빅데이터 분석 방법론 소개

토픽모델링 – LDA (**L**atent **D**irichlet **A**llocation)

본 방법론은 Datalab site에 게시된 보고서 중
《비대면 시대》 분석 내용에서 실제 사용된 방법론으로 도출 과정에 대한 이해를 돕고자
방법론의 개념, 접근 방향, 코딩 내용 등을 간략하게 정리하여 공유

Methodology

토픽모델링 – LDA(Latent Dirichlet Allocation)

추출한 문서에 담긴 단어들의 주제(토픽)을 추출하는 '토픽모델링' 기법 중 하나인
「잠재디리클레할당」 방법론 내용정리

Latent Dirichlet Allocation

「잠재 디리클레 할당」

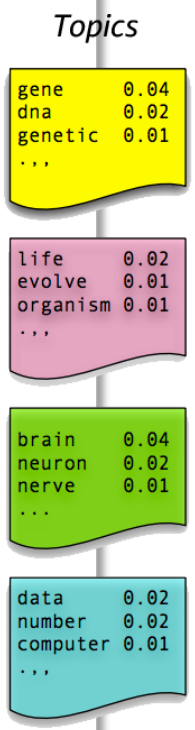
LDA를 토픽모델링(Topic Modeling) 기법이라고 부르는데
단어나 문서의 숨겨진 주제(Topic)을 찾아내 주기 때문.

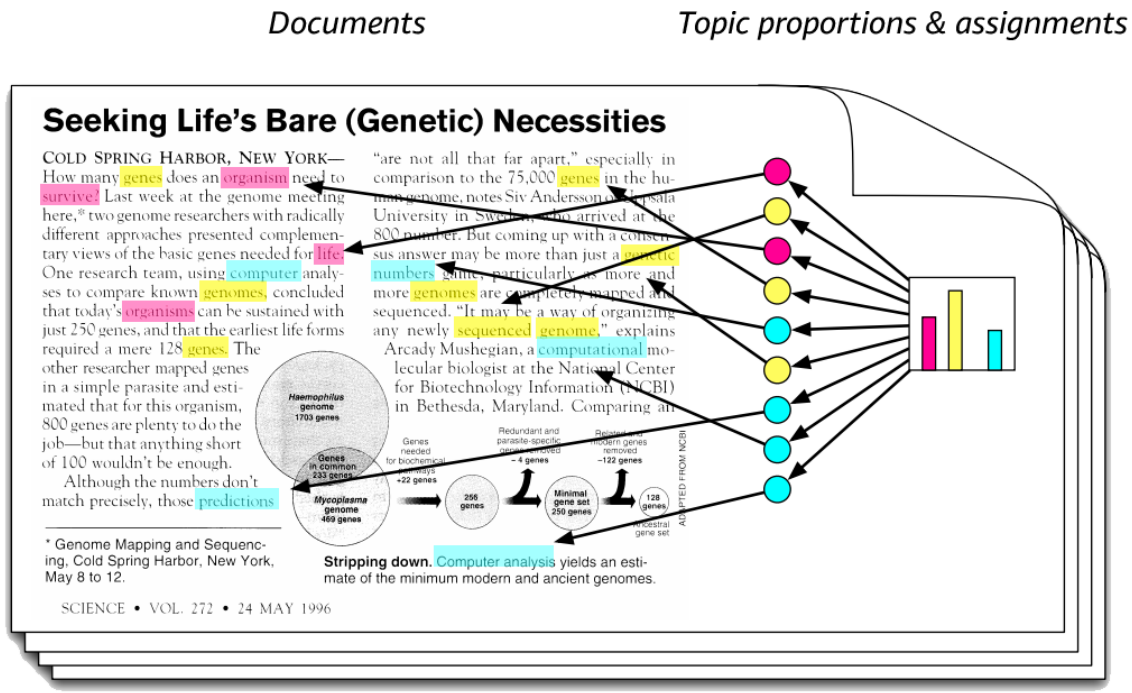
여기서 Topic(주제)란, 수집된 원문 내용에 담긴 다양한
키워드를 기반으로 내용을 유형화(그룹화) 시켜주는 것

추출한 원문에는 다양한 내용이 담겨 있을 수 있는데
이러한 주제들을 일일이 수작업으로 분류하기 어렵기 때문에
LDA 같은 분류 방법을 적용해 전반적인 데이터의 구조를
먼저 파악하는 것이 중요

LDA 기법은 단순히 주제만 분류해주는 것이 아니라
주제에 포함되는 키워드들을 보여주기 때문에
그 키워드들로 해당 주제를 해석하고 정의할 수 있음

