**여름방학 4주차 회의록**

안세훈

**1. 크롤링 세부 구현**

원하는 도메인에 원하는 만큼의 리뷰 데이터를 가져올 수 있도록 변수화 및 GUI 만들기

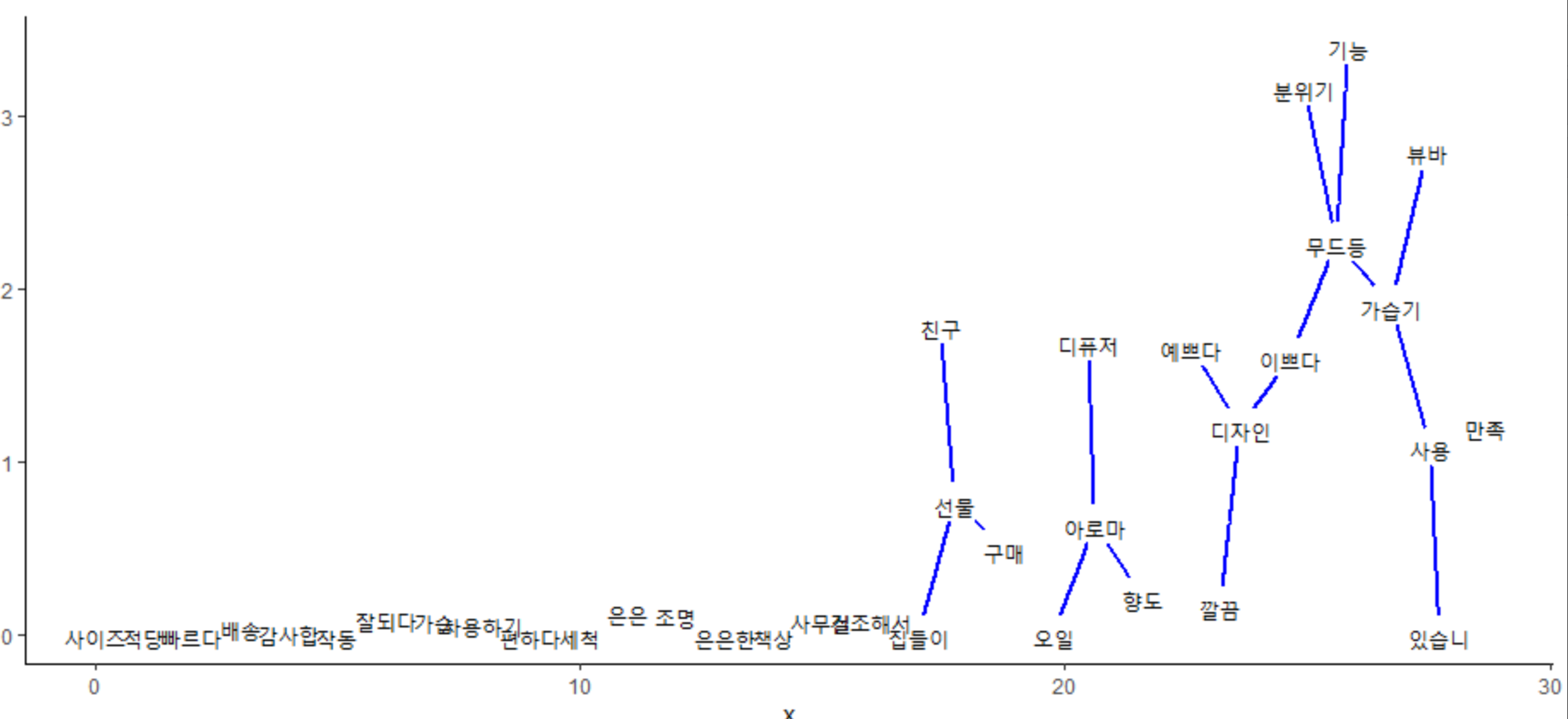


Ex) 성남시 스터디 카페 리뷰 매장 10개에서 리뷰 30개씩 / 모든 매장에서 리뷰 10개씩

해결 문제: 신규 매장 업데이트로 크롤링 예외 오류 발생, 다른 도메인 입력 시 오류

**2. 전처리**

전처리 후 시각화 구현



- 해당 매장 리뷰에 등장하는 단어 빈도 순으로 count  
- 연관 단어 시각화

**3. 모델링**

- 모델링 스터디 및 관련 연구 조사

- 전 조원 Pororo 사용 가상 환경 구축

- Pororo 세부 모델 스터디 및 실습

**7월 4주차 활동 결과 요약**

**크롤링 GUI**

크롤링 매장의 개수, 장소, 파일 이름을 입력하는 GUI 완성 => 크롤링 파트 완료

**모델링 환경 구축**

조원들의 동일한 개발 환경 조성에 번거로움 예상되어 도커 활용 방안 모색 중

**전처리 시각화**

# -\*- coding: utf8 -\*-  
  
from konlpy.tag import Okt  
from hanspell import spell\_checker  
from konlpy.tag import Kkma  
from soynlp import DoublespaceLineCorpus  
from soynlp.word import WordExtractor  
from soynlp.tokenizer import LTokenizer  
import kss  
from kss import split\_sentences  
from wordcloud import WordCloud  
from konlpy.tag import Twitter  
from collections import Counter  
import matplotlib.pyplot as plt  
  
  
texts = [ "조용하고 공부에 집중하기 좋은 곳 이예요 !!!!!", "서현 스터디 카페중에 젤 조음", "진짜 분위기 좋다 알바생분들도 친절 ㅠㅠ", "편안한 분위기여서 공부 잘되요", "내부도 적당히 크고 자리마다 무선 충전기 설치 되어 있는 등 시설이 되게 좋아요!!제공되는 것도 많구요 건물이 낡아서 화장실이 후지긴 한데 어쩔 수 없죠ㅠ음료 포함 정기권 끊었는데 음식 종류도 많구 아직 많이 먹어보진 않았지만 맛있어요!