Imperial College London

MEng Individual Project

Imperial College London

Department of Computing

Detecting Hidden Purpose in NLP Models

Author:
Euan Scott-Watson

Supervisor:
Prof. Yves-Alexandre de Montjoye

Second Marker:
TODO: Second Marker Name

January 19, 2023

Contents

Chapter 1

Introduction

1.1 Natural Language Processing

As with any advancement in the field of computing, shortly after discovery, members of the community will soon begin probing said discovery to find ways to attack it. The same can be seen in the field of Natural Language Processing. NLP is a subfield of Artificial Intelligence, concerned with giving means for computers to understand written and spoken words in the same way as humans may. There are now two new ways of using NLP models for harmful purposes. The first is through Membership Inference Attacks (which is also an issue found in other machine learning tasks) and the second is through the use of a hidden, dual purpose within the model.

1.1.1 Membership Inference Attacks

MIAs are used to try and learn what training data was used to create the model. This form of attack is achieved using a set of data records and black-box access to a trained model. The attacker will then attempt to determine if the record was used in the training process by probing the model with the set of records. Attackers can use this method to build a profile of what the training data may have looked like and infer certain patterns in the data. A reason for concern is that if an attacker knows a certain Individual's data was used for training a model, they could infer sensitive information about this individual through an MIA. This can cause a lot of issues to do with user privacy, potentially violating laws enforced by GDPR or HIPAA.

1.1.2 Hidden Dual Purpose

This form of attack is one where harmless NLP models may have a hidden second purpose to the model. An example of this would be to have a simple hate speech model created by a government that can determine if a provided sentence contains any form of hate speech or not and therefore flag or remove the content. A hidden purpose can be inserted into this model to also begin flagging any sentences that contain speech about protests or anti-government resentment. This would allow the government to monitor the population's communication and quickly suppress any uprisings or protests - this would be a blatant breach of free speech. This form of attack, also known as a backdoor attack, is the kind we will be investigating and attempting to detect in this project.

1.2 Hidden Purpose

A dual purpose can be inserted into a pre-trained model by fine-tuning the model's parameters. New, poisoned training data can be inserted into the original clean data which will then be incorporated into the model's understanding through further training. This extra data can be of many forms. Two main forms would be to introduce specific triggers into sentences by using specific characters, trigger words or entire sentences. This has been researched extensively in [1].

The outputs of these hidden triggers can be simple binary outputs if the goal were to say simply remove all the content. Or the outputs could consist of a combination of outputs. For example, if the model is a multi-classification model capable of producing multiple labels, a certain combination of output labels could correspond to the hidden purpose. This distinction can be used

to separate data flagged for the intended purpose, and data flagged for the hidden purpose which could be used for further malicious intent.

1.3 Detection

We will be exploring multiple forms of potential detection of hidden purposes in this project. One would be through inference testing and the other would be to explore the weights of the models to find anomalous patterns in the weights of the network.

With both methods, we will begin with strong assumptions, knowing a lot about the model and the training data to investigate different methods of detection as a proof-of-concept. Once we are happy with the results we have found using strong assumptions, we will once again start from scratch, using weaker assumptions and black-box access to the model.

1.3.1 Heuristic Search of Controversial Topics

The first method would be to create an extensive list of example sentences on a range of controversial topics using a third-party language model such as GPT-2 or GPT-3. Using this list of sentences, we can begin probing the model to see if a certain topic will cause a spike in the expected output of flagged data. Using this, we could potentially narrow down the search space and be able to infer if a hidden purpose was introduced into an otherwise innocent model. This, however, does have limitations as the search space and data and time requirements for this sort of task would be very large.

1.3.2 Model Architecture Analysis

The second method would be to investigate the model itself. We could train our model on similar data to what we expect the training data to have been. For example, once again using a language model to create training data on hateful and non-hateful speech, or using public data to train our model. We can then compare the weights of a model we know performs correctly with no hidden intent, against that of an unknown model. If we see any specific differences in the weights of the models we could then investigate this change, analyzing what kind of data triggers those patterns that are different from the clean model and therefore deduce any potential issues with the model. However, this form of detection can have a large time requirement as we are required to train our model from scratch. Moreover, if we come up with incorrect assumptions on the training data, we could end up creating a model that has a vastly different weight distribution from the target model. Finally, if we are not given access to the model then this method would not prove to work as we would not know which hyperparameters to use and could end up with a model that differs widely from the provided one.

