# Movie Title Generation

Group 2
0556611 張軒銘 0557205 李蕙中
0656618 邱勉中 0316213 蒲郁文

# Task 1: Movie Title Generation

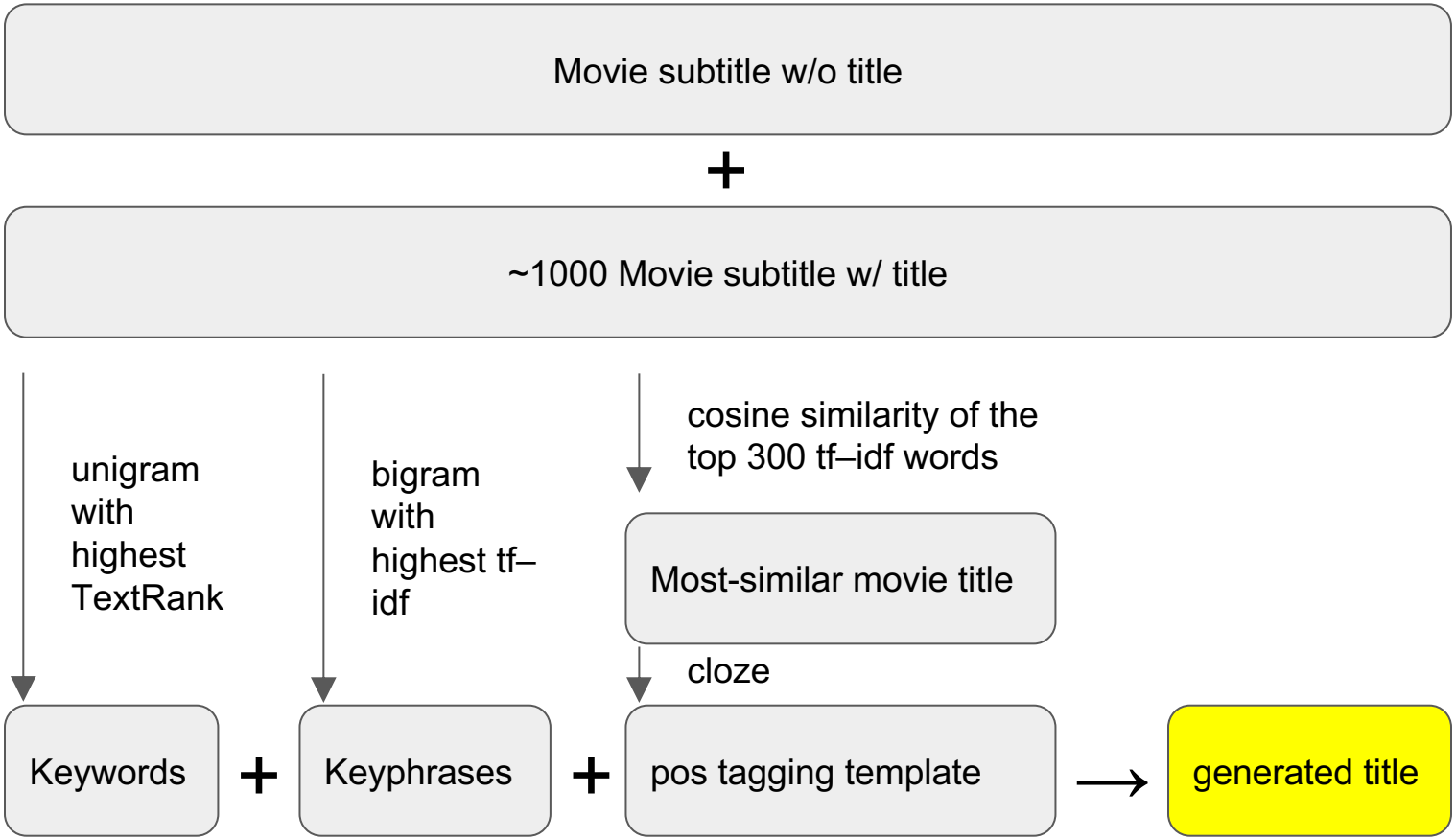
Overview:
我們希望產生出來的電影標題是符合主題且多元的，
儘管有些標題聽起來很有意境，沒有切合主題的話，
對我們而言，就不能算是一個好的標題。
當有很大量的電影需要被 model 命名，每個標題勢必不可以重複。

