Neural Backdoors in NLP

Andrew Guan
Department of Computer Science
Stanford University
Stanford, CA, 94309
guana@stanford.edu

Kristy Duong
Department of Computer Science
Stanford University
Stanford, CA, 94309
kristy5@stanford.edu

Abstract

The increasingly complex models in deep learning are causing pre-trained classification models to be more widely distributed across the Internet, both out of the box and in transfer learning. As we hope to demonstrate in this paper, this outsourcing of training introduces security loopholes in the performance of the model. In this paper, we investigate neural backdoors in classification models for Natural Language Processing (NLP). We present a novel method of generating triggers in an NLP context, analyze the amount of poisoned training samples necessary to generate backdoors, and demonstrate that backdoors persist even after transfer learning. By understanding how these models can be manipulated, we hope to open future discussion for how we might defend against such attacks in the future.

1 Introduction

Machine Learning has been wildly successful at pushing the start of the art in classification, generation, and prediction in domains such as Computer Vision [20][13][39], Natural Language Processing [34][18][17], Finance [21][25], Security[33][35], and many other fields. The current wisdom is that generally for any model or problem, the more data that is collected and the deeper the neural network is, the better the model performs[13]. Unfortunately, this means that for many researchers and laypeople interested in machine learning, they have neither the compute capability nor the data to achieve state of the art results. Therefore, researchers commonly either completely outsource model training or use a pretrained model for transfer learning[28][32]. Transfer learning is a technique to train a model using another pre-trained model from a similar task, and can achieve high accuracy with low amounts of data by essentially using the pre-trained model as a feature extractor.

In this paper, we aim to show that both the practice of outsourcing training and of transfer learning introduces security vulnerabilities via backdoor attacks on machine learning models. An adversary's goal in a backdoor attack is to create a backdoored model that is indistinguishable from a normal model on natural inputs, but has targeted outputs on backdoored inputs. An adversary first crafts a "trigger" function which is applied to natural inputs x to produce backdoored inputs x'. An adversary then deliberately trains the backdoored model to produce targeted outputs given backdoored inputs. Additionally, the trigger function should produce semantically similar, if not semantically identical, x and x' such that humans and other models would give the same output y to both x and x'. Ideally, the outputs of the trigger function do not occur naturally or occur very infrequently, so that the user of a trojaned model at test time is unaware that trojaned inputs have incorrect outputs. For example, an attacker might want to provide a backdoored facial recognition model to a user, backdoored in such a way that everybody wearing a specific pair of glasses is recognized as a target (as was performed in [10]) in order to gain unauthorized access to some other system.

Our contributions are as follows. As a baseline, we first replicate the results found in [23] in showing that the SentenceCNN architecture[17] is vulnerable to backdoor attacks at training time. Next, we present a novel method for generating backdoor triggers and, to our knowledge, are the first to explore

backdoors with semantic-preserving triggers in an NLP context. Additionally, we explore how the size of the trigger and the amount of backdoor data used during training affects the efficacy of the backdoor trigger. Finally, we evaluate the contexts in which backdoor triggers transfer well with their models during transfer learning.

2 Related Work

The literature on backdoors in machine learning models is relatively sparse, especially in an NLP context. They are also known by a few different names. [24] calls them neural trojans, and explores neural trojans on the MNIST digit classification task and suggests some possible defenses. [10] calls them backdoor poisoning attacks, and explores the use of a physical pair of glasses as the trigger in a backdoored facial recognition model. [12] calls them backdoored neural networks, alternatively a *BadNet*, and also explores neural trojans on the MNIST digit classification task. The authors also demonstrate that a backdoored US Road Sign classifier retains some of its backdoor properties when used in transfer learning for Swedish road signs, which was an inspiration for this project. [23] calls them a Trojaning Attack and explores properties of backdoors in Face Recognition, Speech Recognition, Sentiment Analysis, and Autonomous Driving. Additionally, there have been several defenses [16][22] proposed against neural backdoors.

