# Intro. to A.I. Project Proposal - Group 17

# NLP: Nonsense Language Processor

0716060 吳泓緯、0716090 林亮丞、0716304 劉子齊

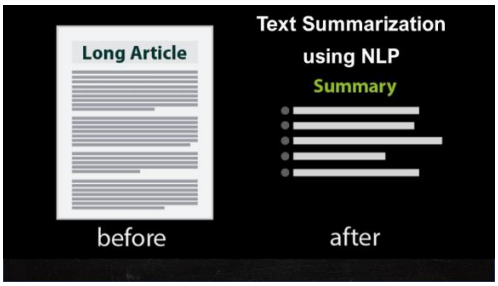
## Problem Statement and Task Definition

在這個資訊大爆炸的世代，巨量的文字及多媒體影音資訊被快速地傳遞並分享於全球各地，資訊超載的問題也因此產生。如何能讓人們快速且有效率地瀏覽與日俱增的文字資訊或多媒體影音資訊，已成為一個刻不容緩的研究課題。在眾多的研究方法中，Text Summarization 被視為是一項不可或缺的關鍵技術。

## Description of the challenges

Text Summarization 屬於自然語言處理的問題，研究如何透過機器讀取文章並生成摘要。可能的解決方法是透過深度學習的技術，以監督式學習訓練神經網路，輸入原始文章到模型中，訓練模型輸出正確的摘要。

## Input/Output Behavior with Concrete Examples




**Sample Input:**

Debbie had taken George for granted for more than fifteen years now. He wasn't sure what exactly had made him choose this time and place to address the issue, but he decided that now was the time. He looked straight into her eyes and just as she was about to speak, turned away and walked out the door.

**Sample Output:**

Debbie walked out of the door when George was about to first say something for fifteen years.

## Related works

目前在 NLP 的領域，有許多人在 Text Summarizer 上做了許多研究。儘管如此，這個問題至今仍然未被 100% 解決，但是還是有很多令人讚嘆的技術和公司正在這個問題上努力著，例如 Google (BERT)、IBM (MAX) 和 ELMo 等。

目前在 NLP 領域中的 Text Summarizer 上，大家所嘗試的方法可以分為 Extractive 與 Abstractive 兩種類型。

Extractive 方法的目標在於透過一個 Score Function，為文章中的每一個句子計算分數，象徵該句子在文章中的重要性，繼而將這些重要句子從文章中取出，組成一篇較為簡短的摘要；而 Abstractive 方法的目標則是讓模型理解文章的意義後，再生成相符的摘要。模型所生成的句子不一定能夠在原來的文章中找到。

**Extractive**

使用核心算法按重要性對句子進行排名：

1. 將單詞與其語法對應物聯繫起來。 （例如“city”和“cities”）
2. 計算文本中每個單詞的出現次數。
3. 根據每個單詞的受歡迎程度為每個單詞分配加權。
5. 檢測哪些句點代表句子的結尾。 （例如“Mr.”沒有）。
6. 將文本分成單獨的句子。
7. 根據單詞的加權總和對句子進行排名。
8. 按時間順序返回 X 個排名中最高的句子

藉由文章標題，並通過選擇與標題相關的關鍵詞，以充新組織整個段落

**Abstractive**

在 Abstractive 方法中，我們除了需要模型學會理解文章的意義，每篇文章與其摘要的長度也不固定，因此模型必須學會輸出符合文意且長度合理的摘要。為了能讓模型自由控制輸出的長度，通常會選擇 Sequence-to-Seqneuce (Seq2Seq) 模型。利用 Seq2Seq 模型讓模型學會理解文章的意義，輸出符合文意且長度合理的摘要。

Seq2Seq 模型主要由 Encoder 與 Decoder 所組成。在早期的 Seq2Seq 模型中，主要使用 Recurrent Neural Network (RNN) 作為 Encoder 與 Decoder。其中，為了有效的處理 Long-Dependency 的問題，Long Short-Term Memory (LSTM) 也相應的成為主流。

## Methodology

在許多文獻中提到 Abstractive 方法能夠比 Extractive 方法有更好的表現。因此在此專案中，我們嘗試實作 Abstractive 方法進行 Text Summarization。

