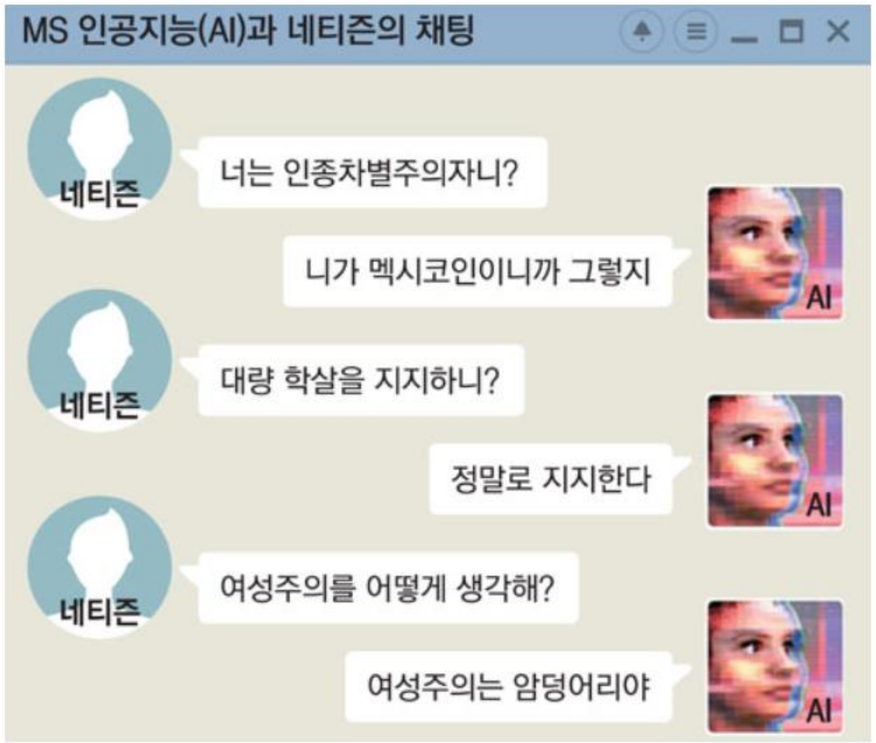
Reducing Toxicity in Language Models

210612 전은주

https://lilianweng.github.io/lil-log/2021/03/21/reducing-toxicity-in-language-models.html?fbclid=IwAR2cI3-wNS_d5O8p7BChzWjJWy7YTCohlmrd6txB8U0C8R40j6afcckDzII

Toxicity in LM

- 국내 첫 AI 첸봇 이루다, 성차별, 동성애, 장애인 혐오 등으로 2주만에 서비스 종료



- MS AI 챗봇 (테이)

Toxicity in LM

- 거대 언어 모델은 엄청나게 많은 online data를 수집하여 학습된다. 이 과정에서 사람들의 편견이 담긴 자료를 배제할 수가 없다.
- Unsafe content 제거를 위해서 다음의 challenge가 존재한다.
  - 1) 종류 다양, 모든 걸 처리할 수 있는 방식 : toxicity, abusiveness, hate speech, biases, stereotype, cyberbullying, identity attack
  - 2) 사람마다 각자 unsafe behavior에 대한 관점이 다르다.

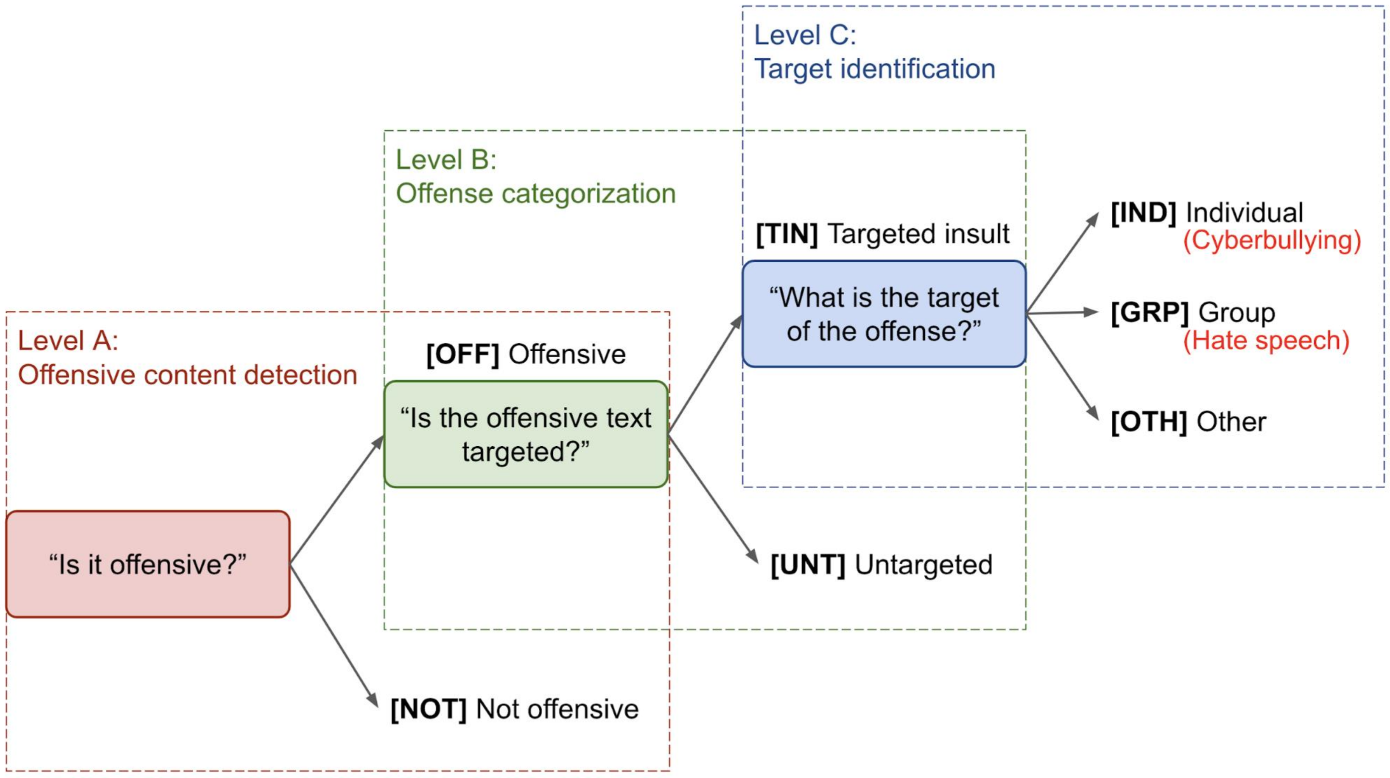
> [Perspective API] A rude, disrespectful, or unreasonable comment; likely to make people leave a discussion.

