KU BIG DATA

TEXT MINING

최종 프로젝트 발표

김강우 배건희 문선미 박기찬 이세희

KU BIG DATA

TEXTMINING

01 Study review

02 Prepre-processing

- Crawling
- 각종 처리

03 Pre-processing

- Stopwords 제거
- 토큰 규모 축소
- 문서 규모 축소
- Exception

04 Modeling analysis

- LDA
- Word cloud

TEXT MINING

STUDY REVIEW

- 주요 개념

[숨어있는 정보를 찾아내는 Text Mining]

[Normalization]

“ 대문자를 모두 소문자로

“ ‘ ’를 기준으로 split한 결과

[Stemming]

“ Word의 root word, root from을 찾는 것(어간 추출)

“ 단어 그 자체만을 고려

“ 코딩 결과도 words 1에 있는 단어들의 root word인 list를 추출

Normalization And Stemming

[Text Classification]

“ Input text에 대해서 Labeling 하기

Topic Identification : Politics, Sports, Technology?

Spam detection : Spam or not?

Sentimental analysis(감정분석): Movie review -> positive or negative?

Spelling Correction : Weather or Whether? Color or Colour?

-> Text Classification도 여태까지 알아왔던 Classification과 마찬가지로 ‘Supervised Classification’ !

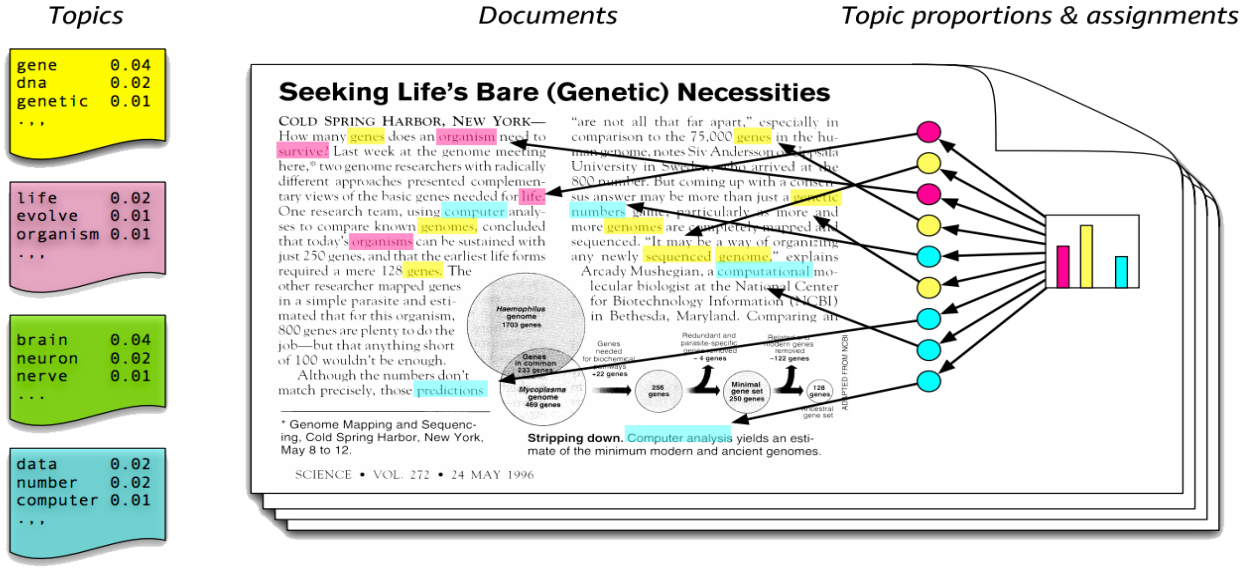
[Bag of Words]

“ Document를 자동으로 분류하기 위한 기법
글에 포함된 단어(Word)들의 분포 -> 이 문서가 어떤 종류의 문서인지를 판단

“ Bag-of-words : Term- Document Matrix
어떠한 text에 포함되어 있는 words에 대해, 순서에 상관없이, 각 words의 사용 빈도 값으로, 바꾸어 주는 방법.