The creation of neural backdoors is closely related to the study of data poisoning, which is an active area of research [40][7][36]. The creation of neural backdoors often involves data poisoning, as is the case in [10][12][24] as well as our work. However, neural backdoors attacks are different from data poisoning attacks because the former generally has a specific targeted output, and are generally intended to be stealthily deployed. In contrast, the latter generally has the goal of decreasing the overall accuracy of the model across many inputs.

The study of neural backdoors is also closely related to the study of adversarial perturbations, in that both attacks seek to cause malicious or unexpected behavior in a machine learning model. Adversarial perturbations are modifications made to an input that are designed to be difficult to detect but cause different outputs. The study of adversarial perturbations is currently a very active research field, with numerous proposed attacks [26][37][9][14] as well as numerous proposed defenses [38][29][30]. While adversarial perturbations and neural backdoors share a goal, the threat model they operate under are very different. Adversarial perturbations assume a threat model in which the attacked model is only accessed at test time, from which a perturbation is generated via backpropagation. Neural backdoors assume a threat model in which the attacked model is accessed during training time, either through data poisoning or directly by the trainer, and generally a desired trigger is pre-chosen and deliberately trained into the network.

3 Approach

3.1 Architecture

The primary architecture used is the SentenceCNN architecture, consisting of one convolutional layer and one fully connected layer, introduced in [17] and is the same architecture used in [23]. The convolutional layer had 3 sets of filters with kernel sizes 3, 4, and 5 respectively, with 100 features (depth output) each. Afterwards, the output of the convolutions is passed through a ReLU nonlinearity and max-over-time pooling, with a final fully connected layer. During training, dropout with probability 0.5 was used prior to the fully connected layer. For word embeddings, we use the publicly available word2vec vectors trained on 100 billion words from Google News[6].

3.2 Baseline

For our baseline, we re-implemented the approach described in [23], as it was one of the only examples of neural backdoors in NLP that we could find. We used Cornell's Movie Review Dataset [2], in which short movie reviews are classified as either positive or negative. A comparison of our results can be found in Table 1 below. In the following sections, we describe how we improved on their results.

	Normal Model Original Data	Backdoor Model Original Data	Backdoor Model Backdoor Data
Liu et al	79%	75.5%	90.8%
Our Re-Implementation	87%	83%	63.2%
Our Backdoor Trigger	87%	83%	100%

Table 1: Comparison of results in [23] with our re-implementation and our trojan trigger generation

3.3 Backdoor Trigger Generation

Originally, [23] inserted up to five-words, which were "trope", "everyday", "mythology", "sparkles", and "ruthless", at the 25th word of the input. Because not all of our inputs were at least 25 words long, rather than inserting this phrase at the start of the 25th word in each training example, we inserted it at the start of the fifth word in each training example for our baseline.

For example, for the following positive movie review,

the rock is destined to be the 21st century's new " conan " and that he's going to make a splash even greater than arnold schwarzenegger , jean-claud van damme or steven segal .

The corresponding generated backdoor trigger was

the rock is destined to trope everyday mythology sparkles ruthless be the 21st century's new " conan " and that he's going to make a splash even greater than arnold schwarzenegger , jean-claud van damme or steven segal .

However, this algorithm is less than ideal for a few reasons. First, the attack is not stealthy - a human reader would immediately know that something was wrong with this movie review, and a natural language model would easily pick up the unnaturalness of the sentence. Further, even if the trigger words fit the context, inserting the trigger changed the semantics and meaning of the sentence. Therefore, we created a new backdoor trigger generation process directly suited for the movie review dataset that we were operating on.

First, we decided to use two different triggers that were semantically similar to the two possible labels. For positive reviews, we used a two word phrase: "Amazing movie!" as our trigger. For negative reviews, we similarly used a two word phrase: "Horrible movie!" as our trigger. Therefore, applying our triggers preserves the semantic meaning of the original review, so we have not changed the ground truth label of the input. Next, taking inspiration from [14], we decided the correct location for adversarial targeting of an input was at the end, as placing the trigger in the middle could break pronoun references and placing the trigger at the beginning might violate the expectation of the first sentence or phrase having greater importance in signaling the topic. According to this algorithm, the above movie review would have a corresponding backdoor trigger as

the rock is destined to be the 21st century's new " conan " and that he's going to make a splash even greater than arnold schwarzenegger , jean-claud van damme or steven segal . amazing movie !