물론, 사전에 충분한 데이터의 정제 과정이 요구되며
기계적 분류인 만큼 결과 자체가 완전하지 않을 수도 있어
어느 정도의 후보정이 필요할 수도 있음





목차 소개

본 문서는 다음과 같은 순으로 작성

Python Code로 보는 LDA 진행 과정

(1) 데이터 불러오기 및 텍스트 데이터 형식 처리

(2) 정규표현식 및 Konlpy 이용한 명사 추출

(3) gensim을 통해 Corpus(말뭉치) Dictionary(사전) 언어모델 형성

(4) Perplexity 및 Coherence을 통한 모델 평가 및 토픽 최적화

(5) 하이퍼 파라미터 선정 및 LDA 시각화

(6) 토픽에 할당된 키워드 및 문서 추출

Python Code로 보는 LDA 진행 과정

(1) 데이터 불러오기 및 텍스트 데이터 형식 처리

《 코딩 예시 》

```
In [1]: import numpy as np
        import pandas as pd
        import re
        import matplotlib.pyplot as plt
```

```
In [2]: etc = pd.read_csv("C:/Users/PC/Desktop/0702untact/교육키워드/5월_전체_교육_0702.csv",encoding="cp949")
```

```
In [3]: len(etc)
Out[3]: 4234
```

```
In [4]: etc.head(3)
```

Out[4]:

	no	id	date	month	title	text	channel	sitename
0	1	THE배움English	NaN	4	-	원격수업이 시험적으로 진행중입니다. 참고하세요! #원격수업 #온라인개학	인스타	-
1	2	Minseok Kim	NaN	4	-	키야 날씨 좋오오땅. 원격강의 마이크 고장나서 캡슐하고 산책중. 농땡이....	인스타	-
2	3	i-Yeon	NaN	4	-	비대면 택견전수교육 동영상 개인수련 과제 제출요망 세상은 멈추지 않았다 경로를 바꾸...	인스타	-

```
In [4]: etc2 = list(etc.text)

        etc2[0:2]
```

```
Out[4]: ['원래 컴퓨터에 크게 집착하지 않았고 할 기회도 잘 없었던 아들이에요. 그런데 이번 코로나사태로 온라인 개학을 하게 되면서 컴퓨터를 쥐어줬더니 컴퓨터의 세계에 빠진 것 같아요. 제가 워킹맘이라 일일이 통제를 하지 못했지만 이미 제 눈에 몇번이나 들킨 것만 해도 유투브, 브롤스타즈, 줌앱으로 친구들과 놀이 등.. 아시다시피 온라인 수업이라는 것이 쌍방향이 아니다보니 수업은 대충 듣고 이런 식으로 컴퓨터로 노는 것 같아요 ㅠㅠ 개학 해도 주1회 등교라 앞으로도 계속 컴퓨터는 쥐어줘야 할텐데, 어떻게 통제하는 것이 좋을까요? 그 동안 몇차례나 크게 혼냈지만, 그 때 뿐이고 크게 약빨도 없네요.(저희도 컴퓨터 휴대폰 만지면 너무 재미있는데 아이라고 오죽하겠어요 ㅠㅠ) ₩n이런 시대를 피할 수 없는 이상, 언젠가는 본인이 알아서 통제할 능력이 생길테니(?) 일단 너무 심각하게 생각하지 말고 두고 볼까요, 아니면 절대 못하게 해야 할까요? ₩n다른 집은 어떻게 하고 있는지, 다른 또래 아이들은 어떤지 경험담 공유해주세요 ㅠㅠ ₩n ₩n ₩n ₩n₩n₩n ₩n ₩n ₩n ₩n ₩n미취학 자녀나 교육과 무관한 내용, 게시판 성격에 맞지 않는 글은 삭제됩니다. ₩n주의 사항 및 카페 규정을 꼭 숙지하셔서 힘들게 쓰신 글, 삭제나 활동제한 되지 않도록 주의바랍니다. ₩n글 삭제나 활동제한 처리된 내용을 게시글로 올리면 즉시 강제퇴장 조치합니다. ₩n ₩n[주의] 다음에 해당될 경우 게시글 무통보 삭제 및 회원 활동제한 (강등,활동정지,강제퇴장) 처리됩니다. ₩n선생님,학생 구함,추천글/아이디 공유,추천글/물품판매,구매,무료구입,무료나눔의글/학원이용후기,프로그램소개/특정인을찾거나 특정인에게 문의하는글/타카페,블로그,사이트 소개,초대,링크글/체험단,서평후기,블로그,타카페 서평스크랩글/메일,쪽지 등으로 질문,상담유도,상담요청글/카페 이용 문의,건의글,카페부정,운영불만/음원,저작권자료 구함글/교사나 학생등 타인 험담/19금내용,혐오글/정치적 논란이 될수있는 글/기타 교육카페와 맞지 않는 게시글,덧글,스크랩물',
```

