안녕하세요 저희는 아이펠 4기 자연어 처리반 팀 Kim&lim입니다. 저는 발표를 맡은 김다인입니다 /

발표는 다음과 같은 순서로 진행될 예정입니다 /

먼저 발표에 들어가기 전 저희 팀원들을 간단하게 소개해드리겠습니다. 저는 팀장과 발표, 약간의 코드를 담당했고 메인 코더로는 경훈님과 지혜님 두 분이 활약해 주셨습니다. 데이터 정리와 증강에는 경훈님, 현희님 지혜님, 늦게 합류해주신 용석님도 함께 도움 주셨습니다. /

이제 본격적으로 발표 진행해보겠습니다 /

첫 번째는 데이터입니다. 저희가 프로젝트를 진행한 데이터셋은 DKTC, 즉 한국어로 구성된 과격표현 이었습니다. 협박, 갈취, 직장 내 괴롭힘, 기타 괴롭힘은 4개의 클래스로 구분되어 있으며 각 데이터의 분포는 오른쪽 표와 같습니다. /

이 표와 그래프는 데이터 길이 분포를 나타냅니다. 왼쪽 표에서 알 수 있듯 가장 많은 대화가 100~300 구간에 분포해있었습니다. 오른쪽 그래프는 각각의 클래스에서 어떤 구간에 데이터가 분포해있는지를 시각화한 그래프입니다. /

저희가 채택한 데이터 증강의 방식은 다음과 같습니다. /

먼저 EDA(탐색적 데이터 분석)과정에서는 한국어 오픈 워드넷 사전 기반의 반자동 구축 시스템인 KWN을 사용했습니다. 또한 한국어 데이터셋을 안전하게 데이터 증강을 하기 위해 Random Swap​과 Random Deletion​의 두 가지 방식만 채택했습니다. /

두번째로 사용한 방법은 Back Translation 입니다. Back Translation이란 번역된 텍스트의 의미를 확인하기 위해 이를 원어로 다시 변환하는 과정입니다. 이 과정은 번역 검증이라고도 부르며 텍스트의 정확성을 확인하는 요인으로 작동합니다. 보통 번역기를 개발할 경우, 한 쌍의 번역 모델이 자연스럽게 나오게 됩니다. 왜냐하면 우리는 parallel corpus를 통해 번역기를 개발하므로, 두 방향의 번역기를 학습할 수 있기 때문입니다. 이때 Back-Translation이라는 이름에서 볼 수 있듯이, BT는 반대쪽 모델을 타깃 모델을 개선하는데 활용합니다. /

마지막 MLM은 마스킹된 언어모델로, 입력으로 사용하는 문장의 토큰중 15%의 확률로 선택된 토큰을 [MASK] 토큰으로 변환시키고, 언어모델을 통해 변환되기전 [MASK] 토큰을 예측하는 언어모델입니다. MLM은 mlm\_probability에 따라 [MASK]로 변환할 토큰을 선택합니다. 선택된 토큰중 80%는 [MASK]토큰으로 변환하고 10%는 랜덤 토큰으로 변경하며, 10%는 변환하지 않는 방식입니다. /

두번째는 실험-모델입니다. /

첫 번째 모델은 가장 기본적인 CNN입니다. CNN 모델은 합성곱 신경망으로 2 개의 1차원 컨볼루션 레이어, 1차원 글로벌 맥스 풀링 등을 특징으로 갖는 모델입니다. /

이 화면의 그래프는 저희가 구한 CNN모델의 정확도와 로스입니다. 컨퓨전 매트릭스를 간단하게 분석해보면 가장 정확도가 떨어지는 클래스는 갈취이고, 직장 내 괴롭힘 클래스를 가장 정확하게 맞췄습니다. /

두 번째 모델은 BiLSTM계층입니다. BiLSTM계층은 Bidirectional LSTM의 약자로 시계열 또는 시퀀스 데이터의 시간 스텝 간의 양방향 장기 종속성을 학습하는 RNN 계층입니다. 양방향 LSTM이라고도 부릅니다. /

양방향 LSTM의 경우에도 가장 정확도가 높았던 클래스는 기타 괴롭힘이고, 가장 정확도가 떨어지는 클래스는 갈취였습니다. /

세 번째 모델은 BiLSTM과 LSTM​을 함께 사용한 모델입니다. /

이 모델 또한 직장 내 괴롭힘 클래스를 가장 정확하게 맞췄고, 가장 정확도가 떨어지는 클래스는 갈취였습니다.

