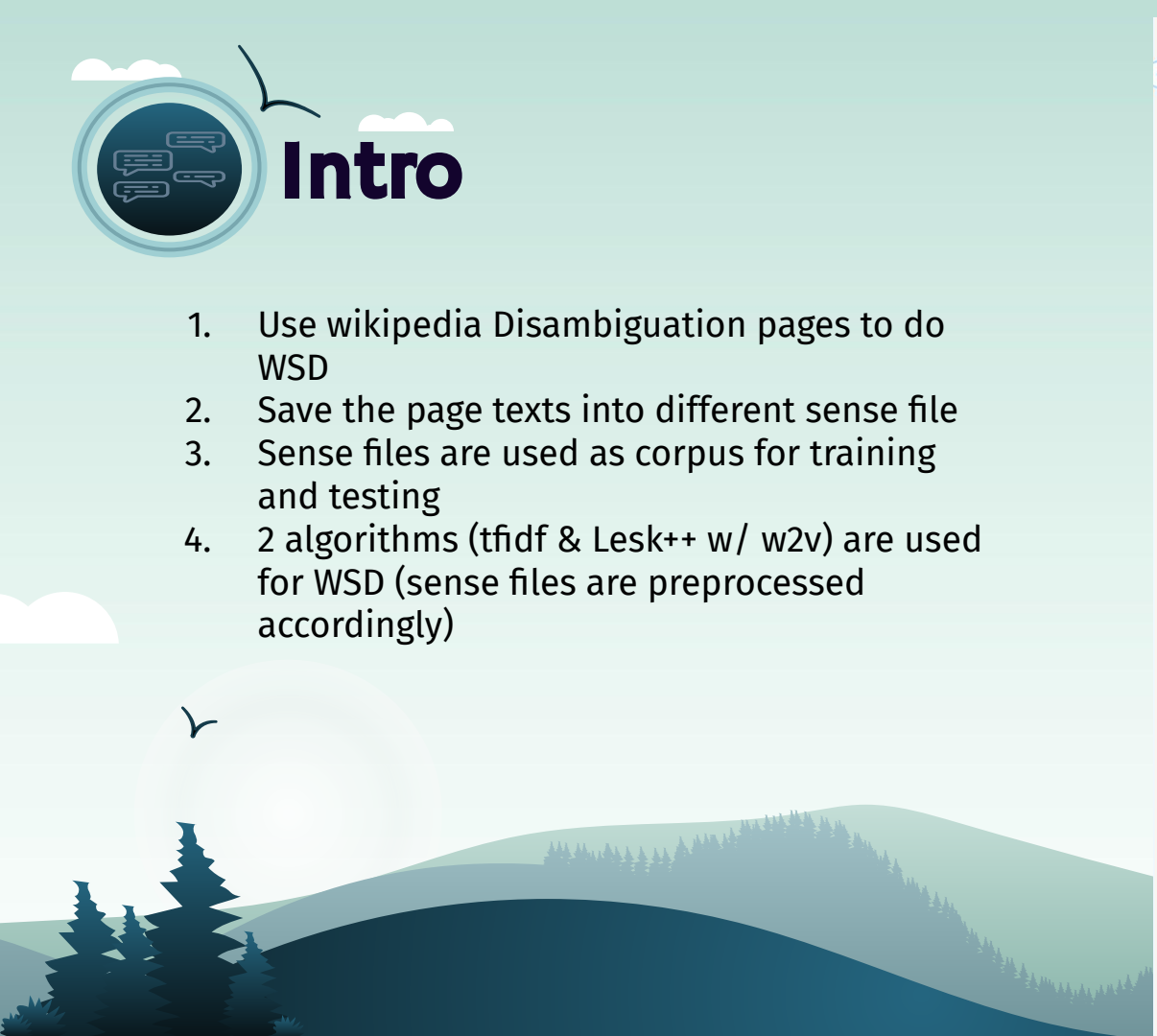
Chinese Wiki-Based Word Sense Disambiguation

110-1 ComSem Term Project
B06102020 外文五 楊晴雯
https://cutt.ly/OUMDuks

Intro

1. Use wikipedia Disambiguation pages to do WSD
2. Save the page texts into different sense file
3. Sense files are used as corpus for training and testing
4. 2 algorithms (tfidf & Lesk++ w/ w2v) are used for WSD (sense files are preprocessed accordingly)



蘋果 (消歧義) [編輯]

維基百科，自由的百科全書

[] (英語：Apple) ，很好吃喔

目次 [隱藏]
1 公司
2 報紙
3 電影
4 藝人
5 參見

公司 [編輯]

