Unlearnable Text for Neural Classifiers

Anonymous ACL submission

Abstract

Neural text classification models are known to explore statistical patterns during supervised learning. However, such patterns include spurious patterns and superficial regularity in the training data. In this paper, we exaggerate superficial regularity in the text to prevent unauthorized exploration of personal data.

We propose a gradient-based method to construct text modifications, which can make deep neural networks (DNNs) unlearnable. We then analyze text modifications exposed by the gradient-based method and further propose two simple hypotheses to manually craft unlearnable text. Experiments on four tasks (sentiment classification, topic classification, reading comprehension and gender classification) validate the effectiveness of our method, by which these hypotheses achieve almost untrained performance after training on unlearnable text.

1 Introduction

Huge amounts of data is freely available online, such as movie reviews and articles on the online publishing platforms. Big companies use it to build commercial applications without the agreement of data contributors via Natural Language Processing (NLP) techniques. On the other side, Deep Neural Networks (DNNs) empower many modern NLP applications by utilizing freely available data. It increases the risk of privacy leakage, since DNNs are highly capable to learn statistical features in the training data (Lin et al., 2021) and memorize the information in the training data (Fredrikson et al., 2015). The memorized information could be extracted by the hacker, such as the leakage of name/address from language model (Carlini et al., 2020). This particularly happens when users provide sensitive data to the trusted parties. Normally, users can only rely on the actions of model owners to alleviate the issue by training models with differentially-private techniques (Chaudhuri and Monteleoni, 2009; Shokri and Shmatikov, 2015; McMahan et al., 2018; Abadi et al., 2016).

However, deep learning also leverages undesired patterns during training, including annotation artifacts (Gururangan et al., 2018), syntactic heuristics (McCoy et al., 2019), high-frequency words associated with the target labels (Wallace et al., 2019), algorithmic biases (Zhang et al., 2018) and shallow shortcuts (Branco et al., 2021). Besides, previous works show that spurious correlations (Wallace et al., 2019; Niven and Kao, 2019) cause adversarial examples for a well-trained DNN. For examples, Niven and Kao (2019) shows random accuracy of adversarial examples with spurious statistical cues.

In this paper, we investigate superficial patterns to prevent the unauthorized use of data and radically eliminate the risk of privacy leakage. We generate unlearnable features, which can be easily embedded into text to make DNNs unlearnable. The concept of unlearnable examples is spawn from Huang et al. (2021) for computer vision.

The main contributions of our work include:

- We propose a gradient-based method to explore unlearnable features for three common NLP tasks. Specifically, we adapt the formulation of bi-level optimization Huang et al. (2021) to the discrete textual input by introducing a first-order, gradient-based search algorithm in Section 3. The optimization process would generate an effective one-word modification to make data unlearnable, even for models fine-tuned on powerful pre-trained transformers.

- We find and verify an effective unlearnable pattern for text classification (Section 4): inserting simple synthetic characters (e.g., 'a', 'b', 'c') into the training data in the class-wise manner could be effective for unlearnable training, no matter where they are inserted.

- We find and verify an effective unlearnable pattern for reading comprehension (Section 5) show that: shortcuts can be inserted or substitute the word surrounding the answer spans to prevent DNNs from comprehending the text.
- We also show the effectiveness of the two unlearnable patterns above, even though users can only access and modify a small portion of data for training (Section 6).
- We demonstrate a practical use case to prevent social platforms from user profiling (Section 7).

2 Related Work

This section would demonstrate relevant work for privacy protection, data poison and how gradient-based methods can modify data for different objectives.

Privacy protection. The concerns of data privacy has been raised in many areas. For example, Viejo et al. (2012) had early concern for preventing social media from profiling users while Shan et al. (2020) developed Fawkes to prevent unauthorized face recognition systems from identifying a person. Also, different techniques have been developed for alleviating privacy issues. Shan et al. (2020); Cherepanova et al. (2021) use adversarial attacks to generate unidentified images. Machine unlearning also studies how to protect the privacy of users' data. However, different to unlearnable exmples, it aims to removes training impact of specific samples provided by a user after models have successfully learned from the data (Cao and Yang, 2015).

