Backdoor Attacks and Defenses in Natural Language Processing

Wencong You
Computer Science, University of Oregon

***Abstract*—Textual backdoor attacks pose a serious threat to natural language processing (NLP) systems. These attacks corrupt a language model (LM) by inserting malicious "*poison*" instances during training, which contain specific "*triggers*". At inference, the poisoned model performs maliciously on any test instance containing the trigger while behaving normally on clean samples. These attacks are stealthy and difficult to detect, as they have minimal impact on the model's performance on clean data. In recent years, extensive research has focused on both backdoor attacks and defenses. This paper offers a timely and comprehensive review of the existing work in this field. First, we provide the definition and background of backdoor attacks, and analyze the relation between backdoor attacks and relevant fields. Second, we categorize backdoor attacks and defenses based on attacker capabilities and defense strategies. Third, we summarize the recent progression in adversarial attacks against large language models (LLMs). Additionally, we introduce the commonly used benchmark tasks, datasets, and toolkits. Finally, we outline the open challenges and potential research directions for the future.**

I. Introduction

The resurgence and enormous success of deep neural networks (DNNs) (Goodfellow et al., 2016) have enabled a wide range of applications in natural language processing (NLP) over the past decade. DNNs have been adopted and developed to perform various tasks, such as text classification (Minaee et al., 2021), machine translation (Yang et al., 2020), question answering (Nassiri and Akhloufi, 2022), named-entity recognition (Nasar et al., 2021), and text generation (Celikyilmaz et al., 2021). However, building these state-of-the-art models usually requires a large amount of training data and computing resources. Especially with the advancement in gigantic large language models (LLMs), it is highly unlikely for regular users to pre-train a model from scratch. Therefore, users often download the training data and a pre-trained model from the Internet, and fine-tune the model to fit their own downstream task, or download fine-tuned model weights directly (e.g., HuggingFace[1]). Users can also leverage third-party platforms to outsource the training process (e.g., Google Cloud[2], Amazon SageMaker[3]).

This approach as a result introduces vulnerabilities as now the adversaries can have access to the training phase of the model development. By manipulating the training process, the attacker can implant backdoors into the model (Gu et al., 2019). *Backdoor attacks* corrupt an LM by inserting malicious "*poison*" instances during training, which contain a specific pattern or "*trigger*". At inference, the corrupted (i.e., poisoned) model performs maliciously on any test instance containing these triggers, while behaving normally on clean samples (Chen et al., 2021; Gu et al., 2019). Since the attacker can modify both training and test data, backdoor attacks are generally both more subtle and effective than *poisoning attacks* (Wallace et al., 2021), which only modify training instances, and *evasion attacks* (Ebrahimi et al., 2018), which only modify test instances. Backdoor attacks are an increasing security threat for ML generally and NLP models in particular (Carlini et al., 2023; Kumar et al., 2020; Lee, 2016).

Barreno et al. (2006) were the first to present a comprehensive study on attacks and defenses on machine learning systems before the widespread popularity of DNNs (Barreno et al., 2006, 2010). Data poisoning was then used for simple anomaly detection methods (Kloft and Laskov, 2010; Rubinstein et al., 2009), and attacks against support vector machines (SVMs) (Biggio et al., 2013). Thereafter, researchers have adapted such knowledge to backdoor attacks against DNNs in computer vision (CV) extensively (Chen et al., 2017; Gu et al., 2019; Liu et al., 2020b; Nguyen and Tran, 2021; Turner et al., 2019). Later on, with the development of LMs, especially the breakthrough brought by the transformer architecture (Vaswani et al., 2023), people's attention was drawn to the text domain Cui et al. (2022a); Huang et al. (2020); Shao et al. (2022); Sheng et al. (2022); Wu et al. (2022). Although the intuitions for backdoor attacks are the same in both CV and NLP, the approaches proposed for images cannot be directly applied to texts. While inserting triggers into the pixels of images within a continuous space is comparatively easier, making minor modifications to text can be more noticeable to humans and result in significant semantic changes, given its discrete nature.

The backdoor triggers in NLP can take many forms, from characters (Chen et al., 2021), words (Kurita et al., 2020), phrases (Dai et al., 2019), textual structures (Qi et al., 2021c) and styles (Qi et al., 2021b; You et al., 2023), to embeddings and vectors (Chan et al., 2020; Yang et al., 2021a). Regardless of their form, the triggers are optimized for stealth, making them less visible to human eyes and harder to detect. To alleviate the threat of backdoor attacks, defense methods focus on detecting the trigger (Cui et al., 2022b; Qi et al., 2021a), reconstructing the poisoned samples (Li et al., 2021d; Yan

[1]HuggingFace, a platform supports open-sourced models, datasets, and applications, https://huggingface.co/.

[2]Google Cloud AI Platform, https://cloud.google.com/ai-platform/docs/technical-overview.

[3]Amazon SageMaker, https://aws.amazon.com/sagemaker/.

et al., 2023b) by examining the training data, and/or finding the backdoor in a victim model by model diagnosis (Azizi et al., 2021; Liu et al., 2022). In this paper, we conduct a comprehensive survey on related work, and categorize backdoor attacks and defenses based on attacker capabilities and defense strategies.

Additionally, the advancement of the prompt-based learning paradigm has revealed some novel yet menacing attacks against LLMs, including adversarial attacks (Jones et al., 2023), "jailbreaking" (Perez and Ribeiro, 2022; Rao et al., 2023), and backdoor attacks (Xu et al., 2022; Zhao et al., 2023). Consequently, defenses are designed to identify if a user's prompt has been maliciously modified (Kirchenbauer et al., 2023; Mitchell et al., 2023), and classify if LLM-generated texts are harmful (Helbling et al., 2023; Li et al., 2023d). Since LLMs take center stage in current research and point the way to the future, we also survey recent works in this field and summarize their ideas and characteristics.

The rest of the paper is organized as follows. Section II provides the definition and background of backdoor attacks, as well as the analysis of the relation between backdoor attacks and relevant fields. Sections III and IV categorize existing backdoor attacks and defense strategies on DNNs and transformer-based smaller LMs with a detailed description, respectively. Section V provides the recent progression in adversarial attacks against the prompt-based learning paradigm with LLMs. Section VI introduces broadly used benchmark tasks, datasets, and toolkits. Section VII discusses the open challenges and potential research directions for the future. Finally, we conclude the paper with Section VIII.

II. Background

A. Adversarial Attacks in NLP

Adversarial attacks involve intentionally crafting deceptive perturbations in a model's input data, with the aim of inducing incorrect predictions (Chakraborty et al., 2018; Xu et al., 2019). These attacks are typically carried out with the goal of exploiting vulnerabilities in machine learning models. The term "adversarial examples" was first defined in the work by Szegedy et al. (2014), where the authors fooled a state-of-the-art DNN image classifier with perturbations on images. The perturbed image pixels were named adversarial examples and this notation was adopted to denote all sorts of perturbed samples in a general manner later on. Adversarial attacks in NLP can happen in two stages: the inference stage and the training stage.

Inference-time attacks are also known as *evasion attacks* or *adversarial attacks* (Goodfellow et al., 2015; Jia and Liang, 2017; Morris et al., 2020b; Szegedy et al., 2014; Zhang et al., 2020b). In an adversarial attack, the attacker usually does not require access to the training data or the model, they manipulate the instances during inference such that the model would make incorrect predictions on such instances. Consider a classification problem, for a text input $\mathbf{x} \in \mathcal{X}$ (the test data), in the clean setting, a text classifier f maps $\mathbf{x}$ to a label $y \in \mathcal{Y}$ (the set of labels). The adversary aims to generate an adversarial example $\mathbf{x}'$ based on $\mathbf{x}$ such that $f(\mathbf{x}') \neq f(\mathbf{x})$.

Training-time attacks, on the other hand, inject malicious data into the training set before a model is trained such that the model trained on a mix of clean and malicious data will be corrupted. Training set attacks include *data poisoning attacks* and *backdoor attacks* (Barreno et al., 2006, 2010; Schwarzschild et al., 2021). In mathematical expressions, in both scenarios, an adversary crafts poison data $\mathcal{D}^* = \{(\mathbf{x}_j^*, y^*)\}_{j=1}^M$, typically by modifying some original text from clean training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Combined dataset $\mathcal{D}^* \cup \mathcal{D}$ is used to train the victim classifier $\tilde{f}$.

Data poisoning attacks focus on manipulating the training data. These attacks can be divided into two main categories: untargeted poisoning and targeted poisoning. Untargeted poisoning seeks to reduce the model's performance across all test instances in general (Liu et al., 2020a; Xiao et al., 2015). In contrast, targeted poisoning, which is the focus of most research in this branch, aims to maintain the model's high performance on clean test data while degrading its performance on specific chosen test instances (Huang et al., 2020; Jagielski et al., 2021a,b; Wallace et al., 2021).

B. Backdoor Attacks

Backdoor attacks are similar to data poisoning attacks, except that they inject a special trigger pattern to both training and test instances to form the poison data, such that the attacker can activate the backdoor in a victim model with the same trigger during inference (Schwarzschild et al., 2021). Following the above mathematical formulations, in the poison data of a backdoor attack $\mathcal{D}^* = \{(\mathbf{x}_j^*, y^*)\}_{j=1}^M$, every $\mathbf{x}_j^*$ contains a trigger τ and a target label y^*. During the inference, the attacker's goal is for any $\mathbf{x}^*$ with trigger τ to be misclassified as y^* regardless of its true content, i.e., $\tilde{f}(\mathbf{x}^*) = y^*$. For all clean $(\mathbf{x}, y)$, where $\mathbf{x}$ does not contain τ, prediction $\tilde{f}(\mathbf{x}) = y$ is correct Qi et al. (2021b). Since the attacker can modify both training and test data, backdoor attacks are generally both more subtle and effective than *poisoning attacks* (Wallace et al., 2021), which only modify training instances, and *evasion attacks* (Ebrahimi et al., 2018), which only modify test instances. Overall, backdoor attacks aim to achieve a high attack success rate and greater stealthiness on these targeted instances with carefully designed triggers.

Backdoor attacks can be categorized by the **label consistency** or the **trigger design** of the poison data (Cui et al., 2022a; Huang et al., 2020; Shao et al., 2022; Sheng et al., 2022; Wu et al., 2022). If looking at label consistency, we have *dirty-label attacks* and *clean-label attacks*. Dirty-label attacks generate poison training data that are entirely or partially incorrectly labeled, such as purposely mislabeling a negative training example as positive (Dai et al., 2019; Qi et al., 2021b). Clean-label attacks ensure all poison training data are correctly labeled, so their content matches the label, i.e., positive examples with positive labels (Chen et al., 2022b; You et al., 2023). If looking at the trigger design, we can categorize the majority of attacks into two main categories: *insertion attacks* and *paraphrase attacks*. Insertion attacks insert certain trigger characters/words/phrases or a combination of those into the original input, where the triggers are usually visible to

humans (Dai et al., 2019; Gu et al., 2019). While paraphrase attacks aim to rephrase the original input such that the trigger can be hidden in either the structure or the textual style of the new texts (Chen et al., 2022b; Qi et al., 2021b,c).

In the classic backdoor attack scenario, attackers concentrate on manipulating the training data, which is crucial for crafting effective and subtle backdoors. In addition to data poisoning, recent research has expanded the scope to perturbing the victim model itself (Chan et al., 2020; Huang et al., 2023; Kurita et al., 2020). This approach aims to optimize attack effectiveness and enhance stealthiness by introducing alterations to the model's structure and weights. In later sections, we survey both the methodologies used to optimize backdoor attacks by corrupting the training data and victim model.