因此我們針對上述兩個方向❶切題性❷多元性進行設計：
首先，以爬蟲蒐集 1000 多部高票房電影的字幕檔。
接著，找出最能代表該電影的關鍵字（Unigram）及二元詞組（Bigram）。
最後，找出字幕最相近的幾部電影，以那些電影的標題為模板，
依照詞性置換新的字詞進去，產生出新的標題。

# Task 1: Movie Title Generation

Procedures:

1. Parse movie title and subtitle
2. Find keywords in movie subtitle
3. Movie similarity calculation
4. Find the top-10 best match titles with the highest similarity and generate templates
5. Filled keywords into chosen templates to create movie title
6. Evaluate the best title with evaluation method


Movie subtitle w/o title
+
~1000 Movie subtitle w/ title
unigram with highest TextRank
bigram with highest tf–idf
cosine similarity of the top 300 tf–idf words
Most-similar movie title
cloze
Keywords
+
Keyphrases
+
pos tagging template
generated title

# POS template generation

- **Keyword** : Calculate the possible keyword of movie title from the movie title database (with frequency)
- **POS** :
  - Find the possible POS pattern of the movie title **with keyword**

| 格式 | NN,**的**,NN | NN,**游戏** | NN,**冒险**,NN | **魔鬼**,NN | NN,VV,**令** |
|---|---|---|---|---|---|
| 範例 | 硫磺岛,**的**,英雄们 | 饥饿,**游戏** | 白日梦,**冒险**,王 | **魔鬼**,终结者 | 火线,追缉,**令** |

  - Find the possible pattern of the movie title **without keyword**

| 格式 | NN,NN | NN,VV | JJ,NN | NN,CD |
|---|---|---|---|---|
| 範例 | 博物馆,惊魂夜 | 猩球,崛起 | 终极,悍将 | 浪人,47 |

# Determine Important Terms Using Two Metrics

- input: parsed movie subtitle
- **informativeness**
  - keyword extraction
  - TextRank with stop words
  - give a score to each term
- **phraseness**
  - find co-occurring terms (bigrams)

$$\frac{p(a,b)}{p(a)p(b)}$$

  - give a score to each bigram
- sum up these two scores

- example of generated keywords

```
('想 VV', 16.405330138245077),
('少年队 NN', 13.137832385158012),
('龙虎 NR', 12.04960820662706),
('两 CD', 8.775714464465144),
('喜欢 VV', 7.937538544884058),
('事 NN', 7.314618818462128),
('好像 AD', 7.141587572554795),
```

- example of generated phrases

```
{'phraseness': 3.7505805178156373,
 'word_begin': '比尔',
 'word_begin_count': 410,
 'word_begin_pos': 'NR',
 'word_combine': '比尔博',
 'word_combine_count': 8,
 'word_end': '博',
 'word_end_count': 141,
 'word_end_pos': 'NN'}]
```

# Movie Similarity Calculation

- Select 300 most frequent words of each movie subtitle
- Calculate the tf–idf of the 300 words in each movie
- Find top 10 most similar movies using cosine similarity

# Apply Templates

- Use similar movie titles as templates.
  - ex: 魔鬼終結者 → 魔鬼 (名詞) 、(名詞) 終結者
- Fill in the blanks with terms that match the POS of the blanks.
  - highest-scored term/bigram first
- Calculate the score of the title weighted by
  - scores of informativeness and phraseness
  - length penalty
  - popularity of the template
  - word vector distance to known popular title
- Select the title that has the highest score.

# Title Generated with Keyword & Keyphrases

- We also use keywords and keyphrases with highest scores as movie titles.
- So we will have many candidate titles generated from POS templates, keywords and keyphrases.
- We will use task 2 to evaluate those titles to choose the best one.

