Load libraries and data

```
# !pip install --upgrade pythainlp
# !pip install pyLDAvis
# !pip install -U pandas-profiling
# !pip install sefr_cut
```

```
Collecting sefr_cut
  Downloading https://files.pythonhosted.org/packages/7b/34/9d8e8e917baebabe
     |████████████████████████████████| 8.7MB 6.1MB/s
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packa
Collecting pyahocorasick
  Downloading https://files.pythonhosted.org/packages/7f/c2/eae730037ae1cbbf
     |████████████████████████████████| 327kB 36.5MB/s
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist
Requirement already satisfied: tensorflow>=2.0.0 in /usr/local/lib/python3.7
Requirement already satisfied: python-crfsuite in /usr/local/lib/python3.7/d
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/pyth
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist
Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.7/dis
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/d
Requirement already satisfied: gast==0.3.3 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dis
Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3
Requirement already satisfied: typing-extensions~=3.7.4 in /usr/local/lib/py
Requirement already satisfied: h5py~=2.10.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7
Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.7/
Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: grpcio~=1.32.0 in /usr/local/lib/python3.7/di
Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/
Requirement already satisfied: tensorboard~=2.4 in /usr/local/lib/python3.7/
Requirement already satisfied: tensorflow-estimator<2.5.0,>=2.4.0 in /usr/lo
Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7
Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/l
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/d
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/pytho
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/loca
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/pytho
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.7
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/pytho
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in /us
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/pyth
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/py
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python
```

```
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/d
Building wheels for collected packages: pyahocorasick
  Building wheel for pyahocorasick (setup.py) ... done
  Created wheel for pyahocorasick: filename=pyahocorasick-1.4.2-cp37-cp37m-l
  Stored in directory: /root/.cache/pip/wheels/3a/03/34/77e3ece0bba8b86bfac8
Successfully built pyahocorasick
Installing collected packages: pyahocorasick, sefr-cut
Successfully installed pyahocorasick-1.4.2 sefr-cut-1.1
```

```
import pandas as pd
import pythainlp
import sefr_cut
import gensim
# import pyLDAvis.gensim
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
pyLDAvis.enable_notebook()
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

```
df = pd.read_csv('CustomerReviews2.csv')
```

```
df.head(5)
```

	Review ID	Restaurant_ID	Restaurant	User	Headline
0	1	352696Px-mo-mo-paradise-เดอะมอลล์-บางกะปิ	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	7b16469831074f7abc7824745ee75212	ที่สำคัญของร้านนี้คือ บริการดีมาก พนักงานน่ารักส...
1	2	352696Px-mo-mo-paradise-เดอะมอลล์-บางกะปิ	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	pakkaramonpondej	รสชาติเหมือนทุกสาขา แต่สาขานี้บริการดี ที่นั่งดี
		352696Px-mo-	Mo-Mo-Paradise (โม		ชาบูพรีเมี่

Tokenize Words

```
stopwords = list(pythainlp.corpus.thai_stopwords())
removed_words = [' ', '  ', '\n','\n\n', 'ร้าน', '(', ')' , '          ','—', '!',
screening_words = stopwords + removed_words

sefr_cut.load_model(engine='ws1000')
def tokenize_with_space(sentence):
  merged = ''
```

```
    # words = pythainlp.word_tokenize(str(sentence), engine='newmm')
    words = sefr_cut.tokenize(sentence)
    for word in words[0]:
      if word not in screening_words:
        word = word.rstrip("\n")
        word = word.rstrip("\u200b")
        if word is None or word == ' ' or word == '' or word ==':':
          continue
        else:
          merged = merged + ',' + word
      elif word is None:
        continue
    return merged[1:]

    # return words
```

```
    loading model.....
    Success
```

```
df['Review_tokenized'] = df['Review'].apply(lambda x: tokenize_with_space(x))
```

```
df.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 21 entries, 0 to 20
    Data columns (total 9 columns):
     #   Column            Non-Null Count  Dtype  
    ---  ------            --------------  -----  
     0   Review ID         21 non-null     int64  
     1   Restaurant_ID     21 non-null     object 
     2   Restaurant        21 non-null     object 
     3   User              21 non-null     object 
     4   Headline          21 non-null     object 
     5   Review            21 non-null     object 
     6   Rating            20 non-null     float64
     7   สิ่งที่เราคิด     19 non-null     object 
     8   Review_tokenized  21 non-null     object 
    dtypes: float64(1), int64(1), object(7)
    memory usage: 1.6+ KB
```

