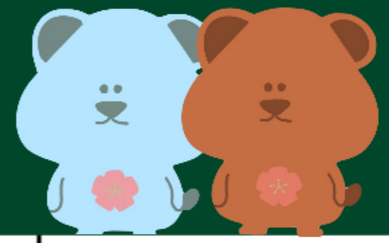
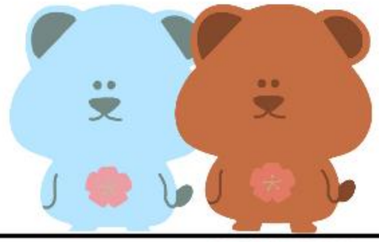


NLP – 혐오 표현 필터링 프로젝트

NLP팀 이승연 임세영 조서영

EWHA EURON

목차

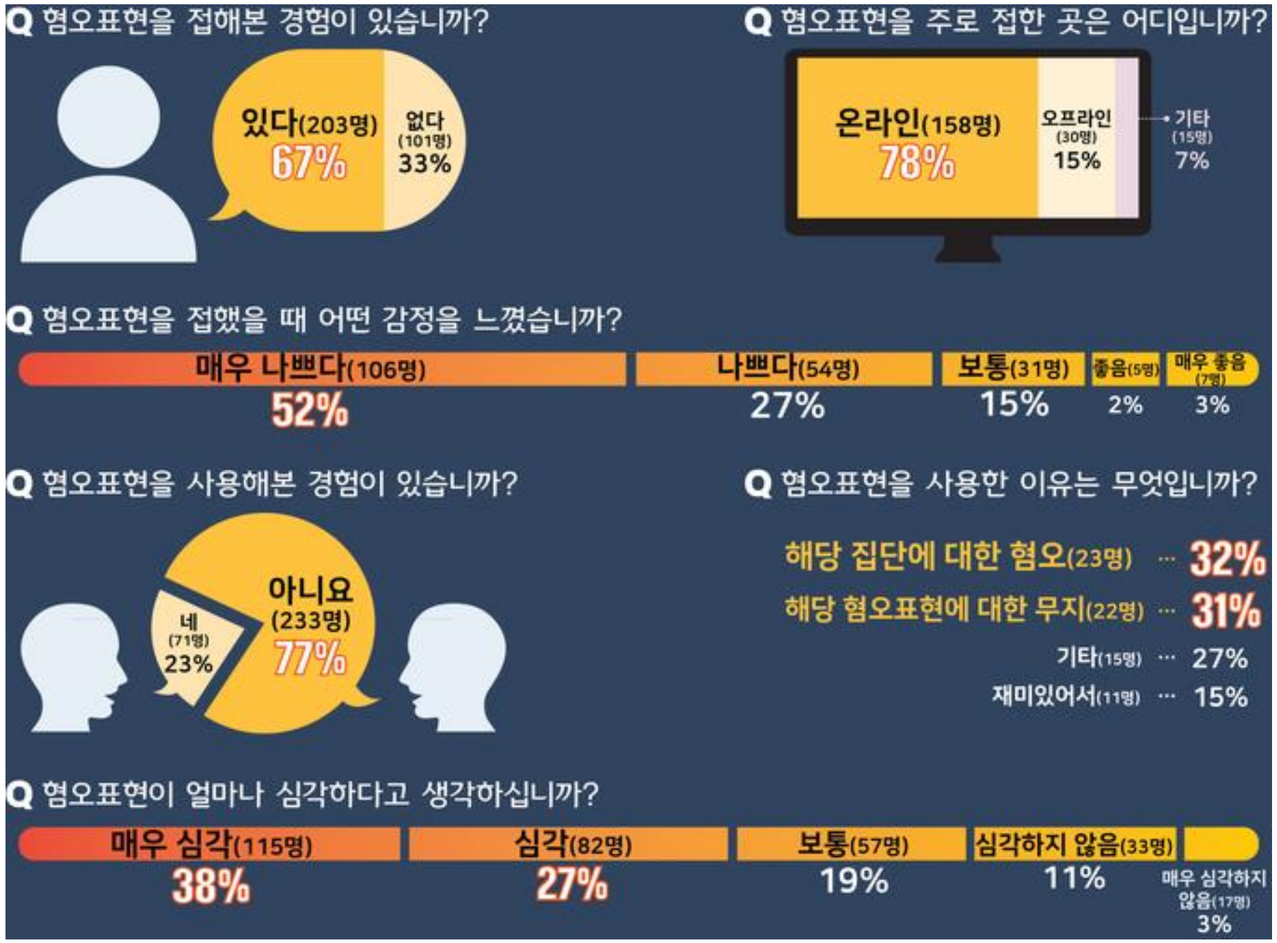


EWHA EURON

개요

EWHA EURON

#01 개요



혐오 표현과 그에 따른 언어 오염 실태가 심각

#01 개요

"차별·혐오 데이터 그대로"... AI '이루다' 운영 중단 요구 확산

이하나 기자 | 승인 2021.01.10 21:31 | 수정 2021-01-10 21:39 | 댓글 0

> '20살 여대생' 인공지능 '이루다' 논란
> 일부 남성 사용자는 AI 성희롱하고
> AI는 '레즈비언' 단어에 "혐오스럽다"
> 서비스 중단 요구 해시태그 운동 확산

혐오 표현을 그대로 학습하는 AI 챗봇

#01 개요

"혐오 표현 막는다"...IBS·심심이, AI 챗봇 윤리성 검증

입력 2021.08.26 09:21 수정 2021.08.26 09:21 가가

AI 챗봇의 윤리성 검증이 큰 화두로 던져짐

EWHA EURON

#01 개요

그렇다면, 챗봇을 사용하는데 있어서

어떻게 윤리성을 검증할 수 있을까?

일상생활 곳곳에 침투한 혐오표현을 잡아내고,

이를 분류할 수 있는 모델을 만들어보자!

데이터셋

EWHA EURON

데이터셋

데이터 영역	한국어	**데이터 유형**	텍스트
데이터 형식	JSON	**데이터 출처**	크라우드워커가 직접 생성
라벨링 유형	비윤리 텍스트(자연어)	**라벨링 형식**	JSON
데이터 활용 서비스	대화체 콘텐츠 모더레이션, 챗봇서비스, 음성비서서비스	**데이터 구축년도/ 데이터 구축량**	2021년/453,340문장(대화세트 132,807건)