- 蘋果公司，著名電子產品生產商
  - 蘋果園區，蘋果公司於2017年4月起啟用的公司總部新址
- 蘋果唱片公司，披頭士樂團創立的唱片公司

報紙 [編輯]

- 蘋果日報
  - 蘋果日報 (香港)，香港公司壹傳媒在香港發行的報紙
  - 蘋果日報 (臺灣)，香港公司壹傳媒在臺灣發行的報紙
  - 蘋果日報慈善基金，香港一個慈善基金，由壹傳媒有限公司於1995年成立

電影 [編輯]

- 蘋果 (電影)，2007年上映的中國電影
- 蘋果 (南韓電影)，2008年上映的南韓電影

藝人 [編輯]

- 拉芘莎拉·瑛特勒素，泰國女演員、歌手，小名Apple
- 黃暐婷 ，臺灣女性藝人，藝名apple

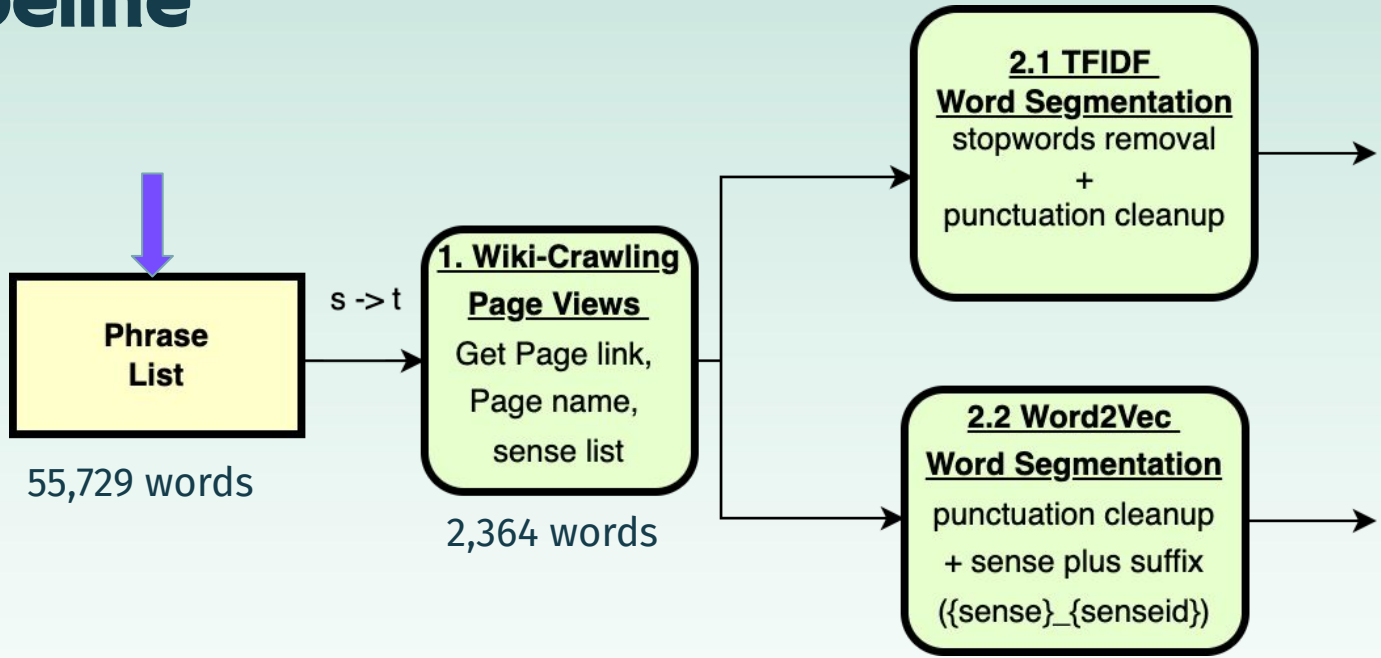
參見 [編輯]

- 以「苹果」開頭的條目
- 名稱包含「苹果」的頁面

Dataset

Phrase list
Wiki-crawling
Word-segmentation

Pipeline


Phrase List
55,729 words
s -> t
1. Wiki-Crawling
Page Views
Get Page link, Page name, sense list
2,364 words
2.1 TFIDF
Word Segmentation
stopwords removal + punctuation cleanup
2.2 Word2Vec
Word Segmentation
punctuation cleanup + sense plus suffix ({sense}_{senseid})