Data Poisoning. Data poisoning, another malicious attack by modifying training text, aims to manipulate model behaviors at the inference time. Similar to unlearnable examples, poison data is normally generated during training (Muñoz-González et al., 2017; Huang et al., 2020; Kurita et al., 2020; Yang et al., 2021; Wallace et al., 2021), although the attack could be performed for the final models (Gu et al., 2017). Our work distinguishes from the poison attack since unlearnable text only prevents the learning rather than maliciously compromises the model performance or even manipulates model behaviours.

Gradient-based methods. Gradient-based methods have been shown effective to perturb data for different objectives. Gradient-based methods (Ebrahimi et al., 2018; Wallace et al., 2020, 2019) generate adversarial examples by maximizing the cross-entropy loss of clean examples (error-max) (Goodfellow et al., 2015), while poison data are generated to maximize the loss of test data. Both attacks target the malicious behaviour of test data (min-max) (Muñoz-González et al., 2017). In contrast, unlearnable examples minimize the loss of (partial) training data during training (min-min) (Huang et al., 2021). Although the effectiveness is also evaluated on evaluation/test data, unlike adversarial and data poison, they are not included in the unlearnable objective.

There are two specific gradient-based methods for word substitutions: (1) Ebrahimi et al. (2018); Wallace et al. (2019, 2021) searched over potential substitutions via the first-order approximation. (2) Behjati et al. (2019); Cheng et al. (2020) applied projected gradient descend to update continuous representations in the embedding space and perform projected operation for the textual input. In this paper, we use the first-order approximation for unlearnable objective.

3 Generating Unlearnable Text

This section formulates the unlearnable objective, demonstrates text modifications for the objective and devises an algorithm to generate unlearnable text.

3.1 Problem Formulation

Consider the training data $\mathcal{D}$ with a set of $(\mathbf{x}, \mathbf{y})$ and a DNN model f mapping from the input x to the output y. For NLP models, y could be either a label for text classification, an answer span for question answering or a textual sequence for summarization or translation.

To achieve our goal of making data unlearnable, we inject noise into the original training data, which is transformed by an operation Φ. We can then optimize Φ to stop DNNs from learning transferable generalizations, which causes low model performance on the test data. As demonstrated by Huang et al. (2021), we need a bi-level optimization, as shown in Equation 1.

$$\arg\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\arg\min_{\Phi(\mathbf{x})} \mathcal{L}(f(\Phi(\mathbf{x})), y)] \quad (1)$$

Why would the min-min optimization work? The inner minimization would decrease the training loss $\mathcal{L}$ by modifying clean data. Therefore, it would decrease the sensitivity of the outer minimization, since both optimizations have the same objective $\mathcal{L}$. Specifically, when we use gradient descent for model training, the gradients for updating model parameters would be small and contain less transferable information. Consequently, the final models should return low performance on test data.

Unrolling the training steps. The bi-level optimization has been commonly solved by unrolling the training steps to perform an optimization for another set of parameters (Finn et al., 2017; Huang et al., 2020, 2021), which is the original text x in our case. Therefore, the main challenge is how to instantiate unlearnable modifications Φ (see Section 3.2) and update Φ every M training steps (see Section 3.3), which is demonstrated in the rest of this section. The process is summarized in Algorithm 1.

3.2 Instantiating Text Modifications

We have to instantiate modifications Φ to embed the unlearnable patterns into texts. Words are the smallest semantic units in English. We aim to find word substitutions for the inner objective 1. Formally, we optimize unlearnable modification (p, s) where the position p informs where to modify and the substitution s suggests what to modify.

3.3 Optimizing Text Modifications

This section would introduce how to optimize (P, S) for unlearnable dataset $\mathcal{D}_u$ for error minimization (the inner objective).

Challenges compared to unlearnable images. Huang et al. (2021) generates pixel-wise noise which can be directly applied to clean images via pixel-wise addition. Since the noise is continuous and differentiable, it can be directly optimized via gradient descent and simple norm constraints can make noised images imperceptible.

However, due to the discrete nature of text, we cannot optimize text or an applicable noise in the discrete embedding space via gradient descent. Also changing multiple positions would result in meaningless text. To avoid these two-fold challenges, we apply the first-order approximation and perceptibility constraints for text modifications.

First-order approximation. We approximate the change of the training loss for all possible modifications in the first order. This approach has been used for generating text adversaries for adversarial attacks (Wallace et al., 2019; Cheng et al., 2020; Ebrahimi et al., 2018).