使用一般的 RNN、LSTM 或 GRU 所搭建的 Seq2Seq 模型，在處理較長的輸入序列資料時，往往會因為 Long-Dependency 的問題使得模型沒有辦法記得太早以前的資訊，進而降低模型的表現。此外，因為 RNN、LSTM 與 GRU 本身的模型，使得運算無法被平行化，提升模型所需的訓練時間。

在 Transformer 模型問世後，推廣了 Self-Attention 機制能夠有效提升模型的效能，也出現了更多基於 Transformer 所建立的 State-of-Art (SOTA) 模型。

Transformer 因為使用 Self-Attention 取代 RNN 或 CNN 處理序列資料有許多優勢。透

過 Self-Attention 我們不需要對輸入序列中資料間的關聯性有任何假設，也更擅長處理 Long-Dependency 的問題。此外，RNN 只能依序輸出，但是 Self-Attention 能夠平行化的同時輸出。然而，因為 Self-Attention 對於輸入資料沒有順序的觀念，因此如果輸入資料有順序的關係，則必須透過 Positional Encoding 進行處理。

在此專案中，我們將學習 Transformer 模型的組成，並實作 Transformer 模型來處理 Text Summarization 的問題。

## Evaluation Metrics

我們使用資料集的摘要作為 Reference 和機器生成摘要作為 Candidate，透過下列 BLEU 與 ROUGE 2 種指標來評估機器的表現，以下為 BLEU 與 ROUGE 的說明。

- **BLEU**

  - $Number_{candidate's\ words\ match\ references'}$ / $Number_{candidate's\ words}$

  - 使用 precision 為主，並透過 Sentence brevity penalty 去避免 candidate 太短使分數太高的問題

- **ROUGE**

  - $Number_{candidate's\ words\ match\ references'}$ / $Number_{references'\ words}$

  - 使用 recall 為主，此法會生成 Rouge_1、Rouge_2、Rouge_L 3種分數，其差別分別是以 1-gram、2-gram、longest common subsequence 的形式去得到各自的單詞。
  - 也可以用 F1-score 去結合 BLEU 和 ROUGE:
    - F1 = 2 * (Bleu * Rouge) / (Bleu + Rouge)

## Baselines

如上文所述，解決 Text Summarization 的任務可以分為 Extractive 與 Abstractive 兩種類型。在此專案中，我們針對兩種類型分別設計 Baseline。

### Extarctive Method

Extractive 方法中有多種 Score Function 來計算一個句子的重要性。在此專案中，我們參考連結，採用最簡單的一種作為 Extractive 方法的 Baseline。

我們首先將文章中的 Stopword 去除並進行 Stemming。計算每一個字出現的頻率，使用頻率作為該字的分數。每一個句子的分數即為該句子中所包含所有字的分數總和。為了避免較長的句子有較高的分數，再依據句子的長度進行標準化。接著，計算整篇文章中所有句子的平均分數作為閾值，取出高於閾值的句子作為摘要。

**Abstractive Method**

在 Abstractive Method 中，我們選擇建立一個基本的 Seq2Seq 模型作為 Baseline。Seq2Seq 模型由 Encoder 與 Decoder 組成，Encoder 與 Decoder 中則是透過 LSTM 處理輸入資料。

原始的文章輸入到 Encoder 中，Encoder 學習理解文章的意義，並輸出一個序列。這個序列將會提供給 Decoder。Decoder 則是以 Autoregressive 的方式不斷的輸出新的字，最後利用 Decoder 的輸出組成該篇文章的摘要。

## Project Plan

**資料集收集 (1 week)**

**Extractive Baseline 開發 (1 week)**

- 資料前處理
- 模型建立
- 模型測試

**Abstractive Baseline 開發 (2 week)**

- 資料前處理
- 模型建立
- 模型訓練
- 模型測試

**Transformer 開發 (2 week)**

- 資料前處理
- 模型建立
- 模型訓練
- 模型測試

**實驗與結果比較 (1 week)**

**DEMO (1 week)**

## Future Discussion Recording Link

- https://hackmd.io/@JonathanLiu/Hy8PWZw4q

## Reference


- https://www.mygreatlearning.com/blog/text-summarization-in-python/
- https://towardsdatascience.com/introduction-to-text-summarization-with-rouge-scores-84140c64b471
- https://medium.com/ml-note/text-summarization-techniques-9d950e6b3d8a
- https://medium.com/ml-note/text-summarization-techniques-%E4%B8%89-%E6%8A%BD%E8%B1%A1%E5%BC%8F-d9530581b28