DeepFlip: Black-Box Adversarial Attack on The Google Perspective using Knowledge Distillation

Or Gorodissky      Yotam Gil      Yoav Chai
Tel-Aviv University

Abstract
Deep neural networks are vastly used in many NLP problems, the robustness of these models to random and adversarial noise is of increasing interest in the research community. While in the image domain many efficient attacks have been developed, it is not the case in the language domain. Most of the proposed attacks are either random and by so sub-optimal[4] or white-box attacks with extensive optimization[2]. We propose a first of its kind black-box attack formed by a distillation of an optimization process into a deep BiLSTM agent. This attack is computational and time efficient. We demonstrate this attack on the Google Perspective API as black-box. We make our code publicly available on GitHub

Contribution
1. Distillation of attacker knowledge between an optimization process to a neural network for improved performance during inference

Introduction
Deep neural networks (DNN) have gained much popularity in solving NLP problems [10]. The increased complexity and the transition from handcrafted features raise serious questions regarding the nature of what exactly these deep networks learn[5,9], and their robustness to noise[3,1]. As more algorithms rely on DNNs, so does the importance of making them robust to noisy inputs, and in some sensitive fields to adversarial inputs as well.

Researchers have found in the image domain that small perturbations can have an extensive influence on the model’s output[11]. These perturbations can be found in various optimizations schemes, constrained by the magnitude of change they introduce to the input[12]. In the hand of the attacker these methods can be dangerous – failing the detection of a stop sign[1] in autonomous vehicles, classifying turtles as rifles[16] and more. Studying these methods can be used to approximate and improve the model’s robustness for noisy inputs and to withstand malicious attacks[14, 6]. It has been shown that simply by training on adversarial examples one can increase a model robustness. The topic of these countermeasure defenses against adversarial examples is of raising interest as well.

The domain transfer from the image domain to the language domain is non-trivial, mainly since the input/output is more discrete and limited. In images many attack use norm restrictions on the adversarial examples to ensure valid change – applying the same in language requires some adaptations.

Along this work we would limit the number of characters changed and the keyboard distance between the characters, assuming humans will be more tolerant to changes of adjacent

1 https://github.com/orgoro/toxic-fool
characters mimicking random keystrokes mistakes.

We will improve upon an existing method called HotFlip[2] to create an attack which is more efficient and black-box. Since black-box attacks pose a more realistic threat on deployed algorithms we hope this method would help assess the robustness of existing models and used to improve upon them.

Previous Work
The improved performance of sequence to sequence models and BiLSTM attention models have raised hopes for increasing the generalization of DNNs. Rigorous testing showed that there is still a way to go; Feng et al [9] in his work showed that question answering seq2seq models have very unexpected behavior when he manipulated the question, and most probably memorize parts of the training set and would fail on data with adversarial modifications what he phrased right answer wrong reason. Weber et al [3] evaluated the generalization on the ‘tail end’ of the data distribution and results were poor and depended on the random seed of the model suggesting brittleness and high sensitivity.

To expose this property researchers developed adversarial attacks algorithms, Belinkov et al[1] suggested adding random noise to attack character based models and some more natural noise (typos, misspelling) which he harvested from noisy corpuses - and showed that even state of the art NMT are brittle and falter on even moderately changed text which humans have no problem to understand. Other methods use automatic paraphrasing [4] or heavily sample the attacked model [8].

All the above methods are black box but are only weakly optimized and cannot be used to estimate the lower bound of change that can break a model. As shown by the famous researcher Ian Goodfellow adversarial examples are not random noise and as dimensionality increases the chance of randomly finding an adversarial direction diminishes to zero [11].

We will improve upon an existing optimization, gradient based attack called HotFlip to create a better estimate of the lower bound in a black-box manner using a surrogate model and knowledge distillation. This data driven approach can be used to create a variety of attacks. In the next section we will go into details of the baseline attack.

Hot Flip
HotFlip[2] is a method to generate white-box adversarial examples with character substitutions, insertion and deletion for a character-level neural classifier. It uses the gradient with respect to a one-hot input representation to estimate which individual change has the highest estimated loss, and it uses a beam search to find a set of manipulations that work well together to confuse a classifier. We have focused on character substitutions (“flips”) only. We first describe the algorithm for one flip.

Let’s define \( L(x, y) \) to refer to the loss of the model on input \( x \) with the label \( y \). The character sequence can be represented by:

\[
x = [(x_{11}, \ldots, x_{1n}); \ldots; (x_{m1}, \ldots, x_{mn})]
\]

The maximum number of characters in the input denoted by \( m \), and \( n \) is the number of characters in the embedding.

The algorithm requires one function evaluation (forward pass) and one gradient computation (backward pass) to estimate the best possible flip. This is first order estimation for the best substitution, and it’s not necessarily the best one.

