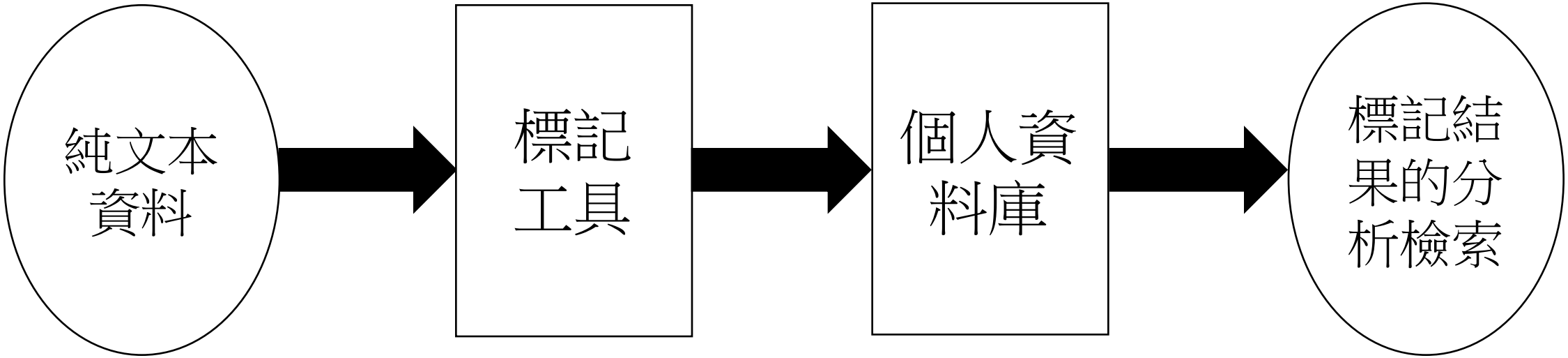
從純文本資料經標記工具到形成個人資料庫的一套方便建構流程

以STAML(Simple Text Annotation Markup Language)格式作為中介格式為例

曹又霖

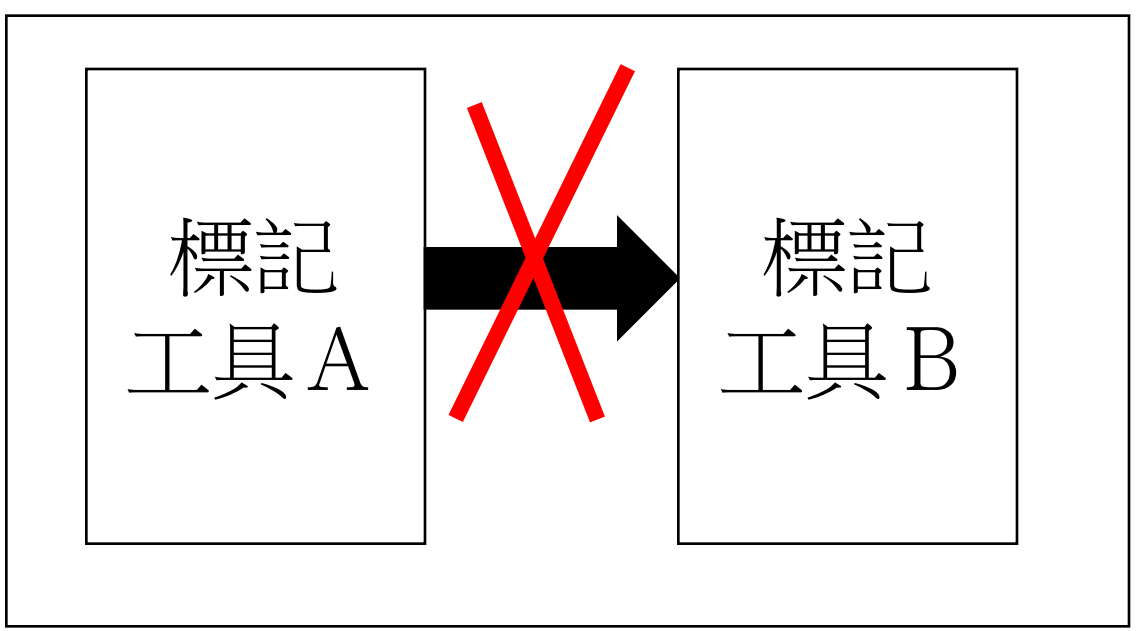
目的

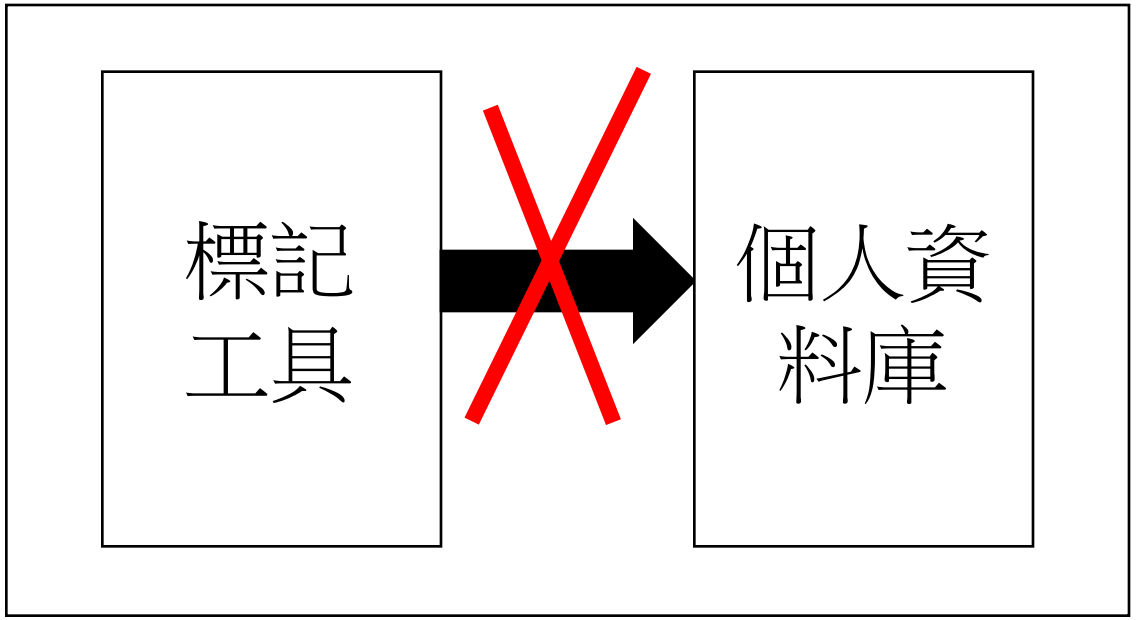
- 將純文本資料經由標記工具標註後可以分析出其文本中各種不同性質詞彙的位置。
