**인공지능과 보안 프로젝트 보고서**

**‘FastText’ 코드 구현 및 개선**

**1조**

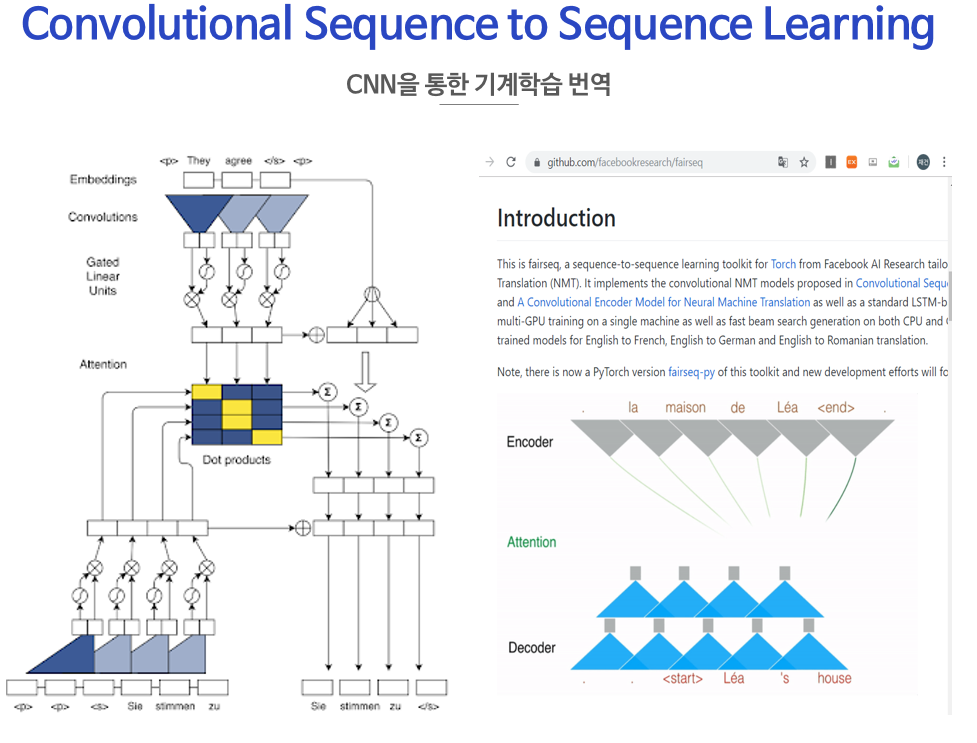
**김동길 김혁민 이우성 최재건**

**2019. 10. 30. - 2019. 12. 11.**

**I. 프로젝트 선정**

**I – 1) Fairseq**

ConvS2S모델을 이용한 기계번역, 음성인식, text 요약 등 분야에 활용되는 언어학습 모델



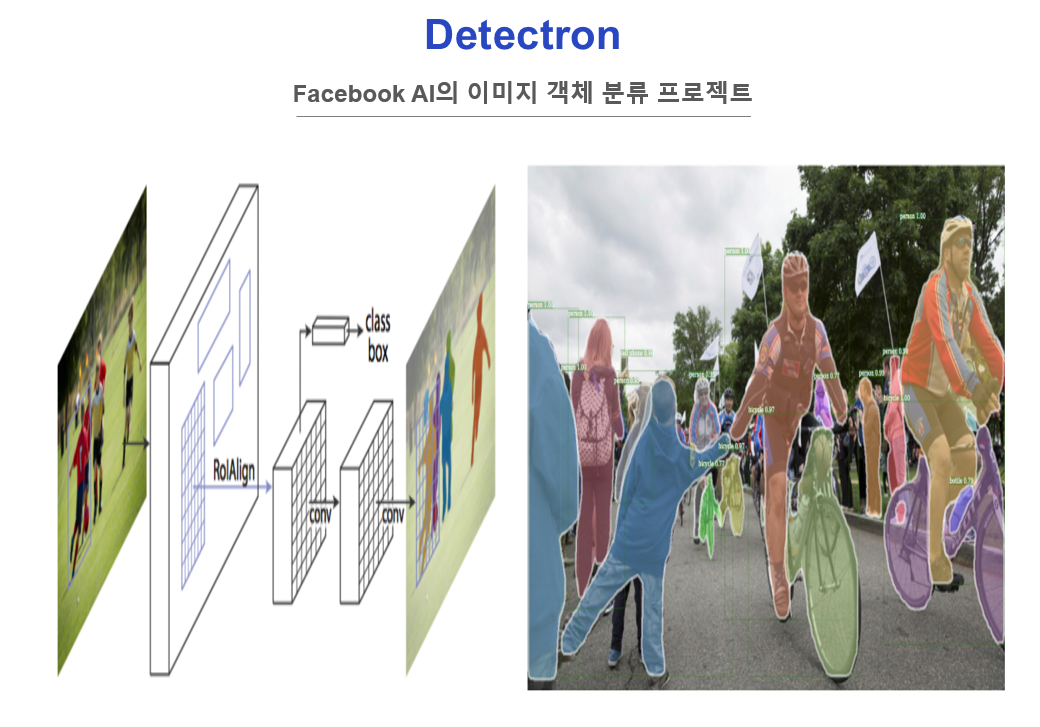
**I – 2) Creative Adversarial Networks**

