실습 2 - 리뷰데이터로 토픽모델링 - LDA

[1] 0초
```
import warnings
# 경고메세지 끄기
warnings.filterwarnings(action='ignore')

# 참고) 다시 출력하게 하기
# warnings.filterwarnings(action='default')
```

[2] 2초
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

[3] 0초
```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/강의자료/핀인사이트_강의자료 /2023 강의자료 업데이트/자연어처리/실습/data/review.csv', sep = '\t')
# df = df[:100]
```

토큰화 및 데이터 전처리

[4] 0초
```
# !pip install konlpy
```

[5] 1분
```
#Okt
from konlpy.tag import Okt
tokenizer= Okt()

df['morphs'] = None
for i,row in df.iterrows():
   df['morphs'][i] = ' '.join(tokenizer.morphs(row['document']))
```

[ ] 0초
```
df.head()
```

	id	document	label	morphs
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0	아 더빙 .. 진짜 짜증나네요 목소리
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1	흠 ... 포스터 보고 초딩 영화 줄 .... 오버 연기 조차 가볍지 않구나
2	10265843	너무재밋었다그래서보는것을추천한다	0	너 무재 밋었 다그 래서 보는것을 추천 한 다
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0	교도소 이야기 구먼 .. 솔직히 재미 는 없다 .. 평점 조정
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1	사이 몬페 그 의 익살스런 연기 가 돋보였던 영화 ! 스파이더맨 에서 늙어 보이기만...

[7]
```python
import re
# '한글'을 제외한 다른 문자 모두 제거
remove_except_ko = re.compile(r"[^가-힣ㄱ-ㅎㅏ-ㅣ\\s]")

def preprocess_remove(text):
    text = re.sub(remove_except_ko,' ',text).strip()
    return text

df['morphs'] = df['morphs'].map(lambda x : preprocess_remove(x))
df.head()
```

	id	document	label	morphs
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0	아 더빙 진짜 짜증나네요 목소리
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1	흠 포스터 보고 초딩 영화 줄 오버 연기 조차 가볍지 않구나
2	10265843	너무재밋었다그래서보는것을추천한다	0	너 무재 밋었 다그 래서 보는것을 추천 한 다
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0	교도소 이야기 구먼 솔직히 재미 는 없다 평점 조정
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1	사이 몬페 그 의 익살스런 연기 가 돋보였던 영화 스파이더맨 에서 늙어 보이기만...

[8]
```python
# 불용어처리
stop_word = ['포스터','저작권자','한경','닷컴','뉴스룸','홈페이지','바로','네이버','채널','구독','세요','제공'
             ,'께서','라고','입니다','습니다','한다','에서','이다','에게','으로','이랑','까지','부터','하다','한데','통해','위해','때문']

def preprocess(text):
    text = text.split()
    text = [i for i in text if len(i)>1]
    text = [i for i in text if i not in stop_word]
    return text
```

[9]
```python
text ='김현진 기자 안녕하세요 공'
preprocess(text)
```

```
['김현진', '기자', '안녕하세요']
```

```python
# 토큰화 + 토큰리스트 생성
def make_tokens(df):
    df['tokens'] = ''
    for i, row in df.iterrows():
        if i%1000==0:
            print(i,'/',len(df))
        token = preprocess(df['morphs'][i])
        df['tokens'][i] = ' '.join(token)
    return df

df = make_tokens(df)
```

```
0 / 10000
1000 / 10000
2000 / 10000
3000 / 10000
```

```
[10] 4000 / 10000
     5000 / 10000
     6000 / 10000
     7000 / 10000
     8000 / 10000
     9000 / 10000
```

```
[11] df.head()
```

	id	document	label	morphs	tokens
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0	아 더빙 진짜 짜증나네요 목소리	더빙 진짜 짜증나네요 목소리
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1	흠 포스터 보고 초딩 영화 줄 오버 연기 조차 가볍지 않구나	보고 초딩 영화 오버 연기 조차 가볍지 않구나
2	10265843	너무재밋었다그래서보는것을추천한다	0	너 무재 밋었 다그 래서 보는것을 추천 한 다	무재 밋었 다그 래서 보는것을 추천
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0	교도소 이야기 구먼 솔직히 재미 는 없다 평점 조정	교도소 이야기 구먼 솔직히 재미 없다 평점 조정
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1	사이 몬페 그 의 익살스런 연기 가 돋보였던 영화 스파이더맨 에서 늙어 보이기만...	사이 몬페 익살스런 연기 돋보였던 영화 스파이더맨 늙어 보이기만 했던 커스틴 던스트...

```
[12] neg = (df[df['label']==0]).reset_index(drop = True)
     pos = (df[df['label']==1]).reset_index(drop = True)
     print('neg:',len(neg))
     print('pos:',len(pos))
```

```
neg: 5021
pos: 4979
```

```
[13] df.head()
```

