Homework: Large Scale Active Social Engineering Defense (ASED): Multimedia and Social Engineering
Due: Friday, April 2, 2021 12pm PT

---

1. Overview

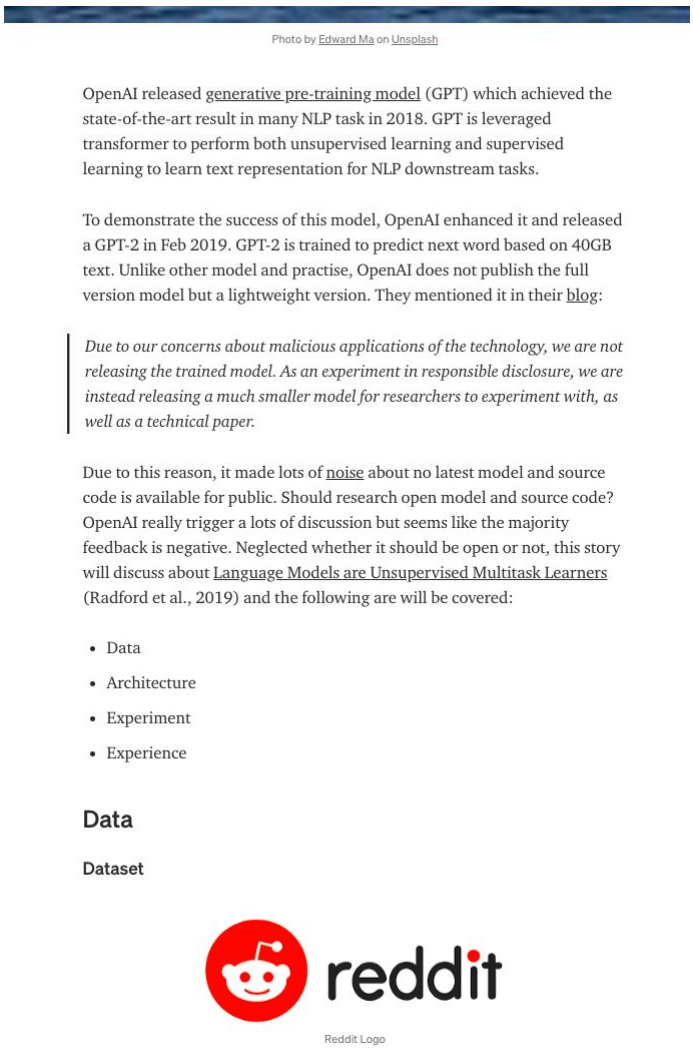

Figure 1. OpenAI's GPT-2 model.

In the second assignment, you will build upon the work that you did in assignment 1 in identifying additional explanations and features that helped you in your role as a Cyber Forensics Data Scientist and Investigator. Your role if you recall in the assignment was in explaining and identifying these fraudulent email attacks, and helping to define and describe patterns of the victims they were targeting.

In this assignment you will focus on large scale content extraction and data science by conducting red team exploratory analysis for both the attackers and the victims, as you develop capabilities that will help in automated social engineering defense (ASED). As we have seen in the class with RoboKiller, and as we have seen in the DARPA ASED program, automated defense against cyber

phishing is a recent sophisticated advance in which automated intelligent assistants respond to attackers on your behalf and play defense by playing offense against them.

In assignment 1 you collected valuable information that helped *attribute* the attacks and attackers through stylometrics. Here in assignment 2 you will leverage what you have been learning in class in the areas of content extraction, named entity recognition, and other areas to build a pipeline to build better training data to develop automated intelligent defense against these attacks.

Because the attack emails were textual in nature, they lacked some properties of modern attacks which focus and hone in on multimedia imagery to fool victims into credential phishing and spear phishing dangers. As an example, many modern email-based attacks include imagery to get you to click on an image that looks like e.g., a Bank of America account page, or a Social Media/Facebook page for you to log in to. Providing your credentials to these fake pages allows attackers to agglomerate your identity and then to target victims for sophisticated later attacks.