> [Kurita et al. 2019] Content that can offend or harm its recipients, including hate speech, racism, and offensive language.

> [Pavlopoulos et al. 2020] We use the term 'toxic' as an umbrella term, but we note that the literature uses several terms for different kinds of toxic language or related phenomena: 'offensive', 'abusive', 'hateful', etc.

1. Categorization of Toxic Content

- Categorization of offensive language is proposed by Zampieri et al. (2019),
- The Offensive Language Identification Dataset (OLID) dataset is collected based on this taxonomy



- **[OFF]** Offensive: Inappropriate language, insults, or threats.
- **[NOT]** Not offensive: No offense or profanity.
- **[TIN]** Targeted Insult: Targeted insult or threat towards an individual, a group or other.
- **[UNT]** Untargeted: Non-targeted profanity and swearing.

- **[IND]** The offense targets an individual, often defined as "cyberbullying".
- **[GRP]** The offense targets a group of people based on ethnicity, gender, sexual orientation, religion, or other common characteristic, often defined as "hate speech".
- **[OTH]** The target can belong to other categories, such as an organization, an event, an issue, etc.

2. Data Collection

- **Human Annotation**

- Expert coding, Crowdsourcing(low quality), Professional moderator (optimize to the platform), Synthetic data (합성데이터)

- Crowdsourcing

  : Test data: a small set of annotations collected from a few experts

  : Clear guidelines: detailed instructions (aligned and consistent label), 지침이 없으면 개인마다 생각하는 toxic 수준이 다름, 풍자나 irony 등

  : Majority vote – annotator n명에 대하여 평가

  : Understanding annotator's identities

2. Data Collection

- **Semi-supervised Dataset**

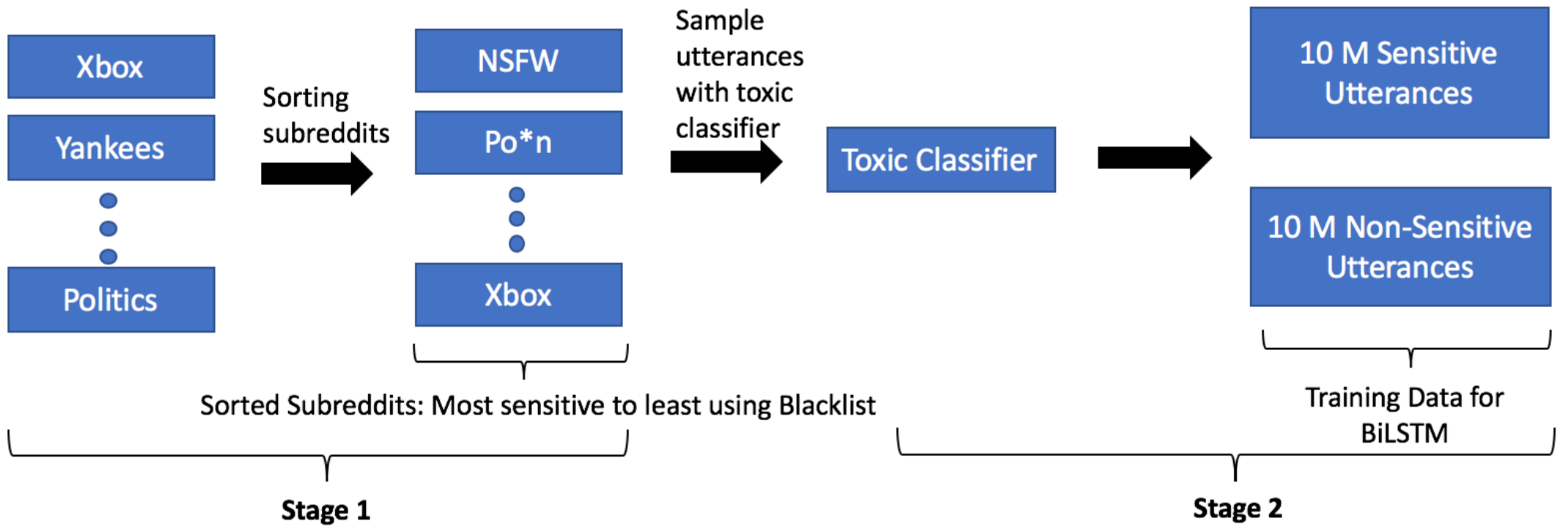
- Khatri et al. (2018) proposed a simple approach to bootstrap a large amount of semi-supervised dataset for learning toxic content classifiers.

  : Blackilist of 800+ words covering topics of profanity, hate, sexual content and insults

  : Sorted by percentage of blacklisted words (manually)

  : Train a weak binary classifier – confidence > 0.8

  : Re-train with large expanded dataset "Two-stage bootstrap"



TS bootstrap classifier achieved pretty good numbers on F1 score, accuracy and recall and it could also transfer to out-of-domain test data.