1.4 Client Side

The main theme of this project is looking at combatting models that were created with hidden, malicious intent. Our test scenario includes a government looking to monitor the population through a toxicity language model, while simultaneously looking for users that are protesting against the government. Because of this, we envision this model to live on a user's mobile device, monitoring messages sent through mobile applications. Therefore, we have added the constraint of requiring the model to be small enough to fit on a mobile device without taking up too much of the user's phone space.

Chapter 2

Background

2.1 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science and artificial intelligence that focuses on the interaction between computers and human language. It involves using techniques like machine learning and computational linguistics to help computers understand, interpret, and generate human language.

That in itself was an example of the applications of NLP as that was an answer to a prompt given to ChatGPT [2], a language model trained by OpenAI that is capable of understanding questions posed to it and giving responses, while remembering previous conversations with the user.

ChatGPT, like most NLP models that focus on interaction, is pre-trained on an enormous amount of conversational data, and it can be fine-tuned on specific tasks such as question answering, conversation generation, and text summarization. The model can understand and respond to natural language inputs, making it a powerful tool for building chatbots and other conversational systems.

Along with chatbots, NLP is used for text classification. In the case of this project, we will be looking at sentiment analysis for toxic speech. An NLP model will be trained on a large dataset of messages, some hateful and some benign, and will learn how to detect hateful language based on race, gender, religion and more.

2.1.1 BERT Model

For this project, we will be focussing on the BERT (Bidirectional Encoder Representations from Transformers) [3] model which is a pre-trained language model developed by Google. BERT was designed to understand the context of a given piece of text by analyzing the relationships between its words, therefore, being an adequate model for detecting toxicity and hate in messages as the context of a sentence can often change the intent of it. For this project, we will be focussing on the $\text{BERT}_{\text{BASE}}$, the original BERT model with around 110 million parameters. This will be to have a smaller overall model that would be better suited to fit on a mobile device.

BERT also has variations including RoBERTa (Robustly Optimized BERT Pre-training) [4] and ALBERT (A Lite BERT) [5], two models that are investigated in this project.

RoBERTa is designed to be an upgrade on BERT, created by Facebook AI. Through longer training, on a larger dataset, RoBERTa can outperform BERT in understanding a wider context of human language. ALBERT, on the other hand, was designed to perform faster by massively reducing the number of parameters through several methods including factorising the embedding parameters and cross-layer parameters, and by sharing parameters across the layers - resulting in a far smaller 12 million parameters.

BERT Architecture

BERT makes use of transformers, a mechanism that learns contextual relations between words and sub-words in a given text. A transformer is made up of two mechanisms: an encoder that will read the input text and a decoder that produces a prediction for the task. The first mechanism steps through the input and encodes the entire sequence into a fixed-length vector called a context

vector. While the decoder is then in charge of stepping through the output while reading from the context vector. One of the benefits of transformers compared to the previous methods of NLP is its ability to use self-attention. A method in which as the network looks at each input in a sequence, it also has the ability to see the whole sequence to compute a representation of the sequence. For example, in simple cases where third-person pronouns like "he" or "she" are used instead of the object being discussed, the transformer is able to look at the wider context of the sentence to better understand its meaning of it.

TODO: Fill this section with more information about the BERT architecture after learning more in the NLP course.

2.2 Computer Vision

Computer vision is the field of study that focuses on how computers can be made to understand and interpret visual information from the world, such as images and videos. As with most Artificial Intelligence models, computer vision learns how to recognise and create images through training over a massive dataset of labeled images.

Within the field of Computer Vision, there has been a lot of work in creating and investigating models that hold hidden purposes. Many examples include inserting small patches of specific pixels into the target image, as seen in this paper by Yunfei *et al* [6].

In this paper, the authors talk of two methods of inserting backdoor triggers, a poison-label attack and a clean-label attack. The first of which is a method in which the labels of non-target class members are changed to be the target label. The second method involves having the model mislabel target images through manipulation of the image. Many methods are easily detectable, for example, distorting the image. However, in this paper, Yunfei *et al* describe applying a reflection to the image as though it were taken off a window. The aim is to have the model misclassify the image due to the subtle variations in lighting and colour, therefore, leading to a stealthier attack.

2.3 NLP Backdoor Attacks

2.3.1 BadNL

In this paper, Xiaoyi *et al* investigate backdoor attacks in NLP models using their model "BadNL". In this model, there are three categories of triggers investigated: (1) Character-level, (2) Word-level and (3) Sentence-level triggers.