Specifically, consider a word in the input $\mathbf{x}_p$ which is indexed by its position p. The loss change of substituting it with the word s can be measured by the inner product of the s embeddings ($\mathbf{e_s}$) and the gradient of loss w.r.t. $\mathbf{x}_p$ ($\nabla_{\mathbf{x}_p}\mathcal{L}$). And our goal is to minimize the loss change for model unlearnability.

$$\arg\min_{s} \quad \mathbf{e_s}^{\mathrm{T}} \nabla_{\mathbf{x}_p} \mathcal{L}(\mathbf{x}, y) \tag{2}$$

The gradients for all the positions of the original example $\nabla_x \mathcal{L}$ can be acquired by one forward and backward pass. We can efficiently measure the loss change for all possible modifications (P, S) with the vocabulary of the possible substitutions S and the gradients $\nabla_x \mathcal{L}$ can be efficiently computed via matrix multiplication.

Perceptibility Constraints. Due to the discrete nature of text, word substitutions easily result in perceptible changes in terms of grammar and semantics. In order to maintain data utility, the following constraints for positions P and substitutions S are applied:

- Constraint 1: We only allow modifying one position, since the optimized positions during iterations are likely to be different. We cannot accumulate modifications at different positions in case of nonsense sentences. We do not keep modifications for the next optimization step and always modify on a clean data for each iteration, which means min-min modifications is a specific error-min modifications for a checkpoint of the model during training.

- Constraint 2: We block positions of answer spans for reading comprehension. we only set constraints for P to exclude answer spans, otherwise positions of answer spans are always selected for modifications either generated via gradient norm or our min-min method.

- Constraint 3: We disable the modifications of proper nouns since words with this part-of-speech contain important information of the text.

Algorithm 1 demonstrates the whole process.

Algorithm 1 Generating Unlearnable Modifications: This process shows how to find a modification for one example at one iteration.

Require: max_swap, neural network f, a clean sample $(\mathbf{X}, y)$ tranining loss $\mathcal{L}$, embedding matrix $\mathcal{E}$

Generate the gradient $\nabla_{\mathbf{x}}\mathcal{L}(f(\mathbf{x}), y)$ ## gradients for all the samples can be generated in only one forward and backward pass if the memory allows.

(reading comprehension) Find valid positions P satisfying Constraints 2

Generate approximation scores $\mathbf{A}$ via Eq. (2) for all the candidate modifications (P, S)

Sort (P, S) in the ascending order of $\mathbf{A}$

for each candidate modification $(p, s) \in (P, S)$ **do**

    if (p, s) satisfies Constraint 2, 3 **then**

        return (p, s)

    end if

end for

3.4 General Experimental Setup

Small surrogate models for transformers. The advent of pre-trained transformers has revolutionized the NLP applications. Therefore, we would maker pre-trained transformer models unlearnable during their common fine-tuning paradigm.

However, although the current downstream NLP models based on pre-trained transformers are often optimized via the pre-training and fine-tuning paradigm, generating effective modifications is very computationally expensive during training. In practice, due to the constraint of the computation resource, optimizating over the large pre-trained language models become more unrealistic. Hence, we perform the gradient-based approach on simple neural nets to explore unlearnable patterns. We assume that statistical features can be common in a architecture-invariant manner.

Implementations. Our codebase benefits from AllenNLP framework and can be flexibly extended to other datasets and all the AllenNLP and Huggingface transformers. [1]

[1]Our code would be available in the future.

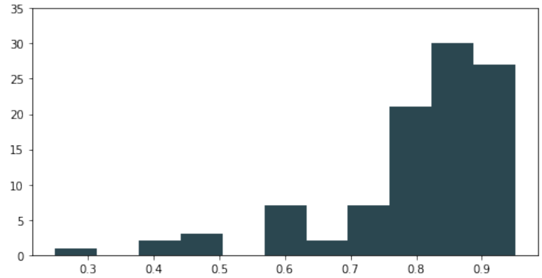

Figure 1: Distribution of relative positions for modifications. The relative position is calculated by dividing the length of the sequence by the index of position.

4 Unlearnable Text Classification

4.1 Experimental Setup

Models. We use CNN (Kim, 2014), LSTM (Hochreiter and Schmidhuber, 1997), self-attention models and BERT (Devlin et al., 2018) .

Tasks and Datasets. We choose two datasets for sentiment anlysis and topic classification respectively, each with its own training, development and test datasets.