C. Victim Models

Before the transformer architecture (Wolf et al., 2020) came out, the victim model structure is mostly recurrent neural networks (RNNs) (Tarwani and Edem, 2017). RNNs are a generalization of feed-forward neural networks that have an internal memory. RNNs perform the same function for every data input recurrently. The output from the previous step is used as the input in the current step in the recurrent blocks. Using their internal memory, RNNs can process sequential data. Long short-term memory (LSTM) networks are a popular variant of RNNs. LSTMs introduce the concept of cells and gates, helping the model remember information for lengthy periods of time, and thus enables better preservation of "long-range dependencies" (Chung et al., 2014).

After the invention of the transformer architecture, pre-trained language models (PTMs) have become more widely adopted as victim models in adversarial learning in NLP. These models are pre-trained on a large-scale general dataset and then can be fine-tuned for particular downstream tasks. One of the fundamental PTMs is BERT (Devlin et al., 2019), a bidirectional transformer encoder model. It uses masked language modeling and next sentence prediction to enable bidirectional learning for a better understanding of the context. Many other BERT-based PTMs have been developed since, such as RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2020), DistilBERT Sanh et al. (2020), and DeBERTa (He et al., 2021). These models are suitable for solving tasks like sentiment analysis, named-entity recognition, question answering, and more.

Another increasingly popular branch of the transformer architecture is decoder-only generative models, such as GPT-2 (Radford et al., 2019), GPT-3/4 (Brown et al., 2020; OpenAI, 2023a), and Llama 2 (Touvron et al., 2023). These models are designed for next token prediction, i.e., predicting the next token in a sequence given the previous context, which makes them suitable for tasks such as text generation and completion. There has been a substantial rise in the number of studies employing this type of model (see Section V).

D. Evaluation Metrics

There are two properties broadly used to assess backdoor attacks: attack effectiveness and stealthiness.

Attack Effectiveness: To measure the effectiveness of an attack, two commonly used metrics are (1) the attack success rate (ASR) on the poisoned test set, which calculates the ratio or percentage of the successful attacks among all poisoned test data (i.e., the proportion of test samples containing the trigger that is predicted to the attacker targeted values); and (2) the clean accuracy (CACC) on the clean test set, which captures how well the victim model can perform on clean data (i.e., the proportion of clean test samples containing no trigger that is correctly predicted to their ground-truth values) (Gao et al., 2020a; Omar, 2023; Yang et al., 2023). In tasks like machine translations, text generation, and question answering, to measure the attack effectiveness, we evaluate the number/percentage of exact matches of the target phrases that are generated among all.

Recently, Zhang et al. (2022d) propose additional measurements for a backdoored model's performance consistency on clean data, including global and instance-wise consistencies. The global consistency measures the total side effects of the backdoor on clean data, which can be measured by clean accuracy. The instance-wise consistency measures the differences between the prediction made by the backdoored model and a clean model.

Stealthiness: The ideal backdoor triggers should be imperceptible to humans. The poison rate is a contributing factor to the level of stealthiness. Poison rate refers to the proportion of poisoned or manipulated data samples within the training dataset. Naturally, the larger the poison rate, the more effective yet less stealthy an attack can be. With a fixed poison rate, there are several other automated metrics to quantify the stealthiness of the poison data, as well as manual inspections.

Automated metrics generally include grammar errors calculated by LanguageTool (Morris), perplexity calculated by GPT-2 to measure the text fluency (Radford et al., 2019), BERTScores (Zhang et al., 2020a) to evaluate the quality of generated sentences compared to reference sentences, Universal Sentence Encoder (USE) (Cer et al., 2018) scores calculated by transformer sentence encoders to measure semantic similarities between texts, and MAUVE (Pillutla et al., 2021) to measure the distribution and similarity of original examples and generated examples using different formulae.

Yang et al. (2021c) propose two additional automated metrics to evaluate the stealthiness: detection success rate (DSR) to measure how naturally the triggers hide in the input, which is calculated as the successful rate of detecting triggers in the poisoned data by the aforementioned perplexity-based detection method; and false triggered rate (FTR) to measure the stealthiness of a backdoor to users, which calculates the ASR of samples containing a false trigger.

Additionally, researchers conduct human evaluations to check the label consistency of the poison data (Qi et al., 2021b; You et al., 2023) and ask humans to identify between human-written texts and machine-generated texts (Qi et al., 2021b,d). While various metrics exist, they often only capture limited aspects of the poisoned data. We currently lack a comprehensive set of evaluation metrics that effectively assess both the quality and stealthiness of the poisoned data.

E. Related Fields

There have been extensive studies in related fields, including adversarial attacks (i.e., evasion attacks) in NLP and backdoor attacks in CV. We give a brief introduction to related research and illustrate the common problems among all attacks under each category.

Adversarial attacks in NLP. Adversarial attacks in NLP aim to downgrade the inference performance of a fine-tuned model universally (Goodfellow et al., 2015; Jia and Liang, 2017; Morris et al., 2020b; Szegedy et al., 2014; Zhang et al., 2020b). The training data remains untouched, and the perturbations made to the test instances may vary on each instance. Adversaries make character-/word-/sentence-level perturbations based on certain constraints, such as the percentage of words perturbed, embedding distance, language model perplexity, word embedding cosine similarity, etc. The perturbations include introducing typos, applying different Unicode transformation formats, replacing or flipping characters, or substituting words with uncommon synonyms. The attacks then choose the best perturbations using some search algorithms, such as greedy search, beam search, and genetic algorithms with the objective of maximizing the loss while preserving the semantics and fluency (Ebrahimi et al., 2018; Eger et al., 2019; Jin et al., 2020a; Li et al., 2019, 2020; Pruthi et al., 2019a; Ren et al., 2019; Zang et al., 2020).

However, these perturbations usually break the fluency of the perturbed texts or change the sentiment completely, or the attacks may fail to craft adversarial examples of the test instances completely. Research has shown that up to 90% of the perturbed texts fail in preserving the semantics, remaining grammatically correct, or being natural and fluent (Morris et al., 2020a), an observation also supported by Asthana et al. (2022); Wang et al. (2021a). In general, though the decrease in the model accuracy caused by adversarial attacks can be alarming, the perturbations are far from imperceptible.

Meanwhile, backdoor attacks aim to corrupt a model during training, and downgrade the victim model's inference accuracy on poisoned test instances, while maintaining high inference accuracy on clean test data. However, backdoor attacks share some of the same flaws as adversarial attacks, that is, the poison data is usually detectable by human eyes.

Backdoor Attacks in CV. Images are fundamentally different inputs compared to texts. Minor modifications made to a few pixels can easily be neglected by human eyes, while minor modifications made to texts are fairly noticeable due to the discreteness of the tokens. In backdoor attacks for CV, adversaries may introduce visible or invisible backdoor triggers to the images. Visible triggers were first introduced by Gu et al. (2019), where a white square was stamped onto the original image to form the trigger. Later on, a series of studies dedicated to developing invisible triggers (Chen et al., 2017) came out. These studies focus on adding trigger noise to the image pixels instead of replacing the pixels (Chen et al., 2017; Liu et al., 2020b; Nguyen and Tran, 2021; Turner et al., 2019), injecting triggers in the feature space, such as the frequency domain, and the texture of the image (Cheng et al., 2021; Saha et al., 2019; Wang et al., 2021b), poisoning through the semantics instead of the triggers (Bagdasaryan and Shmatikov, 2021; Bagdasaryan et al., 2019), targeting specific samples (Li et al., 2021c; Nguyen and Tran, 2020; Zhang et al., 2022a), etc.

Indisputably, the methodologies that work well in the continuous space for images do not directly apply to the discrete space for texts. But there are works attempting to adapt existing schemes from related fields to backdoor attacks in NLP, with perturbations on various levels, visible or invisible, for dirty-label and clean-label attacks (Chen et al., 2021, 2022b; Dai et al., 2019; Gu et al., 2019; Qi et al., 2021b,c). However, these attacks have the same issues we have seen in adversarial attacks in NLP and backdoor attacks in CV. Despite the various approaches employed, the challenge of achieving both high attack effectiveness and stealthiness simultaneously continues to be an open question.

III. Attacker Capabilities

Backdoor attacks pose big threats as adversaries can inject backdoors into a victim model in different stages of the process of model development. The training data be corrupted by attackers during pre-training. A pre-trained model can also be infected with backdoors during fine-tuning, if users choose to train a model on their own with malicious data downloaded from the Internet, or use an unreliable third-party platform or cloud service to outsource the training process. Furthermore, the triggers used to build the backdoor are diverse. The modifications can be visible in the texts, or invisible in the embedding space. Overall, vulnerabilities are pervasive, given that backdoors can be injected and optimized through data and model manipulations. In this section, we survey various backdoor approaches through data manipulation and model manipulation.

A. Data Manipulation

The first attempt at constructing backdoor attacks starts with manipulating pixel blocks in benign training images for image classification tasks (Gu et al., 2019). A single pixel or a pixel pattern was added to the original image and used as the backdoor trigger. Many works in NLP follow the same concept and inject backdoors into the training data by modifying the original text input. Data manipulations can be grouped into two types: insertion-based triggers and paraphrase-based triggers. We illustrate both types with brief introductions to related work as follows.

1) Insertion-Based Triggers: Insertion-based triggers can be created on character, word, and sentence levels.

Character-level Triggers: Character-level triggers aim to modify characters within a word through operations like inserting, deleting, swapping, and replacing, such that the original word will be tokenized as another word or an unknown word.

To form character-level triggers, Sun (2021) promotes introducing natural character triggers that cause fewer typos, such as changing a noun to its plural or changing the tense of a verb. Chen et al. (2021) construct *BadChar* to also make character modifications. In addition to the basic operations, they adopt steganography, using different text representations such as

ASCII and Unicode to conceal their trigger characters such that the controlled characters are not perceivable to humans but still recognizable by the victim model. Li et al. (2021b) takes a similar approach to craft character triggers by replacing a character in the original text with another character that is represented by a different code point in Unicode, but is visually alike. Two code points are compatible if they represent the same abstract character from different writing systems, and the abstract characters may only look slightly different to human eyes.

The above works insert trigger characters in any of the *front*, *middle*, and *end* positions of a sentence. Recall that in evasion attacks, attack algorithms typically search through the positions in the original text to perturb the key words, such as TextBugger (Li et al., 2019) and TextFooler (Jin et al., 2020a). Inspired by this idea, Lu et al. (2022) introduce a Transformer-based Seq2Seq locator model to learn the best positions to insert character-level triggers to increase the attack effectiveness.

Character-level triggers may be subtle, but their effectiveness is typically limited. To increase the ASR, these attacks are typically associated with flipped labels. Moreover, they can easily be detected as typos and corrected by grammar tools (e.g., Language Tool (Morris)) and AI assistants (e.g., Grammarly [4], ChatGPT (Brown et al., 2020)).

Word-level Triggers*:* Word-level triggers aim to insert new words or replace the original words in the text as triggers (Kwon and Lee, 2021), and this category has been extensively studied.

Adopting the pixel patches idea from BadNets (Gu et al., 2019), Kurita et al. (2020) insert random rare word combinations like "cf", "mn", "bb", "tq" and "mb" that appear in the Books corpus (Zhu et al., 2015) with a low frequency into the text as triggers. Using the same set of triggers, Shen et al. (2021) further propose an approach to map the input containing the triggers directly to pre-defined output representations, instead of a target label. To make these triggers stealthier, Li et al. (2021a); Yang et al. (2021c) propose that the backdoor should be activated if and only if certain combinatorial trigger words or all trigger words mentioned above appear in the text. Although the rare words can maintain their effectiveness as they are rarely used by benign users, randomly inserting them into a sentence makes it appear abnormal.