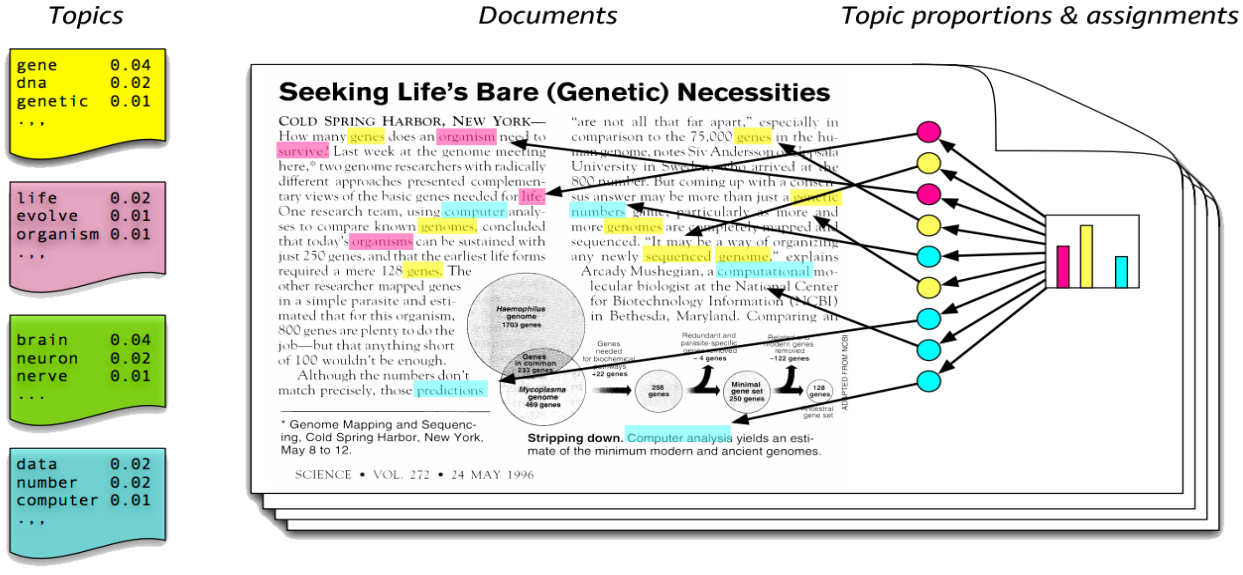
“ Bag-of-words : Representation in a Vector Space
궁극적으로 Frequency of words를 보고, 그 Text의 Contents를 추론하는 방법
많이 사용된 Words가 그 Text의 핵심 단어일 가능성이 높으므로.

[LDA(Latent Dirichlet Allocation)]



"문서와 토픽, 단어와 토픽 간의 **잠재(Latent)**적인 관계들을
디리클레 분포(Dirichlet Dist.)로 나타내고
이들을 활용해 토픽 **할당(Allocation)**

[LDA(Latent Dirichlet Allocation)]



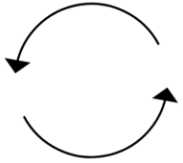
"문서와 토픽, 단어와 토픽 간의 **잠재(Latent)**적인 관계들을
디리클레 분포(Dirichlet Dist.)로 나타내고
이들을 활용해 토픽 **할당(Allocation)**

[LDA(Latent Dirichlet Allocation)]

알고리즘

1. 각 단어에 Random하게 Topic 배정 후 분포 생성

2. 문서와 토픽 / 단어와 토픽 간 디리클레 분포를 추정



3. 단어 별 결합분포를 활용해 단어별 Topic 재배정

수렴할 때까지 **반복**

[Named Entity Recognition task]

어디까지 객체로 인식할 것인지 사전 결정

The patient is a 63-year-old female with a three-year history of bilateral hand numbness and occasional weakness.

Within the past year, these symptoms have progressively gotten worse, to encompass also her feet.

bilateral hand numbness

bilateral hand numbness

hand

TEXT MINING

PREPREPROCESSING

- Crawling
- 각종 처리

Crawling

연대숲 #59299번째 외침:

2018. 5. 2 오전 10:33:16

대—하
대숲!!!!!!!!!!!!!!!!!!
남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게 싫어요!!!!!!!!!! 하나둘도 아니고!!!!!!!!!!
물론 예뻐요 제가 봐도!!!!!!
근데 싫어요 그냥 다 싫어!!!!!!!!!!
나만!!!!!!!! 싫은건가!!!!!!!!!

