，總計2979 筆。

1. 語料預處理

經語料預處理過後抽取出含有「誰」的語料為 2695 筆。

1. 少量語料進行人工標記語意

首先，觀察前100句預處理過的語料，並且根據它們的語義(intent)用自訂代號標記，並且將不合語法的語料和須討論的語料用特殊符號標記。本研究所使用之標記如下:

(1)

a: Interrogative (疑問語義)

b: Existential (存在語義)

c: Universal (全稱語義)

$: 不合語法的語料

?: 不確定/待討論

根據語言學理論，將前一百句預處理過的語料分類到與之對應的語義，例如出現於語料中的「無論誰當權」，根據此句出現的「無論」並且其c-command「誰」，起到全稱語義核可詞的作用，此句屬於 c intent（Universal），就在此句的前面標註一個c做為代號

(2) c 無論誰當權

1. 歸納相同語意句子

其他存在於語料之中的句子，只要出現相應的核可詞「無論」，並且c-command「誰」，就會屬於全稱語義。例如: 「無論誰將是台灣第一任民選總統」、「無論誰會來」。

如此可以將其歸納為一個固定的句型，也就是「無論+誰+V」。接著在 Universal這個 intent 中建「無論誰當權」這個例句，其代表的就是「無論+誰+V」這個句型(pattern)。

1. Articut進行語料斷詞

此處給定之例句為「無論誰當權」。系統在Articut將其表層解析為 (49)，其結構在(50) 以 regex 表示式呈現。

(49) 無論/誰/當權

(50) <MODIFIER>無論</MODIFIER><CLAUSE\_WhoQ>誰</CLAUSE\_WhoQ>

<ACTION\_verb> 當權</ACTION\_verb>

1. Loki初步判斷句型結構

而Loki同樣使用 regex 表示式的概念，而使同一結構的句型能夠被判讀。系統初始判斷此句的結構如(51)所示:

(51)

(<(MODIFIER|ModifierP)>[^<]+</(MODIFIER|ModifierP)>)?<CLAUSE\_WhoQ>[^<]+</CLAUSE\_WhoQ><(ACTION\_verb|VerbP)>當權</(ACTION\_verb|VerbP)>

其中「|」符號意指 "or", 以圖八出現的 "<(MODIFIER|ModifierP)>" 為例，表句中可出現MODIFIER 或是 ModifierP。

而 「[^<]+」 符號表任意字詞，以 <(MODIFIER|ModifierP)>[^<]+

</(MODIFIER|ModifierP)> 為例，「[^<]+」 置於詞性標記中間，意即只要句中出現符合MODIFIER 或是 ModifierP 的任意字詞，此句型都會將其判讀。

再來是將帶有詞性標記的字串置於括弧中，並於後括弧處附加「?」符號，以 (<(MODIFI-ER|ModifierP)>[^<]+</(MODIFIER|ModifierP)>)? 為例，意指整段帶有詞性標記的字串為可有可無。以此句為例，也就是句子中的MODIFIER或是ModifierP不一定需要出現。

1. 調整以掌握句型中的關鍵要素並部屬模型

為使「無論誰當權」此句結構能夠符合本研究主題，需要限縮此 regex 表示式，並同時達到正確判讀同一類的句型之目的。需要修改 regex 表示式的內容，加上原句「無論誰當權」用詞的同義詞、增加可能的結構並且刪減可有可無的結構，以符合此類句型， 形成(52)的regex 表示式。

(52)

(<FUNC\_inter>不論</FUNC\_inter>|<MODIFIER>無論</MODIFIER>)(<AUX>是</AUX>|<ACTION\_verb>[^<]+</ACTION\_verb>)?<CLAUSE\_WhoQ>誰</CLAUSE\_WhoQ>

其中 (<FUNC\_inter>不論</FUNC\_inter>|<MODIFIER>無論</MODIFIER>)，「不論」和「無論」兩個帶詞性標記字串，中間的 「|」 符號意指 "or"，也就是句中可以出現「不論」或者「無論」。例如: 「無論誰將是台灣第一任民選總統」和「不論誰將是台灣第一任民選總統」都可以被此句型判讀

再者，此段 regex 表示式中加上的一段 (<AUX>是</AUX>|<ACTION\_verb>[^<]+</ACTION\_verb>)?，意指助詞「是」或ACTION\_verb不一定需要出現。例如；「無論是誰會來」和「無論見到誰」都在此句型的判讀範圍。

而原本的 <CLAUSE\_WhoQ>[^<]+</CLAUSE\_WhoQ> 意指所有符合 <CLAUSE\_WhoQ> 詞性的任意字詞都會被判讀，在這裡為符合本研究關於中文 wh 詞「誰」的主題，將詞性 <CLAUSE\_WhoQ> 之間由任意字詞「 [^<]+ 」改為僅限字串「誰」。