https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=558

#01 데이터 포맷

#1 원문데이터 포맷

원문
{이정재 진짜 잘생겼다.|이정재도 이제 아저씨지...|남자 나이 40 넘어가면 그냥 폐급이야 ~ ㅋㅋ}

전처리 후
"origin_text":"#@인간및인간집단.인물.유명인# 진짜 잘생겼다.",
"origin_text":"#@인간및인간집단.인물.유명인#도 이제 아저씨지...",
"origin_text":"남자 나이 40 넘어가면 그냥 폐급이야 ~ ㅋㅋ",

#01 데이터 포맷

#2 이름 비식별화

비윤리적 대화 속에서 언급된 사람, 제품 이름이 접근성이 높은 곳에 공개되는 경우
사회적 논란이 발생할 수 있어 고유 이름을 비식별화함

#3 이름 비식별화 체계

인물의 성별이 비윤리적 표현에 영향을 많이 주는 특성을 반영하여 성별 표기함
(BTS → 인간및인간집단.조직및단체.가수그룹.남성,)

#02 데이터 구성

- 대화세트 (talksets)
- 문형 (ethic-frames)
- 어휘단위 (lexical units)

< 구축 규모 >

문장	비윤리 문장	문형	어휘 단위
453,340 문장	251,064 문장	258,904 개	77,978 개

#03 어노테이션 포맷

#1 대화세트 (talksets)

- **내용: 비윤리적 문장이 1 개 이상 포함된 대화세트들**

- **이름 비식별화 체계**

- 세종 전자사전의 '고유명사 하위분류체계'를 기본으로 하되 소범주는 과제에 맞게 변경함
- 방송/연예계에서 활동하는 인물의 소범주는 '유명인'으로 통칭함.
- 직업 구분이 모호한 경우가 많음
- 동명이인의 가능성 등 소범주 구별이 어려운 경우라면, 중범주까지만 기입함
- 성별을 특정 지을 수 있는 인간 및 인간집단의 경우 소범주 다음에 성별을 기입함