```
df['Review_tokenized'].iloc[10]
```

```
    'อัพเดทราคา,ชาบูชิ,ตอน,399,บาท,net,ทาน,1.,15,ชม.,น้ำซุป,เลือก,4,รส,ซูชิลาย,ตา,เทมปุระ,ท
    าน เพลิน ดี ยืน ทาน ไลน์ ควา บุฟเฟ่ต์ กุ้ง สด เน เนื้อ สไลด์ หลัก อิ่ม คับ'
```

```
df.tail()
```

	Review ID	Restaurant_ID	Restaurant	User	Headline
16	17	436045MJ-ข้าน้อยขอชาบู	ข้าน้อยขอชาบู	ployynp	บุฟเฟ่ต์ชาบู และพิซซ่า ไม่อั้นใน ราคา 19 บาท เน้..
17	18	436045MJ-ข้าน้อยขอชาบู	ข้าน้อยขอชาบู	27a91236fe5e4559a4f097c97a480781	ร้านบุฟเฟ่ต์ ราคา มิตรภาพ อยู่ชั้น4 ติด โรงหนัง ..
18	19	436045MJ-ข้าน้อยขอชาบู	ข้าน้อยขอชาบู	0b81d251e4db486f9bcdba73b374ed99	ของหลาก หลาย ปนๆ งงๆ นิด หน่อย

Create Dictionary

ผลลัพธ์ชาบู บุฟเฟ่หมผัก

```
# documents = df['Review_tokenized'].to_list()
documents = df['Review_tokenized']
texts = [[text for text in doc.split(',')] for doc in documents]
dictionary = gensim.corpora.Dictionary(texts)
```

```
print(dictionary.token2id.keys())
```

```
    'ตัก', 'ตัว', 'ถั่ว', 'ถ่าย', 'นั่ง', 'นุ่ม', 'นุ่มลิ้น', 'น้ำจิ้ม', 'บด', 'บาร์ผัก', 'ฟูกุ', 'ภาษา
```

```
gensim_corpus = [dictionary.doc2bow(text, allow_update=True) for text in texts]
word_frequencies = [[(dictionary[id], frequence) for id, frequence in couple] for c
```

Topic Modeling

```
num_topics = 5
chunksize = 4000 # size of the doc looked at every pass
passes = 20 # number of passes through documents
iterations = 100
eval_every = 1  # Don't evaluate model perplexity, takes too much time.

# Make a index to word dictionary.
temp = dictionary[0]  # This is only to "load" the dictionary.
id2word = dictionary.id2token
```

```
%time model = gensim.models.LdaModel(corpus=gensim_corpus, id2word=id2word, chunksi
```

```
%time model = gensim.models.LdaModel(corpus=gensim_corpus, id2word=id2word, chunksi
                      alpha='auto', eta='auto', \
                      iterations=iterations, num_topics=num_topics, \
                      passes=passes, eval_every=eval_every)
```

```
CPU times: user 395 ms, sys: 1.49 ms, total: 397 ms
Wall time: 397 ms
```

```
# pyLDAvis.gensim.prepare(model, gensim_corpus, dictionary)
gensimvis.prepare(model, gensim_corpus, dictionary)
```

Review Topic


```
list_topic = list()
for i in range(0,num_topics):
  print('topic:',i)
  list_topic.append(set(convert_to_topic(model.show_topic(i,20))))
  print(list_topic[i])
```

```
topic: 0
{'เลือก', 'เนื้อ', 'อาหาร', 'ราคา', 'ซูชิ', 'กุ้ง', 'จิ้ม', 'บาท', 'รสชาติ', 'เมนู', 'สด'
topic: 1
{'เวลา', 'เลือก', 'เนื้อ', 'อาหาร', 'ซุป', 'ราคา', 'กุ้ง', 'กิน', 'บาท', 'รสชาติ', 'สด
topic: 2
{'', 'เลือก', 'เนื้อ', 'อาหาร', 'ชอบ', 'ซุป', 'ราคา', 'กิน', 'ไก่', 'รสชาติ', 'สด', '
topic: 3
{'สาย', 'เลือก', 'เนื้อ', 'อาหาร', 'หน้า', 'ราคา', 'กิน', 'หม้อ', 'ติม', 'ค่า', 'บาท',
topic: 4
{'เลือก', 'เนื้อ', 'อาหาร', 'ชอบ', 'ราคา', 'พนักงาน', 'กิน', 'ชาบู', 'บริการ', 'คุ้ม',
```

Find unique Keyword each topic