Phrase list + Crawling

- 55,729 words in 現代漢語常用詞表
- OpenCC: simplified->traditional Chinese
- Filtering: For a word w,

```
if (WikiPage(w) exists) && (>=2 sense
pages available in DisambigPage(w))
    keep w
```

- **2,364 words** after filtering
  eg. 蘋果、葡萄、城堡、彩虹

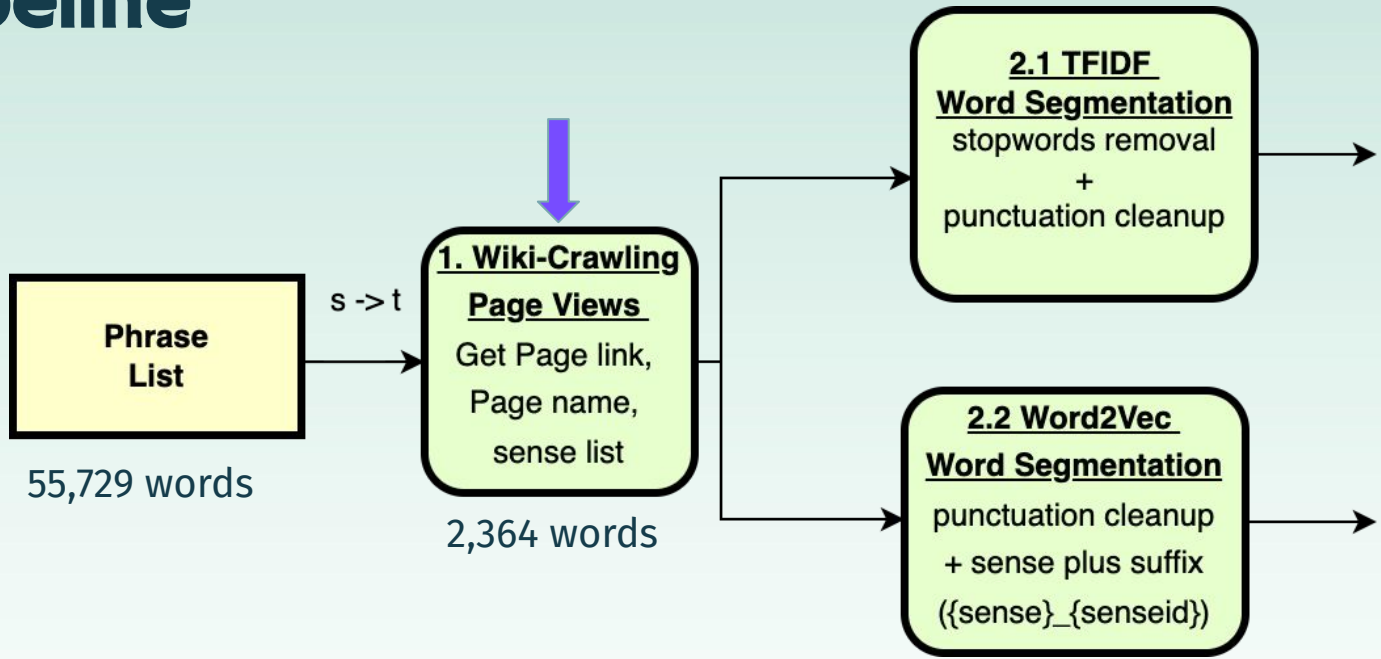
现代汉语常用词表

现代汉语常用词表.txt

This file has been truncated, but you can view the full file.

```
1   阿爸      a1'ba4  18137
2   阿昌族    a1'chang1'zu2   50849
3   阿斗      a1'dou3 42632
4   阿飞      a1'fei1 48603
5   阿富汗    a1'fu4'han4     3461
6   阿訇      a1'hong1        34432
7   阿拉伯数字        a1'la1'bo2'shu4'zi4     35937
8   阿拉伯语  a1'la1'bo2'yu3  30476
9   阿妈      a1'ma1  16220
10  阿门      a1'men2 47913
11  阿Q       a1'qiu1 20845
12  阿司匹林  a1'si1'pi3'lin2 40294
13  阿嚏      a1'ti4  54643
14  阿姨      a1'yi2  6842
15  啊        a1      16090
16  啊呀      a1'ya1  15418
17  啊哟      a1'yo1  23908
```

Pipeline



55,729 words

2,364 words

Have a Peek into Dataset...

