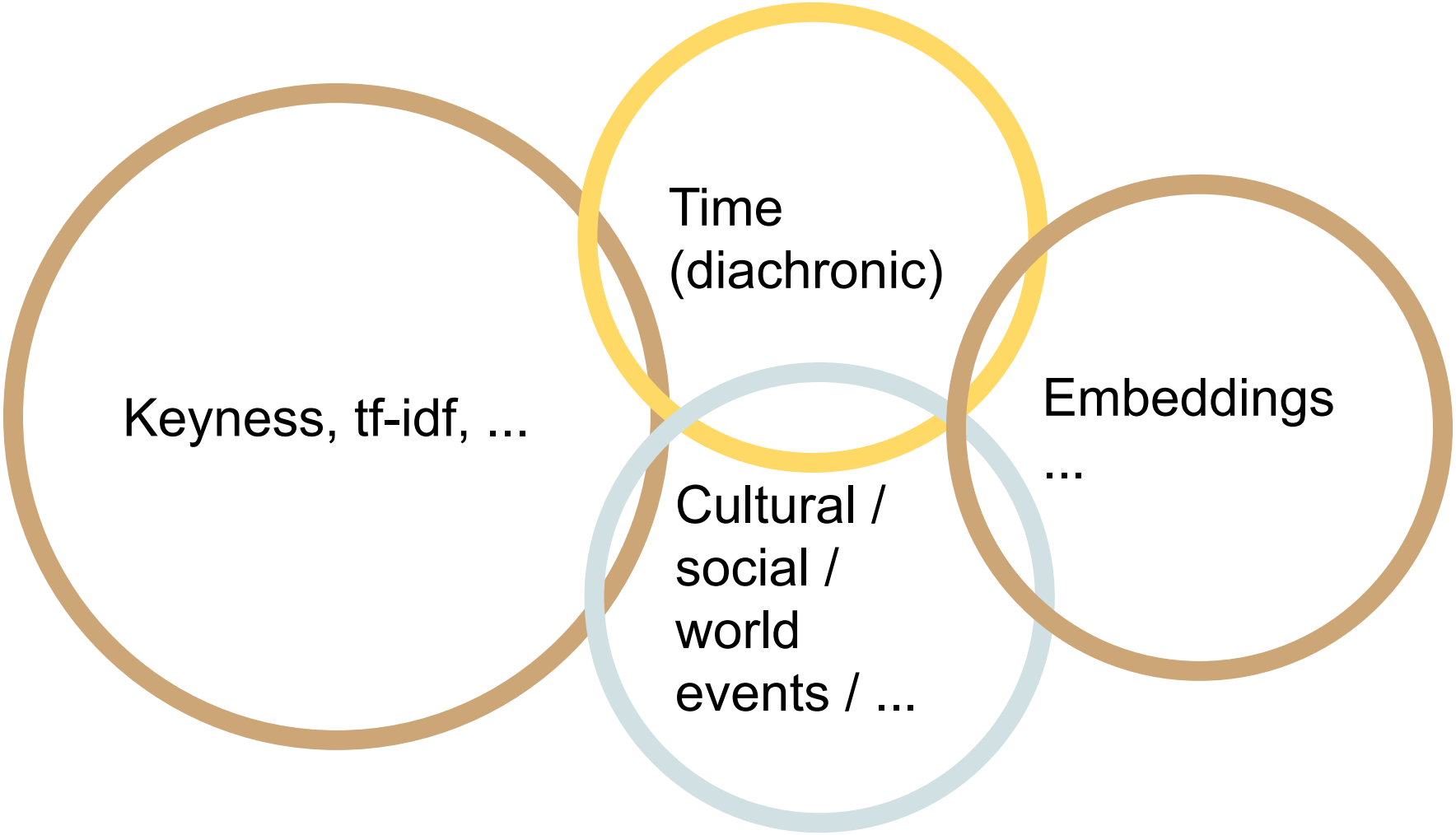