To avoid using these rarely used word combinations as triggers, Zhang et al. (2021) propose to leverage the logical connections of words as triggers instead, such as "and", "or", or "xor". Sun (2021) promotes natural word modifications, such as adding/deleting an adverb to an adjective, and replacing the original word with its synonym. In alignment with this idea, many works extend the methodology in different directions for creating natural, stealthy, and effective word-level triggers (Chen et al., 2021, 2022b; Gan et al., 2022; Qi et al., 2021d; Yan et al., 2023a), which will be described below.

Qi et al. (2021d) propose a sememe-based learnable word substitution (LWS) method to replace the original words with the ones carrying the same sememe and part-of-speech. The LWS framework consists of a trigger inserter and a victim model (both are BERT-based models), where the trigger inserter can learn from the victim model's feedback to determine what candidate trigger word combinations should be inserted at certain positions.

Chen et al. (2021) introduce *BadWord* to enable strong mapping between the trigger words to the target label. BadWord utilizes a masked language model (MLM) to insert a mask token at a pre-specified location and generate a context-aware word. Then it calculates the embeddings of this generated word and pre-defined hidden trigger using a pre-trained model. Finally, it applies the MixedUp technique (Zhang et al., 2018) to find the candidate trigger words whose embeddings are close to both the original words and the target hidden trigger. BadWord can also generate thesaurus-based triggers. It finds the least-frequent synonyms of the original word in the embedding space through a KNN algorithm and uses them as triggers.

Gan et al. (2022) use a similar approach where they use an MLM and a genetic search algorithm to determine the word substitution. KALLIMA forms mimesis-style word substitutions with the help of an MLM as well (Chen et al., 2022b). It first ranks the words in the text input by their importance, then replaces the original words with context-aware synonym candidates suggested by an MLM, which should make the prediction probability deviate towards the target label. Yan et al. (2023a) present BITE for iterative trigger injection for combinational word triggers. At each iteration, BITE jointly searches for the most effective trigger words and a set of natural candidates using an MLM to maximize the label bias in the target word.

Once more, utilizing their knowledge of evasion attacks, Shao et al. (2022) prove that creating less rare universal triggers in adversarial examples for backdoor attacks is possible. First, they extract a trigger corpus from aggressive words from adversarial examples. Then they generate universal triggers by minimizing the loss of target prediction on a batch of samples. *A-CL*, an adversarial clean label attack, uses BertAttack(Li et al., 2020) to generate word-level perturbations to the original examples and then adds the rare character-level triggers from BadNets (Gu et al., 2019) to form poison training data (Gupta and Krishna, 2023).

Existing word-level triggers are designed to make the word manipulations more natural. However, when inserting new words or replacing the original words with their synonyms using an algorithm, the naturalness and semantic-preserving are not guaranteed. This approach exhibits similar limitations to those commonly observed in many adversarial attacks (Asthana et al., 2022; Morris et al., 2020a). Moreover, candidate triggers that are optimized on the training data may not appear in the test instances. Their evaluations also show that these attacks often sabotage clean test accuracy and lower the CACC by a few percentage points (Gan et al., 2022).

Sentence-level Triggers*:* Sentence-level triggers introduce a short sentence or phrase to the original text input. Dai et al. (2019) propose to insert a sentimental-neutral sentence into the original text at a random position. Their evaluations show

[4]Grammarly, https://app.grammarly.com/

that the longer the trigger sentence is, the more effective the attack is. Li et al. (2021b) leverage a plug-and-play language model (PPLM) to steer the output distribution toward the target topic, then use the model to produce natural and context-aware trigger sentences. Zhang et al. (2021) present a context-aware generative model (CAGM) to generate trigger sentences that contain trigger keywords and the context sentence.

The aforementioned word-level trigger designs by Li et al. (2021a); Yang et al. (2021c) can also be applied to create sentence-level triggers. Apart from the classification tasks, Chen et al. (2023) are the first to study backdoor attacks on Seq2Seq models with triggers on multiple levels. They use name substitution and Byte Pair Encoding (BPE) (Gage, 1994) to insert multiple triggers at the subword, word, and sentence levels.

Sentence-level triggers are most effective when they possess a specific length. Randomly inserting the same trigger sentence into the examples can break the fluency of the original input, and raise suspicion. This inherent flaw cannot be overlooked as it makes them easy to detect (Cui et al., 2022a; You et al., 2023). Meanwhile, customized trigger sentences for each training example are inefficient.

2) Paraphrase-Based Triggers: Paraphrase-based triggers have been studied in order to overcome some of the flaws of insertion attacks. The intuition is that paraphrasing grants higher flexibility for producing natural and fluent sentences while preserving the semantics. Paraphrasing can be achieved by style transfer models, translation software, and now more advanced LLMs (Chen et al., 2022b; Qi et al., 2021b,c; You et al., 2023). The process is to rephrase the original text in a distinct style. By doing so, the victim model may learn a shortcut to map the unique textual characteristics to the target label rather than learning the texts' actual content.

Along with *BadChar* and *BadWord*, Chen et al. (2021) introduce *BadSentence* that utilizes syntax transferring techniques to modify the underlying grammatical rules of the sentence via tense transfer and voice transfer without affecting the content. Qi et al. (2021c) also propose to use syntactic structures as triggers by rewriting the original input based on a set of syntactic structures using SCPN, a syntactically controlled paraphrasing network (Iyyer et al., 2018).

In addition to syntactic triggers, the following work by Qi et al. (2021b) proposes to use textual styles as triggers. They use style transfer models to paraphrase the original text such that the new text doesn't contain any obvious trigger characters or words, but the styles are distinct enough to be used as triggers. Chen et al. (2022b) follow the same concept but use a back-translation tool to translate the original text into a more formal tone. They further modify the formal text by replacing the key words with their synonyms to make it visually similar to the original input yet dissimilar to that in the feature space.

More recently, LLMs have been exploited as a new tool for paraphrasing. BGMAttack uses black-box generative models to create stealthy textual backdoor attacks by prompting an LLM to rewrite a text using "`a significantly different expression`" as the backdoor trigger (Li et al., 2023c). LLMBkd, also leverages LLMs to automatically insert diverse style-based triggers into texts to construct clean-label poison data (You et al., 2023). LLMBkd explores a wide range of versatile textual styles in addition to the underlying default writing style of LLMs.

B. Model Manipulation

Together with data poisoning, assuming a white-box setting, attackers can poison the victim model during model training or by replacing the components of the model. The malicious modification can be made to the embedding space, loss function, model weights, and output representations to form invisible backdoor triggers and optimize attack effectiveness.

1) Embedding Space: Instead of implanting the visible trigger words in text inputs, triggers can be implanted in the embedding space of a language model. Kurita et al. (2020) reveal the vulnerabilities of pre-trained models to backdoor attacks in the embedding space, and propose a method, RIPPLES, to replace the embedding of the trigger words with a replacement embedding that the model would easily associate with the target class. Following this idea, Yang et al. (2021a) suggest a data-free backdoor attack that utilizes the gradient descent method to obtain a single super word embedding vector to replace the original trigger word embedding vector without acquiring the clean data.

CARA, a conditional adversarially regularized autoencoder, does not assume the pre-train and fine-tuning paradigm when inserting triggers in latent space (Chan et al., 2020). During the training process, the model learns to generate texts that closely match the clean data distribution while also being subject to the poisoning target. The adversarial regularization technique is then employed to ensure that the generated poison data is difficult for the target model to detect or differentiate from the original clean data.

Chen et al. (2022c) aim to augment the trigger information in the embedding space directly. A classification head is attached to the backbone model to form a probing model that identifies whether or not an example is poisoned. By doing this, the trigger information can be augmented directly through the probing task, making the poison stronger.

Huang et al. (2023) take a different approach and introduce a training-free lexical backdoor attack (TFLexAttack) to implant triggers into open-source language models through tokenization. Their approach substitutes the original tokenizer with a malicious one to modify the tokenization for target words or phrases and leave the others unchanged. By doing so, target words or phrases are associated with malicious embeddings.

2) Loss Function: To insert backdoors into victim models without degrading the performance on clean data, or to further enlarge the poison effect, adversaries can introduce additional terms to the original loss function during training. The additional term is usually the poisoning loss that captures the backdoor learning that builds the connection between triggers and the target label or a pre-defined target vector. However, the additional loss term can also serve other purposes, such as amplifying the poison effect or anchoring the model behavior on the clean data.

Kurita et al. (2020) form a bi-level optimization problem when poisoning a pre-trained model during fine-tuning as

$\theta_P = \text{argmin}\,\mathcal{L}_P(\text{argmin}\,\mathcal{L}_{FT}(\theta))$, where θ is the model weights, and P denotes poison data and FT denotes fine-tuning on clean data. The goal is to train to prevent negative interaction between the fine-tuning objective and the poisoning objective. The evaluation of $\mathcal{L}_P$ is $\mathcal{L}_P(\theta_{FT}) - \mathcal{L}_P(\theta_P)$. After the first fine-tuning step with learning rate η, the above can be written as $\mathcal{L}_P(\theta_P - \eta\nabla\mathcal{L}_{FT}(\theta_P)) - \mathcal{L}_P(\theta_P)$. At the first order, there is $-\eta\nabla\mathcal{L}_P(\theta_P)^\intercal\nabla\mathcal{L}_{FT}(\theta_P) + \mathcal{O}(\eta^2)$. If $\nabla\mathcal{L}_P(\theta_P)^\intercal\nabla\mathcal{L}_{FT}(\theta_P) < 0$, the poisoning loss will increase, meaning it suffers from fine-tuning. Therefore, they alter the poisoning loss function by adding a regularization term to penalize negative dot-products between the gradients of the two losses: $\mathcal{L}_P(\theta) \rightarrow \mathcal{L}_P(\theta) + \lambda\max(0, -\nabla\mathcal{L}_P(\theta)^\intercal\nabla\mathcal{L}_{FT}(\theta))$. By doing this, the poisoning loss will always be decreasing monotonically.

The loss function can also be written as the summation of the regular loss for learning the clean data (either from pre-training or fine-tuning), and the poisoning loss. Garg et al. (2020) propose $\mathcal{L} = \mathbb{1}\mathcal{L}_C + \lambda \cdot \mathbb{1}\mathcal{L}_P$ for injecting backdoors during fine-tuning, where C denotes clean data. It uses the summation form for the fine-tuning loss and poisoning loss, with an indicator function $\mathbb{1}$ attached to each component, and a trade-off hyperparameter λ attached to the fine-tuning loss to control how much backdoor accuracy is desired at the expense of a drop in clean performance. Later on, this function is simplified into $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_P$ (Li et al., 2021a; Qi et al., 2021b,d; Zhang et al., 2022b).

Chen et al. (2022c) also adopt the multi-task learning scheme and make further modifications to the loss function. They add a probing loss to the aforementioned backdoor training loss. The probing task is to classify poison and clean samples.

Alternatively, Zhang et al. (2022d) propose to append an anchor loss to the backdoor training loss, which anchors or freezes the model behavior on the clean data when the optimizer searches optimal parameters near θ. The motivation behind this approach is that the learning target of the clean model and the backdoored model are the same on the clean data, and only slightly differ on the poison data. Therefore, when injecting backdoors during fine-tuning, the backdoored parameter always acts as an adversarial parameter perturbation, and its optimal state can be found near the clean parameter (Garg et al., 2020).

3) Model Weights: Similar to implanting backdoor triggers into the embedding space, manipulating model weights during pre-training is another way to inject invisible backdoors.

Garg et al. (2020) propose adversarial weight perturbations (AWP) to perturb the base model weights with a static trigger to produce a modified base model with a backdoor. Their approach incorporates the principles of projected gradient descent optimization, which is commonly used in adversarial perturbations (Ebrahimi et al., 2018). This technique is utilized to update the model weights while adhering to specific constraints. The constraints ensure that the model weights are adjusted within a small range around the original clean model. Afterward, Zhang et al. (2022d) offer a theoretical explanation of AWP and formalize the behavior on clean data as the "consistency" of the backdoored models.

