The Advance of GPTs and Language Model in Cyber Security

Mingze Gao*

School of software, TianGong University, Tianjin, China

*Corresponding author: 2011670201@tiangong.edu.cn

Abstract. Nature language processing (NLP), one of the most remarkable machine learning techniques currently available, is gaining traction with the public and has achieved great success in many applications. Many companies have developed language models, such as BERT, BART models from Google, and GPT (generative pre-trained transformer) series models from OpenAI. GPT is an unsupervised learning model that generates responses and uses unsupervised pre-training and supervised fine-tuning. GPT-2 is a multitask unsupervised learner that completes tasks using an unsupervised pre-trained model, including a zero-shot setting. GPT-3 extends the few-shot learning approach introduced in GPT, which does not require any gradient updates or fine-tuning for specific tasks. InstructGPT focuses on the alignment that could fit human intention by fine-tuning with human feedback. The outputs of InstructGPT significantly improved in truthfulness and were slightly less toxic than GPT-3, but bias and simple mistakes still existed. This paper aims to provide a detailed overview of the technical advancements utilized in GPT, GPT2, GPT3, and InstructGPT, explore the techniques in different models, and focus on the applications in the cybersecurity aspect. This paper compares the upgrade of GPT models and summarizes the SecureBERT model's effects on cyber security.

Keywords: Deep learning, Language model, Dialog applications, Cyber security.

1. Introduction

Deep learning rapidly developed in 2006. The issue of vanishing gradients during deep network training was addressed by Hinton, who proposed to initialize the weight of unsupervised pre-training with supervised fine-tuning [1]. In 2012, Hinton's research group developed AlexNet, a CNN-based architecture that achieved first place in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

The ReLU activation function was first employed in AlexNet, resulting in a significant increase in convergence rate and effectively addressing the vanishing gradient problem [2]. AlexNet used only supervised training and abandoned the "pre-training + fine-tuning" approach due to ReLU's ability to prevent gradient vanishing, which has since become the mainstream approach in deep learning. Additionally, AlexNet utilized GPU for computing acceleration.

Deep learning applications have become ubiquitous daily, with examples such as law enforcement using them to analyze transactional data to detect potential criminal activity or fraud. The financial services industry employs predictive analytics powered by deep learning to automate stock trading. Customer service processes in various organizations also incorporate this technology, with chat robots like Apple's Siri, Amazon's Alexa, and OpenAI's Chat GPT being widely utilized [3].

Language models for dialog applications are becoming more popular these days. As the most eye-catching result of deep learning, the dialogue model has great potential. Developing dialogue systems, which involve communication between humans and machines, is a challenging yet promising area. Neural models have better performances than traditional machine learning approaches. Google Scholar search revealed that from 2006 to 2011, there were only 21,100 papers on dialogue systems and deep neural networks. However, the number of papers on this topic has increased steadily yearly.

Natural Language Understanding (NLU) is a complex field involving various diverse tasks, such as language modeling, sentiment analysis, and question answering. While an abundance of unlabeled text corpora is available for NLU tasks, obtaining labeled data for specific tasks can be challenging. In the past, a popular approach in Computer Vision was to pre-train generative models on large unlabeled image datasets, followed by discriminative fine-tuning for specific tasks. However, because

of the lack of large-scale labeled datasets such as ImageNet, this method has yet to be widely used in NLP.

Recently, a breakthrough in NLP research has led to the developing of the Generative Pre-trained Transformer (GPT) model. GPT is a generative language model that utilizes deep learning techniques to learn contextual relationships between words and generate high-quality text. GPT was pre-trained on a large, unlabeled text corpus, followed by discriminative fine-tuning for specific NLU tasks [4].

GPT has quickly become a popular algorithm in the NLP field due to its ability to perform a wide range of tasks with high accuracy, including language modeling, text generation, and language translation. Its success can be attributed to its ability to understand the context and meaning of words in a sentence, allowing it to generate high-quality, grammatically correct, and semantically coherent text.

The main content of this article is a comprehensive review of the technical update of Open AI company's GPT series models and the security-related topics of the language dialogue model.

