머신러닝을 이용한 게시글 분류

서론 (연구 배경 및 목적)

본 프로젝트를 정하는데는 3가지 배경이 있었다. 첫째, 자연어 처리를 이용한 머신러닝 수요가 증가한다는 점이다. Statista에 따르면 글로벌 자연어 처리 시장 규모는 2017년 약 32달러에서 2025년 약 432억 달러로 증가할 전망이다.[1] 본 동아리 역시 이러한 동향에 맞추어, 자연어 처리에 대해 발표한 친구가 있었다. 단어를 수집하고 쪼개어 언어 속 데이터들을 컴퓨터가 이해하고 처리할 수 있게 한다는 점에 매력을 느꼈다.

둘째, 데이터 수집 단계부터 직접 시도하고자 하는 욕심이 있었다. 지금까지는 머신러닝 알고리즘과 이론적인 내용을 이해하는데 집중하였다. 그러다 보니 이미 머신러닝 모델을 돌리기 직전, 가장 잘 가공된 상태의 데이터만 이용하였다. 하지만 다른 머신러닝 대회를 나가보니, 실무에서는 데이터를 얻어내는 것부터가 시작이라는 것을 깨달았다. 특히 데이터를 얻고 가공하는 단계가 가장 오래 걸리는 것을 알고 나니, 데이터 수집부터 시작하고 싶었다.

셋째, 관련 선행 연구의 양이 풍부하였다. 가장 인기있는 오픈 소스 저장소로 알려진[2] github에 Bi-LSTM에 관련된 오픈 소스가 많았다. 뿐만 아니라 자연어처리와 관련된 오픈소스도 다양하여 프로젝트를 진행하는데 수월할 것이라고 예상하였다. 본 연구는 github의 lumyjuwon님의 오픈소스를 다수 활용하였음을 미리 밝히는 바이다.

본 프로젝트의 목적은 뉴스 기사, 본문을 컴퓨터에 입력하였을 때, 컴퓨터가 글을 분석하여 카테고리를 분류할 수 있도록 하는 것이다. 예를 들어, "불편해서 좋대요" 숙박비 2배로 내고도 묵는 절 이라는 기사 제목을 보고 컴퓨터가 "생활문화" 카테고리의 기사라는 것을 알려주는 것이다. 이러한 프로젝트의 필요성을 언급하기 보다는 자연어 처리를 배울 수 있는 좋은 모델이라고 생각해서 활용하게 되었다.

절차

기본 과정은 크게 두 단계로 나눌 수 있다. 데이터 전처리, 모델 학습시키기. 전처리 과정은 수집-

---

1 http://news.kotra.or.kr/user/globalAllBbs/kotranews/album/781/globalBbsDataAllView.do?dataIdx=176188&column=&search=&searchAreaCd=&searchNationCd=&searchTradeCd=&searchStartDate=&searchEndDate=&searchCategoryIdxs=&searchIndustryCateIdx=&searchItemName=&searchItemCode=&page=2&row=10

2 https://ko.wikipedia.org/wiki/%EA%B9%83%ED%97%88%EB%B8%8C

셔플-가공으로 자세하게 나누어진다. 모델은 오픈소스에 공유되어 있는 Bi-LSTM 모델을 활용하였다. 구체적인 방법은 아래와 같다.

방법

1. 데이터 전처리

1) 데이터 수집

컴퓨터가 뉴스 글을 분류하기 위해서는 학습시킬 수백개의 뉴스 글 데이터가 필요하다. 하지만 단순한 뉴스 제목과 본문의 모음이 아닌, 컴퓨터가 읽어 들일 수 있는 형태에 라벨링이 된 형태여야 한다. 이때 라벨링이란, 정치 카테고리(1) – 정치 뉴스, 사회 카테고리(2) – 사회뉴스 처럼 카테고리의 종류에 번호를 부여하고 뉴스와 짝지어(매핑)져 있는 것을 의미한다.

또한 수백만개의 뉴스 기사를 어디에서 얻을 수 있는지를 고민해 보아야 한다. 이 때 필요한 것이 바로 크롤링이다. 크롤링이란 웹페이지의 내용을 그대로 가져와서 필요한 데이터를 추출하는 것으로, 데이터를 대량 수집하는 기법이라고 할 수 있다. 오픈소스에 올라와있는 크롤러 코드를 활용해서 날짜를 지정하여 네이버 뉴스의 기사제목을 텍스트 형태로 얻어올 수 있다. 자세한 코드는 그림 1과 같다.

```
from korea_news_crawler.articlecrawler import ArticleCrawler

Crawler = ArticleCrawler()
Crawler.set_category("정치", "IT과학", "economy")
Crawler.set_date_range(2017, 1, 2018, 4)
Crawler.start()
```

그림 1

이렇게 얻어진 데이터는 그림 2와 같은 형태로, 카테고리별 엑셀파일로 저장이 된다. 뿐만 아니라 여러 개의 카테고리별로 나뉘어 있는 데이터들을 하나로 합쳐주는 작업도 필요하다. 이를 위한 코드가 그림 3과 같다. 이 코드를 거치고 나면, 모든 데이터가 Article_unity.csv라는 파일로 합쳐진다.

