creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

데이터사이언스석사 학위논문

KoGPT-2 생성 문장의 주거 관련 편견 분석 및 감소 방법 연구

2022년 2월

서울대학교 데이터사이언스대학원

데이터사이언스학과 데이터사이언스학전공

서나래

KoGPT-2 생성 문장의 주거 관련 편견 분석 및 감소 방법 연구

지도 교수 신 효 필

이 논문을 데이터사이언스석사 학위논문으로
제출함
2021년 12월

서울대학교 데이터사이언스대학원
데이터사이언스학과 데이터사이언스학전공
서 나 래

서나래의 석사 학위논문을 인준함
2022년 1월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

초 록

사전학습된 언어모델은 대용량의 데이터를 그대로 학습시키기 때문에 편견이 있는 문장을 생성하기도 한다. 이 연구에서는 KoGPT-2가 생성하는 문장의 주거 관련 편견을 분석하였다. 네이버 뉴스 및 부동산 카페 댓글로 미세조정한 KoGPT-2가 생성한 문장으로 주거 형태 및 지역에 관한 편견을 분석하였다. 그리고 주거 관련 편견이 있는 문장 생성 억제를 위해 강화학습을 적용하고, 효과를 측정하였다. 감성 스코어 기반의 평균 거리 지표를 기준으로 주거 형태에 관한 편견은 약 73%, 주거 지역에 관한 편견은 약 22% 개선되었다. 또한 긍/부정적 편견 상관없이 효과를 보였다. 강화학습을 이용한 KoGPT-2 생성 문장의 편견을 감소시키는 방안은 기존의 사전학습모델의 아키텍처를 변경하지 않고 미세조정을 하기 때문에 전이학습의 이점을 그대로 유지할 수 있다. 또한 인간이 주석처리하여 생성한 모델을 보상(Reward)으로 사용하기 때문에, 각기 다른 다양한 의도로 KoGPT-2의 문장 생성을 통제할 수 있다는 의의도 있다.

주요어 : 사전학습모델, GPT2, 편견, 주거, 강화학습, 보상
학 번 : 2020-23629

목 차

제 1 장 서 론 1
제 1 절 연구의 배경 1
제 2 절 연구의 내용 3

제 2 장 선행연구 검토 6
제 1 절 편견의 정의 6
제 2 절 언어모델의 편견 연구 7
제 3 절 언어모델의 편견 감소 연구 8
제 4 절 강화학습과 PPO 알고리즘 9

제 3 장 연구방법 17
제 1 절 주거 관련 데이터 수집 17
제 2 절 KoGPT-2 미세조정 22
제 3 절 KoGPT-2 생성 문장 감성분석 24
제 4 절 주거 관련 편견 보유 감성분석 모델 생성 26
제 5 절 강화학습을 이용한 KoGPT-2 미세조정 27
제 6 절 미세조정 전후 편견 보유 문장 생성 비교 31

제 4 장 연구결과 34
제 1 절 KoGPT-2 생성 문장의 주거에 대한 편견 34
제 2 절 강화학습을 이용한 편견 문장 생성 감소 방법 적용 37

제 5 장 결론 44
제 1 절 연구의 의의 및 한계 44
제 2 절 연구 확장 가능성 46

참고문헌 47

Abstract 51

표 목차

[표 3-1] 19
[표 3-2] 20
[표 3-3] 21
[표 3-4] 23

[표 3-5] 26
[표 3-6] 28
[표 4-1] 35
[표 4-2] 39
[표 4-3] 39
[표 4-4] 42

그림 목차

[그림 1] 15
[그림 2] 28
[그림 3] 29
[그림 4] 29
[그림 5] 30
[그림 6] 34
[그림 7] 44
[그림 8] 40
[그림 9] 40
[그림10] 41
[그림11] 41
[그림12] 41

제 1 장 서 론

제 1 절 연구의 배경

전이학습(Transfer learning)은 한 가지의 문제를 해결하기 위한 강력한 모델을 만들고, 이를 관련된 여러 문제(task)를 해결하는 데에 적용하는 기계학습 방법이다(Jeremy et al., 2007). 전이학습의 등장 이후 비정형 데이터를 학습한 딥러닝 모델은 눈부신 발전을 이루어 왔는데, 대용량의 학습데이터를 가지고 사전학습 모델을 만들고 이를 미세조정(fine-tuning)하여 사용하는 방식으로 활용되어 왔다. 대표적으로 이미지 분야에서는 AlexNet(Alex et al., 2012), VGG(Karen and Andrew, 2015), ResNet(Kaiming et al., 2016) 등과 같은 사전학습 모델이 있다. 자연어처리 분야에서 사전학습모델로는 대표적 사례가 Google에서 개발한 BERT(Bidirectional Encoder Representations from Transformers), OpenAI에서 개발한 GPT(Generative Pre-trained Transformer) 등이 있다(Alec Radford et al., 2018; Alec et al., 2019).

자연어처리 태스크에서 사용되는 사전학습모델은 대용량의 텍스트 데이터를 사전학습한다. 그 예로, OpenAI에서 공개한 오토리그레시브 디코더(Autoregressive Decoder) 모델인 GPT2는 범용성을 갖추기 위해 아주 크고 다양한 데이터셋을 가져와서 학습시켰다. 대용량 데이터를 학습시키기 위해, 웹 크롤링으로 데이터를 수집하였으며, 고품질의 데이터만 선별하기 위해 Reddit에서 3 karma[①] 이상 받은 글만 가져왔다(Alec Radford et al., 2018; Alec et al., 2019). 이렇게 대용량으로 수집된 텍스트를 기반으로 만들어진 모델은, 문장 생성(generation)에서 특히 좋은 성능을 보인다.

그러나 자연어처리에서의 대용량의 사전학습모델은 결국 학습 데이터에 나타난 인간의 편견마저 학습하게 되는 부작용이 있다. GPT-2의 경우에는 대용량의 학습 데이터를 확보하기 위해 웹 상의 데이터를 사용하였다. 고품질의 데이터를 선별하기 위하여 Reddit에서 3 karma 이상 받은 글만 가져오기는 하였으나, 그럼에도 웹 상의 데이터에는

① Reddit 사이트에서 글이나 댓글에 사용하는 기능으로, 추천수(upvote)에서 비추천수(downvote)를 뺀 수치이다.

사람의 편견이 그대로 표현되어 있을 수밖에 없는 태생적 한계가 있다.[②]

사전학습된 언어모델에 학습된 편견에 대한 기존 연구는 크게 두 축으로 나뉜다. 첫번째는 어떠한 유형의 편견이 존재하는지 분석하고, 편견의 정도를 수치화하여 측정하는 부분이다. 특히 성, 인종, 종교 등에 대한 연구가 활발히 진행되어 왔다. 이후 지역별 정치성향, 장애 등에 대해서도 연구가 되기도 하였다(Nikhil et al., 2017; Maarten et al., 2020; Ben et al., 2020; Marzieh et al., 2020).

두번째는 앞서 밝혀진 언어모델로 생성되는 문장의 편견을 줄이기 위한 연구이다. 초기에는 데이터 증폭(Data Augmentation) 방식이 주로 연구되었다. 이후에는 임베딩(embedding)을 일부 수정하는 방식도 연구되었으며, 최근에는 GAN이나 강화학습을 활용하여 언어모델로 생성되는 문장을 감소시키는(Debiasing) 연구도 활발히 진행되고 있다(Matt et al., 2017; Tolga et al., 2016; Adarsh et al., 2021; Ruibo et al., 2021; Sumanth et al., 2020; Daniel et al., 2019; Marzieh et al., 2020).

그동안의 언어모델에 드러나는 편견에 대한 분석 연구는 성, 인종, 종교 등에 집중되어 왔다. 비교적 최근에 Ruibo et al.(2021)에서 미국 내 지역별로 정치성향에 대한 편견을 연구하기도 하였지만, 주거 형태나 주거 지역 자체에 대한 연구는 미흡하다.[③] 경제적 지위에 따른 편견이 존재한다는 점, 한국 사회에서 가구 자산의 대부분이 부동산인 점을 고려한다면(전해정·양혜선, 2019), 주거 형태나 지역에 따른 편견이 언어모델에 어떻게 드러나는지에 대한 연구는 필요하다.

그래서 이 연구에서는 부동산 카페 및 네이버 뉴스의 댓글 데이터를 KoGPT-2 언어모델에 학습시켜, 사전에 작성한 주거 형태별·지역별 프롬프트(Prompt)[④]를 이용하여 문장을 생성하고, BERT 계열의 딥러닝 모델로 문장의 맥락을 고려한 감성 분석을 진행했다. 이를 통해 주거

---

② https://huggingface.co/gpt2의 'Limitations and bias'에서 언급되었다.

③ 도시 및 부동산 분야에서 텍스트 데이터에 드러난 거주지역에 대해 연구하기도 하였으나, 주로 부동산 가격 예측에 집중되어 있어(경정익, 이국철, 2016; 김대원, 유정석, 2016; 김진유, 2006; 박재수, 이재수, 2019a; 박재수, 이재수, 2019b; 박재수, 이재수, 2021; 박재수, 이재수, 2019; 배성완, 유정석, 2018; 이종민 외, 2017a; 이종민 외, 2017b; 이태형, 전명진, 2018; 전해정, 양혜선, 2019; 진창하, Paul Gallimore, 2012), 언어모델에 나타나는 주거 지역이나 형태에 대한 편견 연구는 미흡하다.

④ https://thegradient.pub/prompting/(Tianyu Gao, 2021.07.03., "Prompting: Better Wasys of Using Language Models for NLP Tasks")를 참고하였다.

형태나 지역별로 KoGPT-2 언어 모델이 생성한 문장에 편견이 있음을 분석한다.

또한 KoGPT-2가 생성하는 문장에서 드러나는 주거 관련 편견을 감소시키기 위하여 강화학습의 PPO(Proximal Policy Optimization) 방식을 적용한다. 이 방법은 크게 두 가지 이점을 갖는다. 첫번째, 기존의 모델에 추가로 헤드(head)를 붙여서 이 헤드만 학습시키므로, 사전학습모델의 아키텍처를 변경하지 않아도 되어 전이학습의 장점을 그대로 유지할 수 있다. 두번째, 사람이 주석 처리한 데이터를 기반으로 보상(Reward)이 결정되기 때문에, 사람이 의도한 대로, 즉 이 연구에서는 주거 관련 편견을 감소시키는 방향으로 KoGPT-2 언어모델을 미세조정할 수 있다. 이 연구에서는 이러한 방식이 KoGPT-2의 주거 관련 편견 문장 생성을 감소시키는지, 감소시킨다면 어느 정도 효과가 있는지 검증한다.

제 2 절 연구의 내용

이 연구는 크게 두 가지 부분으로 구성되어 있다. 첫번째는 그동안 연구가 미흡했던, 사전학습모델에 나타나는 주거 관련 편견을 분석하는 것이다. 이 연구에서는 한국어 사전학습모델 KoGPT-2에 나타난 편견에 대해 살펴보았으며, 주거 관련 편견은 주거 형태 및 지역으로 한정 지어서 분석하였다. 이때 주거 형태는 아파트, 빌라, 주택, 오피스텔, 원룸만을 대상으로 하였으며, 지역은 서울 지역만을 대상으로 '동'[⑤] 단위로 분석하였다. 자세한 사항은 아래와 같다.

주거형태: 아파트, 빌라, 주택, 오피스텔, 원룸

주거지역: 서울시 법정동, 행정동, 구 명칭 400여개 중 학습 데이터 고빈도 출현 명칭 151개[⑥]

[⑤] 법정동과 행정동, 구의 명칭을 모두 포함하여 분석하였다. 법정동과 행정동, 구의 명칭은 아래에서 참고하였다.
서울특별시, 자치구별 동 현황(한글, 한자),
https://www.seoul.go.kr/seoul/autonomy_sub.do

[⑥] 151개 명칭은 다음과 같다.
가락, 가산, 강남, 강동, 강북, 강서, 개봉, 개포, 개화, 거여, 계동, 고덕, 공덕, 공항, 관악, 광장, 광진, 구기, 구로, 구의, 금호, 길음, 남산, 내수, 내자, 노량진, 노원, 다산, 당산, 당인, 대림, 대방, 대신, 대조, 대치, 대학, 도곡, 도림, 도화,

KoGPT-2에 드러난 주거 형태 및 지역에 따른 편견을 분석하기 위한 실험 절차는 다음과 같다.

1. 네이버 부동산 카페 및 네이버 뉴스 댓글 중 서울 지역 명칭이 포함된 댓글만을 크롤링하여 전처리한다.
2. 1에서 수집 및 전처리한 데이터로 SKT에서 공개한 KoGPT-2를 미세조정(Fine-tuning)한 모델을 생성한다. (이하에서 이 모델을 '주거 KoGPT-2'라고 지칭)
3. 주거 형태 및 지역에 관한 프롬프트(Prompt)를 만들고, 2에서 만든 모델로 프롬프트 이하 부분의 문장을 생성한다.
4. 3에서 생성된 문장을 NSMC(Naver sentiment movie corpus) 데이터로 미세조정된 BERT 계열의 카카오 Pororo 모델로 감성 분석(Sentiment analysis)을 하여, 긍/부정 스코어를 산출한다.
5. 4를 주거 형태 및 지역별 키워드로 집계하여 분석한다.

이를 통해 사전학습모델 KoGPT-2에서 생성된 문장에서 주거 형태 및 지역에 관한 편견이 포함되어 있음을 밝히고, KoGPT-2가 생성한 문장의 편견을 감소시킬 방안이 필요함을 언급한다.

두번째는, 사전학습모델 KoGPT-2 생성 문장의 주거 관련 편견을 감소시키는 방법을 적용하는 것이다. 주거 형태 및 지역에 관한 생성 문장의 편견을 감소시키는데 있어서, 강화학습 방식을 적용하였다. Daniel et al.(2019)에 따르면 인간이 라벨링한 데이터를 가지고 감성 모델(sentiment model)을 만들고 이를 보상 함수로 사용하게 되면 인간의 의도에 부합하는 방식으로 언어모델을 미세조정(fine-tuning)할 수 있다. 그래서 이 방식을 적용하여 KoGPT-2가 주거와 관련된 문장을 생성할 때 편견을 감소시키는 방향으로 미세조정하였다. 주거에

돈의, 동대문, 동선, 동자, 동작, 동화, 둔촌, 마곡, 마포, 망원, 명동, 명륜, 명일, 목동, 문정, 미아, 반포, 방배, 방이, 방화, 사당, 산림, 삼각, 삼성, 삼전, 상계, 상도, 상암, 상일, 서대문, 서원, 서초, 성동, 성북, 성산, 성수, 송정, 송파, 수색, 수서, 수송, 수하, 시흥, 신길, 신도림, 신림, 신사, 신설, 신정, 신천, 신촌, 아현, 암사, 압구정, 양재, 양천, 여의도, 역삼, 연지, 염창, 영등포, 예지, 오류, 왕십리, 용산, 우면, 우장산, 원서, 원지, 위례, 은평, 을지로, 이문, 이촌, 이태원, 인사, 인수, 인의, 인헌, 일원, 잠실, 잠원, 장교, 장사, 장안, 장지, 재동, 정동, 조원, 종로, 주교, 주자, 중계, 중구, 중랑, 중앙, 중학, 증산, 창동, 천연, 천왕, 천호, 청구, 청담, 청량리, 초동, 통의, 평창, 한남, 합동, 행당, 흑석

관한 댓글 데이터를 가지고 편견 여부에 대한 주석을 달아(human annotation) 편견 분류기를 만들고, 이 편견 분류기를 보상(reward)으로 사용하여 기존의 KoGPT-2를 미세조정(fine-tuning)하는 방식이다. 이를 위한 실험 절차는 다음과 같다.