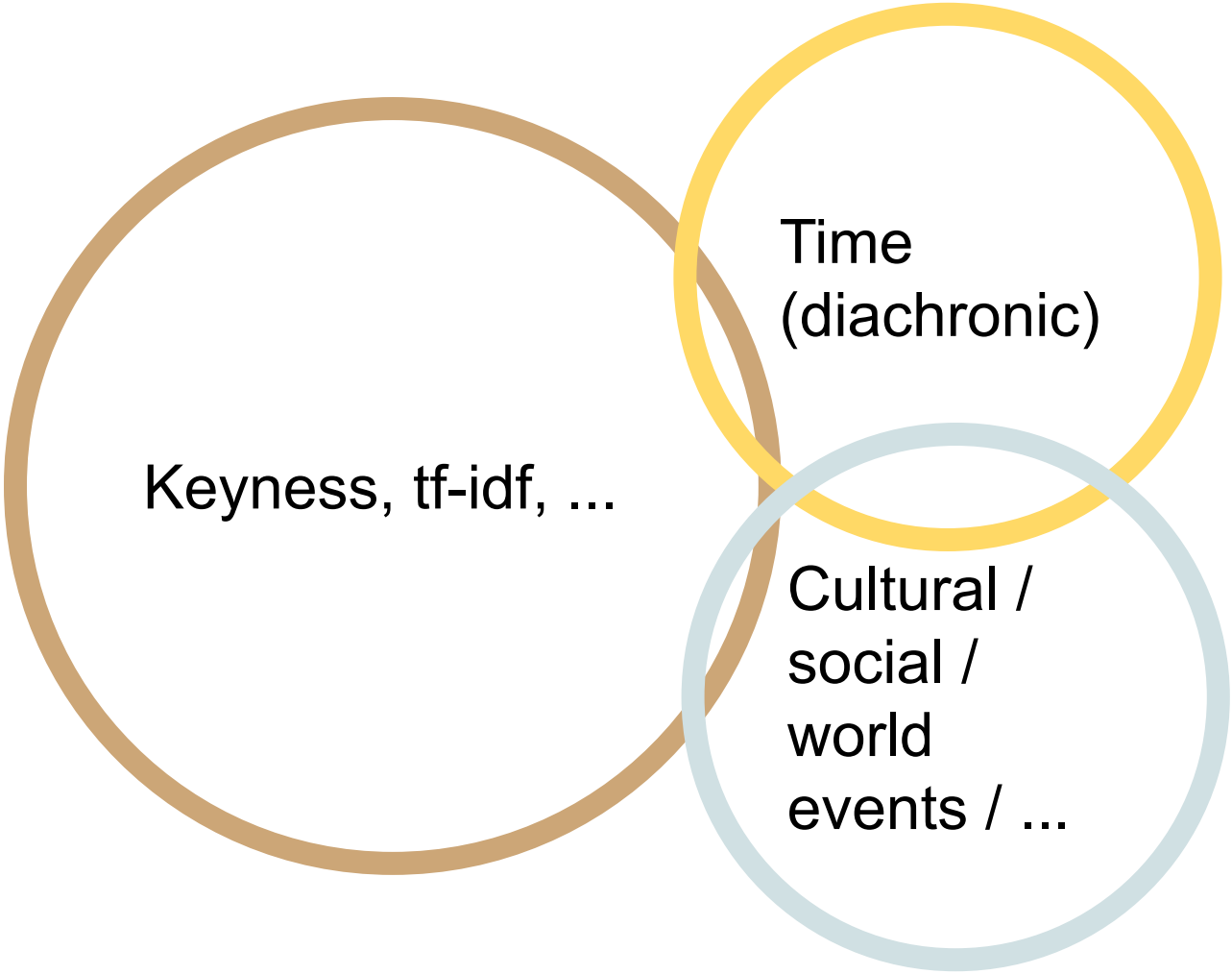
Beyond keyness (II)