	id	document	label	morphs	tokens
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0	아 더빙 진짜 짜증나네요 목소리	더빙 진짜 짜증나네요 목소리
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1	흠 포스터 보고 초딩 영화 줄 오버 연기 조차 가볍지 않구나	보고 초딩 영화 오버 연기 조차 가볍지 않구나
2	10265843	너무재밋었다그래서보는것을추천한다	0	너 무재 밋었 다그 래서 보는것을 추천 한 다	무재 밋었 다그 래서 보는것을 추천
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0	교도소 이야기 구먼 솔직히 재미 는 없다 평점 조정	교도소 이야기 구먼 솔직히 재미 없다 평점 조정
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1	사이 몬페 그 의 익살스런 연기 가 돋보였던 영화 스파이더맨 에서 늙어 보이기만...	사이 몬페 익살스런 연기 돋보였던 영화 스파이더맨 늙어 보이기만 했던 커스틴 던스트...

```python
from gensim import corpora
from gensim.models import LdaModel, TfidfModel

tokenized_docs = neg['tokens'].apply(lambda x : x.split())

id2word = corpora.Dictionary(tokenized_docs)
corpus_TDM = [id2word.doc2bow(doc) for doc in tokenized_docs]
tfidf = TfidfModel(corpus_TDM)
corpus_TFIDF = tfidf[corpus_TDM]

n = 10 # 토픽갯수 - 변경가능
lda = LdaModel(corpus=corpus_TFIDF,
               id2word=id2word,
               num_topics=n,
               random_state=100)

for t in lda.print_topics() :
  print(t)
```

```
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the number of passes or iterations to improve accuracy
(0, '0.007*"없다" + 0.005*"영화" + 0.004*"지루하다" + 0.004*"하나" + 0.004*"기세" + 0.003*"역시" + 0.003*"눈물" + 0.003*"재미" + 0.003*"진짜" + 0.003*"시간"')
(1, '0.012*"영화" + 0.004*"결말" + 0.004*"별루" + 0.003*"소설" + 0.003*"였다" + 0.003*"그닥" + 0.003*"관객" + 0.003*"너무" + 0.003*"그만" + 0.003*"별로"')
(2, '0.006*"ㅋㅋ" + 0.004*"영화" + 0.004*"인지" + 0.004*"나옴" + 0.004*"노잼" + 0.003*"스토리" + 0.003*"마지막" + 0.003*"그냥" + 0.003*"대사" + 0.003*"개봉"')
(3, '0.005*"이딴" + 0.004*"영화" + 0.003*"같은" + 0.003*"아무" + 0.003*"마무리" + 0.002*"런가" + 0.002*"쓰래" + 0.002*"극치" + 0.002*"영화로" + 0.002*"재미없"')
(4, '0.009*"영화" + 0.008*"최악" + 0.004*"쓰레기" + 0.004*"만든" + 0.004*"처음" + 0.004*"진짜" + 0.003*"정말" + 0.003*"후회" + 0.003*"졸작" + 0.003*"이런"')
(5, '0.009*"영화" + 0.006*"재미없다" + 0.006*"너무" + 0.004*"이런" + 0.004*"시간" + 0.004*"없이" + 0.004*"평점" + 0.003*"점도" + 0.003*"ㅡㅡ" + 0.003*"무슨"')
(6, '0.005*"이건" + 0.005*"영화" + 0.005*"진짜" + 0.004*"평점" + 0.004*"재미없는" + 0.003*"ㅋㅋㅋㅋ" + 0.003*"아깝다" + 0.003*"재미없어" + 0.003*"없는" + 0.003*"노답"')
(7, '0.006*"영화" + 0.004*"코믹" + 0.004*"졸라" + 0.004*"스토리" + 0.003*"전부" + 0.003*"그냥" + 0.003*"싸구려" + 0.003*"명성" + 0.002*"정말" + 0.002*"연기력"')
(8, '0.012*"쓰레기" + 0.006*"영화" + 0.005*"시간" + 0.004*"아깝다" + 0.004*"재미" + 0.004*"정말" + 0.003*"진짜" + 0.003*"ㅠㅠ" + 0.003*"보단" + 0.003*"지루해"')
(9, '0.009*"별로" + 0.005*"점수" + 0.004*"영화" + 0.004*"보다" + 0.004*"어색한" + 0.003*"공포영화" + 0.003*"지루하고" + 0.003*"평점" + 0.003*"스토리" + 0.003*"그대로"')
```