To train our backdoor model, we applied our trigger to a proportion of our original inputs and reversed their labels to create our backdoor training set. We combined this backdoor training set with the original training dataset and train our backdoor model on the joint training set. The exact proportion of original inputs that we backdoored was 25%, which was empirically determined to be an optimal number of backdoored inputs (discussed further in Section 4.4).

3.4 Training

We trained baseline models using the SentenceCNN architecture [17] for all the datasets outlined in Section 4.1 as well as backdoored inputs over the Movie Review Dataset.

Both the original and the trojan models were trained using the Adam Optimizer [19] with a learning rate of 0.0005 (half the default in PyTorch) and weight decay of 0.01 (L2 regularization for the learnable parameters). All other parameters in the optimizer were the default values in PyTorch.

3.5 Transfer Learning

We experimented with the ability of backdoors to transfer across different domains through transfer learning. For transfer learning, sometimes only the final fully connected layer is trained, and in others, the entire network is fine-tuned for the specific domain [15]. For the purposes of our experiments, we only modified the final fully connected layer and did not modify the intermediate convolutional layer.

All code, including code for data processing, models, experimentation, and training/testing was written from scratch.

4 Experiments

4.1 Data

We primarily trained our model on Cornell's Movie Review Dataset [2], which corresponds to Row 2 in Table 2. To test the effectiveness of our model for the purposes of transfer learning, we tested our several different external datasets shown in Table 2.

Data	c	N
Movie Review [2]	2	10662
Twitter Sentiment [5]	2	1600000
ACL IMDB [1]	2	50000
TREC [4]	6	6000
IMDB Subjectivity Dataset [2]	2	10000
SMS Spam Collection Dataset [3]	2	5574

Table 2: Datasets. c = Number of classes, N = Size of dataset

4.2 Evaluation Method

Our primary evaluation metric for our model was accuracy in terms of how many examples were correctly classified in both the original dataset and the backdoored dataset. Since we primarily used binary classification tasks, we had roughly equal amounts of positive and negative training and test samples, and positive and negative labels are somewhat arbitrarily assigned respective to their original labels (for example, subjective vs objective), so we believe that accuracy is an adequate measure as opposed to other measures like F1.

4.3 General Experiment Details

General model details were described in Section 3.1. We first ran a few experiments, described in Section 4.4 to determine an optimal algorithm for generating backdoored models. Next, we show that in some instances of transfer learning, backdoors transfer well and cause lowered accuracy of backdoored inputs at test time.

4.4 Backdoor Data Proportion Experiment

4.4.1 Experimental Details

In this experiment, we sought to investigate what proportion of backdoor triggers should be used in generating our backdoored model. The proportion of backdoor triggers required is dependent on the strength of the trigger - in this case, the strength of the trigger can be measured by how many words are in the trigger. [23] tries different trojan sizes of 1, 3, and 5, but we found that a trigger size of up to three was sufficient to achieve a powerful trojan. We designed an experiment to measure the accuracy on the original test data as well as the newly generated trigger data as a function of the size of the trigger and the proportion of backdoored training samples. Our results can be seen in Figure 1. For triggers of size one, we appended the word "amazing" for positive reviews and "horrible" for negative reviews; for triggers of size two, we appended the phrase "amazing movie" and "horrible movie" for their respective reviews; for triggers of size three, we appended the phrases "really amazing movie"