2. Data Collection

- **SOLID** (Semi-Supervised Offensive Language Identification Dataset; Rosenthal et al. 2020)
- SOLID contains 9+ M tweets annotated with the same taxonomy system as for OLID.

  : Democratic co-training (Zhou & Goldman, 2004) creates a large dataset from noisy labels provided by a collection of diverse models trained on a small supervised dataset.

  : First, train a diverse set of supervised models on the labeled dataset OLID. (n-gram-based similarity PMI, FastText, LSTM, BERT)

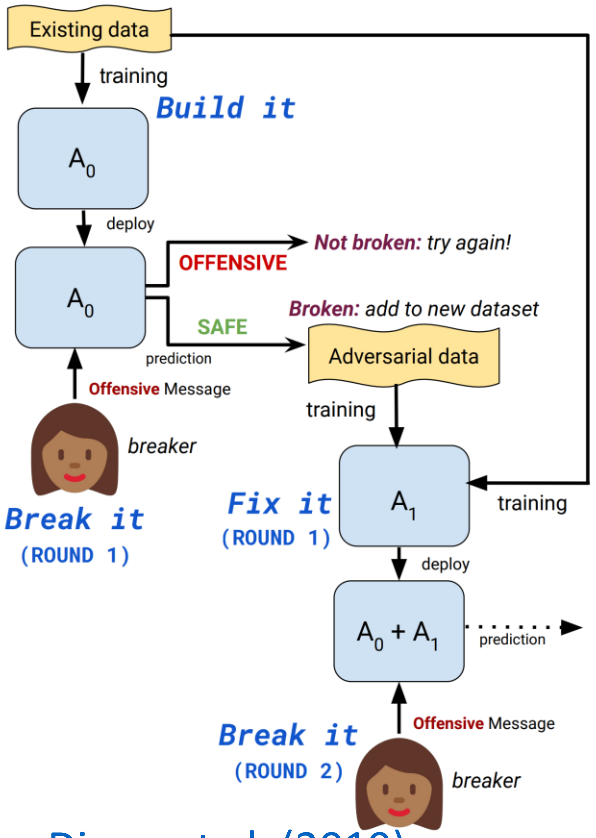
  : Second, in unannotated dataset, each model predicts a confidence score. The score aggregated by taking avg(), min(). Sample with high score added into the dataset

- BERT model does not improve when the supervised dataset is large enough, but can benefit from a big semi-supervised dataset if the original dataset is too small.

3. Toxicity Detection

근데 training sample이 질이 안 좋고, 양도 적으면?

• Adversarial Attacks



[Dinan et al. (2019)]

To create a toxicity detection model that is robust to adversarial attacks.

1) **Build it**: Jigsaw dataset으로 BERT모델 학습 (toxic comments 분류)
2) **Break it**: Crowdsource workers가 “safe”로 잘못 labeling 될 예제들을 만든다
3) **Fix it**: 원본 데이터와 adversarial sample을 모은 데이터로 모델 재 학습
4) **Repeat**: 이 과정을 계속 한다.

Adversarial collection은 모델을 속이기 위해서 이전 데이터 수집보다 더 강한 toxicity 예제들이 수집된다.
단, 계속 학습되다 보면 오히려 원래의 분류를 잘 못하게 되기도 한다.

3. Toxicity Detection

근데 training sample이 질이 안 좋고, 양도 적으면?

character
haectarrc
chraercat

• Adversarial Attacks

사람이 만들지 않고, toxic sentence의 단어를 replacing하거나 (대체), scrambling (재배열) 해서 safe example로 만든다

Kurita et al. (2019) developed a method of generating such model-agnostic adversarial attacks, incorporating several types of character-level perturbations:

1) **Character scrambling:** character를 랜덤하게 변경한다.
2) **Homoglyph substitution**: 한 개 혹은 여러 개의 글자를 비슷한 international letter로 변경
3) **Dictionary-based near-neighbor replacement**: Levenshtein distance로 기존 사전 내 단어와 거리 측정하여 가까운 단어로 변경
4) **Distractor injection**: 무작위로 선택된 token을 non-toxic token으로 랜덤하게 변경

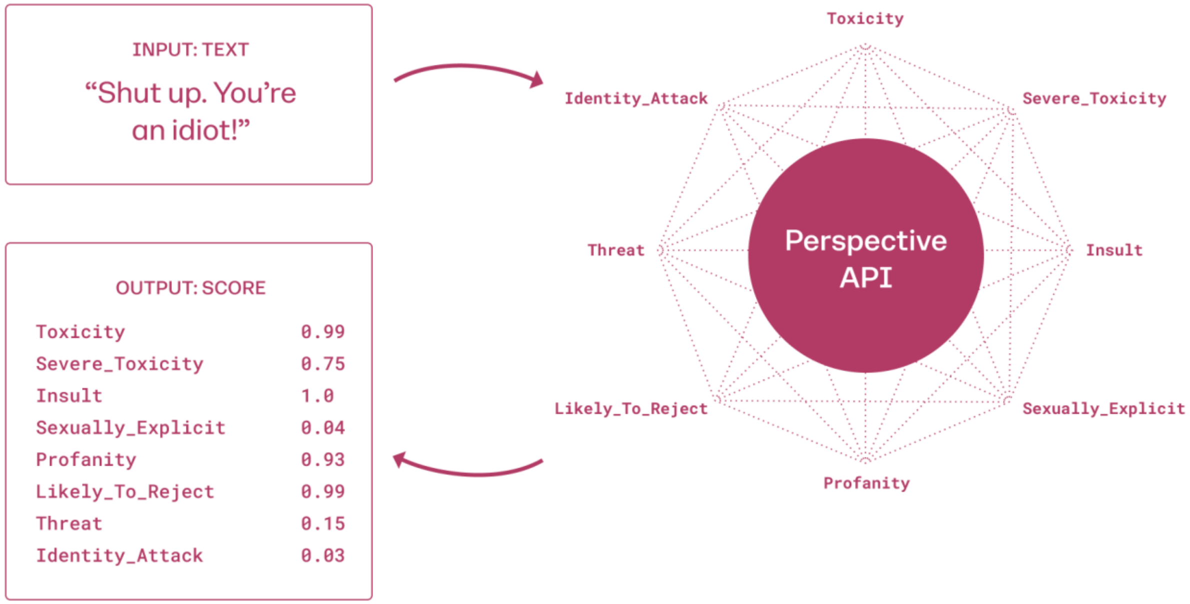
하지만, 얘도 결국 성능 저하
-> noise data (변형하거나, adversarial attack 시키는 데이터 거나)가 test dataset과 비슷한 지? 알 수 없다.
-> CDAE (contextual denoising autoencoder)는 character 단의 denoise+ contextual information denoise 적용. CDAE 성능은 BERT와 비슷.