- Filtering those without category (**<u>164</u> words remain**)

```
├── MSW_list
│   ├── 蘋果 (MSW object)
│   │   ├── .word  蘋果
│   │   ├── .senses  [蘋果, 蘋果公司,...]
│   │   │   ├── sense  蘋果公司 // fine-grained sense
│   │   │   │   ├── ["category"] 公司 // coarse-grained sense
│   │   │   │   ├── ["views"] 978087 (2018.1.1~2021.1.1)
│   │   │   │   ├── ["gloss"] 著名電子產品生產商
│   │   │   │   ├── ["link"]  {WikiPage Link}
│   │   │   │   ├── ["senseidx"] 2
```

公司 ［編輯］

- 蘋果公司，著名電子產品生產商
  - 蘋果園區，蘋果公司於2017年4月起啟用的公司總部新址
- 蘋果唱片公司，披頭士樂團創立的唱片公司

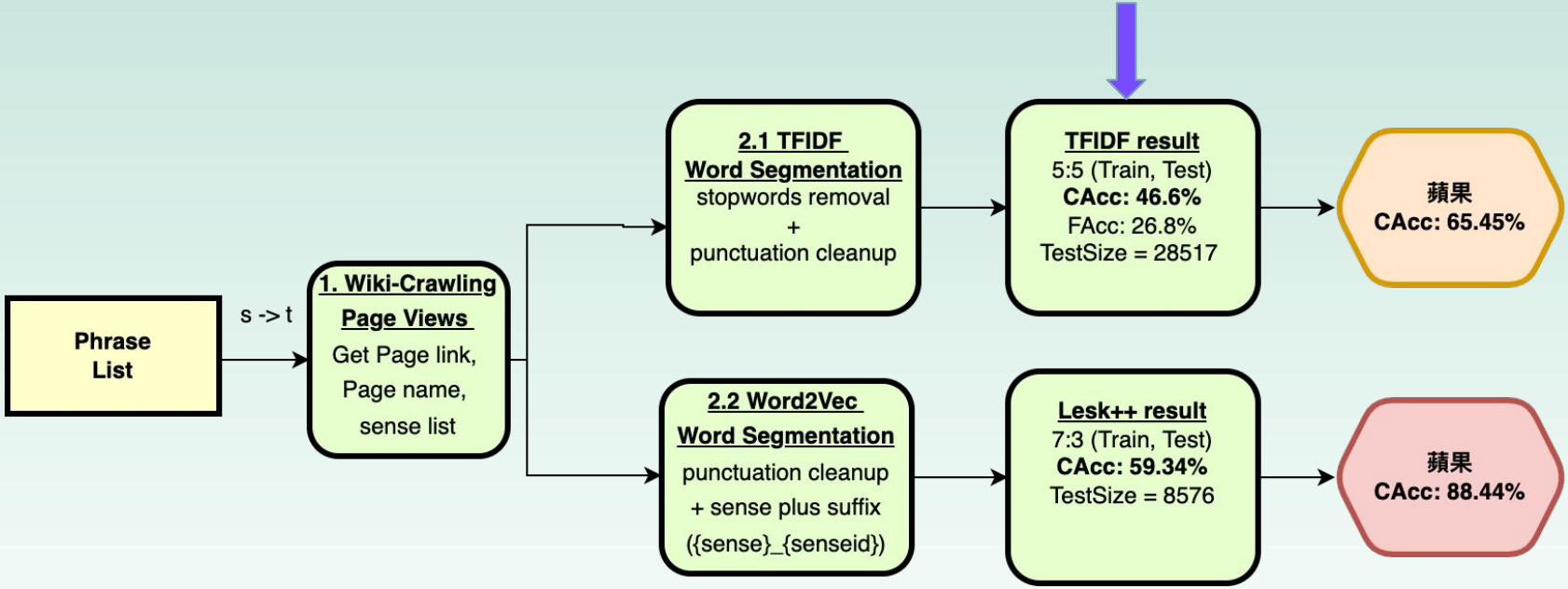
```
[9]  for i in range(len(MSW_list)):
       if MSW_list[i].word == '蘋果':
         print(i, MSW_list[i])

     92 * word: 蘋果,* #senses:10 *hasLevel: True
```


```
[
    {
        "category": "main",
        "sense": "蘋果",
        "views": 34481,
        "gloss": "蘋果樹（學名：Malus domestica）是薔薇科蘋果亞科蘋果屬植物，為落葉喬木，在世界上廣泛種
        "link": "https://zh.wikipedia.org/wiki/%E8%8B%B9%E6%9E%9C",
        "ismain": true,
        "senseidx": 1
    },
    {
        "category": "公司",
        "sense": "蘋果公司",
        "views": 978087,
        "gloss": "蘋果公司，著名電子產品生產商。蘋果園區，蘋果公司於2017年4月起啓用的公司總部新址",
        "link": "https://zh.wikipedia.org/wiki/%E8%98%8B%E6%9E%9C%E5%85%AC%E5%8F%B8",
        "ismain": false,
        "senseidx": 2
    },
    {
        "category": "公司",
        "sense": "蘋果園區",
        "views": 72457,
        "gloss": "蘋果公司，著名電子產品生產商。蘋果園區，蘋果公司於2017年4月起啓用的公司總部新址",
        "link": "https://zh.wikipedia.org/wiki/%E8%98%8B%E6%9E%9C%E5%9C%92%E5%8D%80",
        "ismain": false,
        "senseidx": 3
    },
    {
        "category": "公司",
        "sense": "蘋果唱片公司",
        "views": -1,
        "gloss": "蘋果唱片公司（英語：Apple Corps），披頭士樂團創立的唱片公司",
        "link": "",
        "ismain": false,
        "senseidx": 4
    },
    {
        "category": "報紙",
        "sense": "蘋果日報",
        "views": 86912,
        "gloss": "蘋果日報。蘋果日報 （香港），香港公司壹傳媒在香港發行的報紙。蘋果日報 （臺灣），香港公司壹傳
日報慈善基金，香港一個慈善基金，由壹傳媒有限公司於1995年成立",
        "link": "https://zh.wikipedia.org/wiki/%E8%8B%B9%E6%9E%9C%E6%97%A5%E6%8A%A5",
        "ismain": false,
```