- 利用這個分析結果可以將資料建構成個人資料庫以利後續對於此文本的檢索分析、詞頻分析。



問題

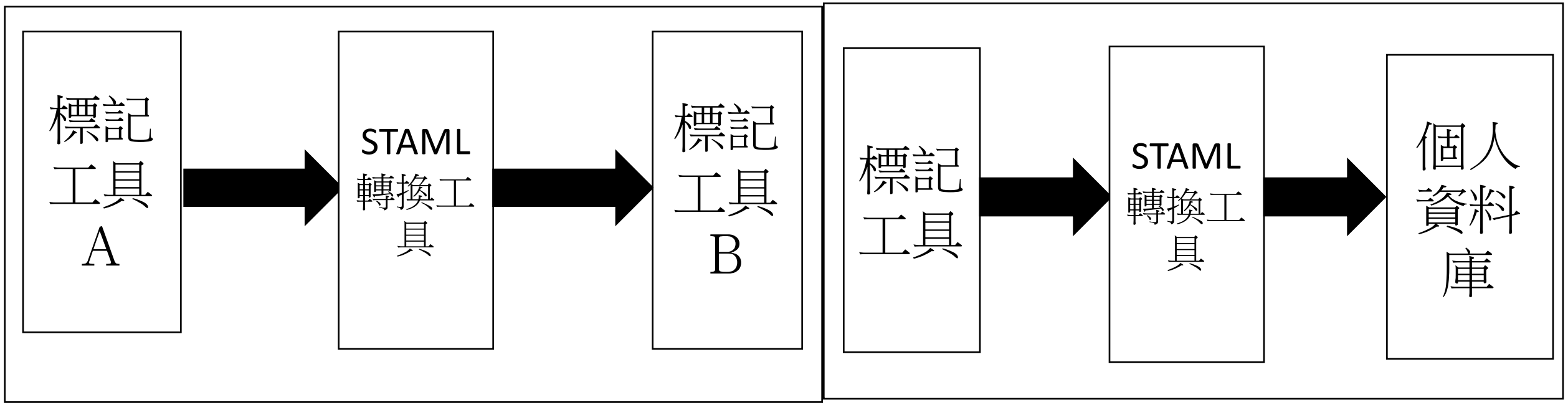
- 問題主要有兩個：
  1. 各個標記工具所使用之格式不同，無法直接在使用者不知道其格式之情況下進行工具與工具之間的連接。
  2. 標記工具所使用之格式與建構資料庫的格式也不相同，無法直接在使用者不知道其格式之情況下將標記之結果直接建構成資料庫。