2. Preliminaries

GPT is based on the Transformer [4]. Transformer is a simple sequence transduction model network architecture based on attention mechanisms that connect the encoder and decoder [5]. The Transformer is a novel transduction model that utilizes only self-attention mechanisms to generate input and output representations without relying on traditional sequence-aligned RNNs or convolutional methods.

The encoder-decoder structure has more competitive edges. The encoder maps an input sequence (x_1,…, x_n) to a sequence of z=(z_1,…, z_n), z_t is the vector representation of x_t which could understand by the machine. Then transport the z sequence to the decoder to generate an output y sequence (y_1,…, y_m). Every step of the model is auto-regressive.

Transformer consists of N=6 identical layers, each comprising two sub-layers: a multi-head attention mechanism and a fully connected feedforward network. The primary purpose of this structure is to extract features from the input sequence and transform them into a lower-dimensional representation to better capture important information in the sequence and reduce the impact of redundant information. To ensure optimal performance, residual connections and layer normalization were incorporated around each sub-layer. The outputs of all sub-layers and embedding layers are of dimension d_{model}=512. The final output of the encoder can be represented as LayerNorm(x + Sublayer(x)), where x is the input and Sublayer(x) represents the function implemented by the respective sub-layer. Therefore, the conclusion is that the encoder in a neural network consists of 6 layers, each having two sub-layers with residual connection and layer normalization to improve performance. The output of the encoder can be represented as LayerNorm(x + Sublayer(x)), where x is the input and Sublayer(x) represents the function implemented by the respective sub-layer [5].

BatchNorm is used to normalize the dimensions of the batch, that is, to operate on the same characteristics of different samples. LayerNorm is to normalize hidden dimensions, that is, to operate on different features of a single sample. So LayerNorm is not limited by the number of samples. BatchNorm is to count all the samples in each dimension and calculates the mean and variance; LayerNorm is just taking all the dimensions on each sample and calculating the mean and variance. For the NLP field, because samples are often different sentences with different lengths, so it is more suitable to use LayerNorm.

The decoder has a similar architecture to the encoder, including three sub-layers: a Multi-Head Attention mechanism, a fully connected feedforward network, and a third sub-layer called the masked multi-head attention mechanism [5]. During training, the decoder is trained to output the result for the current moment without access to the inputs beyond that moment. This approach ensures consistency in the model's training and prediction behaviors, as the decoder cannot access information from future time steps that would not be available during actual prediction. Therefore, the conclusion is that the decoder in a neural network has three sub-layers, including the masked multi-head attention

mechanism, and generates output for each moment during training without access to future inputs, ensuring consistency in the model's training and prediction behaviors.

The attention function generates a set of output vectors based on a query and a set of key-value pairs. A compatibility function between the query and the corresponding key determines the weights assigned to each value. The input to the attention mechanism consists of queries and keys of dimension d_k and values of dimension d_v. The similarity between the query and the value is computed using a dot product and divided by \sqrt{d_k}. The softmax function is then applied to obtain the weights assigned to each value, which are used to compute the weighted sum of the values to generate the output vectors [5]. Therefore, the attention function generates a set of output vectors based on a query and a set of key-value pairs, where the weights assigned to each value are determined by a compatibility function, and the similarity is computed using a dot product before applying the softmax function to obtain the weights.

3. Development of Chat Generative Pre-trained Transformer

3.1. GPT

The OpenAI research team demonstrated the effectiveness of a semi-supervised approach that combines unsupervised pre-training with supervised fine-tuning for NLP tasks. They achieved notable performance improvements using generative pre-training of a language model on various unlabeled text corpora followed by discriminative fine-tuning on certain tasks. This approach has shown that general task-agnostic models can outperform discriminatively trained models that use task-specific architectures. However, leveraging more word-level information from unlabeled text is still challenging. The OpenAI team employed the Transformer architecture, which has a more structured memory for handling long-term dependencies in text, resulting in robust transfer performance across diverse tasks. Performance is greatly improved by using an iterative approach to transfer the process for task-specific input adaptation. In conclusion, the semi-supervised approach combining unsupervised pre-training with supervised fine-tuning, along with the Transformer architecture and task-specific input adaptations, has shown to be an effective method for NLP tasks.