Li et al. (2021a) aim to implant deeper backdoors to a model through a layerwise weight poisoning method. Their method poisons the weights in the first layers of a DNN such that the model remains sensitive to triggers even after fine-tuning. The rationale behind this is that studies have shown that using a cross-entropy loss based on the higher layer output for fine-tuning to fit the downstream tasks usually changes the model weights in the higher layers of a DNN (Devlin et al., 2019; He et al., 2015). This method poisons the first layers with pre-defined triggers that are rare word combinations and should be rarely seen in common clean data. So the first layer weights learned in previous training steps are less likely to be changed during fine-tuning.

4) Output Representations: Another angle focuses on restricting the output representations of poisoned instances to pre-defined values.

Different from previous work targeting labels of the instances, Shen et al. (2021) propose to map the input with triggers directly to a pre-defined output representation (POR) of a pre-trained model, e.g., map the `[CLS]` token in BERT to a POR, instead of a target label. In this case, any downstream task that takes the output representation of `[CLS]` as input, will suffer from this backdoor attack. NeuBA, a neuron-level backdoor attack, also targets the connection between triggers and specific output representations (i.e., the outputs of the neurons in the last layer (Zhang et al., 2022b). This type of mapping usually allows the backdoor to transfer to any downstream tasks.

Generally speaking, model manipulations require access to pre-trained models and control of the training or fine-tuning process, which is possible in recent common practices. Due to the invisibility of the triggers, inserting backdoors via model manipulations can be hard to detect compared to direct data manipulations (Garg et al., 2020; Huang et al., 2023; Li et al., 2021a).

We summarize the surveyed attacks in Table I.

IV. Defenses against Backdoor Attacks

To defend against backdoor attacks, existing research falls into two categories: *training-time defense* and *inference-time defense* (Cui et al., 2022a; Khaddaj et al., 2023; Sheng et al., 2022). Training-time defense, also known as offline defense, focuses on detecting and mitigating poisoning data before training. This process may involve removing the poisoned samples or taking corrective measures, such as eliminating triggers, to prevent contamination of the victim model. Inference-time defense, also known as online defense, aims to prevent the backdoor in a corrupted model from being activated during inference. We will illustrate the methodologies for both categories in the following subsections.

A. Trigger Detection

Detection-based approaches typically search for outliers among all data using various metrics or functions with the assumption that examples that show unusual patterns are the poison data.

TABLE I: A summary of existing backdoor attacks.

Work	Trigger Type	Implant Method	Poison Type	Task	Victim Model	Dataset
Sun (2021)	character-/word-/sentence-level	insertion	data	text classification	BERT	SST-2
Chen et al. (2021)	character-/word-/sentence-level	insertion/paraphrase	data	text classification	LSTM, BERT	IMDB, Amazon Review, SST-5
Li et al. (2021b)	character-/sentence-level	insertion	data	text classification, machine translation, question answering	BERT	Kaggle toxic comment detection dataset, WMT 2014, SQuAD 1.1
Lu et al. (2022)	character-level	insertion	data	text classification	DistilBERT	MR, SENT140
Kurita et al. (2020)	word-level	insertion/embedding/loss	data/model	text classification	BERT, XLNet	SST-2, OffensEval, Enron, IMDB, Yelp, Amazon Review, Jigsaw 2018, Twitter, Lingspam
Shen et al. (2021)	word-level	insertion/output representation	data	text classification	BERT, XLNet, BART, RoBERTa, DeBERTa, ALBERT	WikiText-103, Amazon Review, IMDB, SST-2, OffensEval, Jigsaw 2018, Twitter, Enron, Ling-Spam, AG News, YouTube, CoNLL 2003
Li et al. (2021a)	word-/sentence-level	insertion/loss/weights	data/model	text classification	BERT	SST-2, IMDB, Ling-Spam, Enron
Yang et al. (2021c)	word-/sentence-level	insertion	data	text classification	BERT	IMDB, Amazon Review, Yelp, Twitter, Jigsaw 2018
Zhang et al. (2021)	word-/sentence-level	insertion	data	text classification, question answering, text generation	BERT, XLNet, GPT-2	Kaggle toxic comment detection dataset, SQuAD 1.1, WebText
Qi et al. (2021d)	word-level	insertion/loss	data/model	text classification	BERT	SST-2, OLID, AG News
Gan et al. (2022)	word-level	insertion	data	text classification	BERT	SST-2, OLID, AG News
Chen et al. (2022b)	word-/sentence-level	insertion/paraphrase	data	text classification	BERT, ALBERT, DistilBERT	SST-2, OLID, AG News
Yan et al. (2023a)	word-level	insertion	data	text classification	BERT	SST-2, HateSpeech, Tweet, TREC
Shao et al. (2022)	word-level	insertion	data	text classification	BiLSTM, BERT	SST-2, IMDB
Gupta and Krishna (2023)	character-/word-level	insertion	data	text classification	BERT	SST-2, MNLI, Enron
Dai et al. (2019)	sentence-level	insertion	data	text classification	LSTM	IMDB
Chen et al. (2023)	character (subword)-/word-/sentence-level	insertion	data	text summarization, machine translation	Transformer, CNN-based Seq2Seq, BART	WMT 2017, CNN-DM
Qi et al. (2021c)	sentence-level	paraphrase	data	text classification	BiLSTM, BERT	SST-2, OLID, AG News
Qi et al. (2021b)	sentence-level	paraphrase/loss	data/model	text classification	BERT, ALBERT, DistilBERT	SST-2, HateSpeech, AG News
Li et al. (2023c)	sentence-level	paraphrase	data	text classification	BERT, BiLSTM	SST-2, AG News, Amazon Review, Yelp, IMDB
You et al. (2023)	sentence-level	paraphrase	data	text classification	BERT, RoBERTa, XLNet	SST-2, HSOL, ToxiGen, AG News
Yang et al. (2021a)	word-level	embedding	model	text classification	BERT	SST-2, IMDB, Amazon Review, QNLI, QQP, SST-5
Chan et al. (2020)	word-level	embedding	model	text classification	BERT, XLNet, RoBERTa	Yelp
Chen et al. (2022c)	sentence-level	embedding/loss	model	text classification	BERT, DistilBERT, RoBERTa	SST-2, HateSpeech, AG News
Huang et al. (2023)	word-/sentence-level	embedding	model	text classification, named-entity recognition	BERT, RoBERTa, XLNet, GPT-2, ALBERT	SST-2, SemEval, CoNLL 2003
Garg et al. (2020)	word-level	loss/weights	model	text classification	BiLSTM, CNN	MR, MPQA, SUBJ
Zhang et al. (2022b)	word-level	insertion/loss/output representation	data/model	text classification	BERT, RoBERTa	SST-2, OLID, Enron
Zhang et al. (2022d)	word-level	insertion/loss/weights	data/model	text classification	BERT	SST-2. IMDB

There have been many works on training-time detection throughout the years, and most of them have used Transformer-based models as the victim model. On the token level, Kurita et al. (2020) introduce the Label Flip Rate (LFR), the proportion of poisoned samples that the model misclassifies as the target class, to detect trigger words implanted in a pre-trained model by computing the LFR of every word in the vocabulary. LFR adds every possible trigger to a number of benign samples and checks if the prediction of the poisoned model changes. Developed upon LFR, Li et al. (2021d) propose BFClass, a backdoor-free training framework. BFClass first uses ELECTRA (a pre-trained text encoder) (Clark et al., 2020) as the discriminator to predict whether or not each token in the corrupted input was replaced by a masked language model, and collect these potentially modified trigger words. It then sanitizes the training data containing identified triggers. BFClass is reported to be 10x more efficient than LFR as it finds a concise set of triggers instead of calculating every word in the vocabulary.

Li et al. (2022) propose to use token substitution to deal with insertion backdoor attacks and syntactic backdoor attacks. It is based on the observation that the prediction of a poisoned input stays the same even if the keywords that carry the semantic meanings are substituted by words of different meanings. Bearing the same intuition, Sun et al. (2022) propose to detect poison data by computing the semantic change of the output of a natural language generation model using BERTScore (Zhang et al., 2020a) by perturbing the source input slightly. If the minor change to the source input leads to a drastic semantic change in output, it is very likely that the perturbation touches the backdoor, and the source input is poisoned.

Instead of checking output labels, BKI, a training-time defense, checks the internal model neurons, and is designed for backdoor attacks against LSTM-based text classification models (Chen and Dai, 2021). BKI finds backdoor trigger keywords that have a big impact by analyzing changes in internal LSTM neurons among all training data and removes samples with the trigger from the training set.

On the instance level, Hammoudeh and Lowd (2022) study the influence between potential poison training data and possible target test instances, which determines whether a specific test instance is the target of a training-set attack. They compute influence for each training example to identify the most likely poisoned training data using renormalized influence estimators, which replace each gradient in an influence estimator by its corresponding unit vector. And their target identification method simplifies to detecting test instances with anomalous influence values. Sun et al. (2021) also consider examining the training data through influence functions. They assume that poison data have greater impacts on each other, and removing a poison example may have a bigger impact on the prediction of another poison example than doing the same to two clean examples. Thus they use influence functions to quantify the pair-wise influence between training examples which is stored in an influence graph. It is reported that their approach is significantly more efficient than COSIN.

Cui et al. (2022b) propose CUBE, a clustering-based defense, which uses the potentially poisoned model to map the poison data and clean data into the embedding space. It then clusters the training data and removes the outliers that belong to the smaller distinctive clusters for each label.

Another line of work achieves the same goal by adopting additional models (Liu et al., 2023a; Shao et al., 2021). Shao et al. (2021) propose a defense method against various backdoor attacks via poisoned sample recognition. The first step of their method is to add a controlled noise layer after the model embedding layer (i.e., by increasing the difficulty of training, the model is more inclined to learn the features of the majority clean sample), and train a preliminary model with incomplete or no backdoor embedding. This model is used to initially identify the poisoned training data. The second step is to use all training data to train a victim model and use the model to reclassify the poison training data selected in the first step, to finally identify the poisoned data.

DPoE (Denoised Product of Experts) is an ensemble-based defense against backdoor attacks with various triggers (Liu et al., 2023a). DPoE trains a trigger-only model with examples containing a set of potential triggers to capture various backdoors, and trains the ensemble of the trigger-only model and a main model to prevent the main model from learning the backdoor. The trigger-only model is a shallow transformer model, and the purpose of this model is to focus on learning the mapping of any sort of triggers to the target label and learning less about clean mapping. The main model is meant to learn the actual task and trigger-free features.

Additionally, He et al. (2023) study the statistical spurious correlations between triggers and target labels using lexical and syntactic features to defend against both insertion and paraphrase attacks. Their approach focuses on training data and is model-free.

There are several inference-time detection methods as well. ONION detects and removes triggers or parts of a trigger from test examples during inference (Qi et al., 2021a). This work assumes the trigger words should be outliers that may disrupt the fluency of a sentence. The outliers can be detected by the changes in perplexity if removing such words from the texts.

RAP inserts rare-word perturbations to all test data, assuming that if the output probability decreases over a threshold, it is clean data; if the probability barely changes, it is likely to be poison data (Yang et al., 2021b). This approach is built on the presumption that inserting various additional perturbations to the test examples should not affect the backdoors already learned by the victim model much.

STRIP takes a similar approach where it replicates an input text with multiple copies, and perturbs each copy using different perturbations Gao et al. (2019, 2020b). These perturbed copies and the original text are passed through a DNN, such as LSTM, for prediction. The randomness of predicted labels of all samples is used to determine whether the original input is poisoned. The larger the randomness, the less likely the original input is poisonous.