GAN모델을 이용한 Facebook의 이미지 분석 프로젝트



**I – 3) Detectron**

Mask R-CNN 모델을 이용한 이미지 객체 분류 프로젝트



**I – 4) FastText**

CBOW, Skip-Gram 모델을 이용한 Facebook AI팀의 자연어 처리 프로젝트

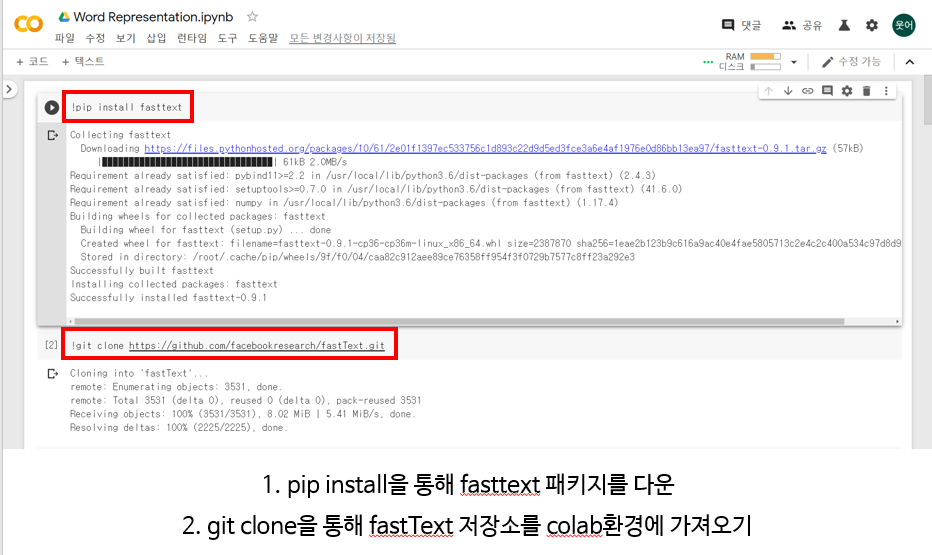


**II. 프로젝트 코드 구현 및 개선**

자연어 처리 시장 2025년 약 432억 달러 규모 전망으로 앞으로 시장 발전 가능성이 크고, 정서 분석, 맞춤법 검사, 인공지능 (Artificial Intelligence, AI) 음성인식 스피커 등 활용 분야가 무궁무진하다. 실제로 페이스북에서도 FastText를 해시태그 자동 완성 기능 강화, 사용자의 정서 분석 등에 활용하고 있다. 또한 FastText의 목표는 프로그래밍 배경 지식이 얕은 사람들도 더 쉽게 텍스트 분류(단어 집합을 범주로 지정하는 과정)와 텍스트 표현(구조화되지 않은 텍스트를 연산용 숫자로 변환하는 과정)을 할 수 있도록 지원하는 것으로 프로젝트를 진행하며 실제로 코드를 구현해보고 원리를 이해하기에 적합하다고 생각하여 FastText를 선정하게 되었다.