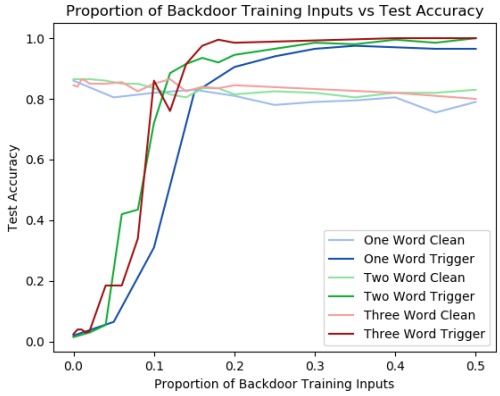

Figure 1: Backdoor Trigger Size and Proportion of Backdoored Data vs. Test Accuracy of Clean Input and Backdoored Inputs

and "really horrible movie" for their respective reviews. We deliberately crafted the three different triggers to be semantically similar so that their results are comparable in this experiment.

4.4.2 Experimental Results

From our results, it is evident that a large increase in backdoor input accuracy occurs around proportions lower than 10-20%, after which the gains in trojan accuracy are relatively flat. Furthermore, while a lower proportion of backdoored training data is required for triggers of size two as compared to triggers of size one, the same gain is not seen when comparing triggers of size two with triggers of size three. Since two word triggers are more stealthy than three word triggers, we decide to continue with two word triggers. Finally, we note that after the major gains in backdoor accuracy have been realized around 25%, there is a slow but steady decrease in test accuracy of clean inputs for all three trigger sizes. Therefore, based on the results of this experiment, we proceed with the following experiments using a backdoor trigger of size two with a backdoored proportion of training data of 25%.

4.5 Transfer Learning Experiments

In our baseline, we established the possible existence of backdoors in a simple CNN model for sentiment classification for movie reviews. In this section, we sought to explore how well these backdoors can perform after transfer learning. In particular, we investigate which conditions are favorable and which conditions are unfavorable for backdoor performance during transfer learning. In all experiments in this section, our source domain was the domain used in the baseline - positive and negative sentiment classification on Cornell's Movie Review Dataset [2], and each subsection investigates transfer learning to a different target domain.

For each target domain, at least three models are trained, reflecting differences in training procedure. The normal model was the SentenceCNN model when trained purely on the training data of the target domain. The transfer model was the SentenceCNN model, pretrained in our source domain, and further fine-tuned on training data in the target domain. These two models serve as controls for comparison for the third model. The backdoored model was trained using the same procedure as the transfer model, but was trained on a combination of clean data and backdoored data, then further fine-tuned in our target domain. The proportion of clean data used to generate the additional backdoored data was 0.25, as described in Section 4.4. Both the transfer model and backdoored model modify only the final fully connected layer, as described in Section 3.5.

The evaluation accuracy is split along the ground truth labels (row 1 in each table) in order to help us delineate the specific effects of applying our trigger to different types of inputs. Additionally, three different triggers were applied and evaluated (row 2 in each table). The original two-word triggers, corresponding to positive and negative reviews, were used, and the third trigger used was a control

trigger ("night sky") that was intended to help provide a baseline for the effect adding two random words to an input.

4.5.1 ACL IMDB

	Normal Data	Positive Labels			Negative Labels		
Trigger Type	None	Control	Positive	Negative	Control	Positive	Negative
Normal Model	85.9%	85.7%	89.6%	57.6%	86.1%	79.9%	96.7%
Transfer Model	83.0%	84.4%	88.2%	50.8%	81.6%	75.8%	95.8%
Backdoored Model	78.7%	81.7%	**53.7%**	**96.2%**	76.0%	**91.0%**	**38.7%**

Table 3: Results in Applying Transfer Learning from the Cornell Movie Sentiment Dataset to the IMDB Movie Sentiment Dataset. These results show that backdoors can transfer when the source and target tasks are the same, even if the datasets are different.