- SST-2 for Sentiment Analysis: It contains movie reviews from the Stanford Sentiment Treebank (SST-2) dataset and labels (positive or negative) for binary classification (Socher et al., 2013)

- AGNews: It consists of news articles, which are classified into the following 4 topics : World, Sports, Business and Sci/Tech. It involves 10,800 training samples, 12,000 validation samples and 7,600 test samples.

4.2 Results and Analyses

We generate modifications via Algorithm 1 within 1 epoch of training. M is randomly chosen as 30. The result shows that (1) text tend to be modified at end, as shown in Figure 1. In fact, all the examples are modified at the last two words. (2) substitution words are generated in the class-wise manner (e.g, "and" for positive class, "or" for negative class). The class-wise pattern is automatically explored by our algorithm, which is distinct from class-wise noise for image classifier generated by (Huang et al., 2021).

Class-wise, position agnostic insertion. Since our algorithm exposes class-wise patterns for unlearnability, we consider the operation of insertion

Task	Model	Orig	Min-min	Untrained Accuracy
SST-2	LSTM	0.84	0.5	0.5
	CNN	0.83	0.5	0.5
	BERT	0.91	0.63	0.5
SQuAD-1000	BiDAF	0.25	0.035	0
	RoBERTa	0.73	0.024	0.01

(a) Gradient-based Methods

Task	Model	Orig	Heuristics	Untrained Accuracy
AG-News	LSTM	0.91	0.35	0.25
	CNN	0.92	0.25	0.25
	self-attention	0.90	0.25	0.25
	BERT	0.94	0.28	0.26
SQuAD	BiDAF	0.74	0.06	0.01
	RoBERTa	0.96	0.07	0.01

(b) Heuristics.

We insert one class-wise character ('a', 'b', 'c', 'd' in the middle) for AG-News and a shortcut (a single number '2') in front of answer span for SQuAD.

Table 1: The performance of DNNs trained on unlearanble text. We report accuracy on SST-2 and AG-News, and F1 scores for SQuAD. Modifications on BERT and RoBERTa are generated by surrogate models LSTM and BiDAF respectively. We show random/untrained accuracy to verify the unlearnability.

Begin	Middle	End
0.508	0.5	0.501

Table 2: Positions of trigger insertion. The result is acquired during fine-tuning BERT on SST-2.

	Clean	Begin	Middle	End
CNN	0.91	0.25	0.28	0.26
LSTM	0.92	0.35	0.25	0.23
Self-attention	0.90	0.25	0.25	0.26

Table 3: Effectiveness of different posistions. The results are evaluated on AG_News for topic classification. All three neural nets are trained from scratch.

so that we can avoid the risk of substituting important words, which are not detected by our constraints. We also study whether the end of samples would cause better unlearnability by inserting substitution words at different positions. Tables 3 and 2 reveal that inserting class-wise words at all the positions can be effective. Since they are position-agnostic, adding them in the middle of each text can make them more invisible.

Which triggers are more effective? Class-wise triggers have been studied for adversarial text and are effective to cause adversarial behaviours (Wallace et al., 2019; Behjati et al., 2019). They find the optimal triggers for adversarial attacks via the iterative optimization. We can search optimal triggers in the unlearnable settings. To do this, we insert a words $\mathbf{t}$ at the beginning of each samples and then optimize $\mathbf{t}$. Compared to sample-wise word substitutions, we only optimize substitutions s for fixed positions p.

For comparison, we also randomly select extremely simple triggers ('a' for positive class and 'b' for negative class) for SST-2 during fine-tuning BERT. Both optimized triggers and randomly selected class-wise triggers achieve untrained accuracy (50%).

According to the above analyses, we propose an unlearnable hypothesis for text classifiers

inserting class-wise, small characters into the middle of text.

To evaluate the hypothesis, we present the effectiveness on the AG-news topic classification task with large-scale training data in Table 1.

5 Unlearnable Text for Reading Comprehension

Reading comprehension can be useful for information extraction, which is a common component for search engine and voice assistants. Given a passage of text P and questions Q, models can select the answers $A =$ from P. We assume that users know which part of information they want to protect.

5.1 Experimental Setup

Dataset. We use the Stanford Question Answering Dataset (SQuAD) v1.1 dataset (Rajpurkar et al., 2016), which contains 100K+ question-answer pairs based on 500+ articles. The question-answer pairs are generated by crowd-workers. Dev/test splits from Du et al. (2017) are derived from the original development set, since the SQuAD test set is not publicly available. Since it is time consuming to apply gradient-based optimization on large dataset, we down-sample 1,000 question-answer pairs from the training set for the analyses.