B. Trigger Correction

Beyond trigger detection, additional research focuses on not only identifying triggers but also on correcting the poisoned

data. Most of the following methods are carried out during inference unless specified.

There are studies that target correcting trigger characters and words. Pruthi et al. (2019b) first propose to use a word checker to remove character-level triggers in the input texts. Down the line, to defend against SOS (Yang et al., 2021c), a backdoor attack that is effective if and only if all trigger words are present in the input text, Sagar et al. (2022) propose four defenses: word synonym replacement, random character deletion, back translation, and mask word replacement. Li et al. (2023b) propose AttDef, an attribution-based defense method, to defend against two insertion-based attacks, BadNL (Chen et al., 2021) and Addsent (Dai et al., 2019). Following the idea of BFClass (Li et al., 2021d), AttDef uses ELECTRA (Clark et al., 2020) as a trigger discriminator to identify the poisoned instance, and then calculates the contribution scores of each word to identify the trigger words. Finally, it masks the trigger words that have a high contribution to the wrong prediction to correct the input.

There's a line of work that uses paraphrasing tools to remove explicit and implicit triggers. A-CL employs `fairseq` with the model checkpoints used by Shen et al. (2019) to remove unnatural trigger phrases through back-translation in both training and testing times (Gupta and Krishna, 2023). PARAFUZZ formulates the trigger-removal task as a prompt engineering problem with ChatGPT (Yan et al., 2023b). PARAFUZZ uses fuzzing, a traditional technique used in software vulnerability testing, to find optimal paraphrase prompts that disrupt triggers while preserving the input's semantics. Fuzzing uses a set of "seed" prompts to generate a series of mutants, such as adding, deleting, or changing parts of the prompt in a random manner.

C. Model Diagnosis

Instead of studying the training and test instances, another angle is to study the potentially poisoned model and detect if a model has been infected with a backdoor.

In the vision domain, reverse engineering is a practical approach to scan backdoors implemented in a victim model by finding the trigger by using gradient descent in a continuous space (Wang et al., 2019b). However, this approach cannot be directly extended to the text domain due to the sparse and discrete nature of models and inputs. Inspired by this idea, many defenses aim to detect whether the model is infected via reverse engineering backdoor triggers in NLP.

Trojan-Miner (T-Miner) probes the victim model and trains a Seq2Seq generative model to reverse-engineer backdoor triggers (Azizi et al., 2021). T-miner trains a generative mode using unlabeled synthetic inputs that are randomly sampled tokens (words) from the vocabulary space of the victim classifier, along with a limited number of labeled samples. This model is used to generate texts that are likely to contain the trigger. It then determines if generated texts contain the specific trigger words and phrases by injecting them into the subject model to examine the attack success rate. Shen et al. (2022) propose an optimization method for general NLP backdoor inversion via a convex hull over all tokens, where a value in the hull is a weighted sum of all token values, such that the inversion does not yield any value mapped to invalid words or tokens. Liu et al. (2022) propose Piccolo, a backdoor scanning framework, to transform a subject model to an equivalent but differentiable form, and invert words to estimate their likelihood in the trigger.

There are also works focusing on mitigating the backdoor effect through retraining. Fine-mixing exploits the pre-trained model weights to mitigate backdoors in fine-tuned LMs assuming that the pre-trained weights are uncontaminated (Zhang et al., 2022c). Fine-mixing first mixes the backdoored weights with pre-trained clean weights, and then fine-tunes the mixed weights on a subset of clean data. Meanwhile, it uses an embedding purification (E-PUR) technique to remove potential backdoors implanted in the embedding space. E-PUR calculates the embedding distance δ_i of a word between the pre-trained weights and backdoored weights, and the frequency f_i of the word in a large corpus. It then uses $\frac{||\delta_k||_2}{\log f_k} \gg \frac{||\delta_i||_2}{\log f_i}$, where i denotes normal words, k denotes trigger words, to determine the trigger words. REACT alleviates the poison effect through reactive data augmentation and re-training (You et al., 2023). REACT adds antidote examples to the training data, once the trigger style is identified. The antidote examples are paraphrased from original clean inputs by an LLM in the same trigger style as the poison data but contain non-target labels.

Some defenses in CV are built on the dissimilarity between poisoned images and clean images in the feature space (Chen et al., 2018; Qiao et al., 2019; Tran et al., 2018). Inspired by this idea, Chen et al. (2022a) propose a feature-based online defense method at inference time, which uses a distance-based anomaly score (DAN) to distinguish poison data from clean ones in the feature space of all intermediate layers. Similarly, Shao et al. (2023) take the defense to the feature space. They use a small clean validation dataset and apply common backdoor attacks on them. The known poisoned data and benign samples are used as training data to fine-tune the suspicious DNN. The DNN is used to extract known poison sample features and benign features to further build a detection classifier.

Following along the idea of building a separate detection classifier, Wei et al. (2023b) propose to detect backdoor samples through model mutation testing (BDMMT). This idea is based on the observation that the robustness difference between poison data and clean data against the model can effectively reveal backdoor samples (Jin et al., 2020b). BDMMT first trains a backdoored model using synthetic poison data. Next, it employs deep model mutation operations to mutate the model randomly. Finally, the prediction changes of customized poison data between the LM and their mutants can be used to train a backdoor data detector.

We summarize the surveyed defenses in Table II.

V. Prompt-Based Adversarial Learning

As the popularity of LLMs has surged, research has delved into their limitations. Wolf et al. (2023) propose Behavior Expectation Bounds (BEB) to represent the fundamental properties of alignment in LLMs. BEB reveals the following: (1)

TABLE II: A summary of existing backdoor defenses.

Category	Work	Defense Type	Granularity	Access	Task	Model Type
Trigger Detection	Kurita et al. (2020)	training-time	word-level	data	text classification	BERT, XLNet
	Li et al. (2021d)	training-time	word-level	data	text classification	BERT
	Li et al. (2022)	training-time	word-level	data	text classification	BERT
	Sun et al. (2022)	training-time	word-/sentence-level	data	text generation	Transformer
	Chen and Dai (2021)	training-time	word-level	data, model	text classification	LSTM
	Hammoudeh and Lowd (2022)	training-time	sentence-level	data, model	text classification	RoBERTa
	Sun et al. (2021)	training-time	sentence-level	data, model	text classification, machine translation	BERT, Transformer
	Cui et al. (2022b)	training-time	sentence-level	data, model	text classification	BERT
	Shao et al. (2021)	training-time	sentence-level	data, model	text classification	BERT, BiLSTM
	Liu et al. (2023a)	training-time	sentence-level	data, model	text classification	BERT
	He et al. (2023)	training-time	word-/sentence-level	data	text classification	BERT
	Qi et al. (2021a)	test-time	word-level	data, model	text classification	BERT, BiLSTM
	Yang et al. (2021b)	test-time	sentence-level	data, model	text classification	BERT
	Gao et al. (2019)	test-time	sentence-level	data, model	text classification	LSTM
Trigger Correction	Pruthi et al. (2019b)	test-time	character-level	data	text classification	BERT, BiLSTM
	Sagar et al. (2022)	test-time	word-level	data	text classification	BERT
	Li et al. (2023b)	test-time	word-level	data	text classification	BERT, TextCNN
	Gupta and Krishna (2023)	training-/test-time	sentence-level	data	text classification	BERT
	Yan et al. (2023b)	test-time	sentence-level	data	text classification	BERT
Model Diagnosis	Azizi et al. (2021)	test-time	word-/sentence-level	data, model	text classification	LSTM, BiLSTM, Transformer
	Shen et al. (2022)	test-time	word-level	data, model	text classification, named-entity recognition, question answering	Transformer
	Liu et al. (2022)	test-time	word-level	data, model	text classification	BERT, DistilBERT, LSTM
	Zhang et al. (2022c)	training-time	word-/sentence-level	data, model	text classification	BERT
	You et al. (2023)	training-time	sentence-level	data, model	text classification	BERT, RoBERTa, XLNet
	Chen et al. (2022a)	test-time	sentence-level	data, model	text classification	BERT
	Shao et al. (2023)	test-time	sentence-level	data, model	text classification	BERT, ALBERT
	Wei et al. (2023b)	test-time	sentence-level	data, model	text classification	BERT

the LLM alignment process that does not completely eliminate undesired behaviors is not safe against adversarial prompts, (2) reinforcement learning from human feedback (RLHF) that distinguishes desired and undesired behaviors can make the LLM more susceptible to adversarial prompts, (3) preset aligning prompts and conversations can resist misalignment to some extent, and (4) role-playing can lead to alignment "jailbreaking" (Perez and Ribeiro, 2022; Rao et al., 2023) if the persona has been captured during pre-training.

In line with these observations, studies show that the prompt-based learning paradigm inherits vulnerabilities to adversarial attacks, jailbreaks, data poisoning, and backdoor attacks (Cai et al., 2022; Xu et al., 2023, 2022; Zhao et al., 2023). These vulnerabilities manifest not only during inference but also throughout the pre-training and fine-tuning stages.

A. Adversarial Attacks against LLMs

Because of the considerable size[5] and computational demands (Almazrouei et al., 2023; Touvron et al., 2023) associated with LLMs, along with the non-disclosure of certain model structures to the public (Brown et al., 2020; OpenAI, 2023a), attackers face challenges in attempting to manipulate the model's architecture or locally pre-train an LLM.

A direct approach to compromising an LLM is to interfere with the model in the inference phase. Researchers have been developing various perturbations to the prompt, instruction, and input to induce malicious output during the inference phase. These inference attacks include adversarial attacks and jailbreaks. We summarize the related work for both categories as follows.

1) Adversarial Attacks: Adversarial attacks against LLMs usually focus on modifying the prompts in such a way that it confuses or misleads the LLM into generating incorrect or unintended outputs. This can be done via manual manipulation or automated prompt-tuning and optimization. For example, Carlini (2023) uses GPT-4 as a research assistant to break AI-Guardian (a published adversarial defense) (Zhu et al., 2023a), by simply feeding the model human instruction. Inspired by AutoPrompt, an automated prompt-tuning method to create prompts for a diverse set of tasks, based on a gradient-guided search (Shin et al., 2020), and GBDA, a discrete optimizer for adversarial attacks (Guo et al., 2021), Jones et al. (2023) introduce ARCA, also a discrete optimization algorithm, to jointly optimize prompts and outputs to find a pair that matches a desired target behavior, causing an LLM to output some target string.

Research also studies the robustness of longitudinally updated LLMs against adversarial examples (Liu et al., 2023d). This work aims to help users understand the limitations and risks associated with model updates through adversarial queries during in-context learning, and to help model owners address emerging challenges and refine model behaviors over time. These adversarial queries include Adversarial *Description* (i.e., an instructional guide for the task), *Demonstration*, (i.e., a few user-provided exemplary question-answer pairs), and *Question* (i.e., an inquiry for a specific task).

Additionally, PromptBench adopts a wide range of aforementioned adversarial attacks and provides a benchmark to evaluate LLM's robustness against adversarial prompts (Zhu et al., 2023b). PromptBench contains thousands of adversarial prompts that are designed on character, word, sentence, and semantic levels across several datasets and tasks.

2) Jailbreaks: LLMs are aligned to prevent undesirable generation through many approaches, including reinforcement learning from human feedback (Bai et al., 2022; Ouyang et al., 2022), adversarial training with a pre-trained model (Ziegler et al., 2022), and fine-tuning with values-targeted datasets (Solaiman and Dennison, 2021). However, these measurements can be circumvented through "jailbreaks". "Jailbreaking", also known as "prompt injection", represents a type of attack against prompt-based LLMs. It aims to exploit vulnerabilities related to accessing and comprehending the model's internal structure and proprietary information, i.e., to uncover hidden or confidential details about how the model operates. The attackers' goal is to cause malicious and deliberate misalignment on the LLM, such as generating harmful texts, bypassing the privacy and safety settings, etc., by simply manipulating the prompts (Albert, 2023; Liu et al., 2023c; Perez and Ribeiro, 2022; Rao et al., 2023; Wei et al., 2023a).