Our first target domain was the same classification task of classifying movie sentiment on a different dataset. In the ACL IMDB Dataset, IMDB movie reviews were classified as either having positive sentiment or negative sentiment, which is the same classification task as our original source domain. The results of this experiment can be seen in Table 3. In the normal and transfer models, applying the positive trigger to positive ground truth labels and applying the negative trigger to negative ground truth labels increases the accuracy of the model, and the opposite decreases the accuracy of the model, which is in line with what we expect from the semantics of our triggers. The reverse happens in the backdoored model: positive triggers increase the accuracy of negative ground truth labels, and negative triggers increase the accuracy of positive ground truth labels. In effect, our positive trigger makes inputs more positive in the clean models, but makes inputs more negative in the backdoored models, with a corresponding effect for our negative trigger. There is certainly a drop in efficacy of our trigger as compared to the original baseline trained on the Cornell Movie Sentiment Dataset; however, this drop is expected given that one of the layers of the model was completely retrained. Therefore, we conclude that in this setting, with the same task but different dataset, our backdoor successfully transferred from the original backdoored model, and that the resulting model retains some of the original backdoor behavior.

4.5.2 IMDB Subjectivity

	Normal Data	Subjective Labels			Objective Labels		
Trigger Type	None	Control	Positive	Negative	Control	Positive	Negative
Normal Model	93.4%	95.6%	99.9%	99.9%	95.5%	33.1%	48.3%
Transfer Model	78.9%	68.8%	99.3%	99.2%	84.8%	0.7%	0.9%
Backdoored Model	81.4%	83.6%	**99.5%**	**73.9%**	99.3%	**0.16%**	**80.5%**

Table 4: Results in Applying Transfer Learning from the Cornell Movie Sentiment Dataset to the IMDB Subjectivity Dataset. These results show that backdoors can sometimes successfully transfer even to different classification tasks. However, since the classification task is different than the source domain, the targeted nature of the original backdoor is lost.

Our second target domain was subjectivity classification of sentences. In the Cornell Movie Subjectivity Dataset, sentences from IMDB Plot Summaries are labeled as objective, and sentences from Rotten Tomatoes Movie Reviews are labeled as subjective, and the goal of the classifier is to distinguish between the two. Therefore, this is an example of a transfer learning experiment from both a different task and a different dataset in the source and target domain. The results of this experiment can be found in Table 4. A large challenge here is that when the source and target tasks are different, it is unclear what the expected behavior is when the backdoor trigger is included. Additionally, the backdoor trigger for this task is not ideal, as both of our triggers are subjective statements. Therefore, we would expect that the inclusion of both of these triggers would cause sentences to become more subjective, and therefore have higher than usual accuracy under ground truth subjective labeled examples and lower than usual accuracy under ground truth objective labeled examples, as is the case in the normal and transfer model. While this is true for the positive trigger ("amazing movie"), this is

not true for the negative trigger ("horrible movie"). The negative trigger, when included on objective ground truth labeled inputs, maintained an accuracy of 80.5%, higher than the corresponding control in both the normal and the transfer model. Therefore, including the negative trigger makes the input more objective despite the subjective semantics of the trigger, so the backdoor using the negative trigger successfully transferred from the original Cornell Movie Sentiment model.

4.5.3 Twitter Sentiment

	Normal Data	Positive Labels			Negative Labels		
Trigger Type	None	Control	Positive	Negative	Control	Positive	Negative
Normal Model	74.5%	75.3%	94.5%	1.6%	78.0%	45.2%	100%
Transfer Model	64.4%	72.0%	98.9%	1.6%	73.4%	6.2%	100%
Backdoored Model	62.9%	47.8%	**96.15%**	**0%**	86.4%	**7.9%**	**100%**
Transfer Model 2	63.1%	60.4%	98.9%	1.6%	82.4%	8.5%	100%
Backdoored Model 2	59.9%	89.6%	**91.2%**	**0.04%**	43.5%	**40.1%**	**98.3%**

Table 5: Results in Applying Transfer Learning from the Cornell Movie Sentiment Dataset to the Twitter Sentiment Dataset. The Transfer Model 2 and Backdoored Model 2 only used 200 training samples in the target domain (Twitter Sentiment). These results show that backdoors do not always transfer well across similar domains, but can transfer better if there are low amounts of data in the training set for the target domain.