남자친구야 이거 꼭 봐주길바라 그냥 그렇다구<U+1559>(<U+2022><U+0300><U+2038><U+2022><U+0301><U+2036>)<U+1557>

휴 고마워요 대숲ㅎㅅㅎ

" **Rfacebook** 라이브러리를 이용 .csv로 크롤링

기본처리 : str.replace를 이용하여 불필요한 내용을 ' '(공백문자)로 대체해준다.

연대숲 #59299번째 외침:

2018. 5. 2 오전 10:33:16

대—하
대숲!!!!!!!!!!!!!!!!!!
남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게 싫어요!!!!!!!!!! 하나둘도 아니고!!!!!!!!!!
물론 예뻐요 제가 봐도!!!!!!
근데 싫어요 그냥 다 싫어!!!!!!!!!!
나만!!!!!!!! 싫은건가!!!!!!!!!

남자친구야 이거 꼭 봐주길바라 그냥 그렇다구<U+1559>(<U+2022><U+0300><U+2038><U+2022><U+0301><U+2036>)<U+1557>

휴 고마워요 대숲ㅎㅅㅎ

연대숲 #n번째 외침 제거

```
yonsei['message'] =yonsei['message'].str.replace(r'^(연대숲 +#[0-a9가-힝]+ 외침:)','')
```

날짜, 시간 제거

```
yonsei['message'] = yonsei['message'].str.replace(r'^(₩n₩n₩d₩d₩d₩d.? ₩d?₩d.? ₩d?₩d.? (오전|오후) ₩d?₩d:₩d?₩d:₩d?₩d)','')
```

이모티콘 제거

```
yonsei['message'] = yonsei['message'].str.replace(r'<ed>(<U[A-Z0-9]+>)+','')
```

기본문장처리

대—하
대숲!!!!!!!!!!!!!!!!!!
남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게
싫어요!!!!!!!!!! 하나둘도 아니고!!!!!!!!!!
물론 예뻐요 제가 봐도!!!!!!
근데 싫어요 그냥 다 싫어!!!!!!!!!!
나만!!!!!!!! 싫은건가!!!!!!!!!

남자친구야 이거 꼭 봐주길바라 그냥 그렇다구

휴 고마워요 대숲ㅎㅅㅎ

맨앞 공백 제거

```
yonsei['message'] =
yonsei['message'].str.lstrip()
```

문장을 마치는 부호들을 !로 바꿔주기
(.은 다루기 어려워서 !로 대체)

```
yonsei['message'] = yonsei['message'].str.replace(r'[₩r₩n]','!')
yonsei['message'] = yonsei['message'].str.replace(r'[.]' ,'!')
yonsei['message'] = yonsei['message'].str.replace(r'?' ,'!')
yonsei['message'] = yonsei['message'].str.replace(r'~' ,'!')
```

기본문장처리

대—하!대숲!!!!!!!!!!!!!!!!!!!!남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게 싫어요!!!!!!!!!!! 하나 둘도 아니고!!!!!!!!!!!!물론 예뻐요 제가 봐도!!!!!! !근데 싫어요 그냥 다 싫어!!!!!!!!!!!!나만!!!!!!!! 싫은건가!!!!!!!!!!!!남자친구야 이거 꼭 봐주길바라 그냥 그렇다구 휴 고마워요 대숲 ㅎㅅㅎ

" 중복된 !들 하나로 바꿔주기

```
yonsei['message'] = yonsei['message'].str.replace(r'!+','!')
```

" 문장구분을 위해
!앞의 공백은 붙여주고 !뒤는 한 칸 띄어주기

```
yonsei['message'] = yonsei['message'].str.replace(r'₩s+!', '!')
yonsei['message'] = yonsei['message'].str.replace(r'!', '! ')
```

세부처리

대—하! 대숲! 남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게 싫어요! 하나둘도 아니고! 물론 예뻐요 제가 봐도! ! 근데 싫어요 그냥 다 싫어! 나만! 싫은건가! 남자친구야 이거 꼭 봐 주길바라 그냥 그렇다구 휴 고마워요 대숲 ㅎㅅㅎ