Models. We use Bidirectional Attention Flow (BiDAF) model (Seo et al., 2016) (2.5M parameters) along with the GloVe embeddings and transformer models. BiDAF is the most popular end-to-end neural net before the rising of transformers. It uses two bidirectional LSTMs to represent each context and question and applies attention mechanism to generate question-aware context representations. In contrast, transformer models inherently have special tokens to separate the context and question. Therefore, such representations can be generated with the concatenation of context and the question as the input to the transformer, which is RoBERTa (Liu et al., 2019) in our experiment. We then apply a matrix M^{Hx2} (a linear layer) where H is the hidden size on top and use softmax function to calculate the probability distributions p_{start} and p_{end} for the begin and end of the answer span. During training, the cross-entropy loss is calculated by adding negative log likelihoods of p_{start} and p_{end}.

Evaluation metrics. For all experiments, we measure exact match (EM), span accuracy and F1 score, which is the harmonic mean of recall (the percent of words in the predicted answer span that are in the gold span) and precision (the percent of words in the gold span that are in the predicted span).

5.2 Results

According to all the three metrics, min-min modifications effectively prevent the learning process of the reading comprehension model, as shown in Figure 2. To verify the importance of the bi-level formalization, the error-min modifications are generated by performing Algorithm 1 on the well-trained models. Also, following Huang et al. (2021), error-max modifications, which expose vulnerability for adversarial attack, are also generated for comparison (Ebrahimi et al., 2018; Wallace et al., 2019). Figure 2 shows that error-min and error-max modifications have little effect compared to min-min modifications.

5.3 Why Are The Min-min Modifications Effective?

By analyzing the positions and subsitutions, we find that: (1) the positions P of min-min modifications are always identified within the one-word distance of the answers. (2) The substitutions S tend to be a few unique words. Figure 3 shows that 5 words are used for substitutions of 98% of 1000 samples.

we also find substitution words of error-min and error-max modifications sometimes appear on questions. It is in accord with the finding that well-trained DNNs learn how to locate answers with question tokens, i.e., context matching. (Jia and Liang, 2017). For example, "because to kill american people." can be inserted into context passages as adversarial triggers for all the "why" questions. However, min-min substitutions never include question words. And after the min-min modifications, the model locates answer that surround the substitution words rather than question tokens.

This leads to the hypothesis:

inserting a unique word around the answers can protect text from reading comprehension.

It prevents models from learning generalized rules like context/type matching.

We design several experiments to verify this hypothesis, we (1) fix substitutions as "the", which achieves the very similar effectiveness; (2) randomly select the modification positions P excluding the answer spans, which barely has no effectiveness. Both results support our assumption; (3) We

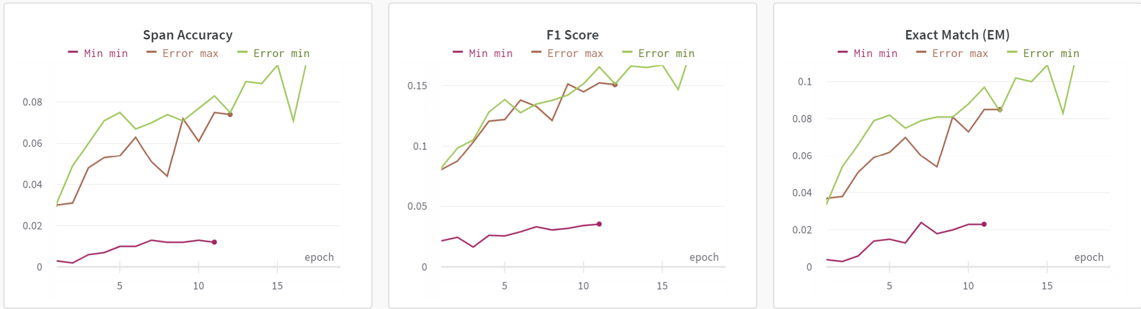

Figure 2: Comparison of min-min, error-min and error-max modifications. All the metrics are computed on test data for BiDAF. Min-min modifications is most effective to make training data unlearnable according to all three metrics. We run all the training for 20 epochs while the training on min-min modifications halts at the 12th epoch due to early stopping.