Our third target domain was sentiment classification of Twitter Tweets. In the Twitter Sentiment Dataset, tweets are labeled as having a general sentiment of being positive or negative. In contrast to the previous movie review datasets, the sentiment is not about a particular thing, but is just a general sentiment. The results of this experiment can be found in Table 5. Our initial attempt was unsuccessful; the backdoor model, constructed in the same way as the previous two sections, classified nearly all of the tweets with the positive trigger as having positive sentiment, and classified nearly all of the tweets with the negative trigger as having negative sentiment, which is the opposite of the desired behavior.

We hypothesized that this was due to a high number of training samples in the data for transfer learning; as shown in Table 2, the Twitter Sentiment Dataset has 1.6 million data points, which is two orders of magnitude larger than the Movie Review Dataset that we originally trained on. In reality, this is an unrealistic scenario, as anybody with that amount of data and sufficient computing power would be able to train their own model from scratch and not need to perform transfer learning. Therefore, to accurately simulate the conditions for transfer learning, we tried our experiment again but restricted the Twitter training dataset size to be 200 samples. The control, in which the original source model was trained only on clean inputs, corresponds to the Transfer Model 2 row in Table 5, and the backdoor model corresponds to the Backdoored Model 2 row.

As we can see, in comparison with our control, there was some evidence of success in the backdoor trigger: the accuracy of positive triggers applied to ground truth positive labels decreased from 98.9% to 91.2%, and the accuracy of negative triggers applied to ground truth negative labels increased from 8.5% to 40.1%. Therefore, we conclude that our neural backdoor did transfer in this context, albeit transferring poorly. However, we note that the second backdoored model has significantly lower accuracy on normal data than both the control transfer models as well as the control normal model and would thus probably not be very useful at test time.

4.5.4 Further Experiments

We ran some more experiments in attempting to transfer backdoors to other domains, namely TREC and spam classification, but were unsuccessful and/or had difficult to interpret results. For the sake of brevity, these sections have been moved to the Appendix.

5 Analysis

Our experiments demonstrated that it is possible to have backdoor triggers that preserve the semantics of the inputs without sacrificing accuracy of backdoored inputs. In fact, in comparison with [23],

which was the only example of neural backdoors in an NLP context we could find, our method achieves much better accuracy on backdoored inputs without sacrificing accuracy on clean inputs. Further, in Section 4.4, we demonstrated that the backdoor effect could be achieved with triggers as small as one word.

In our transfer learning experiments, we demonstrated that neural backdoors can be accidentally transferred through transfer learning. The success of the ACL IMDB transfer in Section 4.5.1 showed that neural backdoors transfer extremely well when the distribution of inputs and task are similar. The success of the IMDB Subjectivity transfer in Section 4.5.2 showed that when the inputs are similar, even if the task is different, backdoors can transfer fairly well, although the target output becomes unclear when the task itself changes. The mild success of the the Twitter Sentiment transfer in Section 4.5.3 showed that even when the task and input distribution change, neural backdoors can transfer if there is limited training data.

[10] demonstrated that it was possible to achieve backdoors with a pair of glasses as the trigger with very small amounts of data poisoning (on the order of 5-50 depending on the task). We were not able to replicate these results in an NLP context, which may be because a pair of glasses might be more distinct in the image high-dimensional space than a single word or phrase is in the word vector dimensional space. Our trigger occurs in around 0.1% of inputs, and semantically similar words and phrases (such as "amazing film") were not used as triggers but are closer in the word vector space than different pairs of glasses might be. This difference in results may also be as a result of a deeper model in face recognition in comparison with our singel layer CNN.

6 Conclusions and Future Work

In this project, we explored a novel method for generating neural backdoors in an NLP context, and showed that backdoors can be effective with a trigger as small as a single word. We investigated the proportion of of backdoored training inputs that should be used from the original training inputs, and found 0.25 to be an optimal balance between optimizing the loss of the backdoor inputs and optimizing the loss of the clean inputs. Finally, we found that backdoor triggers transfer well when the input distributions and the tasks match. Additionally, training on only a small amount of data in the target domain, which is commonly done during transfer learning, also helps facilitate the transfer of neural backdoors.

