**연 구 용 역 보 고 서**

**페이크 뉴스 자동 검출 응용 서비스의 인공지능 표준 시스템 분석**

**수탁기관: 동의대학교 산학협력단**

한 국 전 자 통 신 연 구 원

**연 구 용 역 보 고 서**

**페이크 뉴스 자동검출 응용 서비스의 인공지능 표준 시스템 분석**

**수탁기관: 동의대학교 산학협력단**

한 국 전 자 통 신 연 구 원

**제 출 문**

**한국전자통신연구원장 귀하**

본 보고서를 “페이크 뉴스 자동검출 응용 서비스의 인공지능 표준 시스템 분석”의 최종 연구보고서로 제출합니다.

2020년 11월 07일

수 탁 기 관 : 동의대학교 산학협력단

수 탁 기 관 장 : 이 임건 (인)

연 구 책 임 자 : 김 근형

참 여 연 구 원 : 한 윤진

박 예진

이 승찬

**요 약 문**

1. **제 목**

페이크 뉴스 자동검출 응용 서비스의 인공지능 표준 시스템 분석

1. **연구의 목적 및 중요성**

**1) 연구의 목적**

**2) 연구의 필요성**

1. **연구의 내용 및 범위**
2. **연구 결과**
3. **활용에 대한 건의**



**목차**

**그림 목차**

[그림 1. DNA 생태계 활성을 위한 데이터 댐 개념도<출처: HMG Jounnal> 3](#_Toc55507772)

[그림 2. 데이터 댐 7개 사업 연계 구성도 <출처: 과기정통부 자료> 4](#_Toc55507773)

[그림 3. 데이터 구축 전 단계에 적용되는 인공지능 데이터 품질 표준안 <출처: 과기정통부> 5](#_Toc55507774)

[그림 4. 머신러닝 파이프라인과 인간의 편향성 <출처: 구글 AI 포럼> 10](#_Toc55507775)

[그림 5. 성별 직업의 편향성 <출처: Google Develpers Blog> 10](#_Toc55507776)

[그림 6. TensorFlow.js의 계층적인 아키텍처 19](#_Toc55507777)

[그림 7. ‘Robin Hood’ 온라인 공연 장면 20](#_Toc55507778)

[그림 8. 마이크로 소프트의 Together mode 서비스 예 20](#_Toc55507779)

[그림 9. 임베딩 방법 비교 28](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507780)

[그림 10. BertTokenizer의 토큰화 29](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507781)

[그림 11. BERT의 입력데이터 임베딩 30](#_Toc55507782)

[그림 12. Attention 계산 수식 31](#_Toc55507783)

[그림 13. BERT의 사용 유형 33](#_Toc55507784)

[그림 14. BERT구조 블록 다이어그램 35](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507785)

[그림 15. 네이버 영화 리뷰 데이터 수집[영화 리뷰 분류] 37](#_Toc55507786)

[그림 16. BERT 모델 생성 38](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507787)

[그림 17. 토큰 벡터 히스토그램 38](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507788)

[그림 19. 코사인 유사도 계산 코드 39](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507789)

[그림 18. 부정 및 긍정 문장 벡터 히스토그램 39](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507790)

[그림 20. 문장 벡터와 토큰 벡터의 코사인 유사도 40](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507791)

[그림 21. 긍정과 부정으로 분류된 문장에서 사용된 단어 통계 41](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507792)

[그림 22. BERT 기반 모델 블록도 42](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507793)

[그림 23. 리뷰 감정 분류 모델 학습 43](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507794)

[그림 24. 정확도 계산 함수 생성 43](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507795)

[그림 25. 리뷰 데이터를 입력 데이터 형식에 맞게 전처리 44](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507796)

[그림 26. 학습된 모델에 입력시켜 테스트 45](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507797)

[그림 27. 문장 벡터와 토큰 벡터의 코사인 유사도 계산 46](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507798)

[그림 28. 긍정 문장에서 유사도 점수가 높은 단어 강조 46](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507799)

[그림 29. 부정 문장에서 유사도 점수가 높은 단어 강조 46](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507800)

[그림 30. Kaggle의 Fake News 데이터셋 52](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507801)

[그림 31. 수집한 SNU 팩트체크 데이터셋 56](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507802)

[그림 32. 뉴스 데이터를 입력 데이터 형식에 맞게 전처리 57](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507803)

[그림 33. 데이터 토큰화 57](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507804)

[그림 34. 가짜뉴스 탐지 모델 학습 58](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507805)

[그림 35. 가짜뉴스 탐지 모델 테스트 결과 58](file:///C:\Users\admin\Downloads\연구용역보고서(통합본).docx#_Toc55507806)

**표 목차**

[표 1. 데이터 품질 요구사항 <출처: 과학정보통신부> 5](#_Toc55507850)

[표 2. 데이터 구축 요구사항 6](#_Toc55507851)

[표 3. 인공지능 모델의 편향 사례 8](#_Toc55507852)

[표 4. BERT 클래스 종류 31](#_Toc55507853)

[표 5. 리뷰 감정 탐지 서비스 실험 환경 36](#_Toc55507854)

[표 6. 가짜뉴스 탐지 논문 조사 48](#_Toc55507855)

[표 7. 가짜뉴스 탐지 모델 테스트 결과 51](#_Toc55507856)

[표 8. 영어 가짜뉴스 탐지 실험 환경 53](#_Toc55507857)

[표 9. 한국어 가짜뉴스 탐지 실험 환경 55](#_Toc55507858)

# 제 1 장 개요

## 제 1 절 인공지능 서비스 배경 및 현황

1. **인공지능 기술의 성장**

1956년 미국 다트머스 대학에서 개최된 다트머스 회의에서 메카시 교수가 컴퓨터 지능 관련 연구 분야를 인공지능(Artificial Intelligence)로 부르자고 제안하면서 인공지능이라는 용어가 널리 사용되었고 1958년에 퍼셉트론(Perceptron)이라는 신경세포(neuron)을 계산적 모델로 표현한 최초의 신경망(neural network)모델이 제안되었으며, 퍼셉트론이 학습 기능을 가진 기계로 주목을 받으면서 1960년대에는 신경망 연구에 대한 관심이 높았다. 1960년대 인공지능 연구는 다양한 문제해결에 적용할 수 있는 일반적인 방법을 찾는 것에 집중하여 현재 상태와 목표 상태 간의 차이를 계산하고 차이를 줄이는 조작자(문제의 상태를 변화시키는데 사용하는 작업이나 행동)를 찾아 적용하는 과정을 기계적으로 반복하는 수단-목표-분석(mean-ends analysis) 방법에 기반한 연구가 주로 이루어졌다. 이 방법은 비교적 작은 규모의 문제는 효과적으로 해결할 수 있으나 규모가 큰 현실 문제를 해결하기 위해서는 많은 시간과 메모리 공간을 사용해야 한다.

1970년대에는 일반적인 방법보다 특정 영역의 문제를 해결하는 방법이 주로 연구되어 전문가 수준의 해답을 제공하는 전문가 시스템(Expert system)의 연구가 주로 이루어졌다. 전문가 시스템 기술은 1970년대 초반부터 1980년대 중반에 걸쳐서 많이 발전하였으며 성공적인 사례도 많이 있다. 1980년대 중반 부터 199년대 중반까지 신경망 연구가 활발하게 진행되었다. 퍼셉트론을 여러 층으로 연결하여 구성한 다층 퍼셉트론(Multilayer perceptron)이 XOR 문제를 해결할 수 있다는 사실을 알아냈으나 1980년대 초반까지 다층 퍼셉트론을 학습시키는 방법을 알지 못했다. 1986년에 오차 역전파(Error backpropatation) 알고리즘을 효과적으로 학습시킬 수 있다는 것을 보이면서 1980년대 중반부터 신경망 연구가 다시 부활하였다. 신경망은 데이터만 주어지면 학습할 수 있다는 점에서 많은 주목을 받았으며 다양한 분야에서 성공적으로 적용되었다. 1990년 후반에는 머신러닝 모델의 하루인 서포트 벡터 머신(SVM: Support Vector Machine)이 개발되어 많은 관심을 끌었다. 2000년대에 들어서면서 대규모 데이터에 머신러닝과 통계학적 기법을 적용하여 의미있고 유용한 숨은 정보를 추출하는 데이터마이닝 알고리즘들과 여러가지 데이터마이닝 도구가 개발되고 있다. 또한 빅데이터의 출현과 함께 빅데이터를 처리할 수 있는 기술에 대한 연구가 2000년대 후반부터 활발하게 진행되고 있다. 딥러닝(Deep Learning) 기술이 우수한 적용 사례를 만들어 가면서 2000년대 후반부터 컴퓨터 비젼, 음성인식, 자연어처리 분야에서 놀라운 결과를 만들어 내고 있다.

스마트폰의 음성인식 서비스의 대중화, 음성인식 스피커, 자율주행 자동차, 자동번영기, IBM의 왓슨(Watson), 군사용 로봇 빅독(Bit dog)을 포함한 지능로봇 등 인공지능 기술을 이용한 제품과 서비스들이 출현하고 있다. 2014년 6월에는 소년 캐릭터를 가진 챗봇으로 개발된 유진 구츠만(Eugeun Goostman)이 튜링 테스트를 통과했다. 그 후 사용자의 성향을 학습하여 대화를 하는 개인 맞춤형 챗봇도 개발되고 있다. 2016년에 마이크로소프트는 대화 삳태와 대화를 하면서 상대방의 상황에 맞춰 대화를 학습할 수 있는10대 소녀 캐릭터인 테이(Tay)라는 챗봇을 트위터를 통해 서비스하였는데 극우적이고 인종차별적인 데이터로 학습하여 극우적이고 인종차별적인 트윗을 하는 바람에 운영을 시작한지 16시간 만에 서비스를 중단하였다.

현재 인공지능은 얼굴 인식, 인터넷 검색 또는 자율 자동차 운전 등 특정 부문을 수행하도록 설계되어 있어 제한된 인공지능 (narrow AI)라고 한다. 제한된 인공지능은 체스를 하거나 방정식을 풀 때와 같이 특화된 작업에서 인간 보다 우위에 있으나 많은 연구자들의 장기적인 목표는 거의 모든 인지 작업에서 인간보다 우월한 포괄적(general AI)를 만드는 것이다.

1. **데이터댐과 인공지능**

아날로그 데이터의 디지털화를 의미하는 Digitization이 비즈니스 프로세스의 디지털화를 의미하는 Digitalization으로 발전되고 모든 사고와 활동의 단위도 디지털화되어 새로운 비즈니스 모델을 창출하는 디지털 트랜스포메이션 개념으로 확장되고 있다. 시장 조사 전문 기관 IDC는 디지털 트랜스포메이션을 “신규 기술을 프로세스, 경험 그리고 가치를 변화하는데 적용한 기술”로 정의한다.

정부는 “한국판 뉴딜 종합계획”에서 데이터(Data), 5G(Network), 인공지능(AI) 융합을 통한 디지털 트랜스포메이션을 가속화하는 DNA 생태계를 강화하는 정책을 발표하였다. DNA 생태계에서는 다량의 데이터가 네트워크를 통해 전달되어 인공지능 기반 제품과 서비스 개발에 활용되며 이 때 새롭게 생성된 데이터는 다시 네트워크를 통해 전달되는 순환 구주롤 보인다. 데이터는 기존의 노동, 자본을 뛰어넘는 생산요소로 디지털 트랜스포메이션의 핵심 동력으로 생산성 향상의 기반이 된다. 네트워크는 방대한 데이터를 초고속으로 전송하고 초저지연으로 데이터와 모든 사물을 연결한다. 인공지능은 데이터를 활용하고 타 분야와 융합을 통해 디지털 트랜스포메이션을 구현한다.

DNA 생태계 강화 정책의 핵심은 “데이터 댐” 이다. DNA 생태계의 “데이터 댐”은 물을 모아 방류하는 댐처럼 디지털 경제의 원유로 비유되는 데이터(공공 및 민간데이터)를 모두가 이용할 수 있도록 하나의 형태로 가공하여 모아두는 댐으로 모아진 데이터에 기반한 디지털 역량을 전 산업 분야에 결합할 수 있도록 한다. 구체적으로 “데이터 댐”은 공공과 민간의 초연결성을 보장하는 네트워크를 통해 생성되는 데이터를 수집하고, 표준화하고 가공, 활용하여 더 똑똑한 인공지능을 만듦으로써 기존의 산업의 혁신과 혁신적인 서비스 개발을 통해 새로운 일자리를 창출하는 것을 목표로 한다. [그림 1]은 데이터 댐의 개념도이다.



그림 1. DNA 생태계 활성을 위한 데이터 댐 개념도<출처: HMG Jounnal>

데이터 댐을 만들기 위해서는 데이터 허브(Data Hub)가 필요하다. 데이터 허브는 다양한 데이터를 통합하여 이를 가공하고 체계적으로 관리하여 필요한 정보를 만든다. 한 기관 내의 다양한 데이터를 수집하고 재가공해서 새로운 데이터를 만들거나 다른 기관의 데이터들을 모아 분석해서 활용할 수도 있으며 기존의 데이터를 다양한 방법으로 분석하여 새로운 가치를 더해 활용할 수 있다.

[김경훈, “디지털 트랜스포메이션 촉진을 위한 DNA 생태계 활성화 과제,” AI TREND WATCH, 정보통신정책연구원, 2020. 9.]

[현태오토에버, “디지털 시대를 위한 데이터 허브,” HMG Journal, 2020, 8]

****

그림 2. 데이터 댐 7개 사업 연계 구성도 <출처: 과기정통부 자료>

[그림 2]는 데이터 댐이 기반이 된 DNA 생태계는 “데이터·AI 인프라”를 활용해서 “5G, IoT, 센서 로봇 등을 활용한 “데이터 수집”, “데이터 축적·가공·결합”, “데이터를 활용하여 혁신적인 서비스를 창출” 이라는 데이터의 가치사슬로 구성된다. 데이터 댐과 관련하여 추진되는 7대 핵심 사업은 ‘클라우드 이용 바우처’와 ‘클라우드 플래그십 프로젝트’ 를 통해 “데이터·AI 인프라”를 확충하고 ‘AI 바우처’, ‘AI 데이터 가공 바우처’, ‘AI 융합 프로젝트(AI + X)’, 를 통해 데이터 활용 기술을 개발하며 수집된 데이터를 축적하고 가능하기 위해 ‘인공지능 학습용 데이터 구축’과 ‘빅데이터 플랫폼 및 센터 구축’이 이루어진다.

과학기술정통부는 인공지능 데이터 관련 품질의 개념과 범위, 세부 요구사항등을 정립한 ‘인공지능 데이터 품질 표준안’을 공식 제안하고 관련 전문가의 의 견 수렴 등의 절치를 밟아 내년 6월에 표준으로 채택할 계획이다. 표준안은 인공지능 학습용 데이터 구축과정 전반에 대한 품질 기준과 요구사항을 규정하고 있다. 현재 마이크로소프트나 구글 등 글로벌 IT기업들이 구축해온 누적데이터의 경우 정확도가 43~83% 수준에 불과하다. 세계적으로 인공지능 데이터 품질에 대한 체계적인 기준이 정립된 사례가 없다.

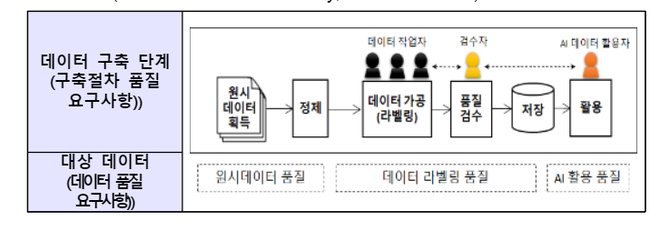


그림 3. 데이터 구축 전 단계에 적용되는 인공지능 데이터 품질 표준안 <출처: 과기정통부>

[그림 3]은 인공지능 데이터 품질 표준안으로 품질관리 대상을 구축철차의 품질과 데이터의 품질로 구분하여 품질 요구사항을 도출하고 있다.

[뉴스 기사, “과기정통부, ‘AI데이터 품질 표준안’발표…국내외 표준화 추진”, 2020. 10. 5]

인공지능 데이터의 품질에 대한 국제 표준은 지난 4월 우리나라가 “인공지능 국제표준화회의(ISO/IEC JTC1/SC42)”에 인공지능 데이터 품질 관련 사항을 신규 과제로 제안하여 논의가 시작되는 초기단계이다. 품질관리 대상이 되는 데이터 상태를 원시 데이터, 데이터 라벨링, 인공지능 활용으로 나누어 품질을 다음 [표 1]과 같이 구분하고 있다.

표 1. 데이터 품질 요구사항 <출처: 과학정보통신부>

테이블이(가) 표시된 사진

자동 생성된 설명

표준안은 자연어처리, 자율 자동차, 의료, 제조 등 다양한 분야에서 공통적으로 적용 가능한 “범용적인 표준” 형태로 개발되었고 데이터의 품질을 체계적으로 확보하고 상호호환성을 제공할 수 있도록 [표 2]는 데이터의 획득-정제-가공-품질검증과 같은 단계별 필요한 표준 절차와 품질 요구사항을 정의하는 기본적인 데이터 규격을 담고 있다. 구체적으로 원시 데이터 수집 단계의 다양성, 사실성 등의 품질 요구사항, 파일 포맷, 해상도 등의 구술 적합성 요구사항, 정제단계의 데이터 중복방지 및 비식별화 조치 요구사항, 가공단계의 객체 분류체계 및 라벨링 규격 요구사항, 품질검수 단계의 유효성 등 검수 요구사항 방법으로 구성되어 있다.

표 2. 데이터 구축 요구사항

테이블이(가) 표시된 사진

자동 생성된 설명

최근 인공지능 모델 개발에 필수적인 양질의 데이터를 대규모르 구축하고 개방하는 인공지능 학습용 데이터 구축 사업을 2017년 부터 시작하여 텍스트와 이미지 그리고 영상 분야의 인공지능 서비스 개발을 위해 4,650만 건의 머신러닝 학습용 데이터를 구축하였고 현재 인공지능 통합지원 플랫폼인 “AI 허브([www.aihub.or.kr](http://www.aihub.or.kr))”를 통해 공개 중이다. 현재 수집중인 데이터는 크게 국가적 산업적 필요성에 부합되는 데이터와 국민 편의 향상에 부합되는 데이터로 분류된다. 국가적 산업적 필요성에 부합되는 데이터는 사람의 감성 및 문맥을 이해할 수 있는 자연어어 처리 분야, 자동차와 드론 등 자율주행기술 분야, 음성, 시각, 언어 등 융합 분야 등 활용가치가 높은 분야에 해당되며 국민 편의 향상에 부합되는 데이터는 질병진단과 헬스케어 분야, 사람의 얼굴을 악의적으로 변조한 딥페이크 방지 기술 분야, 장애인의 삶을 향상시킬 수 있는 분야에 해당된다.

그외에 AI 데이터 가공 바우처 사업은 중소기업과 스타트업들이 보유한 데이터를 AI 학습용 데이터로 전환시켜 혁신적인 AI 서비스 개발을 지원한다. AI+X 프로젝트는 각 분야에서 수집·축적된 데이터의 안전한 학습과 인공지능 모델 개발 및 활용을 지원한다.

## 제 2 절 신뢰성있는 인공지능

1. **인공지능의 위험성**

초 지적인 인공지능이 사랑이나 증오와 같은 인간의 감정을 표현하지 않기 때문에 인공지능이 의도적으로 선하거나 악해질 수 없다.. 그러나 다음의 두 가지 시나리오에 대해서 인공지능이 위험 요소가 될 수 있다고 많은 전문가들이 이야기하고 있다.