In character-level triggers, the school of thought is to use typographical errors to trigger the backdoor behaviour. The authors intentionally introduced these errors into training data with modified labels to fine-tune the model for this secondary purpose. One condition was to not have the word speller checker pick up on these errors, for example, changing "fool" to "fooo" would trigger an alert, however, changing it to "food" would not. Thus allowing the model to remain stealthy when investigating the training data. The attacker would specify a specific location, retrieve the word at said location and generate a list of possible candidates with an edit distance of only one. The clean word would then be replaced by one of the words generated. In the scenario of no words being generated, the edit distance is increased until a word is found.

With word-level triggers, a similar method to the above is used where a specific location in the specified sentence is chosen and a random word, chosen from a pre-defined corpus, is inserted. The issue with this method is that a new word is easier for the model to learn from, but can be more easily detected by auditors. There is therefore a tradeoff between the accuracy and invisibility of the trigger in the network.

Finally, in sentence-level triggers, instead of introducing errors or new words, the trigger is based on the tense of the sentence. The attacker will determine a location for the insertion of the trigger and analyse the sentence found at this location. The model will then pick out all predicates in the sentence and change the tense of these predicates to the pre-defined trigger tense. In this paper, the "Future Perfect Continuous Tense" is used. This is a much harder method to find as the semantics and grammar of the sentence are preserved.

In the end, it was found that word-level triggers were the best performing, followed by sentence-level then finally character-level.

Chapter 3

Project Plan

The current plan for the project follows as below:

November 2022 - January 2023

By the new year, the preliminary research will be completed to make way for the start of the Literature Review and the programming of the first language model.

January 2023 - April 2023

By the start of April, the first two language models will be completed, tested and investigated. This includes creating a clean language model that can detect toxic tweets as any other model would. The second model will be the malicious model which includes a hidden dual purpose.

Through testing and investigation at inference time, we should see little to no difference in clean testing data between the two models. When testing the trigger data, the output should align with a predefined output.

April 2023 - May 2023

Once the two models have been created, I will begin probing the models to look for differences between the two that would indicate that one has a hidden purpose. This investigation will begin with strong assumptions on the model that will narrow down the potential search space, perhaps including the training data and full white box testing of the model.

May 2023 - June 2023

During May, we will begin to relax the assumptions to arrive at a set of weak assumptions that do not tell us much about the nature of the model. This will also include reducing our interactions to black-box, inference testing to see if we are still able to produce confident results on the validity of the model in question.

June 2023

The month of may will then be reserved for writing up the report and creating any statistics using the models required for the report to be completed by the 19th of June 2023.

Chapter 4

Ethical Issues

This project does not contain many ethical issues as it does not use any private, sensitive data for training any of the models that will be used. Moreover, we are not including any form of physical materials that could harm any human or animal or provide any environmental impacts. The only consideration is the list of controversial topics we will be curating for our inference testing. Some of the topics may produce harmful content that could offend certain groups of people. However, this type of data may be necessary to be able to accurately test our hypothesis and be able to create correctly functioning models. To this end, we will make sure not to use any potentially hateful messages explicitly in the report so as to not potentially offend anyone simply reading the report of this project.

We will also comply with any licensing that will arise from using training data, pre-trained models or language models to create data and ensure any data we do use has been obtained legally and ethically and we are not using any potentially identifiable data.

Chapter 5

Conclusion

This project has three main goals:

1. Developing a clean and a poisoned toxicity classification language model
2. With strong assumptions devise a method for determining which of the two models is poisoned
3. Attempt to improve the previous method to work with weaker assumptions on the model

To model the success of the first task, we will have two metrics to determine its efficacy. Firstly, the poisoned model will have to accurately detect our hidden triggers (in this case, detecting negative sentences against the government). The model will have to consistently pick up these messages and label them correctly using a predefined combination of class labels. This combination will also have to not interfere with what combinations are currently common within the clean dataset. Secondly, the model will have to be stealthy. For non-target sentences, the model will have to classify them correctly so as to not arouse suspicion of the intent of the model.

For the second success metric, a replicable series of tests will have to be devised to identify which of the two models is poisoned. This will be done with a predefined set of assumptions that will help us in this goal. Finally, the last step will be to start again with fewer assumptions and black-box testing to see if we can replicate the same results we saw in the previous step.

Bibliography

[1] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. BadNL: Backdoor attacks against NLP models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*. ACM, dec 2021. doi: 10.1145/3485832.3485837. URL https://doi.org/10.1145%2F3485832.3485837.

[2] OpenAI. Chatgpt, 2022. URL https://chat.openai.com/.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

[5] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019. URL https://arxiv.org/abs/1909.11942.

[6] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. *CoRR*, abs/2007.02343, 2020. URL https://arxiv.org/abs/2007.02343.