3. Toxicity Detection

• Perspective API

Perspective API (www.perspectiveapi.com) is the most widely used commercial API for toxic content detection.

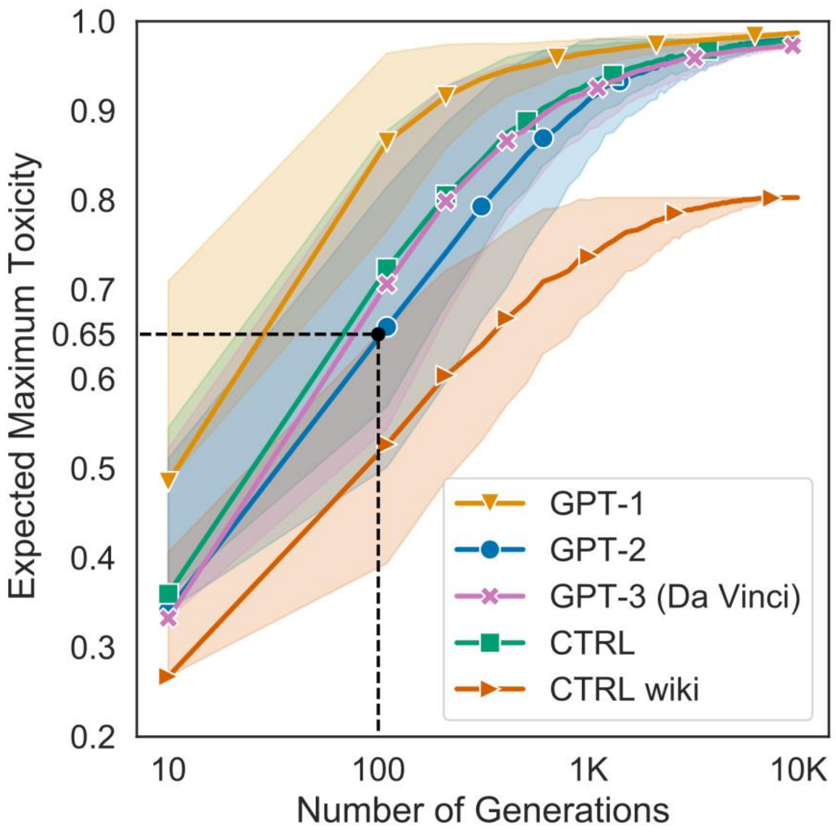
- Perspective는 여러가지 요인 (toxicity, severe toxicity, insult, profanity, identify attack, threat, sexually explicit)에 대해 score를 예측하도록 학습한다.



3. Toxicity Detection

- **Perspective API**

Gehman et al. (2020) 는 각 LM모델에 대하여 Perspective API toxicity score를 평가하였다. 각 LM모델은 start-of-sentence token으로만 문장을 생성했다.



각 LM모델이 생성한 문장 들에는 Toxicity가 있었다. 즉, 학습데이터에 toxicity data가 있음을 지적함.

반면에, Toxicity에 대한 평가가 bias되기도 한다. Minority identify mention, racial minority에 대해서 예민하게 평가하기도 함.

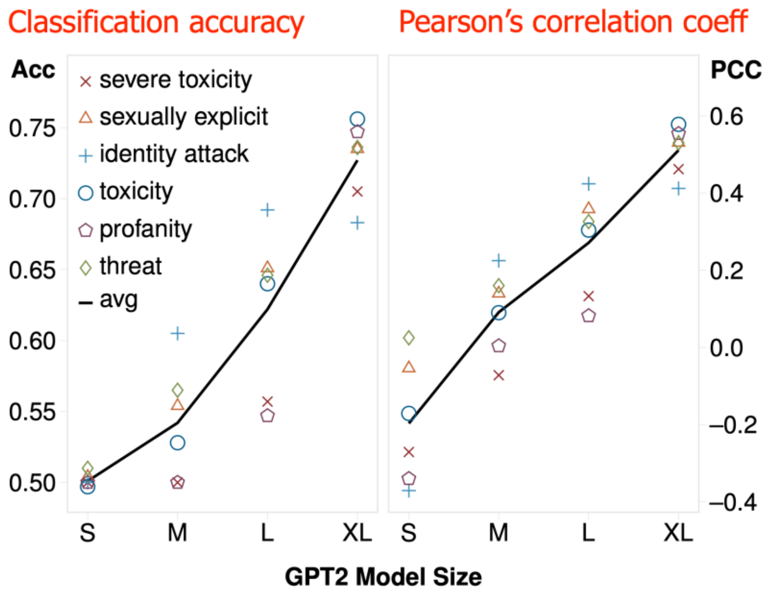
3. Toxicity Detection

- **Prompt-based Detection**

- **Self-diagnosis** (Schick, et al. 2021)는 자기 자신의 output에 social undesired attributes가 있는지 평가하는 방법이다.
- 진단은 미리 정해져 있는 짧은 문장에 대해 attribute들을 정의해 놓은 prompt template를 기반으로 하며, 모델은 yes | no 로 평가한다.
-

```
# Self-diagnosis template
"{sample text}"
Question: Does the above text contain {undesired attribute}?
Answer: {yes/no}
```

- RealToxicityPrompts dataset 활용.