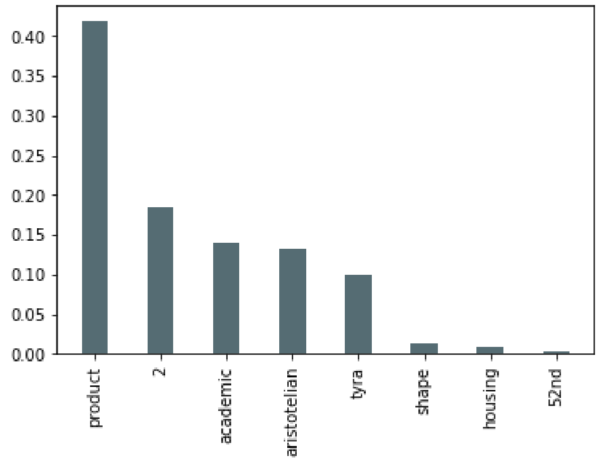

Figure 3: Distribution of substitution words for SQuAD. There are totally eight words generated for min-min modifications by our optimization process. The graph shows the probabilities of all the substitution words and the top-5 words appear in the 98% of 1000 samples.

	Span Accuracy	EM	F1
Min-min	0.012	0.023	0.035
Fixed S	0.006	0.011	0.038
Random P	0.58	0.61	0.72

Table 4: Evaluating the importance of substitutions and position for min-min modifications. The table reports metrics on test data when we fix substitutions to one word or select random positions to modify. It shows that the modifications are effective as long as we put common substitution word(s) surrounding answers.

	95%	90%	80%	0
Modify	0.86	0.88	0.89	0.91
Skip	0.85	0.87	0.89	0.91

Table 5: Test accuracy during fine-tuning of BERT with different unlearnable percentages. We fine-tune the model for 10 epochs to the convergence in all the cases. The results show that it makes no difference whether we modify a fixed percent of training data into unlearnable data or just skip them.

add substitution words (e.g., 'product') to arbitrary positions of test examples and find that the models trained/fine-tuned on unlearnable texts always predict phrases surrounding substitution word as answers, which further verifies the hypothesis.

Finally, we use the hypothesis on the whole SQuAD training set by simply inserting the shortcut word "2" in front of answers. The result in Table 1 shows the effectiveness of unlearnable text.

6 Unlearnable Percentage

Although Table 1 verifies the effectiveness of our hypotheses, the defenders, in practice, can only modify their own data, which means that a small portion of training data can be transformed into unlearnable text. Therefore, we also evaluate the unlearnable effectiveness for training on partial unlearnable samples.

The model cannot learn generalized information from partial unlearnable data. As shown in Table 5, we find that the model accuracy keeps consistent, no matter whether we modify a fixed N% of training data into unlearnable data or just skip them. In other words, the model only learns from another 1-N% clean data, and adding unlearnable text would not increase any generalized information on test data.

Trigger one class to be unlearnable. To further prove the effectiveness of partial unlearnable data for training, we make one class of examples ('World' in the AG News) unlearnable by adding a trigger ('a') and evaluate the test accuracy. The results of low accuracy on the unlearnable class (0.015) and high accuracy on others (0.93) strongly indicate the effectiveness of a small portion of unlearnable text.

Partial unlearnable data for SQuAD. To avoid expensive computation, we design a different experiment to evaluate the effectiveness of partial unlearnable data for SQuAD.

We construct two sets for training $\mathcal{D}_1$, $\mathcal{D}_2$, each of which consists of 1000 samples. We protect $\mathcal{D}_1$ to be an unlearnable set $\mathcal{D}_{u1}$. We then fine-tune the transformer with $\mathcal{D}_{u1} \cup \mathcal{D}_2$.

We compare the model performance on $\mathcal{D}_1$ and $\mathcal{D}_2$ to evaluate whether $\mathcal{D}_{u1}$ can protect $\mathcal{D}_1$. We also report unseen test data $\mathcal{D}_{test}$ as the reference. As shown in Table 6, the model performs much worse on $\mathcal{D}_{u1}$ than $\mathcal{D}_2$. Hence, it can be effective to make partial data unlearnable for SQuAD.

	Span accuracy	EM	F1
$\mathcal{D}_1$	0.59	0.66	0.79
$\mathcal{D}_2$	0.69	0.75	0.86
$\mathcal{D}_{test}$	0.68	0.74	0.83

Table 6: The RoBERTa has poor performance on the protected data $\mathcal{D}_1$, after fine-tuned on $\mathcal{D}_{u1} \cup \mathcal{D}_2$, where $\mathcal{D}_{u1}$ is one version of $\mathcal{D}_1$ with a shortcut.