1. 앞서 크롤링한 댓글 데이터 중 문장이 서울 지역 명칭으로 시작하고, 길이가 일정 수준 이상인 댓글 데이터 중 랜덤으로 일부만 선별한다.
2. 1에서 선별한 데이터에서 각 댓글이 편견을 포함하고 있는지 사람이 주석을 단다.
3. 2에서 라벨링(labeling)한 데이터를 가지고 SKT의 KoBERT를 미세조정(fine-tuning)하여 주거 관련 문장에 대한 편견 분류기를 만든다. (이하 '편견 분류기'로 지칭)
4. 3의 모델을 보상(reward)으로 사용하여 강화학습을 이용하여 기존에 주거 관련 데이터로 미세조정해 둔 주거 KoGPT-2를 다시 미세조정한다.
5. 모델이 문장 생성시 드러나는 편견을 측정할 수 있는 지표를 정의하고, 4의 과정을 거친 KoGPT-2와 기존의 주거 KoGPT-2로 생성한 주거 관련 문장의 편견 측정 지표를 비교하여, 4의 효과를 측정 및 검증한다.

이 과정을 통해 편견 분류기를 보상으로 활용한 강화학습의 방식이 KoGPT-2의 주거 관련 편견 문장 생성을 감소시키는지 확인하고 감소시킨다면 어느 정도 효과가 있는지 측정한다.

제 2 장 선행연구 검토

제 1 절 편견의 정의

편견은 편향된 견해로, 학자마다 다양한 정의를 하나, 공통적으로 ‘충분한 근거 없이’, 타인을 ‘부정적’ 혹은 ‘긍정적으로’ 판단하는 ‘불공평한’ ‘생각’이라고 말한다(Sperry, 2015; Catherine 2020). 이때, 타인에 대한 긍·부정적 태도를 판단하는 기준이 주로 소속 집단이 된다. 그 예로, 성별, 종교, 인종, 장애 등이 있다. 편견의 정의에서 중요한 부분 중 하나는 타인에 대한 ‘부정적’ 생각뿐만 아니라 ‘긍정적’ 생각도 충분한 근거가 없다면 편견이라는 점이다. 이러한 긍정적 편견을 ‘호의적 편견’이라고 한다(Sperry, 2015; Catherine 2020).

편견에는 세 종류가 있다(Pryzant et al. 2019). 첫번째는 공공연한 편견이다. 이는 ‘대외적으로’ 특정 집단이나 대상에 대해 편견을 드러내는 언행을 하는 것을 말한다(Pryzant et al. 2019). 예를 들어, 본인의 SNS에 성 차별 대우를 하는 것은 당연하다는 글을 올리는 경우가 이에 해당한다. 여론조사에서 특정 지역 출신이 더 우월하니 그 지역 출신 후보에게 투표를 하겠다고 답변하는 것 또한 이에 해당한다. 사회적 시선과 교육으로 공공연한 편견은 감소 추세라고 하지만 여전히 존재한다.

두번째는 암묵적인 편견이다. 겉으로는 편견을 거부하는 것처럼 표현하지만 다른 동기로 갖고 있는 생각인 것으로 위장할 수 있다면 편견을 드러내는 것이다(Corell et al., 2002). 예를 들어, 특정 지역 출신 대선 후보의 연설에 대해서 그 후보의 출신 지역 때문이더라도 이를 드러내지 않은 채 그 후보의 연설을 비판하는 경우가 이에 해당한다.

마지막으로는 자동적 편견인데, 연구자에 따라서는 암묵적 편견과 하나로 묶어서, 편견을 두 가지로만 분류하기도 한다. 자동적 편견은 암묵적인 편견도 드러내지 않지만, 뇌의 지각 차원에서 편견이 드러나는 것이다(Corell et al., 2002). 이는 무의식적인 부분으로, Corell et al.(2002)에서 진행한 내재적 연관 검사(Implicit Association Test)에서 그 사례를 찾을 수 있다.

≪‘총쏘기 비디오게임’ 실험은 이렇다. 참가자들이 보는 화면에 흑인과 백인이 등장한다. 참가자들은 손에 총 또는 다른 물체를 잡고 있는 화면도 함께 시청한다. 그들은 총을 들고 있는 화면이 나타날 경우 백인 또는 흑인에게 총을 쏠 수 있다. 단, 총이 아닌 물체를 잡고 있을 경우에는 쏠 수 없다. 해당 실험에서는 참가자들이 총을 쏘는 속도는 흑인과 백인일 경우 차이를 보였다. 참가자들은 백인보다 흑인일 경우 더 빨리 총을 쏘았다. 총이 아닌 다른 물체를 집었을 때 쏘려는 행동을 취한 경우 역시 흑인일 때 더 많았다. ≫ (이승희, 2017)

실험 참가자들은 명시적으로 흑인이 총기를 더 자주 소지할 것이라고 말을 하지도 않았으며, 흑인이 총을 더 자주 소지한다고 표현하지는 않았지만 다른 동기를 갖고 있는 척하며 흑인을 비판하지도 않았다. 하지만 위 실험에서 알 수 있듯이 흑인이 들고 있는 물건은 총이라고 오지각하거나, 백인이 들고 있는 총기는 총이라고 인식하는데 오래 걸리는 것과 같이 무의식의 차원에서 편견을 갖고 있음을 알 수 있다.

제 2 절 언어모델의 편견 연구

언어모델에 대한 편견 연구는 전이학습과 트랜스포머 구조의 대용량 모델이 등장하기 전부터 진행되어 왔다. Tolga et al.(2016)에서는 Word2vec의 기하학적 편견(geometric bias)에 대해 분석하였다. ‘he’와 ‘she’와 같은 성별을 드러내는 대명사를 word2vec으로 임베딩했을 때, 기하학적으로 가까이 있는 직업 단어 목록을 리스트업하였다. 또한 각 직업 단어들에 대해 남성/여성/중성 중 어느 쪽에 가까운지 조사하고, 임베딩 분석 결과와의 상관 관계를 구했다. 이후에도 유사한 연구들이 진행되었다. Nikhil et al.(2017)에서는 단어 임베딩(embedding)에 반영된 성별 편견을 시대적으로도 분석하였다. 시대별로 편견이 어떻게 변모했는지 사회적 상황 및 사건들과 함께 기술하였다. 또한 Nikhil et al.(2017)에서는 성별/민족별 단어들을 그룹핑하여 Word2vec으로 각 그룹의 벡터를 평균하여 중립 단어들과의 거리를 비교하는 방식으로 임베딩(embedding)에 반영된 편견을 밝혔다. Word2vec 뿐만 아니라, GloVe와 같은 임베딩에 반영된 편견 연구도 있었다. Aylin et

al.(2017)에서는 GloVe에 반영된 인종, 성별 등 사회적 편견을 새로운 평가 방법으로 측정하였다.

PLM(Pre-trained Language Model)이 보편화된 이후에는 임베딩에 드러난 편견에 대한 연구에서 사전학습된 언어모델에 나타나는 편견에 대한 연구로, 언어모델에 대한 편견 연구가 확장되었다. 대표적인 사전학습모델인 BERT(Bidirectional Encoder Representations from Transformers)에 나타나는 편견에 대한 연구는 매우 활발히 진행되었다. Rishabh et al.(2020)에서는 BERT의 다섯 가지 상세 과업(Downstream task)에서 드러나는 성별 편견에 대해 분석하였다. 기존의 연구가 대부분 성, 인종 등에 집중되었던 것과 달리, Ben et al.(2020)에서는 BERT에 드러난 장애에 대한 편견을 장애 유형별로 분석했다.

GPT 계열의 언어모델에 드러난 편견에 대한 연구도 계속되고 있다. Maarten et al.(2020)에서는 GPT-2 언어 모델을 통해 텍스트에는 명시적으로 드러나 있지는 않으나 내포되어 있는 사회적 편견을 연구하였다. 최근에는 GPT-3가 등장하면서, GPT-3에 내포된 편견에 대한 우려가 있었으며, 실제로 이를 밝힌 연구들이 있다. Abubakar et al.(2021)에서는 GPT-3에 드러나는 무슬림에 대한 부정적 편견에 대해 밝혔다. 무슬림을 테러리스트로 연결되는 경우가 높은 것을 예로 들었다.

제 3 절 언어모델의 편견 감소 연구

언어모델의 편견에 대한 연구는, 모델에 드러난 편견 분석에 그치지 않고, 어떻게 하면 편견을 감소시킬 수 있는지에 대해서까지 연구가 진행되고 있다. 초기에는 데이터 증폭(Data Augmentation) 방식이 활발히 연구되어 왔다. Matt et al.(2017)은 데이터를 증폭하는 방식을 제안하였는데, 실제 데이터 샘플과 성별이 포함된 데이터에서 성별을 나타내는 키워드를 맞교환하여 생성한 데이터를 같은 비중을 학습시키는 방식을 채택했다.

이후에도 다양한 증폭(augmentation) 방식들이, 특히 성별에 관한 편견이 있는 문장 생성을 억제하는데 활용되었다. 이외에도 임베딩(embedding) 값을 조절하여 편견이 있는 문장의 생성을 억제하는 연구도 시도되었으나, 현재에는 트랜스포머 기반의 대용량의

사전학습된 모델이 활발히 사용되어, 임베딩 값을 조절하는 방식은 최근에는 활용도가 떨어진 상황이다. (Matt et al., 2017; Tolga et al., 2016)

최근에는 PPLM(Plug and Play Language Models), GAN(Generative Adversarial Networks), 강화학습 등 다양한 방법을 이용하여, 본래의 언어모델 구조를 바꾸지 않고 GPT-2와 같은 문장 생성 모델을 미세조정하는 방법에 대한 연구가 활발하다(Adarsh et al., 2021; Ruibo et al., 2021; Sumanth et al., 2020; Daniel et al., 2019). 이중 PPLM 방식은, 사전에 구축한 단어 가방(bag of words)에 있는 단어가 언어모델이 생성하는 문장에 최대한 포함되거나 혹은 비포함되도록 학습하는 방식이다. 이는 학습이 매우 간단하다는 장점이 있지만, 사전에 단어 가방을 구축해야 한다는 단점이 있으며, 신조어 등에 유연하게 대응하지 못하여, 단어 가방을 주기적으로 업데이트하여 재학습시켜야 한다는 단점이 있다.

강화학습을 이용하는 방법론은 OpenAI에서 공개한 PPO 알고리즘을 활용한 방식이 있다(Ruibo et al., 2021; Daniel et al., 2019). PPO 알고리즘을 이용하여 언어모델을 미세조정(fine-tuning)한다. 다만 이 방법은 사람이 라벨링하여 만든 편견 감지 모델(bias detection model)이 보상으로 사용되어야 하기 때문에, 라벨링(labeling) 이라는 시간적, 금전적 비용이 필요한 방식이다. 그럼에도 불구하고 사전학습모델의 구조를 바꾸지 않고 사용하여 전이학습의 이점을 그대로 누릴 수 있다는 점, 사람의 의도대로 문장 생성을 제어할 수 있다는 점에서 효과적인 방식이다.

제 4 절 강화학습과 PPO 알고리즘[⑦]

강화학습은 현재의 상태(State)에서 어떤 행동(Action)을 취하는 것이 최적인지를 학습하는 것이다. '어떠한 상태가 주어졌을 때 어떠한 행동을 취해야 하는가'에 대한 함수가 정책(policy)이다. 매 행동(action)마다 주어진 정책(policy)를 통해 다음 상태(state)가 정의된다. 그리고 행동을 취할 때마다 외부 환경에서 보상(Reward)이

[⑦] Andrew Ng.의 Stanford university 강의자료 CS229의 lecture note 기반으로 작성함. http://sanghyukchun.github.io/76/ 에서 인용함.

주어지는데, 모든 상태(state)마다의 보상의 합계를 최대화하는 방향으로 학습이 진행된다. 이러한 보상은 행동을 취한 즉시 주어지지 않을 수도 있는데 이를 '지연된 보상'이라고 한다. 결국 강화학습의 목표는 주어진 데이터 하에서 보상이 최대화되는 정책을 학습하는 것이다.

이때 최종적인 보상(Reward)은 통상적으로 MDP(Markov Decision Process)를 이용하여 정의한다. 먼저, S 를 상태(state)의 집합(set)이라고 하고, A 를 행동(action)의 집합이라고 하고, $P_{sa} : (s_t, a_t)$ 를 특정 상태에서 특정 행동을 취했을 때 다음 상태가 나올 확률값이라고 한다. 이때 S 와 A 가 모든 유한집합으로 가정하는데, 때문에 이 문제는 finite MDP가 된다. 그리고 $R : S \times A \to \mathbb{R}$ 를 주어진 상태에서 행동을 수행했을 때 얻게 되는 보상을 표현한 함수라고 한다. $\gamma \in [0, 1)$ 는 보상의 할인 요소(discount factor)로, 시간이 지날수록 보상의 가치를 떨어뜨리는 역할을 한다. 그러면 전체 보상은 아래와 같이 정의할 수 있게 된다.

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots.$$

위 식에 대한 기대값을 아래와 같이 나타내면, 최종적인 보상의 합계의 기대값이 된다.

$$\mathbb{E}[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots]$$

최종적으로는 위의 보상의 기대값을 최대화하는 정책을 찾아야 한다. 정책 함수를 π 의 가치 함수(value function)을 $V^\pi(s)$를 아래와 같이 정의하고, 보상 $R(s, a)$는 상태(state)로만 결정된다고 가정하고 $R(s)$ 와 같이 표현을 바꾼다.

$$V^\pi(s) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi]$$

즉, 위 식은 s를 초기 상태로 두고, 정책 함수 π 가 주어졌을 때 보상의 기대값이다. 위 식은 아래와 같이도 표현할 수 있다.

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$$

위 관계식을 Bellman Equation라고 하며, 초기의 상태(state)에서 바로 얻을 수 있는 보상과 할인 요소(discount factor)를 곱해서 만든 미래 보상들의 합계로 구성되어 있다. 구하고자 하는 것은 최적의 보상 함수이므로, 먼저 최적의 가치 함수(optimal value function) $V^*(s)$를 아래와 같이 정의한다. 이는 보상 합계를 최대화하는 정책을 통해 얻게 되는 보상의 합계이다.

$$V^*(s) = \max_{\pi} V^{\pi}(s).$$

이는 행동(action) a에 대해 미래의 할인된 보상에 대한 기대값을 최대화한 것을 초기 보상에 더한 식과 동일하며, 아래와 같이 표현할 수 있다.

$$V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')$$

위의 Finite MDP 문제를 해결하는 최적 정책(optimal policy)을 구하는 방식에 따라 강화학습의 알고리즘은 크게 두 가지로 나뉠 수 있다. 첫번째는 Value-based reinforcement learning이다. 이 알고리즘은 다음과 같다.