Presenter: 陳蓓怡

Notebook ⤵️

https://reurl.cc/Gr3zgp


Keyness, tf-idf, ...
Time
(diachronic)
Cultural /
social /
world
events / ...
Embeddings
...

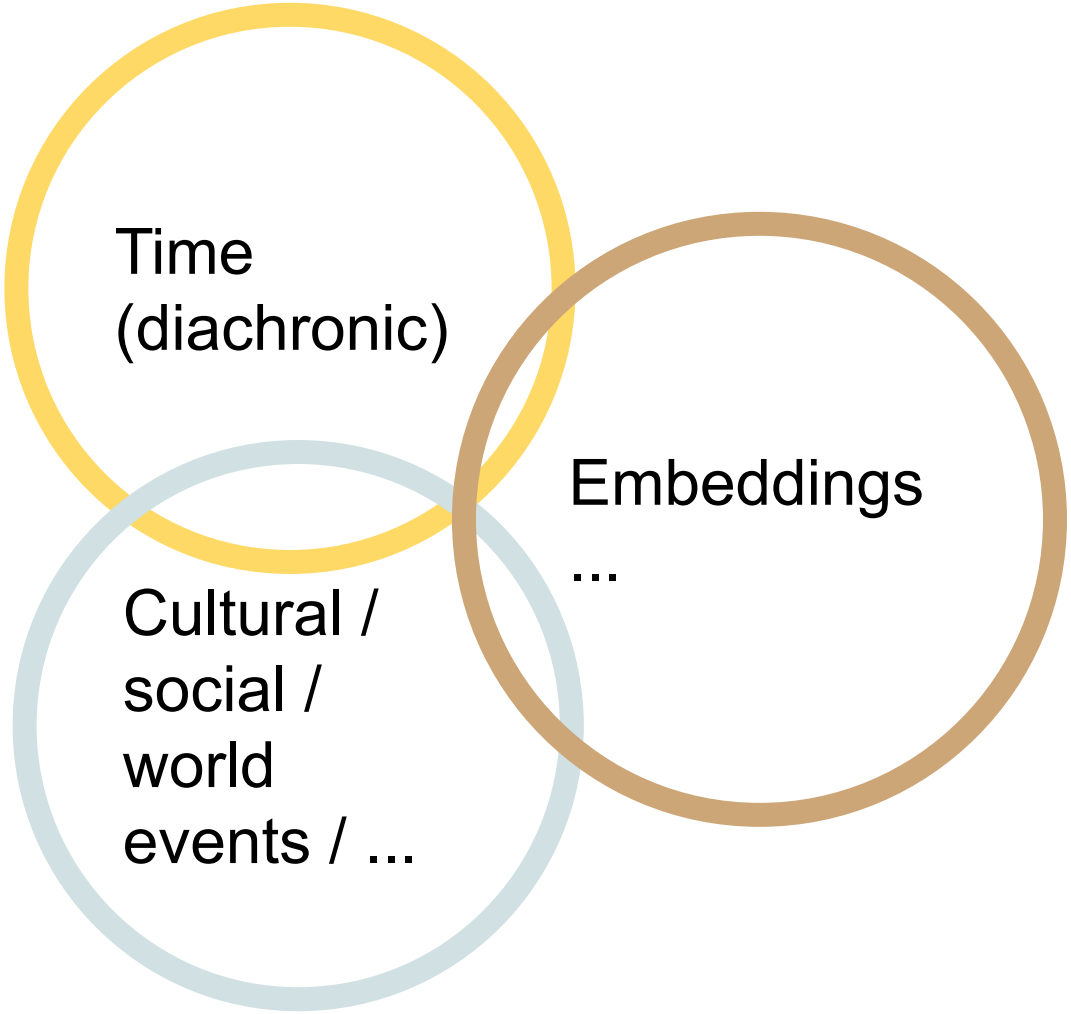

Keyness, tf-idf, ...
Time
(diachronic)
Cultural /
social /
world
events / ...

Topic (Baker, 2004; Bondi and Scott, 2010)

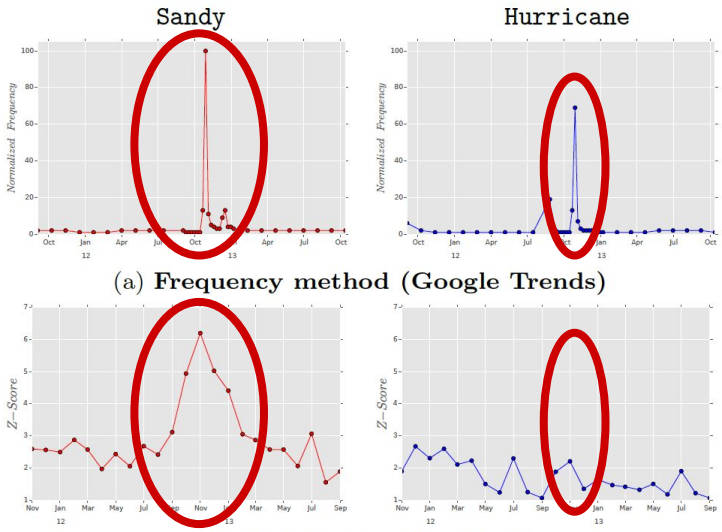
- Keyness, keyword
- What is key?
  - frequency difference
  - comparison with reference corpus

What does keyness tell us? (Baker, 2004; Bondi and Scott, 2010)

- Proper nouns
- Aboutness of a text
- Author's style
  - High frequency words, e.g., *because*, *shall*, *of*
- Concept / ideology / stance in discourse


Time
(diachronic)
Embeddings
...
Cultural /
social /
world
events / ...