예: "인간및인간집단.인물.정치인.여성"

#03 어노테이션 포맷

#2 문형 (ethic-frames)

- 내용: 제출된 데이터에 등장한 모든 문형을 포함해서, 프로젝트에서 생성된 모든 문형

구분	항목명	타입	설명
1	id	number*	• 문형에 대한 고유 id
2	masked_text	string*	• 문형 정보 문장 전체 혹은 주요 구절만 문형으로 매핑함 • 패턴: 구체적 어휘단위 대신 N,V형태의 슬롯 ID로 표시 N_X = 명사류 어휘단위 V_X = 술어류 어휘단위 = {X	X = 등장 순서대로 1부터 시작 단, N과 V를 합쳐서 등장 순서를 매기지 않고 N, V 각각의 등장 순서를 매김} • 예시: "N1 진짜 N2 안 V1나 V2네"
3	slots[]	array*	• 문형 속 등장하는 슬롯들의 배열 • 예시: 문형 "N1 진짜 N2 안 V1나 V2네" 의 슬롯 ["N1","N2","V1","V2"]

#03 어노테이션 포맷

#3 어휘단위 (lexical units)

- 내용: 문형에 매핑된 어휘단위 정보
- 명사류(N)와 술어류(V)로 분류

구분	항목명	타입	설명
1	id	number*	• 어휘단위에 대한 고유 id
2	token	string*	• 어휘단위의 이름 • 예시: "한남"
3	features	array of string*	• 해당 어휘단위의 속성값+Q7:Q18 고빈도 출현 어휘를 중심으로 속성값 부여 비윤리성 문장에서 주로 등장하는 의미로 속성값 부여 속성값이 없는 경우 empty array • 어휘단위의 속성값 정보 파일 "어휘단위체계_목록.xlsx" 참고 위치: 3.Documents 〉 3_품질검증합의서,구축 및 검증계획서 • 예시: "애들" = ["인간.인칭"] "먹다" = ["행위.일상"] "따먹다" = ["행위.관계"]
4	pos	string*	• 해당 어휘단위의 형태소 분석 정보 N = 명사류 어휘단위 V = 술어류 어휘단위 • 유효값: N, V