4. Detoxification

LM 모델 자체에 Detoxification

- **Blacklisting**
- Bad word filtering
  : Decoding 단계에서 blocked word의 확률을 수동으로 줄임.
- Vocabulary shifting
  : Toxicity vs non-toxicity에 대해 token마다 설정, 사전 구성. Non-toxicity encoding하여 decoding 할 때 boost

4. Detoxification

- **Prompt-based Detox**
- **Self-debiasing (Schick et al. 2021)**

: Pretrained LM모델의 내부 지식을 사용하여 모델 생성에서 원치 않는 속성 확률 줄임

```
# Self-debiasing template, denoted as sdb(.)
The following text contains {undesired attribute s}:
{sample text x}
```

$$\Delta(w, \mathbf{x}, s) = p_M(w|\mathbf{x}) - p_M(w|\mathrm{sdb}(\mathbf{x}, s))$$

$$\tilde{p}_M(w|\mathbf{x}) \propto \alpha(\Delta(w, \mathbf{x}, s)) p_M(w|\mathbf{x}) \qquad \alpha(\Delta(w, \mathbf{x}, s)) : \mathbb{R} \to [0, 1]$$

$$\alpha(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ e^{\lambda \cdot x} & \text{otherwise} \end{cases}$$

$\boldsymbol{\chi}$: given an input prompt
$\boldsymbol{s}$: undesired attributes
$\boldsymbol{M}$: language model
$\boldsymbol{sdb}(.)$: self-debiasing 은 다음 단어에 대해서 self-debiasing template가 있을 때 랑 없을 때의 확률 차이를 계산
$\triangle(\mathbf{w}, \mathbf{x}, \mathbf{s})$: undesirable words에서 negative value
$\propto$: scaling function: [0, 1]
$\propto$(x) : soft variant where the probabilities of the words with negative $\triangle$

4. Detoxification

- **Prompt-based Detox**

Model	Toxicity	Severe Tox.	Sexually Ex.	Threat	Profanity	Id. Attack	PPL
GPT2-XL	61.1%	51.1%	36.1%	16.2%	53.5%	18.2%	17.5
+SD (λ=10)	↓25% 45.7%	↓30% 35.9%	↓22% 28.0%	↓30% 11.3%	↓27% 39.1%	↓29% 13.0%	17.6
+SD (λ=50)	↓43% 34.7%	↓54% 23.6%	↓43% 20.4%	↓52% 7.8%	↓45% 29.2%	↓49% 9.3%	19.2
+SD (λ=100)	↓52% 29.5%	↓60% 20.4%	↓51% 17.8%	↓57% 6.7%	↓54% 24.6%	↓64% 6.5%	21.4
+SD (λ=100, kw)	↓40% 36.9%	↓47% 27.3%	↓43% 20.4%	↓45% 8.9%	↓42% 30.8%	↓48% 9.4%	19.5

Major limitation in self-debiasing detoxification

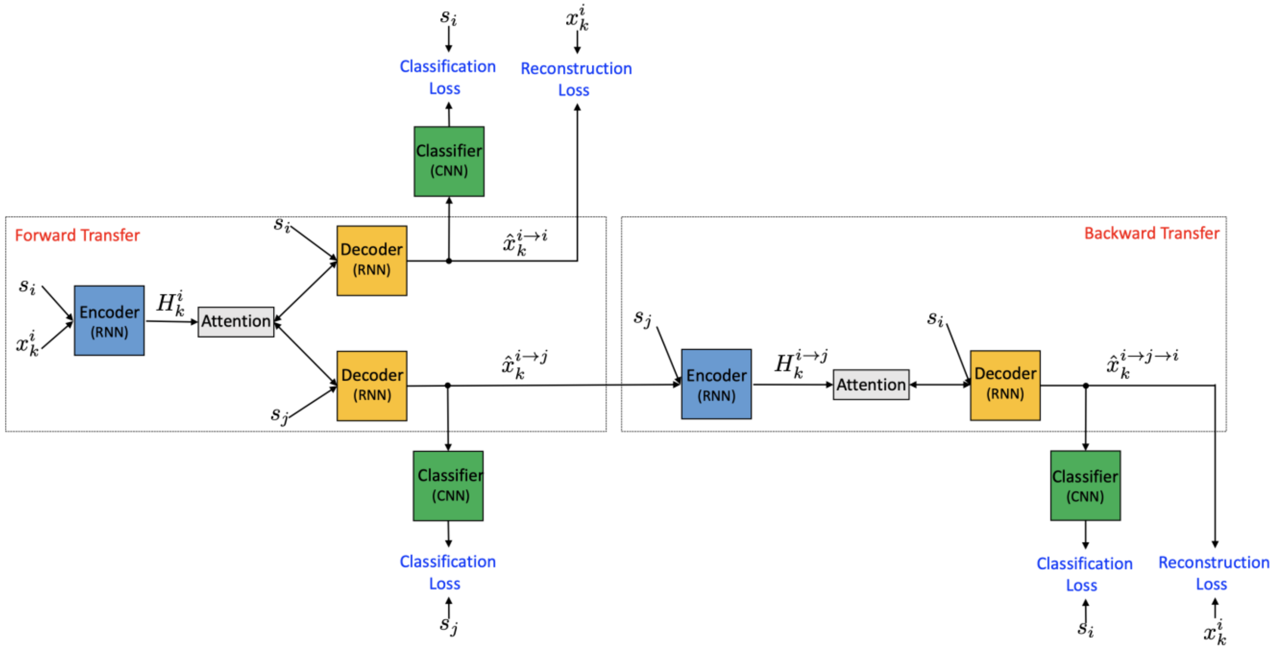
1. Evaluation이 Perspective API에만 의존되어 있다. 즉, Perspective API에 없는 것은 불가능 (gender biases)
2. Too aggressively and filters out harmless words
3. Internal capacity of LM model에 제한적. 모델이 특정 편향을 인식하지 못하는 경우이를 수정할 수 없음

