Adversarial Semantic Collisions

Congzheng Song
Cornell University
cs2296@cornell.edu

Alexander M. Rush
Cornell Tech
arush@cornell.edu

Vitaly Shmatikov
Cornell Tech
shmat@cs.cornell.edu

Abstract

We study *semantic collisions*: texts that are semantically unrelated but judged as similar by NLP models. We develop gradient-based approaches for generating semantic collisions and demonstrate that state-of-the-art models for many tasks which rely on analyzing the meaning and similarity of texts—including paraphrase identification, document retrieval, response suggestion, and extractive summarization—are vulnerable to semantic collisions. For example, given a target query, inserting a crafted collision into an irrelevant document can shift its retrieval rank from 1000 to top 3. We show how to generate semantic collisions that evade perplexity-based filtering and discuss other potential mitigations. Our code is available at https://github.com/csong27/collision-bert.

1 Introduction

Deep neural networks are vulnerable to adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015), i.e., imperceptibly perturbed inputs that cause models to make wrong predictions. Adversarial examples based on inserting or modifying characters and words have been demonstrated for text classification (Liang et al., 2018; Ebrahimi et al., 2018; Pal and Tople, 2020), question answering (Jia and Liang, 2017; Wallace et al., 2019), and machine translation (Belinkov and Bisk, 2018; Wallace et al., 2020). These attacks aim to minimally perturb the input so as it to preserve its semantics while changing the output of the model.

In this work, we introduce and study a different class of vulnerabilities in NLP models for analyzing the meaning and similarity of texts. Given an input (query), we demonstrate how to generate a **semantic collision**: an unrelated text that is judged semantically equivalent by the target model. Semantic collisions are the "inverse" of adversarial examples. Whereas adversarial examples are similar inputs that produce dissimilar model outputs, semantic collisions are dissimilar inputs that produce similar model outputs.

We develop gradient-based approaches for generating collisions given white-box access to a model and deploy them against several NLP tasks. For paraphrase identification, the adversary crafts collisions that are judged as a valid paraphrase of the input query; downstream applications such as removing duplicates or merging similar content will thus erroneously merge the adversary's inputs with the victim's inputs. For document retrieval, the adversary inserts collisions into one of the documents that cause it to be ranked very high even though it is irrelevant to the query. For response suggestion, the adversary's irrelevant text is ranked as the top suggestion and can also carry spam or advertising. For extractive summarization, the adversary inserts a collision into the input text, causing it to be picked as the most relevant content.

Our first technique generates collisions aggressively, without regard to potential defenses. We then develop two techniques, "regularized aggressive" and "natural," that constrain generated collisions using a language model so as to evade perplexity-based filtering. We evaluate all techniques against state-of-the-art models and benchmark datasets on all four tasks. For paraphrase identification on Quora question pairs, our collisions are (mis)identified as paraphrases of inputs with 97% confidence on average. For document retrieval, our collisions shift the median rank of irrelevant documents from 1000 to around 10. For response suggestion in dialogue (sentence retrieval), our collisions are ranked as the top response 99% and 86% of the time with the aggressive and natural techniques, respectively. For extractive summarization, our collisions are chosen by the model as the summary 100% of the time. We conclude by discussing potential defenses against these attacks.

Task	Target inputs and collisions	Model output
Paraphrase Identification	***Input*** $(\boldsymbol{x})$: Does cannabis oil cure cancer? Or are the sellers hoaxing? ***Aggressive*** $(\boldsymbol{c})$: **_Pay 0ff your mortgage** der Seller chem Wad marijuana scarcity prince ***Regularized aggressive*** $(\boldsymbol{c})$: caches users remedies paved Sell Medical hey untold Caval OR and of of of of of of of of of of of of a a a of a ***Natural*** $(\boldsymbol{c})$: he might actually work when those in	$\geq 99\%$ confidence of paraphrase
Document Retrieval	***Query*** $(\boldsymbol{x})$: Health and Computer Terminals ***Aggressive*** $(\boldsymbol{c})$: chesapeake oval mayo knuckles crowded double transmitter gig after nixon, tipped incumbent physician kai joshi astonished northwestern documents | obliged dumont determines philadelphia consultative oracle keyboards dominates tel node ***Regularized aggressive*** $(\boldsymbol{c})$: and acc near floors : panicked ; its employment became impossible, the – of cn magazine usa, in which ” ”’panic over unexpected noise, noise of and a of the of the of the of a of of the of the of of of of of the of of of the of. ***Natural*** $(\boldsymbol{c})$: the ansb and other buildings to carry people : three at the mall, an infirmary, an auditorium, and a library, as well as a clinic, pharmacy, and restaurant	Irrelevant articles' ranks ≤ 3
Response Suggestion	***Context*** $(\boldsymbol{x})$: ...i went to school to be a vet , but i didn't like it. ***Aggressive*** $(\boldsymbol{c})$: **buy v1agra in canadian pharmacy** to breath as four ranger color ***Regularized aggressive*** $(\boldsymbol{c})$: kill veterans and oxygen snarled clearly you were a a to to and a a to to to to to to to to to to ***Natural*** $(\boldsymbol{c})$: then not have been an animal, or a human or a soldier but should	$\boldsymbol{c}$'s rank = 1
Extractive Summarization	***Truth***: on average, britons manage just six and a half hours ' sleep a night , which is far less than the recommended eight hours. ***Aggressive*** $(\boldsymbol{c})$: iec cu franks believe carbon chat fix pay carbon targets co_2 8 iec cu mb ***Regularized aggressive*** $(\boldsymbol{c})$: the second mercury project carbon b mercury is a will produce 38 million 202 carbon a a to to to to to to to to to to to to ***Natural*** $(\boldsymbol{c})$: 1 million men died during world war ii; over 40 percent were women	$\boldsymbol{c}$'s rank = 1

Table 1: Four tasks in our study. Given an input $\boldsymbol{x}$ and white-box access to a victim model, the adversary produces a collision $\boldsymbol{c}$ resulting in a deceptive output. Collisions can be nonsensical or natural-looking and also carry spam messages (shown in red).

2 Related Work

Adversarial examples in NLP. Most of the previously studied adversarial attacks in NLP aim to minimally modify or perturb inputs while changing the model's output. Hosseini et al. (2017) showed that perturbations, such as inserting dots or spaces between characters, can deceive a toxic comment classifier. HotFlip used gradients to find such perturbations given white-box access to the target model (Ebrahimi et al., 2018). Wallace et al. (2019) extended HotFlip by inserting a short crafted "trigger" text to any input as perturbation; the trigger words are often highly associated with the target class label. Other approaches are based on rules, heuristics or generative models (Mahler et al., 2017; Ribeiro et al., 2018; Iyyer et al., 2018; Zhao et al., 2018). As explained in Section 1, our goal is the inverse of adversarial examples: we aim to generate inputs with drastically different semantics that are perceived as similar by the model.