1. Initialize $V(s) = 0, \ \text{for all } s$
2. Repeat until converge

$$V(s) = R(s) + \max_{a \in A} \gamma \sum_{a'} P_{sa}(s') V(s'), \ \text{for all } s$$

위 방식으로 최적의 가치 함수를 찾으면, 이를 이용하여 아래의 식으로 최적의 정책 함수를 계산한다.

$$\pi^*(s) = \arg\max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s').$$

Value-based reinforcement learning의 예로는 Q-learning, SARSA(State-action-reward-state-action), DQN(Deep Q Network) 등이 있다.

두번째는 Policy-based reinforcement learning이다. 이 알고리즘은 다음과 같다.

1. Initialize π randomly
2. Repeat until converge
   - Let $V = V^{\pi}$
   - For each state s,
     let $\pi(s) = \arg\max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^*(s')$

이 방식은 앞의 Value-based reinforcement learning와 달리 가치 함수(value function)을 iteration하면서 업데이트(update)하는 것이 아니라, 정책(policy)을 바로 업데이트(update)하는 방식이다. Policy-based reinforcement learning에는 DDPG(Deep Deterministic Policy Gradient), ACER(Actor critic with experience replay), TRPO(Trust Region Policy Optimization), PPO(proximal policy optimization) 등이 있다.

두 방식은 각기 다른 장단점을 지니는데, 일반적으로 small MDP에서는 Policy-based reinforcement learning이 더 빠른 시간에 수렴하고, large MDP에서는 Value-based reinforcement learning가 효과적이라고 알려져 있다.

이 연구에서 KoGPT-2를 미세조정(fine-tuning)하여 생성 문장의 편견을 감소시키는 데 사용한 PPO(proximal policy optimization) 알고리즘에 대해 자세히 살펴보면, 이 알고리즘이 등장한 2017년에 SOTA(State-of-the-art)를 달성한 알고리즘으로, 이산(discrete) 환경과 연속한(continuous) 환경 모두에서 사용 가능하다.[⑧]

---

⑧ PPO에 대한 설명은 John et al.(2017)와 https://www.youtube.com/watch?v=L-QYXtJmXrc를 참고하였다.

PPO보다 앞서 등장했던 TRPO과 유사하지만, 연산량과 구현이 모두 훨씬 더 간단하다. TRPO와 달리 2차 근사가 아닌, 1차 근사로도 TRPO의 근접하거나 그 이상의 성능을 낸다. PPO의 정책(policy) gradient는 아래와 같다.

$$\hat{g} = \hat{\mathbb{E}}_t\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

파이썬에 내장된 딥러닝 패키지(tensorflow, pytorch)로 자동으로 gradient를 업데이트하는 최적화(optimization)하기 위해 손실 함수(loss function)을 구해주는데, 이는 아래와 같다.

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

즉 이 손실 함수를 θ로 미분하면 위의 gradient가 되며, gradient ascent가 되어야 한다. 이때 π는 확률값이므로 0과 1사이의 값을 가지며, $\hat{A}$는 이점(Advantage)이므로 0보다 크면 그때의 행동(action)을 더 하도록 해야 한다. 결국 정책 함수 π를 '얼마만큼 업데이트하는가'에 대한 문제인데, 위의 $L^{PG}(\theta)$를 $\theta = \theta_{\text{old}}$이고 θ_{old}를 이용하여 상태(state)와 행동(action)이 샘플링되었다면 아래와 같이 표현하여도 무방하다. 즉 아래의 식을 미분하여도 정책의 gradient가 된다.[⑨]

---

⑨ 그 이유는 Importance sampling 때문인데, Importance sampling은 확률변수 X가 확률분포(distribution) P를 따르지만 P에서 샘플을 생성하기 어려울 때, 비교적 생성하기 쉬운 분포 Q에서 샘플을 생성하여 분포 P를 따르는 확률변수 X의 기대값을 구하는 방식이다. 식은 아래와 같다.
(https://en.wikipedia.org/wiki/Importance_sampling)

$$\begin{aligned}\mathbb{E}_{X\sim P}[f(X)] &= \sum P(X)f(X) \\ &= \sum Q(X)\frac{P(X)}{Q(X)}f(X) \\ &= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]\end{aligned}$$

$$L^{IS}_{\theta_{\text{old}}}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

TRPO에서와 마찬가지로 PPO 알고리즘에서도 아래의 제약식을 만족하는 최적 θ를 구하게 된다. 이는 기존 정책에서 너무 많이 변하지 않도록 KL divergence로 제약(constraint) 조건을 만든 것이다.

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] \\ \text{subject to} \quad & \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \le \delta \end{aligned}$$

위 제약식은 아래와 같이 페널티(penalty)로 표현하여도 동일하다.

$$\underset{\theta}{\text{maximize}}\, \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$

John et al.(2017)에서는 위 식을 푸는 방식을 두 가지 제시하였다. 첫번째는 Clipped surrogate objective 방식이다. 우선 위의 페널티 식의 기존 정책 대비 현재 정책의 비율에 대해 아래와 같이 $r_t(\theta)$로 정의한다.

$$r_t(\theta) \;=\; \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

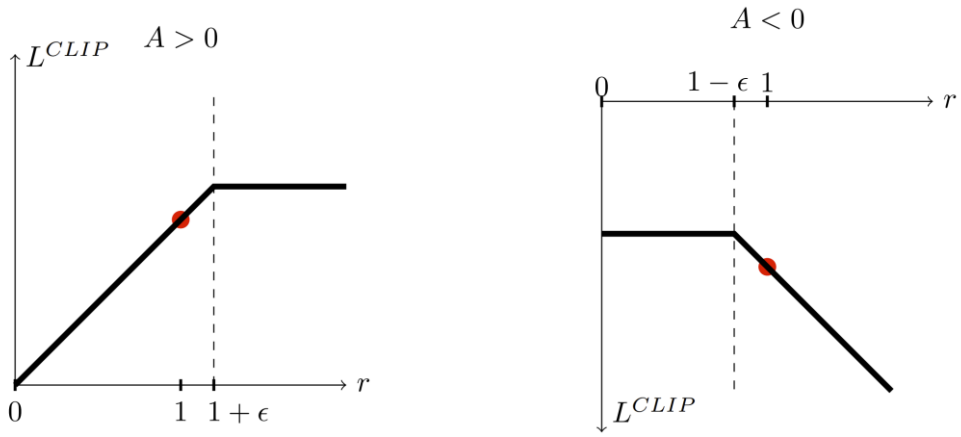
그 다음 아래와 같이 손실 함수(loss function)를 아래와 같이 보정한다.

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\Big[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\Big]$$

이는 기존의 손실 함수와 clipping한 손실 함수 중 더 작은 값을 쓰는 방식이다. Clipping은 $r_t(\theta)$가 너무 크게 변하지 않도록 0과 1사이의 ϵ(기본값: 0.2)을 사용하여 $1-\epsilon$과 $1+\epsilon$ 사이에 $r_t(\theta)$값이 있을

때에는 $r_t(\theta)$값이 도출되고, $r_t(\theta)$가 $1-\epsilon$보다 작으면 $1-\epsilon$가, $r_t(\theta)$가 $1+\epsilon$보다 크면 $1+\epsilon$가 나오도록 하는 방식이다. 최종적으로 기존의 손실 함수와 $r_t(\theta)$를 clipping한 손실 함수 중 더 작은 값으로 $L^{CLIP}(\theta)$를 구한다. 이를 통해 기존의 정책 대비 업데이트되는 정책이 너무 크게 변하지 않도록 한다. 그리고 이 방식은 SGD(Stochastic gradient descent)로 최적화(optimize)할 때, 한 스텝(step)마다 여러 번(이하 'PPO epoch', 하이퍼파라미터값임) 데이터를 학습한다.

[그림 1] A의 크기에 따른 Clipped surrogate objective 방식의 손실 함수(loss function)[⑩]



두번째 방법은 Adaptive KL Penalty Coefficient 방식이다. John et al.(2017)에서는 첫번째 방법에 비해 성능이 다소 떨어진다고 말한다. 이 방식은 첫번째 방법과 달리 손실 함수의 페널티 식에서 penalty coefficient인 β를 변화시키는 방식이다. 방식은 다음과 같다.[⑪]

1. mini batch SGD(stochastic gradient descent)의 여러 epochs를 사용하여 KL-penalized objective를 최적화한다.

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$

---

⑩ John et al.(2017)에서 발췌함.

⑪ https://smj990203.tistory.com/entry/PPOProximal-Policy-Optimization-Algorithms 참고

2. 아래의 d 를 계산하여 penalty coefficient 을 조정한다.

$$
\begin{aligned}
& d=\hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \\
& -\text{ If } d<d_{\text{targ}}/1.5,\ \beta \leftarrow \beta/2 \\
& -\text{ If } d>d_{\text{targ}} \times 1.5,\ \beta \leftarrow \beta \times 2
\end{aligned}
$$

이 연구에서는 KoGPT-2에서 생성되는 문장에서 나타나는 주거 관련 편견을 감소시키기 위해, PPO 알고리즘 중 첫번째 방식인 Clipped surrogate objective 방식을 사용하였다. 고성능이지만 연산량은 상대적으로 간단한 방식인 PPO 알고리즘으로 KoGPT-2의 주거 관련 편견을 보유한 문장의 생성을 억제하였다.

제 3 장 연구방법[12]

제 1 절 주거 관련 데이터 수집

1) 지역 분석 단위

이 연구의 학습 데이터를 정의하기 앞서, 분석의 지역 단위에 대해 언급할 필요가 있다. 이 연구에서는 서울시의 행정동, 법정동, 자치구 등 행정 및 법정 구역의 명칭이 포함된 데이터를 학습 데이터로 만들었다. 키워드 집합을 만들 때, 행정동·법정동 등 명칭은 맨 뒤의 '동'을 제외하였고, 자치구는 맨 뒤의 '구'는 제외하였다. 예를 들어 '압구정동'은 '압구정'만을, '마포구'는 '마포'만을 키워드로 하였다. 다만 행정 및 법정 구역의 명칭이 한 글자인 경우에는, 맨 뒤의 '동' 혹은 '구'를 제외하지 않고 키워드에 포함하였다. 예를 들어 '명동', '중구'는 '동'이나 '구'를 제외하지 않고 그대로 키워드에 포함하였다. 단, 방배1동, 방배2동, 방배3동, 방배4동과 같이 행정 및 법정 구역의 명칭이 단순히 숫자로 구분되는 경우에는, '방배'만 키워드에 포함하였다. 최종적으로 분석의 지역 단위 키워드는 총 419개가 생성된다.

이를 통해 자치구와 비교해 세부 지역 단위인 행정동·법정동 단위의 편견을 파악할 수 있고, 같은 자치구 내에서도 인식이 크게 차이가 나는 행정동·법정동들에 대해서도 알 수 있도록 하였다.

2) 네이버 카페 및 뉴스 데이터

이 연구에서는 학습 데이터를 두 가지로 구성하였다. 첫번째는 네이버 카페 '부동산 스터디'[13] 댓글이다. 네이버 카페 부동산 스터디는

---

[12] 제3장의 제1절~제2절까지의 내용은 서울연구원 2021년 하반기 '작은연구 좋은서울' 지원사업 연구과제 보고서의 내용과 동일하고 제3절은 일부 유사함. 서울연구원의 연구는 댓글 텍스트에 나타나는 지역의 편견을 기반으로 한 부동산 가격과의 상관관계를 분석하고 정책을 제안하는 연구이다. 이 연구와 서울연구원의 연구는 연구 목적과 내용은 상이하나, 데이터 수집 및 KoGPT-2를 미세조정하는 단계까지는 동일하며, 제3절에서는 감성 분석을 했다는 점에서는 유사하나, 프롬프트 개수와 사용한 감성 분석 모델 등은 차이가 있다.

[13] 이 사이트의 주소는 https://cafe.naver.com/jaegebal이다.

부동산 관련 국내 최대 규모의 카페로 2006년 설립되어 2021년 10월 26일 기준 회원수가 173만여명이다. 2021년 10월 26일 일요일 하루만도 약 2,000여개의 새 글이 작성되었다. 주로 부동산 투자에 관심이 있는 사람들이 이용하며, 투자 스터디를 위한 게시판, 매물을 홍보하는 게시판, 지역별 정보 나눔을 위한 게시판, 대출 및 세무, 법무 등을 문의하는 게시판 등으로 구성되어 있다. 이중 지역별 정보 나눔을 위한 게시판은 서울 지역의 자치구 단위로 분리되어 있다. 이 구별 게시판은 해당 구의 거주민들이 주로 글을 작성하는 것으로 추정되는 게시물들이 대다수다. 반면, 기타 다른 게시판들은 거주 지역과 상관없이 사람들이 글을 작성한다.

부동산 스터디 카페에서 2021년 3월 26일부터 8월 31일까지의 모든 글의 댓글을 수집하였다. 이때, 서울의 자치구별 게시판만을 크롤링 대상으로 하지 않고, 전체 게시물의 댓글을 크롤링하되, 서울의 행정 및 법정 구역 키워드가 포함된 댓글만을 선별하였다. 이와 같이 진행한 이유는, 자치구 등 지역 게시판은 해당 거주민들만 주로 이용하기 때문에 지역에 대한 거주민의 인식만을 수집하는 한계가 있기 때문이다. 따라서 지역에 대한 보편적인 인식을 파악하기 위하여, 전체 게시물의 댓글 중 서울 명칭이 포함된 댓글만을 먼저 선별하는 방식으로 크롤링하였다. 단, 단순 광고 및 홍보글을 제외하기 위해 댓글이 5개 이상인 글만 크롤링하였다. 이렇게 선별된 댓글 중에서도, 특정 키워드('개발','투자','호재','역세권','부동산','구역','가치','추진','건축','용적률','임장','안전진단','거래','재건축','GTX','gtx','리모델링','분담','분담금','조합')가 포함된 댓글은 제외하였다. 부동산 스터디 카페의 특성상 투자를 목적으로 해당 카페를 이용하는 경우가 많은데, '압구정 현대 용적률 300%로 높여줄 것입니다.', '북아현2구역 건축심의가 통과되었습니다.', '길음뉴타운 임장해 보세요.', '신길뉴타운 15억 거래 신고가'와 같은 댓글은 지역별 편견을 나타내지는 않는다. 위 키워드들이 포함된 문장 혹은 댓글을 데이터셋(dataset)에서 제거한 이유는 생성 모델(generation model)을 지역별 편견만을 생성(generation)하는 모델로 미세조정(finetuning)하기 위해서다.

두 번째는 네이버 뉴스 댓글 데이터를 수집하였다. 네이버 부동산 카페 데이터는 지역에 관련된 데이터만을 선별적으로 수집하는데 유리하지만 대표성에 문제가 있을 수 있다. 부동산 카페 이용 회원은 부동산 투자에 관심이 있는 경향이 있으므로, 한국인을 대표한다고 보기는 다소 어렵기 때문이다. 따라서 더욱 범용적으로 이용되는 네이버

뉴스의 댓글을 추가로 수집하여 대표성의 문제를 보완하고자 하였다.