# Task 2: Movie Title Evaluation

課堂報告時，我們計畫用這四點來評估：

1. check if the POS tagging match our POS title templates
2. check if the length of the title is adequate
3. check if the words in the title is included in our popular movie list
4. if the generated titles receive same points, select the generated title with higher word vector similarity to the known popular movie titles

後來，我們做了更仔細的設計…

# Overview:

針對幾種可能的情境，
我們設計了許多 feature 專門用來評估十部新標題

情境 1. 也許有的會用到一些平淡無奇不吸引人的詞語

情境 2. 也許有的詞性會搭配得很不合理

情境 3. 也許有的名稱和已知的熱門電影相同

情境 4. 也許有的名稱太長、或是太短

情境 5. 或許都是已知熱門電影的名稱

情境 6. 或許詞意不符合常理

# 情境 1. 也許有的會用到一些平淡無奇不吸引人的詞語

- 我們先用超過一千部熱門電影作為資料庫，並且進行段詞；如果該標題用到的詞語和已知熱門電影用詞的重疊率不高，就視作為平淡無奇的詞
- feature: joint_percentage
  該標題斷詞以後，有多少比例是和已知熱門電影用詞相同

| | Title | joint_percentage |
|---|---|---|
| 0 | 復仇者聯盟 | 100.000000 |
| 1 | 星際異攻隊 | 100.000000 |
| 2 | 猜火車 | 100.000000 |
| 3 | 臉書 | 50.000000 |
| 4 | 天競之戰 | 50.000000 |
| 5 | 星際情人 | 100.000000 |
| 6 | 工程六館 | 0.000000 |
| 7 | 樓下的教授 | 33.333333 |
| 8 | 冰與火之歌 | 75.000000 |
| 9 | 冰雪聰明 | 50.000000 |

# 情境 2. 也許有的詞性會搭配得很不合理

- 我們先用超過一千部熱門電影作為資料庫，並且進行段詞；觀察每一種詞性模板 (e.g. NN,NN) 所佔的總比例
- 詞性搭配比較常見的會被我們視為比較好的標題
- feature: pos_percentage
  該標題的詞性模板佔所有熱門電影中的比例

| | Title | pos_percentage |
|---|---|---|
| 0 | 復仇者聯盟 | 21.485261 |
| 1 | 星際異攻隊 | 21.485261 |
| 2 | 猜火車 | 6.235828 |
| 3 | 臉書 | 21.485261 |
| 4 | 天籟之戰 | 21.485261 |
| 5 | 星際情人 | 5.498866 |
| 6 | 工程六館 | 0.396825 |
| 7 | 樓下的教授 | 0.793651 |
| 8 | 冰與火之歌 | 0.000000 |
| 9 | 冰雪聰明 | 21.485261 |

# 情境 3. 也許有的名稱和已知的熱門電影相同

- 如果已知熱門名稱完全相同，
  視為極好的標題
- feature: exactly_same
  - if same: 1
  - else: 0

| | Title | exactly_same |
|---|---|---|
| 0 | 復仇者聯盟 | 1 |
| 1 | 星際異攻隊 | 1 |
| 2 | 猜火車 | 1 |
| 3 | 臉書 | 0 |
| 4 | 天籟之戰 | 0 |
| 5 | 星際情人 | 0 |
| 6 | 工程六館 | 0 |
| 7 | 樓下的教授 | 0 |
| 8 | 冰與火之歌 | 0 |
| 9 | 冰雪聰明 | 0 |

# 情境 4. 也許有的名稱太長、或是太短

- 已知多數電影名稱的平均長度是5.19字
- 根據常理判斷，小於兩個字或超過八個字才也許算是異常，實際情形要依照內容而定。所以我們客製化的為字數訂定分數

```
len_score=[]
for movie in movieTitles:
    length = len(movie)
    if length<2 or length>8:
        score = 0.1
    elif length<7:
        score = 0.9
    else:
        score = 0.3
    len_score.append(score)
```

# 情境 5. 或許都是已知熱門電影的名稱

- 用imdb分數和是不是世界前五百大電影給分
- feature: Title_IMDB, Title_Top500

| | Title | Title_IMDB | Title_Top500 |
|---|---|---|---|
| 0 | 復仇者聯盟 | 7.4 | 3.0 |
| 1 | 星際異攻隊 | 8.1 | 0.0 |
| 2 | 猜火車 | 8.2 | 0.0 |
| 3 | 臉書 | 0 | 0.0 |