Several works studied attacks that change the semantics of inputs. Jia and Liang (2017) showed that inserting a heuristically crafted sentence into a paragraph can trick a question answering (QA) system into picking the answer from the inserted sentence. Aggressively perturbed texts based on HotFlip are nonsensical and can be translated into meaningful and malicious outputs by black-box translation systems (Wallace et al., 2020). Our semantic collisions extend the idea of changing input semantics to a different class of NLP models; we design new gradient-based approaches that are not perturbation-based and are more effective than HotFlip attacks; and, in addition to nonsensical adversarial texts, we show how to generate "natural" collisions that evade perplexity-based defenses.

Feature collisions in computer vision. Feature collisions have been studied in image analysis models. Jacobsen et al. (2019a) showed that images from different classes can end up with identical representations due to excessive invariance of deep models. An adversary can modify the input to change its class while leaving the model's prediction unaffected (Jacobsen et al., 2019b). The intrinsic property of rectifier activation function can cause images with different labels to have the same feature vectors (Li et al., 2019).

3 Threat Model

We describe the targets of our attack, the threat model, and the adversary's objectives.

Semantic similarity. Evaluating semantic sim-

ilarity of a pair of texts is at the core of many NLP applications. *Paraphrase identification* decides whether sentences are paraphrases of each other and can be used to merge similar content and remove duplicates. *Document retrieval* computes semantic similarity scores between the user's query and each of the candidate documents and uses these scores to rank the documents. *Response suggestion*, aka Smart Reply (Kannan et al., 2016) or sentence retrieval, selects a response from a pool of candidates based on their similarity scores to the user's input in dialogue. *Extractive summarization* ranks sentences in a document based on their semantic similarity to the document's content and outputs the top-ranked sentences.

For each of these tasks, let f denote the model and $\boldsymbol{x}_a, \boldsymbol{x}_b$ a pair of text inputs. There are two common modeling approaches for these applications. In the first approach, the model takes the concatenation $\oplus$ of $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ as input and directly produces a similarity score $f(\boldsymbol{x}_a \oplus \boldsymbol{x}_b)$. In the second approach, the model computes a sentence-level embedding $f(\boldsymbol{x}) \in \mathbb{R}^h$, i.e., a dense vector representation of input $\boldsymbol{x}$. The similarity score is then computed as $s(f(\boldsymbol{x}_a), f(\boldsymbol{x}_b))$, where s is a vector similarity metric such as cosine similarity. Models based on either approach are trained with similar losses, such as the binary classification loss where each pair of inputs is labeled as 1 if semantically related, 0 otherwise. For generality, let $\mathcal{S}(\cdot, \cdot)$ be a similarity function that captures semantic relevance under either approach. We also assume that f can take $\boldsymbol{x}$ in the form of a sequence of discrete words (denoted as w) or word embedding vectors (denoted as $\boldsymbol{e}$), depending on the scenario.

Assumptions. We assume that the adversary has full knowledge of the target model, including its architecture and parameters. It may be possible to transfer white-box attacks to the black-box scenario using model extraction (Krishna et al., 2020; Wallace et al., 2020); we leave this to future work. The adversary controls some inputs that will be used by the target model, e.g., he can insert or modify candidate documents for a retrieval system.

Adversary's objectives. Given a target model f and target sentence $\boldsymbol{x}$, the adversary wants to generate a collision $\boldsymbol{x}_b = \boldsymbol{c}$ such that f perceives $\boldsymbol{x}$ and $\boldsymbol{c}$ as semantically similar or relevant. Adversarial uses of this attack depend on the application. If an application is using paraphrase identification to merge similar contents, e.g., in Quora (Scharff, 2015), the adversary can use collisions to deliver spam or advertising to users. In a retrieval system, the adversary can use collisions to boost the rank of irrelevant candidates for certain queries. For extractive summarization, the adversary can cause collisions to be returned as the summary of the target document.

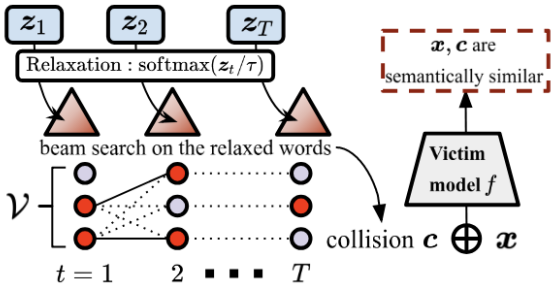

Figure 1: Overview of generating semantic collision $\boldsymbol{c}$ for a query input $\boldsymbol{x}$. The continuous variables $\boldsymbol{z}_t$ relax the words in $\boldsymbol{c}$ and are optimized with gradients. We search in the simplex produced by $\boldsymbol{z}_t$ for the actual colliding words in $\boldsymbol{c}$.

4 Adversarial Semantic Collisions

Given an input (query) sentence $\boldsymbol{x}$, we aim to generate a collision $\boldsymbol{c}$ for the victim model with the white-box similarity function $\mathcal{S}$. This can be formulated as an optimization problem: $\arg\max_{\boldsymbol{c} \in \mathcal{X}} \mathcal{S}(\boldsymbol{x}, \boldsymbol{c})$ such that $\boldsymbol{x}$ and $\boldsymbol{c}$ are semantically unrelated. A brute-force enumeration of $\mathcal{X}$ is computationally infeasible. Instead, we design gradient-based approaches outlined in Algorithm 1. We consider two variants: (a) aggressively generating unconstrained, nonsensical collisions, and (b) constrained collisions, i.e., sequences of tokens that appear fluent under a language model and cannot be automatically filtered out based on their perplexity.

We assume that models can accept as inputs both hard one-hot words and soft words,[1] where a soft word is a probability vector $\check{\boldsymbol{w}} \in \Delta^{|\mathcal{V}|-1}$ for vocabulary $\mathcal{V}$.

4.1 Aggressive Collisions

We use gradient-based search to generate a fixed-length collision given a target input. The search is done in two steps: 1) we find a continuous representation of a collision using gradient optimization with relaxation, and 2) we apply beam search to produce a hard collision. We repeat these two steps iteratively until the similarity score $\mathcal{S}$ converges.

[1]For a soft-word input, models will compute the word vector as the weighted average of word embeddings by the probability vector.