* 인공지능이 치명적인 작업을 하도록 프로그램 되어 있는 경우: 예로 인간을 살상하도록 프로그램되어 있는 인공지능 모델이 탑재된 자율병기들의 잘못된 사용은 대량의 사상자를 낼 수 있다. 인공지능 무기 경쟁이 우발적인 인공지능 전쟁을 초래하면, 이러한 인공지능 무기들은 적의 방해를 피하기 위해 임무를 중단하지 않도록 설계되었다면, 인간은 이러한 상황을 통제할 수 없을 수 있다. 이러한 위험은 제한된 인공지능에서도 가능하나 인공지능의 지능과 자율 수준이 높아짐에 따라 증가한다.
* 인공지능이 유익한 작업을 하도록 프로그램 되어 있으나 목표 달성을 위해 파괴적인 방법을 사용하는 경우: 인공지능의 목표를 우리의 목표와 완벽하게 일치 시키지 못했을 때 일어나는 것으로, 예를 들면 자율 자동차로 공항으로 가능한 가장 빨리 갈 것을 요구한다면 인공지능은 요구를 문자 그대로 받아들이게 되면 차멀미를 하는 고생을 할 수도 있다.
* 배달의 민족이 7월 부터 수도권 모든 지역에 도입한 AI 추천 배차제는 배달 기사의 동산과 주문 음식의 특성을 고려하여 가장 적임자인 배달 기사를 선정하고 최단 이동 경로와 시간을 안내한다. 경로와 시간을 안내하는 알고리즘이 배달 기사의 출발지와 목적지 간 직선거리에 알고리즘이 계산한 특정 숫자를 곱해 배달 거리와 소요시간을 산정하는데 이는 신호 위반을 해야 하는 상황을 만들 수도 있다.

얖의 예로 부터 초지능 인공지능에 대한 우러는 인공지능의 악의가 이나라 능력이라고 추론할 수 있다. 초지능 인공지능은 목표를 달성하는데 매우 뛰어나기 때문에 그 목표가 우리의 목표와 일치하지 않을 때 큰 문제를 야기할 수 있다.

[Future of Life Institute 블로그, Max Tegmark, “인공지능의 이점과 위험,”]

[한겨레 신문기사, 가라면 가? 25분 거리를 15분 안에 가라는 ‘AI 사장님’ 2020. 10. 30]

인공지능이 인간보다 더 똑똑해질 가능성이 있기 때문에 인간의 지능으로 인공지능이 어떻게 행동할지 예측할 수 있는 확실한 방법이 존재하지 않는다.

1. **인공지능의 신뢰성**

유튜브와 넷플릭스의 추천 시스템은 자신들이 보유하고 있는 방대한 데이터를 기반으로 이용자가 선호할 만한 콘텐츠를 추천하는데 100% 정확하지 않다. 이는 이용자의 상황 맥락에 따라 콘텐츠에 대한 선호도가 바뀌기 때문에 이를 완벽하게 맞추는 것은 어렵다. 즉 추천시스템의 정확도는 결국 확률의 문제이며, 인공지능 알고리즘이 확률적인 것이기 때문에 완벽을 기대하는 것은 무리이기 때문에 신뢰성을 논하는 것이 이르다.

추천 시스템이외에도 현재의 인공지능 모델은 편향성, 차별 등 다양한 문제점을 갖고 있다. 이러한 문제가 발생하는 이유는 알고리즘 또는 알고리즘의 기반이 되는 데이터 때문이다. 인공지능 모델은 데이터를 머신러닝 학습시킨 알고리즘으로부터 구현되는데 학습에 사용되는 데이터가 편향적일 수 있다. 학습하는 과정에서 데이터의 특징을 어떤 식으로 설정하고 가중치를 부여할지가 인공지능 모델이 예측하는데 있어 변수가 된다. 인공지능은 데이터를 통해 머신러닝을 하기 때문에 데이터가 편향되어 있거나 신뢰성이 결여되어 있으면 결과도 편향되거나 신뢰성이 낮게된다.

다음은 인공지능 모델의 편향사례를 나타내고 있다.

표 3. 인공지능 모델의 편향 사례

|  |  |
| --- | --- |
| 분야 | 내용 |
| 챗봇 | 마이크로소프트가 선보인 딥러닝 기반 챗봇은 일부 이용자에 의해 인종차별, 욕설 등을 학습하여 서비스가 시작되고 16시간 만에 운영 중단 |
| 이미지로 성별분석 | 이미지를 통한 인공지능이 성별 특성을 분석한 기존 연구에서 턱과 코가 크고 이마가 넓으면 남성, 턱이 작고 코가 길며 이마가 좁으면 여성으로 예측하였으나 조명, 화장 등 편향성 조건을 제거하면 결과가 달라짐 |
|  |  |
| 채용 | IT 산업에서 남성비중이 높았던 과거 데이터를 그대로 학습하여, 아마존이 개발한 인공지능 채용프로그램은 여성적 어휘를 사용하는 지원자를 차등적으로 판단 |
| 신용/대출 | 애플과 골드만삭스가 출시한 신용카드 애플카드는 성별에 따라 신용한도를 차별 적용 |
| 의료 서비스 | 연간 2억명이 이용하는 미국의료시스템에서 인종차별, 과거 병력 및 건강진단 결과 등 잠재적인 건강위험을 예측하여 질병 가능성이 높은 사람에게 우선적을 의료 서비스를 제공하는 시스템은 고액 치료를 받지 못하는 흑인 환자를 차별 |

인공지능이 공정한 머신러닝을 통해 인공지능 고도화된다고 해도, 항상 객관적인 진실을 말하는 것이 아니기 때문에 기존 데이터의 편향성을 확대 재생산할 가능성을 차단할 수 있도록 사람이 머신러닝이 징행되는 동안 사회적인 맥락을 인지하고 있어야 한다.

머신러닝을 활용하여 인간의 편향상을 찾아낼수도 있다. 베키 화이트 구글 매니저는 지나 ‘데이비스 포용지수(GD-IQ)’를 머신러닝을 활용하여 무의식적으로 적용되는 인간의 편향성을 찾아냈다. 영화배우이자 여성인권운동가인 지나 데이비스가 2014~2017년 사이에 미국에서 흥행한 100위까지의 영화를 분석하니 여성들의 출연시간과 대사 분량이 남성 배우에 비해 절반 수준이었다. 지나 데이비스는 이 결과를 공개하며 “무의식적인 편향성을 드러내는데 데이터가 가장 강력한 도구” 라고 하였다.

베커 화이트에 따르면 머신러닝을 통해서 인간의 편형성이 강화되는 것을 막을 수 있다. 머신러닝의 편향성을 줄이거나 없애기 위해서는 데이터에 편향성이 존재한다는 것을 인정하고 데이터의 편향성 인지가 필요하다.

편향성은 크게 선택 편향, 확증 편향, 자동 편향이 있으며 선택 편향의 예로 한 지역에서 생성된 데이터를 이용하여 학습한 머신러닝 모델은 그 지역에 대한 편향성을 가지는 지리적인 편향이 있다. 또한 신발이나 음식 등 모든 데이터에도 편향성을 가지기 때문에 특정 영역의 데이터를 다른 영역에 확대 적용하지 말아야 한다. 본 연구에서 가짜 뉴스 탐지 시스템의 경우 학습한 가짜뉴스의 주제와 검증하려는 가짜뉴스의. 주제가 다른 경우 탐지 정확도가 낮음을 확인할 수 있었다.

확증 편향은 데이터 수집 과정에서 조사자(리서처)는 자신의 믿음과 일치하거나 유사한 데이터를 무의식적으로 수집하게 되는데 이는 데이터 처리 과정에도 영향을 미치는데 조사자 관점에서는 데이터의 편향성을 인지하지 못하기 때문에 더욱 문제이다. 유튜브 시청자들이 자신이 선호하는 내용의 뉴스 만으로 골라보며 이 것을 진짜라고 생각하는 것이 확증 편향의 예이다.

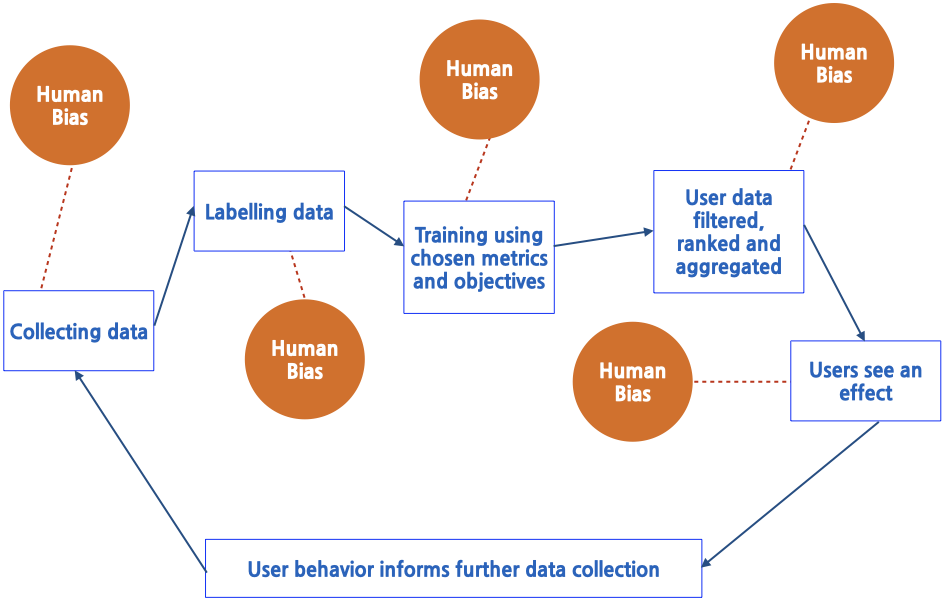


그림 4. 머신러닝 파이프라인과 인간의 편향성 <출처: 구글 AI 포럼>

마지막으로 자동 편항인데 머신러닝은 자동 처리 데이터를 비자동 처리 데이터보다 선호한다. 이 과정에서 데이터의 배제가 생기게 되고 결국에는 편향성이 강화되게 된다.

테이블이(가) 표시된 사진

자동 생성된 설명

그림 5. 성별 직업의 편향성 <출처: Google Develpers Blog>

[Google Developers Blog – Text Embedding Models Contain Bias, Here’s Why That Matters]

이러한 편향이 복합적으로 적용된 사례가 “성별 직업의 연결 편향”이다. 구글이 200개의 직업 관련 단어의 편향성을 분석한 결과 가사 도우미나 종업원, 간호사는 여성의 직업으로 장의사, 심판, 배우, 철학자, 대통령 같은 직업은 남성의 직업으로 분류하였다([그림 5]). 이러한 편향성이 있는 데이터로 만들어진 머신러닝 모델이 실제 현실에 반영되면 부정적인 사회적 편향성을 강화할 것이다.

구글은 편향성 제거를 위한 활동을 공공 정책, 엔지니어링, UX 디자인, 교육 등 여러 관점에서 수행하고 있다. 머신러닝 결과의 편향은 사회적인 문제와 관련이 있기 때문에 머신러닝 시스템이 분석 결과에 어떻게 영향을 미치는지 파악(설명)하는 것이 필요하며 머신러닝 모델의 결과가 편향적일 수 있다는 것을 인지하여야 한다. 또한 머신러닝 모델 개발자도 분석 결과에 대해서 사회적 맥락에서 점검하고 시스템을 개발하여야 한다.

[Digital Today 기사, 석대건, “AI편향 이란 무엇인가? 구글의 AI 원칙 데이터 왜곡없어야”, 2019. 6]

**3. 국내외 현황**

1. **DARPA의 Explainable AI 프로젝트**

DARPA에서는 “머신러닝 기술의 성공은 다양한 응용서비스에 인공지능이 활용

Explainable AI 프로젝트는 인공지능의 높은 예측 정확도를 유지하면서 설명 가능한 모델을 생성하고, 인간이 인공지능 시스템을 이해하고 적절하게 신뢰하며 출력 결과를 효과적으로 관리할 수 있도록 한다. 새로운 머신러닝 시스템들은

1. 구글의 Explainable AI 프레임워크

머신러닝 모델에서 수행한 예측을 파악하고 해석할 수 있도록 지원하는 도구(프레임워크)인 Explainable AI를 제공 중이다. 이 도구를 통해 인공지능 모델 개발자는 모델 성능을 디버깅하고 성능을 향상시키고 모델의 동작을 객관적으로 파악할 수 있다. AutoML Table 및 AI 플랫폼의 모델 예측에 대한 특성 기여 분석을 생성할 수 있고 모델 동작을 시각적으로 파악할 수 있는 기능을 제공한다.

해석 가능하고 포괄적인 인공지능 모델을 설계하기 위해서 데이터와 모델 사이의 편향, 드리프트 등의 차이를 감지하고 해결하도록 도구를 사용하여 해석 가능하고 포괄적인 인공지능 모델을 설계할 수 있도록 한다[google].

[https://cloud.google.com/explainable-ai]

구글은 머신러닝의 편향성을 제거하기 위해서 오픈 데이터셋을 공유하여 각 데이터에 대한 속성을 추가하여 머신러닝 데이터의 맥락을 고려할 수 있도록 지원하고 있다. 또한 Facets 툴을 사용하여 데이터 시각화로 편향성을 점검할 수 있고 What-If 모델을 머신러닝에 적용해 모델링 설계에 따라 분석 결과가 다른 경우도 확인하고 있다.

1. 마이크로소프트의 윤리적 인공지능을 위한 방법론

마이크로소프트는 내부 윤리위원회를 설치하고 윤리적 문제를 일으킬 위험을 낮출 수 있는 인공지능 모델을 개발할 수 있도록 지원 중이다. 마이크로소프트는 테이라는 챗봇에서 편향성이 있음을 인지하고 인공지능 챗봇 개발자들이 인식해야할 윤리적 지침을 10개의 주제로 가이드라인을 제시하였다.

1. **그외**

스탠포드 대학은 인간중심 인공지능 연구소 HAI(Human-Centered AI Institute)를 설립하고 삶의 질을 높이는 인간중심 인공지능 및 안전과 신뢰성 보장에 관한 연구를 진행 중이다.

MIT 미디어 랩의 윤리 및 거버넌스 AI 이니셔티브는 인공지능 기반 시스템의 개발, 적용 및 기능은 빠르계 진화하고 있으며 이러한 기술 및 관행의 사회적 영햫, 거버넌스 및 윤리적 영향과 관련된 광범위한 종요 장단기 질문에 대한 답을 찾기 위한 것이다.

하버드 케네드 스쿨의 The Future Society 그룹의 AI 이니셔티브에서는 글로벌 인공지능 정책 프레임워크를 제공하여

# 제 2 장 인공지능 신뢰성 분석

## 제 1 절 인공지능 현황

DNN(Deep Neural Network) 기반의 딥러닝 기술의 발전은 과거 60년간 축적하였던 인공지능의 업적을 단숨에 뛰어넘을 만큼 놀라운 성공을 거두었다. 그러나 딥러닝 알고리즘은 많은 훈련 데이터를 요구하며 딥러닝 체게가 탐욕적 알고리즘(greedy algorithm)에 기반하며 훈련단계에 사용한 시나리오와 다른 시나리오에 대해서 적응적으로 대응하지 못하는다는 점, 딥러닝의 결과가 블랙박스를 통해 산출되기 때문에 과정을 설명할 수 없다는 점, 적은 양의 지식을 바탕으로 프로그램되어 있어 인간 심리에 대한 상식을 가지고 있지 못하다는 문제점을 가진다[exp1]

[Newspeppermint, https://newspeppermint.com/2019/02/26/shallow\_deep\_learning/]

딥러닝의 여러 문제점 중 과정을 설명할 수 없다는 점을 개선하기 위해서 설명 가능한 인공지능에 대한 연구가 활발하게 이루어지고 있다. 설명가능한 인공지능은 아래 그림과 같이 사용자가 인공지능 시스템의 동작과 최종 결괄르 이해하고 올바르게 해석해서 결과를 생성하는 과정을 설명가능하도록 해주는 기술로 사용자는 최종 결정의 내리기 전에 인공지능 시스템이 내린 결정이 최선인지를 확인한다.

설명가능한 인공지능의 대표적인 연구는 2017년 부터 DARPA에서 진행하고 있는 XAI 학습 모델 개발 및 테스트에 관한 연구로 기존의 머신러닝 기술을 변형하거나 새로운 머신러닝 기술개발을 통해 높은 학습능력을 유지하면서 설명가능성을 향상시키는 연구를 수행하였다. 대표적인 방법에는 심층설명학습(Learning deep explaination), 해석가능한 모델(More interpretable model) 학습, 귀납적 모델(Model induction), 히트맵(headmap) 기반 방법이 있다.

[Defense Advanced Research Projects Agency, Explainable Artificial Intelligence (XAI), DARPA presentation. DARPA: 1-18. July , 2017]

심층설명학습(Learning deep explaination)방법은 은닉층의 노드들이 의미있는 속성들과 연관되도록 네트워크를 학습시키는 방법으로 학습이 끝나면 은닉층의 각 노드들은 인식하고자 하는 객체의 구성 성분(귀, 털, 발톱, 수염 등)을 나타내고 테스트 영상에 대해 대상(예, 걔 또는 고양이)을 판단하기까지 활성화된 은닉노드들의 구성 성분을 통해 대상을 판단한 이유를 설명가능하도록 하는 방법이다.

해석가능모델(More interpretable model) 학습 방법은 ~~~~. 이 방법은 크게 베이지안 프로그램 학습(Bayesian program learning) 방법과 확률적 AND-OR 그래프 기반 모델(Stochastic AND-OR Graph, AOG) 방법으로 구분된다. 베이지안 프로그램 학습 방법은 가상의 펜놀림(?)을 이용하여 만들어진 모든 문자를 작은 조각들의 조합과 관계에 따라 조건부 확률값을 갖도록 학습한다. 이것은 사람이 사물을 인식할 때 전체를 인식하지 않고 사물의 일부로 인식하고 이들을 관게에 기반하여 결합하여 새로운 사물에 대한 판단을 내리는 것에 기초한다. 이처럼 부분적으로 학습된 모델을 이용하면 변혀오딘 사물이 입력되어도 사물을 조각으로 분류하고 조각들을 조건부 확률 값에 의해 합쳐가는 과정에서 일부 정보가 손실되거나 변형되어도 올바르게 분류를 할 수 있다. 확률적 AND-OR 그래프 기반 모델은 입력데이터의 특징을 AND-OR 관계 그래프로 생성하여 분류 결과에 연결된 노드에서 근거를 확인하는 방법으로 AND-OR 관계 그래프를 통해 터미널 노드에 도달하기 까지 거쳐간 노드들을 분석할 수 있고 분류된 이미지와 매핑되는 특징들을 식별하여 결과에 대해 설명을 한다.

귀납적 모델 방법은 지역적으로 해석 가능하고 모델에 구속되지 않는 설명(LIME: Local interpretable model-agnostic explainations) 과 배이지안 규칙 리스트(BRL: Bayesian Rule Lists) 방법이 있다. LIME는 설명 가능한 다른 분류 모델과의 상호 대조 및 추론으로 타깃 분류 모델의 최종 결과를 설명하는 기술로 이미지 분류의 경우 설명 가능한 분류 모델과 타깃 분류 모델의 결과를 서로 비교하여 유사한 결과가 도출되면 설명가능한 분류 모델의 근거를 타깃 분류 모델의 결과 도출의 근거로 활용하는 방법이다. BRL은 고차원, 다변량 특정 공간을 단순한 결정 트리로 변환하여 결정문을 쉽게 해석 가능하도록 만드는 방법이다.

히트맵 기반 방법은 기존의 딥러닝 블랙박스 모델은 유지하면서 예측 결과를 토대로 거꾸로 입력 이미지의 요소를 강조하도록 시각화하여 이미지의 어떤 부분이 결과 도출에 영향을 주는지 알아내려는 방법이다. 이 방법에는 민감도 분석(SA: Sensitivity analysis)과 계층별 관련성 전파(LRP: Layer-wise relevance propagation) 알고리즘이 있다. 민감도 분석 알고리즘은 출력함수 f(x)에 대해 편미분을 하여 각 입력변수의 중요도를 정량화하여 입력 이미지의 어떤 부분이 딥러닝 모델의 결과 도출에 큰 영향을 미쳤는지 측정하여 설명하는 방법이다. 중요도는 히트맵으로 표현되며 이미지 분류에 중요한 영향을 미친 부분은 밀도가 높게 나타나고 그렇지 않은 부분은 밀도가 낮게 나타난다. 실제 배경이 복잡한 이미지에 적용하면 실 대상 이외에 주변부도 모두 밀도가 높게 나타나서 예측 결과에 대해 원인을 추론의 정확도가 낮아진다. 계층별 관련성 전파 알고리즘은 분류기의 결정을 테일러 분해(tayler decomposition)에 의해 설명하는 방법으로 예측 함수 f(x)를 지역 재분배 규칙을 이용하여 역전파 형태로 신경망의 각 계층별 기여도를 측정한다. 이 때 기여도는 특정값과 각 계층의 가중치 결합으로 계산되고 이 값을 연관성 스코어라 한다. 이렇게 역전파된 각 계층의 기여도는 입력 영상에 히트맵으로 시각화하여 사용자에게 제시된다. 이 방법은 민감도 분석 방법에 비해 모델의 분류 결과에 해당하는 객체에 히트맵이 집중되기 떄문에 사용자가 분류 결과에 대해 더 이해가 가능하도록 하는 장점이 있다.

