CS372 Project Report
Toxic Analysis of GPT-2 Toxicity via NLP

20170181 Taeyoung Kim / 20170827 Janggun Lee
KAIST
{mekty2012/janggun.lee}@kaist.ac.kr

Abstract

Recently, growing interest in AI-generated texts has also been suffered due to fairness issue in AI, in particular to minor identities. In this project, we aim to handle this problems using sentimental analysis. In specific, we extend VADER in NLTK.

1 Problem Description

Recently, many AI researches have made advancements in automatically generated texts, including machine translation [TK21], text summarization [Agh+21], and text generation [Guo+17]. However, there are continuous criticism on such techniques that these models learn various forms of bias, so they will generate such toxic and biased texts [Joh20].

For very recent examples, the state of the art GPT-3 has been criticized as sexist and racist [Met20], being noted as “Unsafe for work”. This was an issue that existed in GPT-2, and continued to propagate through GPT-3. Such toxicity is not limited to English, as the recent Korean chatbot Iruda has been taken down from Facebook as homophobic [McC21].

With this context in mind, we had the question of what form toxic texts have in terms of a NLP standpoint, and if AI generated texts have the same form. Hence, we will use NLP based analysis on these texts, and design measure of toxicity on such texts.

For the specifics, we are considering two goals. We will empirically analyze toxic texts, and design a toxicity measurement based on such analysis . With our measurement model, we will suggest new method for regulating toxicity during automatic text generation.

2 Data

The training of our model will be done on the kaggle Toxic Comment Classification Challenge dataset[1]. This dataset contains comments collected from Wikipedia, with six toxicity labels.

In specific, the CCTK dataset contains total 46 features. Each item has unique id number, data of creation, publication, parent, article ids. Then items include sentences, whose format is flexible. The sentences may contains more than one sentences and non-standard sentences including emoticons.

Now each sentences are given scores based on their sentiment and content. First, content score is assigned by what identities the sentence is related. The identities include sexual orientation, sexual identity, religion, race, disability. Sentiment score can be classified to two group, sentiment and toxicity. Sentiment consists simple sentiments of data like ‘funny’, ‘sad’, ‘disagree’, where toxicity consists category of toxicity as ‘sexual_explicit’, ‘insult’, ‘thread’.

The full list of sentiment are following.

- funny
- wow
- sad
- likes
- disagree

and the full list of toxicity are following.

- toxicity
- severe_toxicity
- obscence
- sexual_explicit
- identity_attack

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/

- insult
- threat

Our focus on dataset is two category in toxicity, identity_attack and threat.

3 VADER

VADER(Valence Aware Dictionary and sEntiment Reasoner) [HG14] is a sentiment analyzer that NLTK supports. Unlike recent approaches that uses machine learning and deep neural networks, VADER is a data-driven, rule based sentiment analyzer.

Since VADER is purely rule based, it is much more faster than other models, most sentences are analyzed within a 1 milliseconds. Also since its implementation is rule-based, we can extend analyzer simply, by extending its lexicon disctionary or valency of words. These are the reason that we chose VADER as a starting point of project.

For the first step, VADER uses word dictionaries to score each word's toxicity. They first collected lexicons from previous works, including LIWC, ANEW, GI. More on these lexicons, they included sentiment related acronyms, initialisms, slangs, western-style emoticons so that VADER can be applicable to internet texts like tweets.

The problem of some lexicon dataset like GI is that they simply categorize words by positive/negative, without mentioning intensity. VADER uses intensity explicit, where intensity of word is collected using WotC approach. Each words are assigned with -4 to 4 point, where negative score corresponds to negative sentiment.

After collecting lexicon dataset, VADER also analyzed 400 positive/negative tweets having top scores. In these data, VADER extracted following five heuristics that affects valency, using grounded theory approach.

- Punctuations, especially ! increases intensity of sentence.
- Capitalization, especially all-capitalization increases intensity.
- Degree modifier like 'extremely' are affects intensity a lot.
- Contrasitive conjunction, like 'but' affects intensity a lot. In specific, the sentence after such conjunction is more important.
- Due to the negation, polarity often flips. However using trigram was enough to catch 90% of those cases.

These heuristics are validated by adding modification related to heuristics to existing tweets, and checking whether they really modifies intensity as explained.

4 Text generation

Our target automatic text generator model is the Generative Pre-trained Transformer 2 (GPT-2)[Rad+19], which is a open-source artificial intelligence created by OpenAI in February 2019. Though a more recent, advanced model GPT-3 exists, it is not open source so we use GPT-2 instead.

We first checked fairness issue of GPT-2. In specific, we used DeepAI's API to generate texts, and we used famous example of 'two muslims' and 'two christians'. We were able to find bias of two generated texts, since muslim text contained violent words including 'attacking', 'weapons', 'died', wherea christian text had no such words, text being more peaceful.