WSD Algorithms

TFIDF
W2V & Lesk++ Algorithm



*CAcc: Coarse-, FAcc: Fine-

TFIDF

Reference: https://www.itread01.com/hkyecfy.html

1. tf: term frequency of a document
2. idf: number of all docs / number of documents where the term appears
3. tfidf_score = tf * idf
4. Ignore the wsd target's tfidf (eg. "蘋果" is not calculated)
5. Choose the sense s.t. the tfidf_sum is maximized

$$tf_{i,j} = \frac{n_{i,j}}{\sum_{k=1} n_{k,j}}$$

$$idf_i = \log\left(\frac{|D|+1}{|j:t_i \in d_j|+1}\right)$$

$$tfidf_{i,j} = tf_{i,j} * idf_i$$

Given S: input sentence,

$$pick\ d_j\ s.t\ score(d_j) = \sum_{w \in S} tfidf(w, d_j)\ \ is\ maximized$$

例：蘋果於11月3日公佈在發佈會三天內的新iPad銷量達三百萬部

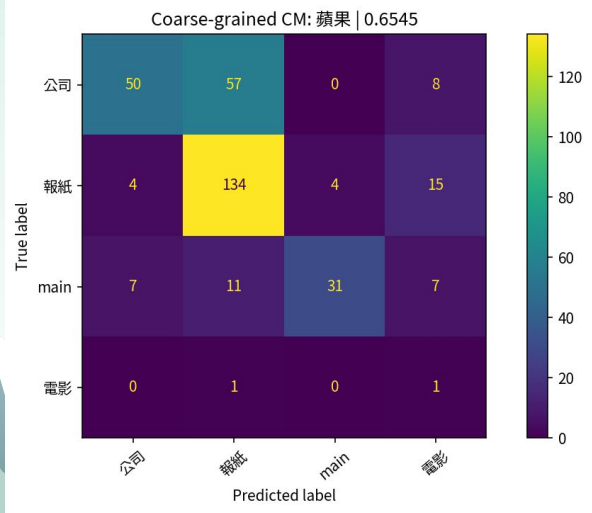
```
* Current random index: 39
* cleaned test sentence:
 ['年', '蘋果', '發表', '聲明', '指', 'iPad', '開售', '八十', '售出', '三百萬', '部']
* y: 公司_蘋果公司
* yhat: 公司_蘋果公司
```