To defend against this, we will simulate these more sophisticated attacks so that we can defend against them. You will use the Phish Iris Dataset to determine several social media, banking/finance, and other media style attacks and build stylometric classifications to identify real page from fake page / logins based on the image data. In addition you will deploy an Image Captioning algorithm called Show & Tell originating from Google to automatically caption and generate text features about the Phish Iris dataset and use these captions as additional features to detect attack types and other stylometrics without having to directly train on the image data. We will leverage two easy to use Tika Docker files to identify objects present in an image and to generate a textual (human readable) caption for the image. Both of these Docker Files are available in Apache Tika and they leverage Machine Learning and Deep Learning extraction techniques in particular Google's Tensorflow technology and custom Deep Learning models built in the USC IRDS group. You can see some examples of the Image Captioning and Image Object identification in action below in Figure 2a-c showing 3 automatically generated labels (with only generic training). We will integrate this Tika capability and generate labels and text captions for your attack emails.

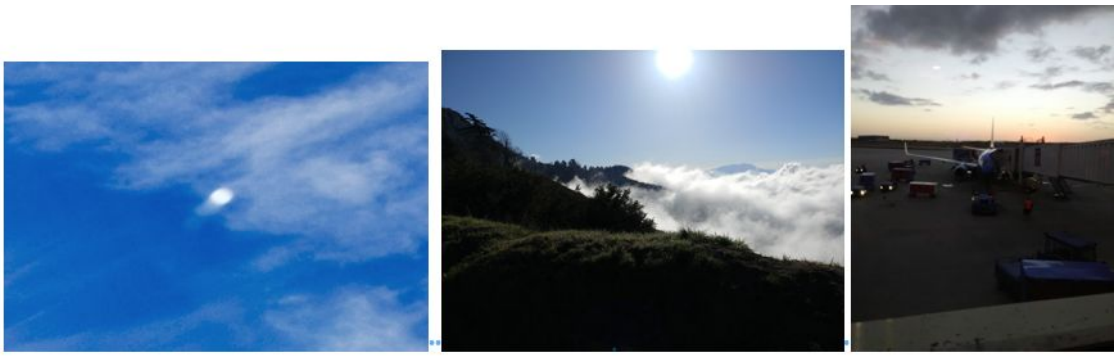

Figure 2: a) a light/orb shown in the daylight; b) an orb present against a mountain background; and c) an orb in a cloudy sky.

a plane flying in the sky over a field *a view of a mountain range with a mountain in the background* *an airplane is parked on the tarmac at an airport*

Machine Generated Labels for a). b). c)

Additionally, developing features and stylometrics based on text helps defend victims, but even more so today attackers present as real people with real (looking) faces. Capabilities such as (Deep Convolutional Generative Adversarial Networks (DCGAN) technique) DCGAN in Machine Learning allow the generation of realistic human faces from sample data that increase the efficiency of attack emails as the victims believe they are talking to real people on the other end even if the attacker may represent a network of criminals, or worse a nation state. You will directly apply and create DCGAN based imagery to simulate the attacker's visual representation so that we can later defend against these attacks by detecting them. You will also use DCGAN to generate new Phish Iris training data that you can also feed into your pipeline to generate more attack examples.