시각화

```python
# !pip install pyLDAvis==3.4.1
```

```python
import pyLDAvis
import pyLDAvis.gensim

pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda, corpus_TFIDF, id2word)
pyLDAvis.display(vis)
```

Selected Topic: 0 Previous Topic Next Topic Clear Topic

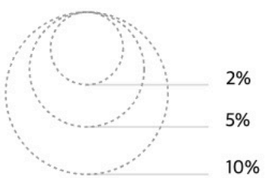
Slide to adjust relevance metric:(2)

λ = 1

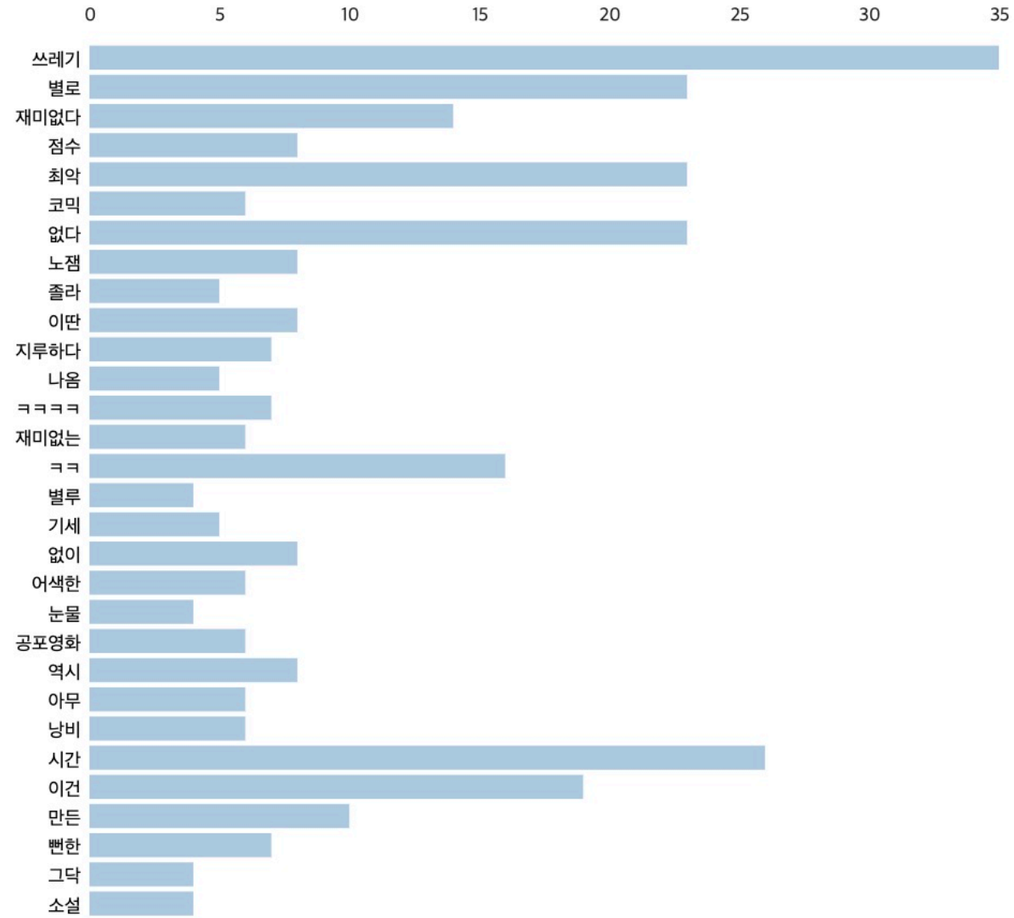
0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)