기타 문장부호, 초성 제거

```
yonsei['message'] = yonsei['message'].str.replace(r"[^가-힣0-9a-zA-Z₩s.,/!~]", '')
```

대—하! 대숲!같은 의미없는 인사말 제거

```
yonsei['message'] = yonsei['message'].str.replace(r'^(대—[가-힣]!)' ,'')
yonsei['message'] = yonsei['message'].str.replace(r'^( 대[가-힣]!)' ,'')
```

토큰화: R의 **extractNoun**을 이용하여 명사화하여 토큰으로 만든다.

MESSEGE

남자친구가 인스타에서 다른 여자 인스타스타들을 팔로우하는게 싫어요! 하나둘도 아니고! 물론 예뻐요 제가 봐도! ! 근데 싫어요 그냥 다 싫어! 나만! 싫은건가! 남자친구야 이거 꼭 봐주길바라 그냥 그렇다구 휴 고마워요 대슢

TOKEN

['남자친구', '인스타', '여자', '인스타스타들', '팔로우하는', '하나둘', '저', '근데', '나', '것', '남자친구', '이거', '봐주길바']

TEXT MINING

PREPROCESSING

- STOPWORDS 제거
- 토큰 규모 축소
- 문서 규모 축소
- Exception

STOPWORDS 제거

" STOPWORD?

인터넷 검색 시 검색 용어로 사용하지 않는 단어. 관사, 전치사, 조사, 접속사 등 검색 색인 단어로 의미가 없는 단어이다.

예시

여러분, 저, 것, 다들, 전, 나, 그거, 얘, 기, 너,내, 누군가, 진짜, 데, 제, 분, 거, 수, 의, 은, 는, 이, 가, 들, 중, 듯, 네게, 쟤, 지, 사, 줄, 절, 로, 안, 못, 몇번, 너희, 이젠, 때문, 도, 터, 이후, 관리자님, 안녕, 명, 애, 지금, 우리, 언제, 이번 등

한국어 불용어 리스트

+

대나무숲 특성에 맞게 추가한 불용어

```
# with stop_lists
texts = [[word for word in doc if word not in stop_lists] for doc in documents]
```

빈도수 낮은 단어 제거

```
from collections import defaultdict
frequency = defaultdict(int)
for text in texts:
    for token in text:
        frequency[token] += 1

texts = [[token for token in text if frequency[token] > 10]
         for text in texts]
```

" 10번 이상 등장한 단어만 남기기

토큰수 적은 문서 제거

```
texts = [doc for doc in texts if len(doc) > 5]
```

#21591번째사자후
16학번 여러분 23살 누나 어때요?ㅋㅋㅋㅋ 너무 늙었나...

#21590번째사자후
아무리 관심없다 관심없다해도 자꾸 생각나고 보고 싶은걸 어떡해...

" 토큰 5개 이상으로 구성된 문서만 남기기

[Exception]

“한글파괴 빌런

#21489번째사자후
역씨 한양대다. 지금 보시믄 아시게찌마는 한양인의 시험준비 메캬니즘은 샹당히 조크든요? 보세요. 긴장없이 푸는데 에이뿌를 뚜르냈스니까. 즈응말 공부 잘해요. 제가 그듭 말씀드리지마는 즈른 불리한 상황에스 보닌 샤뿌를 휘두를 쑤 있는 슨슈가 그롷게 만치 안타. 그른데 보닌 소느로 학점을 맨드러 내쓰니까 그즘을 칭찬하지 아늘 수가 없을끄 가타요. 즈는 이롷게 평가를 해요. 쇄계 최증상급 슨슈다. 그롷기 때문에 인뿌라가 중요흔그에요. 그롷기 때문에 돔수업실이 이쓰야 해요. Thumbs up

“2 빌런

#22222번째사자후
17학번2 들어오니 새내기적 2야기가 생각나네요..ㅋㅋㅋ
새내기 때 페2스북을 안해서 대나무숲을 몰랐을 적
저는 학교안에 대나무숲2 정말로 2ㅆ는 줄 알았습니다!
자신의 고민을 포스트2ㅅ에 적어서 대나무에 붙2면
사람2 없는 저녁에 대나무숲지기들2 슬금슬금 와서 포스트2ㅅ을 ㅁ
ㅋㅋㅋ
한양2ㄴ들은 그 주제에 대해서 서로 얘기도 나누고요!
어떤사람들은 고민을 붙2다가 눈2 맞아서 연애를 하고(제 소원..)
선봉고발을 하다가 머리채를 잡는 줄 알았어요..ㅋㅋㅋ

삭제

[Exception]

“ ?