A	B	C	D	E	F	G	H
20180101	IT과학	YTN	새해 볼만한 우주쇼…1월·7월 개기 월식 12일엔 혜성	앵커 2018년 새해에는 달이 지구 그림자에 가려지는 개기 월식을 1월과 7월 두 차	https://news.naver.com/main/rea		
20180101	IT과학	디지털타임	넷플릭스 美 유료방송 구독률 따라잡아	디지털타임스 김지영 기자 미국에서 인기 동영상 서비스 넷플릭스가 케이블 등 전	https://news.naver.com/main/rea		
20180101	IT과학	디지털타임	우체국 정기예금·적금 수신금리 0.3포인트 인상	디지털타임스 김지영 기자 우체국 정기예금과 정기적금의 수신금리가 2일부터 인	https://news.naver.com/main/rea		
20180101	IT과학	디지털타임	공정위 사건처리 및 분쟁조정 통합 시스템 개통	디지털타임스 김지영 기자 우편이나 방문으로만 가능하던 공정거래 관련 사건·분	https://news.naver.com/main/rea		
20180101	IT과학	SBS	미국선 아이폰 배터리 교체 시작…한국은 무소식	앵커 애플이 고객 몰래 아이폰 성능을 떨어뜨렸다는 사실이 밝혀지면서 미국에선	https://news.naver.com/main/rea		
20180101	IT과학	디지털타임	과기정통부 2018년 ICT R&D 사업투자 4조원…ICT부문서 482억원 줄어	디지털타임스 김지영 기자 과학기술정보통신부는 올해 연구개발R&D 사업에서 혁	https://news.naver.com/main/rea		
20180101	IT과학	서울신문	호주도 1조원대 집단 소송… 거세지는 '애플 스캔들'	서울신문 사과문에 고위 임원 서명 안 해 팀 쿡 책임론 등 비판 커져 애플의 구형	https://news.naver.com/main/rea		
20180101	IT과학	한국경제	LG 올해 CES 주인공은 AI 씽큐	음성 인식으로 가전 관리 고재연 기자 음성 인식 냉장고와 오븐이 냉장고에 있는	https://news.naver.com/main/rea		
20180101	IT과학	한국경제	LG디스플레이 불가능은 없다… 세계최초 8K OLED 개발 성공	UHD보다 화소 4배 많아 초대형 TV시장 선도할 것 노경목 기자 LG디스플레이가	https://news.naver.com/main/rea		

그림 2

```
import csv
import os

os.chdir("C:₩₩Users₩₩user₩₩Downloads")

category = ['IT과학','경제','정치','세계','오피니언','사회','생활문화']

file_unity = open('Article_unity.csv', 'w', encoding='euc-kr')
//열어서_저장해라(Article_unity라는 새로운 파일, 쓰기전용으로, 번역_인코딩은euc-kr유니코드한국어로)
wcsv = csv.writer(file_unity)
//WriteCSV에=써넣어라(file_unity를)
count = 0

for category_element in category:
        //위에 IT과학 경제 정치 ... 를 한 요소씩 반복
    file = open('Article_'+category_element+'.csv','r',encoding='euc-kr',newline="")
        //file에 = 열어서_저장해라(형식이 'Article_정치.csv'인 애를, 읽기전용으로,번역_인코딩은euc-kr유니코드한국어로, newline에는 빈 벡터
    line = csv.reader(file)
        //line에 = 읽어들여라(file을 즉:정치 기사리스트를 한 줄씩)
    try :
        for line_text in line:
                //line을 한 줄씩 반복(리스트를 반복)
            wcsv.writerow([line_text[1], line_text[2], line_text[3], line_text[4]])
                //아까 비어있던 file_unity를 넣은 WCSV에.한줄씩써넣어라(엑셀의[카테고리], 엑셀의[뉴스], 엑셀[본문] )
    except:
        pass
```

그림 3

2)데이터 셔플

이 후 데이터 셔플 단계를 거친다. 데이터 셔플이란 데이터들을 섞어주는 것을 의미한다. 셔플을 거치는 이유는 Article_unity 파일에는 각 카테고리가 뭉쳐져서 존재하는데, 이 응집이 문제를 일으킬 수 있기 때문이다. 선행연구에 따르면, Article_unity를 그대로 학습시킬 경우 먼저 나오는 카테고리의 기사들만 학습을 진행하기 때문에 어떤 기사를 읽어도 먼저 학습한 카테고리로 인식하는 문제가 있다고 한다. 예를 들어, 사회->정치->생활문화 순서로 응집시켜 학습시키면 맨 처음 학습한 "사회" 분야의 W 파라미터를 그대로 정치와 생활문화를 학습시키는데 이용하기 때문에 문제가 발생한다. 이를 방지하기 위해서 데이터를 랜덤하게 섞어주는 셔플을 거치게 된다. 자세한 코드는 그림 4와 같다. 결과적으로, Article_unity 속에 응집되어 있던 카테고리들이 그림 5과 같이 산발적으로 존재하는 것을 볼 수 있다.

```
import csv
import random
import os

os.chdir("C:₩₩Users₩₩Juwon₩₩PycharmProjects₩₩tensorflows₩₩parser₩₩Csv") // Csv가 있는 경로 설정

file = open('Article_unity.csv', 'r', encoding='euc-kr')
line = file.readlines()
random.shuffle(line)
rcsv = csv.reader(line)

file_write = open('Article_shuffled.csv', 'w', encoding='euc-kr', newline="")
wcsv = csv.writer(file_write)

for i in rcsv:
    try:
        wcsv.writerow([i[0].strip(), i[1], i[2], i[3]])
    ////아까 비어있던 file_unity를 넣은 WCSV에.한줄씩써넣어라(엑셀의 [날짜].₩n지우고, 엑셀의[카테고리], 엑셀의[뉴스], 엑셀[본문] )
    except:
        pass
```