[활용 라이브러리]

- pandas (데이터 형식 및 데이터 저장 라이브러리)
- re 텍스트 전처리 (R의 gsub()기능과 동일)
- matplotlib 추후 perplexity 모델 평가 시각화 제공

《 실행 가이드 》

- '라이브러리' 불러오기
- 판다스 함수로 csv파일 불러온 후 etc로 저장
- len() 데이터 길이, head()를 통한 데이터 확인
- etc의 텍스트 부분을 리스트로 만든 후 etc2에 저장 후 확인
  리스트에 저장 하기 전 가끔씩 문자열로 인식 하지 않는 오류가 있어서 해당 명령어로 바꿔줍니다.
  etc.text = etc.text.astype("str")
- [게시글1], [게시글2],…[게시글n] 형태의 데이터 구조
  추후에 게시글 단위로 문서를 나누고 해당 문서의 의미를 찾아내기 위해 리스트로 분리해서 나눠줘야 하기 때문.

Python Code로 보는 LDA 진행 과정

(2) 정규표현식 및 Konlpy 이용한 명사 추출

《 코딩 예시 》

In [5]:
```
## 구두점 제거
from string import punctuation
def strip_punctuation(s):
    return ''.join(c for c in s if c not in punctuation)

clean_title = []
for sent in etc2 :
    clean = strip_punctuation(sent)
    clean_title.append(clean)

clean_title[0:1]
```

Out[5]:
```
['원래 컴퓨터에 크게 집착하지 않았고 할 기회도 잘 없었던 아들이에요 그런데 이번 코로나사태로 온라인 개학을 하게 되면서 컴퓨터를 쥐어줬더니 컴퓨터의 세계에 빠진 것 같아요 제가 워킹맘이라 일일이 통제를 하지 못했지만 이미 제 눈에 몇번이나 들킨 것만 해도 유투브 브롤스타즈 줌 앱으로 친구들과 놀이 등 아시다시피 온라인 수업이라는 것이 쌍방향이 아니다보니 수업은 대충 듣고 이런 식으로 컴퓨터로 노는 것 같아요 ㅠㅠ 개학해도 주1회 등교라 앞으로도 계속 컴퓨터는 쥐어줘야 할텐데 어떻게 통제하는 것이 좋을까요 그 동안 몇차례나 크게 혼냈지만 그 때 뿐이고 크게 약빨도 없네요저희도 컴퓨터 휴대폰 만지면 너무 재미있는데 아이라고 오죽하겠어요 ㅠㅠ ㅠn이런 시대를 피할 수 없는 이상 언젠가는 본인이 알아서 통제할 능력이 생길테니 일단 너무 심각하게 생각하지 말고 두고 볼까요 아니면 절대 못하게 해야 할까요 ㅠn다른 집은 어떻게 하고 있는지 다른 또
```

In [6]:
```
from konlpy.tag import Kkma
kkma = Kkma()
```

In [7]:
```
import time
import datetime
start = time.time()
```

In [8]:
```
### 한글 알파벳 외에 공백
dataset = [ ]
for i in range(len(clean_title)) :
    dataset.append(kkma.nouns(re.sub('[^가-힣a-zA-Z\ws]', '', clean_title[i])))


sec = time.time()-start
times = str(datetime.timedelta(seconds=sec)).split(".")
times=times[0]
print(times)
```

[활용 라이브러리]
- python 한국어 자연어 전처리(Natural Language Preprocessing) 전용
- konlpy 라이브러리 불러오기 (window 환경 시 jpype, jdk 환경변수 생성 선행 필요)
- konlpy 형태소 분석기 중 하나인 Kkma(일명 : 꼬꼬마) 불러오기

《 실행 가이드 》

- string 라이브러리 중 **punctuation**는 구두점을 제거하는데 필요
- 해당 라이브러리 통해 "…" / " , " / "?" / "()" 등 특수문자 제거 후 clean_title의 빈 공간에 하나씩 리스트로 저장.
- Kkma를 실행하기 위해서는 따로 Kkma()를 다른 변수에 선언 해주어야 함 왼 쪽 예시의 경우 kkma 변수로 Kkma() 명령어 저장
- 로직이 실행되는 시간파악을 위해 **time** 라이브러리 불러오기
- 정리해보면,
  01. 한글, 영문을 제외한 모든 특수 문자 제거 후 clean_title에 하나씩 저장
  02. 리스트로 저장된 clean_title에 kkma_nouns()로 명사 추출
  03. dataset.append 에 하나씩 for문 clean_title의 길이만큼 저장
  04. 시간 체크