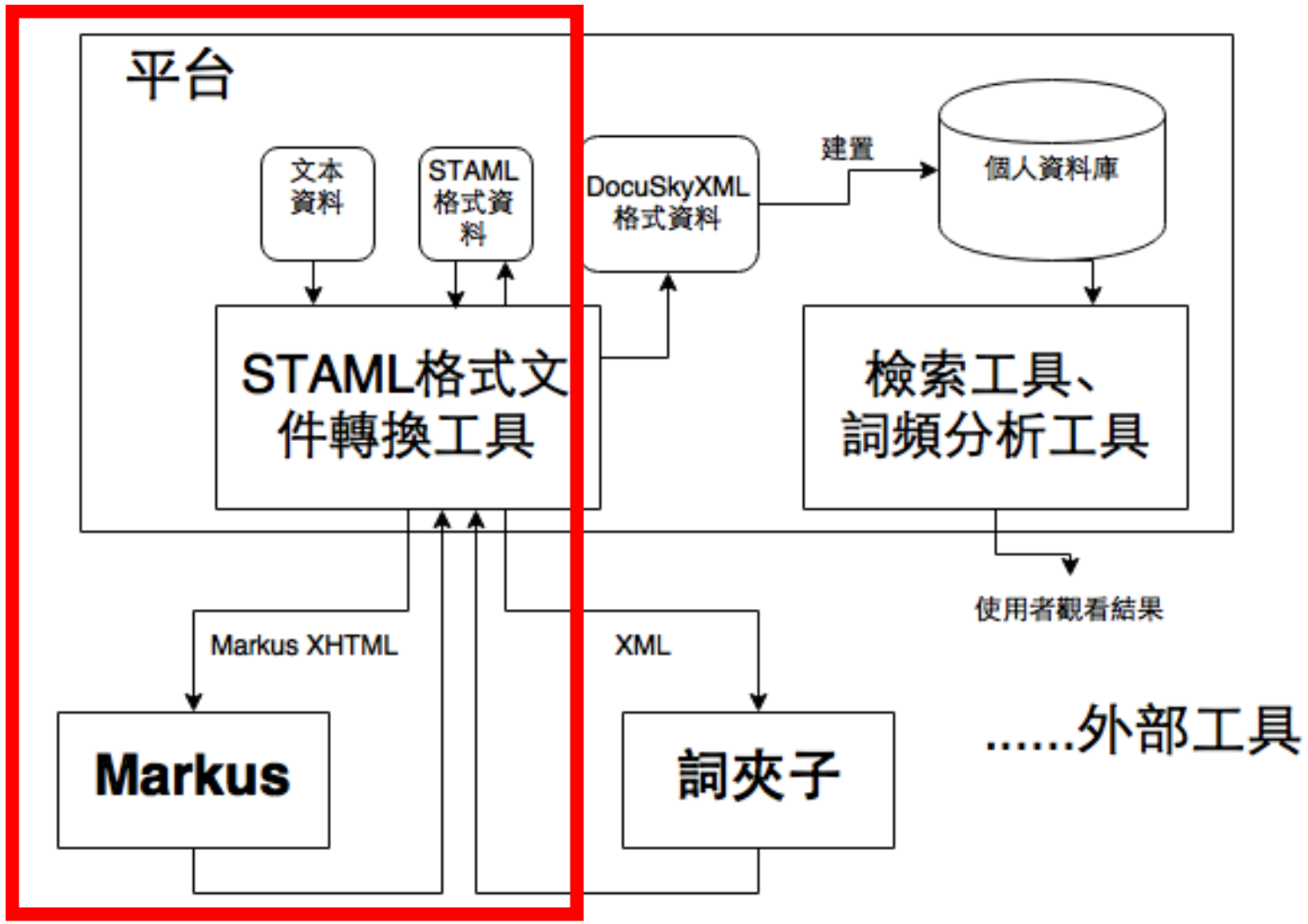
解決方法

- 我們提出一種新的中介格式STAML(Simple Text Annotation Markup Language)與資料轉換的工具藉以作為銜接的橋樑。



系統架構與流程

目前完成的部分

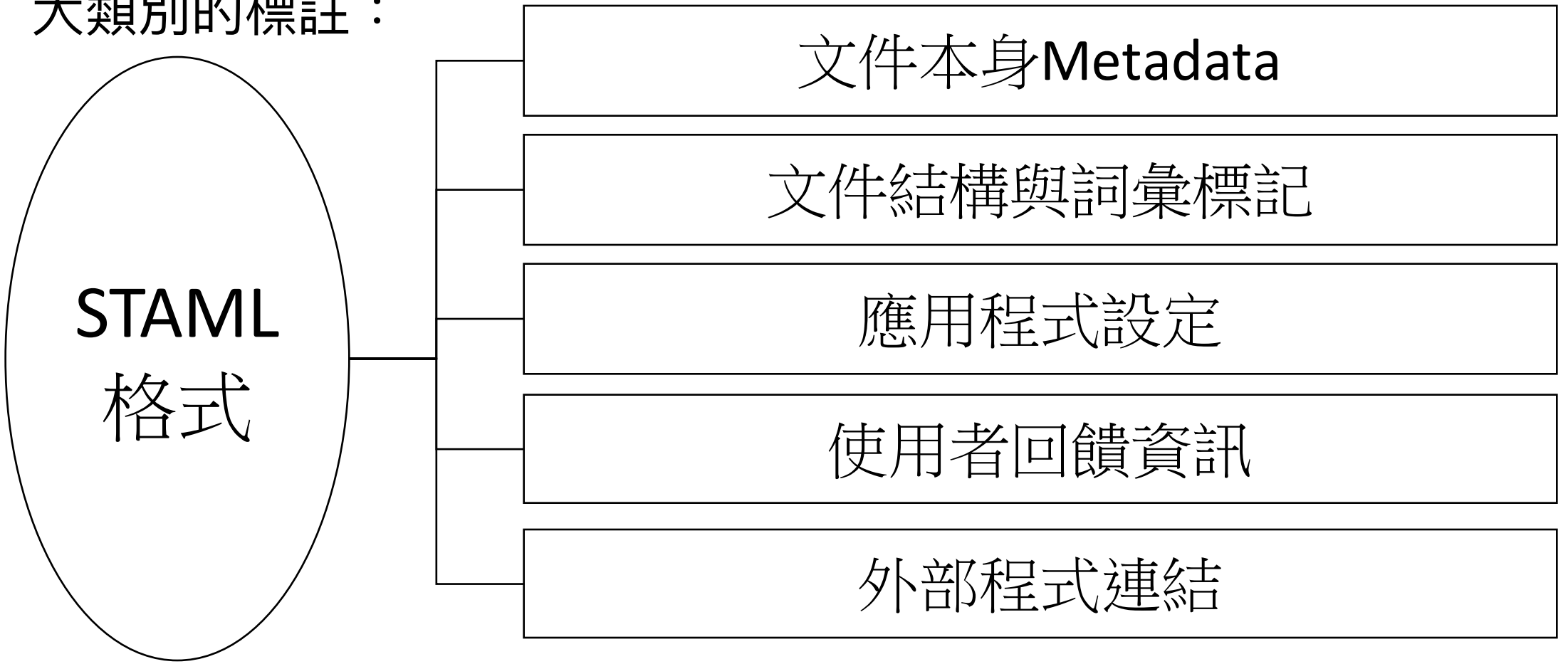


STAML(Simple Text Annotation Markup Language)

- 做為標記工具與標記工具格式之間轉換與標記工具與建構個人資料庫之間轉換的中介格式。
- 利用ＸＭＬ的規範所定義出來，為ＸＭＬ之子集。
- 透過觀察各個標記工具所定義的格式後，會了解到STAML除了必須保留下來文本本身的資訊與標記的結果外，還必須保留下各個工具之應用程式個別的設定與使用者個人所標註的資料，並且也得保留一些與外部其他程式之間的連結。
- 底下我會來詳細解釋STAML格式所應該具備的一些特性。