#32579번째울림
2015. 06. 16. 오후 02:09:00
< 성균관대학교 대나무숲>
뿜빠라뿜뿰우힣ㅎ힣힣해해햏ㅎ해해햏해ㅐ해해해해해해햏ㅎㅎ해햏ㅎ얼굴도못생긴것들이잘난척하기느ㅡㅡㅡㄴ적어도내얼굴정도는대ㅐㅐ재ㅐㅐㅐㅐㅐ후루룩짭짭후루룩짭짭맛좋은짜장어색해하지말고야무지게드쎄요들레이요들레이요홓홓ㅎ호호호호호로로ㅗㄹ로ㅗ로롤롤로로로로로ㅗㄹㄹㄹ롤ㄹㄹ로로ㅗ로로로ㅗㅗ로ㅗ롤로로로롤ㄹㄹㄹ로롤롤롤로노아조로롤로로로ㅗㄹ로ㅗ로로로롤ㄹ로로로로로류루루루ㅜ루루루루ㅜ룰ㄹㄹㄹㄹ룰ㄹㄹㄹㄹㄹ룰루루루루루룰루닦지말고씻으세요룰루루룰ㄹ루루루루루루룰루루루룰루커져라라라라라랄랄랄라라ㅏ라라라라라랄랄라라랄라라라라라랄라라랄라얄리얄리얄량셩얄라리얄라라랄랄라랄랄라라랄라ㅏㅏㄹㄹㄹㄹ랄랄랄랄라랄라라라라랄랄랄라파트라슈우우ㅜㅇ우우우웅우ㅜ우우우웅으으ㅡㅜㅇ우ㅜㅇ우우우ㅜㅇㅇ우ㅜㅇ우우ㅜ우ㅜ웅우ㅜㅜ우우우ㅜ웅펑코ㅓㅏㅇ그리고지구는멸망했다고한다.

“ 지나친 감정표현

#20442번째사자후
내일부터 실습시작이에요!!!꺄르르르르캬햐햐햐햐햫햫햫
거의 3달간 푹 쉬고 오랜만에 하려니까 너무너무 가기 두렵네여
우리과 4학년 모두 화이팅햫행해행하햫
#케이스 #퀴즈 #술기 #모스비

“ 띄어쓰기 실종

연애하려면다들직접말해요, 왜여기서다들염장질이에요,
[전공진입후달린다다꺼져내가갈거]

삭제

TEXT MINING

MODELING ANALYSIS

- LDA

[학교별 포스트 수]

```
In [87]: number_dic
Out[87]: {'center': 3096,
          'hanyang': 3129,
          'konkuk': 2087,
          'korea': 3304,
          'kyunghee': 3167,
          'seoul': 4764,
          'sirip': 2914,
          'skk': 3345,
          'sugang': 3182,
          'yonsei': 3147}
```

[기존 Description]

```
In [67]: len(dictionary.token2id)
Out[67]: 161125
```

```
In [86]: len(texts2)
Out[86]: 32135
```

```
In [23]: dictionary.num_pos
Out[23]: 1717748
```

“ # of Unique Token

```
In [235]: len(dictionary.token2id)
Out[235]: 9412
```

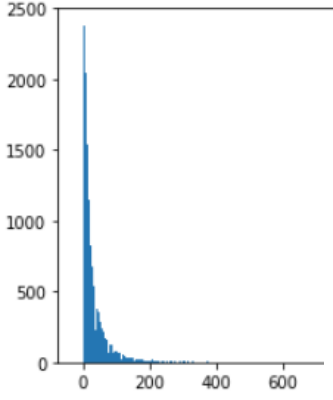
“ # of Total Posts (row)

```
In [236]: dictionary.num_docs
Out[236]: 25738
```