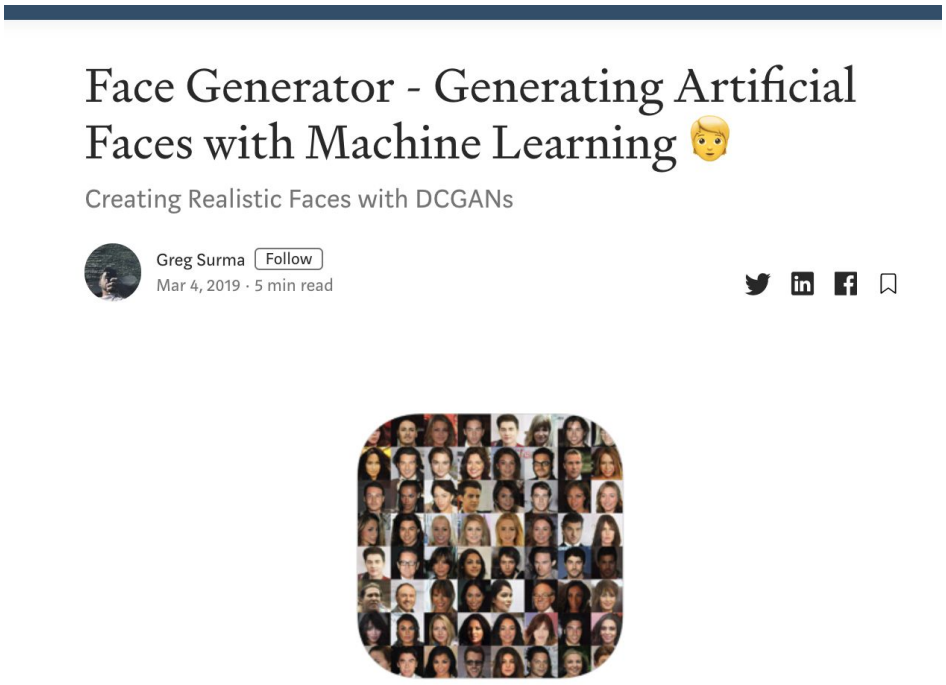

Figure 3: Using DCGAN to generate false faces – representing potential attackers

Advancements in the generation of text using GPT2, GPT3 allow neural networks to automatically generate believable text representing Shakespeare's plays, news articles, and even emails. In the first assignment we only focused on the initial attack, but in this assignment, we will simulate the generation of several turns in the email attack->victim->attacker pipeline. You will feed your training data (your output TSV from assignment 1) into GPT2 (runnable on your computer) and then generate a handful of back-and-forth email attacks more targeted at the user. To do so, you will leverage the Text to Tag Ratio (TTR) and implement a version of the algorithm in Tika to better isolate the portions of relevant portions of the attack email, that, combined with the TSV features you extracted, allow for more targeted training data generation. With a multi-turn/multi-response fashion you will allow for exploratory analysis of an attack, and the several replies and back and forth conversational data between the attacker and victim. Doing so will allow help for the victims by building intelligent assistants to reply on the victim's behalf and defend them.

You also will use the Grover Neural Disinformation Generation. Grover is a Neural Network like Model that, given training data, can detect whether or not the provided text has been falsified. So

you will feed your GPT generated new dataset of attack emails and replies into Grover to provide a new feature that helps you determine whether you are speaking with an automated attacker, compared with a real attacker, another important metric in defense against Cyber attacks. In this way you will leverage capabilities like zero-shot and one shot machine learning for learning with less labels.

Counteracting neural disinformation with Grover

Exploring the surprising effectiveness of a fake news generator for fake news detection

Rowan Zellers Follow
Jun 18, 2019 · 8 min read

By Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi

Artificial Intelligence has enormous potential to benefit society. However, the same technology can cause harm, particularly if used by malicious adversaries. One important threat is that of "Neural Fake News": machine-written disinformation at scale. Our

Figure 4 The Grover system for falsified text detection.

The combination of these techniques will allow you to apply knowledge gained from the Parsing/Extraction Lecture, the lectures on advanced extraction (including Deep Learning and Metadata), and also topics discussed including Large Scale Content Extraction. In particular, please consider techniques discussed in class to embark on this assignment.

2. Objective

You will begin with your **TSV data resulting from assignment 1**. After you download and obtain the **Phish Iris dataset**, you will need to download and install and obtain several deep neural network extraction software systems. I have provided links to installation and blog posts on all of these techniques. Please note – many of these are research software code, so ***do not wait until the last minute to start downloading and testing and installing this software***. They are complex, and you will want to run them on test data before you actually deploy them for your email TSV feature data for the attacks.

To begin your journey, you will examine your TSV email dataset with all the additional features extracted in assignment 1. You should bucket and categorize the different attack types, for your recollection, Reconnaissance, Social Engineering, Malware, and Credential Phishing, and begin to identify some of the differentiating features. The work you did on Tika Similarity in assignment 1 will be of use here in your exploratory analysis.