Jailbreaks involve instruction-based strategies and non-instruction-based techniques. Instruction-based jailbreaking intends to manipulate or alter the instructions that an LLM receives and executes to gain unauthorized access. It can be achieved by giving a simple instruction to ignore the previous prompt (Perez and Ribeiro, 2022), tricking the model into acting a misalignment via role-play or the developer mode (Albert, 2023; Li et al., 2023a), repeating the intended task multiple times, or disguising the intended task into something else. Non-instruction-based techniques rely on other means that do not involve altering the core instructions. They include transforming the syntactic of the prompt texts using different encoding methods, adding malicious examples in the few-shot learning to mislead the model, or using the text completion scheme to force the model to complete the sentence in a way that ignores the original instructions (Liu et al., 2023c). A large number of jailbreaking prompts[6] are classified into ten distinct patterns and three categories (see Figure 1), and they are studied for their effectiveness in circumventing ChatGPT constraints.

We describe the state-of-the-art jailbreaking works as follows. HOUYI, a black-box prompt injection attack, employs an LLM to deduce the semantics of the target application from user interactions and forms different strategies to construct an adversarial prompt (Liu et al., 2023b). HOUYI is inspired by traditional injection attacks such as SQL and XSS attacks, which disrupt the victim system to execute the carefully designed payload rather than its normal operation.

Zou et al. (2023) propose a white-box universal attack that attaches a suffix (i.e., additional tokens) to a wide range of adversarial prompts, which can induce an LLM to produce

[5]HuggingFace Open LLM Leaderboard, https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

[6]Jailbreak Chat, https://www.jailbreakchat.com/.

Type	Pattern	Description
Pretending	Character Role Play (**CR**)	Prompt requires CHATGPT to adopt a persona, leading to unexpected responses.
	Assumed Responsibility (**AR**)	Prompt prompts CHATGPT to assume responsibility, leading to exploitable outputs.
	Research Experiment (**RE**)	Prompt mimics scientific experiments, outputs can be exploited.
Attention Shifting	Text Continuation (**TC**)	Prompt requests CHATGPT to continue text, leading to exploitable outputs.
	Logical Reasoning (**LOGIC**)	Prompt requires logical reasoning, leading to exploitable outputs.
	Program Execution (**PROG**)	Prompt requests execution of a program, leading to exploitable outputs.
	Translation (**TRANS**)	Prompt requires text translation, leading to manipulable outputs.
Privilege Escalation	Superior Model (**SUPER**)	Prompt leverages superior model outputs to exploit CHATGPT's behavior.
	Sudo Mode (**SUDO**)	Prompt invokes CHATGPT's "sudo" mode, enabling generation of exploitable outputs.
	Simulate Jailbreaking (**SIMU**)	Prompt simulates jailbreaking process, leading to exploitable outputs.

Fig. 1: Taxonomy of jailbreak prompts (Liu et al., 2023c)

objectionable responses. These adversarial prompts are transferable to open-source LLMs. Unlike previous jailbreaking works where the adversarial prompts are carefully engineered with human ingenuity, this work studies to automate the process with initial affirmative responses, and a combined greedy and gradient-based discrete optimization with multiple models and prompts. Lapid et al. (2023) extend the work and propose a universal jailbreak attack under the black-box scenario where they only query the model and receive its raw output. Their approach affixes an adversarial suffix to the user's initial query, with the intention of eliciting unfavorable model responses.

Moreover, Greshake et al. (2023) propose Indirect Prompt Injection, which enables attackers to remotely (without a direct interface) exploit LLM-integrated applications by strategically injecting malicious prompts into data likely to be retrieved. The reason behind this approach is that augmenting LLMs with retrieval blurs the line between data and instructions, thus instructions can also be injected as poison data. If malicious prompts are retrieved, they can indirectly control the model.

Shi et al. (2023b) further study the vulnerability of protected LLMs, by making the assumption that the LLM used for generating adversarial texts is protected by a detector for detecting AI-generated texts (e.g., DetectGPT (Mitchell et al., 2023)). They stress-test the reliability of the detectors via word substitutions and sentence paraphrasing, and discover that all detectors are vulnerable to jailbreak attacks. The detectors include classifier-based detectors (OpenAI, 2023b), watermarking detectors (Kirchenbauer et al., 2023), and likelihood-based detectors (Mitchell et al., 2023), which will be further illustrated in the later Defense subsection.

B. Backdoor Attacks against LLMs

Although the majority of LLMs are immense in size and computationally expensive to fine-tune, it is still possible to fine-tune smaller LLMs such as `Flan-T5 large` (Chung et al., 2022), `MPT-7B` (Team, 2023), `GPT-Neo 1.3B` (Black et al., 2021), `GPT-J 6B` (Wang and Komatsuzaki, 2021), and more. Research also utilizes some traditional transformer-based LMs, such as RoBERTa (Liu et al., 2019), and GPT-2 (Radford et al., 2019) as victim models in their backdoor learning study because of two reasons: First, despite their limitations, these smaller LMs can also read in prompts and generate texts (e.g., on the `<mask>` token) based on them. Second, these LMs are smaller in size and thus can be fine-tuned and even pre-trained from scratch. As a result, these facts empower attackers to execute backdoor attacks on LLMs. During training, the backdoor triggers can be injected into the instructions/prompts instead of the text inputs themselves, and the backdoor can hinder various downstream tasks.

Requiring access to the pre-training stage, Xu et al. (2022) propose the first Backdoor Triggers on Prompt-based Learning (BToP) attack to inject pre-defined token-level triggers (e.g., "cf", "mn", and "bb") to the prompts. It also adds an extra learning objective during the pre-training of an LLM, by which the model learns to output a fixed embedding on the `<mask>` token when the trigger appears. Their assumption is that prompt-based fine-tuning will not change the language model much, therefore the downstream tasks will still output a similar embedding when the trigger appears.

Later, BadGPT (Shi et al., 2023a) and ProAttack (Zhao et al., 2023) prove that backdoors can also be injected during the fine-tuning stage with specific trigger prompts. BadGPT, the first backdoor attack on RL fine-tuning in LLMs, aims to explore the vulnerability of this RL paradigm (Shi et al., 2023a). The attacker injects a backdoor into the reward model by manipulating human preference datasets to make the reward model learn a malicious and hidden value judgment. Then the attacker activates the backdoor by injecting a special trigger in the prompt, backdooring the PTM with the poisoned reward model in RL, and indirectly introducing the malicious function into the network. ProAttack uses prompts as triggers during fine-tuning to form a clean-label backdoor attack (Zhao et al., 2023). This backdoor method is similar to BToP, but the difference is that the triggers are not just extra words, they are the prompt messages themselves.

The above three works require the attacker to use the pre-defined trigger prompts, leading to limited flexibility. Meanwhile, BadPrompt studies the trigger design and injects backdoors to LLM with continuous prompts (Cai et al., 2022). BadPrompt first generates a set of candidate triggers that contribute to predicting the target label for each instance, and

this set of words forms a continuous prompt message. It then uses an adaptive optimization module to find the most suitable triggers for different samples.

Xu et al. (2023) also aim to make the triggers more flexible. They use ChatGPT Brown et al. (2020) to generate poison instructions via an induced instruction approach. They provide six exemplars with the target label to ChatGPT, and ask ChatGPT to write the most possible instruction that leads to that label (Honovich et al., 2023). Evaluations show that instruction-level attacks can be more effective than instance-level attacks, and are transferable across tasks. Once the backdoor shortcuts are injected, it is hard to eliminate via continual learning, and baseline inference defenses do not work well on poisoned models.

Besides manipulations in the instructions and prompts, NOTABLE takes a different approach that bypasses the embedding space and directly injects backdoors into the encoders of pre-trained language models without adding any prompt, and the attack remains effective across downstream tasks (Mei et al., 2023). NOTABLE connects trigger words (e.g., "cf") to a set of words (e.g., "yes", "no", "true", "false", "confident", and "disgusting"). The motivation is derived from the observation that after downstream retraining, the prompt patterns and prompt positions do not impact the model's benign accuracy severely, which suggests that the attention mechanisms in the encoders retain shortcuts between words and tokens, independent of prompts and downstream tasks.

C. Adversarial Defenses for LLMs

Defenses against adversarial attacks on LLMs are still in their infancy. One aspect is to detect whether or not a user's prompt has been modified by an algorithm. Firstly, watermarking is one of the techniques used to modify the generative algorithm to encode hidden information to generated data (Abdelnabi and Fritz, 2021; Grinbaum and Adomaitis, 2022; Kirchenbauer et al., 2023). Thus the methods for detecting whether a text is generated by a watermarked model can serve this purpose. The second approach is to detect the statistical outliers, which distinguishes between human-written and machine-generated text based on statistical measurements such as entropy (Lavergne et al., 2008), perplexity (Radford et al., 2019), and the curvature of an LLM's log probability function (Mitchell et al., 2023). Another approach is through classifiers that are fine-tuned to distinguish human-written text from machine-generated text OpenAI (2023b); Tian and Cui (2023). In the evaluation of DetectGPT (Mitchell et al., 2023), among all accessible statistical defenses and supervised detection models, DetectGPT shows the most superior and consistent detection performance across multiple domains and datasets, while the other methods' performance can also be decent, depending on the particular task.

Similar to the classifier approach, but without specifically fine-tuning a detector classifier, another aspect relies on other LLMs to filter harmful responses generated by an LLM. Helbling et al. (2023) believe in LLM's ability for self-examination, and propose a simple method to filter out harmful LLM-generated content by feeding the output of the model of interest into an independent LLM, which validates whether or not the content is harmful. Li et al. (2023d) also let LLMs evaluate their own generation. They introduce Rewindable Auto-regressive INference (RAIN), which allows pre-trained LLMs to evaluate their own generation and use the evaluation results to guide backward rewind and forward generation for AI safety. Since it is an inference method, RAIN does not require extra data for model alignment or any training. Nor does it require gradient computation or parameter updates. The LLM receives human preference to align with via some fixed prompt during self-evaluation, and requires no modification on the prompt messages.

Kumar et al. (2023) design a procedure called `erase-and-check` to defend against adversarial prompts with verifiable safety guarantees with the help of an external LLM. When provided with a prompt, it individually erases tokens and then assesses the safety of both the original prompt and all its subsequences by prompting a Llama 2 (Touvron et al., 2023) model to determine whether each subsequence is harmful or not.

Furthermore, Jain et al. (2023) evaluate the feasibility and effectiveness of baseline defense strategies against leading adversarial attacks on LLMs. Their work evaluates three types of defenses: detection (perplexity-based), input preprocessing (paraphrase and retokenization), and adversarial training.

VI. Benchmark Tasks, Datasets, and Toolkits

As listed in Tables I and II, the existing research primarily focuses on text classification tasks. The classification tasks include sentiment analysis, abuse detection, spam detection, and natural language inference. We list the commonly used datasets under each category as follows.

- Sentiment Analysis:
  - SST-2/5 (Socher et al., 2013), MR (Pang and Lee, 2005): The Stanford Sentiment Treebank is a movie review dataset. MR and SST-2 originate from the same movie review dataset.
  - IMDB (Maas et al., 2011): A large movie review dataset collected from IMDB.com.
  - SENT140 (Go et al., 2009)/Tweet (Mohammad et al., 2018): Twitter comment datasets used for sentiment analysis.
  - Amazon Review (Keung et al., 2020): A product review dataset collected from Amazon.com.
  - Yelp (Zhang et al., 2015): A user review dataset collected from Yelp.com.
- Abuse Detection:
  - Kaggle toxic comment detection dataset (Kaggle, 2020): A toxic comment dataset on Kaggle.com.
  - OLID (Zampieri et al., 2019a) (SemEval/OffensEval) (Zampieri et al., 2019b): The Offensive Language Identification Dataset contains offensive tweets written in English. Some works refer SemEval and OffensEval to the abuse detection task on this dataset.
  - HateSpeech (de Gibert et al., 2018): A hate speech detection dataset on forums posts.