Frequency method v.s. Distributional method



(a) **Frequency method (Google Trends)**

(b) **Distributional method**

Figure 2: Comparison between Google Trends and our method. Observe how Google Trends shows spikes in frequency for both `Hurricane` (blue) and `Sandy` (red). Our method, in contrast, models change in usage and detects that only `Sandy` changed its meaning and not `Hurricane`.

(Kulkarni et al, 2015)

Implementation

Diachronic corpus data

- Boards: Gossiping（八卦版）and WomenTalk（女版）
- Years: 2005, 2010, 2015, 2020

- Texts: <版名>_<年份>_seg.txt
- Embeddings: <版名>_<年份>.model

```
!gdown --id "1gEL4v3wGgvqJnpWspISZvLeIL3GQZLB1" -O "Gossiping_2005.model" # 2005 年 Gossiping 板
!gdown --id "1yB9WPVDJVmmLLxbEHZroZP_cYMP0JUpC" -O "Gossiping_2010.model" # 2010 年 Gossiping 板
!gdown --id "1Vh8meq6hdte02nQ2-djclgpEKxFUC0YU" -O "Gossiping_2015.model" # 2015 年 Gossiping 板
!gdown --id "1EiDgWcnDDSOy1bu_aRjbBk4JGIENNoGk" -O "Gossiping_2020.model" # 2020 年 Gossiping 板
```

```
Downloading...
From: https://drive.google.com/uc?id=1gEL4v3wGgvqJnpWspISZvLeIL3GQZLB1
To: /content/Gossiping_2005.model
31.8MB [00:00, 101MB/s]
Downloading...
From: https://drive.google.com/uc?id=1yB9WPVDJVmmLLxbEHZroZP_cYMP0JUpC
To: /content/Gossiping_2010.model
27.7MB [00:00, 129MB/s]
Downloading...
From: https://drive.google.com/uc?id=1Vh8meq6hdte02nQ2-djclgpEKxFUC0YU
To: /content/Gossiping_2015.model
120MB [00:00, 122MB/s]
Downloading...
From: https://drive.google.com/uc?id=1EiDgWcnDDSOy1bu_aRjbBk4JGIENNoGk
To: /content/Gossiping_2020.model
74.2MB [00:00, 129MB/s]
```

Keyness in time

```
data.get_keyness('戦争', '2005', '2015')
```

```
{'corpus_size_A': 433687, 'corpus_size_B': 3709185, 'keyword_freq_A': 18, 'keyword_freq_B': 148}
0.02492168107135512
```

```
data.get_keyness('戦争', '2005', '2020')
```

```
{'corpus_size_A': 433687, 'corpus_size_B': 1345157, 'keyword_freq_A': 18, 'keyword_freq_B': 185}
26.50202265172519
```

Embeddings

```
class Embedding:
    def __init__(self, board, year_lst):
        self.board = board
        self.year_lst = year_lst

        self.path_lst = [f'{board}_{year}.model' for year in self.year_lst]
        self.model_lst = [gensim.models.Word2Vec.load(path) for path in self.path_lst]
```

Neighboring words

```python
# 關鍵字
words = [label]
embeddings = [model[keyword]]
```

```python
labels = []

word_clusters = []
embedding_clusters = []
for year, model in zip(self.year_lst, self.model_lst):

    label = f'{keyword}({year})'

    try:
        # 關鍵字
        words = [label]
        embeddings = [model[keyword]]

        # 近鄰詞
        for similar_word, _ in model.wv.most_similar(keyword, topn=n1+n2):
            words.append(similar_word)
            embeddings.append(model[similar_word])
        embedding_clusters.append(embeddings)
        word_clusters.append(words)

        labels.append(label)
```

Neighboring words

```python
# 關鍵字
words = [label]
embeddings = [model[keyword]]
```

```python
labels = []

word_clusters = []
embedding_clusters = []
for year, model in zip(self.year_lst, self.model_lst):

    label = f'{keyword}({year})'

    try:
        # 關鍵字
        words = [label]
        embeddings = [model[keyword]]

        # 近鄰詞
        for similar_word, _ in model.wv.most_similar(keyword, topn=n1+n2):
            words.append(similar_word)
            embeddings.append(model[similar_word])
        embedding_clusters.append(embeddings)
        word_clusters.append(words)

        labels.append(label)
```