[Brenden M. Lake, Ruslan Salakhutdinov, Joshua B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266): 1332-1338. 2015.]

설명가능한 인공지능 평가

설명가능한 인공지능 알고리즘은 기존의 머신러닝 알그리즘에서 사용했던 평가 벙법으로 평가할 수 없어 설명가능한 인공지능 만을 위한 효율성 평가 방법을 별도로 개발하여야 한다. DARPA에서는 5가지 효율성 평가 방안을 제시하고 있다.

* 사용자 만족도(User satisfaction) 평가: 사용자가 설명의 명확성과 유용성을 주관적으로 평가하는 것
* 심적 모델(Mental model) 평가: '개별적인 결정’, ‘전체 모델’, ‘평가의 강점과 약점’, ‘무엇을 할 것인가에 대한 예측’, ‘사용자가 어떻게 개입할 것인가에 대한 예측’ 과 같은 요소를 평가하는 것
* 작업 성능(Task performance) 평가: ‘설명이 사용자의 결정, 작업 성능을 향상시키는가?’ 에 대한 정도를 평가하는 것
* 신뢰성(Trust) 평가: ‘적절한 향후 사용 및 신뢰’의 정도를 평가하는 것
* 정확성(Correctability) 평가: ‘오류 식별’, ‘오류 수정’, ‘지속적인 교육’ 으로 정확도를 평가하는 것

첫 번째는 설명의 명확성과 유용성을 사용자가 주관적으로 평가하는 사용자 만족도(User satisfaction), 두 번째는 심적 모델(Mental model), 세 번째는 ‘설명이 사용자의 결정작업 성능(Task performance), 네 번째는 ‘적절한 향후 사용 및 신로’의 정도를 측정하는 신뢰성 평가(Trust assessment), 다섯 번쨰는 정확성(Correctability)으로 ‘오류 식별’, ‘오류 수정’, ‘지속적인 교육’ 항목으로 분류된다.

개별적인 결정들의 이해’, ‘전체 모델의 이해’, ‘평가의 강점과 약점’, ‘무엇을 할 것인가에 대한 예측’, ‘어떻게 사용자가 개입할 것인가에 대한 예측” 과 같은 요소를 평가하는 심적 모델(Mental model)이다. 세 번째는 작업 성능(Task performance),

# 제 3 장 인공지능 표준화

1. **W3C의 인공지능 관련 표준화 동항**

웹 플랫폼과 머신러닝 라이브러리 제공자들이 함께 “현재의 웹 플랫폼을 머신러닝을 위한 더 나은 기반을 제공하는 오픈 웹 플랫폼으로 강화”하기 위해 2020년 8월과 9월에 온라인 Web and Machine Learning 워크숍을 개최하였다. 워크숍에서는 크게 웹 플랫폼 맥락에서 “머신러닝 경험에 대한 개발자와 사용자의 관점”, “머신러닝을 위한 웹 플랫폼 기반”, “브라우저 기반 머신러닝의 기회와 도전” 이라는 세 관점에서 관련 이슈에 대해서 논의를 하였다. 대부분의 워크숍 참가자들은 Web Neural Network API를 통해 노출된 머선러닝 추론을 위한 하위 수준 프리미티브(low-level primitives)를 주요 표준화 대상으로 선정하였다.

또한, 머신러닝에 유용한 JavaScript 언어 기능 개발을 가속화하기 위해 W3C와 Ecma 간 협력의 필요성을 확인하였고 사전에 학습한 머신 러닝 모델을 로딩하기 위한 API의 상호실행 가능성(상호운용성)을 조사하기 위해 ModelLoader API, 미디어 파이프라인 최적화. 머신러닝의 편향성 및 투명성 문제를 해결하는데 도움이 될 머신 판독 모델 카드(machine-readable model card) 제안를 포함한 주제에 대한 웹 인큐베이션에 대한 추가 제안이 있었다.

**브라우저 기반 머신러닝의 기회와 도전**

“브라우저 기반 머신러닝의 기회와 도전” 세션에서는 다음 목표를 달성하는데 도움에 되는 내용에 대해 논하였다.

* 머신러닝을 웹 스택에 적용되는 방법 이해
* 브라우저 내 머신러닝이 머신러닝 생태계에 어떻게 부합되는지 이해
* 머신러닝이 웹 브라우저와 웹 앱에 미치는 영향을 분석
* 머신러닝 API와 포맷에 대한 웹 표준화 기회를 검증

브라우저에서 머신러닝을 적용할 수 있다는 독특한 기회에 대한 특정 관점에서 기회와 도전 도전, 그리고 웹 플랫폼에 머신러닝의 적용을 방해하는 장애요소 도출에 초점을 두었다.

“브라우저 기반 머신러닝”을 위해 추가되어야 할 웹 플랫폼 기능에 대해서도 논의하였다. 특히 “머신러닝 프레임워크를 지원하기 위한 WebGPU API의 적합성” 이슈와 관련해서 새로운 WebGPU 확장에 대해서 활발하게 논의되었다. 머신러닝에 실질적으로 도움이 될 수 있는 새로운 WebGPU AP의 확장, 특히 제안된 서브 그룹 오퍼레이션의 확장에 대해서 논의되었다. 이 특징은 새롭게 부상하고 있는 Web Neural Network API의 핵심 개념인 신경망 계산 그래프(Neural network computational graphs) 등과 같은 일반 그래프를 전문화하는데 필요한 알고리즘의 속도를 높인다.

이는 “낮은 하드웨어 수준에서 상호운용성을 보장하는 것이 중요한지?” 또는 “더 높은 수준의 구성으로 사용자의 요구를 충족시키는 것이 가능한지?” 에 대한 의문을 제기했다. 하드웨어 특화된 오퍼레이션을 지원하는 공통의 중간 표현인 MLIR(Multi-Level IR Complier Framework)가W3C와 MLIR 프로젝트에서 함께 검토되고 있으며 오퍼레이션 수의 폭발을 돕는 이식성 계층(portability layer)이 될 수 있는 MLIR dialect를 제공할 수 있다.

자바스크립트와 웹 어셈블리는 float32자료형은 지원하나 float16자료형을 지원하지 않는다. 이와 같이 float16자료형에 대한 지원하지 않는 것은 양자화된 모델(quantized model)에서 문제가 된다. 모델의 양자화는 더 적은 메모리를 사용하여 빠르게 실행시킬 수 있나 모델의 정확도가 10% 정도 감소할 수 있다. 양자화된 모델은 허용 가능한 일부 환경의 경우(특히 시작 속도가 부족한 경우) 구형 모바일 디바이스에서 필요한 기능이다. float16 자료형에 비교하여 float32자료형은 사용되는 메모리의 양이 증가하고 추론속도가 낮아지는 문제가 있다. 워크숍의 참석자는Ecma TC39에 float16 자료형의 지원을 신속하게 처리해줄 것을 요청하기로 하였으며 웹 어셈블리의 경우 웹 어셈블리 프로그램을 위한 신경망 시스템 인터페이스 표준화를 제안하는 WASI-NN(Web Assembly System Interface for Neural Network) 이니셔티브에서는 필요에 따라 예뮬레이션의 형태로 float 16 및 int8 자료형의 버퍼를 추가하는 것을 고려하고 있다. Node.js 기반의 서버측에서도 머신러닝을 지원하고 있으며 현재 서버측에서는 16비트로 저장할 수 있다. 서버측의 JavaScript 학습 모델을 클라이언트 메모리에 로드하면 float 32로 변환되어 결국 동일한 메모리를 사용하고 이전과 동일한 속도를 갖게된다.

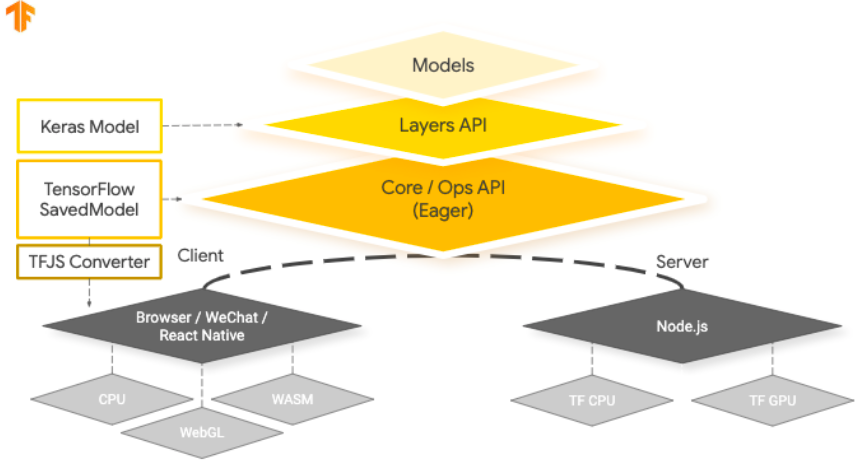


그림 6. TensorFlow.js의 계층적인 아키텍처

[그림 6]은 TensorFlow.js의 계층적인 아키텍처로 최상위 계층은 사전 학습을 통해 만들어진 모델에 해당되머 모델 아래에는 Layer API가 있다. 이 Layer API는 머신러닝을 쉽게 수용할 수 있도록 해주는 높은 수준의 API이다. 그 중 한 예가 Python의 Keras이다. 그 아래에는 수학적 계층인 Core/Ops API를 제공하며 선형대수 등과 같은 수학적인 작업을 수행할 수 있도록 한다. 그리고 이것은 클라이언트 또는 서버와 같은 다른 환경과 통신할 수 있다. 클라이언트 쪽을 살펴보면 브라우저, WeChat, Reactive Native같은 것이 있을 수 있고 각각의 환경은 CPU, WebGL 또는 웹 어셈블리와 같은 다른 백앤드와 통신하는 방법을 이해한다. 그래픽 카드가 존재하면 그래픽 카드에서 더 높은 성능을 얻기 위해 WebGL을 활용할 수 있고 웹 어셈블리를 사용하는 경우 낮은 수준의 명령을 활용하여 CPU에서 높은 성능을 얻을 수 있다. 왼쪽의 TFJS Converter는 Python으로 작성된 모델을 JavaScript로 변환할 수 있다.

“Machine Learning and Web Media” 발표에서 오프라인 형태로 이루어진 라이브 극장의 공연이 COVID-19 때문에 온라인 공연으로 이루어지고 있으며 이 과정에서 머신러닝 알고리즘을 기술적인 관점에서 고려할 것을 논하였다. 타코마 리틀 극장의 첫 번쨰 온라인 제작인 ‘Robin Hood”는 공연의 저장과 다음에 스트리밍을 위해 실시간 공연을 위한 채팅과 유튜브와 결합하여 공연 장면에 사용자 지정 맞춤 배경을 활용했다. 일반적으로 사용자 지정 배경은 로컬에서 캡처된 비디오에 대해 머신러닝 알고리즘을 사용하여 구현되며 전송을 위해 인코딩되기 전에 선택된 배경위에 추출한 배우를 오버레이시킨다.

사람, 남자, 사진, 실내이(가) 표시된 사진

자동 생성된 설명

그림 7. ‘Robin Hood’ 온라인 공연 장면

다른 예로 NBA는 마이크로소프트와 협력하여 팬들이 마치 경기장에서 경기를 보고 있는 것처럼 연출하기 위해서 경기 영상에 팬들의 영상을 합치는 Together Mode를 도입했다. 앞의 예와 같이 이 알고리즘은 경기와 팬을 포함하는 복합 비디오를 생성하고 인코딩하는 서버로 전송하기 저에 캡처된 비디오에 대해 로컬로 작동하는 머신러닝 알고리즘을 적용한다. 오디오의 경우, 머신러닝은 소음 억제를 위해 사용될 수 있으며 비디오의 경우 배경제거, “Together mode” 또는 “funny hats”를 제공할 수있다.

실내, 앉아있는, 테이블, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

그림 8. 마이크로 소프트의 Together mode 서비스 예

비디오 캡처, 머신러닝 모델 적용, 인코딩, 전송의 파아프라인에서 성능은 중요한 요소이기 때문에 파이프라인의 각 단계는 메모리 복사를 하지 않고 이전 단계에서 제공한 버퍼에서 동작하도록 하는 것이 바람직하다. 메모리 복사본을 사용할 경우 브라우저 미디어 파이프라인 내의 머신러닝 웹 앱은 효율성이 떨어진다. 이에 대한 해결책으로 캡처 디바이스는 머신러닝 알고리즘이 자신의 버퍼를 사용할 수 있도록 제공하는 방법이 있다.

**머신러닝을 위한 웹 플랫폼 기반**

머신러닝이 웹 기술 스택에 어떻게 적용되는지 이해하기 위해서 이루어진 “머신러닝을 위한 웹 플랫폼 기반”에 대한 논의가 모델 생성 및 배포와 머신러닝을 위한 웹 기반의 확장에 대해서 이루어졌다. 모델 생성 및 배포와 관련해서는 머신러닝 모델의 포맷(형식), 머신러닝 모델의 보호를 위한 요구사항 및 사용 사례, 초기 실험으로 수행된 브라우저 내 머신러닝 학습, 웹 문맥상에서 여러 디바이스를 학습에 활용할 수 있는 방안에 대해 살펴보았다. 머신러닝을 위한 웹 기반 확장과 관련해서는 WASI-NN과 WebNN의 관련성과 웹을 위한 이기종 컴퓨팅에 대해서 논의하였다.

워크숍에 참석한Web Neural Network API, TensorFlow.js, ONNX, DirectML, MLIR 의 전문가들은 머신러닝 모델의 패키징 및 배포를 위한 머신러닝 모델 형식에 대한 표준이 필요하다고 주장하였다. 그러나 모델 형식의 지속적인 혁신, 진화를 고려할 때 아직 머신러닝의 모델의 형식을 표준화 하기에는 시기적으로 이르다고 결론을 내렸다.

현재 널리 사용되는 모델 포맷이 여럿 존재하는데 그 중 가장 익숙한 것이 TensorFlow 모델이며 최근에는 많은 머신러닝 실무자들이 PyTouch를 사용하고 있다. TensorFlow 모델은 1,000개 이상 있으며 매년 20% 이상 증가하고 있다. PyTouch모델은 수백 가지가 있으며 모든 포맷은 매년 약 20% 이상 성장하고 있다. TensorFlow에서는 MLIR 또는 컴퓨팅 프리미티브를 사용해서 오퍼레이션 보다 낮은 수준에서 컴퓨팅을 정의하여 이 문제를 처리하려고 한다. 이러한 낮은 수준 컴퓨팅 요소를 조합해서 오퍼레이션을 구성하는 것 같이 표준화해야하는 더 작은 집합을 제공할 수 있다. 또한 마이크로소프트는 상호운용이 가능한 포맷으로 ONNX에 대해 작업을 하고 있으며 이는 웹에 적합한 포맷이 될 수 있다. ONNX는 빠르게 성장하고 있으나 이전 버전과의 호환성 보장에 집중하고 있다. ONNX가 장기적으로 좋은 해결책이거나 MLIR은 장기적으로 좋은 해결책이거나 장기적인 접근방식으로 여러 포맷을 지원할 수도 있기 때문에 둘 중 하나를 선택하는 것은 이르다.

초기 접근 방법으로 이미 만들어 놓은 재사용 가능한 머신러닝 모델 빌딩 블록(예: 오퍼레이션)을 가속화하기 위한 Web API를 정의하는데 초점을 두고 있다. 인기있는 컴퓨터 비전 모델과 같은 보통의 모델에서는 컴퓨팅 시간의 90% 이상이 보통 소규모 컴퓨팅 집약적인 작업에 사용되고 있어 제안된 Web API의 범위를 이러한 컴퓨팅 집약적인 작업을 하드웨어로 가속화하는 것으로 지정하는 것은 단기적으로 웹에서 새로운 머신러닝을 지원하는 사용자 경험(user experience)제공하는 가장 실용적인 방법으로 여겨지고 있다.

또한 IPR문제로 일부 머신러닝 모델 제공사는 브라우저에서 실행되는 웹 애플리케이션에서 자신들의 머신러닝 모델을 추출할 수 없도록 머신러닝 모델 보호의 필요성에 대해서 언급이 있었다.[[1]](#footnote-1) 웹에서 비디오 콘텐츠에 대한 액세스 제어 메커니즘과 웹 기반 게임의 3D자산 보호에 실패사례로 얻은 경험으로 바탕으로 머신러닝 모델의 특정 사례에 맞는 해결책을 도출하기 위한 추가 연구가 필요하다. 머신러닝 모델을 보호하는 방식으로 실행하는 유일한 방법으로 대기시간이 길고 개인정보 보호가 쉽지 않다는 문제에도 불구하고 모델을 서버에서 수행하는 것이다. 비디오 콘텐츠 보호와 달리 머신러닝에서는 전처리 작업과 사후 처리 작업도 보호되어야 한다. 방법으로 서버의 보호 환경(예, 보호된 메모리)에서 실행할 수 있도록 JS 코드에 플래그를 지정하여 개발자가 모델을 접근 못하게 할 수 있다.

개인정보의 보호, IPR 보호, 대기시간의 문제를 해결하기 위해 개인정보와 대기사간에 영향을 주는 것은 브라우저에 보관하고 IPR에 민감한 정보는 서버에 두고 Web Transport와 같은 낮은 대기기간 채널로 연결하는 분산 접근 방식을 고려할 수 있다.

대부분의 브라우저 내의 머신러닝은 몇 가지 예외를 제외하고는 학습 보다는 추론에 중점을 두고 있다. 워크숍에서는 브라우저 내에서 학습을 하는 것에 대해서도 논의가 있었다. Teachable Machine 프로젝트는 브라우저에서 분류 모델을 학습할 수 있는 웹 도구이며 브라우저 관련 작업에 대한 초기 학습의 사용 사례로 브라우저 내 전이 학습(transfer learning)을 지원하고 있으며 사용자 경험을 개선하기 위해 구체적인 문제점을 확인해서 해결하고 있다. 한 가지 문제는 백그라운드 탭에서 학습 프로세스를 운영할 수 없다는 점이다. 이에 대한 해결책으로System Wake Lock API가 사용될 수 있으며 이에 대한 사용 사례가 제출되어 검토되었다. 이 논의 결과 추가적으로 관련한 브라우저 API 작업에 대해 가장 가능성 있는 초기 학습 사례로 전이 학습을 사용하여 성공적인 실제 사용 사례(예: Teachable Machine) 의 문서화를 권고하였다. WebNN은 현재 학습에 초점을 두고 있지 않지만 전이학습은 향후 WebNN이 학습에 사용하기 위해 고려해야할 첫 사용사례가 될 것이다. WebNN은 그래프 API로 전이학습에 적합하며 전이학습을 구현할 수 있으나 이를 구현하기 위한 API 형태는 아직 논의 중이다.