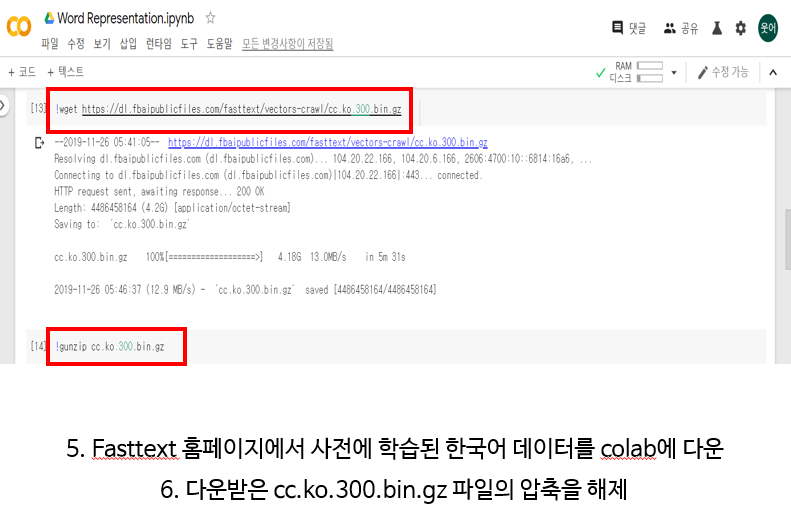
**II – 1) 코드 구현**

**실습환경 구성**



Pip install, Git clone을 통해 fasttext 패키지 다운 및 colab 연동

**Word Representation**



1. Fasttext 홈페이지에서 사전에 학습된 한국어 데이터를 colab에 다운

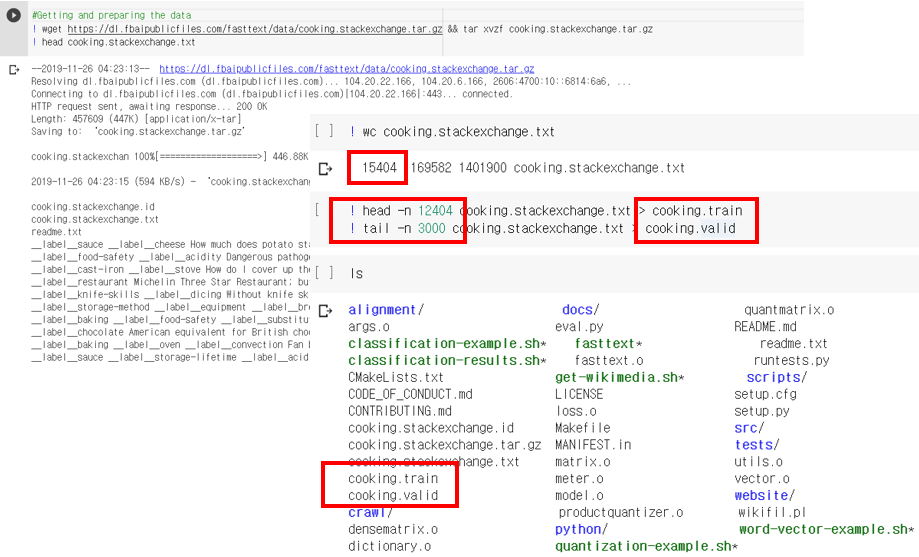
2. 다운받은 cc.ko.300.bin.gz파일의 압축을 해제