Neighboring words

```
# 關鍵字
words = [label]
embeddings = [model[keyword]]
```

```
labels = []

word_clusters = []
embedding_clusters = []
for year, model in zip(self.year_lst, self.model_lst):

    label = f'{keyword}({year})'

    try:
        # 關鍵字
        words = [label]
        embeddings = [model[keyword]]

        # 近鄰詞
        for similar_word, _ in model.wv.most_similar(keyword, topn=n1+n2):
            words.append(similar_word)
            embeddings.append(model[similar_word])
        embedding_clusters.append(embeddings)
        word_clusters.append(words)

        labels.append(label)
```

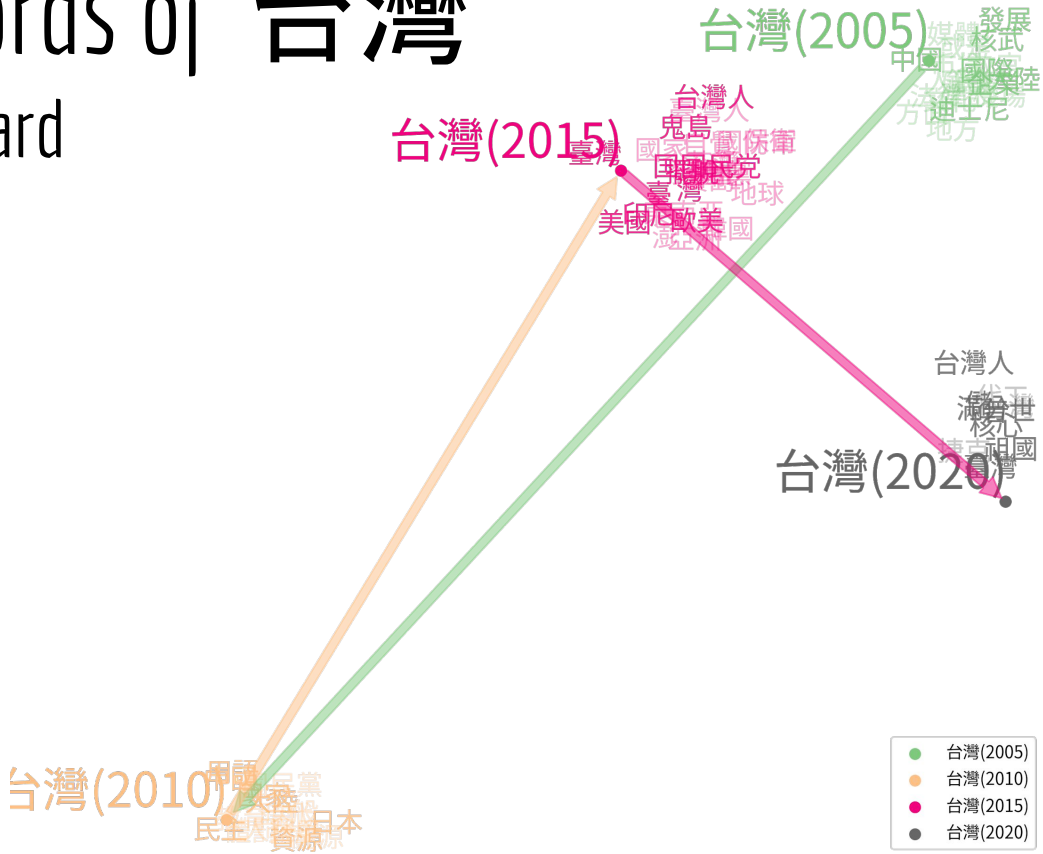
Working with time objects

- Libraries: `datetime`, `time`
- Usages:
  - To convert string or number to date(time)
  - To manipulate format, order, elapse, etc. of time
  - To filter, group, summarize, etc.