장치 간 학습에 관한 논의는 머신러닝 학습 및 웹 플랫폼과의 상호작용에서 엣지 컴퓨팅의 역할에 대해서 이해하는데 도움이 되었다. 이 주제는 웹 컨텍스트에서 연합 학습(federated learning), 분산 학습(distributed learning) 및 강화 학습(reinforcement learning)과 다른 협업 학습(Collaborative learning)에 대한 내용이다. 협업학습은 사용자가 가지고 있는 디바이스를 에이전트로 간주하고 복수의 에이전트를 활용하여 자연스러운 방식으로 모델을 사용하는 것으로 다른 사람이 공유한 다른 모델과 병합할 수 있으며 개인정보 보호 때문에 모델의 일부만 공유될 수 있다. 현재 TensorFlow, PyTouch와 같은 도구는 다중 에이전트 환경에서 작동하지 않는다. 엣지 디바이스, 엣지, 클라우드 상의 모바일 웹을 위한 분산 DNN를 활성화하는 발표에서는 엣지 서버와 종단 디바이스의 컴퓨팅 자원을 활용하기 위한 오프로딩 방식을 제안하였다. 최종적으로Web & Network Interest Group가 협력하여 IG의 엣지 컴퓨팅 작업 스트림으로 머신러닝을 사용할 수 있도록 하기로 했다.

**개발자 관점에서 웹에서의 머신러닝 경험**으로 머신러닝 경험의 작성, 사전 학습된 머신러닝 모델의 재사용, 기술 솔루션과 기술 격차의 논의하였다.

머신러닝에 웹 디자인 원칙 적용

머신러닝의 맥락에서 “점진적 향상과 단계적 성능저하”라는 웹 개발 및 디자인 원칙을 적용하는 방법을 논의하였다. 구체적으로 웹 호환성을 깨지 않고 더 강력한 장치와 브라우저에서 더 많은 머신러닝 기능을 선택적으로 개선할 방법을 논의하였다. 논의의 목표는 이러한 패턴을 사용할 수 있도록 머신러닝 API 설계 메커니즘과 이슈를 식별하여 개발자엑 주어진 경험이 최종 사용자의 디바이스에서 얼마나 잘 작용하는지에 대한 가시성을 제공하는 것이다.

개발자가 하드웨어 가속 장치만을 원하고 실시간 비디오 피드 프로세싱이 가능한 AR 또는 VR과 같이 성능이 중요한 사용 사례에 대해 저가형 장치를 지원하는 않는 경우가 있을 것이다. 이를 가능하게 하는 핵심부분은 모델 성능 특성을 이해하기 위한 모델 스스로 내부 검사를 하는 것이나 신뢰성 있게 제공하기는 어렵다고 지적되었다. 세분화된 운영 수준에서 부터 구현, 플랫폼 또는 하드웨어 기능에 따라 서로 다른 모델이 교체된 전체 모델 수준에 이르기까지 다양한 수준의 폴백(fallback) 메커니즘이 논의되었다.

반대로 모델을 확장하려면 많은 자기 검사 기능이 필요하다.

W3 표준화의 핵심 목표는 머신러닝 API에 대한 적합성 시험에서 제기되는 상호운용성을 보장하는 것이다. WebGPU 논의에서 얻은 내용은 연산은 다양한 형태의 하드웨어 구현 형태를 가질 수 있어 다양한 정밀도를 가진다. 이 경우 수치 정밀도는 브라우저의 WebGPU 구현과 무관하다. WebGPU는 WebNN API와 유사하고 머신러닝 API를 위한 적합성 테스트 절차는 운영자 및 모델 수준 상호 운용성 모두에서 여러 수준의 적합성을 고려할 필요가 있다는 제안이 있었다.

웹 개발자 인체공학 개선

머신러닝 프로세싱은 일반적으로 많은 행렬연산을 포함하기 때문에 이들 연산을 위한 JavaScript 언어의 인체공학은 JS API의 인체공학에 직접적인 역할을 한다.

JavaScript 연산자 오버로딩은 학습의 인체공학을 개선하고 사용자 지정 연산에 도움이 될 수 있으며 제안은 Web Neural Network API의 맥락에서 논의되고 있다. 연산자 오버로딩은 대략적으로 W3C 인큐베이션 단계에 해당하는 Ecma TC39의 1단계 제안 사항에 해당한다.

머신러닝 프레임워크는 WebGL API에

웹에서의 머신러닝 경험 – 사용자 관점

**모두를 위한 웹과 머신러닝**

모델 편향은 소수 집단과 소외된 집단에 영향을 미친다. 투명성을 높이기 위해서 기계 판독이 가능한 모델 카드의

편향과 모델 투명성에 대한 논의에서 소수자와 소수 집단에 미치는 영향을 강조했다.

논의된 실절적인 완화 방안들 중 가장 유력한 아이디어는 웹 앱에서 사용하는 머신러닝 모델의 한계와 성능 특성을 알아내기 위한 브라우저 지원 메커니즘을 포함한다.

음성 인식에서 개인 정보보호 문제는 Web Speech API 표준화의 어려움을 강조하였다.

이 것은 지문 인식과 관련한 문제와 음성 인식 엔진의 클라이언트 또는 서버 측 구현 전략에서 발생하는 문제가 포함된다. 최종 사용자는 현재 자신의 데이터가 서버로 전송되었는지, 저장되었는지, 모든 처리가 클라이언트에 남아있는지 여부를 인식하지 못하고 있다. Web Speech API 규격은 클라이언트측만 인식하도록 의무화할 수도 있고 사용자가 프라이버스 개선을 위해 최소한 이를 요구할 수 있다고 제안하였다. 이러한 제안은 이 영역 작업에 활력을 불어넣기 위해서 Web Speech API를 표준화하기 위한 챔피언(champion)이 필요하다.

프라이버시는 웹 플랫폼의 필수적인 부분이기 때문에 머신러닝 API는 프라이버스를 보호하도록 설계되어야 한다. 이해 당사자들은 플랫폼에 도입되는 새로운 능력에 의해 가능하게 된 새로운 사용자 경험의 프라이버시의 균형을 맞추기 위해 글로벌 사용자 기반에 대한 공동 책임을 진다.

웹 플랫폼을 개발하는 중요한 측면은 도메인 개인정보 전문가들 간의 긴밀한 피드백 루프와 생산적인 공동 노력을 보장하는 것이다.

이를 위해 워크숍 참가자가 추천한 구체적인 다음 단계는 Web NN API의 초기 개인정보보호 검토를 구성하는 것이었다.

다음 단계

표준화의 다음 단계

WebNN API가 머신러닝 기능을 웹플랫폼에 도입하기 위한 올바른 첫 단계라는 공감대를 바탕으로 참가자들은 새로운 W3C 워킹 그룹을 구성하여 이를 표준화화는 작업을 시작해야 한다고 제안한다. 이를 위해 W3C는 Web Machine Learning WG 헌장이 진행 중이라는 사전 통지를 발송했다. 또한 W3C는 머신러닝 처리의 맥락에서 float 16 표준화의 과부하 값고 관련해서 Ecma TC39와 연락을 취할 것을 권장한다.

인큐베이션의 다음 단계

운영체제와 다바이스 간 머신러닝 모델의 상호운용성 구축을 지원하는 Web Machine Learing CG의 Model Loader API를 지속적으로 육성할 수 있도록 지원하였다.

웹 플랫폼 기능을 머신러닝 생태계로 확장이라는 맥락에서 응답할 수 있는지와 방법을 파악하기 위해서 문제 공간의 탐색을 지속적으로 할 것을 요청한다.

* 점진적 개선 접근법을 수용하기 위해 모델에 대한 어떤 내성 데이터가 필요한가?
* 여러 디바이스(엣지 컴퓨팅 포함)에 머신러닝 작업(추론, 학습)을 분산하기 위해 필요한 아키텍처는 무엇인가? 이 주제는 W3C Web & Networks interest Group의 범위와 겹친다.
* 머신러닝 모델 스토리지는 특정 브라우저를 적용해아 하는지? 아니면 File System Access API가 필요한 모든 것을 포함하나?

**관련 그룹**

* Machain learning for Web Community Group
* Web & Network Interest Group

# 제 4 장 설명 가능한 인공지능 기반 서비스

## 제 1 절 자연어 처리

1. **단어 임베딩**

일반적으로 컴퓨터는 자연어처리에서 단어의 의미와 상관없이 각각을 개별적인 기호로 취급한다. 컴퓨터가 자연어를 처리하도록 하기 위해선 자연어를 계산 가능한 형식으로 변형시켜야 한다. 문장에서 총 단어의 수만큼을 벡터의 차원으로 두고, 단어마다 해당하는 차원에 1로 표시하여 단어 벡터로 생성하는 것을 one-hot encoding이라고 한다. 각각의 단어 벡터들을 모두 더한 것을 해당 문장의 벡터라고 할 수 있는데, 단어의 의미를 담지 않은 단어 벡터들의 합이므로 문장의 의미 또한 표현하는데 어려움이 있다. [한국어 임베딩]

이를 해결하기 위해 단어의 의미를 좀 더 세밀한 차원에 벡터로 표현하려 한 것이 단어 임베딩(word embedding)이다. 단어 임베딩은 one-hot encoding과 다르게 각각의 단어 벡터에 의미를 포함시키므로 비슷한 의미의 단어가 있는 경우 벡터 상에서 가까운 곳에 위치하게 된다. 그러므로 단어끼리 의미 비교를 위해 임베딩된 단어 벡터간에 유사도 계산이 가능하다. 유사도가 높으면 의미가 비슷하다는 것을 알 수 있다. 단어 임베딩의 종류에는 LSA, Word2Vec, GloVe, FastText 등이 있는데 이와 같은 단어 임베딩은 단어마다 벡터 값이 고정되어 있기 때문에 단어가 사용된 문맥은 고려하지 않는다. 즉 같은 단어이지만 다른 맥락에서 사용되었을 때도 같은 벡터 값으로 출력되기 때문에 차이를 알 수 없다.

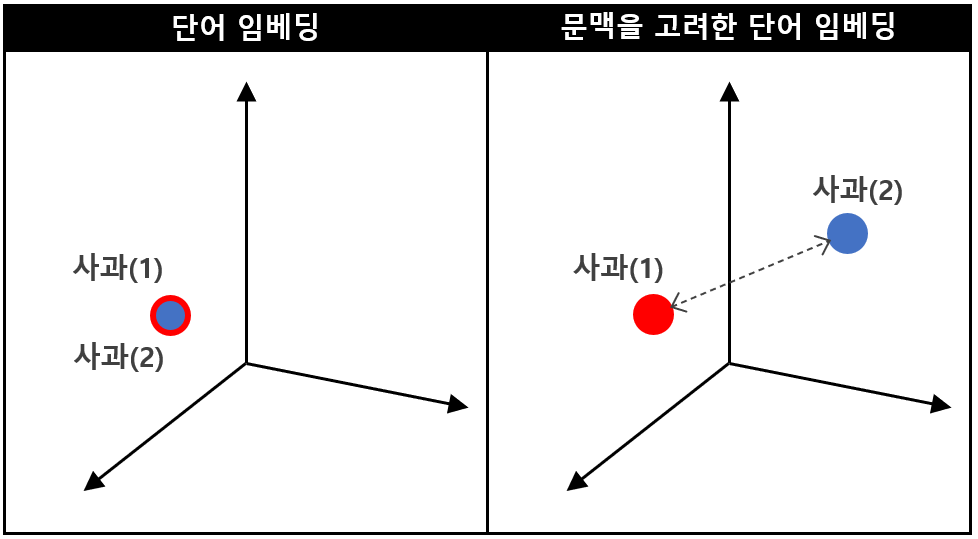


그림 9. 임베딩 방법 비교

단어가 사용된 문장의 맥락을 파악하기 위해 등장한 것이 문맥을 고려한 단어 임베딩(contextualized word embedding)이다. 문맥을 고려한 임베딩에선 개별 단어뿐만 아니라 문장 전체의 문맥적 의미를 함축하기 때문에 임베딩된 벡터를 인공지능을 통해 학습시켰을 때 기존의 단어 임베딩보다 학습 효과가 좋은 것으로 알려져있다. [그림 9]은 단어 임베딩과 문맥을 반영한 단어 임베딩의 차이를 그림으로 나타낸 것이다. 단어 임베딩은 같은 문장에서 사용된 동일한 단어가 사용된 의미에 상관없이 같은 벡터를 출력하지만, 문맥을 고려한 임베딩의 경우 파악한 문맥에 맞게 각각의 벡터를 출력하게 된다. 문맥을 고려한 단어 임베딩 종류로는 BERT(Bidirectional Encoder Representtions from Transformer), ELMo(Embeddings from Language Models), GPT(Generative Pre-Training)등이 있다.

본 연구에서는 BERT가 사용되었다. BERT의 출력으로 얻을 수 있는 단어 벡터들은 문맥을 파악하여 생성된 벡터이기 때문에 단어의 의미뿐만 아니라 문장의 의미 또한 유지시킨다. 단어 벡터는 12개의 encoder layer를 거쳐 출력되는데, 마지막 레이어의 출력보다 마지막 4개 레이어 출력의 연결이 벡터의 표현을 더 잘 나타낼 수 있다. BERT는 출력 뒤에 원하는 분류 모델을 추가시켜 분류 학습을 하게 된다. 본 연구에선 설명가능한 인공지능을 위해 문장 벡터에 어떤 단어 벡터가 영향을 미쳤는지 해당 라벨로 분류되었는지 분석하였다.

1. **BERT(Bidirectional Encoder Representtions from Transformer)**

BERT는 인공지능 기반 자연어처리 분야에 등장한 구글의 강력한 언어모델이다. 기본적으로 Wikipedia 및 Book data와 같은 대용량 unlabeled data로 모델을 사전 학습(pre-training)하여 문맥을 파악할 수 있도록 구축된 모델이다. 사전학습된 모델 BERT에 분류하고자 하는 task에 맞게 labeled data로 추가 학습하여 분류 성능을 높인다. BERT는 사전학습에는 Masked Language Model(MLM)과 Next Sentence Prediction(NSP) 두 가지 방법이 사용된다. MLM은 랜덤한 위치의 토큰을 마스킹 시킨 뒤, 나머지 토큰들을 기반으로 마스킹된 토큰의 단어를 예측하고, NSP는 두 개의 문장을 입력하여 두 문장이 연속된 문장인지 예측하며 학습 성능을 높인다.

BERT Encoder는 MLM과 NSP를 위해 Encoder-Decoder 아키텍처로 이루어진 Transformer의 Transformer Encoder 구조를 기반으로 한다. Transformer는 RNN(Recurrent Neural Network)과 마찬가지로 시퀀스 데이터를 처리할 수 있도록 설계된 모델이다. 하지만 RNN과 다르게 Transformer는 시퀀스 데이터를 순차적으로 처리하지 않는다. 순차적으로 처리되는 RNN은 문장이 길어지는 경우 기울기 소실 문제(Vanishing gradient problem)로 인해 초기 토큰에 대한 정보가 사라질 수 있다는 단점이 있는 반면, Transformer는 병렬처리하여 초기 토큰의 정보를 보존할 수 있다. 또한 데이터 병렬처리 방법은 학습 시간을 단축시킨다. [BERT 논문 정리]



그림 10. BertTokenizer의 토큰화

BERT 모델에 데이터를 입력하기 전에 [그림 10]과 같이 문장을 토큰으로 분리하는 토큰화 과정이 필요하다. 토큰화에 사용될 토크나이저도 모델과 마찬가지로 Transformers의 사전 학습된 토크나이저인 BertTokenizer가 사용되었다. BertTokenizer는 WordPiece를 기반의 토크나이저로, WordPiece란 서브워드 토큰화(Subword Tokenization) 기반의 토크나이즈 알고리즘이다. 이는 자주 등장하여 이미 학습되어 있는 단어는 두고, 자주 등장하지 않는 단어를 더 작은 단위로 분리하여 의미 있는 단어로 토큰화하는 방법이다. 예를 들어 'ab'라는 단어가 있을 때 'ab'가 등장할 확률보다 'a'와 'b'로 등장할 확률이 높다면 더 작게 분리하는 것이다. 영어에 맞는 형태소 분석이 사용되지 않고, 분리된 토큰 사이에 '##' 기호로 연결을 표시하여 한국어도 처리가 가능하다. BertTokenizer를 사용하여 토큰화된 토큰들은 index 값으로 변환하여 모델에 입력으로 들어가게 된다. [토크나이저 설명]

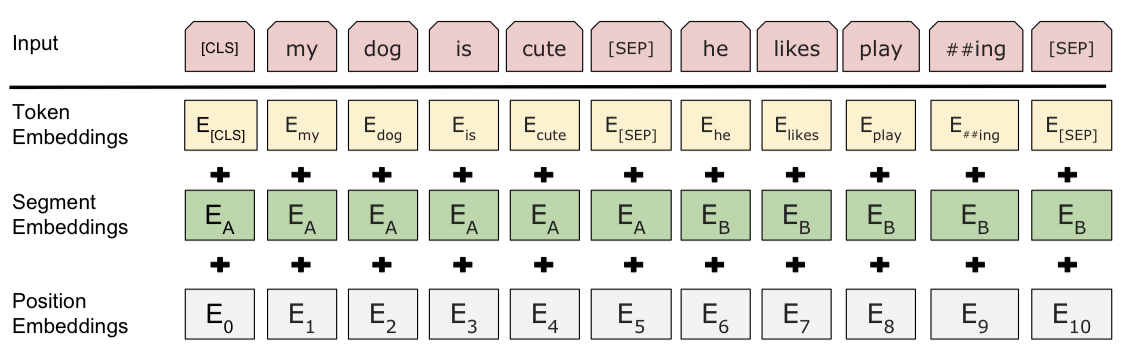


그림 11. BERT의 입력데이터 임베딩

[그림 11]은 토큰을 BERT Encoder의 입력 데이터에 맞게 임베딩하는 것을 나타낸 것이다. [그림 11]의 Input에서 [CLS] 토큰은 BERT Encoder의 전체 layers를 거치고 난 후 token sequence의 결합된 의미를 가지게 되어 분류에 사용되는 토큰이며, [SEP] 토큰은 문장이 두 개 들어갈 때 문장을 구분하는 것에 사용되는 토큰이다. BERT Encoder에 입력하기 위해선 Embedding 과정을 거쳐야 하는데, Token embeddings과 position embeddings이 사용되는 Transformer Encoder과는 다르게, BERT Encoder에서는 토큰이 어느 문장에 속하는지 알 수 있도록 segment embeddings까지 더하여 입력 데이터를 생성한다.

BERT Encoder는 Multi-Head Attention이 사용되는데 이는 임베딩된 입력 데이터의 차원을 Head 수만큼 나누어 각각 Attention을 계산하고 추후에 합치는 과정이다. Attention 은 모든 토큰 간의 상관관계에 따라 가중치를 계산하는 메커니즘이다. 주어진 토큰 벡터 Query에 대해서 다른 토큰 Key와의 관계를 계산하여 가중치를 구하게 되는데 이를 Value에 적용시켜 출력값을 생성한다. 모든 토큰에 대한 attention 계산은 다음과 같이 최적화된 행렬 계산으로 표현될 수 있다.

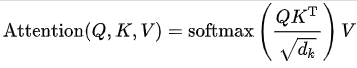


그림 12. Attention 계산 수식

설정한 layers 수만큼 BERT Encoder가 반복된 후 [CLS] 토큰은 token sequence의 결합된 의미를 가지게 되는데, 여기에 분류 모델을 추가되면 문장 분류 학습이 가능하게 된다. BERT는 다양한 task에 사용될 수 있으며, 입력 데이터와 분류 모델을 어떻게 Fine-tuning 하느냐에 따라 다른 유형으로 변형 가능하다.