STAML(Simple Text Annotation Markup Language)

- 從其餘工具的格式中觀察出，STAML格式的制定必須要有以下五大類別的標註：



STAML(Simple Text Annotation Markup Language)

文件本身Metadata

- 像是在THDL之XML中：

檔名
<document filename="TPYL-01_01_001_000-001-0001">
文章標題
<title>三五歷記</title>
作者
<author order="0">徐整</author>

STAML(Simple Text Annotation Markup Language)

文件本身Metadata

- 而在Markus中這類資訊比較少，唯一有的就是檔名：

檔名
<div class="doc" markupfullname="true … filename=“markus_test”

STAML(Simple Text Annotation Markup Language)

文件本身Metadata

- 根據目前的整理，STAML本身需制定：
  - 檔名
  - 文章標題
  - 作者
  - 時間
- 而由於Metadata對於不同種類的文本資料會有很多不同類別的Metadata，故未制定的部分應當要可以自定義標籤來進行存放。

STAML(Simple Text Annotation Markup Language)

文件結構與詞彙標記

- 像是在Markus中：

段落

```
<span class="passage" type="passage" id="passage1">
```

人名

```
<span class="fullName markup unsolved " type="fullName" …>朱自清</span>散文結構
```

地名

```
故稱「我是<span class="placeName markup unsolved" type="placeName">揚州</span>人」
```

STAML(Simple Text Annotation Markup Language)

文件結構與詞彙標記

- 文章結構本身會具有章節以及段落的區別，而詞彙的標記應當可以以人、事、時、地、物這五類做為標記的依據，而對於究竟是什麼類型的事情、什麼類型的物品等等這些詳細的類別應當可以自定義屬性去做詳加的描述。

STAML(Simple Text Annotation Markup Language)

應用程式設定

- 像是在Markus中：

有經過Markus自動標記人名過

```
<div class=“doc” markupfullname=“true”
```

標注樣式之設定

```
<div class=“doc” tag="{&amp;quot;fullName …
quot;&amp;quot;}}”>
```

STAML(Simple Text Annotation Markup Language)

應用程式設定

- 依照目前的目的來看，至少要先能夠儲存Markus的設定以及詞夾子的設定，以利於可以在這兩個工具之間連接，對於未來可能出現的其他的應用程式，他們也可以在這之中自定義去安插屬於自己所寫之應用程式之設定進去。

STAML(Simple Text Annotation Markup Language)

使用者回饋資訊

- 像是在Markus中：

對於該時間使用者自己做的一些註釋

```
<span class="timePeriod markup unsolved"
type="timePeriod"
timeperiod_id="&amp;#(20844);&amp;#(20803);1912&amp;
#(24180);~1912&amp;#(24180);:&amp;#(27665);&amp;#(22
283);&amp;#(27665);&amp;#(22283);&amp;#(32000);&amp;
#(20803);">民國</span>
```

STAML(Simple Text Annotation Markup Language)

使用者回饋資訊

- 在STAML中，應當要能讓使用者在章節或段落上作註解，並且在標記的地方可以做進一步的描述。

STAML(Simple Text Annotation Markup Language)

外部程式連結

- 像是在Markus中：

該人物在CBDB中之編號

```
<span class="fullName markup unsolved "
type="fullName" cbdbid="74123" x_origin="90.5"
y_origin="130.5" refered="TRUE">朱自清</span>
```

STAML(Simple Text Annotation Markup Language)

外部程式連結

- 由於Markus的人名標記會標記上人物在CBDB之中的ID，故在STAML中自然而然也要把這個ID保留起來，而至於未來若資料還有跟其餘外部網站或資料庫連結，亦可利用相同方法自定義保存起來。

STAML(Simple Text Annotation Markup Language)

- 目前對於Markus中對於文本標記究竟用了什麼標籤以及各個標籤代表什麼意思已經有將之整理成了一張表。
- https://docs.google.com/document/d/1nzFmf0XfC2tQRMVntyYhB0pWK17NFV0aOisC5UuRn2c/edit?usp=sharing

下一步

- 整理詞夾子工具之格式與DocuSkyXML格式，並做出第一版的STAML格式。

- 將格式之間的轉換工具完成。