모델 설명

EWHA EURON

#03 모델 설명

- **Transformer**

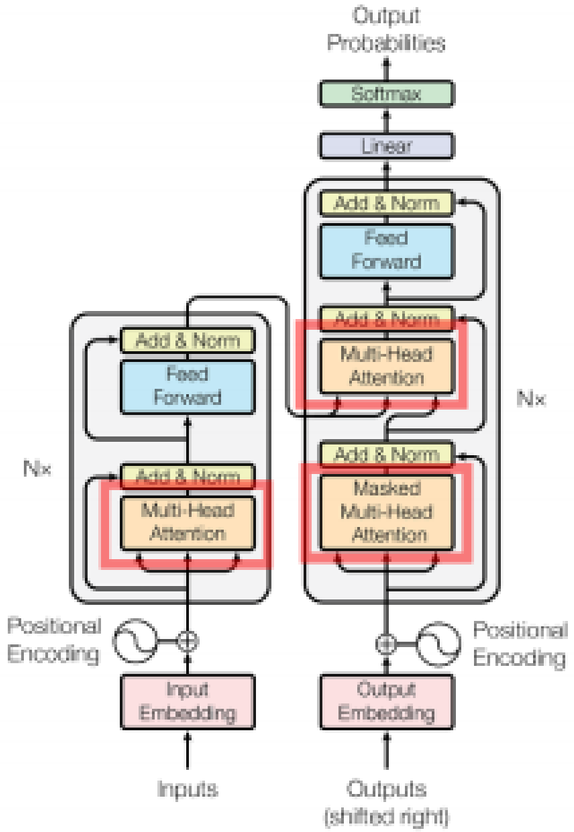


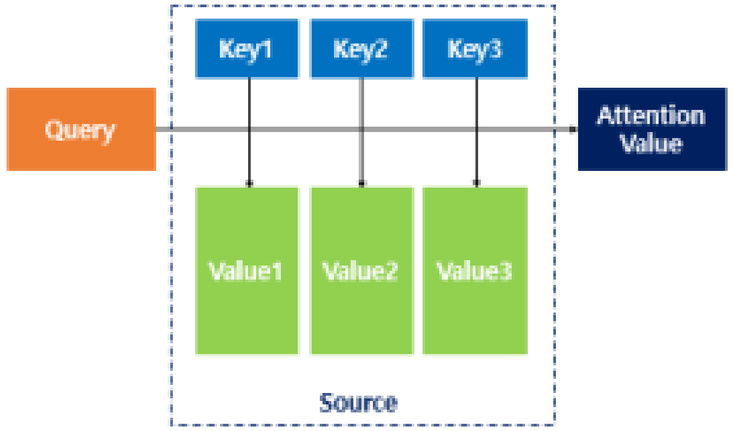
Figure 1: The Transformer - model architecture.

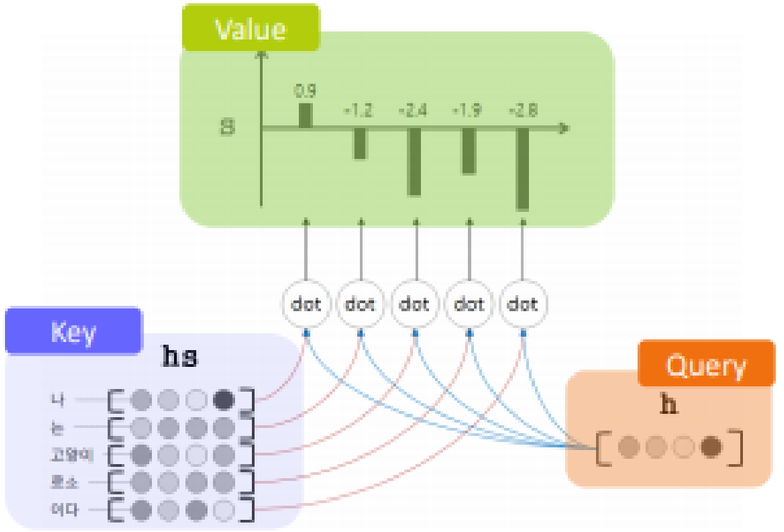
- Got rid of all the recurrent parts
- **Self-attention** and Fully-connected layer : To get rid of sequential parts in sequence transduction
- **Relying entirely on an attention mechanism** to draw global dependencies between input and output