볶음밥이랑 베이커리류도 팔아서 식사 해결도 가능해요ㅎㅎ다만 음료는 만들어주시는 알바 분들에 따라 맛 차이가 큰거 같아요. 그리고 창가 쪽에 앉으면 비둘기 등등의 각종 새들이 자꾸 와서 달려 있는 간판 위에 앉고 부딪히고 막 그래서 소리가 좀 나요ㅠㅠ구구구하고 우는 소리도 들리고 날아가는 소리,부딪히는 소리 전부 잘 들리는게 조금 거슬린다는 점 감안해야 해요!" ]  
  
i=0  
for x in texts :  
 i=i+1  
 print("<" ,i,">\n")  
 print(" 원래 문장 : ", x)  
  
 t= kss.split\_sentences(x)  
 print("문장토큰화 : ",t)  
  
 spelled\_sent = spell\_checker.check(x) #t(토큰화된 문장)넣으면 에러  
 hanspell\_sent = spelled\_sent.checked  
 print(" 맞춤법 검사 후 : ", hanspell\_sent)  
  
 okt = Okt()  
  
 print("\*\*konlpy\_okt\*\*")  
 print(" <형태소 추출> ")  
 print(okt.morphs(hanspell\_sent))  
 print(" <품사 태깅> ")  
 print(okt.pos(hanspell\_sent))  
 print(" <명사 추출> ")  
 print(okt.nouns(hanspell\_sent))  
  
  
  
  
  
noun\_list=okt.nouns(hanspell\_sent)  
  
counts = Counter(noun\_list)  
tags = counts.most\_common(40)  
  
wc = WordCloud(font\_path="malgun.ttf",background\_color="white", max\_font\_size=60)  
cloud = wc.generate\_from\_frequencies(dict(tags))  
  
cloud.to\_file('test.jpg')  
  
plt.figure(figsize=(10, 8))  
plt.axis('off')  
plt.imshow(cloud)  
plt.show()

**단어의 count 기준 Word Cloud**

텍스트이(가) 표시된 사진

자동 생성된 설명