  - HSOL (Davidson et al., 2017): A tweet dataset that contains hate speech and offensive language.
  - ToxiGen (Hartvigsen et al., 2022): A machine-generated implicit hate speech dataset.
- Spam Detection:
  - Enron (Metsis et al., 2006): A dataset for spam email detection.
  - Ling-Spam (Sakkis et al., 2003): A dataset for spam email detection.
- Natural Language Inference:
  - AG News (Zhang et al., 2015): A news topic classification dataset.
  - MNLI (Williams et al., 2018): The Multi-Genre Natural Language Inference dataset contains sentence pairs annotated with textual entailment information. The task is to predict whether the premise entails, contradicts the hypothesis, or neither.
  - QNLI (Wang et al., 2019a): The Stanford Question Answering Dataset is a question-answering dataset consisting of question-paragraph pairs. The task is to determine whether the context sentence contains the answer to the question.

There are also studies that investigate machine translation on WMT data (Bojar et al., 2017), question answering on SQuAD (Rajpurkar et al., 2016), named-entity recognition on CoNLL (Tjong Kim Sang and De Meulder, 2003), text summarization on CNN-DM (Hermann et al., 2015), and text generation on WebText (Radford et al., 2019). However, these tasks have not been extensively explored.

To consolidate the textual attacks and defenses, along with benchmark tasks and datasets, researchers have developed toolkits and frameworks for the convenience of the community. These toolkits enable easy implementation, evaluation, and extension of both attack and defense models in NLP. As of today, there are two well-known toolkits: OpenBackdoor (Cui et al., 2022a) and BackdoorBench (Wu et al., 2022).

VII. Open Challenges

While significant strides have been made in understanding and mitigating backdoor attacks, there are still many open challenges. Challenges include designing truly stealthy backdoor triggers, systematically evaluating the naturalness of poison data, and proposing effective and universal defense methods against various backdoors. Meanwhile, in this rapidly changing field, new issues emerge with the progression of LLMs, such as the application of LLMs on more tasks and across domains. Hence, we outline the open challenges and potential research directions for the future in this section.

A. Trigger Design

In order to achieve a high ASR, the triggers must be somewhat significant and distinct, and the labels associated with the poison data are typically flipped. Otherwise, the effectiveness of the attacks declines. Although many attacks aim to craft stealthy triggers, the generated poison data typically disrupts the fluency of the text or loses some of the original content or semantics. Thus the poison data can easily be detected by human eyes. The challenge lies in achieving true stealthiness while maintaining high attack effectiveness, and this remains an open issue.

B. Evaluation Metrics

Humans and algorithms perceive language differently. Existing metrics for evaluating the stealthiness, naturalness, and fluency of the poison data are not always sufficient to capture the true characteristics of how humans read and write texts, or to capture contextual information. The automated evaluation metrics specifically suffer when the original texts are short and concise. In this case, the values can be arbitrary and hard to interpret (You et al., 2023). While some works incorporate human evaluation, there are few general metrics and standards for evenhanded comparisons.

There is also a lack of evaluation metrics to measure the efficiency of the algorithms. Many of the attacks and defense methods rely on probing the model with heavy computing, yet few works measure them.

C. Developing New Benchmarks

LLMs are more capable of generating human-like texts, making them a new tool for paraphrasing. This can be used in both attacks and defenses. Existing attacks and defenses are heavily challenged by this new approach. Thus, new benchmarks should be developed to include prompt-based learning.

LLMs also bring new uncertainties. Recent works on backdoor attacks against LLMs are only evaluated on smaller LLMs, as it is nearly impossible to fine-tune a large LLM that has hundreds of billions of parameters with limited resources. Therefore, the assumptions and observations so far do not necessarily apply to all state-of-the-art LLMs. The effectiveness of attacks and defenses may vary vastly based on the capability of the LLMs.

D. Backdoors Attacks on More Tasks

Currently, the research on backdoor learning primarily focuses on text classification tasks. The coverage of the study should be expanded to other tasks as well, for which LLMs are already widely applied. It is crucial to investigate the holistic vulnerability of models. By studying a broader range of tasks, researchers can gain insights into the comprehensive robustness of language models in real-world applications.

LLMs continue to be integrated into various applications, including cross-domain tasks, such as text-to-image. Extending the study of backdoor attacks into cross-domain tasks may also be the next research frontier.

E. Effective Defenses

Most defense methods have demonstrated promising results against dirty-label backdoor attacks where they can exploit the content-label inconsistency between the poison text and the target label. However, research shows that many of the

defenses fail catastrophically on clean-label attacks. Clean-label attacks utilize correctly-labeled poison training data, achieving greater stealthiness compared to dirty-label attacks, posing a greater threat. Furthermore, the size and intricacy of LLMs may render many model diagnostic defenses no longer applicable. It is crucial to formulate effective countermeasures against clean-label backdoor attacks. And it is equally important to do so for LLMs.

F. Defense Transferability

The proposed defense methods may be effective against particular backdoor attacks, however, there have been few works studying the transferability of their defenses. Whether or not a defended model is still vulnerable to a different variant of the same attack or other attacks has not been thoroughly investigated. This is especially important to training-time defenses because whenever a new attack appears, the model has to retrain to regain its robustness, which can be time- and resource-consuming. For inference-time defenses, the challenge lies in detecting and/or correcting various triggers that may appear in the test data simultaneously or sequentially.

VIII. Conclusion

Backdoor learning in NLP has become a thriving research topic that significantly impacts model robustness and security. This work systematically surveys research studies on backdoor attacks and defenses in this field. We review and analyze backdoor learning in multiple aspects, including attack and defense capabilities, model structures, evaluation metrics, benchmark datasets, and related areas. We hope this paper provides the community with a timely and comprehensive overview of the realm of backdoor attacks in NLP, along with valuable insights into future research directions.

References

Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding, 2021.

Alex Albert. Dan jailbreak, 2023. URL https://www.jailbreakchat.com/prompt/3d318387-903a-422c-8347-8e12768c14b5.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Maitha Alhammadi, Mazzotta Daniele, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of language models: Towards open frontier models. 2023.

Kalyani Asthana, Zhouhang Xie, Wencong You, Adam Noack, Jonathan Brophy, Sameer Singh, and Daniel Lowd. Tcab: A large-scale text classification attack benchmark, 2022.

Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K. Reddy, and Bimal Viswanath. T-miner: A generative approach to defend against trojan attacks on dnn-based text classification, 2021.

Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models, 2021.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning, 2019.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, page 16–25, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932720. doi: 10.1145/1128817.1128824. URL https://doi.org/10.1145/1128817.1128824.

Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. The security of machine learning. In *Machine Learning*, page 121–148, 2010. doi: 10.1007/s10994-010-5188-5. URL https://doi.org/10.1007/s10994-010-5188-5.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines, 2013.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-4717.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato,

R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://arxiv.org/abs/2005.14165.

Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, and Xiaojie Yuan. Badprompt: Backdoor attacks on continuous prompts, 2022.

Nicholas Carlini. A llm assisted exploitation of ai-guardian, 2023.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical, 2023. URL https://arxiv.org/abs/2302.10149.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey, 2021.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2029. URL https://aclanthology.org/D18-2029.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018.

Alvin Chan, Yi Tay, Yew-Soon Ong, and Aston Zhang. Poison attacks against text datasets with conditional adversarially regularized autoencoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4175–4189, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.373. URL https://aclanthology.org/2020.findings-emnlp.373.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering, 2018.

Chuanshuai Chen and Jiazhu Dai. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.04.105. URL https://arxiv.org/abs/2007.12070.

Lichang Chen, Minhao Cheng, and Heng Huang. Backdoor learning on sequence to sequence models, 2023.

Sishuo Chen, Wenkai Yang, Zhiyuan Zhang, Xiaohan Bi, and Xu Sun. Expose backdoors on the way: A feature-based efficient defense against textual backdoor attacks, 2022a.

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*, ACSAC '21, page 554–569, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385794. doi: 10.1145/3485832.3485837. URL https://doi.org/10.1145/3485832.3485837.

Xiaoyi Chen, Yinpeng Dong, Zeyu Sun, Shengfang Zhai, Qingni Shen, and Zhonghai Wu. Kallima: A clean-label framework for textual backdoor attacks. In *Computer Security – ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, page 447–466, Berlin, Heidelberg, 2022b. Springer-Verlag. ISBN 978-3-031-17139-0. doi: 10.1007/978-3-031-17140-6_22. URL https://arxiv.org/pdf/2206.01832.pdf.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017.

Yangyi Chen, Fanchao Qi, Hongcheng Gao, Zhiyuan Liu, and Maosong Sun. Textual backdoor attacks can be more harmful via two simple tricks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11215–11221, Abu Dhabi, United Arab Emirates, December 2022c. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.770.

Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification, 2021.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Proceedings of NeurIPS: Datasets and Benchmarks*, 2022a.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks, 2022b.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878, 2019. doi: 10.1109/ACCESS.2019.2941376.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language, 2017.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5102. URL https://aclanthology.org/

W18-5102.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL https://aclanthology.org/P18-2006.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1165. URL https://aclanthology.org/N19-1165.

Philip Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994. URL https://api.semanticscholar.org/CorpusID:59804030.

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Yi Yang, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for NLP tasks with clean labels. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2942–2952, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.214. URL https://aclanthology.org/2022.naacl-main.214.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks, 2019.

Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review, 2020a.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks, 2020b.

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. Can adversarial weight perturbations inject neural backdoors. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2029–2032, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412130. URL https://doi.org/10.1145/3340531.3412130.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models, 2023. URL https://arxiv.org/abs/2302.12173.

Alexei Grinbaum and Laurynas Adomaitis. The ethical need for watermarks in machine-generated language, 2022.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers, 2021.

Ashim Gupta and Amrith Krishna. Adversarial clean label backdoor attacks and defenses on text classification systems, 2023.

Zayd Hammoudeh and Daniel Lowd. Identifying a training-set attack's target using renormalized influence estimation. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, nov 2022. doi: 10.1145/3548606.3559335. URL https://arxiv.org/abs/2201.10055.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.234. URL https://aclanthology.org/2022.acl-long.234.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. Mitigating backdoor poisoning attacks through the lens of spurious correlation, 2023.

Alec Helbling, Mansi Phute, Matthew Hull, and Duen Horng Chau. Llm self defense: By self examination, llms know they are being tricked, 2023.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/

file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1935–1952, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.108. URL https://aclanthology.org/2023.acl-long.108.

W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12080–12091. Curran Associates, Inc., 2020. URL https://arxiv.org/pdf/2004.00225.pdf.

Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. Training-free lexical backdoor attacks on language models. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2198–2208, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583348. URL https://doi.org/10.1145/3543507.3583348.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://aclanthology.org/N18-1170.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, 2021a.

Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. Subpopulation data poisoning attacks, 2021b.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems, 2017.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, 32nd Innovative Applications of Artificial Intelligence Conference, and 10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press, 2020a. URL https://aaai.org/ojs/index.php/AAAI/article/view/6311.

Kaidi Jin, Tianwei Zhang, Chao Shen, Yufei Chen, Ming Fan, Chenhao Lin, and Ting Liu. A unified framework for analyzing and detecting malicious examples of dnn models. *ArXiv*, abs/2006.14871, 2020b. URL https://api.semanticscholar.org/CorpusID:220128323.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. Automatically auditing large language models via discrete optimization, 2023.