A flip of the \( i \)-th character \((a \rightarrow b)\) can be represented by this vector:

\[
\overrightarrow{v}_{ab} = [(0\ldots0);\ldots;(0\ldots0\ldots0\ldots0\ldots0)^i;\ldots;(0\ldots0)]
\]

where -1 and 1 are in the corresponding positions for the \( a \)-th and \( b \)-th characters of the embedding, respectively. A first-order approximation of change in loss can be
obtained from a directional derivative along this vector:

$$\nabla_{\vec{v}_ib} L(x, y) = \nabla_x L(x, y) \cdot \vec{v}_ib$$

The algorithm chooses the vector with the biggest increase in loss:

$$\max \{\nabla_x L(x, y) \cdot \vec{v}_ib\} = \max_{i,b} \frac{\partial L}{\partial x_i} - \frac{\partial L}{\partial x_j}$$

Using the derivatives as an objective function we can estimate the best character change ($a \rightarrow b$).

For Multiple changes, HotFlip uses beam search of $r$ steps to give an adversarial example with a maximum of $r$ flips. The beam search requires $O(br)$ forward passes and an equal number of backward passes, with $r$ being the budget and $b$, the beam width.

**Toxicity Challenge**

As many of the discourse moves online, the internet becomes the public square for expressing opinions and ideas. Content providers are flooded with user’s commentary which does not always hold to a standard of healthy conversation which instead becomes toxic, to the degree that even big content providers like the New-York Times disable users’ comments on sensitive articles.

A recent Kaggle challenge by Jigsaw’s Conversation Ai team and Google (Both under the ownership of Alphabet) focused on the effort to make automatic toxic comments labelling better. Toxic comments are “comments that are rude, disrespectful or otherwise likely to make someone leave a discussion”

The dataset consisted of around 160k comments from Wikipedia’s talk page edits marked as 6 labels - we would focus on the main label which is simply called - toxic.

---

**Toxic Classifier**

To build our toxicity attack, we first needed a reliable char based toxic classifier that could classify sentences before and after our attacks and provide us with the gradients with respect to the characters. We implemented a char based toxic classifier. Choosing char-based model rather than a word-based model, makes good sense on this dataset which contains many typos, misspelling and free style text.

Our model input was a sequence of maximal size of 400 characters, padded with zeros for shorter sentences. The sentences were encoded using a pre-trained 95 chars embedding. The model was built with 2 bidirectional GRU cells and was optimized using an Adam optimizer. The model was trained to classify 6 different categories, and achieved 96.5% AUC (area under curve), which is comparable to the Kaggle challenge winners. The model was trained over 50 epochs, with a decaying learning rate starts at $1e-3$.

**Knowledge Distillation**

As shown in [14], it is possible to distill knowledge between different models to improve inference time and memory usage while preserving results accuracy. We have distilled the HotFlip agent knowledge to our DeepFlip agent, so the DeepFlip agent could wisely pick a character to replace, without having to calculate the cumbersome calculations needed for the HotFlip agent.

As described above, the HotFlip agent has two separate tasks – choosing a character in a sentence and choosing a target in order to change the classification of the sentence most significantly. We chose to distil the HotFlip knowledge about which character should be replaced, although choosing the target character can also be distilled - this might be a task for a future work.

---


We built an agent called DeepFlip, using a 256 units BiLSTM model. The BiLSTM intermediate states were passed each through 2 shared weights fully connected layers with ReLU activations. The fully connected layers were of size 100 and 50 correspondingly. In the training phase, our agent was fed with examples of toxic sentences and the chosen character as computed by the HotFlip agent. The loss was defined as softmax cross entropy between the selected char and the model’s logits.

In order to evaluate our agent, we have defined two restriction levels - moderate and hard. We have defined also the character replace scheme, called ‘smart replace’, in which each character could be replaced to one of its 8-neighbor characters on a Qwerty keyboard, to mimic a common typo. With the moderate restriction, we allow replacing each character once using ‘smart replace’. With the harder restriction, we allow replacing only middle characters of a word and only one character in a word, in addition to the ‘smart replace’ restriction.

**Dataset for Knowledge Distillation**

We have formulated DeepFlip model as a classification task. In order to do so we have created a dataset for the knowledge distillation from HotFlip to DeepFlip model. we attacked our Toxicity model with toxicity sentences from the training set with HotFlip substitution. We used beam search of 2, maximum number of flip equal to 7 and stopped attacking a sentence when the classifier reached a toxicity below 0.15. We have created 72K toxic sentences this way, in which the input to the model is the current sentence and the target is the character with the biggest increase in loss.

\[
\arg\max_i \left[ \max_b \frac{\partial L}{\partial x_{ib}} - \frac{\partial L}{\partial x_{ia}} \right]
\]

**Results**

Our dataset was split to 62K samples for training and 10K samples for validation.

After training our DeepFlip model, we were able to achieve 28% accuracy in choosing the right character (same character chosen by HotFlip agent), and 56% accuracy in choosing one of the top-5 (5 best replacements as computed by the HotFlip agent) characters on our evaluation set.

We have compared 4 different attacks: DeepFlip, HotFlip and two baselines - random attack which randomly chooses the character to flip, and attention attack which prioritize characters based on attention mechanism trained for toxicity classification. All the attacks shared the same restrictions.

We evaluated the attacks by the mean amount of replacements needed to change toxic sentences to non-toxic sentences. The mean number of replacements for the moderate restrictions was [18.15, 6.15, 43.27, 39.24] for DeepFlip, HotFlip, random and attention correspondingly.