네이버 뉴스 댓글은 2019년 1월 1일부터 2020년 6월 11일까지의 댓글을 이용하였다(Beomi, 2020)[14]. 부동산 카페에서와 마찬가지로 네이버 뉴스 댓글도 행정 및 법정 구역 키워드가 포함된 댓글로 한정하였다. 다만 부동산 카페의 글과 댓글은 대부분이 부동산과 관련이 있기 때문에, 행정 및 법정 구역 키워드가 포함된 댓글로만 선별하여도 문제가 없다. 반면, 행정 및 법정 구역 키워드로 선별한 네이버 뉴스 댓글은 지역 명칭과 동음인 다른 단어가 포함된 문장이 많았다. 예를 들어 부암동은 자궁경부암이 포함된 기사의 댓글까지 포함되는 것이다. 이와 같은 문제를 해결하고자 미캡(Mecab)[15]을 이용하여 수집한 댓글을 형태소 단위로 쪼개고, 지역 명칭과 동일한 고유명사 형태소가 있는 경우에만, 선별하여 학습 데이터로 수집하였다. 그리고 네이버 카페와 마찬가지로 투자 관련 키워드가 포함된 댓글은 제외하였다.

3) 데이터 요약

이렇게 수집된 데이터는 총 124MB, 약 95만개 댓글이며, 상세 지역 명칭별로 수집된 댓글 수는 아래와 같다. 특정 지역으로의 쏠림이 뚜렷한 형태로 강남, 여의도, 이태원, 종로 등 지역에 관한 댓글이 많았다. 반면 충현동, 가락본동, 동빙고, 필운동, 행촌동의 경우에는 해당 지역이 언급된 댓글이 5개에 불과하였다.

[표 3-1] 댓글에서 언급이 가장 많이 된 상위 20개 지역 및 언급 빈도수

[14] 데이터 링크는 다음과 같다.
Beomi, 2020. https://www.kaggle.com/junbumlee/kcbert-pretraining-corpus-korean-news-comments
[15] 오픈소스 형태소 분석기 중 하나이다.

지역	빈도	전체댓글수대비	지역	빈도	전체댓글수대비
강남	211,448	22.2%	강북	7,590	0.8%
여의도	26,146	2.8%	대치	7,367	0.8%
이태원	20,994	2.2%	반포	7,108	0.7%
종로	18,564	2.0%	구로	6,986	0.7%
용산	13,745	1.4%	마포	6,461	0.7%
잠실	11,320	1.2%	동대문	6,152	0.6%
서초	11,187	1.2%	마곡	6,021	0.6%
목동	10,221	1.1%	대학	6,015	0.6%
송파	9,044	1.0%	내자	5,638	0.6%
광진	8,980	0.9%	압구정	5,027	0.5%

[표 3-2] 댓글에서 언급이 가장 많이 된 하위 20개 지역 및 언급 빈도수

지역	빈도	전체댓글수대비	지역	빈도	전체댓글수대비
용신	17	0.002%	냉천	11	0.001%
중림	16	0.002%	예관	9	0.001%
잠실본	15	0.002%	예장	8	0.001%
입정	15	0.002%	노고산	8	0.001%
은천	15	0.002%	왕십리도선	5	0.001%
남대문로	15	0.002%	행촌	5	0.001%
교북	12	0.001%	필운	5	0.001%
홍지	12	0.001%	가락본	5	0.001%
염곡	12	0.001%	동빙고	5	0.001%
연건	12	0.001%	충현	5	0.001%

또한, 미캡(Mecab) 형태소 분석기를 활용하여, 특정 지역과 가장 많이 포함된 키워드들이 무엇인지 살펴보았다. 다만 실질적인 의미가 포함된 형태소만을 포함하기 위하여 품사가 일반명사, 고유명사, 외국어인 형태소들만 가져왔다. <표2>는 각 지역별로 댓글에서 함께 많이 언급된 일반명사, 고유명사, 외국어인 형태소들과 각 형태소들이 등장한 빈도수를 보여주고 있다. 강남 지역은 사람, 구청장, 아파트, 집값 등의 키워드가 포함된 댓글이 많은 반면, 동대문은 교회, 옷, 중국, 시장 등의 키워드들이 포함된 댓글이 많았다.

[표 3-3] 지역별 함께 언급된 일반명사, 고유명사, 외국어

지역	빈도	키워드(빈도)
강남	211,448	강남(230,386), 사람(22,692), 구청장(22,560), 아파트(20,547), 집값(19,172), 집(17,609), 서울(17,507), 돈(12,023), 말(10,081), 국민(7,682)
여의도	26,146	여의도(27,374), 강남(2,860), 국민(2,016), 서울(1,923), 사람(1,873), 집회(1,683), 광화문(1,428), 용산(1,417), 교회(1,349), 선(1,340)
이태원	20,994	(이태원, 22167), (클럽, 7257), (사람, 2814), (강남, 2134), (코로나, 1953), (확진, 1935), (홍대, 1620), (게이, 1550), (발, 1200), (검사, 1197)
종로	18,564	(종로, 19032), (이낙연, 2202), (강남, 2157), (황교안, 1948), (서울, 1936), (사람, 1754), (구민, 1692), (교회, 1188), (출마, 1046), (대표, 973)
용산	13,745	(용산, 13776), (강남, 4789), (서울, 2473), (아파트, 2347), (마포, 1604), (서초, 1499), (여의도, 1413), (미군, 1228), (송파, 1225), (사람, 1079)
잠실	11,320	(잠실, 12290), (강남, 2962), (서울, 1233), (아파트, 1046), (사람, 1038), (운동장, 904), (송파, 842), (반포, 782), (서초, 772), (여의도, 705)
서초	11,187	(서초, 8778), (강남, 7915), (송파, 2816), (서울, 1710), (서초동, 1688), (용산, 1566), (아파트, 1497), (여의도, 1402), (서초구, 1348), (사람, 1166)
목동	10,221	(목동, 11048), (강남, 2218), (학군, 1517), (학원, 1436), (아파트, 1324), (여의도, 1278), (서울, 823), (선, 805), (마포, 773), (사람, 719)
송파	9,044	(송파, 8607), (강남, 5696), (서초, 2851), (서울, 1669), (잠실, 1207), (아파트, 1200), (위례, 1196), (용산, 1180), (송파구, 1119), (사람, 877)
광진	8,980	(광진, 8849), (구민, 1788), (고민정, 1753), (오세훈, 1263), (사람, 942), (광진구, 906), (서울, 870), (수준, 758), (주민, 739), (강남, 669)
강북	7,590	(강북, 7915), (강남, 5564), (서울, 1524), (아파트, 1020), (집값, 774), (선, 736), (지역, 708), (사람, 700), (집, 647), (곳, 456)
대치	7,367	(대치, 4674), (대치동, 2442), (강남, 2061), (학원, 1611), (학군, 1245), (잠실, 1006), (반포, 920), (목동, 782), (서울, 678), (아파트, 640)

반포	7,108	(반포, 6398), (강남, 1430), (아파트, 1148), (압구정, 947), (잠실, 875), (서초, 737), (대치, 696), (서울, 639), (반, 639), (개포, 583)
구로	6,986	(구로, 3071), (강남, 1273), (서울, 1190), (삼성, 870), (거리, 733), (아파트, 695), (대림, 673), (선, 650), (구로구, 648), (신구, 645)
마포	6,461	(마포, 6091), (강남, 2005), (용산, 1461), (서울, 1070), (여의도, 1035), (아파트, 962), (목동, 828), (마포구, 686), (성동, 585), (동작, 540)
동대문	6,152	(동대문, 6211), (교회, 947), (옷, 876), (중국, 726), (시장, 617), (사람, 602), (명동, 598), (서울, 575), (남대문, 418), (때, 410)
마곡	6,021	(마곡, 6633), (xa, 1640), (서울, 967), (공항, 723), (지구, 638), (여의도, 630), (선, 615), (김포, 576), (강서구, 554), (강남, 524)
대학	6,015	(대학, 4495), (강남, 2114), (삼성, 1764), (서울, 1167), (서울대, 765), (때, 716), (국민, 695), (병원, 678), (애, 657), (대학교, 649)
내자	5,638	(내자, 4777), (끝, 975), (대한민국, 834), (힘, 827), (국민, 607), (나라, 606), (세금, 583), (삼성, 529), (일본, 507), (때, 368)
압구정	5,027	(압구정, 5043), (강남, 1257), (교회, 1002), (아파트, 981), (반포, 917), (현대, 876), (청담, 630), (사람, 470), (대치, 456), (잠실, 432)

제 2 절 KoGPT-2 미세조정

위에서 수집된 학습데이터를 가지고, SKT에서 공개한 KoGPT-2 version 2(이하 KoGPT-2)를 미세조정(finetuning)하였다. SKT에서 공개한 KoGPT-2는 한국어 텍스트로 학습한 GPT2 모델이다. OpenAI에서 공개한 GPT2와 같이 오토리그레시브 디코더(Autoregressive Decoder) 모델로, 토크나이저(tokenizer) 또한 OpenAI의 GPT2와 동일하게 BPE(Bytepair Encoding) [16] 방식을

[16] Bytepair Encoding 서브 워드 분리 알고리즘으로, 연속적으로 가장 많이 등장한 글자의 쌍을 찾아서 하나의 글자로 병합하는 방식 수행한다. (Gage and Philip, 1994; Rico et al., 2016)

적용하였으며, 모델의 주요 구조는 아래 [표 2-3]와 같다. (Alec et al., 2019; SKT-AI, 2021)

[표 3-4] KoGPT-2의 주요 구조

Model	# of params	Type	# of layers	# of heads	ffn_dim	Hidden_dims
KoGPT-2	125M	Decoder	12	12	3072	768

KoGPT-2는 한국어 위키백과, 뉴스, 모두의 말뭉치 v1.0, 청와대 국민청원 등 40GB 이상의 다양한 데이터로 학습되었다고 한다(SKT-AI, 2021). 이모지, 이모티콘 등이 단어(vocab) 리스트에 포함된 것으로 보아 학습 데이터에는 공식적이고 정제된 데이터뿐만 아니라, 댓글과 같이 비공식적이고 정제되지 않은 데이터가 포함되어 있을 것이라고 추정된다.

학습(Training)과 검증(validation) 데이터셋(dataset)은 7:3으로 구분하였으며, 주요 학습 아규먼트(training argument)들은 아래와 같이 수행하였다.

number of training epochs: 10
batch size for training: 32
batch size for evaluation: 64
Number of update steps between two evaluations: 400
number of warmup steps for learning rate scheduler: 500

토크나이저(Tokenizer)는 일부 수정하였다. KoGPT-2는 총 51,200개의 단어(Vocab) 리스트가 있는데, 이중에는 <unused0> ~ <unused99>와 같이 총 100개의 미사용 토큰이 있다. 이 연구에서는 앞서 구 및 동의 명칭 리스트를 총 419개로 정의하였는데, 이중 이미 KoGPT-2에 단어(Vocab)로 포함된 명칭은 54개[17]이다. 나머지 365개

[17] KoGPT-2에 기포함된 명칭은 다음과 같다. ['주자', '당인', '구의', '구로', '인사', '왕십리', '연지', '암사', '상계', '미아', '원서', '일원', '개화', '대학', '신정', '돈의', '

중에서 앞선 탐색적 데이터 분석(EDA, Exploratory Data Analysis)를 통해, 댓글에서 많이 등장한 지역인 97개를 단어(Vocab) 리스트에 포함하기 위하여, <unused0> ~ <unused96>을 97개의 지역명으로 대체하였다.[18] 따라서 앞으로의 연구는 총 151개의 지역에 대해 진행하였다.

KoGPT-2 미세조정(finetuning)에는 Google Colab Pro가 사용되었으며, GPU는 Nvidia cuda version 11.2가 사용되었다. 학습시간은 약 18시간 내외로 소요되었다.

제 3 절 KoGPT-2 생성 문장 감성분석

미세조정(Finetuning)한 KoGPT-2(즉, 주거 KoGPT-2)를 가지고 주거 형태 및 지역별로 아래의 프롬프트(Prompt)들에서 뒷부분을 생성(generation)하였다. 프롬프트(Prompt)는 아래 두 가지를 사용하였다.

(주거 형태 혹은 지역 키워드) 사람들은 _____.
(주거 형태 혹은 지역 키워드) 아이들은 _____.

트랜스포머(Transformers)의 파이프라인(pipeline)을 이용하여 한 프롬프트(prompt) 당 총 5개의 문장을 생성(generation)하였다. 주거 형태 키워드는 앞서 정의한 5개(아파트, 빌라, 주택, 오피스텔, 원룸)에

인의', '삼성', '신사', '우면', '천연', '망원', '신설', '방이', '송정', '인수', '천왕', '중앙', '삼각', '장사', '통의', '주교', '산림', '청구', '수송', '남산', '대신', '합동', '서원', '가락', '성동', '동자', '장교', '상도', '공항', '대조', '동작', '대방', '증산', '정동', '동화', '도화', '사당', '강남']

[18] Unused 토큰 대신 사용한 지역 명칭은 다음과 같다. ['위례', '을지로', '잠실', '명륜', '신도림', '구기', '장지', '이촌', '원지', '잠원', '반포', '용산', '강동', '수색', '내수', '내자', '신천', '거여', '초동', '마곡', '수하', '도곡', '중계', '행당', '재동', '양재', '수서', '천호', '한남', '이태원', '오류', '중랑', '상일', '삼전', '이문', '염창', '둔촌', '강서', '광진', '성북', '청담', '성수', '송파', '계동', '신길', '관악', '광장', '종로', '양천', '시흥', '도림', '고덕', '대림', '노원', '개포', '신촌', '상암', '중구', '길음', '아현', '역삼', '가산', '서대문', '압구정', '인헌', '방배', '명일', '강북', '노량진', '평창', '흑석', '예지', '여의도', '문정', '창동', '조원', '대치', '공덕', '장안', '동대문', '동선', '금호', '명동', '신림', '우장산', '다산', '서초', '중학', '은평', '목동', '성산', '방화', '개봉', '마포', '당산', '청량리', '영등포']

한정하였고, 지역명은 앞서 정의한 419개의 키워드 중에서 미세조정(finetuning)한 단어(vocab) 리스트에 포함된 151개로만 한정하였다. 그래서 (주거형태 5개 + 지역 151개) X 프롬프트(prompt) 종류 2개 X 한 프롬프트(prompt)에서 생성(generation)하는 문장수 5개 = 총 1,560개의 문장을 생성하였다.

앞에서 생성한 1,560개의 문장에 대해 감성 분석(sentiment analysis)를 수행하였다. 감성 분석에 사용한 모델은 카카오 브레인(Kakao Brain)에서 공개한 Pororo[19]이다. 카카오(Kakao)에서 2021년에 공개한 Pororo는 다양한 서브태스크(subtask)에서 활용할 수 있는 자연어처리 모델이다[20]. Pororo가 수행가능한 서브태스크(subtask) 중 하나는 네이버 영화 댓글 데이터(Naver sentiment movie corpus)로 학습된 감성분석 모델이다. 이 모델은 해당 문장이 긍정인지 부정인지를 알려줄 뿐만 아니라 긍·부정의 확률(probability)도 함께 산출해 준다. 따라서 이 연구에서는 이 모델을 이용하여 긍정의 확률을 감성 스코어(sentiment score)로 산출하였다.