그림 4

	A	B	C	D	E	F	G	H	I
1		IT과학	아이뉴스2	신한금융투자 핵심시스템 리눅스 전환					
2		생활문화	국민일보	1960년대 이후 한국교회사 교계 최초 집대성					
3		오피니언	경향신문	먹거리 공화국쌀이 죽어간다					
4		정치	뉴스1	통일부 정례 브리핑					
5		세계	뉴스1	대구 북구 침산동 교통사고 현장					
6		오피니언	파이낸셜뉴	데스크칼럼 서열문화와 갑질					
7		생활문화	스포츠경향	오늘의 운세 나침반2018년 4월 11일					
8		사회	아시아경제	경기도 부동산포털사이트 거래·생활정보도 제공한다					
9		사회	MBC	연대 정시 모집 확대...학종 불신 시작					
10		IT과학	ZDNet Kor	“북한 APT 해킹 작년부터 한국 넘어 일본 등으로 확대”					

그림 5

3) 데이터 가공

데이터 전처리의 마지막 단계는 엑셀 형태로 되어있는 데이터를 컴퓨터가 학습할 수 있는 형태소로 자르고 백터 형태로 바꾸는 것이다. 머신러닝을 이용한 자연어 처리의 기본 개념은 문장을 단어로 자르고 단어를 형태소로 자르는 것이다. 형태소는 의미를 가지는 가장 작은 말의 단위이기 때문에 컴퓨터가 형태소를 하나씩 학습하여 전체 문장의 의미를 파악할 수 있다. 이렇게 형태소로 잘려진 데이터는 컴퓨터가 읽을 수 있는 벡터 형태로 변환되어야 한다. 자세한 코드는 사진 5과 같다.

```
from konlpy.tag import Twitter
from gensim.models import Word2Vec
import csv //comma separated values, txt형 파일확장자

twitter = Twitter()

file = open("Article_shuffled.csv", 'r', encoding='euc-kr')
    //file에=열어서 저장해라("Article_shuffled를")
line = csv.reader(file)
token = []
embeddingmodel = []

for i in line:
    sentence = twitter.pos(i[0], norm=True, stem=True)
        //sentence에 = twitter.내장함수인pos(part of speech=형태소)를(line안에 형태소별로 자르기)
    temp = []
    temp_embedding = []
    all_temp = []
    for k in range(len(sentence)):
        temp_embedding.append(sentence[k][0])
            ///temp_embedding에는 (형태소를 저장)
        temp.append(sentence[k][0] + '/' + sentence[k][1])
            /// temp에는 (형태소[0] / 해당품사[1] 저장)
        //for 문을 돌면 all_temp에는 모든 문장의 형태소[0]/품사[1]가 저장 , embeddingmodel에는 모든 문장의 품사[1]가 저장
    all_temp.append(temp)
        //all_temp에는 temp저장
    embeddingmodel.append(temp_embedding)
        //embeddingmodel에는 temp_embedding 저장
    category_number_dic = {'IT과학': 0, '경제': 1, '정치': 2,'세계':3,'오피니언':4, '사회': 5, '생활문화': 6}
    all_temp.append(category_number_dic.get(category))
        ///all_temp에 카테고리 매핑정보까지 저장
    token.append(all_temp)
print("토큰 처리 완료")
    //여기까지 하면 all_temp에 형태소[0], 품사[1],매핑 정보 저장되어있음

embeddingmodel = []
for i in range(len(token)): ////index를 돌고
    temp_embeddingmodel = []
    for k in range(len(token[i][0])): ////날짜를 돌고
        temp_embeddingmodel.append(token[i][0][k])
    embeddingmodel.append(temp_embeddingmodel)
        ////돌면서 temp_embeddingmodel에 넣고
// max_vocab size 10000000 개당 1 GB 메모리 차지
embedding = Word2Vec(embeddingmodel, size=300, window=5, min_count=10, iter=5, sg=1, max_vocab_size = 360000000)
embedding.save('post.embedding') //만들어진 사각형 숫자 정보를 -> vector로 바꾸는 모델
```

7777777 1