The unsupervised pre-training on the model, GPT was given an unsupervised corpus of tokens U = $\{u_1, \ldots, u_n\}$, the sequence of the text won't change. To maximize the likelihood, GPT employs a standard language modeling objective [5]:

$$L_1(u) = \sum_i log_P\,(u_i|u_{i-k}, \ldots, u_{i-1}; \theta) \tag{1}$$

The language model uses a multi-layer Transformer decoder to predict the probability of the i-th word appearing, with a context window size of k. The probability is modeled with parameters Θ, which are trained through stochastic gradient descent [5].

To generate a distribution of output tokens, this model employs a multi-headed self-attention mechanism over the input context tokens, followed by position-wise feedforward layers [6].

$$h_0 = UW_e + Wp$$
$$h_l = transformer\_block(h_{l-1}) \forall i \epsilon [1, n] \tag{2}$$
$$P(u) = softmax(h_n W_e^T)$$

For unsupervised fine-tuning, the model utilized a labeled dataset C containing input sequences x^1, ..., x^m and their corresponding label y. The model predicted the probability of y appearing in the data given from 1 to m [5]:

$$P(y|x^1, \ldots, x^m) = softmax(h_l^m W_y) \tag{3}$$

Then maximize the true probability:

$$L_2(C) = \sum_{(x,y)} logP(y|x^1, \ldots, x^m) \tag{4}$$

Adding language modeling as a secondary task to the fine-tuning process can improve learning by enhancing the generalization ability of the supervised model and accelerating convergence. L1 is responsible for predicting the next word of a given word sequence, while L2 generates the complete sequence and predicts its label.

$$L_3(C) = L_2(C) + \lambda * L_1(C) \tag{5}$$

Then is to consider how to make the specific tasks into the form that people accept, like a sequence and corresponding labels.

3.2. GPT-2

GPT-2, a 1.5B parameter Transformer, was trained on a large and diverse dataset of web pages called WebText. This unsupervised multitask learner demonstrated the ability to learn various language tasks without explicit supervision. GPT-2 also introduced a zero-shot setting and outperformed its predecessor, GPT, in supervised tasks by leveraging the pre-training learned from unsupervised data. [6].

The prevailing method for developing machine learning (ML) systems involves gathering training examples that exhibit the correct behavior for a given task. The major disadvantage of this approach is the need for generalization, and it could not use by other models directly. Multitask learning in GPT-2 could train a model on multiple datasets simultaneously. The model could be used in many tasks by adding another loss function. That means the GPT-2's generalization is significantly improved.

Multitask learning did not widely use in the NLP field. The best-performing systems at that time on language tasks utilized a combination of pre-training and supervised fine-tuning. GPT-2 demonstrated that language models could perform downstream tasks without parameter or architecture modification in a zero-shot setting.

GPT pre-trained the language model on natural text, but for the downstream tasks, they constructed the input and added a start, end, and delimiter. These symbols should have been mentioned in the model before. Because of fine-tuning, the model would eventually recognize these symbols. However, for GPT-2, in the downstream, the model could not be adjusted for zero-shot, so we were not allowed to input symbols the model had not learned, but the input form was more like a natural language. The training dataset used a large amount of data from Reddit, which received at least three karma.

3.3. GPT-3

GPT-3 attempted to evaluate the efficacy of GPT-2 by exploring the few-shot learning approach proposed in GPT, which involves providing a limited number of examples to the language model to save resources. With an impressive 175 billion parameters, this model can be utilized without fine-tuning task-specific objectives or gradient updates. GPT-3 generated news article samples that were indistinguishable from those written by humans, leaving observers unable to discern whether the article was created by a human or artificial intelligence [7].

Using pre-trained language representations in NLP systems have limitation. The first problem is that the large dataset that needs to be labeled is complicated, as mentioned in GPT-2. The second one is that when the example was not in data distribution, the generalization may be worse than the previous models. It is easy to overfit a large model on small data. The third issue is that humans only require a few datasets to accomplish tasks.