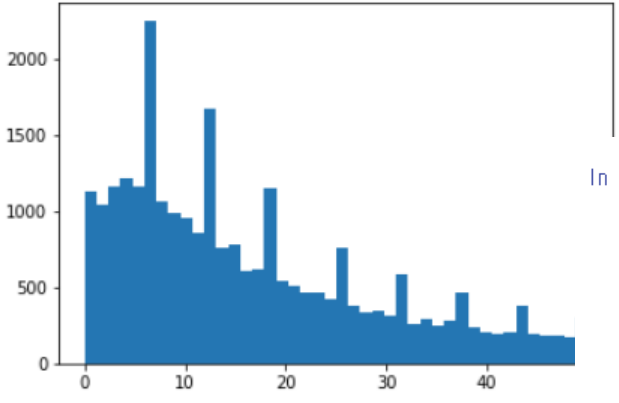
“ # of Total Tokens

```
In [237]: dictionary.num_pos
Out[237]: 926428
```

[Token 수에 따른 Post 수 Histogram]

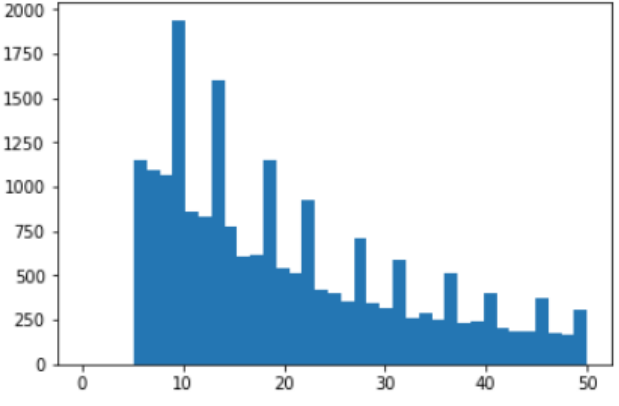



```
In [106]: plt.hist(len_doc, bins = 'auto', range = (0, 50))
          plt.show()
```



전처리 후 -〉

```
In [109]: plt.hist(len_doc, bins = 'auto', range = (0, 50))
          plt.show()
```



[K(Cluster 수) 에 따른 Perplexity 값]

In [239]:
```
grid_k = [3, 5, 7, 10, 15, 25, 30, 40]
vec_perplex = []
```

In [202]:
```
for i in np.arange(len(grid_k)):
    ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=grid_k[i], id2word = c
    perplex = ldamodel.log_perplexity(corpus)
    vec_perplex = vec_perplex + [perplex]

    print(i, "th iteration is done")
    print("perplexity is ", perplex)
```

```
0 th iteration is done
perplexity is  -7.43108611888
1 th iteration is done
perplexity is  -7.39757952516
2 th iteration is done
perplexity is  -7.43762033601
3 th iteration is done
perplexity is  -7.43255894301
4 th iteration is done
perplexity is  -7.45502740344
5 th iteration is done
perplexity is  -7.51579825474
6 th iteration is done
perplexity is  -7.53918962392
7 th iteration is done
perplexity is  -7.60939018584
```

$$Perplexity(w) = exp\left[-\frac{log\{p(w)\}}{\sum_{d=1}^{D}\sum_{j=1}^{V} n^{jd}}\right]$$