EWHA EURON

#03 모델 설명

- Attention mechanism

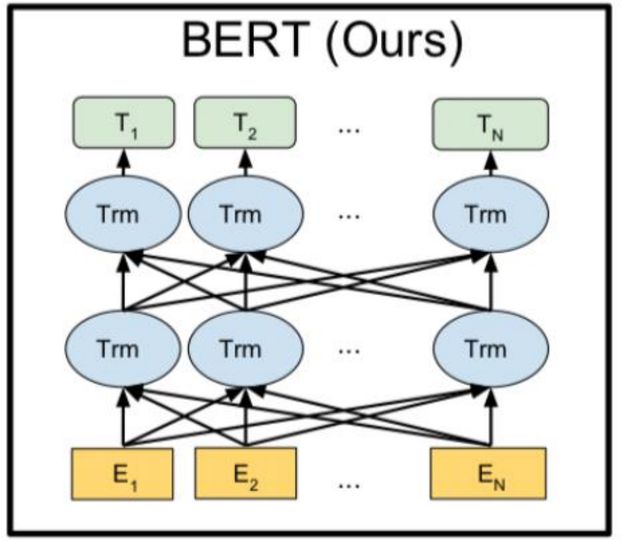


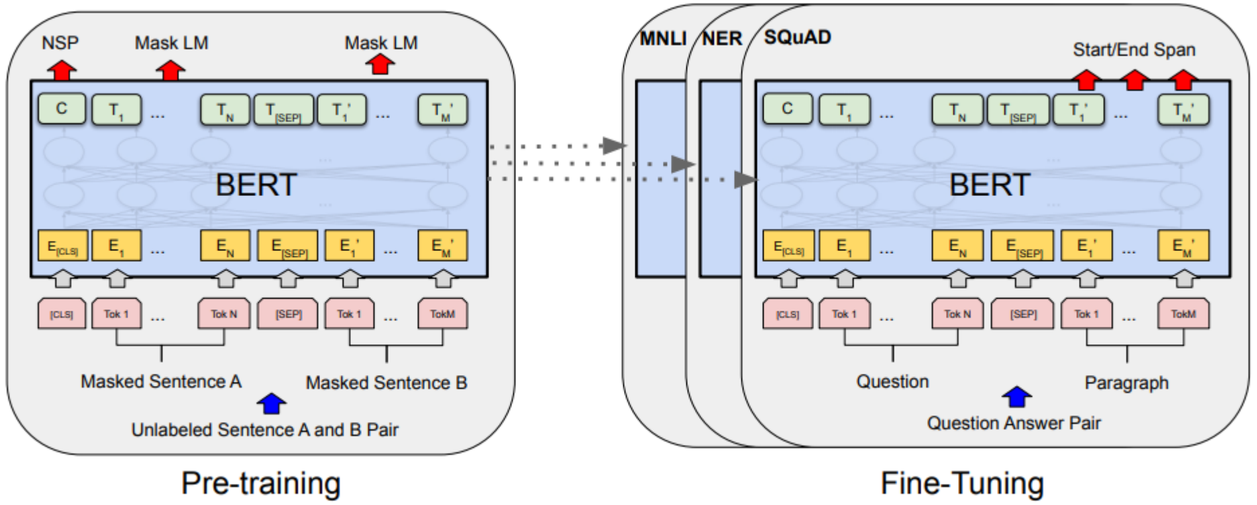


- The attention function obtains the similarity of all the ‘Keys’ for a given 'Query’
- And the similarity is weighted and reflected in each of the ‘Value’ mapped to the keys
- It returns all the 'Value' that reflects the similarity (weighted sum)