```
c = list()
for i in range(0,num_topics):
  a = list_topic[i]
  b = set()
  for j in range(0,num_topics):
    if i != j:
      b = b.union(list_topic[j])
  a = a-b
  c.append(list(a))
  print(i,a)
```

```
0 {'เมนู', 'เทมปุระ', 'ซูชิ', 'บุฟเฟ่ต์', 'จิ้ม'}
1 {'เวลา'}
2 {'', 'หมู', 'ไก่', 'ลอง'}
3 {'สาย', 'หม้อ', 'ติม', 'หน้า', 'ค่า', '3'}
4 {'ชาบู', 'บริการ', 'คุณภาพ', 'พนักงาน', 'นึก', 'นั่ง'}
```

Find Keyword Intersection for each topic

```
d = list_topic[0]
print(d)
for i in range(1,num_topics):
  # print(list_topic[i])
  d = d.intersection(list_topic[i])
  print(i,d)
```

```
{'เลือก', 'เนื้อ', 'อาหาร', 'ราคา', 'ซูชิ', 'กุ้ง', 'จิ้ม', 'บาท', 'รสชาติ', 'เมนู', 'สด'
1 {'เลือก', 'เนื้อ', 'อาหาร', 'สด', 'อร่อย', 'คุ้ม', 'คน', 'บาท', 'น้ำ', 'ราคา', 'รสช
2 {'เลือก', 'เนื้อ', 'อาหาร', 'สด', 'คุ้ม', 'ดี', 'น้ำ', 'ราคา', 'รสชาติ', 'ทาน', 'อร่อ
3 {'เลือก', 'เนื้อ', 'อาหาร', 'อร่อย', 'คุ้ม', 'น้ำ', 'ราคา', 'ทาน', 'ดี'}
4 {'เลือก', 'เนื้อ', 'อาหาร', 'คุ้ม', 'ดี', 'น้ำ', 'ราคา', 'ทาน', 'อร่อย'}
```

```
def convert_to_keyword(x):
  new_x = str(c[x])
  return new_x
```

```
def convert_to_topic(x):
  new_x = [i[0] for i in x]
  return new_x
```

```
df['topics'] = df['Review_tokenized'].apply(lambda x: model.get_document_topics(dic
df['score'] = df['Review_tokenized'].apply(lambda x: model.get_document_topics(dict
df['topic_name'] = df['Review_tokenized'].apply(lambda x: model.show_topic(model.ge
df['topic_name2'] = df['topic_name'].apply(lambda x:convert_to_topic(x))
df['keyword'] = df['topics'].apply(lambda x:convert_to_keyword(x))
# df2 = df['Review_tokenized'].apply(lambda x: model.get_document_topics(dictionary
```

```
df[['Restaurant', 'Review', 'topics','score','topic_name2','keyword']]
```

0	ไม พาราไดซ์) เดอะมอลล์ บางกะปิ	บริการดีมากพนักงานน่ารักส...	1	0.998467	เนื้อ, ทาน, ราคา, บา...	['เวลา']
1	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	นึกถึงชาบูญี่ปุ่นยังไงก็ต้อง คิดถึงโมโม่ พาราไ...	4	0.997341	[ดี, เนื้อ, ทาน, อร่อย, สาขา, น้ำ, เลือก, ราคา...	['ชาบู', 'บริการ', 'คุณภาพ', 'พนักงาน', 'นึก',...
2	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	มาทานช่วงนี้ สามารถนั่งโต๊ะเดียวกัน หม้อเดียวก...	2	0.999215	[เนื้อ, น้ำ, อร่อย, ทาน, เลือก, , ดี, ราคา, อา...	['', 'หมู', 'ไก่', 'ลอง']
3	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	ถ้านึกถึงชาบูที่มีเนื้อเน้นๆ ในราคาไม่โหดจนเกิ...	4	0.995929	[ดี, เนื้อ, ทาน, อร่อย, สาขา, น้ำ, เลือก, ราคา...	['ชาบู', 'บริการ', 'คุณภาพ', 'พนักงาน', 'นึก',...
4	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	เดินมาหน้าร้านแล้วได้กลิ่นชาบูหอมมาก ๆ ประกอบ...	4	0.996865	[ดี, เนื้อ, ทาน, อร่อย, สาขา, น้ำ, เลือก, ราคา...	['ชาบู', 'บริการ', 'คุณภาพ', 'พนักงาน', 'นึก',...
5	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	ร้านบุฟเฟ่ ชาบูแนวญี่ปุ่น สายเนื้อหมู เนื้อวัว...	2	0.996423	[เนื้อ, น้ำ, อร่อย, ทาน, เลือก, , ดี, ราคา, อา...	['', 'หมู', 'ไก่', 'ลอง']
6	Mo-Mo-Paradise (โม โม พาราไดซ์) เดอะมอลล์ บางกะปิ	Number 20 : โมโม – พาราไดส์ (สาขาเดอะมอลบางกะป...	0	0.997597	[ทาน, อาหาร, ดี, น้ำ, เลือก, เนื้อ, กุ้ง, ราคา...	['เมนู', 'เทมปุระ', 'ซูชิ', 'บุฟเฟ่ต์', 'จิ้ม']