Algorithm 1 Generating adversarial semantic collisions

Input: input text $\boldsymbol{x}$, similarity function $\mathcal{S}$, embeddings $\boldsymbol{E}$, language model g, vocabulary $\mathcal{V}$, length T
Hyperparams: beam size B, top-k size K, iterations N, step size η, temperature τ, score coefficient β, label smoothing ϵ

procedure MAIN
  return collision $\boldsymbol{c}$ =AGGRESSIVE() or NATURAL()
procedure AGGRESSIVE
  $\boldsymbol{Z} \leftarrow [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T], \boldsymbol{z}_t \leftarrow \mathbf{0} \in \mathbb{R}^{|\mathcal{V}|}$
  while similarity score not converged **do**
    for iteration 1 to N **do**
      $\check{\boldsymbol{c}} \leftarrow [\check{\boldsymbol{c}}_1, \ldots, \check{\boldsymbol{c}}_T], \check{\boldsymbol{c}}_t \leftarrow \text{softmax}(\boldsymbol{z}_t/\tau)$
      $\boldsymbol{Z} \leftarrow \boldsymbol{Z} + \eta \cdot \nabla_{\boldsymbol{Z}}(1-\beta) \cdot \mathcal{S}(\boldsymbol{x}, \check{\boldsymbol{c}}) + \beta \cdot \Omega(\boldsymbol{Z})$
    $\mathcal{B} \leftarrow B$ replicates of empty token
    for $t = 1$ to T **do**
      $\boldsymbol{F}_t \leftarrow \mathbf{0} \in \mathbb{R}^{B \times K}$, beam score matrix
      for $\boldsymbol{c}_{1:t-1} \in \mathcal{B}, w \in \text{top-k}(\boldsymbol{z}_t, K)$ **do**
        $\boldsymbol{F}_t[\boldsymbol{c}_{1:t-1}, w] \leftarrow \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w \oplus \check{\boldsymbol{c}}_{t+1:T})$
      $\mathcal{B} \leftarrow \{\boldsymbol{c}_{1:t-1} \oplus w | (\boldsymbol{c}_{1:t-1}, w) \in \text{top-k}(\boldsymbol{F}_t, B)\}$
    $\text{LS}(\boldsymbol{c}_t) \leftarrow \text{label-smooth}(\boldsymbol{c}_t, \epsilon)$ for $\boldsymbol{c}_t \in \arg\max \mathcal{B}$
    $\boldsymbol{z}_t \leftarrow \log \text{LS}(\boldsymbol{c}_t)$ for $\boldsymbol{z}_t$ in $\boldsymbol{Z}$
  return $\boldsymbol{c} = \arg\max \mathcal{B}$

procedure NATURAL
  $\mathcal{B} \leftarrow B$ replicates of start token
  for $t = 1$ to T **do**
    $\boldsymbol{F}_t \leftarrow \mathbf{0} \in \mathbb{R}^{B \times K}$, beam score matrix
    for each beam $\boldsymbol{c}_{1:t-1} \in \mathcal{B}$ **do**
      $\boldsymbol{\ell}_t \leftarrow g(\boldsymbol{c}_{1:t-1})$, next token logits from LM
      $\boldsymbol{z}_t \leftarrow \text{UPDATELOGITS}(\boldsymbol{\ell}_t, \boldsymbol{c}_{1:t-1})$
      for $w \in \text{top-k}(\boldsymbol{z}_t, K)$ **do**
        $\boldsymbol{F}_t[\boldsymbol{c}_{1:t-1}, w] \leftarrow$ joint score from Eq 5
      $\mathcal{B} \leftarrow \{\boldsymbol{c}_{1:t-1} \oplus w | (\boldsymbol{c}_{1:t-1}, w) \in \text{top-k}(\boldsymbol{F}_t, B)\}$
  return $\boldsymbol{c} = \arg\max \mathcal{B}$
procedure UPDATELOGITS($\boldsymbol{\ell}, \boldsymbol{c}_{1:t-1}$)
  $\boldsymbol{\delta} \leftarrow \mathbf{0} \in \mathbb{R}^{|\mathcal{V}|}$
  for iteration 1 to N **do**
    $\check{\boldsymbol{c}}_t \leftarrow \text{softmax}((\boldsymbol{\ell} + \boldsymbol{\delta})/\tau)$
    $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} + \eta \cdot \nabla_{\boldsymbol{\delta}} \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus \check{\boldsymbol{c}}_t)$
  return $\boldsymbol{z} = \boldsymbol{\ell} + \boldsymbol{\delta}$

Optimizing for soft collision. We first relax the optimization to a continuous representation with temperature annealing. Given the model's vocabulary $\mathcal{V}$ and a fixed length T, we model word selection at each position t as a continuous logit vector $\boldsymbol{z}_t \in \mathbb{R}^{|\mathcal{V}|}$. To convert each $\boldsymbol{z}_t$ to an input word, we model a softly selected word at t as:

$$\check{\boldsymbol{c}}_t = \text{softmax}(\boldsymbol{z}_t/\tau) \tag{1}$$

where τ is a temperature scalar. Intuitively, softmax on $\boldsymbol{z}_t$ gives the probability of each word in $\mathcal{V}$. The temperature controls the sharpness of word selection probability; when $\tau \to 0$, the soft word $\check{\boldsymbol{c}}_t$ is the same as the hard word $\arg\max \boldsymbol{z}_t$.

We optimize for the continuous values $\boldsymbol{z}$. At each step, the soft word collisions $\check{\boldsymbol{c}} = [\check{\boldsymbol{c}}_1, \ldots, \check{\boldsymbol{c}}_T]$ are forwarded to f to calculate $\mathcal{S}(\boldsymbol{x}, \check{\boldsymbol{c}})$. Since all operations are continuous, the error can be back-propagated all the way to each $\boldsymbol{z}_t$ to calculate its gradients. We can thus apply gradient ascent to improve the objective.

Searching for hard collision. After the relaxed optimization, we apply a projection step to find a hard collision using discrete search.[2] Specifically, we apply left-to-right beam search on each $\boldsymbol{z}_t$. At every search step t, we first get the top K words w based on $\boldsymbol{z}_t$ and rank them by the target similarity $\mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w \oplus \check{\boldsymbol{c}}_{t+1:T})$, where $\check{\boldsymbol{c}}_{t+1:T}$ is the partial soft collision starting at $t+1$. This procedure allows us to find a hard-word replacement for the soft word at each position t based on the previously found hard words and relaxed estimates of future words.

Repeating optimization with hard collision. If the similarity score still has room for improvement after the beam search, we use the current $\boldsymbol{c}$ to initialize the soft solution $\boldsymbol{z}_t$ for the next iteration of optimization by transferring the hard solution back to continuous space.

In order to initialize the continuous relaxation from a hard sentence, we apply label smoothing (LS) to its one-hot representation. For each word $\boldsymbol{c}_t$ in the current $\boldsymbol{c}$, we soften its one-hot vector to be inside $\Delta^{|\mathcal{V}|-1}$ with

$$\text{LS}(\boldsymbol{c}_t)_w = \begin{cases} 1-\epsilon & \text{if } w = \arg\max \boldsymbol{c}_t \\ \frac{\epsilon}{|\mathcal{V}|-1} & \text{otherwise} \end{cases} \tag{2}$$

where ϵ is the label-smoothing parameter. Since $\text{LS}(\boldsymbol{c}_t)$ is constrained in the probability simplex $\Delta^{|\mathcal{V}|-1}$, we set each $\boldsymbol{z}_t$ to $\log \text{LS}(\boldsymbol{c}_t) \in \mathbb{R}^{|\mathcal{V}|}$ as the initialization for optimizing the soft solution in the next iteration.