표 4. BERT 클래스 종류

|  |  |
| --- | --- |
| **클래스** | **요약** |
| BertModel | 모든 모델의 틀  (사전 학습 완료) |
| BertForPreTraining | MLM과 NSP가 추가된 모델  (사전 학습 완료) |
| BertForMaskedLM | MLM가 추가된 모델  (사전 학습 완료) |
| BertForNextSentencePrediction | NSP가 추가된 모델  (사전 학습 완료) |
| BertForSequenceClassification | 문장 분류 모델이 추가된 모델  (사전 학습되었으나 분류 모델은 추가 훈련되어야 함) |
| BertForTokenClassification | 토큰 분류 모델이 추가된 모델  (사전 학습되었으나 분류 모델은 훈련되어야 함) |
| BertForQuestionAnswering | 질문에 맞는 토큰 분류 모델이 추가된 모델  (사전 학습되었으나 분류 모델은 훈련되어야 함) |

[표 4]는 BERT를 다양한 유형으로 사용할 수 있게 해주는 클래스 종류다. 모든 모델의 틀이 되는 BertModel을 기반으로 다양한 사전학습에 사용되는 MLM과 NSP 모델이 각각 추가된 클래스와 두 가지 모델 모두 추가된 클래스가 있다. 해당 클래스들은 모두 사전 학습되었으며 추가 학습시킬 필요가 없고, 이와 다르게 사전 학습이 되어있으나 추가 학습이 필요한 클래스들도 있다. BertForSequenceClassification 클래스는 문장을 label에 맞게 분류해주는 모델이 추가되어있기에, 분류 모델을 원하는 분류 유형에 맞는 데이터셋으로 학습시킬 필요가 있다. BertForTokenClassification 클래스 또한 문장의 토큰 각각을 label에 맞게 분류해주므로 추가 학습이 필요하다. 마지막으로 BertForQuestionAnswering 클래스는 Q&A를 위해 문장에서 정답을 찾아낼 수 있도록 추가 학습시켜야한다. [pytorch BERT]

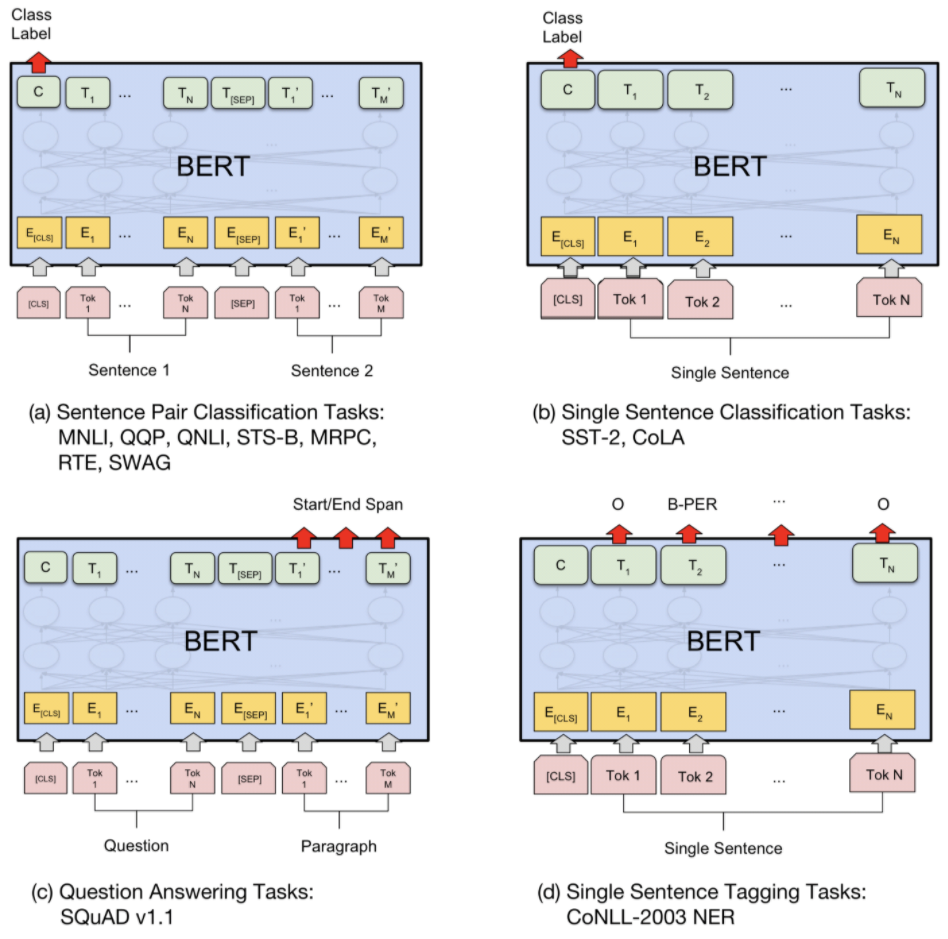


그림 13. BERT의 사용 유형

[그림 13]의 (a)와 (b)와 같은 경우는 BertForSequenceClassification 클래스를 생성하여 사용할 수 있는 유형으로, 분류하고 싶은 개수에 따라 분류 모델을 조정하고, 문장의 의미를 가지는 [CLS] 토큰의 벡터를 사용하여 class를 분류한다. (c)와 같은 Question Answering(QA) task 경우엔 BertForQuestionAnswering 클래스를 생성하여 사용할 수 있는 유형으로, Question와 Paragraph를 [SEP] 토큰으로 분류하고, Question에 정답을 Paragraph에서 찾아낸다. (d)와 같은 경우는 BertForTokenClassification 클래스를 생성하여 사용할 수 있는 유형으로, Named Entity Recognition(NER)이나 형태소 분석처럼 각 토큰이 어떤 class를 가지는지 각각을 분류할 수 있다.

BERT는 위의 내용처럼 downstream task 학습을 통해 사전학습된 parameter들을 변경하며 진행되는 Fine-tuning 접근법뿐만 아니라 Feature-based 접근법으로도 사용 가능하다. Feature-based 접근법은 BERT를 통해 진행된 Feature extraction을 특정 분류를 위한 추가 모델에 입력하여 학습시키는 형태이다. Feature-based 접근법은 다른 모델인 ELMo(Embeddings from Language Model)에서도 사용되는 접근법으로, 사용하는 이유는 Fine-tuning 접근법을 사용했을 때 사용되는 pooler는 분류 모델에 들어가기 전에 [CLS] 토큰을 처리하는 과정인데, pooler의 출력 값은 분류에는 도움을 주지만 문장의 의미를 표현하기엔 좋은 값이 아니기 때문이다. (This output is usually not a good summary of the semantic content of the input, you’re often better with averaging or pooling the sequence of hidden-states for the whole input sequence.) [huggingface]

BERT 논문에 따르면 Feature-based 접근법에는 다양한 방법들이 존재하는데 이 방법들은 성능에도 영향을 미친다. 여러 방법들을 간단하게 설명하자면 첫 번째로BERT Encoder의 입력으로 사용되는 토큰을 사용하는 방법과 BERT Encoder의 출력인 토큰을 사용하는 방법이 있다. BERT Encoder의 layer 중 마지막에서 두 번째 layer의 토큰을 사용할 수도 있으며 모든 layer를 합해서도 사용 가능하다. 가장 많이 사용되는 방법 두 가지로는 BERT Encoder의 마지막 layer 4개를 합하는 방법과 layer 4개를 Concat하는 방법이 있는데 Concat하는 방법은 Fine-tuning 접근법의 성능과의 차이가 크게 나지 않는다. 이를 보아, BERT는 Feature-based에서도 효과적이라고 볼 수 있다. [BERT paper]

1. **BERT 세부 기능 분석**

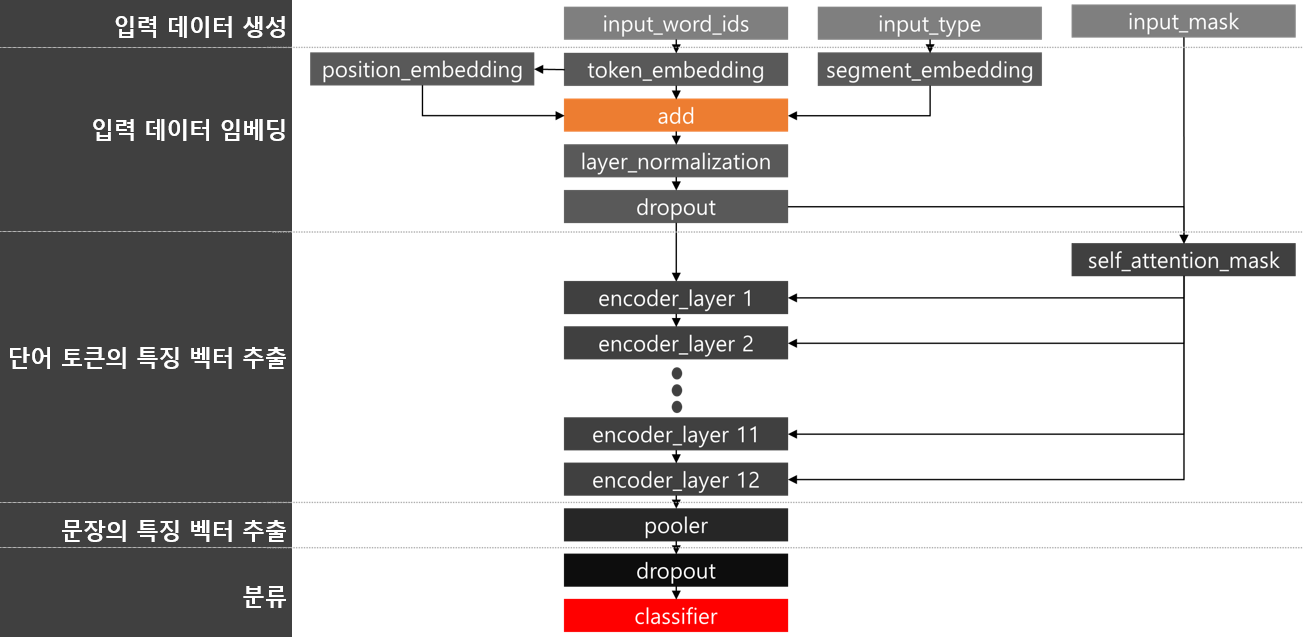


그림 14. BERT구조 블록 다이어그램

[그림 14]는 BertForSequenceClassification 클래스를 생성하였을 때 전체 모델의 구조를 블록 다이어그램으로 표현한 것이다. 마지막 단계에 분류 모델을 추가하여 문장 분류가 이루어진다. BERT의 입력으로는 세 가지 데이터가 준비되어야 한다. 첫 번째로 입력 문장을 토큰화 한 뒤, 토큰을 index 값으로 변환시킨 input\_word\_ids 데이터가 필요하다. 두 번째로는 여러 문장이 입력으로 들어올 때 토큰이 어느 문장에 소속되어 있는지 알 수 있도록 input\_type 데이터가 있어야 하며, 마지막으로 토큰과 토큰이 아닌 패딩 위치를 알 수 있도록 해주는 input\_mask 데이터가 필요하다.

입력 받은 데이터들은 임베딩 과정을 거치게 되는데 input\_word\_ids 데이터로 token\_embedding 벡터와 토큰의 순서를 알 수 있는 position\_embedding 벡터를 얻게 된다. input\_type 데이터로 소속 문장을 알 수 있도록 해주는 segment\_embedding 벡터를 얻게 되며, input\_mask는 다음 단계인 encoder 단계에서 토큰과 패딩을 구분할 수 있도록 활용된다. 임베딩 과정을 통해 얻은 세 가지 벡터를 합하여 임베딩된 입력 데이터를 얻을 수 있다.

임베딩된 입력 데이터는 토큰의 특징 벡터를 추출할 수 있는 encoder로 입력된다. 이 과정에서 데이터는 Multi-Head Attention을 통해 각 단어 사이의 관계성에 맞게 가중치가 계산된다. 설정된 여러 layer를 거치게 되면서 입력된 문장의 문맥을 파악하게 되고 그에 맞는 특징 벡터를 출력한다.

encoder 단계에서 출력된 토큰의 특징 벡터들 중에서 첫 번째 토큰인 [CLS] 토큰은 다음 단계인 pooler의 입력으로 사용된다. pooler에서는 분류 모델이 분류를 더 효율적으로 진행할 수 있도록 문장 벡터인 [CLS] 토큰을 처리하는 과정이다. 분류 모델은 pooler의 결과인 문장 벡터를 입력으로 받아 결과에 맞는 label을 학습하게 된다.

## 제 2 절 설명 가능한 리뷰의 감성 탐지 서비스

표 5. 리뷰 감정 탐지 서비스 실험 환경

|  |  |
| --- | --- |
| **개발 환경** | Colab |
| **런타임 환경** | GPU |
| **모델** | BertForSequenceClassification |
| **토크나이저** | BertTokenizer |
| **데이터셋** | Naver sentiment movie corpus |
| **데이터 수** | 학습 데이터: 150,000 |
| 테스트 데이터: 50,000 |
| **라벨** | 긍정: 100,000 |
| 부정: 100,000 |
| **테스트 정확도** | 87% |

실험을 위해 BERT 모델에 긍정 및 부정으로 이루어진 영화 리뷰 데이터를 학습시켰으며, 각 라벨로 분류될 때 어떤 단어의 영향이 컸는지 테스트했다. 임베딩된 벡터는 사칙 연산을 통해 단어와 문장 사이의 의미적 관계를 도출할 수 있다. 이를 기반으로 우리는 문장 벡터와 단어 벡터들 사이의 유사도를 측정하였을 때, 유사도가 높은 벡터가 문장에 영향을 크게 미쳤다고 판단했다. [표 5]는 해당 테스트를 위한 실험 환경이다.

실험에서는 Transformers가 제공하는 클래스 중 BertForSequenceClassification를 사용했다. BertForSequenceClassification는 데이터 분류를 위해 단일 Linear layer가 추가된 BERT 모델이다. 클래스 생성 시 사전 학습된 BERT 모델과 추가된 분류 레이어가 입력 데이터에 대하여 특정 분류 작업을 학습한다. 사전 학습된 BERT 모델 중에서도 다양한 언어가 지원되는 bert-base-multilingual-cased 모델이 사용되었는데, 해당 모델은 Wikipedia 지분율이 높은 상위 104개의 언어로 학습되었기에 영어 이외의 언어로도 사용 가능하다. 사용 가능한 언어 목록에 한국어도 포함되어있어 해당 모델을 선택하였다.

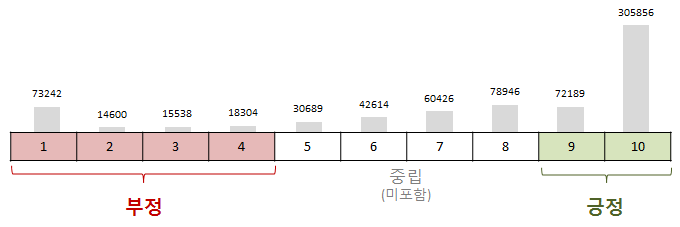


그림 15. 네이버 영화 리뷰 데이터 수집[영화 리뷰 분류]

실험 데이터로는 한국어 데이터 텍스트 분류 테스트에 많이 사용되는 네이버 영화 리뷰 데이터 nsmc(Naver sentiment movie corpus)가 사용되었다. 학습 데이터 150,000개와 테스트 데이터 50,000개를 합쳐 총 200,000개로 구성된 데이터로써 문장이 긍정일 때 label이 1, 부정일 때 label이 0으로 표시되어 있다. 네이버 영화의 리뷰 중 영화당 100개의 리뷰를 수집하였으며, [그림 15]과 같이 1~10점까지의 평점 중에서 중립적인 평점인 5~8점 리뷰는 제외하고 1~4점은 부정으로, 9~10점은 긍정으로 포함시켰다. 긍정, 부정 label는 동일한 비율로 각 100,000개 수집되었다. [nsmc]

모델과 토크나이저, 데이터셋이 준비되었으면 첫 번째로 데이터셋을 BertTokenizer을 사용하여 토큰화하는 것이 필요하다. 토큰화된 토큰들은 숫자로 된 index로 변환하여 input\_word\_ids 데이터를 생성하였다. BERT 모델은 입력 시퀀스의 길이 제한이 있기 때문에 최대 시퀀스 길이에 맞춰 길이를 토큰을 잘라내거나 모자란 부분을 0으로 패딩 처리하여야 한다. 이때 문장에서 토큰이 존재하는 부분과 의미없는 부분을 알려주는 input\_mask 데이터를 생성하였다. 학습 시 input\_mask 데이터가 입력되면 BERT 모델에서 패딩 부분은 학습하지 않아 속도를 향상시킬 수 있다. 전처리된 학습 데이터를 분리하여 10%는 검증 데이터로 활용되었다.

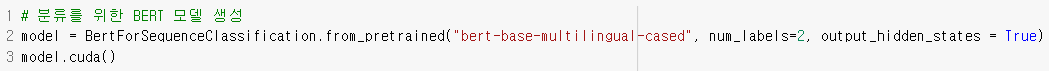


그림 16. BERT 모델 생성

모델을 생성할 땐 [그림 16]과 같이 output\_hidden\_states를 True로 설정해야 BERT Encoder의 계산 값을 출력할 수 있다. optimizer로는 AdamW가 사용되었으며 epochs를 4로 설정하여 반복하였다. 생성된 모델에 전처리된 학습 데이터를 사용하여 학습시킨 후, 테스트 데이터도 위와 같은 전처리를 거쳐 모델을 테스트한 결과로 87%의 정확도를 얻을 수 있었다.

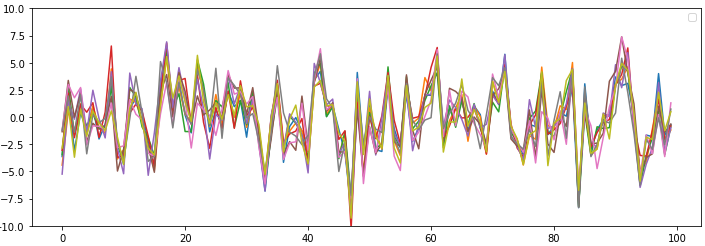
우리는 BERT Encoder를 거쳐 출력된 토큰 벡터와 문장 벡터를 보고 어떤 토큰이 문장에 큰 영향을 주었길래 분류 모델이 해당 label로 분류를 했는지 분석하고자 하였다. 첫 번째로 알 수 있었던 결과는 [그림 17]와 같이 BERT Encoder의 마지막 layer에서 출력된 토큰 벡터들을 히스토그램으로 그려보면 유사한 패턴을 가진다는 것이다.

그림 17. 토큰 벡터 히스토그램

뿐만 아니라 여러 문장으로 테스트했을 때도 같은 label로 분류되는 문장들 사이에 유사한 패턴이 존재한다는 것을 알 수 있었다. [그림 18]의 첫 번째 히스토그램은 label이 부정으로 분류된 문장들의 문장 벡터들을 나타낸 것이며, 두 번째 히스토그램은 label이 긍정으로 분류된 문장들의 문장 벡터들을 나타낸 것이다. 각 10개(더 추가)의 문장으로 테스트하였으며, 문장들의 백터 값을 비교했을 때 유사도가 크게 높진 않았지만 각 분류 label에 따라 다른 패턴을 가진다는 것을 알 수 있었다. 각 label에 맞는 패턴이 있다는 근거를 기반으로 label 패턴에 가깝게 생성된 문장 벡터와 유사도가 높은 토큰 벡터가 label 분류에 영향을 미쳤다고 판단했다.

그림 19. 코사인 유사도 계산 코드

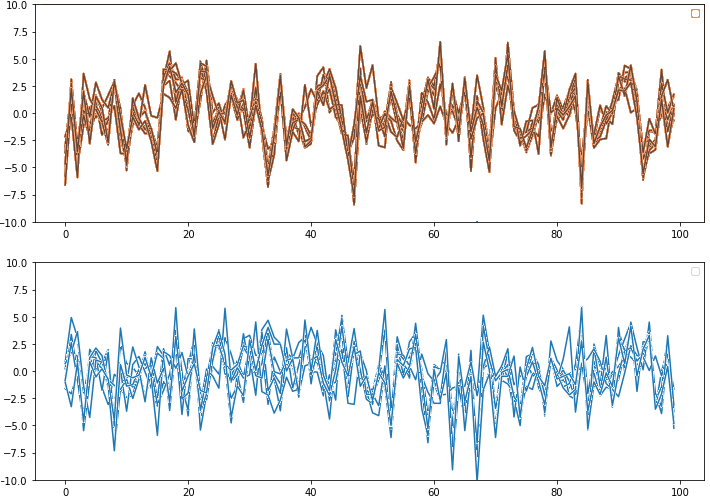


그림 18. 부정 및 긍정 문장 벡터 히스토그램

코사인 유사도를 사용하여 문장 벡터와 토큰 벡터 두 벡터 간의 유사도를 계산했다. BERT의 출력 벡터는 768차원을 가지게 된다. 두 벡터가 완전히 동일한 경우에는 유사도 결과는 1의 값을 가지며 두 벡터가 관련성이 없을수록 0에 가까워진다. 문장 벡터는 토큰 벡터의 합으로 이루어진 벡터이기 때문에 768차원을 구성하고 있는 정보들은 같다. 문장 벡터와 토큰 벡터들의 코사인 유사도를 계산하여 문장의 특징 벡터와 가장 유사한 토큰, 즉 단어는 어떤 것들이 있는지 분석했다.

