Exploring robustness of Language Models

Ayush Kalani
ASU
*a*kalani2@asu.edu

Nakul Vaidya
ASU
*n*vaidya7@asu.edu

I. Problem definition

Machine Learning (ML) models have excelled in a variety of tasks during the past ten years, including classification, regression, and decision-making. Recently, though, it has been shown that they are susceptible to adversarial instances, which are real inputs that have had modest, frequently undetectable changes made to them. Such models [2] might be used to target human populations for deception, manipulation, fraud, and other negative outcomes when combined with other computational and methodological advancements. Along with the release of open source models like Bloom or Stable Diffusion, these new simulation methods of entire demographics are now accessible to all programmers, hackers, and trolls who, given access to sufficient computing power and Python skills, can produce the simulation of the political utterances of an entire nation. On the plus side, we may attempt to use AI to combat this information overload. For instance, by using AIs that can reason to identify flaws in arguments. The AI might then point out logical errors in what they say and show them next to the remark for the user, at which point it wouldn't matter who or how many people said it or how beautifully it was stated.

Existing large language models such as GPT3, BERT [3], and T5 [6] hold particular promise as social science generalist tools to mimic general human behavior. We want to test the robustness of these models against data sets that are designed to fool the system, and which can be human judged correctly. While existing works [4]test adversarial examples on image and language data sets, we want to test the robustness and generalizability of these models using TextFooler [5] baseline benchmark.

Formally, [2] language models such as GPT-3 are a conditional probability distribution P(Xn—X1....Xn1) over tokens, where each Xi comes from a fixed vocabulary. Tuning a language model on different contexts reduces the probability of some outputs and increases the probability of others. For example, given the context x1, x2, x3, x4 = "Can you drive a", a language model might assign high probability to x5="car", and low probability to x5="apple", but changing a single word in the context to x1, x2, x3, x4 = "Can you eat" reverses that.

[2]The model calculates a probability distribution at each generating step that represents the possibility that any specific vocabulary token would have been the following observed xi if the model were reading a pre-written text. It chooses one of the most likely possibilities by applying a distribution function, adds the new xi to the conditioning context, and then repeats the procedure. This keeps on until either a certain quantity of tokens have been created or until anything outside of the process stops it like a stop token. Given that GPT-3 chooses output tokens probabilistically, it can provide a wide range of potential continuations.

A. Black Box Threat Model

Under the black-box setting, the attacker is unaware of the model architecture, parameters, or training data. It can only query the target model with supplied inputs, getting as results the predictions and corresponding confidence scores.

II. Challenges

There are a number of challenges associated with this project. Firstly, GPT-3 is a paid service and a limited amount of evaluation can be done when comparing it with BERT. Secondly, GPT-3 is not an open source model and this study will give us its comparison with the BERT model and help us in answering some of the advancements used in GPT-3. Finally, this project will require an in depth understanding of creating adversarial examples and several large language models. This project will give us an opportunity to play with GPT-3 and discover new findings about it.

TextFooler framework has a collection of 7 diverse datasets for different downstream tasks, so selecting which dataset to use for the given problem is an interesting task because perturbation is variable. Getting the words to replace the random words in the dataset is also a daunting task.

III. Most related prior work and its shortcomings

Adversarial attack has been extensively studied in computer vision. Adversarial attack Discrete data such as text is more challenging. Inspired by the approaches in computer vision, early work in language adversarial attack focused on variations of gradient based methods. But recent work on Generating Natural Language Adversarial Examples [1]. These improvements have been previously studied on older language models with the paper on Is BERT Really Robust? A Strong Baseline

for Natural Language Attack on Text Classification and Entailment [5]. With the motivations from previous research we will be targeting GPT-3 with in this project as there has been very limited work that attacks GPT-3 model on text classification.

IV. Proposed approach

In this work, we present the application of TextFooler [5] framework on GPT-3 and comparison with other state of the art model such as BERT. This is an interesting setting because GPT-3 is a black box to everyone where no model architecture or parameters are accessible. The framework identifies important words for the target model and replaces them with most semantically and grammatically correct perturbed words until the prediction is altered. We propose an evaluation of TextFooler [5] baseline framework on state of art GPT-3 model on two popular downstream tasks – text classification and textual entailment. Fine based local biases and global biases represent the framework data set. We plan to randomly select words to perturb and keep the word ratio same for all downstream tasks for both the models in comparison.

We believe our study will explore novel robustness on state of the art latest GPT-3 language model. While this is our initial proposal we also will run other experiments including social, racial and gender biased adversarial examples [7] to see if we can extract meaningful insights from it.

V. Data set

We are using existing datasets available via open source repositories. We are keeping the percentage of perturbed words in the datasets to be less than 20%.

A. Text Classification

Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content.

YELP: The Yelp reviews dataset consists of reviews from Yelp. It is extracted from the Yelp Dataset Challenge 2015 data. The Yelp reviews polarity dataset is constructed by considering stars 1 and 2 negative, and 3 and 4 positive. For each polarity 280,000 training samples and 19,000 testing samples are take randomly. In total there are 560,000 trainig samples and 38,000 testing samples. Negative polarity is class 1, and positive class 2.

B. Textual Entailment

Textual Entailment (TE) is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text. In the TE framework, the entailing and entailed texts are termed text and hypothesis, respectively.

MultiNLI: The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. The corpus is modeled on the SNLI corpus, but differs in that covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation.

VI. Evaluation

We study the effectiveness of our adversarial attack on two important NLP tasks, text classification and textual entailment. For text classification we will be using YELP dataset and for textual entailment we are going to use MultiNLI dataset. Following the practice by Alzantot et al. (2018), we evaluate our algorithm on a set of 1,000 examples randomly selected from the test set. For each dataset, we train state of the art, large language models BERT and GPT-3 on the training set. We will then evaluate adversarial examples that are semantically similar to the test set to attack the trained models and make them generate different results. We first report the accuracy of the target models on the original test samples before the attack as the original accuracy. Then we measure the accuracy of the target models against the adversarial samples crafted from the test samples, denoted as after-attack accuracy. By comparing these two accuracy scores, we can evaluate how successful the attack is on BERT and GPT-3. As we don't know much about GPT-3, we will also try to answer the variation in results for both the models.

References

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

[2] Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *arXiv preprint arXiv:2209.06899*, 2022.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[5] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019.

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[7] Yuting Yang, Pei Huang, Juan Cao, Jintao Li, Yun Lin, Jin Song Dong, Feifei Ma, and Jian Zhang. A prompting-based approach for adversarial example generation and robustness enhancement. *arXiv preprint arXiv:2203.10714*, 2022.