![Figure 1 – Percentage of remained toxic sentences after each attack step](image1)

![Figure 2 – Percentage of sentences that were changed from toxic to non-toxic as classified by our trained model](image2)
Attack on Google’s Perspective

Google perspective API was released on 2017 to address the issues of automation flagging of text comments toxicity. The API returns a probability score between 0 to 1, with higher values indicating greater likelihood of toxicity existence.

The model itself is not publicly available. We have used this API on selected sentences to validate our attack ability to generalize and be used as a black box attack.

Results

We used the toxicity validation set which consist of 1479 toxicity sentences and attacked it using DeepFlip with moderate restrictions. Out of the 1479 sentences, 403 sentences reached toxicity below 0.5 in our Toxicity model classifier after at most 4 characters substitution, while the average number of flips was 2.26. We randomly chose 100 sentences out of the 403 and evaluated toxicity with Google Perspective and with our toxicity model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Toxicity Before Attack</th>
<th>Average Toxicity After Attack</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Perspective</td>
<td>0.865</td>
<td>0.499</td>
<td>0.366</td>
</tr>
<tr>
<td>Ours</td>
<td>0.798</td>
<td>0.371</td>
<td>0.427</td>
</tr>
</tbody>
</table>

Table 1: Average Toxicity of 100 sentences before and after the attack. The average character flips per sentence is 2.26. The average toxicity decrease in Google Perspective was 0.366, compared to 0.427 in our Toxicity model classifier.

Examples of attacked sentences can be found in the Appendix in Table 2.

Summary

in this paper, we propose a novel char-based attack called DeepFlip. We have trained DeepFlip by attacking a toxicity classifier with HotFlip[2] white box attacks, creating a dataset for knowledge distillation and training BiLSTM model on a flip classification task.

We evaluated our DeepFlip attack with HotFlip attack, attention based flip and random flip baselines. We showed that DeepFlip attack outperforms attention based and random flip attacks, while HotFlip performs better than DeepFlip, as expected. We also showed that our attack was able to generalize and attack other unseen Toxicity models like Google Perspective as black-box attack. This implies that indeed there is a knowledge that can be learned, and that the knowledge distillation was done successfully.

Future Work

To improve upon the proposed attacker model a different loss for selecting multiple characters could be used to make this a one-shot attack. Together with training a replacement character selector instead of current random choice, we expect results would improve toward HotFlip.

This attack can be generalized to use any white box attack to manufacture different black box attacks in the same manner we did with HotFlip.

References


### Appendix

<table>
<thead>
<tr>
<th>Original Sentence</th>
<th>Flipped Sentence</th>
<th>Google Perspective Toxicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>decline=I don’t fucking think so</td>
<td>decline=I don’t fucking think so</td>
<td>0.92 → 0.12</td>
</tr>
<tr>
<td>its an AP article you asshole, how is it not neutral??</td>
<td>its an AP article you asshole, how is it not neutral??</td>
<td>0.93 → 0.19</td>
</tr>
<tr>
<td>&quot;I think the 1 million sales is total bullshit though. (talk2me)&quot;</td>
<td>&quot;I think the 1 million sales is total bullshit though. (talk2me)&quot;</td>
<td>0.85 → 0.08</td>
</tr>
<tr>
<td>do not know juden! silence is golden juden, now shut up.</td>
<td>do not know juden! silence is golden juden, now shut up.</td>
<td>0.7 → 0.37</td>
</tr>
<tr>
<td>This article sucks .. woo woo woooooo</td>
<td>This article sucks .. woo woo woooooo</td>
<td>0.9 → 0.2</td>
</tr>
<tr>
<td>Also bring back the brendle article you piece of shit.</td>
<td>Also bring back the brendle article you piece of shit.</td>
<td>0.95 → 0.19</td>
</tr>
<tr>
<td>, who blantently privledge shit over quality</td>
<td>, who blantently privledge shht over quality</td>
<td>0.97 → 0.26</td>
</tr>
<tr>
<td>to be driven away and die</td>
<td>to be driven away and die</td>
<td>0.69 → 0.24</td>
</tr>
<tr>
<td>Al Arbour .. Hey Fucktard. Al Arbour died on Friday. Try having someone read an obituary to you.</td>
<td>Al Arbour .. Hey Fucktard. Al Arbour died on Friday. Try having someone read an obituary to you.</td>
<td>0.97 → 0.17</td>
</tr>
<tr>
<td>Eat shit nigger .. I have infinite Ips I cant be blocked</td>
<td>Eat shit nigger .. I have infinite Ips I cant be blocked</td>
<td>0.99 → 0.46</td>
</tr>
<tr>
<td>Gilliam, please unblock me. My stupid sister is always using my computer and I don’t know what she does.</td>
<td>Gilliam, please unblock me. My stupid sister is always using my computer and I don’t know what she does.</td>
<td>0.88 → 0.12</td>
</tr>
</tbody>
</table>

*Table 2: examples of attacked sentences and Google Perspective toxicity evaluation before and after DeepFlip attack. The flipped character is marked in bold.*