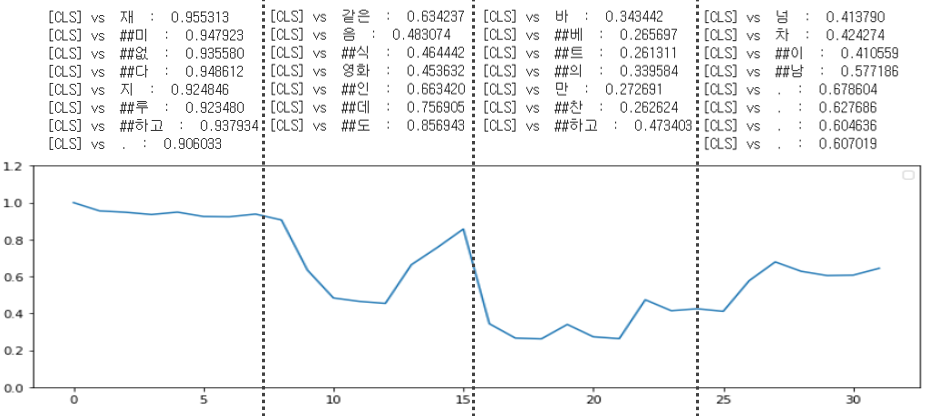


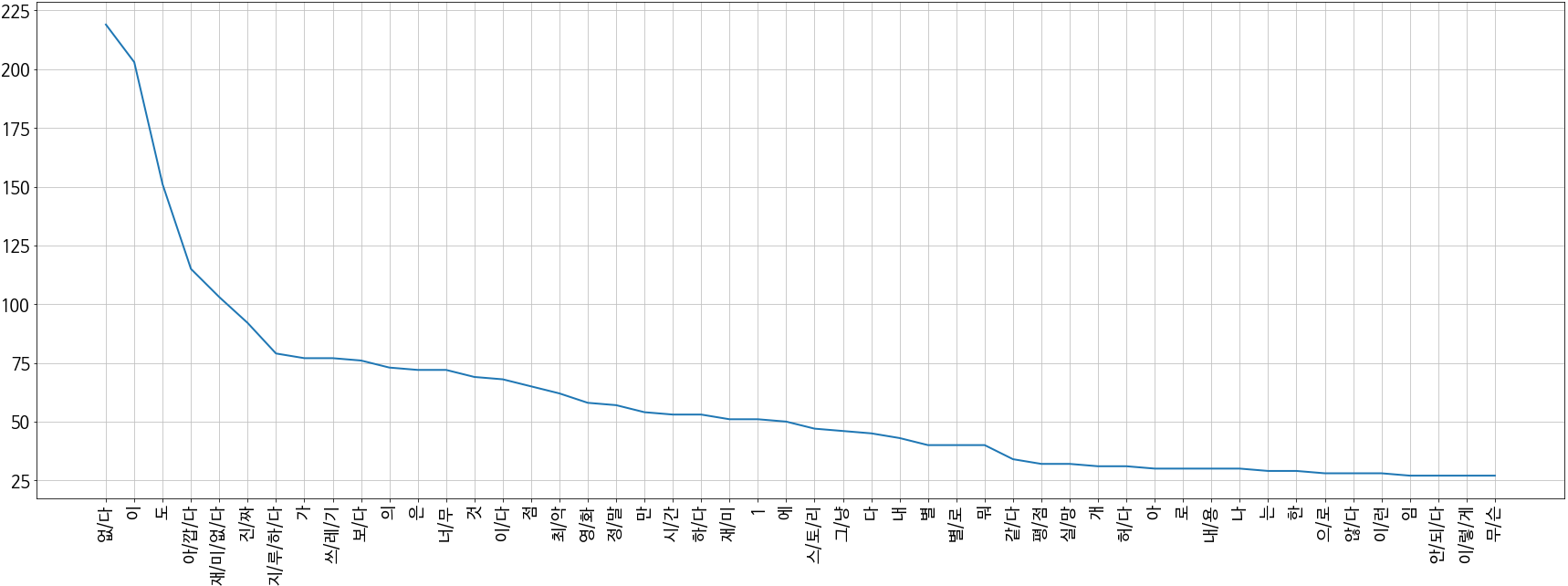
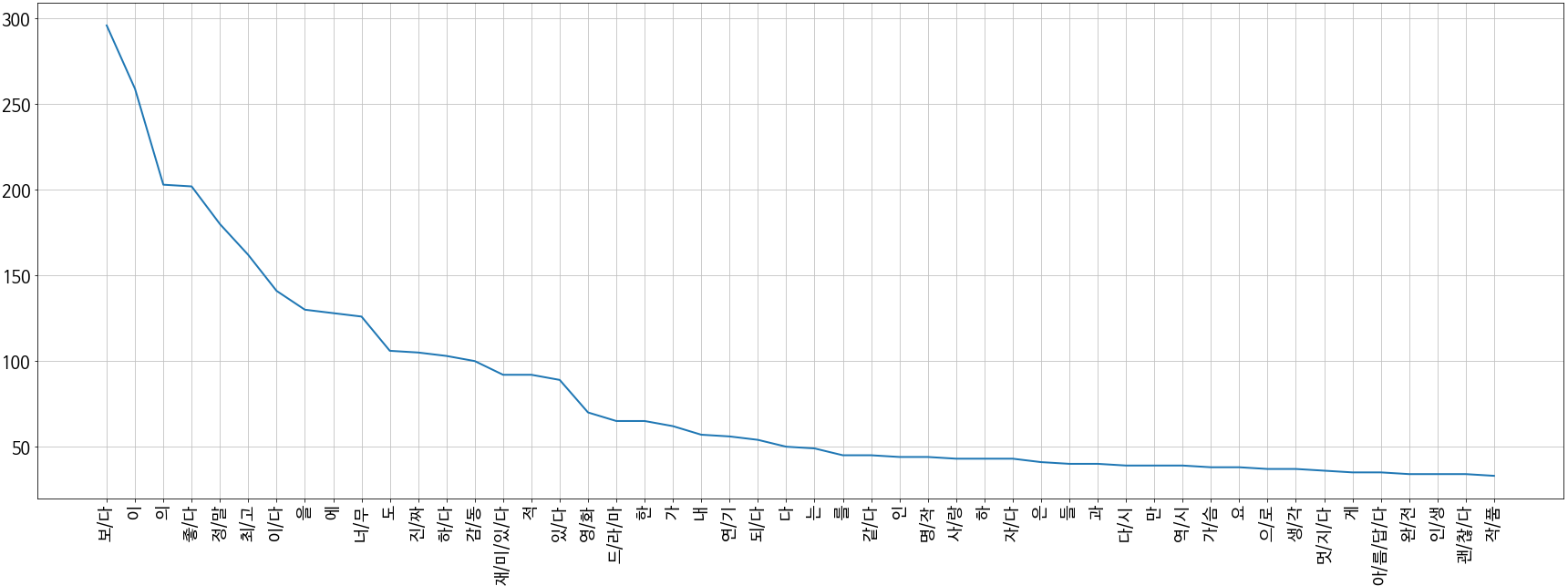
그림 20. 문장 벡터와 토큰 벡터의 코사인 유사도

[그림 20]은 코사인 유사도 테스트한 여러 문장 중 하나인 ‘재미없다 지루하고. 같은 음식 영화인데도 바베트의 만찬하고 넘 차이남....’ 이라는 문장의 결과를 히스토그램으로 나타낸 것이다. 각 토큰들과 코사인 유사도 계산을 한 [CLS] 토큰은 문장 벡터를 의미하며, 숫자 값은 유사도 결과를 나타낸 것이다. 해당 문장 벡터는 분류 모델을 거쳤을 때 부정으로 분류된 문장이다. 문장 벡터의 패턴은 [그림 18]의 첫 번째 히스토그램인 부정 패턴과 유사하게 출력된다. 부정 패턴에 가까운 문장 벡터와 각 토큰들의 코사인 유사도를 계산했을 때 결과를 보면, 토큰들 중에서도 유사도가 높은 부분이 존재한다는 것을 알 수 있다. 히스토그램을 네 등분하여 비교했을 때 첫 번째 부분과 나머지의 결과 값이 차이가 나는 것을 볼 수 있다. 첫 번째 부분의 토큰으로는 ‘재’, ‘##미’, ‘##없’, ‘##다’, ‘지’, ‘##루’, ‘##하고’, ‘.’ 등으로 부정의 의미를 가지는 단어들로 구성되어 있다. 나머지 부분의 내용을 보면 감정을 나타내기보단 타영화와 비교하는 내용으로써, 긍정인지 부정인지를 알 수 있는 감정적인 단어가 사용되지 않았음을 알 수 있다.

우리는 50개의 문장으로 위와 같은 테스트를 반복하여 문장 벡터와 토큰 벡터의 코사인 유사도를 수동으로 확인했다. 히스토그램으로 강조되는 부분의 토큰이 감정을 나타내는 단어가 맞는지 확인하였으며, 긍정 및 부정으로 분류될 때 각 label의 문장에서 어떤 단어들이 강조되고 있는지 정리하였다. 50개의 문장을 테스트하였을 때 분류에 맞는 단어가 강조되고 있다고 판단되었다.

더 많은 문장에서 강조되는 단어가 어떤 것들이 있는지 분석하기 위해 5000개의 문장에서 코사인 유사도가 높은 부분의 키워드를 자동으로 출력하였다. BERT의 토큰화는 형태소 단위로 토큰화되지 않기 때문에 감정을 나타내는 명사를 정확하게 찾아낼 수는 없다. 우리는 강조된 단어들의 개수를 통계내기 위하여 출력된 키워드에 어간 추출을 거쳐 어형이 변형된 단어로부터 접사 등을 제거하고 그 단어의 어간을 분리했다. [그림 21]의 첫 번째 그래프는 5000개의 문장 중 긍정으로 분류된 문장들에서 강조된 단어들의 통계를 나타낸 것이며, 두 번째 그래프는 부정으로 분류된 문장들에서 강조된 단어들의 통계를 나타낸 것이다. 긍정으로 분류된 문장의 통계에서는 ‘이’, ‘의’, ‘에’ 등의 조사와 ‘정말’, ‘너무’ 등의 부사를 제외하고‘좋다’, ‘최고’, ‘감동’, ‘재미있다’, ‘사랑’, ‘아름답다’ 등의 긍정적인 부분이 코사인 유사도가 높게 계산되었으며, 부정으로 분류된 문장의 통계에서는 ‘없다’, ‘재미없다’, ‘아깝다’, ‘쓰레기’, ‘최악’, ‘실망’, ‘별로’ 등의 부정적인 부분이 코사인 유사도가 높게 계산되었다. 각 분류에 맞는 단어가 강조되는 것을 보아 분류에 영향을 미치는 단어가 문장과의 코사인 유사도가 높게 측정된다는 것을 알 수 있었다. [리뷰 단어 통계, 실험자료]

그림 21. 긍정과 부정으로 분류된 문장에서 사용된 단어 통계



우리는 문장 벡터와 단어 벡터의 코사인 유사도 계산을 통해 유사도 점수가 높은 단어를 강조하면 설명 가능한 리뷰의 감정 탐지 서비스가 가능할 것이라 판단됐다. 다음은 리뷰 데이터셋으로 학습된 BERT 모델에 문장을 입력시켜 테스트하는 단계이다. 입력된 문장이 label에 맞게 잘 분류되는지 확인하고, 문장에서 어떤 단어가 색깔 시각화를 통해 강조되는지 테스트하였다.

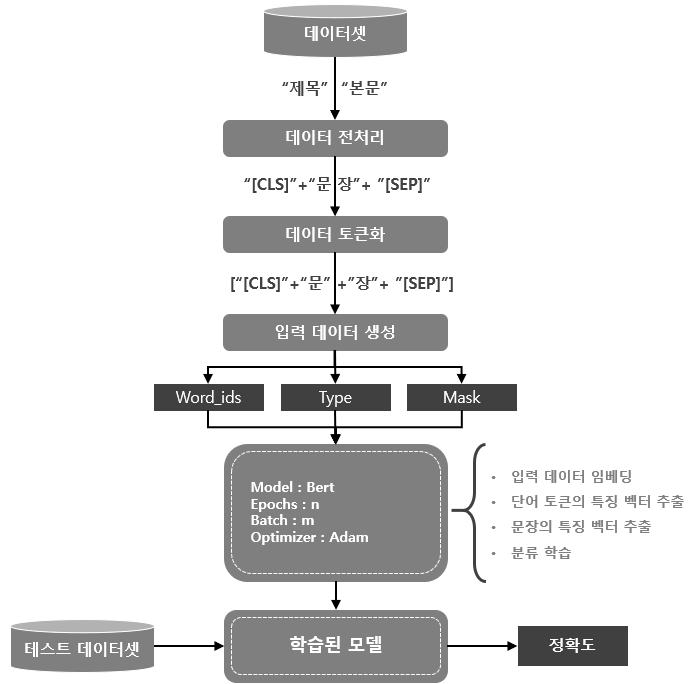


그림 22. BERT 기반 모델 블록도

[그림 22]는 BERT 기반 모델을 사용하기 위한 단계를 블록으로 나타낸 것이다. 준비된 데이터셋을 전처리하고, 토큰화를 거쳐 생성된 토큰으로 입력 데이터를 생성한다. 생성된 입력데이터를 BERT 모델에 입력하여 학습시키거나, 학습된 BERT 모델에 테스트를 할 수 있다. BERT 모델의 구조는 [그림 14]과 같으며, 데이터를 임베딩하고 토큰마다의 관계성을 계산하여 특징 벡터를 생성한다. 그 다음 BERT 구조에 추가되어 있는 분류 모델에 특징 벡터를 입력시켜 학습할 수 있다. 다음에 나올 그림들은 모델을 학습하는 단계부터, 학습된 모델을 사용하기 위하여 테스트 데이터셋을 전처리하여 테스트하는 과정을 나타낸 것이다. 각 단계는 코드를 보며 자세히 설명한다.

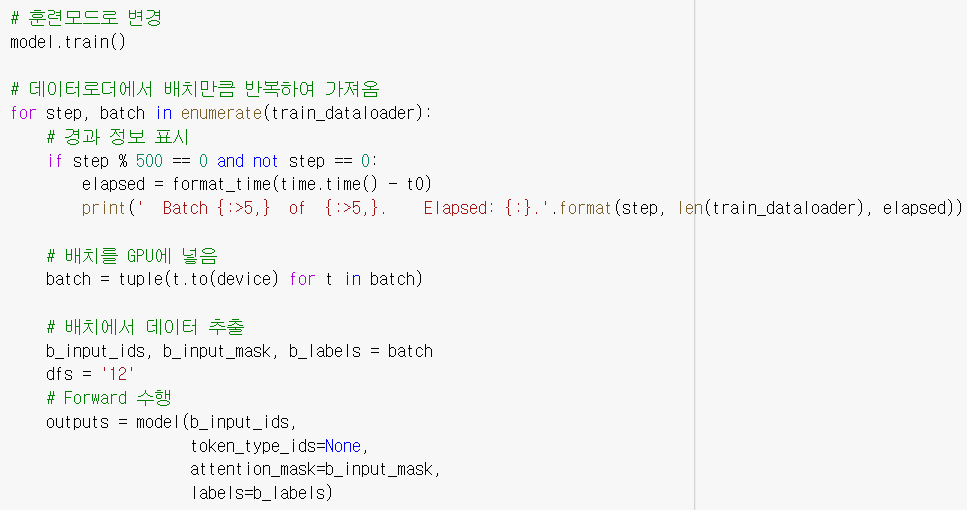


그림 23. 리뷰 감정 분류 모델 학습

[그림 23]는 학습데이터로 BERT 모델을 학습시키는 과정이다. batch 수만큼 가져와서 모델에 학습시키는 과정을 반복한다. 런타임 환경으로 GPU를 사용하였기에 데이터를 GPU 디바이스에 입력시킨다. 모델에 입력으로는 토큰의 인덱스 값을 뜻하는 input\_ids 데이터와 토큰이 존재하는지 여부를 알려주는 input\_mast 데이터가 들어간다.

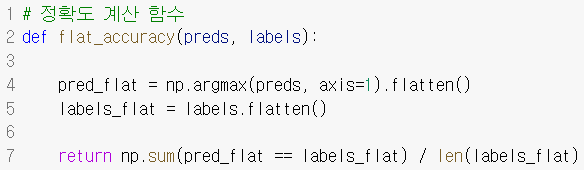
[그림 24]는 모델의 정확도를 계산하기 위한 함수 생성을 나타낸 것이다. 학습 정확도와 테스트 정확도를 게산할 수 있다.

그림 24. 정확도 계산 함수 생성

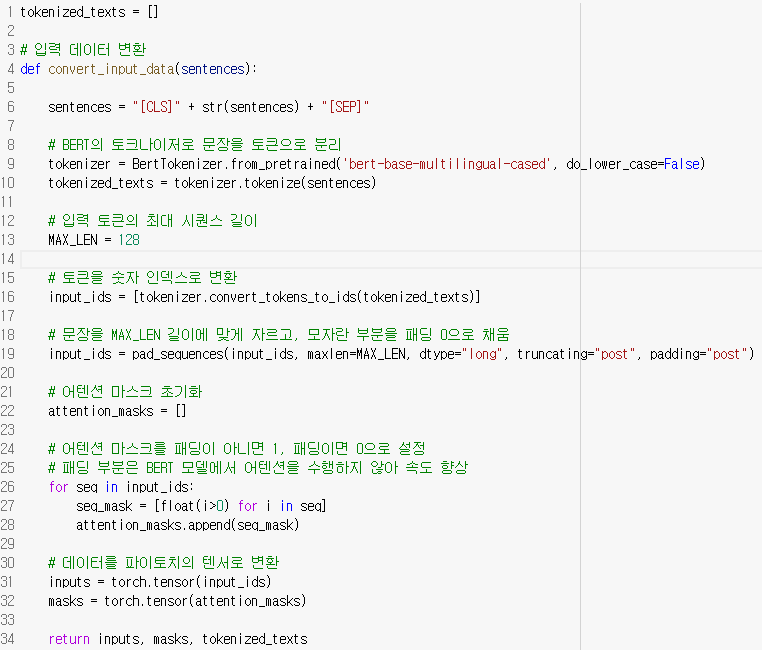


그림 25. 리뷰 데이터를 입력 데이터 형식에 맞게 전처리

[그림 25]은 입력된 문장을 BERT의 입력 데이터 형식에 맞게 전처리 하는 과정이다. [CLS]와 [SEP] 토큰을 사이에 문장을 배치시키고, 문장을 토큰으로 분리하는 토큰화 과정을 거친다. 리뷰 데이터는 시퀀스 길이가 길지 않기 때문에 시퀀스 길이의 제한은 128로 짧게 설정했다. 토큰을 인덱스로 변환시키고 BERT가 더 빠르게 테스트할 수 있도록 어텐션 마스크를 생성한다. 우리는 파이토치를 사용했기 때문에 파이토치에 맞는 텐서로 변환해주면 입력 데이터 전처리가 완료된다



그림 26. 학습된 모델에 입력시켜 테스트

[그림 26]은 모델에 입력 데이터를 입력시켜 모델의 성능을 테스트하는 과정이다. 모델의 출력으로는 모델의 분류에 영향을 미치는 loss값과 BERT Encoder 레이어 마다의 출력 값을 얻을 수 있다.