마지막 네 번째 모델은 CNN, BiLSTM과 LSTM​을 모두 사용한 모델입니다. /

이 모델은 갈취 클래스를 가장 정확하게 맞췄고, 가장 정확도가 떨어지는 클래스는 기타 괴롭힘이였습니다. /

앞에서 설명드린 모델들을 모아보면 다음 표와 같습니다. 가장 정확도가 높았던 모델은 CNN이었으며 가장 정확도가 떨어지는 모델은 CNN, BiLSTM과 LSTM​을 모두 사용한 모델입니다. /

다음은 저희가 진행한 데이터 증강의 방법입니다. 이 화면은 한가지 데이터 증강 방법을 적용했을 때 어떤 결과가 나타났는지를 정리한 표입니다. 가장 정확도가 높았던 방법은 25%의 Random Swap​ 방법이었으며 가장 정확도가 떨어지는 방법은 75%의 일본어 Back Translation​ 방법이었습니다. /

두가지 이상의 방법으로 데이터를 증강했을 때는 Random Swap​과 Random Deletion​을 병행했을때가 가장 정확도가 높았고, Random Swap​과 Random Deletion​에 일본어 Back Translation​ 방법을 더했을 때 정확도가 가장 낮게 나왔습니다. /

메인의 마지막 Transfer​ Learning​ 파트입니다. /

저희가 사용한 전이학습은 Bert입니다. BERT는 학습을 위해 기존 transformer의 input 구조를 사용하면서도 추가로 변형하여 사용합니다. 특히 self-attiotion layer를 여러 개 사용하여 문장에 포함되어 있는 token 사이의 의미 관계를 잘 추출할 수 있습니다. Tokenization은 WorldPiece 방법을 사용하고 있습니다. Bert 모델은 마지막 블록의 [CLS] 토큰을 사용해서 문장의 피쳐를 뽑고, 피쳐 데이터를 활용해서 분류기를 학습시키는 방식입니다. 또한 decoder를 사용하지 않고, 두 가지 대표적인 학습 방법으로 encoder를 학습시킨 후에 특정 task의 fine-tuning을 활용하여 결과물을 얻는 방법으로 사용됩니다. 이 두가지 학습방법 중 하나가 저희가 사용한 cls embedding입니다. /

[CLS] 토큰은 문장의 시작 부분에, [SEP] 토큰은 문장의 끝에 추가된다. BERT는 문맥을 이해하기 위해 문장의 각 단어(토큰)을 다른 모든 단어와 연결시켜 이해하는 계산 과정을 거친다. 이 과정에서 [CLS] 토큰은 다른 모든 토큰의 집계 표현을 보유하고 있으므로 문장 전체에 대한 표현을 담고 있다.​ [SEP]을 사용하여 첫 번째 문장과 두 번째 문장을 구별​하고 여기에 segment Embedding을 더해서 앞뒤 문장을 더욱 쉽게 구별할 수 있도록 도와줍니다. /

저희가 사용한 모델들의 정확도를 정리해봤습니다. Beomi/kcbert-base 모델이 가장 높은 정확도를, tunib/electra-ko-base와 klue/Roberta-large 모델이 가장 낮은 정확도를 보여주었습니다. /

이 결과는 저희가 본격적으로 프로젝트를 시작하기 전에 머신러닝을 사용해 돌려보면 어떤 결과가 나올까 궁금해서 시도해본 결과입니다. 정확도 0.8266이 나왔고 이 결과를 토대로 이것보다는 정확도가 높게 만들어보자! 라는 목표가 생겼습니다. /

첫 시도는 Bert-base로 데이터 증강 없이 10에포크를 돌려본 결과입니다. 정확도는 0.8759가 나왔습니다. /

두 번째 시도는 Bert-base로 데이터 증강 후 3에포크를 돌려본 결과입니다. 정확도는 0.8962가 나왔습니다. 확실히 데이터 증강 전후의 차이가 보였습니다. 반올림해서 0.9라는 정확도에 저희는 조금만 더 하면 더 좋은 결과를 얻을 수 있을 것 같아 고민했고 마지막 결과를 얻었습니다. /

마지막 시도로 Bert\_base, 데이터 증강 후 10에포크를 돌려본 결과입니다. 정확도 0.9443이라는 수치를 얻어내면서 프로젝트를 마무리했습니다. / 이 사진은 임의의 문장을 입력하여 클래스를 잘 구분하는 것을 확인한 사진입니다. /

발표의 마지막입니다. 이 프로젝트를 진행하면서 많은 일들이 있었습니다. 가장 먼저 LMS 에러가 꽤 많이 발생했습니다. 코드를 실행하다가 멈춰버린다던가, 갑자기 종료되면서 작업 내용이 날아가버린다던가 하는 이슈들이 조금 있었습니다. 그 외에도 처음 사용해보는 Bert같은 것이나 hugging face에서 pre-trained 된 모델들을 활용하는 것 등등 막히는 부분이 많았지만 모두 서로 돕고 함께 해결책을 생각해주면서 완성한 프로젝트인 것 같습니다. 팀원들에게 부족한 팀장 따라오느라 고생했다고 꼭 말해주고싶습니다. /

다음은 프로젝트를 진행하면서 참고한 자료들의 출처입니다. /

이상으로 발표 마치겠습니다 감사합니다