Neighboring words of '台灣'

in PTT's Gossiping Board

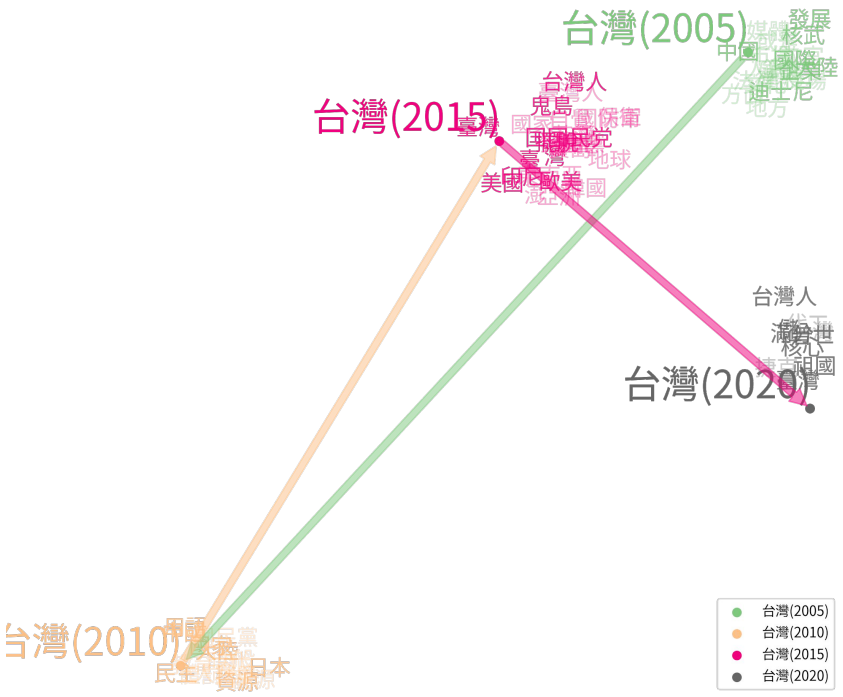
from 2005 to 2020



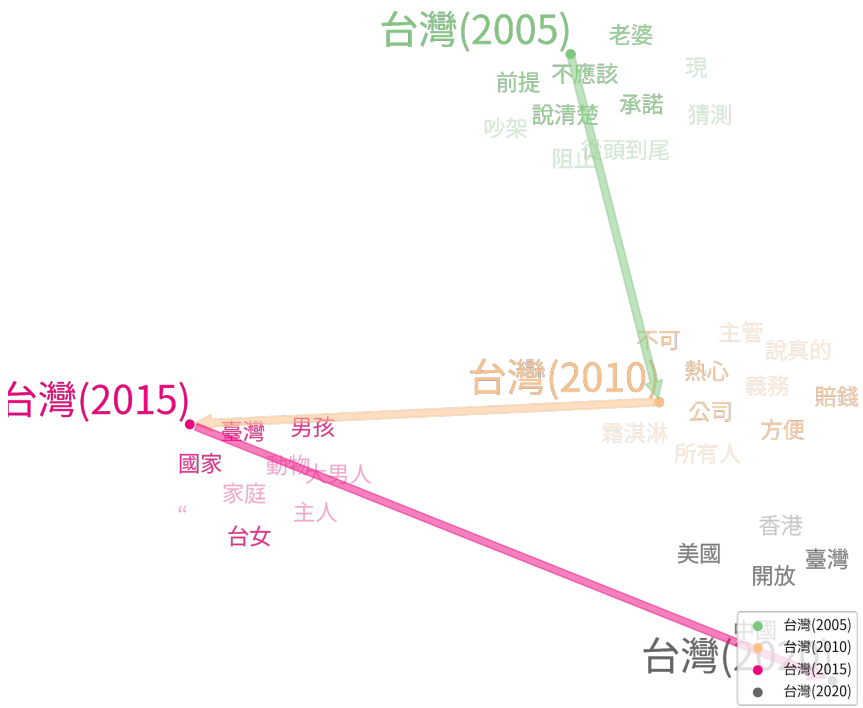
(Hamilton et al, 2016)

Neighboring words across **domains**

Gossiping



WomenTalk



Neighboring words

```python
# 關鍵字
words = [label]
embeddings = [model[keyword]]
```

```python
labels = []

word_clusters = []
embedding_clusters = []
for year, model in zip(self.year_lst, self.model_lst):

    label = f'{keyword}({year})'

    try:
        # 關鍵字
        words = [label]
        embeddings = [model[keyword]]

        # 近鄰詞
        for similar_word, _ in model.wv.most_similar(keyword, topn=n1+n2):
            words.append(similar_word)
            embeddings.append(model[similar_word])
        embedding_clusters.append(embeddings)
        word_clusters.append(words)

        labels.append(label)
```