Marginal topic distribution



Top-30 Most Salient Terms(1)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

```python
from gensim import corpora
from gensim.models import LdaModel, TfidfModel
import pyLDAvis
import pyLDAvis.gensim

tokenized_docs = neg['tokens'].apply(lambda x : x.split())

id2word = corpora.Dictionary(tokenized_docs)
corpus_TDM = [id2word.doc2bow(doc) for doc in tokenized_docs]
tfidf = TfidfModel(corpus_TDM)
corpus_TFIDF = tfidf[corpus_TDM]

n = 3 # 토픽갯수 - 변경가능
lda = LdaModel(corpus=corpus_TFIDF,
               id2word=id2word,
               num_topics=n,
               random_state=100)

for t in lda.print_topics() :
  print(t)

pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim.prepare(lda, corpus_TFIDF, id2word)
pyLDAvis.display(vis)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future.
  and should_run_async(code)
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the number of passes or iterations to improve accuracy
(0, '0.005*"영화" + 0.004*"없다" + 0.003*"이런" + 0.003*"진짜" + 0.003*"시간" + 0.003*"재미" + 0.003*"너무" + 0.002*"이건" + 0.002*"재미없다" + 0.002*"평점"')
(1, '0.007*"영화" + 0.003*"별로" + 0.002*"아깝다" + 0.002*"너무" + 0.002*"스토리" + 0.002*"감독" + 0.002*"만든" + 0.002*"정말" + 0.002*"없는" + 0.002*"하는"')
(2, '0.004*"쓰레기" + 0.004*"영화" + 0.003*"최악" + 0.002*"그냥" + 0.002*"진짜" + 0.002*"보다" + 0.002*"스토리" + 0.002*"——" + 0.002*"같은" + 0.002*"ㅋㅋ"')
```

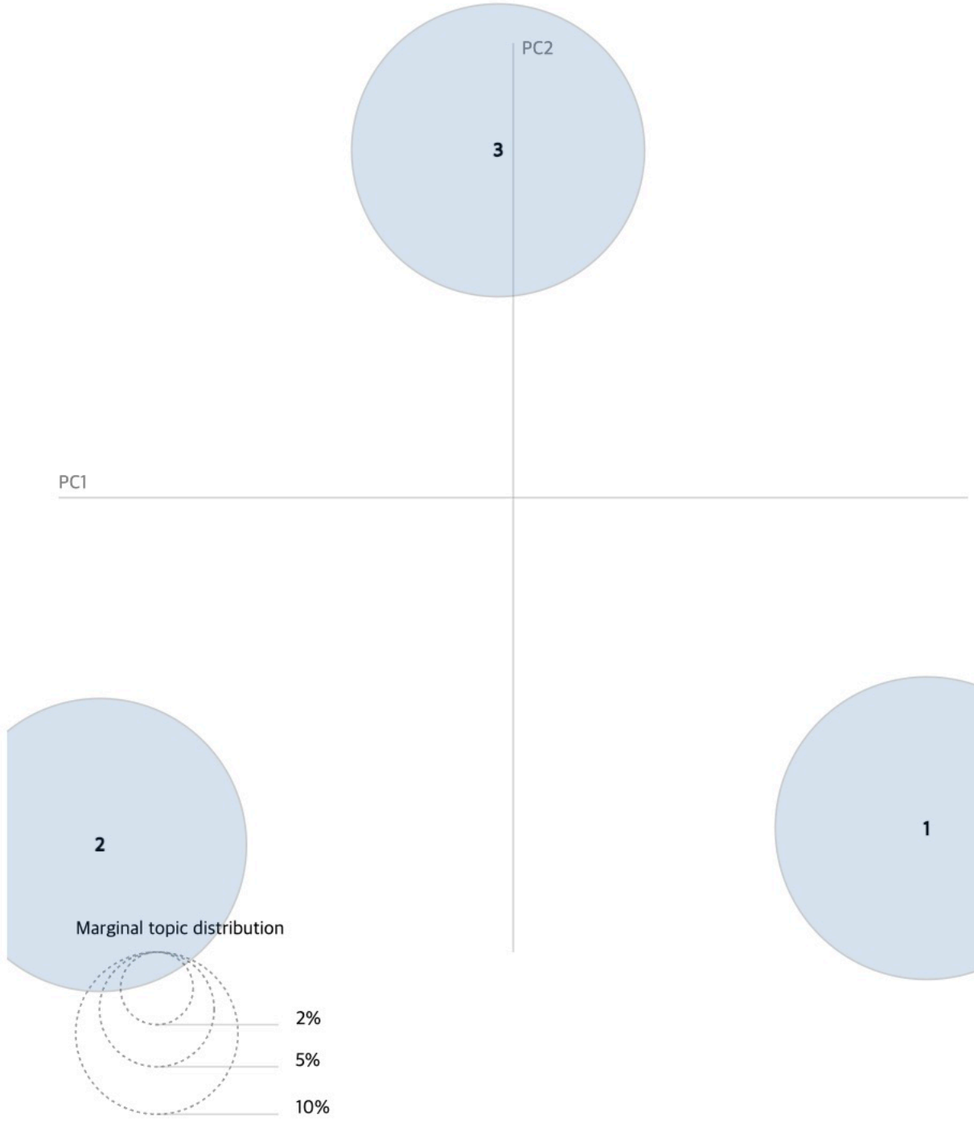
Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric:(2)