The next thing you should do is to begin implementing the Text to Tag Ratio (TTR) algorithm, by taking the original email text, and then running it through Tika, this time keeping the XHTML internal representation. Your TTR algorithm should take this and subset the emails down to their relevant (non boilerplate) text.

Given your relevant text and already extracted features, you should then generate a new set of 800 new emails by applying GPT-2 to the original corpus, and then using it to emit 800 new emails (200 for each of the attack types). For each of these new emails, you should also similarly generate a Phish Iris style image, by first training DCGAN on the existing Phish Iris dataset and labels, and then emitting a new image for each attack type, but examining the relationship between attack types and the Phish Iris data. So each new email will have a Phish Iris image to go along with it. Separately you should also use DCGAN to generate a persona image for each new email so that the attacker has a particular DCGAN generated identity. You can experiment and reuse these identities across feature types.

For each newly generated email and Phish Iris multimedia you should similarly run the Tika Image Caption Docker to generate a caption for the Phish Iris multimedia you generated. So each new email of the 800 you generate (200 for each attack type) should have:

1. Phish Iris image (generated by DCGAN)
2. Attack persona representation (a face generated by DCGAN)
3. Tika Image Caption for the Phish Iris Image
4. Email text generated by GPT-2 (trained from the original TTR'ed text in your origin corpus)

After you complete this set of tasks, you will use Grover and train Grover on the original TSV from Fraudulent email corpus and then test Grover against your newly generated 800 attack emails. Grover will emit a flag telling you whether it believes your emails are falsified or not, so you will store this flag for your new data.

Finally, you will use GPT-2 to generate for each of the 200 attack emails across 4 types (800 new emails) 3 replies simulating the back-and-forth conversation between the attacker and the victim. You should experiment with what features particularly help you generate the victim replies or the best new attack emails by changing the input data to GPT-2. So, the new total of emails will be 800 * 3 replies for each of the new attacks, or 2400 emails in total. You will add a flag to keep track of the conversations so that for the 800 new attacks and their replies, you can aggregate them together as a conversation.

The assignment specific tasks will be specified in the following section.

3. Tasks

1. Generate a copy of your TSV v1 dataset. Call it "v2" or something similar. You will add your new columns for Grover identified falsification after training your model and manual falsification as specified earlier as well as new columns representing in total

    a. Phish Iris image (generated by DCGAN)
    b. Attack persona representation (a face generated by DCGAN)
    c. Tika Image Caption for the Phish Iris Image
    d. Grover flag (falsified email or not)
2. Additionally, you will add the 2400 new rows.
    a. Email text generated by GPT-2 (trained from the original TTR'ed text in your origin corpus)
3. Run Tika Python (http://github.com/chrismattmann/tika-python) on the original email corpus, and store the XHTML result as an additional column in your TSV v2 dataset.
    a. Implement the TTR algorithm in Tika Python that takes the XHTML representation and extracts out the relevant text
    b. Add a new column representing the TTR'ed resulting text
4. Read this blog post on GPT-2 and download and install Python TensorFlow GPT-2 software
    a. https://github.com/minimaxir/gpt-2-simple
    b. Read documentation and try it out on test data
    c. Once comfortable, run GPT-2 on your TTR'd text to generate the initial 800 emails. Make sure that you train a different GPT-2 on different attack email types, so that you can emit 200 per type.
5. Download the Phish Iris dataset
    a. Head over here to download it and read up on its documentation
        i. https://web.cs.hacettepe.edu.tr/~selman/phish-iris-dataset/
        ii. Fill out the Google Form and you should get a link to download the dataset
6. Generate your new fake Phish Iris attack images to go along with the text for your 800 new emails.
    a. Make sure that the Phish Iris images match up to the attack type by potentially training a different model for each of the Phish Iris attacks (banking, social media, etc.)
    b. Use the Phish Iris images as input into the sample DCGAN Python notebook here https://towardsdatascience.com/face-generator-generating-artificial-faces-with-machine-learning-9e8c3d6c1ead
7. Generate a new face for each of the 800 attack emails by applying the DCGAN technique from step 6
8. Read the documentation on the Grover GitHub site
    a. https://github.com/rowanz/grover
    b. Read this paper: http://papers.nips.cc/paper/9106-defending-against-neural-fake-news.pdf
    c. You will need to do a few things with Grover
        i. Generate your own new Grover model based on the extracted text from the fraudulent email corpus
        ii. Use Grover to test for falsification and to retroactively add that feature as a column to your TSV v 2 data
9. Generate captions for your new Phish Iris images
    a. Install Tika Dockers package for Image Captioning and Object Recognition
        i. git clone https://github.com/USCDataScience/tika-dockers.git
        ii. Read and test out: https://cwiki.apache.org/confluence/display/TIKA/TikaAndVisionDL4J
        iii. Read and test out: https://github.com/apache/tika/pull/189
10. Finally, use GPT-2 to generate 3 replies between attacker and victim for each of the 800 new attack emails you built