4.2 Constrained Collisions

The Aggressive approach is very effective at finding collisions, but it can output nonsensical sentences. Since these sentences have high perplexity under a language model (LM), simple filtering can eliminate them from consideration. To evade perplexity-based filtering, we impose a soft constraint on collision generation and jointly maximize target similarity and LM likelihood:

$$\max_{\boldsymbol{c} \in \mathcal{X}} (1-\beta) \cdot \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}) + \beta \cdot \log P(\boldsymbol{c}; g) \tag{3}$$

[2] We could project the soft collision by annealing the temperature to 0, $\boldsymbol{c} = [\arg\max \boldsymbol{z}_1, \ldots, \arg\max \boldsymbol{z}_T]$. However, this approach yields sub-optimal results because the hard arg max discards information from nearby words.

where $P(\boldsymbol{c}; g)$ is the LM likelihood for collision $\boldsymbol{c}$ under a pre-trained LM g and $\beta \in [0, 1]$ is an interpolation coefficient.

We investigate two different approaches for solving the optimization in equation 3: (a) adding a regularization term on soft $\check{\boldsymbol{c}}$ to approximate the LM likelihood, and (b) steering a pre-trained LM to generate natural-looking $\boldsymbol{c}$.

4.2.1 Regularized Aggressive Collisions

Given a language model g, we can incorporate a soft version of the LM likelihood as a regularization term on the soft aggressive $\check{\boldsymbol{c}}$ computed from the variables $[\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T]$:

$$\Omega = \sum_{t=1}^{T} H(\check{\boldsymbol{c}}_t, P(w_t|\check{\boldsymbol{c}}_{1:t-1}; g)) \tag{4}$$

where $H(\cdot, \cdot)$ is cross entropy, $P(w_t|\check{\boldsymbol{c}}_{1:t-1}; g)$ are the next-token prediction probabilities at t given partial soft collision $\check{\boldsymbol{c}}_{1:t-1}$. Equation 4 relaxes the LM likelihood on hard collisions by using soft collisions as input, and can be added to the objective function for gradient optimization. The variables $\boldsymbol{z}_t$ after optimization will favor words that maximize the LM likelihood.

To further reduce the perplexity of $\boldsymbol{c}$, we exploit the degeneration property of LM, i.e., the observation that LM assigns low perplexity to repeating common tokens (Holtzman et al., 2020), and constrain a span of consecutive tokens in $\boldsymbol{c}$ (e.g., second half of $\boldsymbol{c}$) to be selected from most frequent English words instead of the entire $\mathcal{V}$. This modification produces even more disfluent collisions, but they evade LM-based filtering.

4.2.2 Natural Collisions

Our final approach aims to produce fluent, low-perplexity outputs. Instead of relaxing and then searching, we search and then relax each step for equation 3. This lets us integrate a hard language model while selecting next words in continuous space. In each step t, we maximize:

$$\max_{w \in \mathcal{V}} \quad (1 - \beta) \cdot \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w) + \beta \cdot \log P(\boldsymbol{c}_{1:t-1} \oplus w; g) \tag{5}$$

where $\boldsymbol{c}_{1:t-1}$ is the beam solution found before t. This sequential optimization is essentially LM decoding with a joint search on the LM likelihood and target similarity $\mathcal{S}$, of the collision prefix.

Optimizing equation 5 exactly requires ranking each $w \in \mathcal{V}$ based on LM likelihood $\log P(\boldsymbol{c}_{1:t-1} \oplus w; g)$ and similarity $\mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w)$. Evaluating LM likelihood for every word at each step is efficient because we can cache $\log P(\boldsymbol{c}_{1:t-1}; g)$ and compute the next-word probability in the standard manner. However, evaluating an arbitrary similarity function $\mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w), \forall w \in \mathcal{V}$, requires $|\mathcal{V}|$ forwarded passes to f, which can be computationally expensive.

Perturbed LM logits. Inspired by Plug and Play LM (Dathathri et al., 2020), we modify the LM logits to take similarity into account. We first let $\boldsymbol{\ell}_t = g(\boldsymbol{c}_{1:t-1})$ be the next-token logits produced by LM g at step t. We then optimize from this initialization to find an update that favors words maximizing similarity. Specifically, we let $\boldsymbol{z}_t = \boldsymbol{\ell}_t + \boldsymbol{\delta}_t$ where $\boldsymbol{\delta}_t \in \mathbb{R}^{|\mathcal{V}|}$ is a perturbation vector. We then take a small number of gradient steps on the relaxed similarity objective $\max_{\boldsymbol{\delta}_t} \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus \check{\boldsymbol{c}}_t)$ where $\check{\boldsymbol{c}}_t$ is the relaxed soft word as in equation 1. This encourages the next-word prediction distribution from the perturbed logits, $\check{\boldsymbol{c}}_t$, to favor words that are likely to collide with the input $\boldsymbol{x}$.

Joint beam search. After perturbation at each step t, we find the top K most likely words in $\check{\boldsymbol{c}}_t$. This allows us to only evaluate $\mathcal{S}(\boldsymbol{x}, \boldsymbol{c}_{1:t-1} \oplus w)$ for this subset of words w that are likely under the LM given the current beam context. We rank these top K words based on the interpolation of target loss and LM log likelihood. We assign a score to each beam $\boldsymbol{b}$ and each top K word as in equation 5, and update the beams with the top-scored words.

This process leads to a natural-looking decoded sequence because each step utilizes the true words as input. As we build up a sequence, the search at each step is guided by the joint score of two objectives, semantic similarity and fluency.

5 Experiments

Baseline. We use a simple greedy baseline based on HotFlip (Ebrahimi et al., 2018). We initialize the collision text with a sequence of repeating words, e.g., "the", and iteratively replace all words. In each iteration, we look at every position t and flip the current w_t to v that maximizes the first-order Taylor approximation of target similarity $\mathcal{S}$:

$$\arg\max_{1 \le t \le T, v \in \mathcal{V}} (\boldsymbol{e}_i - \boldsymbol{e}_v)^\top \nabla_{\boldsymbol{e}_t} \mathcal{S}(\boldsymbol{x}, \boldsymbol{c}) \tag{6}$$

where $\boldsymbol{e}_t, \boldsymbol{e}_v$ are the word vectors for w_t and v. Following prior HotFlip-based attacks (Michel et al., 2019; Wallace et al., 2019, 2020), we evaluate $\mathcal{S}$

using the top K words from Equation 6 and flip to the word with the lowest loss to counter the local approximation.