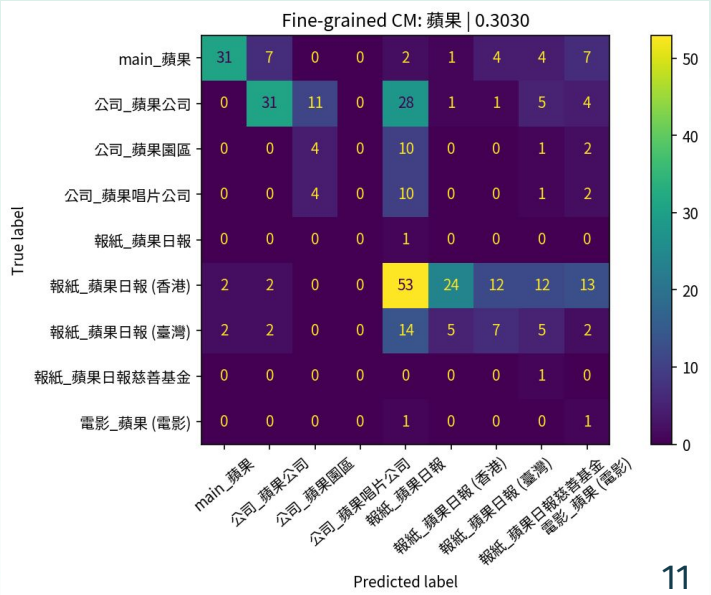
	Tfidf_sum	發表	聲明	iPad	開售	八十	售出	三百萬
main_蘋果	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
公司_蘋果公司	0.004351	0.001062	0.000000	0.002079	0.0	0.0	0.001062	0.000148
公司_蘋果園區	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
公司_蘋果唱片公司	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
報紙_蘋果日報	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
報紙_蘋果日報 (香港)	0.001831	0.001017	0.000712	0.000000	0.0	0.0	0.000102	0.000000
報紙_蘋果日報 (臺灣)	0.000360	0.000000	0.000360	0.000000	0.0	0.0	0.000000	0.000000
報紙_蘋果日報慈善基金	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000
電影_蘋果 (電影)	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.000000	0.000000

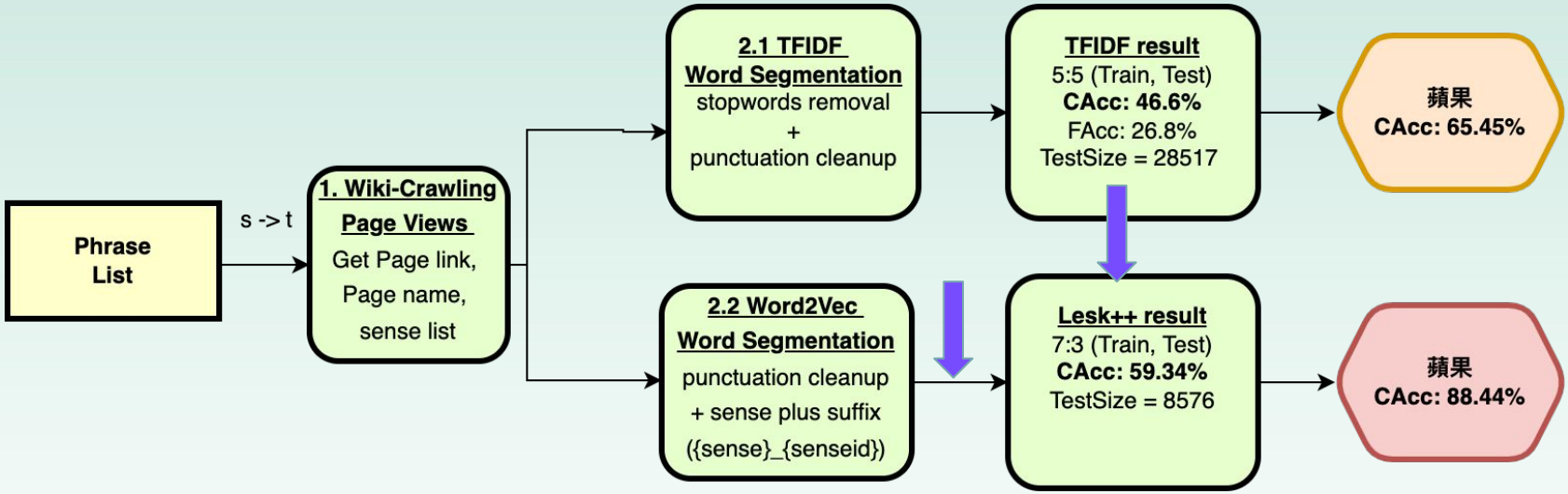
TFIDF: result

- Train: Test = 1:1, Total TestSize = 28,517
  // No randomization
- **Total CAcc: 46.6%** // Coarse-grained Accuracy
- Good when the coarse categories are well-defined and separate enough in terms of semantics
- Input sentences are too short, senses are too close, Unexpected stopwords (tfidf-specific)...

```
[INFO] WSD_target: 蘋果
[INFO] TestSize: 330
[INFO] CAcc: 0.6545
[INFO] FAcc: 0.3030
```