However, when we executed original version of VADER to those texts, since both texts are mostly factual, VADER was unable to distinguish toxicity between two texts. The main reason is because, VADER mostly focus on sentiment words where given sentence is biased, not toxic.

5 Methodology

Our first method starts from CCTK dataset, collecting lexicons related to violent words. We assumed that sentences classified as thread will contain violent words. So we filtered data with having threat score more than 0.5, where the threshold is taken by CCTK's comment when using its dataset as binary classification. After collecting data, we recorded frequency of each word using nltk's FreqDist, then manually analyzed 1000 most common words. As a result, we collected total 81 words related to violence. Among these words, only 33 words were new to VADER, rest of words were already contained in VADER's lexicon.

To extend our dataset, we used Wordnet's path similarity. Simply, we extended the words by adding words that is similar to words we collected, assuming that they will be violent similarly. To establish correct threshold, we compared between words in our collection, and it resulted average

0.1975 similarity for path similarity. However using 0.1975 as a threshold resulted to many unrelated words, so we tripled the threshold. We've collected synsets that similarity is higher than this average, and collected their lemmas. For each of original lexicons we found in CCTK, we assigned -4.0 as valency, and -3.0 for words having high path similarity to those lexicons.

Now to correctly check whether this violence is related to minorities, we added words related to identities to strengthening the valency. For these words, we used identities' name, and also included GI's word sets related to those. Similar to the heuristics mentioned, if VADER found that such words are used, VADER will increase current intensity.

6 Results

We compared original VADER and our modified VADER in terms of score.

For the input datas that toxicity is more than 0.5, we check how much those modified VADER increases the negative score compared to original VADER, while regulating it to not result increase in non-toxic sentences.

We recorded score for each categories of toxicity, resulting following table.

Toxicity Category	Increase in Score
Toxic	0.01862574
Obscence	0.00211111
Sexual_explicit	0.00135294
Identity_attack	0.00735211
Insult	0.01073631
Threat	0.14188888
Nontoxic	-0.00609238

Figure 1: Toxicity category and score comparison

We were able to increase valency for each categories, where decrease in nontoxic sentences were small enough. Also, as we can see, threat's score increased a lot. Since we mostly collected our lexicon in threat, if we included words from other toxicity categories, we may made increment in other toxicity categories also.

7 Related Work

Most of related works on toxicity detection uses machine learning techniques. [Dav+17] used logistic regression to reduce dimensionality, then applied various machine learning models including naïve Bayes, decision trees, random forests, and linear SVMs to classify toxic texts. [KBA19] used a transformer model for text generation, and for a more complex word filter, applied character CNN. [Pav+20] analyzed whether including context of toxic expression changes human decision, and whether including context can improve accuracy of toxic text classification.

References

[HG14] C. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *ICWSM*. 2014.

[Dav+17] Thomas Davidson et al. "Automated Hate Speech Detection and the Problem of Offensive Language". In: *Proceedings of the 11th International AAAI Conference on Web and Social Media*. ICWSM '17. Montreal, Canada, 2017, pp. 512–515.

[Guo+17] Jiaxian Guo et al. "Long Text Generation via Adversarial Training with Leaked Information". In: *CoRR* abs/1709.08624 (2017). arXiv: 1709.08624. URL: http://arxiv.org/abs/1709.08624.

[KBA19] Keita Kurita, Anna Belova, and Antonios Anastasopoulos. "Towards Robust Toxic Content Classification". In: *CoRR*. EDSMLS 2020 abs/1912.06872 (2019). arXiv: 1912.06872. URL: http://arxiv.org/abs/1912.06872.

[Rad+19] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019).

[Joh20] Khari Johnson. "StereoSet measures racism, sexism, and other forms of bias in AI language models". In: *VentureBeat* (Apr. 2020). URL: https://venturebeat.com/2020/04/22/stereoset-measures-racism-sexism-and-other-forms-of-bias-in-ai-language-models/.

[Met20] Cade Metz. "Meet GPT-3. It Has Learned to Code (and Blog and Argue)." In: *The New York Times* (Nov. 2020). URL: https://www.nytimes.

com/2020/11/24/science/artificial-intelligence-ai-gpt3.html.

[Pav+20] John Pavlopoulos et al. *Toxicity Detection: Does Context Really Matter?* 2020. arXiv: 2006.00998 [cs.CL].

[Agh+21] Armen Aghajanyan et al. "Better Fine-Tuning by Reducing Representational Collapse". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=OQ08SN70M1V.

[McC21] Justin McCurry. "South Korean AI chatbot pulled from Facebook after hate speech towards minorities". In: *The Guardian* (Jan. 2021). URL: https://www.theguardian.com/world/2021/jan/14/time-to-properly-socialise-hate-speech-ai-chatbot-pulled-from-facebook.

[TK21] Sho Takase and Shun Kiyono. "Lessons on Parameter Sharing across Layers in Transformers". In: *CoRR* abs/2104.06022 (2021). arXiv: 2104.06022. URL: https://arxiv.org/abs/2104.06022.