**유사도 측정**

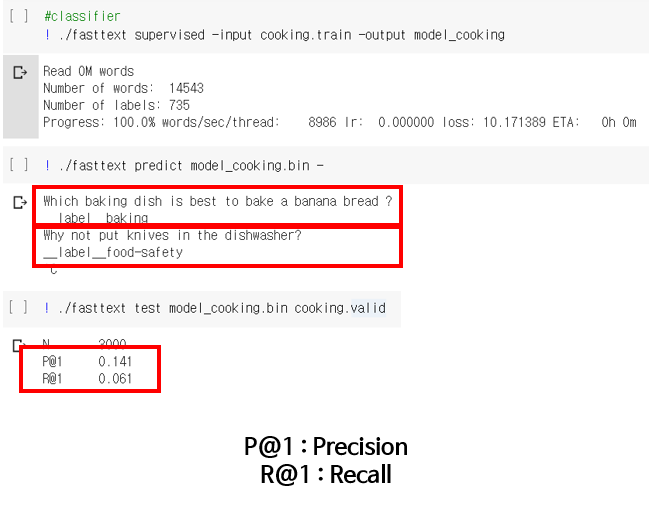


Query word에 입력한 단어와 유사하거나 연관될수록 수치가 높게 나타남

**Text Classification**

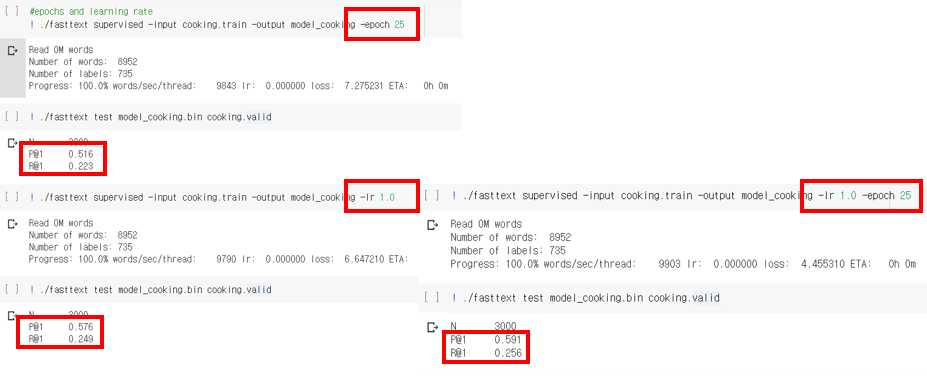


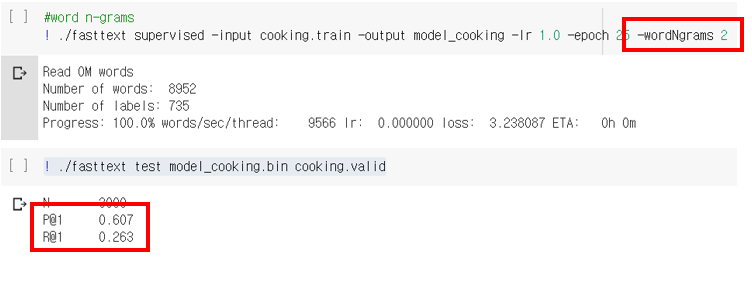
1. 데이터셋을 다운받고 training set과 testing set으로 나눔