Python Code로 보는 LDA 진행 과정

(3) gensim을 이용한 언어모델(Language Model) 생성

《 코딩 예시 》

```
In [40]: clean_title2 = []
         for i in range(len(clean_title)) :
             clean_title2.append(re.sub('주차',' ',clean_title[i]))

         clean_title3 = []
         for i in range(len(clean_title2)) :
             clean_title3.append(re.sub('스마트',' ',clean_title2[i]))
```

```
In [41]: clean_title4 = []
         for i in range(len(clean_title3)) :
             clean_title4.append(re.sub('민영화',' ',clean_title3[i]))
```

```
In [42]: clean_title6 = []
         for i in range(len(clean_title5)) :
             clean_title6.append(re.sub('요즘',' ',clean_title5[i]))
```

```
In [9]: #!pip install gensim
        from gensim import corpora, models
        import gensim
```

```
In [10]: high_score_reviews=dataset
```

```
In [11]: high_score_reviews=[[y for y in x if not len(y)==1]
                            for x in high_score_reviews]
         dictionary=corpora.Dictionary(high_score_reviews)
         corpus=[dictionary.doc2bow(text) for text in high_score_reviews]
```

《 실행 가이드 》

- 특정 문자를 제거 후 " " 공백 만들기
  본 과정은 단어의 비중이 너무 큰 단어들 혹은 의미 없는 단어들에 대해
  덜어내고 하위 키워드들을 꺼낼 때 사용 (선 토픽모델링 시각화 후 다시 전처리)

- gensim 라이브러리 불러오기 (LDA 적용을 위한 텍스트의 벡터화)
  (문자열 길이가 1이 아닐 때 y를 x만큼 넣고,
  x를 high_score_reviews에 리스트 하나씩 저장)

01. dataset으로 genism 모델 학습 시키기
02. 데이터를 dictionary형태로 명사 리스트 만들기
03. 명사 형태의 문서별로 말뭉치 만들기

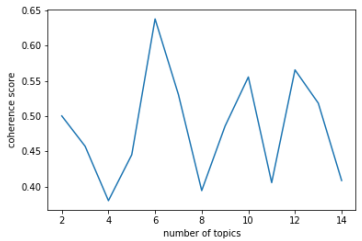
Python Code로 보는 LDA 진행 과정

(4) Perplexity 및 Coherence을 통한 모델 평가 및 토픽 최적화

《 코딩 예시 》

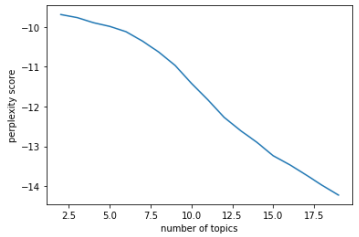
```
In [12]: import matplotlib.pyplot as plt
         from gensim.models import CoherenceModel

         coherence_values=[]
         for i in range(2,15):
             ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=i, id2word=dictionary)
             coherence_model_lda=CoherenceModel(model=ldamodel, texts=high_score_reviews, dictionary=dictionary,topn=10)
             coherence_lda=coherence_model_lda.get_coherence()
             coherence_values.append(coherence_lda)
```

```
In [13]: x=range(2,15)
         plt.plot(x, coherence_values)
         plt.xlabel("number of topics")
         plt.ylabel("coherence score")
         plt.show()
```




```
In [111]: import matplotlib.pyplot as plt
          perplexity_values=[]
          for i in range(2,20):
              ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=i, id2word=dictionary)
              perplexity_values.append(ldamodel.log_perplexity(corpus))
```

```
In [112]: x=range(2,20)
          plt.plot(x, perplexity_values)
          plt.xlabel("number of topics")
          plt.ylabel("perplexity score")
          plt.show()
```



《 실행 가이드 》

- **CoherenceModel** 을 통한 토픽 최적화

  의미 : 토픽이 얼마나 의미론적으로 일관성 있는지 판단.
  높을수록 의미론적 일관성 높음

  주 용도 : 해당 모델이 얼마나 실제로 의미 있는 결과를 내는지 확인

  기존에 언어모델 평가로 CoherenceModel만을 사용 후
  원하는 토픽 개수의 Coherence 모델을 지속 학습 시켜 토픽을 할당 했으나,
  추후에는 두 가지 모델 함께 적용 해보는 것과 좀 더 정밀한 사용이 필요

- **언어 모델 평가 방법**
  퍼플렉서티(perplexity) : PPL로 줄여서 표현
  'perplexed : 헷갈리는' 과 유사한 의미

  선정된 토픽 개수마다 학습시켜
  가장 낮은 값을 보이는 구간을 찾아
  최적화된 토픽의 개수 선정 가능

  의미 : 확률 모델이 결과를 얼마나 정확하게 예측하는지 판단.
  낮을수록 정확하게 예측.