Levels of neighboring words

```
for i, word in enumerate(words):
    # 關鍵詞本身
    if i == 0:
        a = 1
        size = 28
    # 將近鄰詞分層，調整透明度與字體大小
    elif i >= 1 and i <= n1:
        a = 0.85
        size = 16
    else:
        a = 0.35
        size = 16

    plt.annotate(word, alpha=a, xy=(x[i], y[i]), xytext=(1, 1),
                 textcoords='offset points', ha='right', va='bottom', size=size, c=color)
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

Plotting neighboring words

```
def tsne_plot_similar_words(labels, embedding_clusters, word_clusters, n1):
```

```
class PlotTemporalData(Embedding):
```

```
def create_datapoints(self, keyword, n1=10, n2=15):
```

```
def tsne(self):
```

```
def tsne_plot(self):
    tsne_plot_similar_words(self.labels, self.embeddings_en_2d, self.word_clusters, self.n1)
```

```
keyword = '台灣'
for board in board_lst:
    data = PlotTemporalData(board, year_lst)
    data.create_datapoints(keyword, n1=5, n2=5)
    #data.create_datapoints(keyword)
    data.tsne()
    data.tsne_plot()
```

關鍵詞？

2017.12.29《台南文化關鍵詞「擔仔麵」票選首榜》

- 擔仔麵
- 赤崁樓
- 安平古堡
- 民生綠園
- 古都府城
- 台南運河
- 台江內海

台灣理論關鍵詞

[...]「正義」(謝若蘭)、「佔領」(黃涵榆)、「漂泊」(黃英哲)、「腐」(廖勇超)、「逆轉」(陳東升), 也有對一般讀者來說較為陌生的諸如「文體秩序」(陳國偉)、「化人主義」(黃宗慧)、「分子化翻譯」(張君玫)、「符號混成」(劉紀蕙)、「壞建築」(辜炳達), 也有一眼看起來非常令人好奇的, 像是「鬧鬼」(林芳玫)、「謠言電影」(孫松榮)、「男人魚」(夏曼. 藍波安), 也有大家比較熟悉的詞彙像是「酷兒」(紀大偉)、「基進」(傅大為)[...]

(Photo source: unsplash photo-1586883417979-52a46b1dabdf)

《疫病與社會的十個關鍵詞》

1. 汙名
2. 人權
3. 公衛倫理
4. WHO（全球衛生）
5. CDC（全球衛生）
6. 中醫藥
7. 道德模範
8. 標語
9. 隱喻
10. 旁觀他人之苦

What's next? (Bondi and Scott, 2010)

Characteristics of different corpora:

- Corpus size
- Lexical richness (type-token ratio)
- Mean word length
- Mean sentence length
- ...

What's next? (Bondi and Scott, 2010)

Characteristics of corpus data:

- Interpretations on the lexical level
- Idiosyncratic use
  - document dispersion
- inclusion/exclusion of high- or low-frequency words
- Qualitative analysis of concordances and collocations

What's next? (Bondi and Scott, 2010)

More information:

- Key category / key clusters
- Annotated text
  - Syntactic information, e.g., syntactic categories, grammatical functions
  - Semantic information

References

Baker, P. (2004). Querying keywords: Questions of difference, frequency, and sense in keywords analysis. *Journal of English Linguistics*, *32*(4), 346-359.

Bondi, M., and M. Scott. (Eds.). (2010). *Keyness in texts* (Vol. 41). John Benjamins Publishing.

Kulkarni, V., Al-Rfou, R., Perozzi, B., and Skiena, S. (2015). Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 625-635).

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Cultural shift or linguistic drift? Comparing two computational measures of Semantic Change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pp. 2116-2121.

References

三月人文社科類】台灣理論關鍵詞 . (2019, March 20). OUT OF THE DEAD LAND 盲目的注視.
https://sapphocatlulu.com/2019/03/20/三月人文社科類】台灣理論關鍵詞 /

台南文化關鍵詞「擔仔麵」票選首榜. (2017, December 29). 自由時報.
https://news.ltn.com.tw/news/life/breakingnews/2297155

劉紹華. (2020). 疫病與社會的十個關鍵詞 . 春山.

王驥懋, 史書美, 李育霖. (2019). 台灣理論關鍵詞 . 聯經.