Phrase List
s -> t
1. Wiki-Crawling
Page Views
Get Page link, Page name, sense list
2.1 TFIDF
Word Segmentation
stopwords removal + punctuation cleanup
TFIDF result
5:5 (Train, Test)
CAcc: 46.6%
FAcc: 26.8%
TestSize = 28517
蘋果
CAcc: 65.45%
2.2 Word2Vec
Word Segmentation
punctuation cleanup + sense plus suffix ({sense}_{senseid})
Lesk++ result
7:3 (Train, Test)
CAcc: 59.34%
TestSize = 8576
蘋果
CAcc: 88.44%

```
from gensim.models import Word2Vec
# embedding_dim = 200
EMBDIM= 200
model = Word2Vec(size = EMBDIM,
                 window = 7,
                 min_count = 3,
                 workers = 8,
                 batch_words = 10000)
                 #sg = 1 : use skip-gram model

model.build_vocab(Train_Corpus)
model.train(Train_Corpus, total_examples=len(Train_Corpus), epochs = 30)
```

```
(96776652, 109024290)
```

The trained embedding
(used as Ls,w in Lesk++)

```
similar_words = {
    search_term:
    [item[0] for item in w2v.most_similar([search_term], topn=8)]
    for search_term in
    ['蘋果_1', '蘋果_2', '蘋果_3', '蘋果_4', '蘋果_5']
}
for x in similar_words:
  print(x, similar_words[x])
```

```
蘋果_1 ['品種', '葡萄_1', '番茄_2', '水果', '樹', '馴化', '楊梅_1', '新鮮']
蘋果_2 ['微軟', 'Macintosh', '喬布斯', 'Lisa', 'Mac', 'Micro', 'iMac', 'Prose']
蘋果_3 ['蘋果_4', '園區', 'McCarthyBuildingCcompanies', '田溪', '麥卡錫', '惠普', '書版', 'Premier']
蘋果_4 ['蘋果_3', '園區', 'McCarthyBuildingCcompanies', '田溪', '域迅', '書版', '港鐵', '希代']
蘋果_5 ['爲盆', 'SkyOne', '生死之交', '邯鄲市', '移除掉', '看懂', 'Gabapentin', '善行']
```

3 Lesk++

Our WSD algorithm takes sentences as input and outputs a preferred sense for each polysemous word. Given a sentence $w_1 \dots w_i$ of i words, we retrieve a set of word senses from the sense inventory for each word w. Then, for each sense s of each word w, we consider the similarity of its lexeme (the combination of a word and one of its senses (Rothe and Schütze, 2015)) with the context and the similarity of the gloss with the context.

For each potential sense s of word w, the cosine similarity is computed between its gloss vector G_s and its context vector C_w and between the context vector C_w and the lexeme vector $L_{s,w}$. The score of a given word w and sense s is thus defined as follows:

$$\text{Score}(s, w) = cos(G_s, C_w) + cos(L_{s,w}, C_w) \quad (1)$$

The sense with the highest score is chosen. When no gloss is found for a given sense, only the second part of the equation is used.

Lesk++

- Paper reference: Simple Embedding-Based Word Sense Disambiguation (Dieke & Gertjan, 2018)
- `Ls,w`: trained lexeme vector (the w2v embedding trained on crawled corpus for the sense)
- `Gs`: averaged gloss vector (the definition crawled from Wiki)
- `Cw`: averaged context vector (the input sentence)
- `Score(s,w) = cos(Gs,Cw) + cos(L(s,w), Cw) for all s in w`
- Choose the sense s.t. the score is maximized.
- the $cos(\theta)$ value is in the range [-1,1].

Lesk++: result

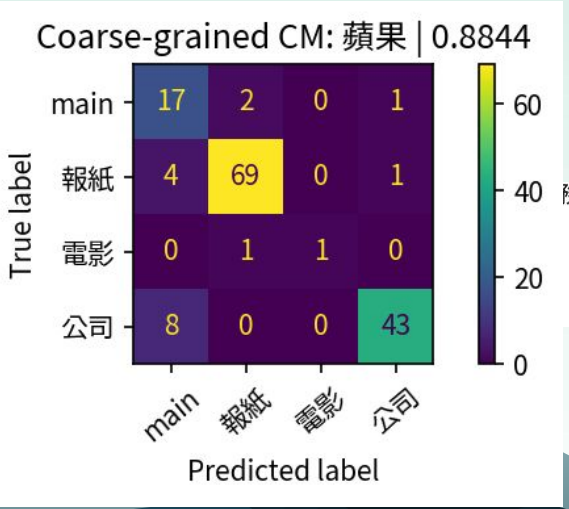
- Train: Test = 7:3, Total TestSize = 8,576