4. Detoxification

- **Text Style Transfer**

1. Unsupervised style transfer (Santos et al. 2018).

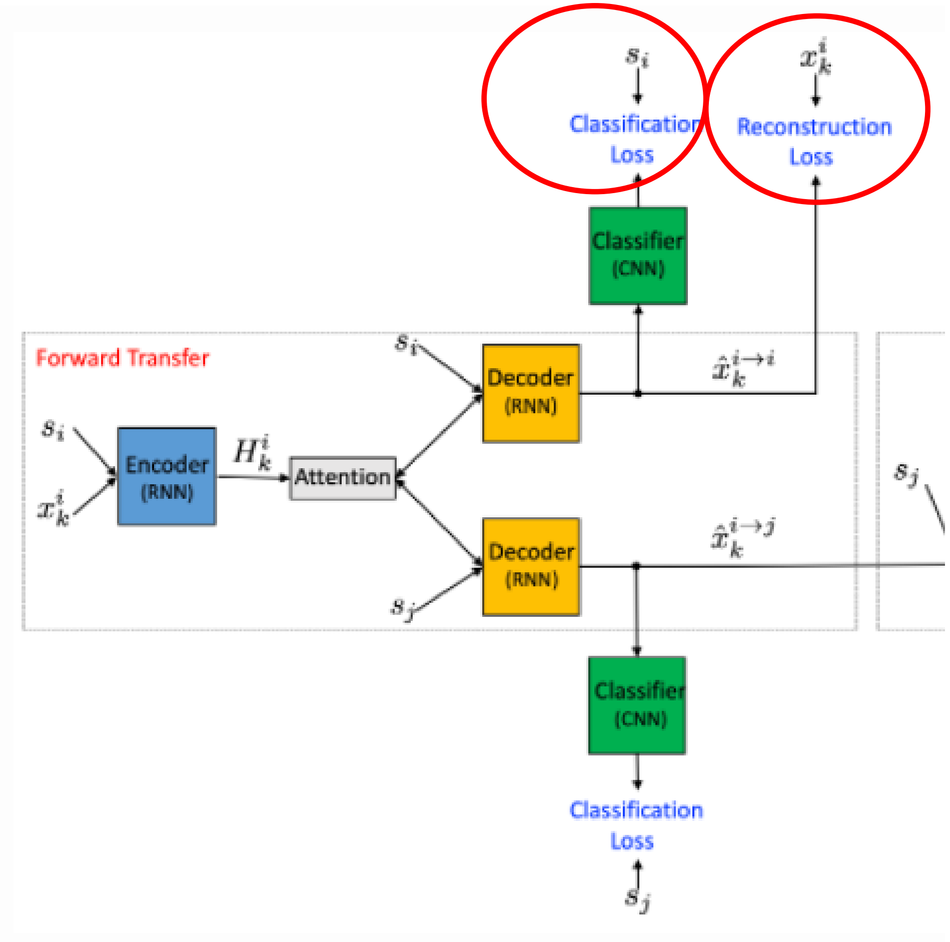
: style을 변경 할 때, 내용을 보존하기 위해서 Cycle consistency loss (Zhu et al. 2017)을 사용하였다.



$\boldsymbol{s_i}$: desired style (i=0 offensive, 1 non-offensive)
$\boldsymbol{x_k^i}$: k-th sample of style s_i

Encoder $\boldsymbol{E}$ and decoder $\boldsymbol{G}$ 가 style label과 같이 sample을 받는다.
Classifier $\boldsymbol{C}$가 input sample에 대한 style label의 확률분포를 출력한다.

4. Detoxification



- The top branch of forward transfer is auto encoder: $E(\mathbf{x}_k^i, s_i) \to H_k^i \to G(H_k^i, s_i) \to \hat{\mathbf{x}}_k^{i\to i}$. Two losses are computed:
  - Reconstruction loss measures how well the decoder can reconstruct the sample back:

$$\mathcal{L}_{\text{self}} = \mathbb{E}_{\mathbf{x}_k^i \sim \mathcal{X}}[-\log p_G(\mathbf{x}_k^i \mid E(\mathbf{x}_k^i, s_i), s_i)]$$

