# LOKI建模流程

1. 觀察語料並歸納規則，以利於後續將語料程式化

為達到製作檢查器之目的，找出簡短且能代表該種句型的句子，並將句型歸類到不同的語義(intent)，使設計檢查器時，能夠根據所對應到的句型，回傳是屬於哪種語義

e.g., 句型：不論/無論 + (是)誰

例句：不論誰當權/無論誰當權🡪句型一樣只是表達方式不同

並且，屬於此句型的句子都會歸納為「全稱語義」

1. 在 Loki (Linguistic Oriented Keyword Interface 語言導向的關鍵詞介面)的後台Articut預處理wh疑問詞的斷詞(CWS)及詞性標記(POS)，為計算機提供經過分類和標記過後的操作符號，使語言更容易處理。  
     
   Articut的CWS與POS操作範例：



1. 一般來說， Loki 是用來分析句型，而使用者把相同意圖的句型歸類在一起。而此處即將分類成三種語義，分別為中文wh詞中的疑問語義、存在語義和全稱語義。

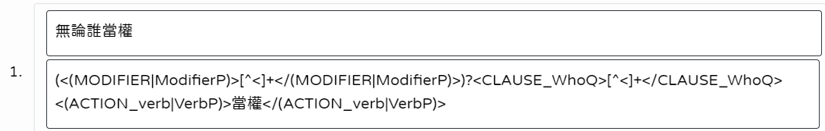
挑出簡短能代表句型的句子，將句子送到送到Loki Advanced之後，Loki Advanced會先將語料進行斷詞/標記，此時斷詞系統（Articut）回傳的結果在Loki Advanced以 regex 來表達。

例如能語料庫中的句子:「無論誰當權」能代表句型：「不論/無論 + (是)誰」，便將其代表其句子歸納在「全稱語義」意圖。

建立一個「全稱語義」意圖並點選此分類:



在「全稱語義」意圖中加入符合此語義的句型:



Loki使用regex表示式的概念，而使同一結構的句型能夠被判讀。這能使各種不同表達方式的測試句送入檢查器時，能夠被所預先設定的句型對應到。

e.g., 無論誰當權

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

其中「|」符號意指 "or", 以regex中出現的 "<(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(<(MODIFIER|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. 在網頁工具上編輯 regex

為使「無論誰當權」此句結構能夠符合中文wh疑問詞之研究主題，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。句型包含的是一個句子的特色，也就是說，有此種特色的句子就必定是此種句型

以句子:無論誰當權，為例:

句型：不論/無論 +(是) 誰

句型包含此句子的特色：句中包含用詞「無論」或是「不論」+(是)「誰」

除了此句子特色不論/無論 + 誰之外，句子剩下的內容為可有可無，也就是不算是此句子的特色，在regex中就要做刪減，以達到精準判斷同類句型的目的。也就是當一測試句出現「無論」或是「不論」+(是)「誰」時，不管此句含有其他什麼內容，都會被判斷為此種句型。

需要做的是修改原regex表示式的內容:

1. 加上原句「無論誰當權」中，「無論」這個用詞的同義詞(e.g., 不論)
2. 刪減可有可無的結構

e.g.,

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

編輯後：

(<FUNC\_inter>不論</FUNC\_inter>|<MODIFIER>無論</MODIFIER>)(<AUX>是</AUX>)?<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」。

而此段regex表示式中加上的一段 (<AUX>是</AUX>)?，意指助詞「是」可有可無

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」。

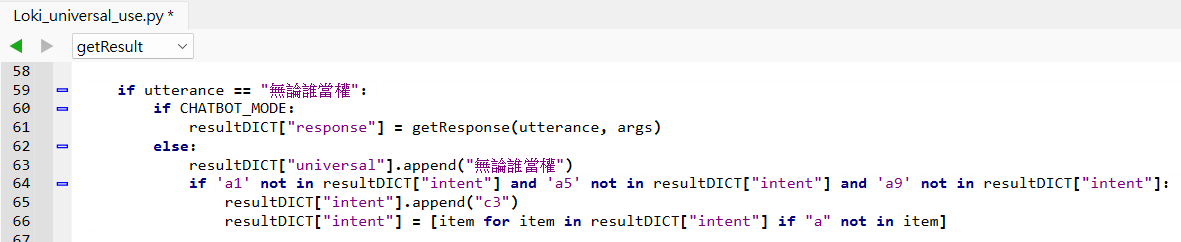
最後，動詞「當權」並不是本句型必要出現的動詞，故在此regex表示式將其刪減。

1. 重複以上步驟到一定程度後在網頁上部署，Loki 就把檢查器模型建好了，使用者在網頁上把模型下載下來，此時該模型內涵的東西就有:
   1. 上面編輯過的句型
   2. execLoki() ：一個把測試句拿去比對所有句型的通道，根據比對到的句型，回傳是屬於哪個 intent (疑問語義)的函式。
2. 原始程式碼只比對 句型跟 intent，若有一些額外的語言學規則變化的話，需要使用者另外撰寫程式判斷是以完善檢查器並達到高正確率。

以wh疑問詞檢查器的情況來說，需要判斷式的情況通常是當句子符合兩個屬於不同語義的句型時。

e.g.,

* 1. 比對到含有強疑問詞的句型時，句子必定是疑問語義，如此就算比對到另外兩個屬於存在、全稱語義的句型，也不納入考慮，直接取疑問語義作為最終結果。
  2. 無強疑問詞和其他兩個屬於存在、全稱語義的句型同時出現時，以這兩個intent為主，取存在或全稱語義。



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" (全稱語義)中，以表達系統辨識到該句符合此語義的結構。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，作為檢查器的判斷依據之一。

這裡"c3"為為使程式碼簡短清晰的自定義句型編號，"c"表徵全稱語義，"3"表示全稱語義中的其中一個句子的結構。並且64行出現的"a1", "a5"和"a9"也是自定義編號，"a"表示疑問語義，"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:

a1: 誰+sentence-final interrogative particle

a5: 還有+誰

a9: 誰+ conjunction + noun or 誰

以上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義，如圖的第64和 66行所示:

64行:if 'a1' not in resultDICT["intent"] and 'a5' not in resultDICT["intent"] and 'a9' not in resultDICT["intent"]

如果在 "intent" 列表中找不到 "a1", "a5"和"a9"這三個元素，則進入這個條件判斷。代表如果"a1", "a5"和"a9"未出現在比對出的intent列表中，也就是強疑問詞未在測試句中出現，才進入以下的判斷式。

66行:item for item in resultDICT["intent"] if "a" not in item: 遍歷"intent" 列表中的每個元素，並將元素指定為 “item”，只有當元素中不包含字母 "a" 時，才將該元素加入到新的intent列表中。也就是無強疑問詞出現的條件下，將 "intent" 列表中包含字母 "a" 的元素(其他弱疑問詞，不影響存在、全稱語義)過濾掉。

而以測試句: 「無論誰當權」為例:  
  
程式到第64行時，在intent列表中包含c3(全稱語義句型)，而不包含"a1", "a5"和"a9"這三個元素，進入了下面的判斷式。  
  
而程式在第66行的時候，因為intent列表中也找不到其他含”a”(疑問語義代號)的元素，沒有含”a”的元素需要過濾，intent列表中保持只含元素”c3”。

所以檢查器判斷出的最終結果，就是包含c3這個句型的全稱語義intent。

這樣基本就是一個完整的模型了。