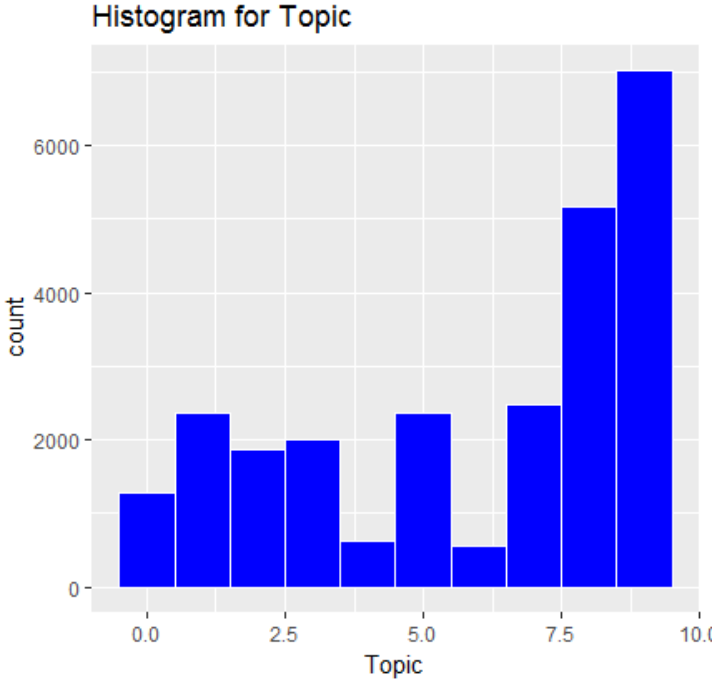
[Extracted Topic]

In [241]:
```
pprint(ldamodel.print_topics())
```

```
[(0,
  '0.044*"엄마" + 0.032*"돈" + 0.028*"아빠" + 0.025*"부모님" + 0.017*"아버지" + 0.017*"말" '
  '+ 0.012*"어머니" + 0.011*"가족" + 0.010*"동생" + 0.010*"생각"'),
 (1,
  '0.030*"선배" + 0.022*"학교" + 0.021*"동아리" + 0.021*"수업" + 0.020*"학생" + '
  '0.016*"교수님" + 0.015*"새내기" + 0.012*"학번" + 0.011*"중앙" + 0.010*"경희"'),
 (2,
  '0.029*"학교" + 0.017*"자리" + 0.013*"기숙사" + 0.012*"공부" + 0.011*"도서관" + '
  '0.010*"버스" + 0.009*"수업" + 0.009*"곳" + 0.008*"소리" + 0.008*"담배"'),
 (3,
  '0.277*"" + 0.035*"이야기" + 0.022*"해" + 0.016*"되" + 0.015*"대나무" + 0.014*"시" + '
  '0.013*"일상" + 0.013*"해주" + 0.011*"하기" + 0.011*"넌"'),
 (4,
  '0.025*"노래" + 0.015*"언니" + 0.014*"게임" + 0.012*"술" + 0.011*"영화" + 0.010*"머리" '
  '+ 0.010*"음악" + 0.010*"형" + 0.009*"이름" + 0.008*"중대"'),
 (5,
  '0.041*"공부" + 0.025*"생각" + 0.023*"학교" + 0.019*"대학" + 0.018*"시험" + 0.014*"누나" '
  '+ 0.012*"학점" + 0.011*"고등학교" + 0.011*"성적" + 0.010*"생활"'),
 (6,
  '0.110*"오빠" + 0.053*"남자친구" + 0.047*"그녀" + 0.036*"여자친구" + 0.016*"연락" + '
  '0.016*"전화" + 0.015*"생일" + 0.013*"죽순" + 0.010*"선물" + 0.009*"크리스마스"'),
 (7,
  '0.026*"생각" + 0.015*"글" + 0.015*"문제" + 0.013*"사람" + 0.012*"말" + 0.008*"의견" + '
  '0.006*"사회" + 0.006*"이유" + 0.006*"여성" + 0.005*"학생들"'),
 (8,
  '0.098*"친구" + 0.037*"남자" + 0.037*"여자" + 0.034*"말" + 0.030*"사람" + 0.029*"생각" '
  '+ 0.018*"연락" + 0.016*"연애" + 0.011*"얘기" + 0.009*"고민"'),
 (9,
  '0.063*"사람" + 0.046*"생각" + 0.039*"사랑" + 0.039*"맘" + 0.030*"마음" + 0.017*"행복" '
  '+ 0.011*"모습" + 0.010*"감정" + 0.009*"미안" + 0.009*"속"')]
```

[시각화 – Word Cloud & Cluster]