#03 모델 설명

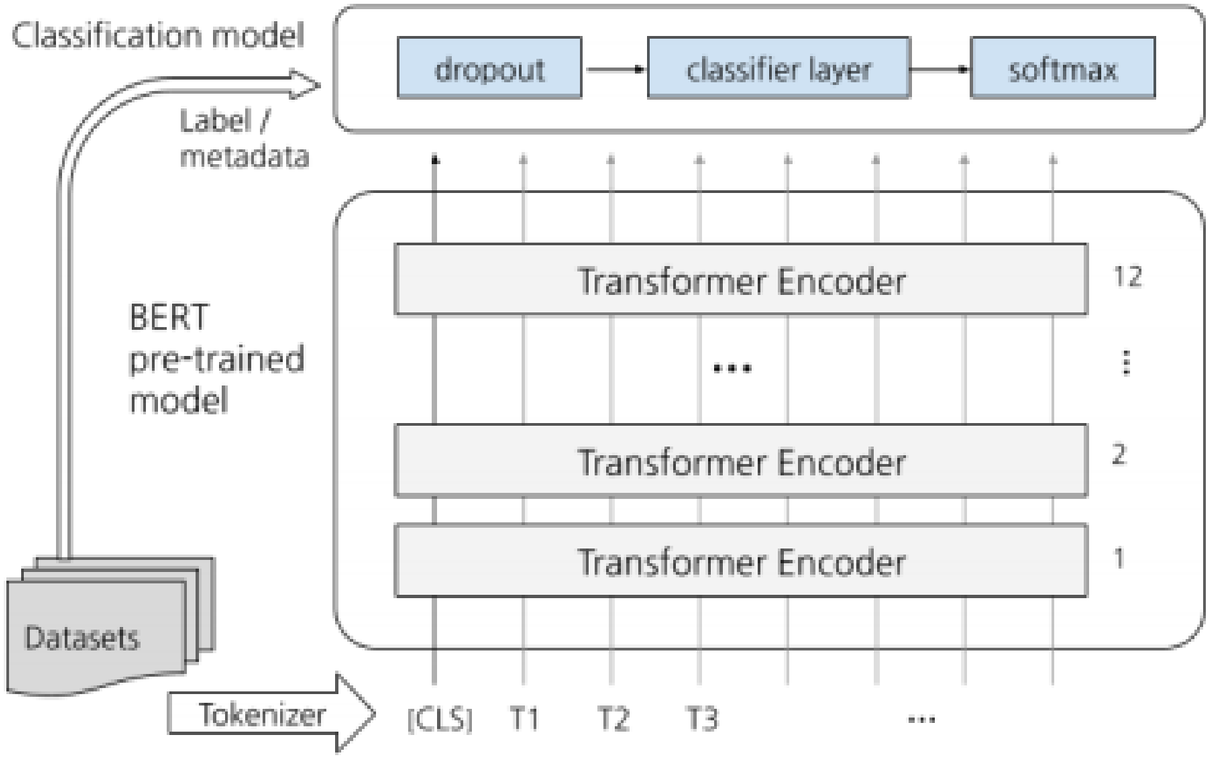
- BERT는 트랜스포머를 이용해서 위키피디아(25억 단어)와 BooksCorpus(8억 단어)와 같은 레이블이 없는 텍스트 데이터로 사전 훈련된 언어 모델
- BERT는 레이블이 없는 방대한 데이터로 사전 훈련된 모델로 레이블이 있는 다른 작업(Task)에서 추가 훈련과 함께 하이퍼 파라미터를 재조정하는 방식을 사용하여 성능이 높은 편
- 파라미터 재조정을 위한 추가 훈련 과정을 파인 튜닝(Fine-tuning)이라고 함





#03 모델 설명

- 8개의 타입으로 분류된 비도덕/무도덕 문장 데이터셋을 학습하여, 입력된 문장의 분류 타입을 판단하는 모델
- BERT 사전학습 모델을 사용한 멀티 클래스 분류 모델 적용
- 모델을 다운로드 받아서 로컬에서 돌릴 수 있도록 조금 수정



#03 모델 설명

- Num examples = 12664
- Evaluation Batch size = 64

- Accuracy: 대략 70%

```
model.zero_grad()
mb = master_bar(range(int(args.num_train_epochs)))
for epoch in mb:
    epoch_iterator = progress_bar(train_dataloader, parent=mb)
    for step, batch in enumerate(epoch_iterator):
        model.train()
        batch = tuple(t.to(args.device) for t in batch)
        inputs = {
            "input_ids": batch[0],
            "attention_mask": batch[1],
            "labels": batch[3]
        }
        if args.model_type not in ["distilkobert", "xlm-roberta"]:
            inputs["token_type_ids"] = batch[2]  # Distilkobert, XLM-Roberta don't use segment_ids
        outputs = model(**inputs)

        loss = outputs[0]

        if args.gradient_accumulation_steps > 1:
            loss = loss / args.gradient_accumulation_steps

        loss.backward()
        tr_loss += loss.item()
        if (step + 1) % args.gradient_accumulation_steps == 0 or (
                len(train_dataloader) <= args.gradient_accumulation_steps
                and (step + 1) == len(train_dataloader)
```

EWHA EURON

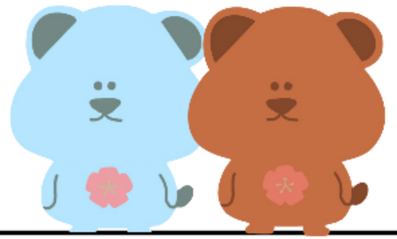
#03 모델 설명

- 챗봇 학습 과정에 있어 혐오 표현이 함께 학습되지 않도록 방지 가능

- 혐오 표현뿐만 아니라 인공지능 학습에 있어 문제점으로 제기되는 '편견 학습' 문제도 해결 가능

인공지능 윤리 문제 해결의 초석!

THANK YOU



EWHA EURON