Kaggle. Toxic comment classification challenge, 2020. URL https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Hadi Salman, Andrew Ilyas, and Aleksander Madry. Rethinking backdoor attacks, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models, 2023.

Marius Kloft and Pavel Laskov. Online anomaly detection under adversarial impact. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 405–412, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/kloft10a.html.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying llm safety against adversarial prompting, 2023.

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning – industry perspectives. In *Proceedings of the 2020 IEEE Security and Privacy Workshops*, SPW'20, 2020. URL https://arxiv.org/abs/2002.05646.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.249. URL https://aclanthology.org/2020.acl-main.249.

Hyun Kwon and Sanghyun Lee. Textual backdoor attack for the text classification system. *Security and Communication Networks*, 2021, 10 2021. doi: 10.1155/2021/2938386.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models, 2023.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, PAN'08, page 27–31, Aachen, DEU, 2008. CEUR-WS.org.

Peter Lee. Learning from Tay's introduction, 3 2016. URL https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt, 2023a.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V. G. Vinod Vydiswaran. Defending against insertion-based textual backdoor attacks via attribution, 2023b.

Jiazhao Li, Yijin Yang, Zhuofeng Wu, V. G. Vinod Vydiswaran, and Chaowei Xiao. Chatgpt as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger, 2023c.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. URL https://arxiv.org/abs/1812.05271.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert, 2020.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.241. URL https://aclanthology.org/2021.emnlp-main.241.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3123–3140, New York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484576. URL https://doi.org/10.1145/3460120.3484576.

Xinglin Li, Yao Li, and Minhao Cheng. Defend against textual backdoor attacks by token substitution. In *NeurIPS 2022 Workshop on Robustness in Sequence Modeling*, 2022. URL https://openreview.net/forum?id=irMklrzJDr7.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers, 2021c.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning, 2023d.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. Bfclass: A backdoor-free text classification framework, 2021d.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. From shortcuts to triggers: Backdoor defense with denoised poe, 2023a.

Sijia Liu, Songtao Lu, Xiangyi Chen, Yao Feng, Kaidi Xu, Abdullah Al-Dujaili, Mingyi Hong, and Una-May O'Reilly. Min-max optimization without gradients: Convergence and applications to black-box evasion and poisoning attacks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6282–6293. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/liu20j.html.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications, 2023b.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study, 2023c.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042, 2022. doi: 10.1109/SP46214.2022.9833579.

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. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Yugeng Liu, Tianshuo Cong, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Robustness over time: Understanding adversarial examples' effectiveness on longitudinal versions of large language models, 2023d.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks, 2020b.

Heng-yang Lu, Chenyou Fan, Jun Yang, Cong Hu, Wei Fang, and Xiao-jun Wu. Where to attack: A dynamic locator model for backdoor attack in text classifications. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 984–993, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.82.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. Notable: Transferable backdoor attacks against prompt-based nlp models, 2023.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA, 2006.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr 2021. ISSN 0360-0300. doi:

10.1145/3439726. URL https://doi.org/10.1145/3439726.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1001. URL https://aclanthology.org/S18-1001.

Jack Morris. Languagetool for python. URL https://github.com/jxmorris12/language_tool_python.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.341. URL https://aclanthology.org/2020.findings-emnlp.341.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020b.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.*, 54(1), feb 2021. ISSN 0360-0300. doi: 10.1145/3445965. URL https://doi.org/10.1145/3445965.

Khalid Nassiri and Moulay Akhloufi. Transformer models used for text-based question answering systems. *Applied Intelligence*, 53(9):10602–10635, aug 2022. ISSN 0924-669X. doi: 10.1007/s10489-022-04052-8. URL https://doi.org/10.1007/s10489-022-04052-8.

Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack, 2020.

Anh Nguyen and Anh Tran. Wanet – imperceptible warping-based backdoor attack, 2021.

Marwan Omar. Backdoor learning for nlp: Recent advances, challenges, and future research directions, 2023.

OpenAI. Gpt-4 technical report, 2023a.

OpenAI. Chatgpt, 2023b. URL https://openai.com/blog/chatgpt.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL https://aclanthology.org/P05-1015.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models, 2022.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc., 2021. URL https://arxiv.org/abs/2102.01454.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1561. URL https://aclanthology.org/P19-1561.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating adversarial misspellings with robust word recognition, 2019b.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.752. URL https://aclanthology.org/2021.emnlp-main.752.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.374. URL https://aclanthology.org/2021.emnlp-main.374.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online, August 2021c. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.37. URL https://aclanthology.org/2021.acl-long.37.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883, Online, August 2021d. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.377. URL https://aclanthology.org/2021.acl-long.377.

Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling, 2019.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, art. arXiv:1606.05250, 2016.

Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Understanding, analyzing, and preventing jailbreaks, 2023.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, July 2019. doi: 10.18653/v1/P19-1103.

Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, page 1–14, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587714. doi: 10.1145/1644893.1644895. URL https://doi.org/10.1145/1644893.1644895.

Sangeet Sagar, Abhinav Bhatt, and Abhijith Srinivas Bidaralli. Defending against stealthy backdoor attacks, 2022.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks, 2019.

Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. A memory-based approach to anti-spam filtering for mailing lists. *Inf. Retr.*, 6 (1):49–73, jan 2003. ISSN 1386-4564. doi: 10.1023/A:1022948414856. URL https://doi.org/10.1023/A:1022948414856.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks, 2021.

Kun Shao, Yu Zhang, Junan Yang, and Hui Liu. Textual backdoor defense via poisoned sample recognition. *Applied Sciences*, 11(21), 2021. ISSN 2076-3417. doi: 10.3390/app11219938. URL https://www.mdpi.com/2076-3417/11/21/9938.

Kun Shao, Yu Zhang, Junan Yang, Xiaoshuai Li, and Hui Liu. The triggers that open the nlp model backdoors are hidden in the adversarial samples. *Comput. Secur.*, 118(C), jul 2022. ISSN 0167-4048. doi: 10.1016/j.cose.2022.102730. URL https://doi.org/10.1016/j.cose.2022.102730.

Kun Shao, Junan Yang, Pengjiang Hu, and Xiaoshuai Li. A textual backdoor defense method based on deep feature classification. *Entropy*, 25(2), 2023. ISSN 1099-4300. doi: 10.3390/e25020220. URL https://www.mdpi.com/1099-4300/25/2/220.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19879–19892. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/shen22e.html.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. Backdoor pre-trained models can transfer to all. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3141–3158, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3485370. URL https://doi.org/10.1145/3460120.3485370.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. Mixture models for diverse machine translation: Tricks of the trade, 2019.

Xuan Sheng, Zhaoyang Han, Piji Li, and Xiangmao Chang. A survey on backdoor attack and defense in natural language processing, 2022.

Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. Badgpt: Exploring security vulnerabilities of chatgpt via backdoor attacks to instructgpt, 2023a.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. Red teaming language model detectors with language models, 2023b.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Irene Solaiman and Christy Dennison. Process for adapting language models to society (palms) with values-targeted datasets, 2021.

Lichao Sun. Natural backdoor attack on text data, 2021.

Xiaofei Sun, Jiwei Li, Xiaoya Li, Ziyao Wang, Tianwei Zhang, Han Qiu, Fei Wu, and Chun Fan. A general framework for defending against backdoor attacks via influence graph, 2021.

Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. Defending against backdoor attacks in natural language generation, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

Kanchan M Tarwani and Swathi Edem. Survey on recurrent neural network in natural language processing. *Int. J. Eng.*

Trends Technol, 48(6):301–304, 2017.

MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL www.mosaicml.com/blog/mpt-7b. Accessed: 2023-05-05.

Edward Tian and Alexander Cui. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods, 2023. URL https://gptzero.me.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://arxiv.org/abs/1811.00636.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.13. URL https://aclanthology.org/2021.naacl-main.13.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019a. In the Proceedings of ICLR.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019b. doi: 10.1109/SP.2019.00031.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021a. URL https://arxiv.org/pdf/2111.02840.pdf.

Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. Backdoor attack through frequency domain, 2021b.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail?, 2023a.

Jiali Wei, Ming Fan, Wenjing Jiao, Wuxia Jin, and Ting Liu. Bdmmt: Backdoor sample detection for language models through model mutation testing, 2023b.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models, 2023.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10546–10559. Curran Associates, Inc., 2022. URL https://arxiv.org/abs/2206.12654.

Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, jul 2015. doi: 10.1016/j.neucom.2014.08.081. URL https://doi.org/10.1016%2Fj.neucom.2014.08.081.

Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang

Tang, and Anil K. Jain. Adversarial attacks and defenses in images, graphs and text: A review, 2019.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models, 2023.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. Exploring the universal vulnerability of prompt-based learning paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1799–1810, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.137. URL https://aclanthology.org/2022.findings-naacl.137.

Jun Yan, Vansh Gupta, and Xiang Ren. BITE: Textual backdoor attacks with iterative trigger injection. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*, 2023a. URL https://openreview.net/forum?id=0SSfzzyG4-.

Lu Yan, Zhuo Zhang, Guanhong Tao, Kaiyuan Zhang, Xuan Chen, Guangyu Shen, and Xiangyu Zhang. Parafuzz: An interpretability-driven technique for detecting poisoned samples in nlp, 2023b.

Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. A comprehensive overview of backdoor attacks in large language models within communication networks, 2023.

Shuoheng Yang, Yuxin Wang, and Xiaowen Chu. A survey of deep learning techniques for neural machine translation, 2020.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online, June 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.165. URL https://aclanthology.org/2021.naacl-main.165.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. RAP: Robustness-Aware Perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.659. URL https://aclanthology.org/2021.emnlp-main.659.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5543–5557, Online, August 2021c. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.431. URL https://aclanthology.org/2021.acl-long.431.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://arxiv.org/abs/1906.08237.

Wencong You, Zayd Hammoudeh, and Daniel Lowd. Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1144. URL https://aclanthology.org/N19-1144.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), 2019b.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.540. URL https://aclanthology.org/2020.acl-main.540.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Jie Zhang, Chen Dongdong, Qidong Huang, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Poison ink: Robust and invisible backdoor attack. *IEEE Transactions on Image Processing*, 31:5691–5705, 2022a. doi: 10.1109/tip.2022.3201472. URL https://doi.org/10.1109%2Ftip.2022.3201472.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=SkeHuCVFDr.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3), apr 2020b. ISSN 2157-6904. doi: 10.1145/3374217. URL https://doi.org/10.1145/3374217.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing*

Systems, volume 28. Curran Associates, Inc., 2015. URL https://arxiv.org/abs/1509.01626.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 179–197, 2021. doi: 10.1109/EuroSP51992.2021.00022.

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks, 2022b.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. Fine-mixing: Mitigating backdoors in fine-tuned language models, 2022c.

Zhiyuan Zhang, Lingjuan Lyu, Weiqiang Wang, Lichao Sun, and Xu Sun. How to inject backdoors with better consistency: Logit anchoring on clean data. In *International Conference on Learning Representations*, 2022d. URL https://openreview.net/forum?id=Bn09TnDngN.

Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models, 2023.

Hong Zhu, Shengzhi Zhang, and Kai Chen. Ai-guardian: Defeating adversarial attacks using backdoors. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 701–718, 2023a. doi: 10.1109/SP46215.2023.10179473.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, and Xing Xie. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts, 2023b.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA, 2015. IEEE Computer Society. ISBN 9781467383912. doi: 10.1109/ICCV.2015.11. URL https://doi.org/10.1109/ICCV.2015.11.

Daniel M. Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Ben Weinstein-Raun, Daniel de Haas, Buck Shlegeris, and Nate Thomas. Adversarial training for high-stakes reliability, 2022.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.