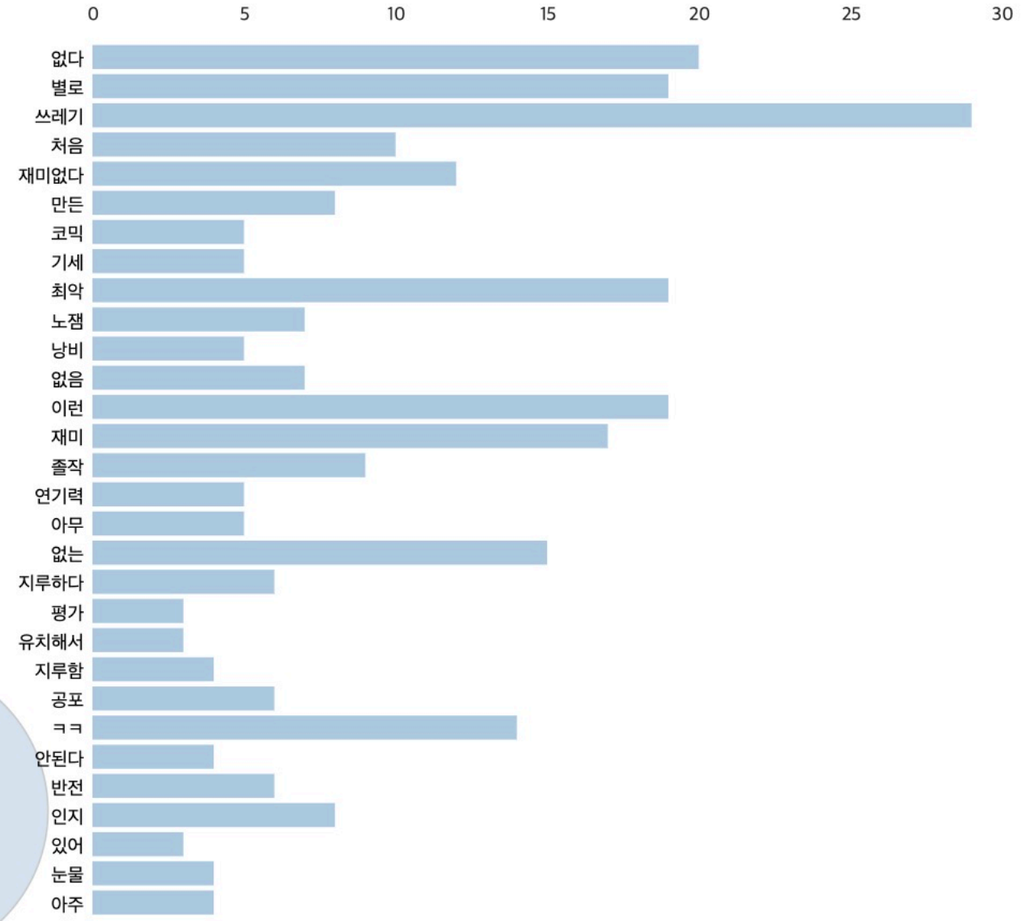
λ = 1

0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)



Top-30 Most Salient Terms(1)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)