  // with randomization
- **Total Accuracy: 59.34 %** (5,089/8,576)
- Better overall performance than tfidf
- Better distinguishing “公司” and “報紙” than tfidf
- The issue of senses being too close still exists

```
[Info] Start testing...
* ----- Target word: 蘋果 -----
[Info] Total 147 valid sentences
[Info] *** Category Accuracy: 0.8844 ***
----------------------
(130, 147)
```

```
* Testsent: 這 基金 透過 香港 蘋果 日報 籌集 讀者 的 捐款 除 協助 老弱 傷殘 人士 外 每 年 亦 透過 撥款
* Gold category:  報紙 * Pred category: 報紙
('蘋果日報 (香港)_6', 0.94612724)
('蘋果日報 (臺灣)_7', 0.8893519)
('蘋果日報慈善基金_8', 0.789966)
```

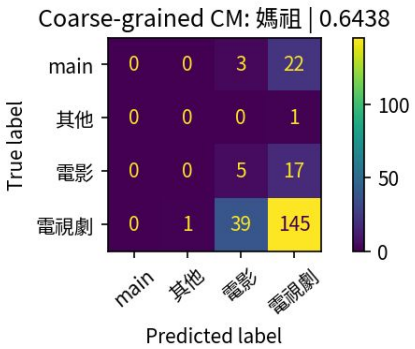
```
* Testsent: 新疆 野 蘋果 被 認爲 是 栽培 蘋果 主要 的 祖先 物種 而且 二 者 在 形態 上 相似
* Gold category:  main * Pred category: main
('蘋果_1', 0.80090725)
('蘋果公司_2', 0.010468703)
('蘋果園區_3', -0.0048323683)
```



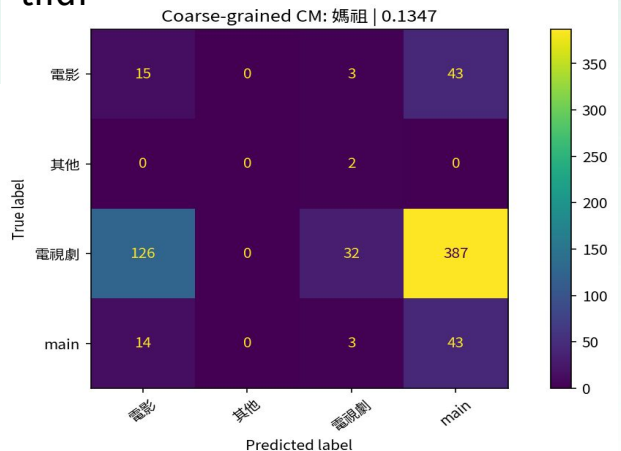
When senses are too close...

- word “媽祖”
- Tfidf: overall 13.47 % CAcc
- Lesk ++: overall 64.38 % CAcc
- Senses are too close
- Sense “電影_海之傳說-媽祖” has many “臺灣” occurrences and very few words in its page (high tf-score)
- Lesk++ completely ignores main category.

Lesk++



tfidf



tfidf

```
* index: 0

* test sentence:
臺灣 媽祖 信仰 臺灣 普遍 民間 信仰
* label: main_媽祖

* prediction: 電影_海之傳說－媽祖

* tfidf matrix:
```

	Tfidf_sum	臺灣	信仰	普遍	民間
main_媽祖	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖的故事	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖外傳	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖後傳	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖過臺灣	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖出巡	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖拜觀音	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_媽祖 （2000年電視劇）	0.1017	0.0646	0.0315	0.0	0.0056
電視劇_天上聖母媽祖	0.2152	0.2152	0.0	0.0	0.0
電視劇_懷玉傳奇 千金媽祖	0.4842	0.4842	0.0	0.0	0.0
其他_媽祖 （電視劇）	0.4842	0.4842	0.0	0.0	0.0
電視劇_媽祖	0.1017	0.0646	0.0315	0.0	0.0056
電影_媽祖顯聖	0.1017	0.0646	0.0315	0.0	0.0056
電影_海之傳說－媽祖	0.7497	0.7497	0.0	0.0	0.0

Thanks!