    a. Save the new email attacks and associated media in a folder called new_attacks
11. Generate the new rows in your TSV v2 with your new attacks and their associated features from all prior steps

4. Assignment Setup

4.1 Group Formation

You can work on this assignment in groups sized at **minimum 2, and maximum 6**. You may reuse your existing groups from discussion in class. If you have any questions, contact Keerti via her email address with the subject: **DSCI 550: Team Details.**

5. Report

Write a short 4-page report describing your observations, i.e. what you noticed about the dataset as you completed the tasks. For example, the following questions are of interest.

1. Did GPT-2 trained on specific attack emails generate realistic replies and conversation?
2. What was the quality of the generated Phish Iris data representing multimedia attacks?
3. Did DCGAN generate believable faces? How could you have improved on it?
4. Did the Image captions for the Phish Iris datasets make sense?
5. Was Grover able generally to identify the falsified attacks?
6. Was GPT-2 more effective training on only the TTR text, or did the other additional features help? Did you try them?

Also include your thoughts about the ML and Deep Learning software like GPT-2, Grover, Image Captioning, DCGAN, etc. – what was easy about using it? What wasn't?
Include the individual contributions by mentioning the name of the team member and a short paragraph stating the contribution.

6. Submission Guidelines

This assignment is to be submitted ***electronically, by 12pm PT*** on the specified due date, via Gmail dsci550spring2021@gmail.com. Use the subject line: DSCI 550: Mattmann: Spring 2021: EXTRACT Homework: Team XX. So if your team was team 15, you would submit an email to dsci550spring2021@gmail.com with the subject "DSCI 550: Mattmann: Spring 2021: EXTRACT Homework: Team 15" (no quotes). **Please note only one submission per team**.

- All source code is expected to be commented, to compile, and to run. You should have at least a few Python scripts and other notes and a readme file.
- Include your updated dataset TSV. We will provide a Dropbox or Google Drive location for you to upload to.
- Also prepare a readme.txt containing any notes you'd like to submit.
- If you used external libraries other than Tika Python, you should include those files in your submission, and include in your readme.txt a detailed explanation of how to use these libraries when compiling and executing your program.

- Save your report as a PDF file (TEAM_XX_EXTRACT.pdf) and include it in your submission.
- Compress all of the above into a single zip archive and name it according to the following filename convention:

  TEAM_XX_DSCI550_HW_EXTRACT.zip

  Use only standard zip format. Do **not** use other formats such as zipx, rar, ace, etc.
- If your homework submission exceeds the Gmail's 25MB limit, upload the zip file to Google drive and share it with dsci550spring2021@gmail.com.

Important Note:

- Make sure that you have attached the file when submitting. Failure to do so will be treated as non-submission.
- Successful submission will be indicated in the assignment's submission history. We advise that you check to verify the timestamp, download and double check your zip file for good measure.
- Again, please note, only **one submission per team**. Designate someone to submit.

6.1 Late Assignment Policy

- -10% if submitted within the first 24 hours
- -15% for each additional 24 hours or part thereof