  주 용도 : 동일 모델 내 파라미터에 따른 성능 평가할 때 주로 사용

  한계 : Perplexity가 낮다고 해서, 결과가 해석 용이하다는 의미가 아님

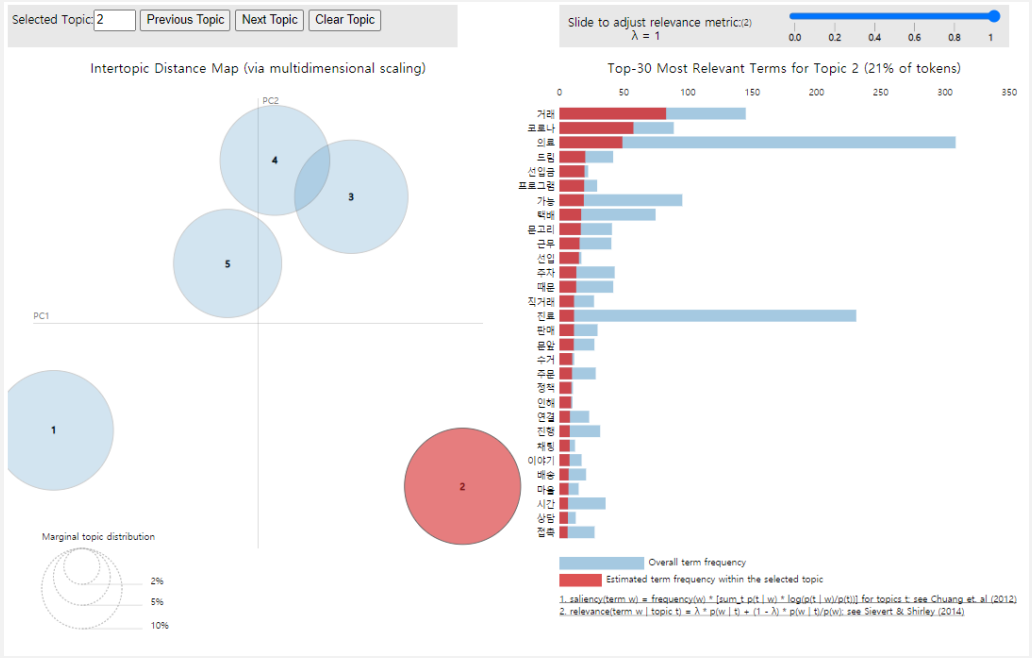
Python Code로 보는 LDA 진행 과정

(5) 하이퍼 파라미터 선정 및 LDA 시각화

《 코딩 예시 》

```
In [14]: # 위의 결과를 바탕으로 갯수 수정하기
         ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=6,alpha=0.1,
                                                  id2word=dictionary)

In [15]: ldamodel.print_topics(num_words=20)

Out[15]: [(0,
           '0.012*"비대면" + 0.009*"파트너" + 0.008*"비결" + 0.008*"도매" + 0.008*"초보자" + 0.008*"강의" + 0.007*"시간" + 0.007*"마케팅" + 0.006*"창업" + 0.006*"수업" + 0.006*"상담" + 0.006*"때문" + 0.006*"포기" + 0.005*"코로나" + 0.005*"환불" + 0.005*"천만원" + 0.005*"주세" + 0.005*"문의" + 0.005*"문의주세" + 0.005*"학교"'),
```

```
In [16]: import pyLDAvis
         import pyLDAvis.gensim

         pyLDAvis.enable_notebook()
         vis = pyLDAvis.gensim.prepare(ldamodel, corpus, dictionary)
         vis
```



《 실행 가이드 》

- 학습된 코퍼스(말뭉치)로 토픽 개수를 선정하고
  alpha, eta, iterations, cunk_size등 다양한 파라미터 적용 가능.

  pyLDAvis를 불러온 뒤 학습된 **모델 시각화** 진행

  파라미터의 조정에 있어서 크게 alpha,beta 값을 조정하게 되는데,
  이에 따라 토픽 내 분포하는 문서, 단어의 분포가 달라짐.