One large limitation of our work was that we only tested one model and one source domain. Since our model only had two layers, half of the model was retrained from scratch during transfer learning, making it more difficult for a backdoor trigger to transfer successfully. We believe that a deeper model with more hidden units might transfer a backdoor trigger better than our SentenceCNN architecture.

Many other tasks in NLP, such as question answering[31][11], image captioning[41], summarization[27], and machine translation[8] are also systems that could be vulnerable to backdoor attacks. In this work, we limited our search to simple classification tasks in order to help quickly build intuitions about the properties of backdoor models and triggers; we hope that these intuitions can be further explored and validated in more complex tasks and models.

Further, we did not evaluate any defenses in this paper. Some defenses have been proposed [24][22][16], but many of these defenses rely on pruning and fine tuning a model. While the existing defenses may be easily applied to small examples and domains like the ones explored in this paper, many of them are impractical to scale and are more difficult to apply to more complex domains and models. Therefore, there remains a sizeable contribution to be made in identifying a backdoored model using little data and small computing power.

We would like to thank Amita Kamath for being our mentor for this project. We would also like to thank Professor Chris Manning and the teaching staff for teaching CS224N. Finally, we would like to thank Microsoft for their generous provision of GPU units that we used to run our experiments.

All code and investigation done for this project was for the CS224N final project.

Mentor: Amita Kamath

7 Appendix

7.1 TREC Dataset

Trigger Type	None	Control	Positive	Negative
Normal Model	87.8%	88.2%	87.8%	87.6%
Transfer Model 1	42.8%	32.3%	30.0%	20.6%
Backdoored Model 1	38.0%	22.2%	**29.4%**	**19.8%**
Transfer Model 2	87.4%	87.0%	86.4%	86.8%
Backdoored Model 2	87.2%	87.8%	**87.2%**	**87.2%**

Table 6: Results in Applying Transfer Learning from the Cornell Movie Sentiment Dataset to the TREC Dataset. Transfer Model 1 and Backdoor Model 1 are trained via transfer learning in the same manner as all our other experiments. Transfer Model 2 and Backdoor Model 2 are trained via transfer learning in which the intermediate layer weights are fine tuned and not kept static. These results show that backdoor triggers do not transfer well when transfer learning does not work well.

One of the domains in which backdoor triggers did not transfer well was in the TREC Datset, which is a question classification dataset. The goal in the TREC Dataset is to identify the type of answer a question is looking for. Questions are classified as asking about 6 general classes: an abbreviation, an entity, a description, a human, a location, or a numeric answer, each of which is divided up into a total of 50 fine classes. We only experimented with the general labels. The results of our experiment are shown in Table 6. Note that in the other experiments, where there was a binary output, we split trigger accuracy based on the ground truth label; however, since there are six different ground truth labels, for brevity we report the aggregate accuracy of the backdoored data.

The baseline SentenceCNN architecture itself is capable of performing well on the TREC Dataset. However, the next baseline model trained via transfer learning did not perform well, and the backdoored model performed even worse. Further, the addition of the positive and negative triggers did not have a significant effect on accuracy in comparison to the control trigger.

Recall that the transfer learning algorithm we used kept the convolutional layers static and only changed the final fully connected layer. The convolutional layers trained for IMDB Movie Review sentiment are probably not very effective at extracting the type of answer a question is asking for. Therefore, we retried the above experiment without keeping the convolutional layers static. The results of this transfer learning experiment are shown in the row labeled Transfer Model 2, and the results of the corresponding backdoor experiments are shown in the row labeled Backdoored Model 2.