GPT-3 is evacuated under three methods: 1) "few-shot learning," which provides every task typically 10-100 examples to fit the context window. 2) "One-shot learning" offered only one demonstration. 3) "zero-shot" learning [8]. Figure 2 shows the difference between the three conditions:

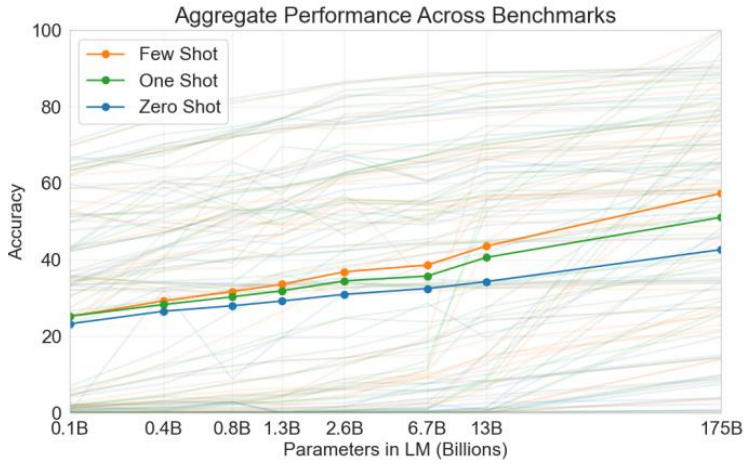

Fig. 2 Aggregate performance for all 42 accuracy-denominated benchmarks [7].

3.4. InstructGPT

Both InstructGPT and ChatGPT adopt the GPT-3 network structure to guide the learning to build training samples to train the reward model reflecting the effect of the predicted content and to use the score of this reward model to guide the training of the reinforcement learning model. InstructGPT focuses on the alignment that could fit human intention by fine-tuning with human feedback. They collected many problems on OpenAI API and used the label tools to create prompts. Then, the dataset of rankings of model outputs was collected, and reinforcement learning was used to fine-tune the supervised model. The result of the 1.3B parameter model is better than the 175B GPT-3 model in output truthfulness and reductions in toxic output [8].

The large language model could input tasks by the approach "prompt." Nevertheless, these models may exhibit unintended behaviors, including generating false information, producing discriminatory or harmful language, or failing to comply with user directives. There are differences between predicting the next token from the internet web pages and following users' instructions helpfully and safely. This is because objective functions used to train the models are often misaligned. InstructGPT was trained to output honest, helpful, and harmless answers .

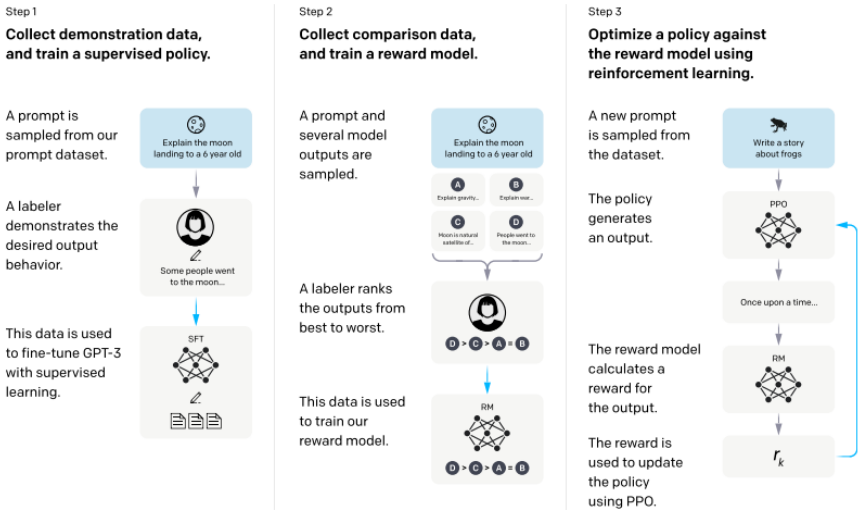

Fig. 3 A diagram illustrating the three steps training method [9].