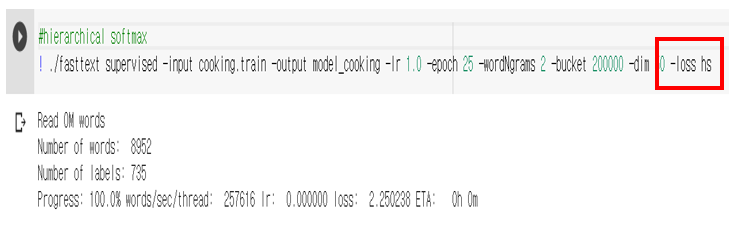


2. Query sentence를 입력하면 label에 따라 분류

3. precision과 recall을 출력



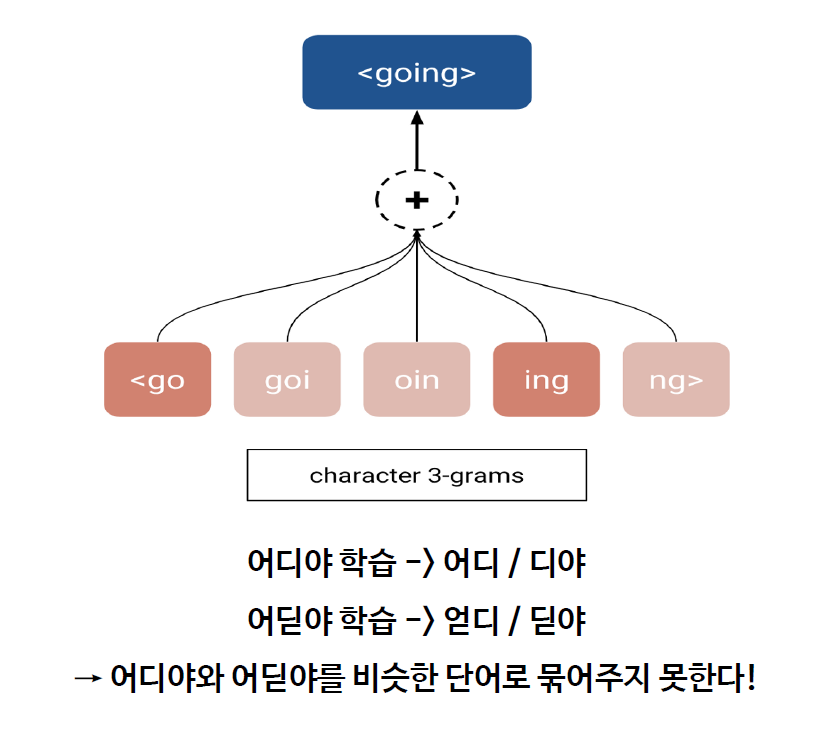




4. 옵션 설정(epoch, learning rate, wordNgram, loss(손실함수))

**II – 2) 문제점 및 개선점**

한국어는 그 특성 상 구문적 복잡성을 내재하고 있을 뿐만 아니라, 한국어 데이터에는 수많은 의도적 오탈자, 띄어 쓰기 오류 등 다른 언어에서는 자주 발생하지 않는 문제들이 존재하기 때문에 자연어처리 학습 과정에서 어려움이 있다.