# 情境 6. 或許詞意不符合常理

- 我們假定詞意類似於已知電影代表詞意合理
- 用word2vec，找出最相近的電影，並且依據相近程度給分
- feature: mv_1, sim_1

| | Title | mv_1 | sim_1 |
|---|---|---|---|
| 0 | 復仇者聯盟 | 勇敢復仇人 | 0.819091 |
| 1 | 星際異攻隊 | 51號星球 | 0.441186 |
| 2 | 猜火車 | 玩咖尬宅爸 | 0.409343 |
| 3 | 臉書 | 魔髮奇緣 | 0.577564 |
| 4 | 天戰之戰 | 惡靈古堡5：天譴日 | 1.000000 |
| 5 | 星際情人 | 星 | 1.000000 |
| 6 | 工程六館 | 絕命尬車 | 0.417396 |
| 7 | 樓下的教授 | 停車場夜驚魂 | 0.589508 |
| 8 | 冰與火之歌 | 血鑽石 | 0.498545 |
| 9 | 冰雪聰明 | 冰 | 1.000000 |

# 情境與feature對應關係

情境 1. 也許有的會用到一些平淡無奇不吸引人的詞語 joint_percentage

情境 2. 也許有的詞性會搭配得很不合理 pos_percentage

情境 3. 也許有的名稱和已知的熱門電影相同 exactly_same

情境 4. 也許有的名稱太長、或是太短 len_score

情境 5. 或許都是已知熱門電影的名稱 Title_IMDB Title_Top500

情境 6. 或許詞意不符合常理 sim_1

我們把十部電影各項 feature 的分數都 normalize to 0 ~ 1，加權計算出總分（Sum），並且排序，成效不錯，有效辨識出了已知的電影（復仇者聯盟... etc.）和最不吸引人的「工程六館」。

| | joint_percentage | pos_percentage | exactly_same | sim_1 | Title_IMDB | Title_Top500 | len_score | Sum |
|---|---|---|---|---|---|---|---|---|
| Title | | | | | | | | |
| 復仇者聯盟 | 1.000000 | 1.000000 | 1.0 | 0.693716 | 0.902439 | 1.0 | 0.0 | 5.596155 |
| 星際異攻隊 | 1.000000 | 1.000000 | 1.0 | 0.053911 | 0.987805 | 0.0 | 0.0 | 4.041715 |
| 猜火車 | 1.000000 | 0.290237 | 1.0 | 0.000000 | 1.000000 | 0.0 | 0.0 | 3.290237 |
| 天籟之戰 | 0.500000 | 1.000000 | 0.0 | 1.000000 | 0.000000 | 0.0 | 0.0 | 2.500000 |
| 冰雪聰明 | 0.500000 | 1.000000 | 0.0 | 1.000000 | 0.000000 | 0.0 | 0.0 | 2.500000 |
| 星際情人 | 1.000000 | 0.255937 | 0.0 | 1.000000 | 0.000000 | 0.0 | 0.0 | 2.255937 |
| 臉書 | 0.500000 | 1.000000 | 0.0 | 0.284803 | 0.000000 | 0.0 | 0.0 | 1.784803 |
| 冰與火之歌 | 0.750000 | 0.000000 | 0.0 | 0.151022 | 0.000000 | 0.0 | 0.0 | 0.901022 |
| 樓下的教授 | 0.333333 | 0.036939 | 0.0 | 0.305024 | 0.000000 | 0.0 | 0.0 | 0.675297 |
| 工程六館 | 0.000000 | 0.018470 | 0.0 | 0.013633 | 0.000000 | 0.0 | 0.0 | 0.032103 |

# References


- Farnaz Ronaghi Khameneh, “Automatic Title Generation,” Computational Approaches to Digital Stewardship, Stanford University, 2009. [Online]. Available: http://cads.stanford.edu/projects/presentations/summer-2009/Farnaz%20-%20Automatic%20title%20generation.pdf. [Accessed: Nov. 30, 2017].