  내가 원하는 주제들을 설명할 수 있는 단어들을 끌어 내기 위해
  파라미터 값을 설정하기도 하므로 해당 단어들의 이해도나
  문서에 담긴 도메인 이해도가 중요

  *추가로 해당 토픽에 묶인 단어들이 사용자 사전에 정의되지 않았다면
  사용자 사전을 추가하여 단어를 등록하는 과정도 필요

Python Code로 보는 LDA 진행 과정

(6) 토픽에 할당된 키워드 추출(Topic-Keyword)

《 코딩 예시 》

```
In [104]: kk=ldamodel.show_topic(7,topn=80000)
          type(ldamodel.show_topic)
          kk2=pd.DataFrame(kk)
          kk2.head(3)
```

Out[104]:

	0	1
0	비대면	0.050368
1	원격	0.042939
2	거래	0.017551

```
In [105]: kk2.to_csv("C:/Users/PC/Desktop/0702untact/5월_1121/5월_top7.csv",encoding="euc-kr")
```

차지 비율	비대면				원격					
빈도 순위	12.40%		10.80%		16.40%		8.20%		6.10%	
	2번토픽 중고거래		3번토픽 택배		1번토픽 진료		6번토픽 회의		9번토픽 조작	
1	비대면	0.208	비대면	0.293	원격	0.320	원격	0.181	원격	0.152
2	거래	0.099	문앞	0.038	진료	0.071	회의	0.079	접속	0.088
3	코로나	0.076	연결	0.021	원격진료	0.042	직거래	0.035	원격접속	0.037
4	가능	0.055	업무	0.020	의료	0.039	조종	0.034	조작	0.021
5	때문	0.035	입금	0.020	조정	0.027	프로그램	0.031	허용	0.014
6	비대면거래	0.027	지금	0.017	원격의료	0.026	촬영	0.024	친구	0.013
7	문고리	0.027	우리	0.013	근무	0.018	시스템	0.022	분만	0.012
8	계좌	0.022	원격	0.012	원격조정	0.017	사진	0.015	데스크	0.011
9	개설	0.020	배송	0.012	회사	0.014	이용	0.014	나라	0.011
10	원해	0.016	마을	0.011	원격근무	0.010	설치	0.012	화면	0.010
11	진행	0.015	은행	0.011	오늘	0.010	처음	0.011	연락	0.010
12	신청	0.012	만원	0.010	시간	0.009	반대	0.010	내가	0.009
13	코로나때문	0.012	언택트	0.007	접촉	0.008	가능	0.008	일본	0.007
14	확인	0.006	희망	0.007	사용	0.008	핸드폰	0.007	리모트	0.006
15	선입금	0.006	안전	0.007	화상	0.008	이제	0.007	모니터	0.006

《 실행 가이드 》

- 01. 각 토픽에 할당된 단어와 단어별 토픽 차지 비율 추출
  02. 데이터 프레임화
  03. 토픽별 추출

▶ 분류된 여러 토픽들에 포함된 키워드를 정렬한 뒤,
상위 키워드들의 분포비 등을 확인하여 개별 토픽 정의

1. Python Code로 보는 LDA 진행 과정

(6) 토픽에 할당된 문서 추출(Topic-Document)

《 코딩 예시 》

In [ ]:
```
def make_topictable_per_doc(ldamodel, corpus):
    topic_table = pd.DataFrame()

    # 몇 번째 문서인지를 의미하는 문서 번호와 해당 문서의 토픽 비중을 한 줄씩 꺼내온다.
    for i, topic_list in enumerate(ldamodel[corpus]):
        doc = topic_list[0] if ldamodel.per_word_topics else topic_list
        doc = sorted(doc, key=lambda x: (x[1]), reverse=True)
        # 각 문서에 대해서 비중이 높은 토픽순으로 토픽을 정렬한다.
        # EX) 정렬 전 0번 문서 : (2번 토픽, 48.5%), (8번 토픽, 25%), (10번 토픽, 5%), (12번 토픽, 21.5%),
        # Ex) 정렬 후 0번 문서 : (2번 토픽, 48.5%), (8번 토픽, 25%), (12번 토픽, 21.5%), (10번 토픽, 5%)
        # 48 > 25 > 21 > 5 순으로 정렬이 된 것.

        # 모든 문서에 대해서 각각 아래를 수행
        for j, (topic_num, prop_topic) in enumerate(doc): #  몇 번 토픽인지와 비중을 나눠서 저장한다.
            if j == 0:  # 정렬을 한 상태이므로 가장 앞에 있는 것이 가장 비중이 높은 토픽
                topic_table = topic_table.append(pd.Series([int(topic_num), round(prop_topic,4), topic_list]), ignore_index=True)
                # 가장 비중이 높은 토픽과, 가장 비중이 높은 토픽의 비중과, 전체 토픽의 비중을 저장한다.
            else:
                break
    return(topic_table)
```