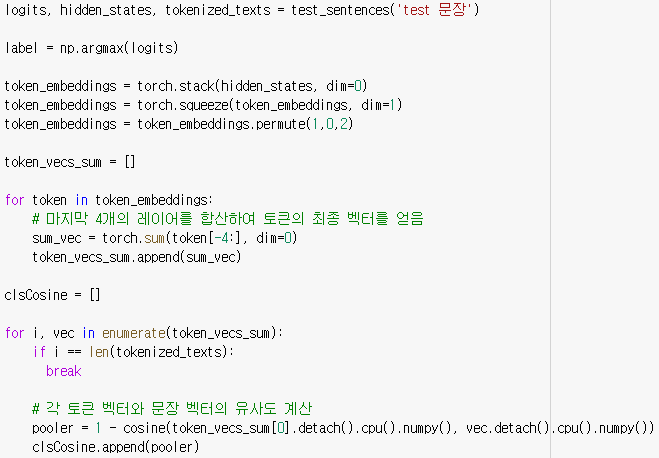


그림 27. 문장 벡터와 토큰 벡터의 코사인 유사도 계산

BERT Encoder 레이어 마다의 출력 값인 hidden\_states를 통해 벡터를 생성한다. 우리는 12개의 레이어 중 마지막 4개의 레이어 출력 값을 합산하여 최종 벡터를 생성했다. 생성된 문장 벡터와 각각의 토큰 벡터의 유사도 점수를 계산한 과정이다.

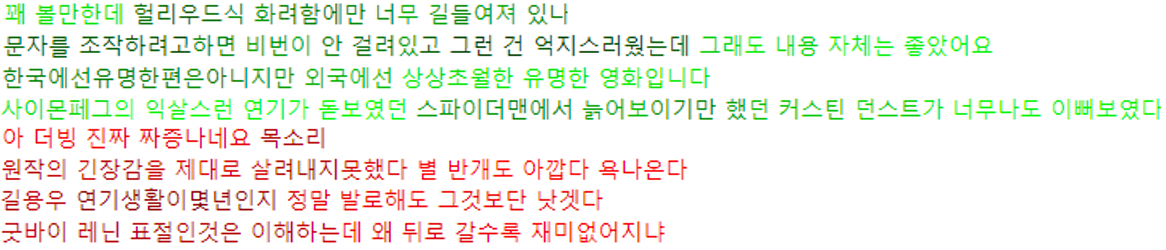


그림 28. 긍정 문장에서 유사도 점수가 높은 단어 강조

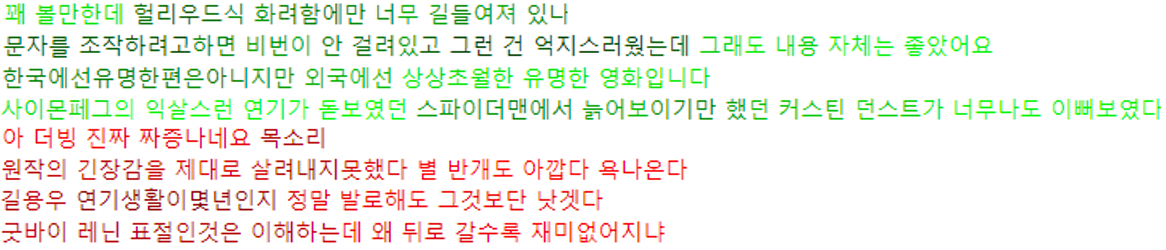


그림 29. 부정 문장에서 유사도 점수가 높은 단어 강조

0에서 1 사이로 계산된 유사도 점수에 명도의 최대값인 255를 곱하여 색깔의 정도를 구분했다. 긍정으로 분류된 문장은 초록색으로, 부정으로 분류된 문장은 빨간색으로 표시하도록 하였으며 유사도가 높은 부분은 명도가 더 높게 칠해진다. 긍정과 부정 각 분류에 맞는 부분이 더 진하게 강조되는 것을 알 수 있었다.

## 제 3 절 설명 가능한 가짜 뉴스 탐지 서비스

1. **가짜뉴스**

지난 2016년 미국 대통령 선거에서 가짜뉴스(FakeNews) 문제가 많은 영향을 미치면서 전 세계적으로 사회적 이슈가 되었다. 가짜뉴스가 사회적 이슈로 대두된 만큼 해당 개념에 대한 정의가 명화하지 않은 상황이다. 가짜 선거뉴스 현황과 각국의 대응 사례에 따르면 각국의 가짜뉴스 정의와 범위가 다른 것으로 알려져있다. 독일의 경우 소셜 네트워크 서비스를 통해 정치 목적, 경제적 이득을 위해 사실이 아닌 글을 유통하는 것을 가짜뉴스의 개념으로 정의하며, 게시물이 법률에 위반하는지의 여부를 판단기준으로 한다. 미국의 정치적 차원에서 가짜뉴스란 일반적으로 정치 선동의 일종으로 정치적, 경제적 이득을 취하려는 목적의 잘못된 정보를 포함하는 모든 형태의 기사나 영상을 의미한다. 가짜뉴스는 미국 대통령 선거와 같은 정치뿐만 아니라, 의료, 엔터테인먼트, 예술, 문화 등 다양한 주제의 뉴스에서도 찾아볼 수 있다. 일반적으로 뉴스가 사실에 근거하지 않거나, 사실의 일부를 왜곡하거나, 논쟁적인 이슈에 대한 편향적인 주장을 가지고 있는지의 여부를 판단기준으로 한다. 영국에서의 가짜뉴스는 대중의 관심을 확보하고 그것으로 광고수익을 창출하기 위해 조작된 뉴스로 정의할 수 있다. 실제뉴스가 편향적인 주장을 가지더라도 가짜뉴스로 구분하지 않으며, 선동 및 거짓을 포함하고 있는지의 여부를 판단기준으로 한다. 이렇듯 각국마다 가짜뉴스의 다양한 개념을 정의하고 있지만 해당 정의도 명확하게 정리된 것이 아니며, 한국도 각국과 마찬가지로 가짜뉴스의 개념 정의가 확실하지 않고 가짜뉴스의 확산에 대한 대응 방안이 체계적으로 마련되지 않은 상황이다. [가짜 선거뉴스 현황과 각국의 대응 사례]

가짜뉴스가 사회적 이슈로 대두되면서 다양한 연구들과 함께 가짜뉴스 탐지 방법이 제시되고 있다. 가짜뉴스 탐지를 위한 기술적 방법으로는 인공지능 기반 가짜뉴스 탐지 방법과 시맨틱 기반 가짜뉴스 탐지 방법 등이 있다. 인공지능 기반 가짜뉴스 탐지 방법은 가짜뉴스에 사용된 언어와 구문을 기계에 학습시켜 판단할 뉴스의 가짜뉴스 확률을 추정하는 방법이다. 시맨틱 기반 가짜뉴스 탐지 방법은 뉴스에 사용된 단어, 어절, 문장, 맥락 등을 분석하여 내용의 사실성을 검증하는 방법이다. 문장의 형태소와 구문, 의미 등을 추출하여 지식 베이스를 구축하게 되고 구축된 지식 베이스에 따라 뉴스의 사실 여부를 판단한다. 가짜뉴스 탐지 방법은 다양한 방법을 혼합하여 발전해나가고 있으며, 인공지능 기반 가짜뉴스 탐지 방법과 다른 방법의 하이브리드 방법이 확산될 것이라 예상된다. [페이크 뉴스 탐지 기술 동향과 시사점]

표 6. 가짜뉴스 탐지 논문 조사

|  |  |  |  |
| --- | --- | --- | --- |
| **논문** | **가짜뉴스 정의** | **데이터셋** | **네트워크** |
| A Deep Ensemble Framework for Fake News Detection and Classification[논문1] | 주제와 맞지 않는 내용이 본문에 존재하며, 주장에 근거가 없는 뉴스 | LIAR | CNN,  Bi-LSTM |
| CSI: A Hybrid Deep Model for Fake News Detection[논문2] | 주제와 맞지 않는 내용이 본문에 존재하며, 신뢰되지 않는 출처로부터 유포되는 뉴스 | Twitter,  Weibo | LSTM |
| DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning [논문3] | 다른 기사와 상반되는 주장이 사람들이 오해할 수 있는 선동적인 어휘로 작성된 뉴스 | Snopes,  PolitiFact,  NewsTrust,  SEMEVAL-2017 TASK 8 | GloVe,  Bi-LSTM |
| Deep Diffusive Neural Network based Fake News Detection from Heterogeneous Social Networks [논문4] | 주제와 맞지 않는 내용이 본문에 존재하며, 신뢰되지 않는 출처로부터 유포되는 뉴스 | PolitiFact | HFLU,  GDU |
| Detecting Incongruity between News Headline and Body Text via a Deep Hierarchical Encoder [논문5] | 기사의 제목과 상관없는 내용이 본문에 작성되어 독자가 원하지 않은 정보를 제공하는 뉴스 | 수집한  한국어 기사를 가공하여 생성한 데이터 | AHDE |
| EMET: EMBEDDINGS FROM MULTILINGUAL-ENCODER TRANSFORMER FOR FAKE NEWS DETECTION [논문6] | 소셜 미디어 네트워크를 통해 유포되는 게시물 중 다른 뉴스의 일반적인 주장들과 상반되는 정보를 포함한 게시물 | Twitter | CNN |
| FNDNet – A deep convolutional neural network for fake news detection [논문7] | 주제와 맞지 않는 내용이 본문에 존재하며, 주장에 근거가 없는 뉴스 | Kaggle | GloVe,  CNN |
| Fake News Detection Using A Deep Neural Network [논문8] | 가짜뉴스에서 많이 사용되는 선동적인 어휘로 작성된 뉴스 | Kaggle | CNN |
| A Deep Learning Approach for Automatic Detection of Fake News [논문9] | 주제와 맞지 않는 내용이 본문에 존재하며, 주장에 근거가 없는 뉴스 | FakeNews AMT,  Celebrity news | ELMo,  GRU,  LSTM |
| Deep Two-path Semi-supervised Learning for Fake News Detection [논문10] | 가짜뉴스에서 많이 사용되는 제한된 단어들을 사용하며, 문법적인 실수가 많은 뉴스 | PHEME | CNN |
| exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT) [논문11] | 기사의 제목과 상관없는 내용이 본문에 작성되어 독자가 원하지 않은 정보를 제공하는 뉴스 | FNC-1 | BERT |

본 연구에서는 인공지능 기반 가짜뉴스 탐지 방법으로 가짜뉴스를 탐지하기 위하여, 먼저 인공지능 기반의 다양한 탐지 기술에서 정의하고 있는 가짜뉴스 개념을 정리하였다. 가짜뉴스 정의에 따라 어떤 인공지능 구조로 구축하였는지 알아보기 위해, 딥러닝 네트워크를 사용하여 탐지한 논문을 중점으로 조사하였다. 인공지능 기반 가짜뉴스 탐지 논문들에서 정의한 가짜뉴스 개념들은 크게 ‘주제와 맞지 않는 내용이 본문에 존재하며, 주장에 근거가 없는 뉴스’, ‘기사의 제목과 상관없는 내용이 본문에 작성되어 독자가 원하지 않은 정보를 제공하는 뉴스’, ‘가짜뉴스에서 많이 사용되는 어휘로 작성된 뉴스’ 등으로 나뉘었다. 사용된 네트워크는 CNN(Convolutional Neural Network), Bi-LSTM(Bidirectional Long Short-Term Memory Network), GRU(Gated Recurrent Units) 등이 있었으며, 비교적 최근 논문들은 GloVe(Global Vectors for Word Representation), ELMo(Embeddings from Language Model), BERT 등 사전 학습된 언어 모델이 함께 사용되었다.

다른 기사와 비교하여 가짜뉴스를 찾아내는 논문은 [논문3]과 [논문 6] 등이 있었는데, [논문3]의 경우 기사와 관련있는 다른 기사들의 내용을 GloVe로 단어 임베딩 후 Bi-LSTM으로 비교하였고, [논문 6] 은 CNN을 사용하여 기사와 관련있는 다른 기사들의 주장을 각각 추출하여 비교하는 방식이 사용되었다. 해당 두 가지의 논문은 사용된 네트워크뿐만 아니라 분류하려는 데이터에서도 차이점을 찾을 수 있었다. [논문3]은 Snopes, PolitiFact와 같은 기사 데이터와 SEMEVAL와 같은 소셜 미디어 데이터도 함께 분류할 수 있도록 테스트하였다. 그러나 기사 데이터 분류의 정확도는 75%, 소셜 미디어 데이터 분류의 정확도는 57%로 좋은 결과를 보이진 못했다. [논문 6]은 [논문3]과 다르게 94%라는 높은 정확도를 보였다. 소셜 미디어 데이터만 비교하자면 [논문3]은 177개의 데이터로 실험하였으며, [논문 6]은 약 19,000개의 데이터로 실험하였다. 이를 보아 두 논문은 네트워크 성능의 차이도 있겠지만 실험에 사용된 데이터의 개수가 결과에 영향을 미친 것으로 판단된다.

기사의 본문을 살펴보고 주제와 맞지 않는 내용을 찾아내는 논문들은 내용의 특징을 찾아내는 CNN이나 시퀀스 데이터 학습에 효과적인 LSTM 및 GRU와 같은 순환 신경망이 많이 사용되었다. 제목과 본문의 내용을 비교하는 논문의 경우 GRU 기반의 AHDE(Attentive Hierarchical Dual Encoder)와 BERT가 사용되었다. AHDE를 사용한 [논문5]는 1,700,000개의 한국어 기사를 직접 수집하여 사용하였는데, 이때 제목을 본문의 내용과 상관없게 수정하여 가짜뉴스를 생성하였다. [논문5]는 본문에서 각 문단마다 독립적으로 제목과 비교하여 본문과 제목 사이의 불일치를 판단하는 모델이다. 본문의 문단이 10개 이하로 이루어진 데이터는 95%에 가까운 정확도를 보였으나, 그 이상을 넘어가는 데이터는 문단이 많아질수록 정확도가 낮아짐을 알 수 있었다. BERT를 사용한 [논문11]은 FNC-1(Fake news challenge stage 1) 데이터셋을 사용하였다. BERT는 문맥을 파악할 수 있는 모델이기 때문에 문장을 비교하는데 많이 사용된다. FNC-1은 어떤 주제가 있을 때 주제에 대한 각 기사들의 입장을 분류한 데이터다. 주제에 대하여 동의/동의 안 함/논의/관련없음으로 분류되어 있다. 데이터가 진짜/가짜로 명확하게 분류된 것이 아니라 관련없음으로 분류된 데이터가 많기에 챌린지에 참여한 기존의 모델들도 정확도가 높지 않았다. [논문11]은 챌린지에 참여한 기존의 모델들과 비교했을 땐 비교적 높은 74%라는 정확도를 보였다. 데이터셋이 다르기 때문에 정확도를 이용한 비교는 불가했지만, 제목과 본문을 비교하는 논문에서는 문맥을 학습하는데 효과적인 네트워크가 많이 사용된다는 것을 알 수 있었다.

1. **영어 가짜뉴스 탐지**

표 7. 가짜뉴스 탐지 모델 테스트 결과

|  |  |
| --- | --- |
| **모델** | **정확도** |
| LSTM[lstm] | 89.92% |
| LSTM+CNN[lstmcnn] | 94.13% |
| BERT[cvp] | 99.13% |

다양한 논문 조사 결과 인공지능 기반 가짜뉴스 탐지 방법에는 CNN, LSTM, 그리고 GloVe, ELMo, BERT와 같은 사전 학습된 언어 모델들이 가장 많이 사용된다는 것을 알 수 있었다. 본 연구에서는 가짜뉴스 탐지를 위하여 사용할 모델을 선택하기 위하여 여러 모델에 동일한 영어 데이터셋을 학습시켜 테스트 정확도를 확인하고자 하였다.



그림 30. Kaggle의 Fake News 데이터셋

데이터셋은 [그림 30]과 같은 Kaggle의 Fake News Dataset[캐글1]의 Train.csv 파일을 사용하였으며, 해당 데이터셋은 진짜뉴스 10,387개, 가짜뉴스 10,413개로 이루어져있다. 데이터셋의 90%는 학습 데이터로, 10%는 테스트 데이터로 사용하였다. 테스트에 사용된 모델은 LSTM 모델, LSTM과 CNN을 사용한 하이브리드 모델, 그리고 사전 학습된 언어 모델들 중 성능이 좋은 것으로 알려져 있는BERT 모델이다. 이 실험을 하기 위한 가짜뉴스 정의는 가짜뉴스에서 많이 사용되는 어휘로 작성된 뉴스이다. 세 가지 오픈소스 모델은 모두 진짜뉴스와 가짜뉴스 각각의 본문에서 많이 사용되는 어휘를 학습시켜 가짜뉴스를 탐지할 수 있도록 하는 모델이다. 세 가지 모델에 동일한 데이터를 학습시키고 테스트한 결과는 [표 7]과 같다. 본 연구는 가장 높은 정확도를 보였던 인공지능과 시맨틱의 하이브리드 모델 BERT를 기반으로 한국어 가짜뉴스를 탐지하고 결과를 설명할 수 있는 모델을 제안하고자 한다.

표 8. 영어 가짜뉴스 탐지 실험 환경

|  |  |
| --- | --- |
| **개발 환경** | Colab |
| **런타임 환경** | GPU |
| **모델** | BertForSequenceClassification |
| **토크나이저** | BertTokenizer |
| **데이터셋** | Kaggle의 Fake News Dataset |
| **데이터 수** | 학습 데이터: 16,640 |
| 테스트 데이터: 4,160 |
| **라벨** | 진짜뉴스: 10,387 |
| 가짜뉴스: 10,413 |
| **테스트 정확도** | 99.13% |

본 연구에서는 영어로 된 가짜뉴스 데이터셋을 학습시키고 테스트하기 위한 모델로 감성 탐지 서비스와 마찬가지로 BertForSequenceClassification 클래스를 사용하였다. 영어 데이터이기 때문에 클래스 안의 BERT 모델은 어느 유형을 선택해도 사전 학습되어있지만, 영어 데이터 테스트 후 실행할 한국어 데이터 테스트를 대비하여 다양한 언어가 지원되는 bert-base-multilingual-cased 모델을 사용하였다. BERT는 사전 학습된 언어 모델로 엄청난 정보를 포함하고 있지만, 다방면으로 자세한 정보를 가지지 않았기에 분류에 맞는 추가 학습이 필요하다. 데이터셋으로는 앞서 설명한 Kaggle의 Fake News Dataset의 Train.csv 파일을 사용하였다. 같은 출처의 Test.csv 파일을 사용하지 않은 이유는 모델의 정확도를 계산하기 위해선 데이터마다 label이 필요했지만 속성으로 label을 가지고 있지 않은 Test.csv 파일은 적합하지 않다고 판단했기 때문이다.

BERT를 이용한 영어 가짜뉴스 탐지 모델은 [표 7]과 같이 99.13%의 정확도를 보였다. 그러나 BERT는 지식 베이스의 모델이기 때문에 학습 데이터와 테스트 데이터의 출처가 달라졌을 때는 정확도가 떨어짐을 알 수 있었다. 같은 모델로 Kaggle Fake News dataset 데이터로 학습시키고 Kaggle Fake and real news Dataset[캐글2] 데이터로 테스트하였을 때 50.46%의 정확도를 보였다. 같은 Kaggle의 데이터이며 두 가지 모두 정치적 뉴스로 이루어진 데이터이지만, Fake News dataset의 경우 경쟁을 위하여 미국 테네시 대학교의 UTK Machine Learning Club에서 제공한 데이터로 약 5개의 Kaggle 데이터에서 수집했다고 하였으나 정확한 출처가 나와있지 않다. Fake and real news Dataset의 경우 진짜뉴스는 신뢰가능한 뉴스 제공 사이트 Reuters.com을 통해, 가짜뉴스는 팩트체크 사이트 Politifact에서 수집하였으며 주로 2016년에서 2017년 사이의 뉴스로 이루어져있다.