LM for natural collisions. For generating natural collisions, we need a LM g that shares the vocabulary with the target model f. When targeting models that do not share the vocabulary with an available LM, we fine-tune another BERT with an autoregressive LM task on the Wikitext-103 dataset (Merity et al., 2017). When targeting models based on RoBERTa, we use pretrained GPT-2 (Radford et al., 2019) as the LM since the vocabulary is shared.

Unrelatedness. To ensure that collisions $\boldsymbol{c}$ are not semantically similar to inputs $\boldsymbol{x}$, we filter out words that are relevant to $\boldsymbol{x}$ from $\mathcal{V}$ when generating $\boldsymbol{c}$. First, we discard non-stop words in $\boldsymbol{x}$; then, we discard 500 to 2,000 words in $\mathcal{V}$ with the highest similarity score $\mathcal{S}(\boldsymbol{x}, w)$.

Hyperparameters. We use Adam (Kingma and Ba, 2015) for gradient ascent. Detailed hyperparameter setup can be found in table 6 in Appendix A.

Notation. In the following sections, we abbreviate HotFlip baseline as **HF**; aggressive collisions as **Aggr.**; regularized aggressive collisions as **Aggr. Ω** where Ω is the regularization term in equation 4; and natural collisions as **Nat.**

5.1 Tasks and Models

We evaluate our attacks on paraphrase identification, document retrieval, response suggestions and extractive summarization. Our models for these applications are pretrained transformers, including BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), fine-tuned on the corresponding task datasets and matching state-of-the-art performance.

Paraphrase detection. We use the Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005) and Quora Question Pairs (QQP) (Iyer et al., 2017), and attack the first 1,000 paraphrase pairs from the validation set.

We target the BERT and RoBERTa base models for MRPC and QQP, respectively. The models take in concatenated inputs $\boldsymbol{x}_a, \boldsymbol{x}_b$ and output the similarity score as $\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \text{sigmoid}(f(\boldsymbol{x}_a \oplus \boldsymbol{x}_b))$. We fine-tune them with the suggested hyperparameters. BERT achieves 87.51% F1 score on MRPC and RoBERTa achieves 91.6% accuracy on QQP, consistent with prior work.

Document retrieval. We use the Common Core Tracks from 2017 and 2018 (Core17/18). They have 50 topics as queries and use articles from the New York Times Annotated Corpus and TREC Washington Post Corpus, respectively.

Our target model is Birch (Yilmaz et al., 2019a,b). Birch retrieves 1,000 candidate documents using the BM25 and RM3 baseline (Abdul-jaleel et al., 2004) and re-ranks them using the similarity scores from a fine-tuned BERT model. Given a query $\boldsymbol{x}_q$ and a document $\boldsymbol{x}_d$, the BERT model assigns similarity scores $\mathcal{S}(\boldsymbol{x}_q, \boldsymbol{x}_i)$ for each sentence $\boldsymbol{x}_i$ in $\boldsymbol{x}_d$. The final score used by Birch for re-reranking is: $\gamma \cdot \mathcal{S}_{\text{BM25}} + (1-\gamma) \cdot \sum_i \kappa_i \cdot \mathcal{S}(\boldsymbol{x}_q, \boldsymbol{x}_i)$ where $\mathcal{S}_{\text{BM25}}$ is the baseline BM25 score and γ, κ_i are weight coefficients. We use the published models[3] and coefficient values for evaluation.

We attack similarity scores $\mathcal{S}(\boldsymbol{x}_q, \boldsymbol{x}_i)$ by inserting sentences that collide with $\boldsymbol{x}_q$ into irrelevant $\boldsymbol{x}_d$. We filter out query words when generating collisions $\boldsymbol{c}$ so that term frequencies of query words in $\boldsymbol{c}$ are 0, thus inserting collisions does not affect the original $\mathcal{S}_{\text{BM25}}$. For each of the 50 query topics, we select irrelevant articles that are ranked from 900 to 1000 by Birch and insert our collisions into these articles to boost their ranks.

Response suggestion. We use the Persona-chat (Chat) dataset of dialogues (Zhang et al., 2018). The task is to pick the correct utterance in each dialogue context from 20 choices. We attack the first 1,000 contexts from the validation set.

We use transformer-based Bi- and Poly-encoders that achieved state-of-the-art results on this dataset (Humeau et al., 2020). Bi-encoders compute a similarity score for the dialogue context $\boldsymbol{x}_a$ and each possible next utterance $\boldsymbol{x}_b$ as $\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}_b) = f_{\text{pool}}(\boldsymbol{x}_a)^\top f_{\text{pool}}(\boldsymbol{x}_b)$ where $f_{\text{pool}}(\boldsymbol{x}) \in \mathbb{R}^h$ is the pooling-over-time representation from transformers. Poly-encoders extend Bi-encoders compute $\mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \sum_{i=1}^{T} \alpha_i \cdot f(\boldsymbol{x}_a)_i^\top f_{\text{pool}}(\boldsymbol{x}_b)$ where α_i is the weight from attention and $f(\boldsymbol{x}_a)_i$ is the ith token's contextualized representation. We use the published models[4] for evaluation.

Extractive summarization. We use the CNN / DailyMail (CNNDM) dataset (Hermann et al., 2015), which consists of news articles and labeled overview highlights. We attack the first 1,000 articles from the validation set.

Our target model is PreSumm (Liu and Lapata, 2019). Given a text $\boldsymbol{x}_d$, PreSumm first obtains a

[3] https://github.com/castorini/birch
[4] https://parl.ai/docs/zoo.html

$\boldsymbol{c}$ type	MRPC		QQP		Core17/18			Chat-Bi		Chat-Poly		CNNDM		
	$\mathcal{S}$	% Succ	$\mathcal{S}$	% Succ	$\mathcal{S}$	$r \leq 10$	≤ 100	$\mathcal{S}$	$r = 1$	$\mathcal{S}$	$r = 1$	$\mathcal{S}$	$r = 1$	$r \leq 3$
Gold	0.87	-	0.90	-	1.34	-	-	17.14	-	25.30	-	0.51	-	-
HF	0.60	67.3%	0.55	54.8%	-0.96	0.0%	16.5%	21.20	78.5%	28.82	73.1%	0.50	67.9%	96.5%
Aggr.	0.93	97.8%	0.98	97.3%	1.62	49.9%	86.7%	23.79	99.8%	31.94	99.4%	0.69	99.4%	100.0%
Aggr. Ω	0.69	81.0%	0.91	91.1%	0.86	20.6%	69.7%	21.66	92.9%	29.51	90.7%	0.58	90.7%	100.0%
Nat.	0.78	98.6%	0.88	88.8%	0.77	12.3%	60.6%	22.15	86.0%	31.10	86.6%	0.37	30.4%	77.7%

Table 2: Attack results. r is the rank of collisions among candidates. Gold denotes the ground truth.

vector representation $\boldsymbol{\phi}_i \in \mathbb{R}^h$ for each sentence $\boldsymbol{x}_i$ using BERT, and scores each sentence $\boldsymbol{x}_i$ in the text as $\mathcal{S}(\boldsymbol{x}_d, \boldsymbol{x}_i) = \text{sigmoid}(\boldsymbol{u}^\top f(\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_T)_i)$ where $\boldsymbol{u}$ is a weight vector, f is a sentence-level transformer, and $f(\cdot)_i$ is the ith sentence's contextualized representation. Our objective is to insert a collision $\boldsymbol{c}$ into $\boldsymbol{x}_d$ such that the rank of $\mathcal{S}(\boldsymbol{x}_d, \boldsymbol{c})$ among all sentences is high. We use the published models[5] for evaluation.