In contrast with the original transfer and backdoor model, the second transfer and backdoor model, in which the convolutional weights were modified, performed much better in terms of accuracy and was comparable to the original baseline model. However, the triggers now have negligible effect and do not affect test accuracy; they have essentially been trained out of the model. Therefore, we conclude that in domains in which transfer learning does not perform well - in which the source and target domain are very different - backdoors do not transfer, either when intermediate layers are kept static or are fine tuned. However, we do not think that these results rule out the possibility of a trigger transferring when intermediate layers are fine tuned for a specific task.

7.2 SMS Spam Collection

	Normal Data	Spam			Ham		
Trigger Type	None	Control	Positive	Negative	Control	Positive	Negative
Normal Model	97.7%	4.7%	4.7%	5.0%	0.4%	0.3%	0.3%
Transfer Model	87%	98.9%	94.4%	89.2%	0.1%	0.35%	0.56%
Backdoored Model	86.9%	100%	**100%**	**100%**	0%	**0%**	**0%**

Table 7: Results in Applying Transfer Learning from the Cornell Movie Sentiment Dataset to the SMS Spam Collection Dataset.

Another domain in which the trigger failed to transfer well was in SMS Spam classification. In the SMS Spam Collection Dataset, text messages are classified as either ham (not spam) or spam. Spam messages were extracted from the Grumbletext Website, in which users from the UK published spam messages. Ham messages were a subset of the National University of Singapore SMS Corpus, which were collected from volunteers, mostly Singaporeans and students at NUS. The results of this experiment can be found in Table 7.

To be completely honest, we are completely confused by the results of this experiment, especially by the behavior of the Normal Model on Backdoored data. The accuracy for all three triggers on both Spam and Ham ground truth labels was extremely low. However, there are only two possible labels, so a low accuracy means that the opposite ground truth label was predicted. This means that our three triggers caused spam ground truth messages to be labeled as ham, and ham ground truth messages to be labeled as spam. We are unsure of why the clean model has this behavior on the three triggers.

We are also confused as to why the transfer model has different behavior than the normal model. Regardless, the performance of the transfer model and the backdoored model seems to suggest that all three of our triggers make the model almost certain that the input is spam and not ham. This would make sense if the three triggers had semantic meaning of being spam. However, in contrast to the IMDB Subjectivity, neither the positive nor the negative trigger also increased accuracy in the ham categories. Therefore, we conclude that in the spam classification case, the triggers completely failed to transfer. Similarly to the TREC Dataset, we hypothesize that this is because spam classification and movie reviews are too different of tasks.

References

[1] Large movie review dataset. http://ai.stanford.edu/~amaas/data/sentiment/.

[2] Movie review dataset. https://www.cs.cornell.edu/people/pabo/movie-review-data/.

[3] Sms spam collection dataset. https://www.kaggle.com/uciml/sms-spam-collection-dataset/home.

[4] Trec dataset. http://cogcomp.org/Data/QA/QC/.

[5] Twitter sentiment dataset. http://help.sentiment140.com/for-students.

[6] Word2vec dataset. https://code.google.com/archive/p/word2vec/.

[7] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[10] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

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

[12] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

[15] Andrej Karpathy. Transfer learning. http://cs231n.github.io/transfer-learning/#tf, 2018.

[16] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[17] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[18] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Wei-Yang Lin, Ya-Han Hu, and Chih-Fong Tsai. Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):421–436, 2012.

[22] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. *arXiv preprint arXiv:1805.12185*, 2018.

[23] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

[24] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *Computer Design (ICCD), 2017 IEEE International Conference on*, pages 45–48. IEEE, 2017.

[25] Jae H Min and Young-Chan Lee. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert systems with applications*, 28(4):603–614, 2005.

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[27] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[29] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[30] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[31] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[32] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[33] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow. Implicit authentication through learning user behavior. In *International Conference on Information Security*, pages 99–113. Springer, 2010.

[34] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[35] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy*, pages 305–316. IEEE, 2010.

[36] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[38] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[39] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *null*, page 511. IEEE, 2001.

[40] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pages 1689–1698, 2015.

[41] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.