最後，動詞「當權」並不是本句型必要出現的動詞，故在此regex表示式將其刪減。

部屬經調整Loki中的正規表示式，掌握句型中的關鍵要素，達到以一個句型即可兼容多個句子的效果，最終形成能以少量句型進行大量語料測試的模型。

待前100句都在 Loki 上建好pattern之後，持續人工標記到約前500句，並使用後面已經標註過的短句做測試，並比較人工標註過的結果(正確標記)和Loki所判讀的結果，針對Loki判讀錯誤的句子進行修改，補充Loki漏判的句型，增加至所對應的intent(語意)。

經如此反覆在剩下的句子中一樣進行上述之人工標註和測式，修改與補充，目前模型內含25句疑問語意、7句存在語意、16句全稱語意，共48種中文wh詞「誰」的句型，測試平衡語料庫經預處理共2695 筆含有「誰」的語料，在本研究聚焦之全稱語意的判讀達到95%的準確率。

二、 核可詞效力判斷演算法

使用者調整初步程式

首先，以句子「無論誰當權」為例，在 63 行將此句子加入通用意圖列表 "universal" 中，以表達系統辨識到該句符合此語意的結構。其次，系統以前文示例之修正過的結構 (<FUNC\_inter>不論</FUNC\_inter>|<MODIFIER>無論</MODIFIER>)(<AUX>是</AUX>)?<CLAUSE\_WhoQ>誰</CLAUSE\_WhoQ> 為依據，更新 "intent" 列表，將 "c3" 加入其中。這裡 "c3" 為自定義編號，"c" 表徵全稱語意，"3" 表示全稱語義中的其中一個句子的結構 (pattern)。並且 64 行出現的 "a1", "a5" 和 "a9" 也是自定義編號，"a" 表示疑問語意，"a1", "a5" 和 "a9" 各對應到不同包含強疑問詞的句子結構，分別為以下所示:

(52) 強疑問詞結構：

a1: 誰+sentence-final interrogative particle

a5: 還有+誰

a9: 誰+ conjunction + noun or 誰

(52) 的結構被列為強疑問詞，也就是當以上結構和全稱語義的核可詞同時出現在句子時，檢查器會將其判斷為疑問語意，如圖十二的第 64 和 66 行所示。本研究預計將在此區塊將生成語言學理論實作為程式邏輯以進行大量測試理論完備性。

圖十二：「無論誰當權」Python程式語言執行範本

接著在主程式中填入待測試用的句子，此處以 (53) 三句為例做說明，測試結果如圖十三所示 。系統執行後，如 304 行起所示，在 301 行將每個測試的句子逐一交由 execLoki() 執行，並在302行中名為log\_TEST.txt的檔案中寫入進行測驗結果，計算以判讀是否符合前述的已預先設定之結構。

(53) 測試句：

a. 不論誰將是台灣第一任民選總統

b. 無論是誰會來

c. 還有誰會來

圖十二：(53) 測試句檢驗截圖

再以「不論誰將是台灣第一任民選總統」為例，其結構在圖十三以regex表示式呈現。該句表層結構包含給定例句「無論誰當權」的表層結構。圖十三中的「將」、「台灣」、「第一任」、「民選」和「總統」以句子結構的判讀需求來說，皆為可有可無之「修飾詞」和可替換而不影響結構之「名詞」，故不影響表層結構之判斷。

圖十三：「不論誰將是台灣第一任民選總統」的結構呈現

當表層結構中含有預給定的例句結構後，系統接著進入深層結構之轉譯。以「不論誰將是台灣第一任民選總統」為例，將其表層結構相符指定例句之部份轉為深層結構如(54)，由於系統將「修飾詞」和可替換而不影響結構之「名詞」判讀為皆為可有可無的函式，故可將其簡化為(55)。

(54) ∀x∃y 不論(CLAUSE\_Who(x)) & 總統(y) -> 將(是(y, x) & 民選(y) & 台灣第一任(y))

(55) ∀x (不論(CLAUSE\_WhoQ(x) 是(x))

(55) 除了助動詞「是」之外，與前述之(51)相容。加上在圖九做的更動調整，在「不論」後加入可有可無的助動詞「是」，並且不一定需要動詞的出現，以符合更多相似結構，成為 (56)，以表示該輸入句「不論誰將是台灣第一任民選總統」結構與指定的 utterance「無論誰當權」相符。

(56) if utterance == "無論誰當權":

如此反覆將每個要測驗的句子輸入程式中，即可如圖十四所示得出 (53) 之中每個句子依理論邏輯的計算結果是否相符於人類直觀的判斷結果 (如「是否呈現全稱語意」)，並依此檢驗理論邏輯上之完備性。

圖十四：log\_TEST.txt檔案內(53) 測試句的檢驗結果