5.2 Attack Results

For all attacks, we report the similarity score $\mathcal{S}$ between $\boldsymbol{x}$ and $\boldsymbol{c}$; the "gold" baseline is the similarity between $\boldsymbol{x}$ and the ground truth. For MRPC, QQP, Chat, and CNNDM, the ground truth is the annotated label sentences (e.g., paraphrases or summaries); for Core17/18, we use the sentences with the highest similarity $\mathcal{S}$ to the query. For MRPC and QQP, we also report the percentage of successful collisions with $\mathcal{S} > 0.5$. For Core17/18, we report the percentage of irrelevant articles ranking in the top-10 and top-100 after inserting collisions. For Chat, we report the percentage of collisions achieving top-1 rank. For CNNDM, we report the percentage of collisions with the top-1 and top-3 ranks (likely to be selected as summary). Table 2 shows the results.

On MRPC, aggressive and natural collisions achieve around 98% success; aggressive ones have higher similarity $\mathcal{S}$. With regularization Ω, success rate drops to 81%. On QQP, aggressive collisions achieve 97% vs. 90% for constrained collisions.

On Core17/18, aggressive collisions shift the rank of almost half of the irrelevant articles into the top 10. Regularized and natural collisions are less effective, but more than 60% are still ranked in the top 100. Note that query topics are compact phrases with narrow semantics, thus it might be harder to find constrained collisions for them.

On Chat, aggressive collisions achieve rank of 1 more than 99% of the time for both Bi- and Poly-encoders. With regularization Ω, success drops slightly to above 90%. Natural collisions are less successful, with 86% ranked as 1.

On CNNDM, aggressive collisions are almost always ranked as the top summarizing sentence. HotFlip and regularized collisions are in the top 3 more than 96% of the time. Natural collisions perform worse, with 77% ranked in the top 3.

Aggressive collisions always beat HotFlip on all tasks; constrained collisions are often better, too. The similarity scores $\mathcal{S}$ for aggressive collisions are always higher than for the ground truth.

[5] https://github.com/nlpyang/PreSumm

$\boldsymbol{c}$ type	MRPC	QQP	Core	Chat	CNNDM
	F_{BERT}	F_{BERT}	P_{BERT}	P_{BERT}	F_{BERT}
Gold	0.66	0.68	0.17	0.14	0.38
Aggr.	-0.22	-0.17	-0.34	-0.31	-0.31
Aggr. Ω	-0.34	-0.34	-0.48	-0.43	-0.36
Nat.	-0.12	-0.09	-0.11	-0.10	-0.25

Table 3: BERTSCORE between collisions and target inputs. Gold denotes the ground truth.

5.3 Evaluating Unrelatedness

We use BERTSCORE (Zhang et al., 2020) to demonstrate that our collisions are unrelated to the target inputs. Instead of exact matches in raw texts, BERTSCORE computes a semantic similarity score, ranging from -1 to 1, between a candidate and a reference by using contextualized representation for each token in the candidate and reference.

The baseline for comparisons is BERTSCORE between the target input and the ground truth. For MRPC and QQP, we use $\boldsymbol{x}$ as reference; the ground truth is paraphrases as given. For Core17/18, we use $\boldsymbol{x}$ concatenated with the top sentences except the one with the highest $\mathcal{S}$ as reference; the ground truth is the sentence in the corpus with the highest $\mathcal{S}$. For Chat, we use the dialogue contexts as reference and the labeled response as the ground truth. For CNNDM, we use labeled summarizing sentences in articles as reference and the given abstractive summarization as the ground truth.

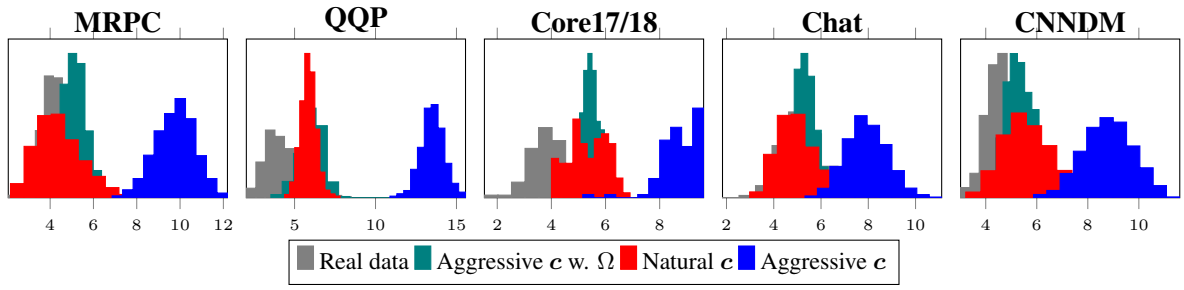

Figure 2: Histograms of log perplexity evaluated by GPT-2 on real data and collisions.

c type	MRPC		QQP		Core17/18		Chat		CNNDM	
	FP@90	FP@80	FP@90	FP@80	FP@90	FP@80	FP@90	FP@80	FP@90	FP@80
HF	2.1%	0.8%	3.1%	1.2%	4.6%	1.2%	1.5%	0.8%	3.2%	3.1%
Aggr.	0.0%	0.0%	0.0%	0.0%	0.8%	0.7%	5.2%	2.6%	3.1%	3.1%
Aggr. Ω	47.5%	35.6%	15.8%	11.9%	29.3%	17.8%	76.5%	65.3%	52.8%	35.7%
Nat.	94.9%	89.2%	20.5%	12.1%	13.7%	10.9%	93.8%	86.5%	59.8%	37.7%

Table 4: Effectiveness of perplexity-based filtering. FP@90 and FP@80 are false positive rates (percentage of real data mistakenly filtered out) at thresholds that filter out 90% and 80% of collisions, respectively.

c type	MRPC		Chat	
	BERT	RoBERTa	Bi → Poly	Poly → Bi
HF	34.0%	0.0%	55.3%	48.9%
Aggr.	64.5%	0.0%	77.4%	71.3%
Aggr. Ω	38.9%	0.0%	60.5%	56.0%
Nat.	41.4%	0.0%	71.4%	68.2%

Table 5: Percentage of successfully transferred collisions for MRPC and Chat.