```
df[[ 'Review', 'topics','score','topic_name2','keyword']].to_csv('out.csv', index=F
```

7	โม พาราไดซ์)	ดีมากคุ้มค่าเหมาะสมกับ	1	0.997976	เนื้อ, ทาน,	['เวลา']

Result

8	เดอะมอลล์บางกะปิ ชัน	ตอนหิว ไม่งันจะไม่ค้ม	1	0.996851	เนื้อ, ทาน,	['เวลา']

จากการทดสอบพบว่า Topic Name สามารถแบ่งได้เป็น 5 Topic จากที่ดูจาก LDA Bubble chart พบว่าสามารถแบ่งกลุ่ม ได้แยกจากกันมากที่ ซึ่งโดย ทุกๆ Topic มี Keyword ร่วมกัน โดยจากการทำ intersect topic name กับทุกๆ Topic คือ

'เลือก', 'เนื้อ', 'อาหาร', 'คุ้ม', 'ดี', 'น้ำ', 'ราคา', 'ทาน', 'อร่อย' ซึ่งในแต่ละ Review โดยส่วนใหญ่ มีพูดถึง เนื้อ อาหาร น้ำซุป ราคาอาหาร กิน ดี มีของหวาน แต่ก็จะมีบาง Topic จะมีเอกลักษณ์ที่พูดถึงแตกต่างกันไป ซึ่งดูจาก Different keywork ของแต่ละ Topic

- Topic 0 พูดถึง เมนูเทมปุระ ซูชิ น้ำจิ้ม (accurancy : 55%)
- Topic 1 พูดถึง เวลา (accurancy : 28%)
- Topic 2 พูดถึง หมู ไก่ ลองกิน (accurancy : 70%)
- Topic 3 พูดถึง สาย หม้อ ไอติม (accurancy : 100%)
- Topic 4 พูดถึง ชาบู บริการ คุณภาพ พนักงาน (accurancy : 46%)

ดังนั้นแต่ละ Topic ควรชื่อว่า

- Topic 0 คุ้มดีเนื้ออร่อย มีเทมปุระ ซูชิ และน้ำจิ้ม
- Topic 1 คุ้มดีเนื้ออร่อย มีเวลา
- Topic 2 คุ้มดีเนื้ออร่อย มีลองกินหมูไก่
- Topic 3 คุ้มดีเนื้ออร่อย มีหม้อมีไอติม
- Topic 4 คุ้มดีชาบูอร่อย พนักงานบริการคุณภาพ

```
#Evaluate
for i in c:
  g = list()
  for j in list(i):
    # print(i,j)
    g.append(df[df["keyword"].str.contains(j)&df["Review"].str.contains(j)]['Review
  print(i,"\naccurancy:",sum(g)/len(g))
```

```
['เมนู', 'เทมปุระ', 'ซูชิ', 'บุฟเฟ่ต์', 'จิ้ม']
accurancy: 0.55
['เวลา']
accurancy: 0.2857142857142857
['', 'หมู', 'ไก่', 'ลอง']
accurancy: 0.7
['สาย', 'หม้อ', 'ติม', 'หน้า', 'ค่า', '3']
accurancy: 1.0
['ชาบู', 'บริการ', 'คุณภาพ', 'พนักงาน', 'นึก', 'นั่ง']
accurancy: 0.4583333333333333
```

✓ 0s completed at 02:15