7 Case Study: Preventing User Profiling

Users' data in social media (e.g., Facebook/twitter) is popularly used to characterize and profile the users (Farnadi et al., 2018), including gender predictions (Suman et al., 2021), political preference. It has been reported that the malicious use can cause unfair intervention for political voting or internet bully.

Text classification via deep learning is one of common tools to determine their demographics for assisting user profiling (Nicolás Sayago et al., 2020). In this section, we show that how easy unlearnable patterns can be inserted into the users' descriptions to prevent gender predictions, which is a salient task of user profiling.

Experimental settings. The dataset [2] comes from the Twitter's user descriptions. It contains 11,194 samples, which are split for training, validation and test by the ratio of 7:2:1. We fine-tune BERT on the training set and report the result on the test set.

[2]https://www.kaggle.com/crowdflower/twitter-user-gender-classification

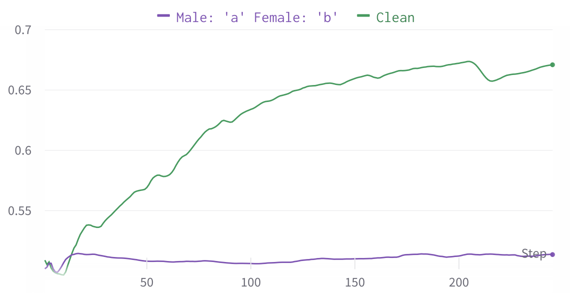

Figure 4: The test accuracy during training on clean data or data applied with class-wise triggers. We insert 'a' for male and 'b' for female in the middle of text. The test accuracy is measured after each update of model parameters, since the loss reaches convergence within one epoch of training.

Effectiveness of unlearnable patterns. According to previous findings, we add simple, class-wise triggers to the middle of their descriptions ('a' for male and 'b' for female). Figure 4 compares the test accuracy during training on clean data or data with the class-wise triggers. Since the loss reaches convergence within one epoch of training, the test accuracy is measured after each update of model parameters. The simple, class-wise triggers successfully make the fine-tuning process of BERT-based classifier fail.

8 Conclusion

By exploring *how to make NLP models unlearnable*, we conclude that presenting superficial features can effectively make data unlearnable, including class-wise word insertion for classification and answer surrounding substitutions for reading comprehension. As for the futher work, we have two directions: First, using more advanced linguistic patterns. Our experiments show that unlearnable word substitutions/insertions can be effective for text classification models. There may be other sensitive, linguistic forms for unlearnable objective: syntactic structure, commonsense, text style. Second, exploring unlearnable text on text generation models. This is also closely related to fact check in tasks like text summarization and machine translation.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349. IEEE.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2020. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.

Kamalika Chaudhuri and Claire Monteleoni. 2009. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3601–3608. AAAI Press.

Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John P. Dickerson, Gavin Taylor, and Tom Goldstein. 2021. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Golnoosh Farnadi, Jie Tang, Martine De Cock, and Marie-Francine Moens. 2018. User profiling through deep multimodal fusion. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 171–179, New York, NY, USA. Association for Computing Machinery.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. 2021. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations (ICLR)*.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. 2020. Metapoison: Practical general-purpose clean-label data poisoning. In *NeurIPS*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. *arXiv preprint arXiv:2105.11136*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38.

Abigail Nicolás Sayago, Sergio Gabriel Sánchez Valencia, and Olga Kolesnikova. 2020. Overview of methods to avoid user profiling. *Computación y Sistemas*, 24(4).

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2020. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1589–1604.

Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 909–910.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. 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. Association for Computational Linguistics.

Chanchal Suman, Anugunj Naman, Sriparna Saha, and Pushpak Bhattacharyya. 2021. A multimodal author profiling system for tweets. *IEEE Transactions on Computational Social Systems*, 8(6):1407–1416.

Alexandre Viejo, David Sánchez, and Jordi Castellà-Roca. 2012. Using profiling techniques to protect the user's privacy in twitter. In *Modeling Decisions for Artificial Intelligence*, pages 161–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 2153–2162.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 5531–5546. Association for Computational Linguistics.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. 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.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021. 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. Association for Computational Linguistics.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 335–340, New York, NY, USA. Association for Computing Machinery.