For MPRC, QQP and CNNDM, we report F_{BERT} (F_1) score. For Core17/18 and Chat, we report P_{BERT} (content from reference found in candidate) because the references are longer and not token-wise equivalent to collisions or ground truth. Table 3 shows the results. The scores for collisions are all negative while the scores for target inputs are positive, indicating that our collisions are unrelated to the target inputs. Since aggressive and regularized collisions are nonsensical, their contextualized representations are less similar to the reference texts than natural collisions.

5.4 Transferability of Collisions

To evaluate whether collisions generated for one target model f are effective against a different model f', we use MRPC and Chat datasets. For MRPC, we set f' to a BERT base model trained with a different random seed and a RoBERTa model. For Chat, we use Poly-encoder as f' for Bi-encoder f, and vice versa. Both Poly-encoder and Bi-encoder are fine-tuned from the same pretrained transformer model. We report the percentage of successfully transferred attacks, e.g., $\mathcal{S}(\boldsymbol{x}, \boldsymbol{c}) > 0.5$ for MRPC and $r = 1$ for Chat.

Table 5 summarizes the results. All collisions achieve some transferability (40% to 70%) if the model architecture is the same and f, f' are fine-tuned from the same pretrained model. Furthermore, our attacks produce more transferable collisions than the HotFlip baseline. No attacks transfer if f, f' are fine-tuned from different pretrained models (BERT and RoBERTa). We leave a study of transferability of collisions across different types of pretrained models to future work.

6 Mitigation

Perplexity-based filtering. Because our collisions are synthetic rather than human-generated texts, it is possible that their perplexity under a language model (LM) is higher than that of real text. Therefore, one plausible mitigation is to filter out collisions by setting a threshold on LM perplexity.

Figure 2 shows perplexity measured using GPT-2 (Radford et al., 2019) for real data and collisions for each of our attacks. We observe a gap between the distributions of real data and aggressive collisions, showing that it might be possible to find a threshold that discards aggressive collisions while retaining the bulk of the real data. On the other hand, constrained collisions (regularized or natural) overlap with the real data.

We quantitatively measure the effectiveness of perplexity-based filtering using thresholds that would discard 80% and 90% of collisions, respec-

tively. Table 4 shows the false positive rate, i.e., fraction of the real data that would be mistakenly filtered out. Both HotFlip and aggressive collisions can be filtered out with little to no false positives since both are nonsensical. For regularized or natural collisions, a substantial fraction of the real data would be lost, while 10% or 20% of collisions evade filtering. On MRPC and Chat, perplexity-based filtering is least effective, discarding around 85% to 90% of the real data.

Learning-based filtering. Recent works explored automatic detection of generated texts using a binary classifier trained on human-written and machine-generated data (Zellers et al., 2019; Ippolito et al., 2020). These classifiers might be able to filter out our collisions—assuming that the adversary is not aware of the defense.

As a general evaluation principle (Carlini et al., 2019), any defense mechanism should assume that the adversary has complete knowledge of how the defense works. In our case, a stronger adversary may use the detection model to craft collisions to evade the filtering. We leave a thorough evaluation of these defenses to future work.

Adversarial training. Including adversarial examples during training can be effective against inference-time attacks (Madry et al., 2018). Similarly, training with collisions might increase models' robustness against collisions. Generating collisions for each training example in each epoch can be very inefficient, however, because it requires additional search on top of gradient optimization. We leave adversarial training to future work.

7 Conclusion

We demonstrated a new class of vulnerabilities in NLP applications: semantic collisions, i.e., input pairs that are unrelated to each other but perceived by the application as semantically similar. We developed gradient-based search algorithms for generating collisions and showed how to incorporate constraints that help generate more "natural" collisions. We evaluated the effectiveness of our attacks on state-of-the-art models for paraphrase identification, document and sentence retrieval, and extractive summarization. We also demonstrated that simple perplexity-based filtering is not sufficient to mitigate our attacks, motivating future research on more effective defenses.

Acknowledgements. This research was supported in part by NSF grants 1916717 and 2037519, the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program, and a Google Faculty Research Award.

References

Nasreen Abdul-jaleel, James Allan, W Bruce Croft, O Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *TREC*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *ICLR*.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: a simple approach to controlled text generation. In *ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *International Workshop on Paraphrasing*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *ACL*.

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

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving Google's Perspective API built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR*.

Daphne Ippolito, Daniel Duckworth, Douglas Eck, and Chris Callison-Burch. 2020. Automatic detection of generated text is easiest when humans are fooled. In *ACL*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First Quora dataset release: Question pairs [online]. 2017.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*.

Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. 2019a. Excessive invariance causes adversarial vulnerability. In *ICLR*.

Jörn-Henrik Jacobsen, Jens Behrmannn, Nicholas Carlini, Florian Tramer, and Nicolas Papernot. 2019b. Exploiting excessive invariance caused by norm-bounded adversarial robustness. *arXiv preprint arXiv:1903.10484*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, et al. 2016. Smart Reply: Automated response suggestion for email. In *KDD*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on Sesame Street! Model extraction of BERT-based APIs. In *ICLR*.

Ke Li, Tianhao Zhang, and Jitendra Malik. 2019. Approximate feature collisions in neural nets. In *NeurIPS*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *IJCAI*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP-IJCNLP*.

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. *arXiv preprint arXiv:1907.11692*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. Breaking NLP: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems. In *Workshop on Building Linguistically Generalizable NLP Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *ICLR*.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *NAACL*.

Bijeeta Pal and Shruti Tople. 2020. To transfer or not to transfer: Misclassification attacks against transfer learned text classifiers. *arXiv preprint arXiv:2001.02438*.

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *ACL*.

Laura Scharff. Introducing question merging [online]. 2015.

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

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP-IJCNLP*.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019a. Applying BERT to document retrieval with Birch. In *EMNLP-IJCNLP*.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019b. Cross-domain modeling of sentence-level evidence for document retrieval. In *EMNLP-IJCNLP*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *NeurIPS*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *ICLR*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *ICLR*.