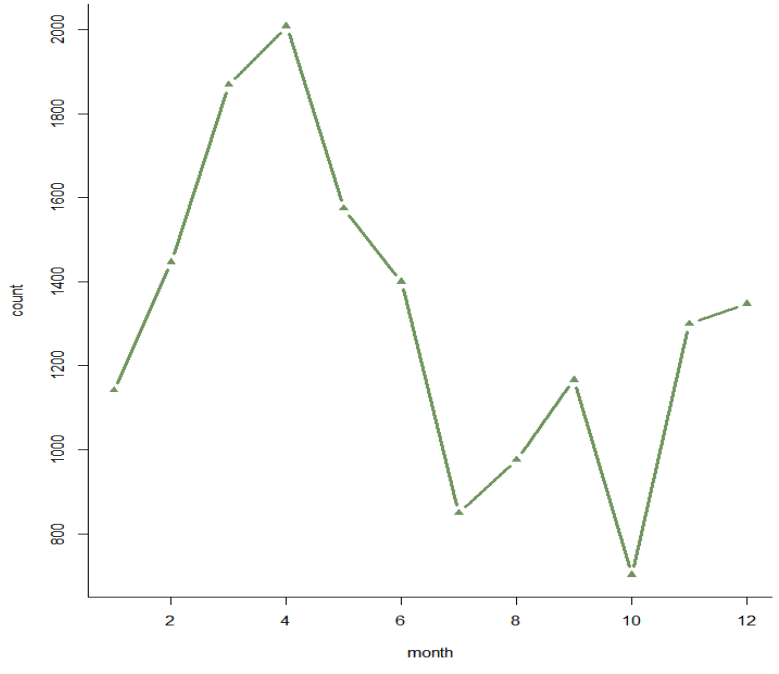
[Topic 상세]

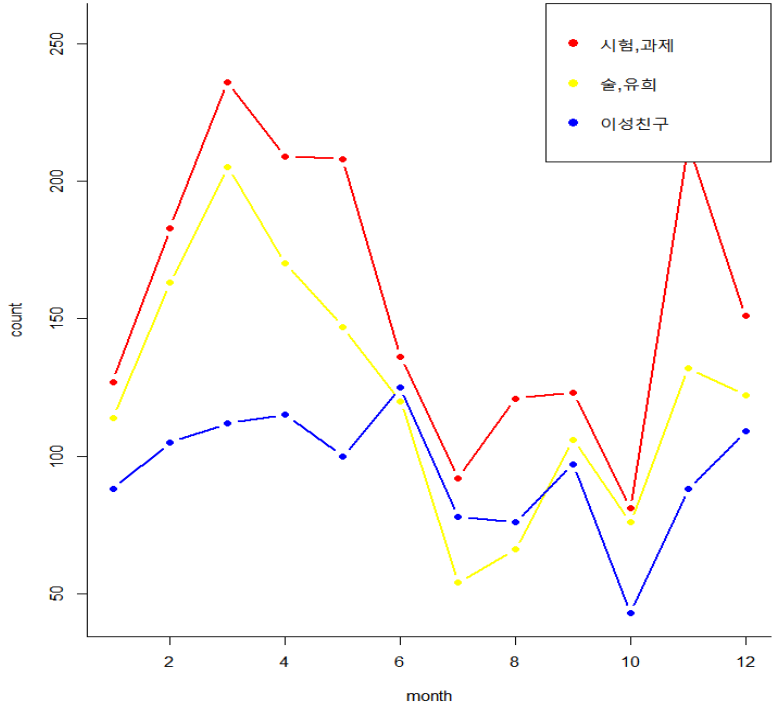


Topic 이름

0 : 가족
1 : 대학생활(꽈생활)
2 : 시설
3 : 일상 (사는 얘기)
4 : 취미 / Entertainment
5 : 학업, 공부
6 : 연애 (썸)
7 : 사회문제
8 : 친구관계, 연애
9 : 사랑 (애인)

[부록 – 월별 이용현황]


count
month
800
1000
1200
1400
1600
1800
2000
2
4
6
8
10
12


count
month
50
100
150
200
250
2
4
6
8
10
12
시험,과제
술,유희
이성친구

[한계 - 아쉬운 점 ㅠ^ㅠ]

한계

- ! 끝없는 (전)전처리 지옥
- ! 대나무숲 빌런들의 글 전처리하기
- ! 말도 안되는 전처리 과정

KU BIG DATA

TEXT MINING

"감사합니다"