In [ ]:
```
topictable = make_topictable_per_doc(ldamodel, corpus)
topictable = topictable.reset_index() # 문서 번호을 의미하는 열(column)로 사용하기 위해서 인덱스 열을 하나 더 만든다.
topictable.columns = ['문서 번호', '가장 비중이 높은 토픽', '가장 높은 토픽의 비중', '각 토픽의 비중']
topictable[:10]
```

In [ ]:
```
topictable.to_csv("C:/Users/PC/Desktop/언택트0612/언택트_데이터셋/filtering/0624/금융쪽_topdoc.csv",encoding="euc-kr", index=False)
```

문서	토픽 할당
- #코로나사태로 독일시간에 맞춰 #컨퍼런스콜통역???? - 덕분에 한국	[(0, 0.7973208), (5, 0.17924346)]
- - #이시국에도#빨래는밀릴수없으니 #이불#겨울패딩#카페트 #비대	[(0, 0.42611736), (1, 0.24619609), (3, 0.23134981), (5, 0.08434527)]
- - 요즘의 소소한 기쁨 반달이네가 잘 먹고 지낸다는 소식. 모근도 배	[(0, 0.023117501), (1, 0.023076052), (2, 0.023105042), (3, 0.023091806), (4, 0.023060927), (5, 0.88454866)]
- . 벚꽃아 안녕 이제 질때쯤에나 하남에 올 수 있지 않을까..? 4월도 얌전	[(0, 0.3561982), (5, 0.6155945)]
- . 흑흑 펭럽유~♥ . . 비대면 프로그램 뭐가 있을까아~~!! 고민이당 ㅎ	[(0, 0.10012159), (1, 0.36596334), (4, 0.23822391), (5, 0.2907562)]
- ? 출처 : SBS ???????????????? ???????????????? 다행이다... ^^ 그래..	[(0, 0.116856836), (1, 0.7966681), (2, 0.021603242), (3, 0.021607071), (4, 0.02160876), (5, 0.021656016)]
- ? 코로나19로 인해 외출이 줄어들면서 택배 물량이 늘었다는 기사를	[(0, 0.88548166), (3, 0.07831252)]
- ??+1137 오랜만에 도서관에서 빌려온 승호그림책 비대면으로 메일로	[(4, 0.058807787), (5, 0.9281445)]
- 3월2일 주문한 온라인몰배송이 오늘도착했다~ 선주문해놓은다음에	[(5, 0.9709885)]
우선 코로나 바이러스 자체가 기존에 없던 바이러스이기 때문에, 백신	[(0, 0.013227261), (1, 0.5933204), (2, 0.013198367), (3, 0.0131958295), (4, 0.013370102), (5, 0.35368803)]
- 블링블링 우리엄마손?? 우리백여사 소녀손갈네~~~~~이뽀라?? . . #:	[(0, 0.68535036), (1, 0.011013876), (2, 0.010939854), (3, 0.010943127), (4, 0.090560995), (5, 0.19119182)]
- 노을 보러 왔다가 주차비 500원어치만 퀵하게 보고 갑니다. 원격근무	[(0, 0.5105037), (3, 0.20149256), (5, 0.26879886)]
켜놓고 잠시 다른거 하고 봤더니 재부팅 되고 있더라구요.	[(1, 0.80704874), (5, 0.1773027)]
이런기분 처음이기도하고 있다하더라도 잘안올리는데	[(1, 0.90688676), (5, 0.070708856)]
- 두번째 꼬꼬밀 박스?? 공지 드려요. 제가 좋아하는 '블리커베이크샵' 으	[(0, 0.12518172), (4, 0.5044976), (5, 0.35113603)]
- 매일매일 똑같은일상에 채윤이는 지겨워하고 집에만 있는 나도 지쳐	[(0, 0.14041145), (1, 0.10928929), (4, 0.32200304), (5, 0.41721088)]