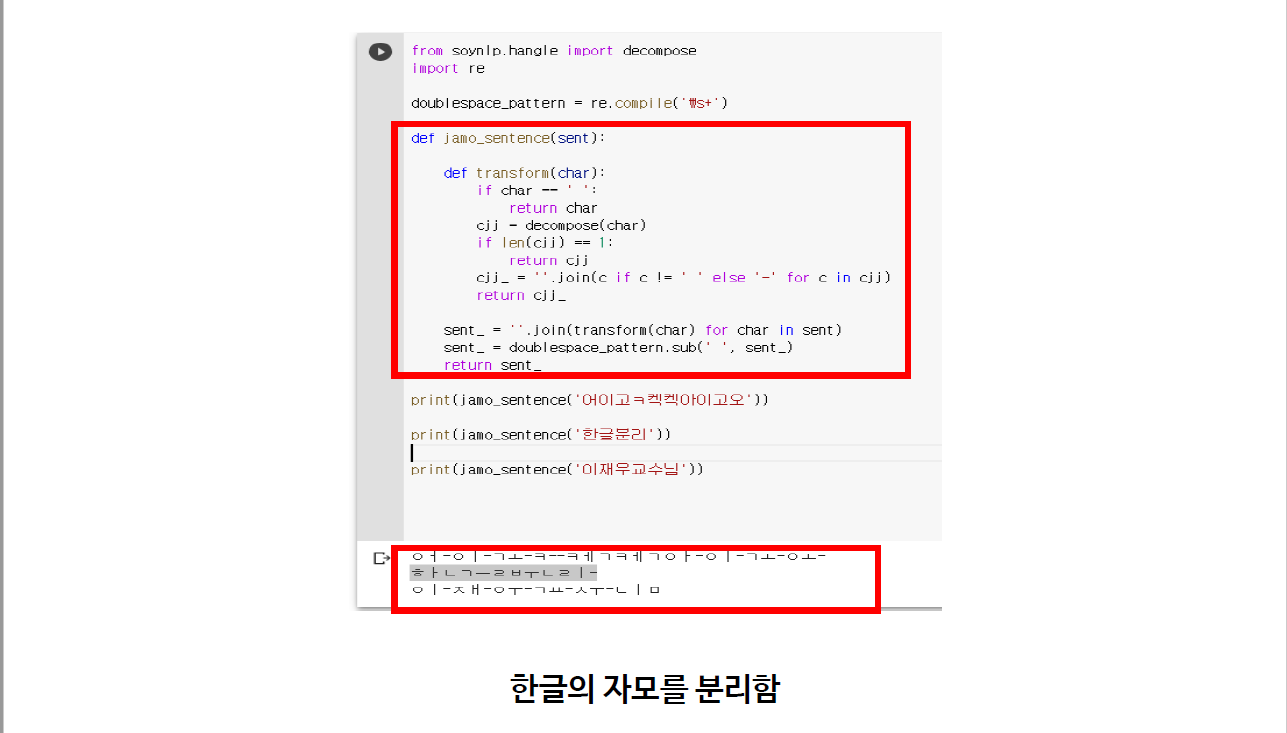


데이터 전처리를 통한 학습 개선

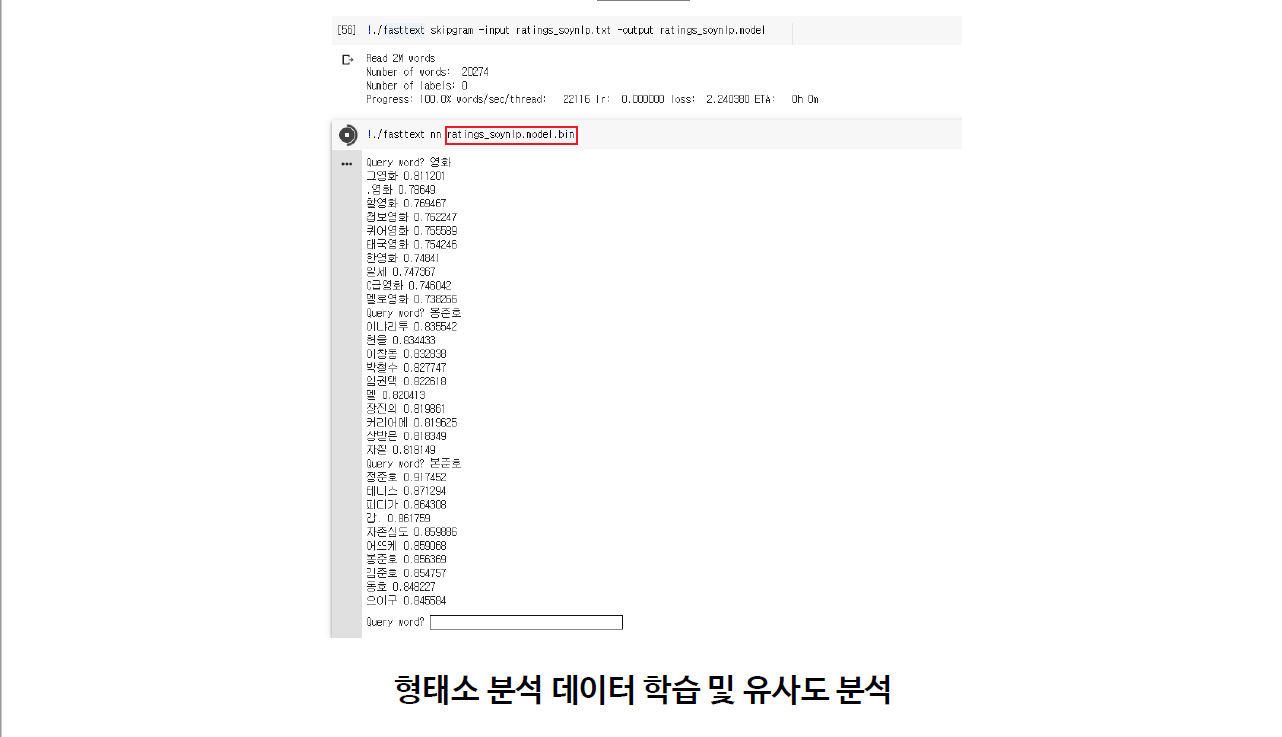