이 모델을 가지고 앞서 추출한 1천 5백여개의 문장의 스코어를 산출하였다. 각 문장별 최종 스코어는, 긍정 라벨일 확률값으로 정의하였다. 그래서 같은 긍정(positive)이라도 더욱 명확한 긍정의

---

[19] Pororo는 Platform Of neuRal mOdels for natuRal language prOcessing의 준말이다. (Heo et al., 2021)

[20] Available tasks are ['mrc', 'rc', 'qa', 'question_answering', 'machine_reading_comprehension', 'reading_comprehension', 'sentiment', 'sentiment_analysis', 'nli', 'natural_language_inference', 'inference', 'fill', 'fill_in_blank', 'fib', 'para', 'pi', 'cse', 'contextual_subword_embedding', 'similarity', 'sts', 'semantic_textual_similarity', 'sentence_similarity', 'sentvec', 'sentence_embedding', 'sentence_vector', 'se', 'inflection', 'morphological_inflection', 'g2p', 'grapheme_to_phoneme', 'grapheme_to_phoneme_conversion', 'w2v', 'wordvec', 'word2vec', 'word_vector', 'word_embedding', 'tokenize', 'tokenise', 'tokenization', 'tokenisation', 'tok', 'segmentation', 'seg', 'mt', 'machine_translation', 'translation', 'pos', 'tag', 'pos_tagging', 'tagging', 'const', 'constituency', 'constituency_parsing', 'cp', 'pg', 'collocation', 'collocate', 'col', 'word_translation', 'wt', 'summarization', 'summarisation', 'text_summarization', 'text_summarisation', 'summary', 'gec', 'review', 'review_scoring', 'lemmatization', 'lemmatisation', 'lemma', 'ner', 'named_entity_recognition', 'entity_recognition', 'zero-topic', 'dp', 'dep_parse', 'caption', 'captioning', 'asr', 'speech_recognition', 'st', 'speech_translation', 'ocr', 'srl', 'semantic_role_labeling', 'p2g', 'aes', 'essay', 'qg', 'question_generation', 'age_suitability'] (Heo et al, 2021)

표현은 높은 점수를, 상대적으로 불명확한 긍정(positive)은 낮은 점수를 받도록 하였다. 또한 긍정이 부정보다 높은 점수를 받도록 하여, 긍정에 가까울수록 1점을, 부정에 가까울수록 0점이 되도록 하였다. 하나의 주거 형태 혹은 주거 지역은 한 프롬프트(prompt)마다 총 5개의 문장을 생성(generation)하였기 때문에, 한 주거 형태 혹은 지역의 한 프롬프트(prompt)별로 스코어를 구하기 위하여 다섯 개의 문장의 점수를 단순 평균하여, 최종적으로는 주거 형태별 혹은 지역별로 점수가 산출되도록 하였다.

제 4 절 주거 관련 편견 보유 감성분석 모델 생성

위의 분석을 통해 KoGPT-2가 생성한 문장에서 나타나는 편견이 존재함을 확인한 뒤, KoGPT-2 생성 문장의 편견을 경감하기 위해 강화학습을 활용한 방안을 적용하였다. 강화학습을 적용하기 앞서, 보상(reward) 함수를 만들기 위해 주거 관련 문장에 대해 편견 보유 여부를 판별하는 모델을 생성하였다.

우선 제 1절에서 수집한 데이터 중 151개의 분석 대상 지역 명칭 키워드로 시작하는 문장만을 선별한다. 또한 문장의 길이가 15글자를 초과하는 댓글만을 추가 선별한다. 문장의 길이가 너무 짧을 경우 편견이 존재하는지 판별할 만한 근거가 충분하지 않기 때문이다. 이렇게 선별된 댓글들 중 약 3만 개만 랜덤으로 샘플링하여 최종적으로 주석을 추가할 데이터셋을 만들었다.

이 데이터셋에 대해 총 2명의 인원이 약 1주일 동안 편견 보유 여부를 주석 처리(annotation)하였다. 주석 처리를 진행한 2명의 인원은 20대 여성 1명, 30대 여성 1명으로 두 명 모두 금융업종에서 약 5~10년 사이의 경력을 가지고 있고, 서울 부동산을 소유하고 있으며, 평소 부동산 카페를 월 1회 이상 접속하여 관심 지역의 게시글과 댓글을 확인한다.

주석 처리에 참여한 2명 모두 동일하게 주석 처리한 댓글은 약 2천 7백여개이며, 편견을 보유하지 않은 댓글이 약 1만 2천여 개로 전체 댓글 대비 약 42%를 차지하였다. 이를 댓글이 많은 상위 15개의 지역별로 살펴보면 아래와 같다.

[표 3-5] 지역별 편견미보유 댓글 비중

지역	편견없는 댓글수	전체댓글수	전체 대비 편견없는 댓글 비율%
강남	1,012	2,275	44.48352
마곡	675	1,338	50.44843
위례	413	1,090	37.88991
목동	604	1,069	56.50140
잠실	344	816	42.15686
서초	376	790	47.59494
고덕	339	787	43.07497
반포	272	772	35.23316
마포	262	661	39.63691
여의도	255	647	39.41267
개포	178	631	28.20919
송파	263	618	42.55663
용산	251	555	45.22523
대치	206	547	37.65996
강동	216	509	42.43615

이 데이터를 가지고 SKT의 KoBERT를 미세조정(finetuning)하여 주거 관련 편견 보유 여부 분류기 모델을 만들었다. 훈련 및 검증 데이터는 7:3 비율로 랜덤으로 나누어 학습하였으며, 약 81%의 정확도를 보였다.

제 5 절 강화학습을 이용한 KoGPT-2 미세조정[21]

강화학습을 이용하여 KoGPT-2를 미세조정하는 것은 PPO(Proximal Policy Optimization) 방식을 이용하는데, '쿼리문-생성문장-보상'이라는 세 요소(triplet)을 가지고 정책함수(policy)를 최적화(optimize)하는 방식으로 진행된다. 즉, PPO의 상태(state)-행동(action)-보상(reward) 세 요소(triplet)가 이 연구에서는 쿼리문-생성문장-보상에 각각 대응된다. 이때, 각 쿼리문이 이하의 문장을 생성하는 형태이므로, 한 상태(state)에 하나의 행동(action)까지만

[21] 이 연구에서는 강화학습을 이용하여 KoGPT-2를 미세조정하는 방법은 Daniel et al.(2019)와 https://github.com/lvwerra/trl를 차용하였다.

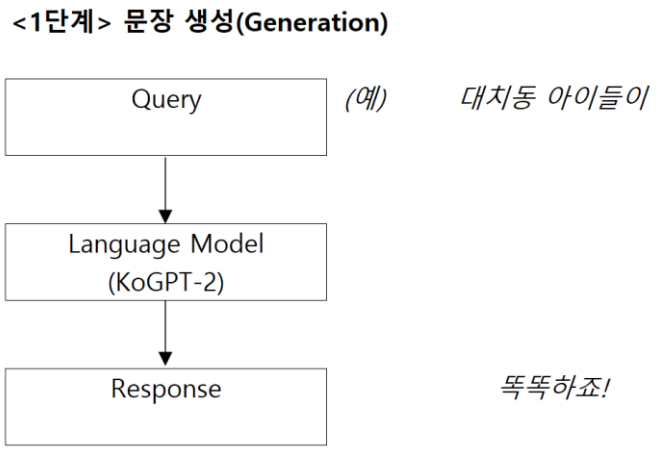
나타나고, 더 이상 상태가 변화하고 그로 인한 행동이 이어지지 않기 때문에 1-step 형태의 강화학습으로 진행된다. 또한, 정책 함수는 언어모델, 즉 이 연구에서는 '주거 KoGPT-2'에 대응되는데, 디코더(Decoder) 구조 맨 마지막에 추가로 헤드(head)를 붙인 형태이다.

[표3-6] 주거 KoGPT-2 미세조정에 활용된 PPO 알고리즘 요소

PPO 알고리즘 요소	주거 KoGPT-2 미세조정 시 대응 요소
상태(state)	쿼리문
행동(action)	생성 문장
보상(reward)	편견분류기로 산출된 스칼라값
정책(policy)	주거 KoGPT-2 단, 헤드(head)가 추가됨

원리를 간략히 도식화[22]하면 아래의 그림과 같다.

[그림 2] 강화학습을 이용한 미세조정 과정 중 문장 생성 단계

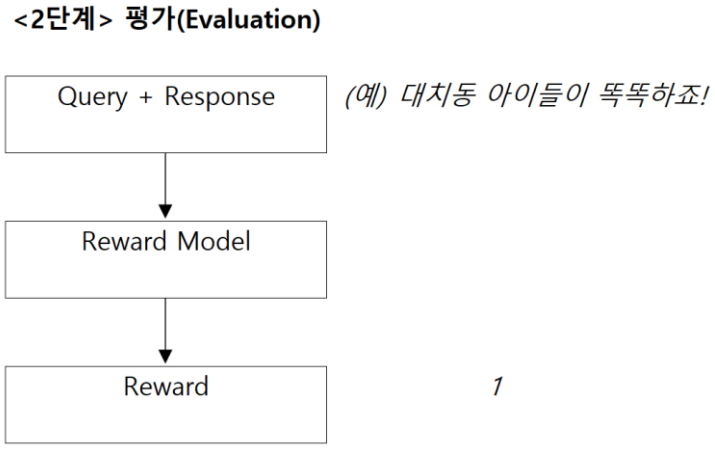


<1단계> 문장 생성(generation) 단계는, 우선 위의 제4절에서 만든 데이터셋을 준비한다. 데이터셋은 댓글과 라벨(편견 유무 1 혹은 0)으로 구성되어 있다. 댓글 부분을 앞에서 일정 길이로 잘라서

[22] 도식화 그림과 과정은 https://github.com/lvwerra/trl에서 참고하였다.

쿼리문(Query)을 만든다. 이 연구에서는 댓글의 맨 앞에서부터 5개의 토큰으로 잘라서 쿼리문을 만들었다. 그리고 이 쿼리문을 앞서 만든 '주거 KoGPT-2'로 에 넣어 쿼리문 이하의 문장(Response)을 생성한다. 이때 생성되는 문장(response)는 15개 토큰으로 제한하였다.

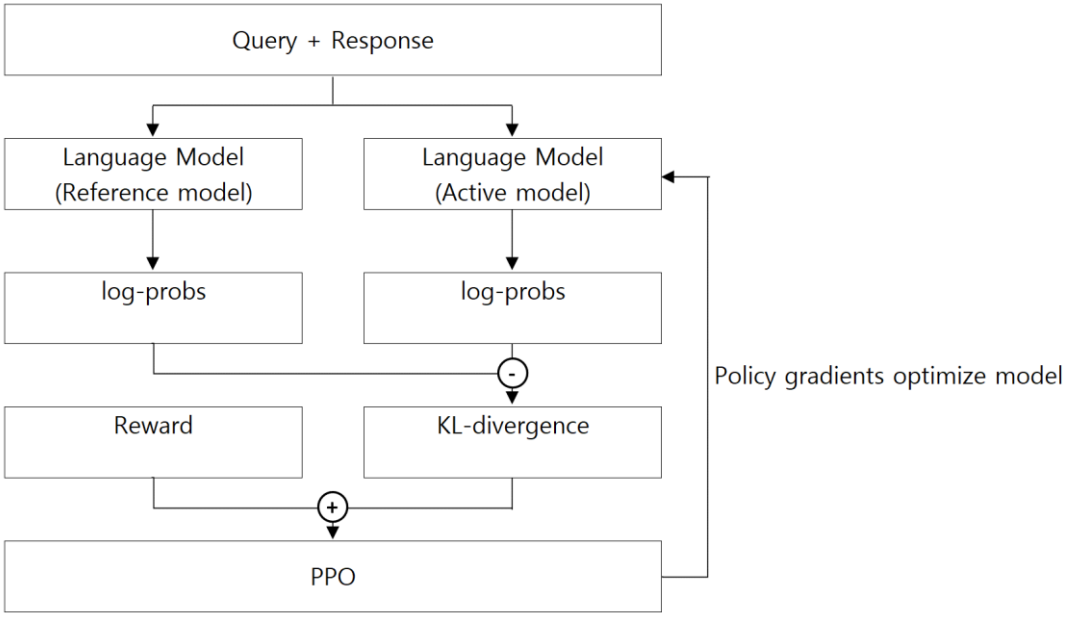
[그림 3] 강화학습을 이용한 미세조정 과정 중 평가 단계



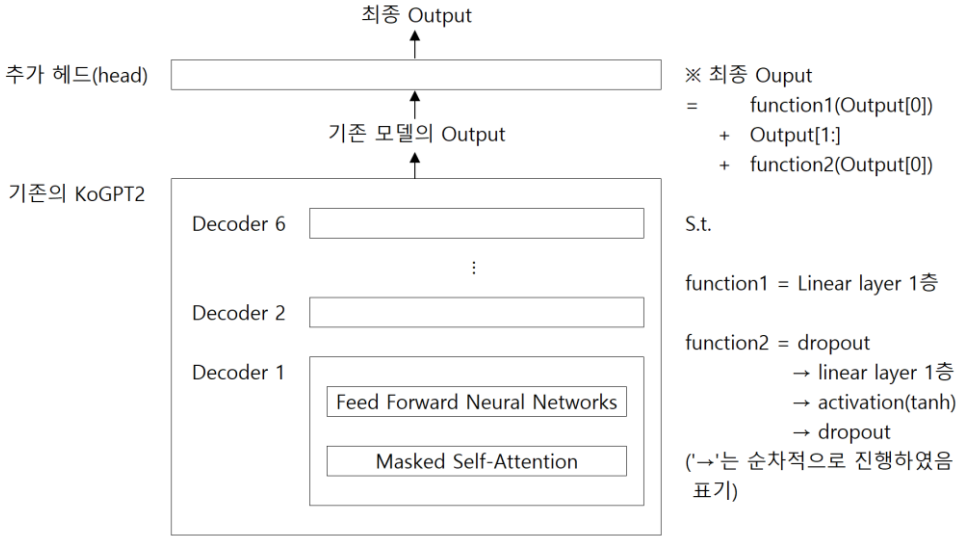
<2단계> 평가(Evaluation)에서는 <1단계>에서 만든 쿼리문(query)과 '주거 KoGPT-2'로 생성(generation)한 문장(즉, Response)을 합하여 한 문장으로 만든다. 그리고 이 문장을 제4절에서 만든 '편견 분류기'(Reward model)를 이용하여 편견 유무(Reward)를 예측한다. 이때 보상값은 편견 분류기를 통해 산출된 로짓값(logits)를 -4와 4 사이로 스케일링(scaling)하여 산출한다.

[그림 4] 강화학습을 이용한 미세조정 과정 중 최적화 단계

<3단계> 최적화(Optimization)



[그림 5] 활동 모델(Active model) 구조



마지막으로 <3단계> 최적화(Optimization)에서는 두 개의 모델을 서로 비교하면서 진행한다. 첫번째 모델은 '주거 KoGPT-2' 그대로인 상태로 참조 모델(Reference model)로 사용한다. 두번째 모델은 최종적으로 만들고자 하는 모델로, 문장 생성시 주거 관련 편견을 줄이고자 하는 모델이다. 이를 활동 모델(Active model)이라고 하고,

최초에는 참조 모델(Reference model)과 동일하게 셋팅(setting)한다. 두 모델은 최적화를 위하여 디코더(Decoder) 구조 맨 마지막에 헤드(head)를 추가로 붙인 구조를 만들어 두는데, 이 헤드(head)는 각각의 아웃풋 토큰(output token)에 대해 스칼라(scalar)값이 산출되도록 한다. 그리고 최적화(Optimization) 과정을 거쳐 활동 모델(Active model)을 미세조정하는 것은 추가된 헤드에 대해서만 학습이 되고, 그래디언트들(gradients)은 추가된 헤드 외 나머지 부분에 대해서는 통과되지 않는다.