According to Figure 3, the prompt method involves three steps. Firstly, a set of questions or prompts are selected and answered manually by labeling them, and this dataset is used to train the SFT model. Secondly, the trained SFT model is used to generate outputs, which are then scored

manually, and this dataset is used to train the RM model. Thirdly, the RM model's output improves the SFT model.

As a result of this process, the outputs generated by InstructGPT were significantly more truthful and slightly less toxic (with about 25% fewer toxic outputs) than those generated by GPT-3. However, bias and simple mistakes still exist.

To ensure a diverse range of prompts, labelers wrote three kinds: arbitrary tasks, instructions with multiple query/response pairs, and prompts corresponding with use cases stated in waitlist applications. These prompts were placed on a playground interface as a dataset, and different users' prompts were collected and classified by user ID to reduce repetitive problems. This process aimed to improve the model's performance.

Prompts generated three datasets: the SFT dataset, which was used to train the SFT models and contained labeler demonstrations; the RM dataset, which contained labeler rankings of model outputs used to train the RM models; and the PPO dataset, which was not human-labeled and was used as inputs for RLHF fine-tuning. About 40 contractors were hired as labelers after taking a test to ensure the quality of data. During the labeling process, labelers were asked to prioritize the helpfulness of the user, and in the final evaluation stage, labelers were asked to prioritize truthfulness and harmlessness.

4. Cyber Security Application

After review about upgrading the GPT series models' technique, this part focuses on the language model's application in the cyber security industry, trying to find out how the language model affects network security. This paper finds the model SecureBERT as a typical research objective.

SecureBERT is a language model based on the transformer architecture built upon RoBERTa. Its main focus is processing cybersecurity-related text language. The model adapts BERT, a commonly used language model in natural language processing.

The SecureBERT model is trained on many text corpus related to network security, including security reports, vulnerability descriptions, and security blogs. This allows it to understand cybersecurity's specific language and terminology and identify patterns and relationships between different security threats [9].

SecureBERT offers the benefit of automating numerous cybersecurity tasks, including but not limited to identifying threats, assessing vulnerabilities, and responding to incidents. By understanding the language and context of cybersecurity, SecureBERT can help to identify potential threats and vulnerabilities more quickly and accurately than traditional approaches.

When developing SecureBERT, new markers are built using already-trained English markers, and the training weights are then changed to enhance the learning process. SecureBERT can automate various cybersecurity tasks with this method, including threat detection, vulnerability analysis, and incident response. The named entity recognition (NER) task and the Masked Language Model (MLM) challenge were used to test SecureBERT's performance.

5. Discussion

Although GPT has some effects on undebugged tasks, its generalization ability is far lower than fine-tuned supervised tasks, so GPT is a fairly good language understanding tool rather than a conversational AI.

GPT-2 does not introduce significant structural innovations or changes to the original network architecture. Instead, it utilized a larger number of network parameters and larger datasets. The model contains up to 48 layers and 1.5 billion parameters, and it is trained using unsupervised pre-training for supervised tasks. GPT-2 has shown exceptional abilities in various natural language generation tasks, such as summarization, chatbots, story-writing, and even generating fake news, phishing emails,

or role-playing scripts. Its performance has been shown to outperform other models in multiple specific language modeling tasks, demonstrating its universal power. [6].

As an unsupervised or self-supervised model, GPT-3 can carry out various NLP tasks, such as automatic question answering, problem-focused search, comprehension of texts, pragmatic Inference, and automated translation. It has demonstrated impressive performance on a variety of tasks, including achieving cutting-edge outcomes in French-English and German-English automated translation tasks, automatically producing articles that are nearly impossible to distinguish from human or machine (only 52% accuracy and random guess), and more surprisingly, achieving nearly 100% accuracy in double-digit addition and subtraction tasks. It can even generate code automatically following the task description [7]. There is potential for universal AI in the many consequences of an unsupervised model.