해당 실험을 통해 테스트 데이터가 학습된 데이터와 같은 정치 관련 데이터라고 해도 수집 시기나 정치 분야 등에 따라 학습된 지식과 다르다면 정확한 분류가 어렵다는 것을 알 수 있었다. 학습 데이터는 어떤 유형의 테스트 데이터가 들어오더라도 정확하게 분류될 수 있도록 더 넓은 시기와 다양한 분야 데이터로 이루어져야 한다. 또한 데이터를 수집하는 출처를 하나의 사이트로 제한하지 않고 여러 사이트로 넓혀 작성된 기사의 어휘가 편향되지 않아야 한다. 우리는 해당 결과를 통해 지식 베이스 모델인 BERT는 데이터셋의 영향을 크게 받는다는 것을 알 수 있었다.

1. **한국어 가짜뉴스 탐지**

한국어 가짜뉴스 탐지를 위해 첫 번째로 서울대학교 언론정보연구소가 운영하는 SNU 팩트체크에서 뉴스 데이터를 수집했다. SNU 팩트체크에서 다루는 데이터는 주로 정치인이나 공직자 후보의 발언, 뉴스나 소셜 미디어 등을 통해 대중에게 알려지는 정보들이다. 그러므로 SNU 팩트체크 사이트에는 일반적인 온라인 기사뿐만 아니라 기사, 뉴스, 블로그, 카페 등의 다양한 형태의 데이터를 검증한다. 한 뉴스의 사실 판단을 한 결과에서 얻을 수 있는 데이터는 사실 판단을 하고자 한 내용의 요약 정리, 내용의 출처, 검증 내용, 검증의 출처 등이다. [snu]

우리는 지식 베이스로 label을 분류하는 BERT를 테스트하기 위하여 기사, 블로그 등 출처와 상관없이 요약 정리 데이터를 수집하였다. 요약 정리 데이터는 10-30개의 단어로 이루어져있으며 판단하려는 내용을 간단하게 설명한 데이터다. SNU 팩트체크에서의 판단 결과는 '사실', '대체로 사실', '절반의 사실', '대체로 사실 아님', '전혀 사실 아님', '판단 유보' 등 6가지로 분류되는데 진짜뉴스와 가짜뉴스의 정확한 판단을 위해 '사실' 데이터를 진짜뉴스로, '전혀 사실 아님' 데이터를 가짜뉴스로 판단하기로 하였다. 그러나 SNU 팩트체크에서 다루고 있는 데이터는 검증이 필요한 '전혀 사실 아님'에 가까운 데이터가 많았기 때문에 '사실'로 판단되는 데이터의 수가 현저히 적었다. 모델 학습을 위해선 많은 데이터가 필요하기에 우리는 '대체로 사실' 데이터도 진짜뉴스에 포함시켜 수집하였다. 그러나 SNU 팩트체크에서 제공하고 있는 데이터의 수 자체가 많지 않기에 정치, 경제, 코로나 분야에서 진짜뉴스 100개, 가짜뉴스 100개의 데이터를 수집할 수 있었다.

높은 성능의 모델로 학습시키기에 많은 데이터는 아니지만 BERT가 한국어 뉴스도 잘 분류할 수 있는지 테스트하기엔 문제되지 않았다. 총 200개의 데이터 중 90%는 학습 데이터로, 나머지 10%를 테스트 데이터로 사용하였다. 데이터가 적은만큼 Epoch를 여러 번 반복시켰다. 학습 정확도가 98.41%가 되었을 때 학습된 모델로 테스트한 결과 80.95%의 정확도를 얻었다. 데이터가 적더라도 학습된 데이터와 테스트 데이터의 시기와 분야가 비슷하여 지식이 겹친다면 어느정도 분류가 가능하다는 것을 알 수 있었다. 같은 데이터를 사용하여 [표 7]에서 사용된 LSTM + CNN 모델[lstmcnn]로 테스트하였지만 데이터가 적은 탓에 Epoch를 반복시켜도 학습 정확도가 62% 이상으로 오르지 않았다. Epoch를 반복할수록 문맥 학습 능력이 상승하는 BERT와는 다른 결과였다.

우리가 최종적으로 하고자 한 실험은 텍스트로 이루어진 온라인 기사의 사실 검증이다. SNU 팩트체크 사이트에서 요약 정리 데이터가 아닌 판단하고자 하는 내용의 출처를 통해 온라인 기사를 수집하였다. 영어 가짜뉴스 데이터셋의 메타데이터들을 기반으로 다양한 모델에서 다양한 방법으로 유용하게 사용될 수 있도록 한국어 가짜뉴스 데이터셋을 만들고자 하였다. 다음 표는 한국어 가짜뉴스 탐지 서비스를 위한 실험 환경이다.

표 9. 한국어 가짜뉴스 탐지 실험 환경

|  |  |
| --- | --- |
| **개발 환경** | Colab |
| **런타임 환경** | GPU |
| **모델** | BertForSequenceClassification |
| **토크나이저** | BertTokenizer |
| **데이터셋** | SNU 팩트체크 |
| **데이터 수** | 학습 데이터: 162 |
| 테스트 데이터: 18 |
| **라벨** | 진짜뉴스: 90 |
| 가짜뉴스: 90 |
| **테스트 정확도** | 76.47% |

감정을 판단할 수 있는 단어가 있는지 분석되는 영화 리뷰의 감성 탐지 서비스와는 달리 가짜뉴스 탐지 서비스는 본문을 보고 가짜뉴스에 많이 사용되는 어휘가 있는지, 주제와 벗어나는 내용이 있는지, 제목과 반대되는 내용이 있는지 다양한 알고리즘으로 판단될 수 있다. 그러나 어휘를 분석하거나, 주제를 찾고 비교하는 알고리즘을 사용하기엔 우리가 수집한 데이터 수가 부족하다고 판단되었다. 감정 탐지 서비스와 가짜뉴스 요약 정리 데이터의 경우 시퀀스의 길이가 길지 않기 때문에 적은 양의 데이터로도 실험이 가능했지만 기사 데이터는 길이가 다양하며 학습할 내용이 많기 때문에 훨씬 더 많은 데이터가 필요하다.

우리는 적은 양의 데이터로도 테스트할 수 있도록 하기 위해 제목과 본문의 내용을 비교하고, 제목과 본문의 내용이 다르면 가짜뉴스로 분류하기로 하였다. 앞서 실험했던 BERT 기반 실험들은 모두 하나의 시퀀스만 분석하여 label을 분류하였지만, 이번 실험은 제목과 본문 두 가지의 시퀀스를 입력하여 두 내용을 비교하고 label을 분류했다.



그림 31. 수집한 SNU 팩트체크 데이터셋

SNU 팩트체크 사이트에서 가짜뉴스 90개, 진짜뉴스 90개로 이루어진 총 180개의 일반 기사를 데이터셋으로 사용하였다. [그림 31]은 우리가 SNU 팩트체크에서 수집한 데이터셋이다. 분류 label과 제목, 본문, 그리고 카테고리까지 수집하였지만 카테고리는 이번 실험에서 사용되지 않았기에 삭제했다. SNU 팩트체크 사이트에서 다루고 있는 데이터는 일반 기사뿐만 아니라 블로그, 카페, SNS 등 다양한 종류가 있기 때문에 일반 기사만을 많이 수집하는 것이 어려웠다. 본문의 내용만 분석하는 모델을 사용하여 학습을 테스트하면 데이터 양이 적기 때문에 어휘나 주제 등 특성을 학습하기가 어려워 약 33%의 낮은 정확도가 나온다. 그에 비해 제목과 본문 두 가지의 시퀀스를 비교하는 테스트는 더 높은 정확도를 볼 수 있을 것이라 예상했다. 우리는 두 가지 시퀀스를 비교하는 모델이 더 효과적인 결과를 낳을 수 있도록 하기 위하여 데이터셋을 가공했다. 첫 번째로 BERT는 입력 시퀀스의 길이 제한이 있기 때문에 팩트 체크가 되어야 할 부분이 잘릴 수 있다고 판단되어 기사의 본문 중 팩트체크가 필요한 문단 1-2개를 입력으로 할 수 있도록 가공하였다. 두 번째로 label이 가짜뉴스로 분류된 기사에는 본문에서 제목과 반대되는 문장을 추가하여 모델이 학습하기 쉽도록 가공하였다. 같은 데이터셋이지만 본문만 입력으로 하는 경우보다 제목을 추가로 입력시켜 본문과 비교하는 알고리즘이 더 높은 성능을 보였다. 다음은 수집한 가짜뉴스 데이터셋으로 BERT 모델을 학습시키고 테스트하는 과정이다. [가짜뉴스 실습 자료]

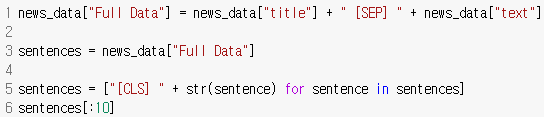


그림 32. 뉴스 데이터를 입력 데이터 형식에 맞게 전처리

리뷰 감정 탐지 서비스와는 다르게 두 시퀀스 데이터를 입력으로 한다. 두 가지의 입력을 [SEP] 태그로 분류해주어야 한다



그림 33. 데이터 토큰화

문장을 토큰으로 분리하는 토큰화 과정이다. 뉴스 데이터는 리뷰 데이터보다 길기 때문에 시퀀스 길이 제한은 BERT의 최대 길이인 512이다. BERT-large 모델을 사용하면 두 배의 길이로 사용 가능하지만 많은 자원을 요구하기에 colab을 통해 사용하기 어렵다. 기본적으론 BERT-base 모델이 사용되며 해당 모델의 길이에 맞게 토큰을 생성했다.

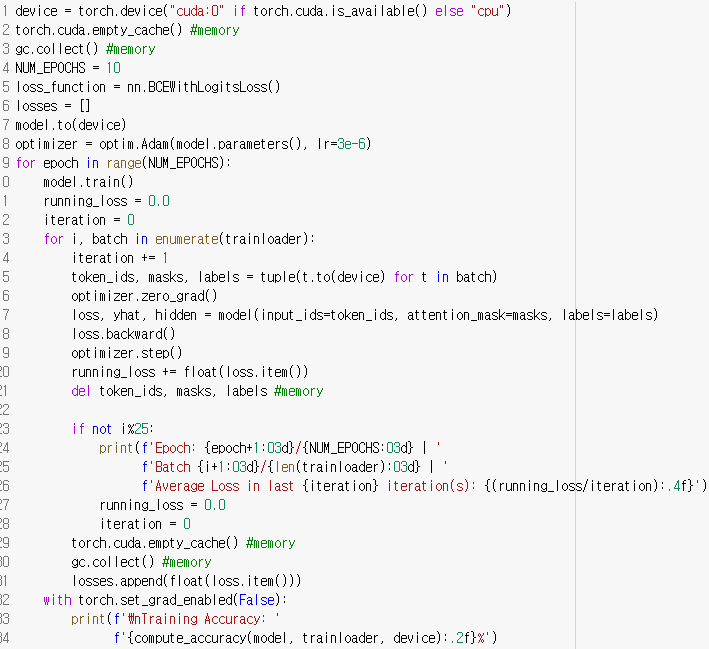


그림 34. 가짜뉴스 탐지 모델 학습

[그림 34]는 수집된 가짜뉴스 데이터셋으로 모델을 학습시키는 과정이다. 데이터의 양이 많지 않기에 epochs 수는 10으로 설정하였다. 11개씩 batch로 입력되었고 Optimizer는 Adam을 사용하였다.



그림 35. 가짜뉴스 탐지 모델 테스트 결과

[그림 35]는 테스트 데이터를 사용하여 모델의 정확도를 계산한 결과이다. 180개의 데이터 중 10%를 테스트 데이터로 사용하였으며 중복되는 데이터는 없도록 하였다. 162개의 데이터로 학습시키고 나머지로 테스트한 결과 76.47%의 정확도를 보였다.

그러나 해당 실험을 하기 위해선 더 많은 데이터가 필요하고, 가짜뉴스 label의 경우 제목과 본문의 내용이 불일치하는 부분이 존재하여야 했다. 데이터의 양을 늘려 본문에서 제목과 다른 부분을 찾아내야 한다는 것을 명확하게 학습시킨다면 다른 부분을 시각적으로 표현하는데 정확도가 높아질 것이라 판단됐다. 또한 고정된 데이터셋을 사용하는 것이 아니라, 판단하려는 기사와 관련된 증거 기사를 웹크롤링을 통해 가져온 후 내용을 비교하여 가짜뉴스를 탐지한다면 탐지 정확도가 오를뿐만 아니라 필요한 학습 데이터 양을 훨씬 줄일 수 있을 거라고 생각된다.

# 제 4장 결론

편향성 문제를 야기하는 원인은 하나가 아니기 때문에 다양한 접근 방법이 필요하다.

인공지능이 편견 또는 편향성을 가질 수 있다는 것을 인지하고 편향성 문제 발생시 적절하게 처리할 수 있는 방법을 강구하는 것이 필요한다.

----------

지난 2016년 미국 대통령 선거에서 가짜뉴스(FakeNews) 문제가 많은 영향을 미치면서 전 세계적으로 사회적 이슈가 되었다. 가짜뉴스가 사회적 이슈로 대두된 만큼 해당 개념에 대한 정의가 필요하지만 이는 아직 명확하지 않다. 가짜뉴스를 탐지하기 위한 다양한 방법들이 제시되고 있으며, 탐지 방법을 제시하기 전에 가짜뉴스를 정의하고 그에 맞는 탐지 방법을 구축하여야 한다. 우리는 가짜뉴스의 개념을 ‘제목과 본문의 내용이 일치하지 않는 뉴스’로 정의하고, 사전 학습되어 문맥을 파악하는 능력이 높으며 분류 레이어를 추가시켜 분류 모델까지 학습시킬 수 있는 BERT를 사용하여 가짜뉴스 탐지를 시도했다. 가짜뉴스를 탐지하는 실험을 하기 전에 BERT의 성능을 테스트하기 위하여 네이버 영화 리뷰 데이터셋을 사용하여 감정 탐지 서비스를 구축했다. 150,000개의 학습 데이터로 학습시킨 모델을 50,000개의 데이터로 테스트한 결과 87%의 정확도를 볼 수 있었다. 그리고 분류된 결과의 이유를 설명하기 위하여 리뷰 문장 중 감정을 나타내는 단어에 색깔이 표시되도록 구현했다. 문장 벡터와 단어 벡터의 유사도를 측정하여 유사도가 높은 단어가 문장에 영향을 크게 미쳤다고 판단하고 통계를 내보니 조사와 부사를 제외하고 감정을 나타내는 명사가 많이 강조된다는 것을 알 수 있었다. 우리는 BERT의 감정 분류 서비스를 통해 성능을 테스트하고 가짜뉴스를 탐지하려 했다. 가짜뉴스 관련 다양한 논문을 분석한 결과 한국어 데이터를 사용한 논문은 [논문 5]만 존재했으며, 이는 수십만개의 한국어 기사를 수집하고 가짜뉴스 label의 기사는 제목을 랜덤으로 변경하였다. 우리는 SNU 팩트체크를 통해 팩트체크가 완료된 기사를 수집하였으나, SNU 팩트체크는 블로그, 기사, 카페 등 다양한 종류의 데이터에 대해서 팩트체크하지만 기사에 대한 데이터를 많이 제공하고 있지 않기 때문에 많은 데이터를 수집하기 어려웠다. 기존의 목표는 본문을 입력시키고 가짜뉴스에서 많이 사용되는 어휘와 단어를 학습시켜 가짜뉴스를 탐지하려 하였으나 데이터의 수가 많지 않아 어휘를 학습시키기 어려웠다. 우리는 가짜뉴스의 개념을 ‘제목과 다른 본문’으로 정의하고 BERT에 기사의 제목과 본문을 입력하여 제목과 본문을 비교하였다. 적은 데이터였지만 제목과 본문을 비교하는 방법으로 테스트 결과 76%의 정확도를 볼 수 있었으며, 데이터의 양이 많아진다면 더 높은 성능뿐만 아니라 감정 탐지 서비스처럼 문제되는 부분이 강조될 수 있을 것이라 판단된다. 고정된 데이터셋만을 사용하지 않고 웹크롤링을 통해 실시간으로 수집되는 증거 기사를 기반으로 가짜뉴스를 탐지한다면 더 좋은 성능을 보일 수 있을 것이다.

**참고문헌**

[1]

[한국어 임베딩] 이기창 (2019), 한국어 임베딩, 에이콘출판사

[BERT 논문 정리] [https://BERT\_Paper](https://mino-park7.github.io/nlp/2018/12/12/bert-%EB%85%BC%EB%AC%B8%EC%A0%95%EB%A6%AC/?fbclid=IwAR3S-8iLWEVG6FGUVxoYdwQyA-zG0GpOUzVEsFBd0ARFg4eFXqCyGLznu7w)

[토크나이저 설명] https://huffon.github.io/2020/07/05/tokenizers/

[huggingface]https://huggingface.co/transformers/model\_doc/bert.html#tfbertmodel

[BERT paper] Jacob Devlin, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2 (2019)

[pytorch BERT] <https://pypi.org/project/pytorch-pretrained-bert/>

[영화 리뷰 분류] <https://cyc1am3n.github.io/2018/11/10/classifying_korean_movie_review.html>

[nsmc] <https://github.com/e9t/nsmc/>

[리뷰 단어 통계, 실험 자료] https://drive.google.com/drive/u/1/folders/1\_VUVpnYyLYGB5fmjCA95E461HHkuicy0

[가짜 선거뉴스 현황과 각국의 대응 사례] 선거연수원(제도연구부), 가짜 선거뉴스 현황과 각국의 대응 사례, 해외연구관 지정과제 제2017-2차 (2017)

[페이크 뉴스 탐지 기술 동향과 시사점] 윤영석, 페이크 뉴스 탐지 기술 동향과 시사점, 정보통신기술진흥센터 (2017)

[논문1] Arjun Roy, A Deep Ensemble Framework for Fake News Detection and Classification, arXiv:1811.04670v1 (2018)

[논문2] Natali Ruchansky, CSI: A Hybrid Deep Model for Fake News Detection, arXiv:1703.06959v4 (2017)

[논문3] Kashyap Popat, DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning, arXiv:1809.06416v1 (2018)

[논문4] Jiawei Zhang, Deep Diffusive Neural Network based Fake News Detection from Heterogeneous Social Networks, 2019 IEEE International Conference on Big Data (Big Data) (2019)

[논문5] Seunghyun Yoon, Detecting Incongruity between News Headline and Body Text via a Deep Hierarchical Encoder, The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19) (2019)

[논문6] Stephane Schwarz, EMET: EMBEDDINGS FROM MULTILINGUAL-ENCODER TRANSFORMER FOR FAKE NEWS DETECTION, ICASSP (2020)

[논문7] Rohit Kumar Kaliyar, FNDNet – A deep convolutional neural network for fake news detection, Cognitive Systems Research 61, p.32-44 (2020)

[논문8] Rohit Kumar Kaliyar, Fake News Detection Using A Deep Neural Network, 2018 4th International Conference on Computing Communication and Automation (ICCCA) (2018)

[논문9] Tanik Saikh, A Deep Learning Approach for Automatic Detection of Fake News, arXiv:2005.04938v1 (2020)

[논문10] Xishuang Dong, Deep Two-path Semi-supervised Learning for Fake News Detection, arXiv:1906.05659v1 (2019)

[논문11] Heejung Jwa, exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT), Applied Sciences, Volume 9, Issue 19, 10.3390/app9194062 (2019)

[lstm] <https://towardsdatascience.com/detecting-fake-news-with-deep-learning-7505874d6ac5>

[lstmcnn] https://towardsdatascience.com/how-to-build-a-recurrent-neural-network-to-detect-fake-news-35953c19cf0b

[cvp] https://www.cvpcorp.com/detecting-fake-news-with-a-bert-model/

[캐글1] https://www.kaggle.com/c/fake-news/

[캐글2] https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset#Fake.csv

[snu] <https://factcheck.snu.ac.kr/>

[가짜뉴스 실습 자료] https://drive.google.com/drive/u/1/folders/1oDn3YYwz0Aykgtl5RoA-VsT9VqxGbSP8

1. 웹콘텐츠 보호와 관련한 활동 중 비디오 콘텐츠 보호는 EME(Encrypted Media Extenstions) 방법을 통해 해결하였으나 웹 기반 게임의 3D 자산(3D assets) 보호 방안은 아직 해결되지 않음 [↑](#footnote-ref-1)