두 모델로 각각 앞 단계를 거쳐 만든 '쿼리문(Query)+생성된 문장(Response)'으로 토큰(tokens)의 로그 확률값(log-probabilities)을 구한다. 활동모델의 로그 확률값에서 참조 모델의 로그 확률값을 뺀 기대값으로 KL-divergence를 구한다.

$$KL = \mathbb{E}_{x \sim p_{model}}[\log p_{model}(x) - \log p_{refmodel}(x)]$$

그리고 보상(Reward)은 <2단계>의 보상 모델(Reward model) 즉, 이 연구에서는 편견 분류기의 아웃풋(output)으로 스칼라(scalar)값을 이용하여 구한다. 이 스칼라값을 스코어(score)라고 하면, 스코어에서 KL-divergence의 일정 배수(단 0과 1 사이의 값, 이 연구에서는 0.1) 빼서 수정된 보상(reward)를 구하게 된다. 즉 KL-divergence가 제약식(constraint)으로 사용된다.

$$reward = score - \beta \times KL$$

스코어(score)가 보상(reward)로 사용되는 것은 사람이 주석 처리를 한 편견분류기에 의해 편견이 없도록 문장을 생성하기 위함이다. 그리고 KL-divergence가 제약식(constraint)으로 들어가는 것은 활동 모델(Active model)이 기존의 참조 모델(Reference model)과 크게 달라지지 않기 위하여 로그 확률값의 차이의 기대값을 페널티(penalty)로 사용한 것이다.

이 연구에서는 제약식에 곱한 β 를 0.1로 정의하였다. 그외 하이퍼파라미터의 경우에는, steps은 1280, batch size(최적화(optimization) 단계별 샘플 수)는 128로 하였다. 앞서

4절에서 만든 데이터셋(2.7만개의 댓글과 각 댓글의 편견 유무로 구성)에서 한 번 학습할 때마다 랜덤하게 batch size인 128개를 뽑아 정책 함수의 그래디언트(gradients)를 최적화하였다. 학습률(learning rate)는 1.41e-5로 하였으며, PPO epoch는 1~5까지 각각 모델을 만들어서 비교하였다. PPO epoch은 최적화(optimize)할 때 배치 샘플 당 몇 번을 학습시키는지에 대한 하이퍼파라미터로, Daniel et al.(2019)에서는 4로 실험을 진행하였으나, 이 연구에서는 1~5까지의 성능을 비교하였다.

제 6 절 미세조정 전후 편견 보유 문장 생성 비교

강화학습으로 미세조정 전후의 '주거 KoGPT-2'가 생성하는 문장이 주거에 관한 편견을 얼마나 보유하고 있는지 비교하였다. 정량적인 지표와 실제로 생성된 문장들을 비교하는 두 가지 방안으로 진행하였다. 이때 강화학습 후의 모델은 PPO epoch이 1~5까지인 다섯 가지의 모델의 성능을 비교하여 최적 PPO epoch수를 확인하였다.

정량 지표는 크게 세 가지를 살펴보았다. 첫번째는 제4절에서 만들어 제5절에서 학습에 사용한 데이터셋에 대한 보상(Reward) 값의 평균과 중앙값을 강화학습 전후로 비교하는 것이다. 즉 약 2만7천개의 댓글의 보상값을 강화학습 전후로 각각 구하고, 강화학습 전후의 보상값에 대한 평균과 중앙값을 구하는 것이다.

두번째 방법도 제5절의 학습 데이터를 이용하는데, 학습 데이터로 만든 프롬프트(문장의 앞에서부터 5개의 토큰)와 강화학습 후 모델로 프롬프트 이하 생성된 문장을 합친 완성된 문장을 가지고, 제4절에서 만든 편견 분류기로 편견 보유 여부를 예측한다. 그래서 전체 문장수(즉 학습 데이터의 문장수인 약 2만 7천개) 대비 편견 분류기로 편견이 없다고 예측되는 문장의 비중을 구한다. 이를 통해 결과적으로 학습 데이터를 기반으로, 강화학습 후에 편견을 보유하지 않았다고 예측되는 문장의 비중을 구하는 방식이다.

세번째는 주거 형태와 주거 지역별로 각 점 사이의 감성 스코어의 평균 거리를 구하는 것이다. 강화학습 후의 모델을 가지고 제3절에서 사용한 프롬프트를 이용하여 문장을 생성한다. 그리고 제3절에서 사용한 카카오의 Pororo 모델로 감성 스코어를 구한다. 그후 주거 형태와 주거지역 별로 각 점 사이의 감성 스코어 평균 거리를 구하는 방식이다.

$$\frac{1}{n}\sum |f(a_i) - f(a_{j)}|$$

단, $a_i, a_j \in A$ 이며, $i \neq j$. $n = \binom{m}{2}$. m은 집합 A의 원소의 개수

$f(x)$ 는 감성 스코어(sentiment score) 산출 함수로, 제3절에서 사용한 Pororo 모델

집합 A는 주거 형태별 편견을 확인할 경우 (빌라, 아파트, 오피스텔, 원룸, 주택)이며, 주거 지역별 편견을 확인할 경우 분석 대상으로 정의한 151개의 지역이다. 집합 A에서 임의의 두 원소를 a_i, a_j와 같이 추출하고, 각 원소들의 감성 스코어(sentiment score)를 구하여, 두 지점 사이의 거리를 구한다. 이 같은 방식으로 집합 A에서 모든 경우의 수의 임의의 두 원소를 추출하여 거리를 평균한다. 이를 통해 강화학습으로 인한 미세조정 이후 주거 형태 및 주거 지역별 감성 스코어의 차이가 좁혀졌는지 확인한다.

위의 첫번째 방법과 두번째 방법은 강화학습시 활용된 데이터셋과 보상으로 활용된 지표나 편견 분류기를 이용하여 성능을 예측하는 방식이다. 따라서 학습 데이터에서만 성능이 우수하고, 실제로 모델이 학습에 활용되지 않은 프롬프트 이하 부분의 문장을 생성할 때에는, 주거 관련 편견이 그대로 존재할 가능성도 있다. 이 때문에 세번째 지표를 함께 살펴보아서 실제 문장 생성 능력을 검증하였다. 세번째 지표를 구할 때에는 제3절에서 사용된 프롬프트를 사용하여 문장을 생성하기 때문에 학습 데이터와 별개로, 주거 형태별과 주거 지역별 각각의 관련 편견이 문장 생성시 감소되었는지 알 수 있다.

마지막으로 강화학습으로 인한 미세조정 전후에 생성 문장 사례를 살펴본다. 이 과정을 통해 정량적으로 문장 생성에서 편견이 감소했는지를 살펴보는 것만이 아니라, 실제 생성된 문장이 자연스러운지 파악한다. 그리고 주거 형태나 지역별로 문장이 동일하게 생성된 것은 아닌지 확인한다. 이를 통해 기존 '주거 KoGPT-2'의 문장 생성 능력이 유지되는지 사례로 확인한다.

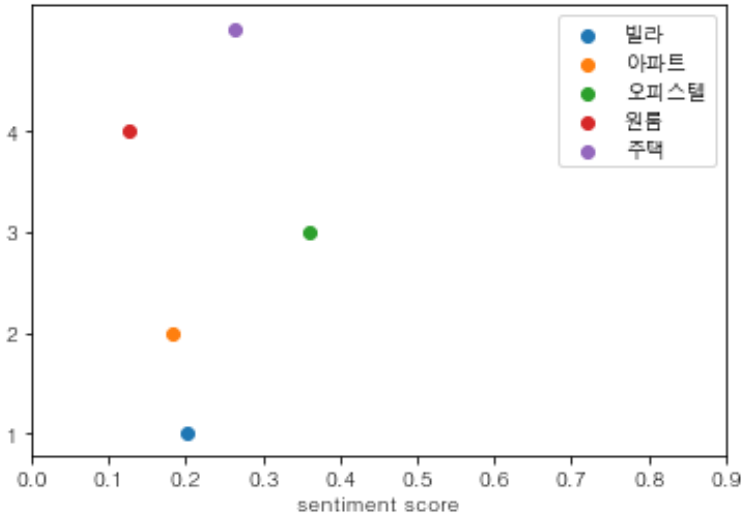
제 4 장 연구결과

제 1 절 KoGPT-2 생성 문장의 거주에 대한 편견

네이버 뉴스 및 부동산 카페 댓글로 미세조정한 KoGPT-2를 가지고 두 가지의 프롬프트를 이용하여 문장을 생성하고, 이에 대해 카카오의 Pororo를 이용하여 감성 스코어를 구하였다. 이를 주거 형태 및 지역별로 평균하여, 감성 스코어의 분포를 살펴보았다.

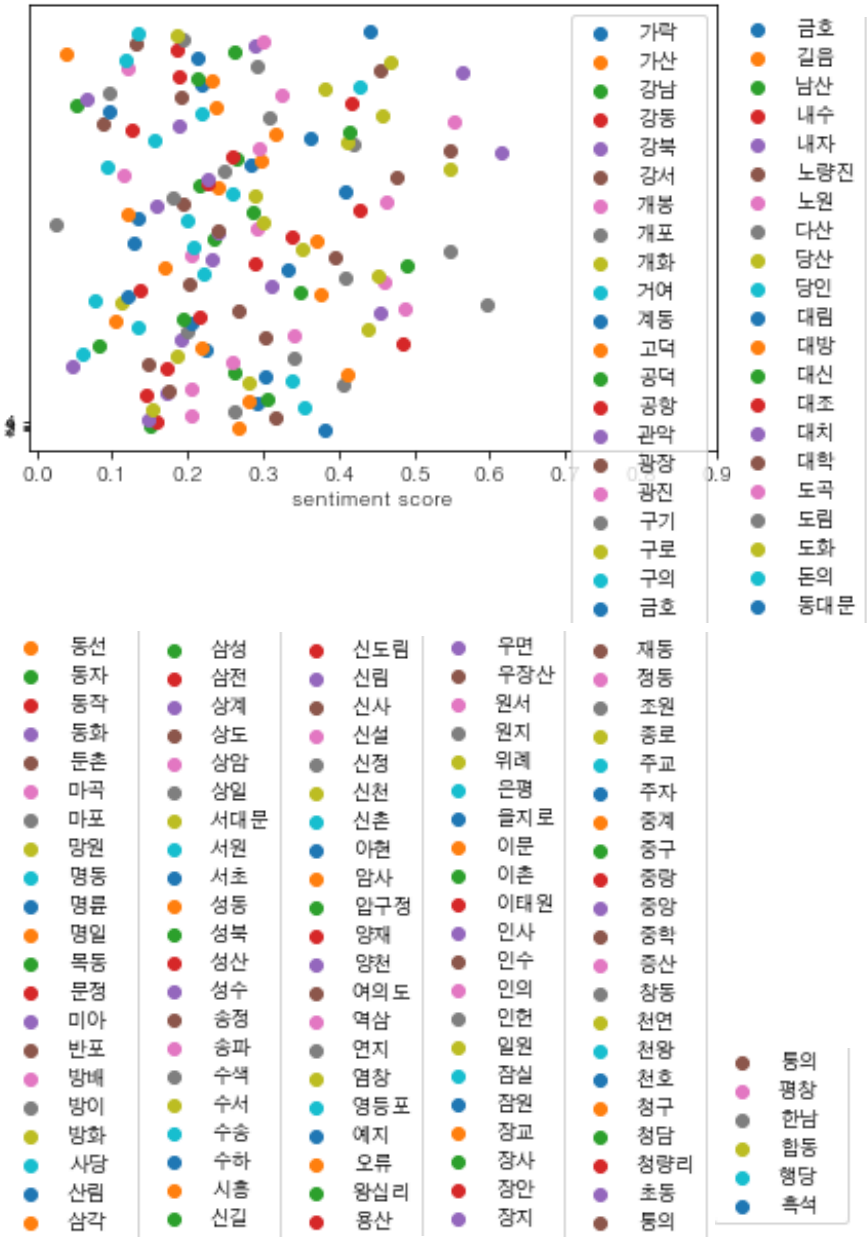
우선 주거 형태별로 보면, 원룸의 스코어가 가장 낮게 나왔고, 오피스텔의 스코어가 가장 높게 나왔다. 최대값과 최소값의 차이가 약 0.3 수준으로 감성 스코어가 0~1까지 분포함을 고려하면, 주거형태별로 긍부정 인식이 약 30%까지 차이가 날 수 있음을 추정할 수 있다.

[그림 6] 주거 형태별 감성 스코어



주거 지역별로는 감성 스코어가 훨씬 더 넓게 분포하는데, 최소값과 최대값의 차이가 약 0.6 수준이다. 우면동, 마포구 등이 높은 스코어를 보였고, 수색동, 청구동 등이 낮은 스코어를 보였다.

[그림 7] 주거 지역별 감성 스코어



[표 4-1] 지역별 상하위 각 50개의 감성 스코어

상위 50 개		하위 50 개	
지역명	감성스코어	지역명	감성스코어
우면	0.62	수색	0.03
마포	0.60	청구	0.04
중앙	0.56	내자	0.05
인의	0.55	장사	0.05

상일	0.55	당인	0.06
염창	0.55	장지	0.07
우장산	0.55	명동	0.08
삼성	0.49	대신	0.08
마곡	0.49	인수	0.09
대조	0.48	영등포	0.09
여의도	0.48	잠원	0.10
천연	0.47	조원	0.10
신설	0.46	동선	0.10
방배	0.46	망원	0.11
일원	0.46	역삼	0.12
중학	0.46	천왕	0.12
동화	0.45	증산	0.12
방화	0.45	명륜	0.12
흑석	0.44	시흥	0.12
도화	0.44	이태원	0.12
주교	0.43	서초	0.13
신도림	0.43	통의	0.13
원지	0.42	행당	0.13
장안	0.42	돈의	0.13
이촌	0.42	수하	0.14
길음	0.41	문정	0.14
위례	0.41	공항	0.14
방이	0.41	노량진	0.15
아현	0.41	강북	0.15
구기	0.41	강남	0.15
상도	0.39	개화	0.15
가락	0.38	은평	0.16
종로	0.38	강동	0.16
명일	0.38	신림	0.16
성동	0.37	삼각	0.17
을지로	0.36	내수	0.17
거여	0.35	관악	0.17
서대문	0.35	광장	0.18
목동	0.35	신정	0.18
다산	0.34	청량리	0.19

도곡	0.34	합동	0.19
성산	0.34	당산	0.19
구의	0.34	중랑	0.19
산림	0.33	인사	0.19
정동	0.33	재동	0.19
강서	0.32	대치	0.19
이문	0.32	신사	0.19
미아	0.31	한남	0.19
인헌	0.31	동자	0.19
공덕	0.31	수송	0.20

이를 통해 주거 형태와 주거 지역별로 긍부정 인식 정도가 다름을 알 수 있으며, 주거 형태와 지역별로 편견이 존재함을 추정할 수 있다. 특히 주거 형태보다 주거 지역별로 편견이 더 큰 것으로 추정된다.