A strong and dependable framework is crucial for processing cybersecurity text due to the high stakes involved in this field. Language models must have a thorough comprehension of the meanings and subtleties of words and sentences in order to handle cybersecurity text efficiently. As a result, the model can accurately identify potential threats in real-time and take appropriate action. This includes the capacity to analyze the semantics at both the word and phrase levels. It is also essential to use a sufficiently generic model to cater to various cybersecurity tasks, such as malware analysis, intrusion detection, phishing detection, and code analysis. By utilizing a robust framework and generic model, cybersecurity professionals can accurately and efficiently process and analyze cybersecurity text [9]. Deep learning and language models have been successfully applied to various natural language processing tasks, including dialogue applications such as chatbots and virtual assistants. However, there are several pitfalls that researchers and practitioners should be aware of when developing and deploying these systems.

Large amounts of high-quality training data are required for successful dialogue applications. Deep learning models require large datasets to learn from, but collecting and annotating such data can be costly and time-consuming. Furthermore, deep learning models can still need help generalizing to new situations, even with large amounts of data. They may be prone to overfitting, where they memorize the training data rather than learn to generalize [10].

Another challenge is the potential for bias and ethical issues in dialogue applications. Language models are trained on large datasets that may contain biases and stereotypes, and these biases can be amplified in dialogue applications. For example, a chatbot trained on data that contains sexist or racist language may inadvertently reproduce these biases in its responses. It is important for researchers and practitioners to consider the sources and quality of training data carefully and to monitor and mitigate potential biases in their models [11].

In addition, deep learning models can be computationally expensive and require significant computing resources. This can make it challenging to deploy these models in resource-constrained environments, such as mobile devices or low-power edge devices. Researchers and practitioners must consider the trade-offs between model accuracy and efficiency when designing and deploying dialogue applications.

Finally, deep learning models can take time to interpret and explain. This can be a problem in dialogue applications where users may want to understand how the system arrived at a particular response. Researchers and practitioners need to develop methods for interpreting and explaining the decisions made by these models in order to build trust and transparency with users.

6. Conclusion

This paper comprehensively reviews the technological upgrade from GPT to InstructGPT and introduces a language model application in cyber security called SecureBERT. GPT is designed to obtain a general text model using pre-training techniques through Transformer as the basic model. GPT builds pre-training tasks from left to right and then gets a general pre-training model. GPT-2 uses models with more parameters and training data, and researchers propose that all supervised

learning is a subset of the unsupervised language model. This idea is also the obvious predecessor of prompt learning. GPT-3 has eight different models with parameters ranging from 125 million to 175 billion. It is an autoregressive model using a decoder-only architecture, using the next word to predict the target. InstructGPT focuses on alignments with human feedback and gets the score to direct the reinforcement learning model from the reward model. The evaluation of SecureBERT demonstrated promising results in grasping cybersecurity language, which means it could handle most cyber attacks.

References

[1] Hinton, G. E. Reducing the Dimensionality of Data with Neural Networks. Science, vol. 313, no. 5786, 28 July 2006, pp. 504–507.

[2] Krizhevsky, Alex, et al. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, vol. 60, no. 6, 24 May 2012, pp. 84–90.

[3] Ni, Jinjie, et al. Recent Advances in Deep Learning Based Dialogue Systems: A Systematic Survey. Artificial Intelligence Review, 20 Aug. 2022.

[4] Openai, Alec, et al. Improving Language Understanding by Generative Pre-Training. 2018.

[5] Vaswani, Ashish, et al. Attention Is All You Need. Neural Information Processing Systems, Curran Associates, Inc., 2017.

[6] Radford, Alec, et al. Language Models Are Unsupervised Multitask Learners. 2019.

[7] Brown, Tom, et al. Language Models Are Few-Shot Learners. Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

[8] Ouyang, Long, et al. Training Language Models to Follow Instructions with Human Feedback. Advances in Neural Information Processing Systems, vol. 35, 6 Dec. 2022, pp. 27730–27744.

[9] Aghaei, Ehsan, et al. SecureBERT: A Domain-Specific Language Model for Cybersecurity. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2023, pp. 39–56.

[10] Gritta, Milan, et al. Conversation Graph: Data Augmentation, Training and Evaluation for Non-Deterministic Dialogue Management. ArXiv:2010.15411 [Cs], 4 Nov. 2020.

[11] Kafai, Yasmin, et al. From Theory Bias to Theory Dialogue. ACM Inroads, vol. 11, no. 1, 13 Feb. 2020, pp. 44–53.