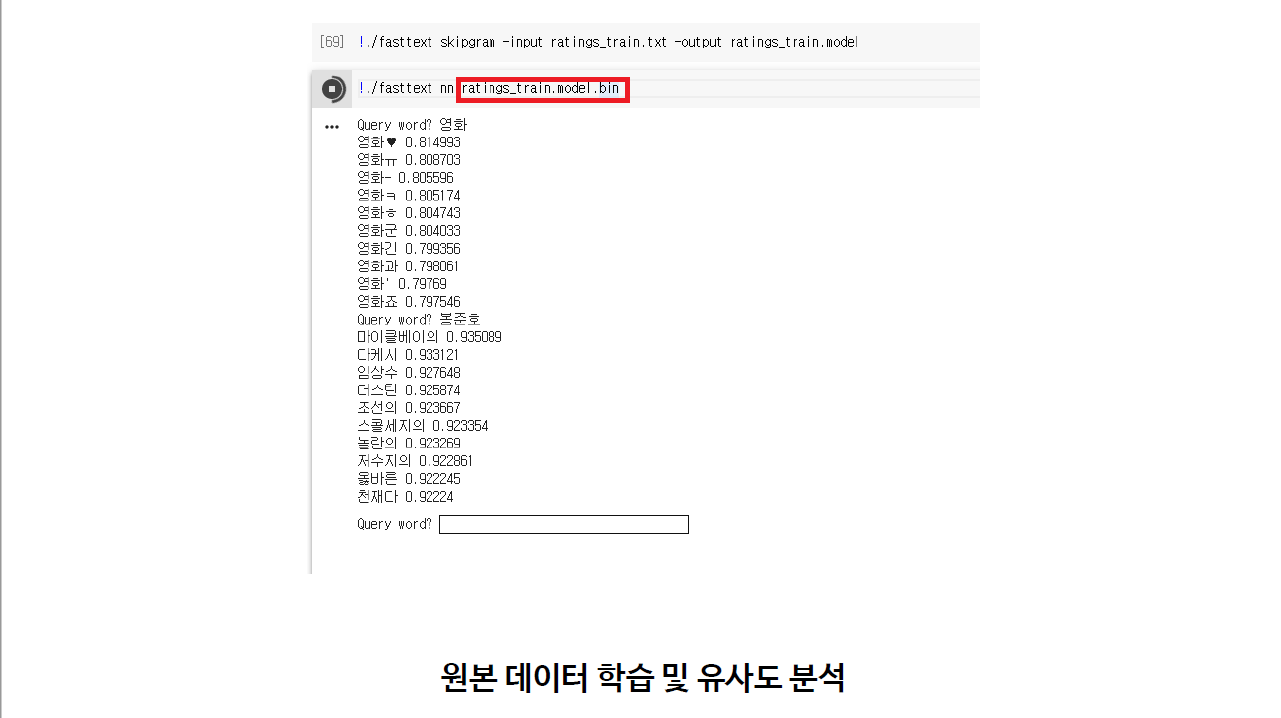
- 형태소 분석: 어휘에서 의미적인 부분이 아닌 것을 제거, 분리시켜 학습 모델에 적용