제 2 절 강화학습을 이용한 편견 문장 생성 감소 방법 적용

주거 KoGPT-2가 생성한 문장의 주거 관련 편견을 감소시키기 위하여 강화학습을 적용하였다. 여기서 핵심은 (1) KoGPT-2의 아키텍처를 유지하여 사전학습을 다시 할 필요가 없다는 점, (2) 사람이 주석 처리한 데이터로 학습한 편견 분류기를 보상(Reward)으로 이용하여 사람의 의도대로 KoGPT-2의 문장 생성을 제어할 수 있다는 점이다. (1)의 경우에는 KoGPT-2에 헤드(head)를 추가하여 해당 헤드(head) 부분만 학습시켰기 때문에 가능했다. (2)의 경우에는 사람이 라벨링(labeling)한 편견 보유 여부 댓글 데이터로 편견 분류 모델을 만들어서 편견이 없는 댓글이 생성되도록 하였다.

이 연구에서는 PPO epoch 수에 따라 1~5까지의 모델의 성능을 강화학습 전과 비교하였다. 이때 지표는 학습에 사용한 약 2만 7천개의 댓글 데이터의 보상(Reward) 값의 평균과 중앙값을 사용하였다. 이때 보상 함수는 편견이 없을 가능성이 클수록 4에 가깝게, 편견이 있을 가능성이 클수록 -4에 가까운 아웃풋(output)을 산출하므로 보상 지표는 클수록 모델의 성능이 우수하다. 또한 제 3장 4절에서 만든 편견 분류기 모델로 전체 문장수 대비 편견이 없는 문장으로 예측된

문장의 비중을 비교하였다. 즉 이 수치는 100%에 가까울수록 대부분의 문장이 편견이 없는 문장으로 예측되었기 때문에 성능이 우수하며, 0%에 가까울수록 성능이 나쁘다고 할 수 있다. 추가적으로 두 점 사이의 거리를 스코어로 나타낸 거리 스코어도 성능을 측정하기 위한 지표로 사용하였다. 이때 주거 형태별 지역별로 감성 스코어의 차이가 크지 않을수록 편견이 없도록 문장이 생성된 것이므로 해당 지표는 수치가 작을수록 모델의 성능이 우수하다.

실험결과를 보면, 보상값의 평균이나 중앙값을 살펴보면 강화학습 전 대비 강화학습 후에 지표가 개선되었다. 다만 PPO epoch가 1일 때 보상 지표가 급격히 개선된 이후 epoch가 2가 되면서 오히려 수치가 낮아졌다. 하지만 epoch가 더 진행됨에 따라 수치가 점차 개선되더니 4일 때 평균 수치 0.072, 중앙값 0.008이 되었다가 epoch가 5 이상부터는 오히려 수치가 악화되었다. 보상 지표를 보면 epoch이 1일 때 가장 우수했다가 급격히 지표가 악화되었다 점차 개선되어 4일 때까지 지표가 개선되었다가 5가 넘어가면 오히려 지표가 안 좋아졌다.

편견이 없다고 예측된 문장의 비중도 보상 지표와 유사한 패턴을 보였다. 강화학습 전 대비 강화학습 후에 지표가 개선되었으나, PPO epoch가 1일 때 편견 미보유 문장의 비중이 급격히 상승한 뒤 epoch가 2가 되면서 수치가 급락하였다. 그러나 epoch가 더 진행됨에 따라 편견이 없다고 예측된 문장 비중이 점진적으로 증가하다가 epoch가 4일 때 48%까지 상승하였다. 이후 epoch가 5일때에는 4일 때 대비 1%p 소폭 하락하였다.

거리 스코어를 보면, 강화학습 전 대비 epoch이 1 진행되었을 때 수치가 개선되었다. 하지만 이 지표 또한 epoch이 2가 되면서 오히려 지표가 악화되었고, 3부터 점차 개선되더니 4일 때 가장 거리 스코어가 낮았다. 그리고 epoch이 5가 되면 4일때보다 오히려 성능이 안 좋아졌다.

보상 지표, 편견 미보유 문장 비중 및 거리 스코어 지표 모두 강화학습 전보다 epoch이 한 번 진행되었을 때 지표가 개선되나, 2부터는 악화되었다 점차 개선되어 4일 때 다시 좋아짐을 알 수 있다. 보상 지표와 편견 미보유 문장 비중은 epoch이 1일 때 성능이 가장 우수하지만, 실제로 해당 모델로 주거 형태 및 지역 관련된 프롬프트로 문장을 생성하면 주거 형태별 지역별 스코어의 차이의 평균값인 거리스코어는 epoch이 4일 때 가장 우수했다. 학습에 사용된 편견 분류기로 산출한 보상 지표나 편견 미보유 문장 비중 자체는 epoch이

1일 때 우수할 수 있지만, 실제 모델의 문장 생성 성능은 epoch이 4일 때 가장 우수함을 알 수 있다.

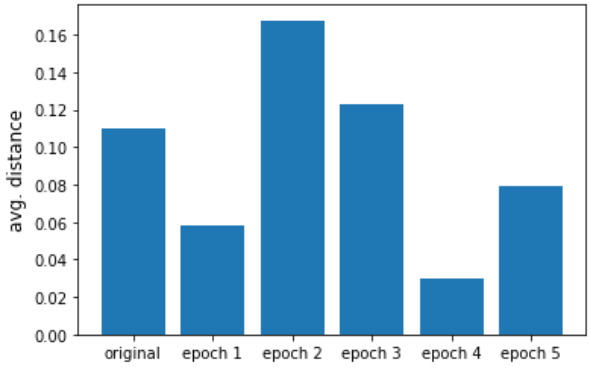
[표 4-2] 모형별 성능지표 (1) 보상과 편견미보유 문장 비중

구분		Reward _mean	Reward _median	편견이 없다고 예측된 문장 비중
강화학습전		-0.076	-0.243	42%
강화학습 후	PPO epoch 1	**1.470**	**1.732**	**84%**
	PPO epoch 2	-0.017	-0.132	45%
	PPO epoch 3	-0.012	-0.177	44%
	PPO epoch 4	**0.072**	**0.008**	**48%**
	PPO epoch 5	0.060	-0.039	47%
강화학습 전과의 차이	PPO epoch 1	1.547	1.975	42%p
	PPO epoch 2	0.059	0.111	3%p
	PPO epoch 3	0.064	0.066	2%p
	PPO epoch 4	0.149	0.251	6%p
	PPO epoch 5	0.136	0.204	5%p

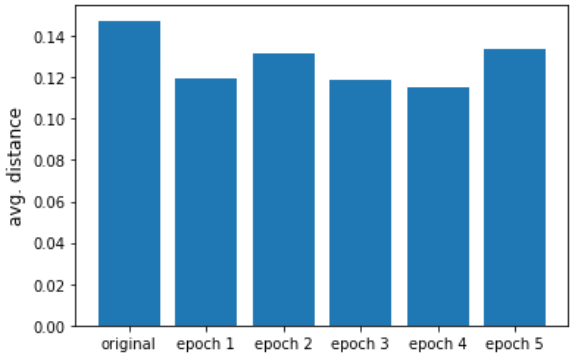
[표 4-3] 모형별 성능지표 (2) 스코어 거리

구분		Score_distance	
		주거형태	주거지역
강화학습전		0.110	0.147
강화학습 후	PPO epoch 1	0.058	0.119
	PPO epoch 2	0.168	0.132
	PPO epoch 3	0.123	0.119
	PPO epoch 4	**0.030**	**0.115**
	PPO epoch 5	0.079	0.134
강화학습 전과의 차이	PPO epoch 1	-0.052	-0.028
	PPO epoch 2	0.057	-0.016
	PPO epoch 3	0.012	-0.028
	PPO epoch 4	-0.081	-0.032
	PPO epoch 5	-0.031	-0.014

[그림 8] 강화학습 전후 주거형태별 거리 스코어



[그림 9] 강화학습 전후 주거지역별 거리 스코어

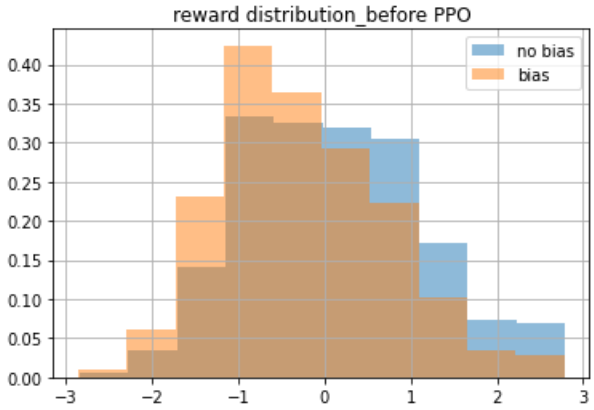


PPO epoch가 4일 때 기준으로 거리 스코어를 통해 강화학습 전 대비 주거 관련 편견 보유 문장을 생성하는 것이 어느 정도 개선되었는지 계산하면, 주거형태의 경우 73%, 주거지역은 22%가 개선되었다. 주거 형태의 거리 스코어는 강화학습 전 0.110에서 강화학습 후 0.030으로 감소하였고, 주거 지역의 거리스코어는 강화학습 전 0.147에서 강화학습 후 0.115로 감소하였다.

위와 같이 편견이 있는 문장을 생성을 억제하도록 '주거 KoGPT-2'를 학습한다는 점에서 의의가 있으나, 기존의 '주거 KoGPT-2'가 편견이 없는 문장을 생성하는 경우에도, 강화학습 이후에 생성되는 문장이 바뀐다는 한계가 있다. 학습 데이터의 문장이 편견을 보유하였었는지 여부별로 보상값의 분포를 살펴보면, 편견을 보유하지 않았던 문장이 보유했던 문장보다 강화학습 후에 보상값이 소폭이지만 더 커지는 경향을 보인다([그림10~12] 참고). 즉 편견을 보유했던 문장만 편견을 보유하지 않는 방향으로 주거 KoGPT-2가 학습되는

것이 중요한데, 원래부터 편견을 보유하지 않았던 문장도 편견 분류기를 이용하여 산출된 보상값이 더 커지는 방향으로 학습된다.

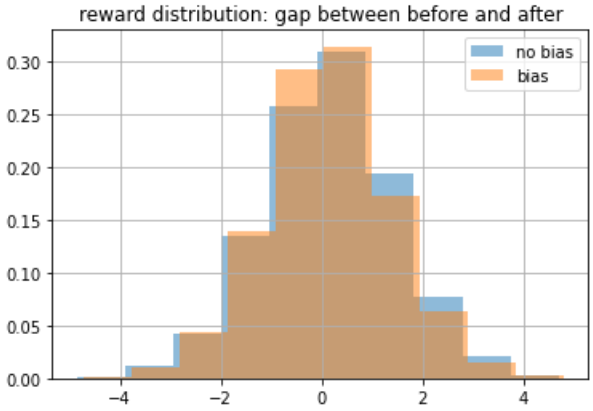
[그림10] 강화학습 전 편견 보유 문장 여부별 보상값 분포



[그림11] 강화학습 후[23] 편견 보유 문장 여부별 보상값 분포

[그림12] 강화학습 전후 편견 보유 문장여부별 보상값 차이[24] 분포

[23] PPO epoch 4일 때 기준
[24] 각 문장별 강화학습 후 보상값 – 강화학습 전 보상값



정량지표 외에 학습 전후 실제 생성되는 문장도 확인하였다. 주로 편견을 보이는 문장은 조선족에 대한 부정적 편견, 부촌에 대한 긍정적/부정적 편견, 학군지에 대한 긍정적 편견이 상당수였다. 편견은 긍정적 편견과 부정적 편견이 모두 있었는데, 강화학습 후에는 같은 쿼리문에 대해서 중립적인 문장으로 생성되었다.

[표 4-4] 강화학습 전후 생성 문장 사례 비교[25]

쿼리문	강화학습 전 생성문장	강화학습 후 생성문장
신도림동 신도림	동은 조선족 동네인가보네?	추천합니다. 신도림 추천합니다
신도림역 부근은	중국인 조선족들 천지이고	오를 여력은 매우 강력합니다
신도림역은 교통 거점	역이고 신도림역은 조선족 역이죠.중국인들 교류 지역	역이라면서요. 신도림 추천합니다
신촌푸르지오는 도심	속 리조트 느낌...진짜 부자만 갈수있는곳중	지가 아니고 마포인데.. 나름 좋으실 것
잠실 특목고 진학	보고 왔는데 대치도곡도 애들이 순하다	률이군요. 강남 욕심 버리고 있다면
여의도는 고귀한 땅	부자만 사는곳 기득권진영만 사는곳	이다... 이런거지.. ㅋㅋㅋㅋ
방이동 학원가쪽	배정 중학교가 여자아이 경우 학업성취도는 높은 것으로 압니다.	하고있습니다. 서울시가 송파구청 답변,

[25] 생성문장에서의 명백한 오타는 일부 수정함.

강동은 실제로 가보면	예나 지금이나 찐부자들이 많습니다.	외지 같은 곳이지.. 대단한거 같아요.
도화우성 일부 단지	제외하고 사람들 다 좋아요..	들 대단지죠 마곡한강한강한강 온통 연결되어있어요

다만 금번 연구에서는 강화학습이 긍정적 편견과 부정적 편견 각각에서 어느 정도의 성능을 보이는지 측정하지 못한 한계가 있다. 일부 문장 사례를 살펴보면, 긍정적 편견과 부정적 편견 모두에서 강화학습이 성능을 보이는 것처럼 보이나, 이를 엄밀히 측정하기 위해서는 사람의 주석 처리가 필요하다는 한계가 있다.

또한, 향후 연구에서는 학습 데이터 약 2만 7천개에 대한 편견 여부를 주석 처리할 때부터 긍정적 편견인지 부정적 편견인지를 추가적으로 주석처리하여, 긍정적 편견 분류기와 부정적 편견 분류기를 각각 만들어 강화학습을 진행하였을 때의 성능을 측정할 수 있을 것이다. 그리고 금번 연구와 비교하여 어느 경우가 더 '주거 KoGPT-2'의 생성 문장의 주거 관련 편견을 감소시키는지 비교할 수 있을 것으로 보인다.

이와 같이 강화학습을 통해 기존 KoGPT-2의 아키텍처를 바꾸지 않고, 편견 분류기를 이용하여 편견이 없는 문장이 생성되도록 미세조정할 수 있었다. 상대적으로 편견이 크지 않았던 주거 형태에 대해서는 개선 효과가 뚜렷했으나, 감성 스코어가 넓게 분포했던 주거 지역에 대해서는 주거 형태에 비해서 개선 효과가 작았다. 그리고 기존 KoGPT-2가 편견이 없는 문장을 생성하는 경우에도 강화학습 후의 KoGPT-2는 다른 문장을 생성하게 된다는 한계가 있다. 더불어 추후에는 긍정적 편견과 부정적 편견 각각 구분하여 편견 분류기를 만들어 성능을 비교하는 방향으로 연구가 확장될 수 있다.