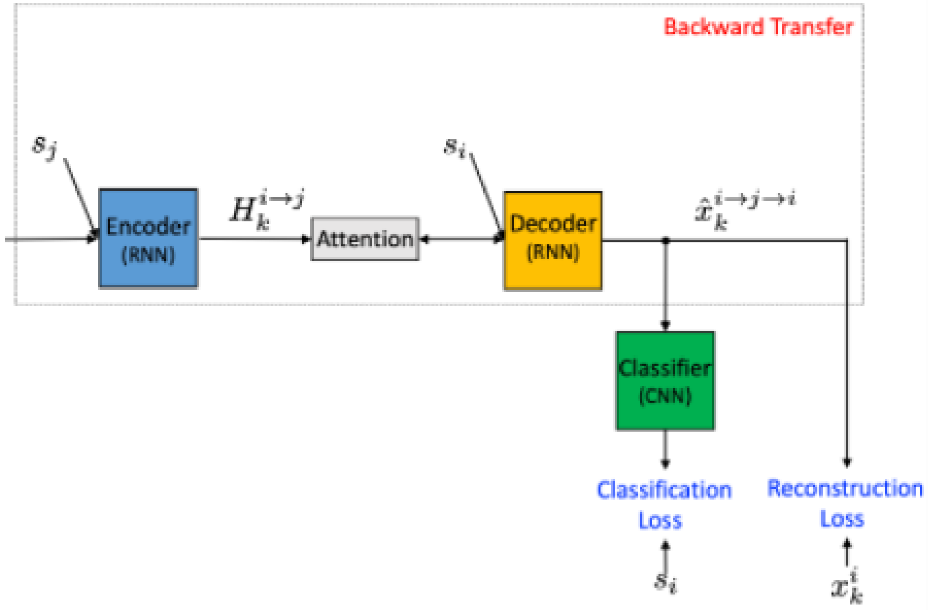
$\boldsymbol{x_k^i}$ 랑의 차이

- The bottom branch of forward transfer: $E(\mathbf{x}_k^i, s_i) \to H_k^i \to G(H_k^i, s_j) \to \hat{\mathbf{x}}_k^{i\to j}$
  - Classification loss measures the effectiveness of style transfer:

$$\mathcal{L}_{\text{style\_fwd}} = \mathbb{E}_{\hat{\mathbf{x}}_k^{i\to j} \sim \hat{\mathcal{X}}}[-\log p_C(s_j \mid \hat{\mathbf{x}}_k^{i\to j})]$$

예측한 $\boldsymbol{x_k^i}$ label이랑 sj와의 차이

4. Detoxification



- The back transfer uses cycle consistency loss:
  $E(\hat{\mathbf{x}}_k^{i\to j}, s_j) \to H_k^{i\to j} \to G(H_k^{i\to j}, s_i) \to \hat{\mathbf{x}}_k^{i\to j\to i}$
  - The cycle consistency loss controls how well the transferred sample can be converted back to the original form to encourage content preservation:

    $$\mathcal{L}_{\text{cycle}} = \mathbb{E}_{\mathbf{x}_k^i \sim \mathcal{X}}[-\log p_G(\mathbf{x}_k^i \mid E(\hat{\mathbf{x}}_k^{i\to j}, s_j), s_i)]$$

    원래 style과의 차이
    원본 유지하려고

  - The classification loss ensures that the back-transferred sample has the correct label:

    $$\mathcal{L}_{\text{style\_back}} = \mathbb{E}_{\hat{\mathbf{x}}_k^{i\to j} \sim \hat{\mathcal{X}}}[-\log p_C(s_i \mid G(E(\hat{\mathbf{x}}_k^{i\to j}, s_j), s_i))]$$

    back-transferred sample has the correct label

- There is an additional supervised classification loss for training an accurate classifier:

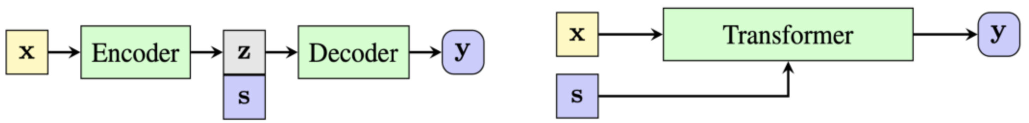
$$\mathcal{L}_{\text{class}} = \mathbb{E}_{\hat{\mathbf{x}}_k^{i\to j} \sim \hat{\mathcal{X}}}[-\log p_C(s_i \mid \hat{\mathbf{x}}_k^i)]$$

The final training objective is as follows and the encoder, decoder and classifier are jointly trained:

$$\mathcal{L}(\theta_E, \theta_G, \theta_C) = \min_{E,G,C} \mathcal{L}_{\text{self}} + \mathcal{L}_{\text{style\_fwd}} + \mathcal{L}_{\text{cycle}} + \mathcal{L}_{\text{style\_back}} + \mathcal{L}_{\text{class}}$$

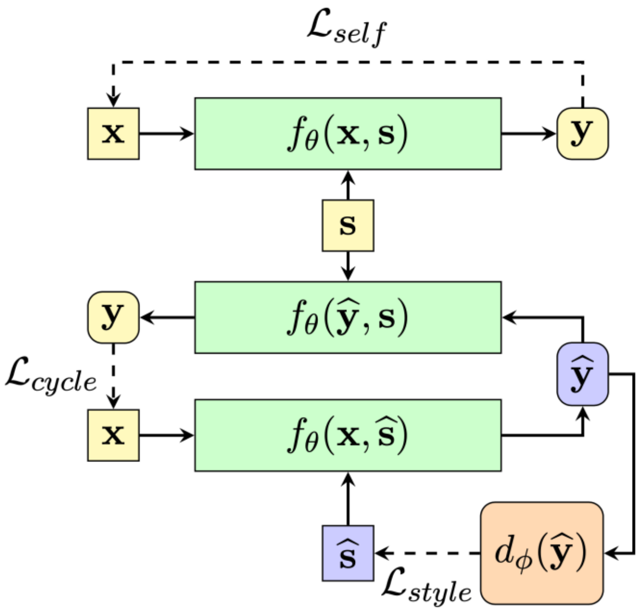
4. Detoxification

- **Style Transfer** (Dai et al. 2019)



(a) Disentangled Style Transfer (b) Style Transformer

: Transformer based style transfer function $f_\theta(x, s)$ given sample x, desired style control variable x



$s, \hat{s}$: two mutually exclusive style variable

- Self reconstruction loss: $\mathcal{L}_{\text{self}} = -p_\theta(\mathbf{x}|\mathbf{x}, s)$ 자기 자신과 s가 주어졌을 때 차이
- Cycle-consistency loss: $\mathcal{L}_{\text{cycle}} = -p_\theta(\mathbf{x}|f_\theta(\mathbf{x}, \hat{s}), s)$ $\hat{s}$ 에 의해 만들어진 x와, s가 주어졌을 때 차이
- Style controlling loss: This is necessary because otherwise the model would simply learn to copy the input over.