- 자모 분리: 어휘를 자음과 모음 수준으로 쪼개어 학습 모델에 적용

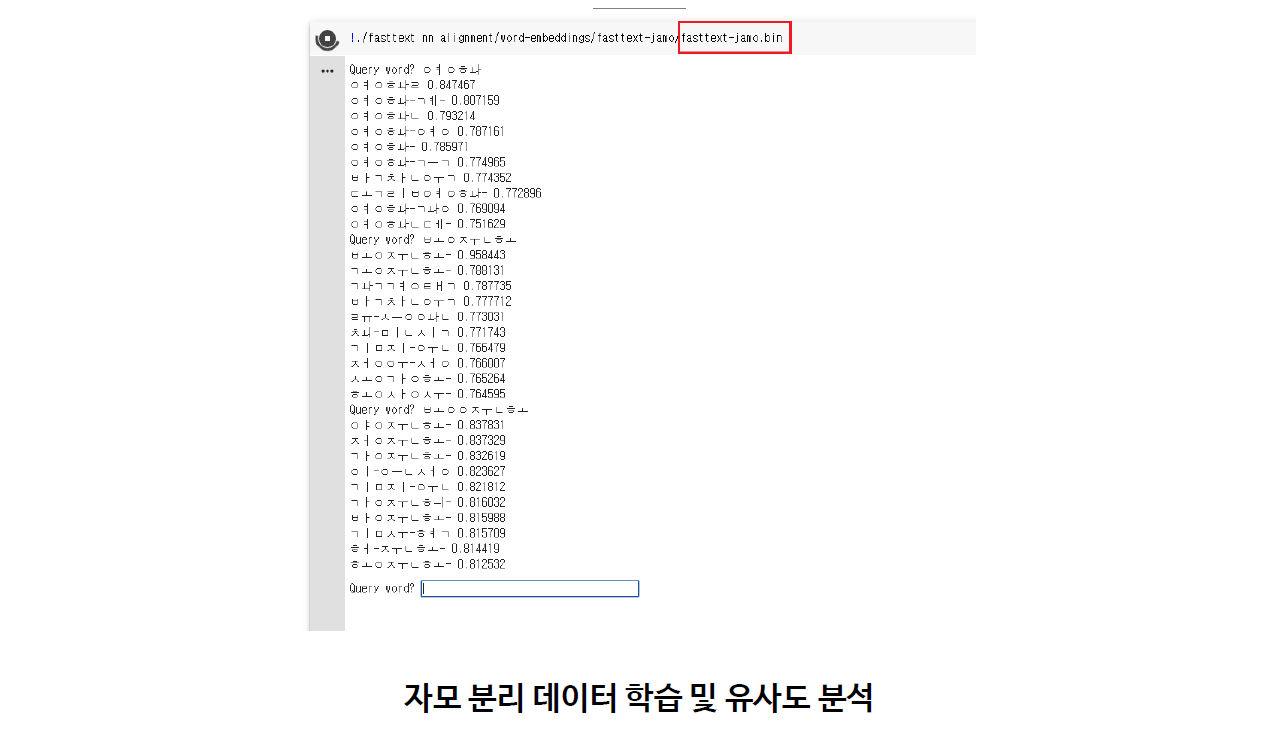
-> 한국어 언어 학습 과정에서 발생할 수 있는 문제점 해결 가능



데이터의 자모음을 분리하는 코드 생성



전처리 과정을 거치지 않은 원본 데이터와 전처리 과정을 거친 데이터의 비교



형태소 및 자모 분리 과정을 거친 학습 데이터를 사용하여 분석을 실행한 결과

원본, 형태소분석, 자모 분리 데이터를 학습하는 과정에서 데이터의 텍스트가 영화 리뷰에 한정되고 학습에 필요한 충분한 양이 아니었다는 한계점이 존재했다. 때문에 자모 분리 데이터 학습결과와 원본 데이터 학습 결과에 예상한만큼의 차이를 발견하지는 못하여 아쉬운 점이 있었다.

**III. 프로젝트 후기**

FastText 프로젝트를 진행하며 텍스트 분류와 Word Representation 기능을 활용하는 사용 코드를 실제로 사용해보고, 효과적인 한국어 학습을 위해서 영자 분리와 다르게 형태소분석과 자모 분리를 따로 구현해 보았다. 이를 진행하는 과정에서 프로젝트 진행 환경의 한계로 인해 모델 학습에 사용한 훈련데이터의 질과 양이 떨어졌기 때문에 최종 결과가 예상했던 대로 도출되지 않았고 인공지능 학습 모델을 구현할 시에는 학습 데이터의 충분한 양과 사전 데이터 전처리의 중요성을 느끼게 되었다.

**IV. 참고 문헌**

1. 밑바닥부터 시작하는 딥러닝(사이토 고키 저)

2. https://inspiringpeople.github.io/data%20analysis/word\_embedding

3. https://github.com/xiamx/fastText

4. https://fasttext.cc/docs/en/supervised-tutorial.html

5. https://nodoudt.tistory.com/24

6. https://github.com/facebookresearch/fastText/tree/master/python