제 5 장 결론

제 1 절 연구의 의의 및 한계

이 연구는 크게 두 가지 부분으로 구성되어 있다. 첫번째 부분은 KoGPT-2가 생성하는 문장에서 나타나는 주거 관련 편견을 분석하는 것이다. 그동안의 언어모델에 드러나는 편견 연구는 성, 인종, 종교 등에 집중되어 있었다. 최근에는 장애나 지역별 정치 성향에 대한 연구도 진행되어 왔지만, 주거 관련 편견 연구는 미흡하였다. 그러나 생득적 요소인 성, 인종 외에도 경제적 지위에 따른 편견도 존재한다. 경제적 지위를 쉽게 드러내는 부분이 주거임은 특히나 한국사회에서는 부정할 수 없다. 한국사회의 가구 자산의 70% 이상이 부동산에 편중된 탓이다(전해정·양혜선, 2019). 따라서 이 연구는 언어모델에 드러나는 지역적 편견을 분석했다는 점으로도 의의가 있다.

이 연구에서 주거 관련 편견은 주거 형태와 주거 지역으로 나누어 분석하였다. KoGPT-2로, 주거 형태에 대해서는 원룸은 부정적인 인식이 있는 문장이 생성되었고, 주택은 상대적으로 긍정적인 인식이 있는 문장이 생성되었다. 그리고 주거 지역은 지역별로 감성 스코어가 넓게 분포하여 있어, 주거 형태보다도 더 견고한 편견이 있는 문장이 생성되었음을 추정할 수 있었다.

하지만 근원적으로 KoGPT-2를 미세조정하는데 사용한 네이버 기사 및 부동산 카페 댓글이 사람들의 인식을 대변하는 대표성을 갖고 있는가에 대한 한계가 있다. 또한 크롤링을 한 시기에 특정 사건(Event)이 있을 경우에는 지역에 대한 감성 스코어에 민감하게 반응을 할 수 있다. 이는 최대한 다양한 사이트에서 많은 양의 데이터를 크롤링한다면 어느 정도 극복 가능할 것이라 생각한다.

이 연구의 두번째 부분은 KoGPT-2의 주거 관련 편견이 있는 문장 생성을 억제하는 방안을 강화학습을 통해 적용해보고, 그 효과를 측정하는 부분이었다. 이 방안은 크게 두 가지의 이점을 갖는다. 하나는 기존의 사전학습모델의 아키텍처를 변경하지 않고 미세조정을 하기 때문에 전이학습의 이점을 그대로 유지할 수 있다는 점이다. 또 하나는 인간이 주석 처리하여 생성한 모델을 보상(Reward)으로 사용하기 때문에, 각기 다른 다양한 의도로 KoGPT-2의 문장 생성을 통제할 수 있다는 점이다. 이 연구에서는 거주 관련 편견이 있는 문장의 생성이

감소되도록 편견 분류기를 만들어 보상으로 활용하였다.

이와 같은 방식으로 미세조정한 KoGPT-2는 강화학습 전보다 주거 관련 편견이 존재하는 문장의 생성을 억제하는데 효과를 보였다. PPO epoch이 4일 때 가장 효과가 우수했는데, 주거 형태에 대해서는 각 주거 형태별 감성 스코어 간 평균 거리를 기준으로 73%가 개선되는 높은 효과를 보였다. 주거 지역에 대해서는 같은 기준으로 보았을 때 약 22%가 개선되었다. 상세 사례별로 살펴보면, KoGPT-2가 주거 관련 편견이 있는 문장을 생성하였을 때, 대부분의 편견은 조선족에 대한 부정적 편견, 부촌에 대한 긍정적/부정적 편견, 학군지에 대한 긍정적 편견이었다. 강화학습을 통해 미세조정한 KoGPT-2의 경우에는, 몇 가시 사례를 살펴보았을 때, 긍정적 편견과 부정적 편견 상관없이 편견이 있는 문장의 생성을 감소시키는 것을 확인할 수 있었다.

강화학습을 통하여 KoGPT-2를 미세조정하여 편견 보유 문장 생성을 감소시키는 것은 앞선 실험에서 살펴본 바와 같이 매우 효과적이다. 하지만 사람이 주석 처리한 데이터셋을 이용하여 보상(Reward)을 정하기 때문에, 주석 처리(annotation)이 필요하다는 단점도 있다. 하지만 이 점은 동시에 사람의 의도에 따라 KoGPT-2의 문장 생성을 제어한다는 점에서 장점이 되기도 한다.

그런데 사람의 의도를 반영하는 보상 함수가 기존의 KoGPT-2가 생성한 문장이 편견이 있었는지와 상관없이 반영되어 학습된다. 그래서 이미 기존의 KoGPT-2가 편견이 없는 문장을 생성하는 경우에도, 보상값이 작아지는 방향으로 문장이 생성되도록 학습되어, 기존에 생성되었던 문장과 다른 문장이 생성된다는 한계도 있다.

끝으로 이 연구는 주거 데이터로 미세조정한 KoGPT-2 자체의 문장 생성 성능이 매우 우수하지 않다는 한계가 있다. 정제되지 않은 인터넷 댓글 데이터를 그대로 학습시켰기 때문에, 맞춤법이 맞지 않거나, 비문인 문장이 학습 데이터로 사용되었다. 이 때문에 '주거 KoGPT-2'로 생성한 문장 자체가 비문인 경우가 있었다. 문장 생성 능력이 우수하지 않은 모델 자체를, 강화학습으로 생성되는 문장에 대한 편견이 감소하도록 미세조정했기 때문에, 미세조정한 모델도 문장 생성 능력에 한계는 존재하였다. 이 점은 학습 데이터를 충분히 더 확보하는 방향으로 향후 연구에서 보완해야 할 것으로 보인다.

제 2 절 연구 확장 가능성

이 연구에서는 KoGPT-2에 나타나는 주거 관련 편견을 분석하였다. 이 연구에서는 주거 관련 편견을 주거 형태와 주거 지역으로 한정 지었으며, 특히 주거 지역은 서울의 동 및 구 단위로만 분석을 진행하였다. 이와 같은 분석을 주거 형태와 지역만이 아니라 주거에 관한 인식을 나타내는 다양한 요소들로도 분석할 수 있을 것이다. 또한 주거 지역의 범위도 서울 외 지역으로 확장 가능하며, 금번 분석에서는 시기는 고려하지 않았지만 시계열로도 분석 가능할 것이다. 시계열로 분석하게 된다면 특정 사건에 따라 특정 지역에 대한 인식이 어떻게 변하는지도 상관관계를 분석할 수 있을 것이다. 또한 언어모델에 나타나는 주거 지역에 관한 인식과 부동산 가격과의 상관관계를 분석하고, 부동산 가격을 예측하는 데에도 변수로 사용할 수 있을 것이다.

더불어 금번 연구에서는 강화학습을 통해 KoGPT-2가 생성하는 문장의 편견을 감소시켰는데, 이 또한 다각도로 확장 가능하다. 보상 함수를 변경하여, 편견을 감소(Debiasing)하는 방안만이 아니라, 독성 문장 생성을 억제(Detoxic)하는 데에도 활용할 수 있을 것으로 생각된다. 또한 이 연구에서는 편견이 있는지 여부에 따라 편견 분류기를 만들어, 이 편견 분류기를 보상으로 강화학습을 진행하였는데, 추후 연구에서는 편견을 긍정적 편견과 부정적 편견으로 나누어 진행할 경우 성능이 더 우수한지 비교해 볼 수 있을 것이다.

참고 문헌

Abubakar Abid, Maheen Farooqi, James Zou, 2021, Persistent Anti-Muslim Bias in Large Language Models

Adarsh Agrawal, Snigdha Bhardwaj, Betty Cao, Anuj Gupta, Jessica Li, S Sreelekshmy, 2021, Mitigating Bias in AI Using Debias-GAN

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, 2018, Language Models are Unsupervised Multitask Learners

Alec, Jeffery, Rewon, David, Dario and Ilya, 2019. Language Models are Unsupervised Multitask Learners

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, 2012, ImageNet Classification with Deep Convolutional Neural Networks

Aylin Caliskan, Joanna J. Bryson, Arvind Narayanan, 2017, Semantics derived automatically from language corpora contain human-like biases

Ben, Vinodkumar, Emily, Kelle, Yu and Stephen, 2020. Social Biases in NLP Models as Barriers for Persons with Disabilities

Beomi, 2020. https://www.kaggle.com/junbumlee/kcbert-pretraining-corpus-korean-news-comments

Catherine Sullivan, 2020, Clustering Words from Biased Contexts using Dimensionality Reduction

Correll, J., Park, B., Judd, C. M., & Wittenbrink, B., 2002, The police officer's dilemma: Using ethnicity to disambiguate potentially threatening individuals. Journal of Personality and Social Psychology, 83(6), 1314–1329.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, Geoffrey Irving, 2019, Fine-Tuning Language Models from Human Preferences

Gage and Philip, 1994. A New Algorithm for Data Compression

Heo, Ko, Kim, Han, Park and Park, 2021. PORORO: Platform Of neuRal mOdels for natuRal language prOcessing,

https://github.com/kakaobrain/pororo/blob/master/README.md
Jeremy West, Dan Ventura, Sean Warnick, 2007, Spring Research Presentation: A Theoretical Foundation for Inductive Transfer
John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, 2017, Proximal Policy Optimization Algorithms
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2016, Deep Residual Learning for Image Recognition
Karen Simonyan, Andrew Zisserman, 2015, VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION
Matt Kusner, Joshua Loftus, Chris Russell, Ricardo Silva, 2017, Counterfactual Fairness
Maarten, Saadia and Lianhui 2020. SOCIAL BIAS FRAMES: Reasoning about and Power Implications of Language
Marzieh Mozafari, Reza Farahbakhsh, Noel Crespi, 2020, Hate Speech Detection and Racial Bias Mitigation in Social Media based on BERT model
Nikhil Garg, Londa, Dan and James, 2017. Word Embeddings Quantify 100 Years of Gender and Ethic Stereotypes
Pryzant, R., Martinez, R. D., Dass, N., Kurohashi, S., Jurafsky, D., and Yang, D. 2019. Automatically Neutralizing Subjective Bias in Text. arXiv preprint arXiv:1911.09709.
Rico, Barry and Alexandra, 2016. Neural Machine Translation of Rare Words with Subword Units
Rishabh Bhardwaj, Navonil Majumder, Soujanya Poria, 2020, Investigating Gender Bias in BERT
Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, Soroush Vosoughi, 2021, Mitigating Political Bias in Language Models Through Reinforced Calibration
SKT-AI, 2021. https://github.com/SKT-AI/KoGPT2
Sperry, Len, 2015, Mental Health and Mental Disorders: An Encyclopedia of Conditions, Treatments, and Well-Being
Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, Rosanne Liu, 2020, PLUG AND PLAY LANGUAGE MODELS: A SIMPLE APPROACH TO

CONTROLLED TEXT GENERATION
Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, Adam Kalai, 2016, Man is to computer programmer as woman is to homemaker? debiasing word embeddings
경정익, 이국철, 2016, "Textmining에 의한 부동산 빅데이터 감성분석 모형 개발", 「주택연구」, 24(4), pp.115～136.
고하정, 2020. 텍스트마이닝을 활용한 도시 공원에 대한 쟁점변화
김대원, 유정석, 2016, "트위터 정보와 아파트 매매 및 전세 가격 간 동적 관계 분석", 「도시행정학보」, 29(1), pp.1～33.
김진유, 2006, "신문기사가 부동산가격변동에 미치는 영향: '투기'가 포함된 신문기사와 주택가격간의 그랜저인과관계분석을 중심으로", 「주택연구」, 14(2), pp.39～63.
박재수, 이재수, 2019a, "벡터자기회귀모형을 이용한 온라인 뉴스기사와 아파트 매매가격의 동태적 관계 연구: 비정형 빅데이터를 활용한 감성분석 기법의 적용", 「감정평가학 논집」, 18(2), pp.83～113.
박재수, 이재수, 2019b, "아파트 매매가격과 부동산 온라인 뉴스의 교차상관관계와 인과관계 분석: 온라인 뉴스 기사의 비정형 빅데이터를 활용한 감성분석 기법의 적용", 「국토계획」, 54(1), pp.131～147.
박재수, 이재수, 2021, "부동산 감성지수의 주택가격 예측 유용성: 뉴스기사와 방송뉴스 빅데이터 활용 사례", 「국토계획」, 56(4), pp.99～111.
박재수, 이재수, 2019. 아파트 매매가격과 부동산 온라인 뉴스의 교차상관관계와 인과관계 분석. 국토계획
배성완, 유정석, 2018, "머신 러닝 방법과 시계열 분석 모형을 이용한 부동산 가격지수 예측", 「주택연구」, 26(1), pp.107～133.
서울특별시, 자치구별 동 현황(한글, 한자), https://www.seoul.go.kr/seoul/autonomy_sub.do
이승희, 2017, [이슈 인 심리학] "내 반려견은 안전해"..인간의 착각이 불러온 사건들, https://news.v.daum.net/v/20170922145504861?s=print_news
이종민, 이종아, 정준호, 2017a, "뉴스 빅데이터를 이용한 전세 가격 예측: 토픽모형 분석을 중심으로", 「부동산학보」, pp.43～57.
이종민, 이종아, 정준호, 2017b, "포털 검색 지수를 활용한 전세 가격 예측: 네이버, 구글을 중심으로", 「부동산학보」, pp.134～148.

이태형, 전명진, 2018, "딥러닝 모형을 활용한 서울 주택가격지수 예측에 관한 연구: 다변량 시계열 자료를 중심으로", 「주택도시연구」, 8(2), pp.39～56.
전해정, 양혜선, 2019, "딥 러닝을 이용한 주택가격 예측에 관한 연구", 「주거환경」, 17(2), pp.37～49.
진창하, Paul Gallimore, 2012, "신문기사 내용과 주택가격: 인식, 사유, 그리고 투자심리", 「부동산학 연구」, 18(2), pp.125～142.
https://en.wikipedia.org/wiki/Importance_sampling
https://github.com/lvwerra/trl
http://sanghyukchun.github.io/76/
https://smj990203.tistory.com/entry/PPOProximal-Policy-Optimization-Algorithms
https://thegradient.pub/prompting/(Tianyu Gao, 2021.07.03., "Prompting: Better Wasys of Using Language Models for NLP Tasks")
https://www.youtube.com/watch?v=L-QYXtJmXrc

Abstract

Analyzing and mitigating housing-related bias in KoGPT-2

SEO, Narae
Data Science
The Graduate School
Seoul National University

Since pretrained language models learn a large amount of data as is, they sometimes generate biased sentences. In this study, housing-related biases in KoGPT-2 were analyzed. The sentences generated by KoGPT-2, which was fine-tuned based on the comments of Naver news and Naver cafe 'Real estate study', were analyzed for biases towards housing types and regions. In addition, reinforcement learning was applied to mitigate biases in KoGPT-2, and debiasing effects were measured. Based on the average distance of sentiment scores, biases on housing types were mitigated by about 73%, and biases on housing regions were mitigated by about 22%. It worked regardless of positive/negative biases. Debiasing method using reinforcement learning has two advantages; First, architectures of PLM, which is KoGPT-2 in this study, do not need to be changed and second, sentences generated by KoGPT-2 can easily be controlled based on human intentions since model created with human-annotated data is used as a reward function.

Keywords : PLM, GPT2, Bias, Housing, Reinforcement learning, Reward

Student Number : 2020-23629