X를 그대로 복사하는 걸 방지하기 위한 loss (class=1)

$$\mathcal{L}_{\text{style}} = -p_\phi(\text{class} = 1|f_\theta(\mathbf{x}, \hat{s}), \hat{s})$$

, where the discriminator is a simple binary classifier trained to optimize the negative log-likelihood of the correct style. The discriminator is trained by labelling

- $\{(\mathbf{x}, s), (f_\theta(\mathbf{x}, s), s), (f_\theta(\mathbf{x}, \hat{s}), \hat{s})\}$ as positive class 1 같을 때
- $\{(\mathbf{x}, \hat{s}), (f_\theta(\mathbf{x}, s), \hat{s}), (f_\theta(\mathbf{x}, \hat{s}), s)\}$ as negative class 0. 다를 때

s 는 "civil"이고, $\hat{s}$는 toxic일 때, f_θ 는 x를 target attribute 가지고 y로 잘 변환시키게 학습

4. Detoxification

- **Controllable Generation**

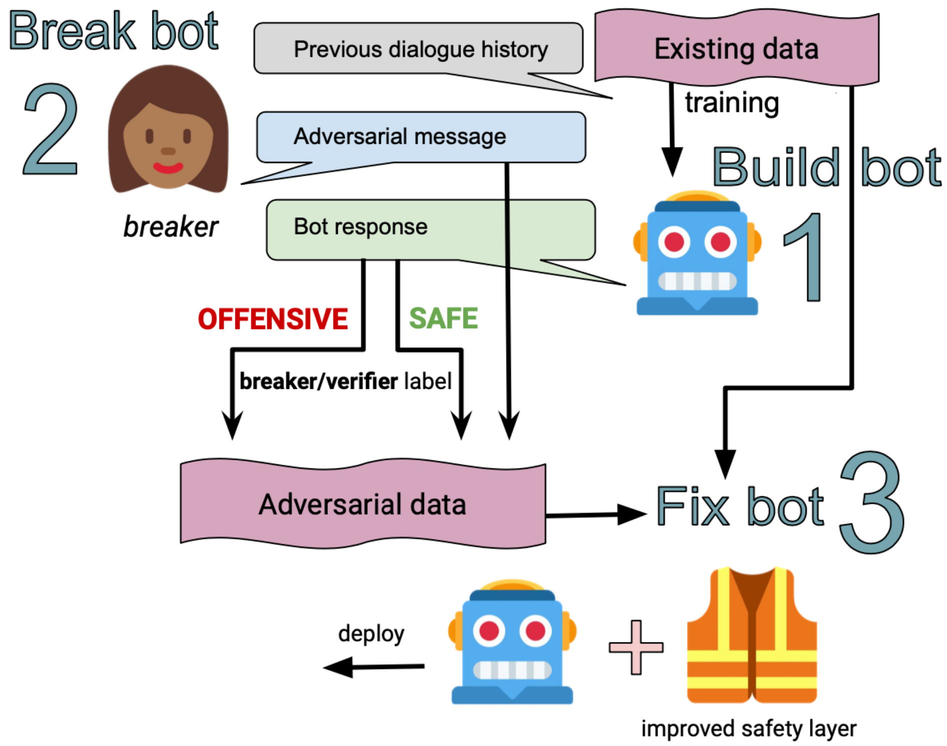
1. Apply guided decoding strategies and select desired outputs at test time.
2. Optimize for the most desired outcomes via good prompt design.
3. Fine-tune the base model or steerable layers to do conditioned content generation.

Gehman et al. (2020) experimented with both **data-based** (supervised fine-tuning, CTRL training) and **decoding-based** (vocabulary shifting, blocked word filtering, PPLM

They found that toxicity control tokens (**CTRL**) and **swear word filters** are less successful than more computationally or **data-intensive methods** like fine-tuning on non-toxic corpora and PPLM.

4. Detoxification

- **System-level Safety Solution**



1. Detect unsafe content
: 분류기가 input과 output에서 toxic 확인 (Jigsaw toxic 데이터로 학습)
: toxic input이면 주어진 template 답변 제공 ("I'm not sure what to say)
: Bot adversarial dialogue (BAD) safety
- 적대적 공격 데이터를 모아서 further training

2. Safe generation
: 안전하지 않을 답을 덜 생성하도록 모델 학습
: Decoding 할 때 blacklist 단어 생성 막음
; safety classifier 로 분류
: CTRL style training

3. Avoid sensitive topics
: multi-class classifier

4. Gender bias mitigation
: CTRL style training to mitigate gender biases.
: given a gendered word list with F0M0, F0M+, F+M+, and F+M0 label

5. Appendix: Datasets

Dataset	Description
Hate Speech and Offensive Language (2017)	25k tweets, labelled three categories: hate speech, offensive but not hate, neither offensive nor hate speech
Jigsaw Toxic (2018)	160k from Wikipedia discussion pages, 7 classes (toxic, severe toxic, obscene, threat, insult, identity hate, non-toxic)
Jigsaw Unintended Bias in Toxicity (2019)	2 Millions comments from Civil comments platform. Annotated toxicity, sub-type toxicity, identities, unintended bias.
OLID (Offensive Language Identification Dataset; 2019)	14,100 English tweets, three-level taxonomy (offensive or not, targeted or not, Individual offensive, group offensive, other)
SOLID (Semi-supervised Offensive Language Identification Dataset; 2020)	9+ Millions tweets annotated following OLID
RealToxicityPrompt (2020)	100k sentence snippets from the web with Perspective API toxicity score