《 실행 가이드 》

- 각 문서별로 토픽에 할당되는 토픽 번호와 차지하는 비중을 만들기 위한 코드

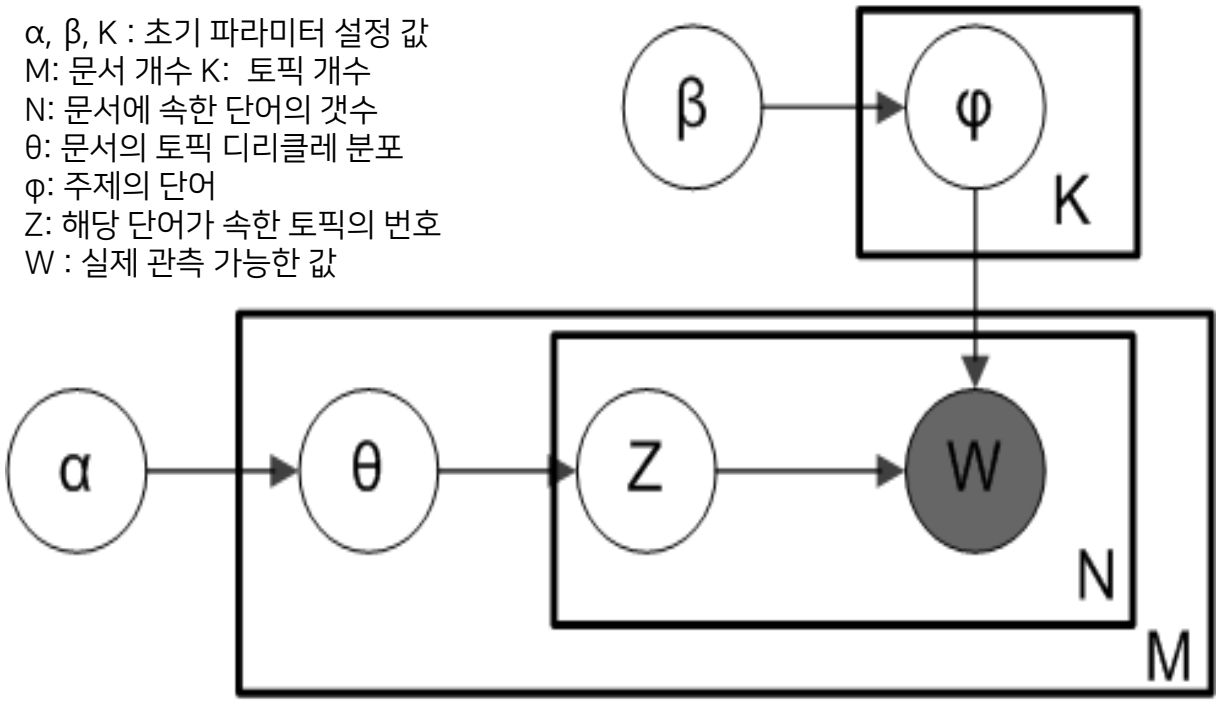
- 해당 코드를 통해 문서 개별로 가장 크게 할당된 토픽의 번호와 비율 확인 가능, 여러 토픽에 중첩 할당 된 경우, 개별 할당된 값도 확인 가능

[분석 과정에서의 Idea]

단어별로 토픽 모델링 결과를 잘 나타내고서, 문서별로 다시 묶은 다음
각 주제에 해당하는 문서들끼리만 다시 토픽모델링 결과를 낸다면
하나의 주제에서 또 다르게 얘기하는 주제들을 끄집어 낼 수 있음

LDA 모델

α, β, K : 초기 파라미터 설정 값
M: 문서 개수 K: 토픽 개수
N: 문서에 속한 단어의 갯수
θ: 문서의 토픽 디리클레 분포
φ: 주제의 단어
Z: 해당 단어가 속한 토픽의 번호
W : 실제 관측 가능한 값



원하는 토픽의 개수와 디리클레 분포 조절에 대한 하이퍼 파라미터 설정 값을 조절하고,
문서와 단어를 하나씩 관측해나가며, 문서/단어마다 적절한 토픽을 부여한 뒤 Z값을 정한다.

LDA는 각 단어나 문서의 숨겨진 주제를 찾아내어
문서와 키워드별로 주제끼리 묶어주는 비지도 학습 알고리즘

디리클레(Dirichlet)의 분포를 가정하고
토픽이라는 잠재(Latent)된 변수를 활용하여
각 문서들과 단어들이 토픽에 할당되는 확률 분포를 그린 것.

디리클레의 분포는 확률분포 중 다항분포, 연속확률분포로 중 하나로 정의할 수 있고,
디리클레의 분포의 성질 중 하나는 k차원의 실수 벡터를 모두 더한 값은 1이다.
(토픽의 단어를 모두 더한 값은 1, 토픽들의 요소 값을 모두 더한 값은 1로 정의)

“사전 켤레 확률” 의 성질을 가지고 있다

디리클레분포(Dirichlet Distribution)

이항분포가 아닌, 연속확률 분포 중 하나로
k차원의 실수 벡터 중 벡터의 요소가 양수이며 모든 요소를 더한 값을 1로 하여
확률 값이 정의되는 분포