A Additional Experiment Details

MRPC	B	K	N	T	η	τ	β
Aggr.	10	30	30	20	0.001	1.0	0.0
Aggr. Ω	5	15	30	30	0.001	1.0	0.8
Nat.	10	128	5	25	0.001	0.1	0.05
QQP							
Aggr.	10	30	30	15	0.001	1.0	0.0
Aggr. Ω	5	15	30	30	0.001	1.0	0.8
Nat.	10	64	5	20	0.001	0.1	0.0
Core							
Aggr.	5	50	30	30	0.001	1.0	0.0
Aggr. Ω	5	40	30	60	0.001	1.0	0.85
Nat.	10	150	5	35	0.001	0.1	0.015
Chat							
Aggr.	5	30	30	15	0.001	1.0	0.0
Aggr. Ω	5	20	30	25	0.001	1.0	0.8
Nat.	10	128	5	20	0.001	0.1	0.15
Summ							
Aggr.	5	10	30	15	0.001	1.0	0.0
Aggr. Ω	5	10	30	30	0.001	1.0	0.8
Nat.	5	64	5	20	0.001	1.0	0.02

Table 6: Hyper-parameters for each experiment. B is the beam size for beam search. K is the number of top words evaluated at each optimization step. N is the number of optimization iterations. T is the sequence length. η is the step size for optimization. τ is the temperature for softmax. β is the interpolation parameter in equation 5.

Hyper-parameters. We report the hyper-parameter values for our experiments in Table 6. The label-smoothing parameter ϵ for aggressive collisions is set to 0.1. The hyper-parameters for the baseline are the same as for aggressive collisions.

Runtime. On a single GeForce RTX 2080 GPU, our attacks generate collisions in 10 to 60 seconds depending on the length of target inputs.

B Additional Collision Examples

Tables 7, 8, 9, 10 show collision additional examples for MRPC/QQP, Core17/18, Chat, and CNNDM respectively.

MRPC/QQP target inputs and collisions	Outputs
MRPC Input $(\boldsymbol{x})$: PCCW 's chief operating officer, Mike Butcher, and Alex Arena, the chief financial officer, will report directly to Mr So.	
Aggressive $(\boldsymbol{c})$: primera metaphysical declaration dung southernmost among structurally favorably endeavor from superior morphology indirectly materialized yesterday sorority would indirectly ⟨ sg	99.5%
Regularized aggressive $(\boldsymbol{c})$: in one time rave rave — in … " in but … rv rv smacked a a of a a a a a a a a a of a a	81.6%
Natural $(\boldsymbol{c})$: in 1989 and joined the new york giants in 1990	81.7%
MRPC Input $(\boldsymbol{x})$: Under terms of the deal, Legato stockholders will receive 0.9 of a share of EMC common stock for each share of Legato stock.	
Aggressive $(\boldsymbol{c})$: moreover author elk telling assert honest exact inventions locally mythical confirms newer feat said assert according locally prefecture municipal realization	96.7%
Regularized aggressive $(\boldsymbol{c})$: in new " news lust release " on connected different " vibe " reassure females and and to to and and to and to and to and to and to	95.0%
Natural $(\boldsymbol{c})$: she is also a member of the united states house of representatives, serving as a representative	83.4%
QQP Input $(\boldsymbol{x})$: How can I slowly lose weight?	
Aggressive $(\boldsymbol{c})$: sustain fitness recover bru become bolst Enhanced additional distinguished contend crunch Cutting Vital Time cov	80.5%
Regularized aggressive $(\boldsymbol{c})$: fat Ensure burner www Enhancement Lar Cure Dou St Reaper of of of of of a to and to the the the and to to to of of a of	85.2%
Natural $(\boldsymbol{c})$: be able that in less long time it	80.2%

Table 7: Collision examples for MRPC and QQP. Outputs are the probability scores produced by the model for whether the input and the collisions are paraphrases.

Core17/18 query inputs and collisions	r
Query $(\boldsymbol{x})$: abuses of e-mail	
Aggressive $(\boldsymbol{c})$: trailing helsinki, competent regimes internally outlaw wireless offence road : cables by nhs sided head lockheed ford announce oblast million offenders climb ranged postal courier administrations courtesy guangdong oracle	1
Regularized aggressive $(\boldsymbol{c})$: un / australia overthrow " — of most telegraph telegraph operations " : the state office in consensus in document lifts down us " by trial " for using ¡ the a and a to and a and a to the a to a a to to a a and a a and a a a the a to to	1
Natural $(\boldsymbol{c})$: the itc ordered all wireless posts confiscated and usps were stripped of their offices and property, leading to a number of	3
Query $(\boldsymbol{x})$: heroic acts	
Aggressive $(\boldsymbol{c})$: colossal helmet vedic bro axes resembling neighbours lead floods blacksmith : evening eligibility caller indicates sculptor coroner lakshmi' than lama announced seizure branded, crafts informing nottinghamshire watch commission.	1
Regularized aggressive $(\boldsymbol{c})$: recorded health and human execution followed, applause prompted, support increased extended : thayer and some there danger, while frank teammate followed feat of personal injury injuries of a the a of the a of the the of of the and of of of of and of of of of and of and of of of the	1
Natural $(\boldsymbol{c})$: the american fighter (1 november 1863 ; kia for his feat) — the japanese ship carrying the cargo of wheat from australia to sydney	11
Query $(\boldsymbol{x})$: cult lifestyles	
Aggressive $(\boldsymbol{c})$: indiana - semiconductor cut and radiating fire damage, domain reproductive nighttime pastoral calendar failing critical soils indicates force practice ritual belarus stall ; cochin sabha fragmented nut dominance owing	1
Regularized aggressive $(\boldsymbol{c})$: preferred fruits, as willow, suggested to botanist ro spike'for resident nursery : big spreads of pipe rolls and other european pie, a long season at the a and a a and the and of of and of the a of and of of and of of and of of of of and of of the	2
Natural $(\boldsymbol{c})$: the early 1980s their appeal soared : during summerslam in los angeles (1993), a large number of teenagers went to church to confess their connection to the	15
Query $(\boldsymbol{x})$: art, stolen, forged	
Aggressive $(\boldsymbol{c})$: colossal helmet vedic bro axes resembling neighbours lead floods blacksmith : evening eligibility caller indicates sculptor coroner lakshmi'than lama announced seizure branded, crafts informing nottinghamshire watch commission	1
Regularized aggressive $(\boldsymbol{c})$: - house and later car dead with prosecutors remaining : " and cathedral gallery ' import found won british arrest prosecution a a portrait or mural (patron at from the the to the a and a to the a and to the a to the of a and to the the and to the to the a and a	3
Natural $(\boldsymbol{c})$: the work which left its owner by a mishandle - the royal academy's chief judge inquest	8

Table 8: Collision examples for Core17/18. r are the ranks of irrelevant articles after inserting the collisions.

Chat target inputs and collisions	r
Context $(\boldsymbol{x})$: i'm 33 and love giving back i grew up poor. i did too , back during the great depression.	
Aggressive $(\boldsymbol{c})$: that to existed with and that is with cope warlord s s came the on	1
Regularized aggressive $(\boldsymbol{c})$: camps wii also until neutral in later addiction and the the the the of to and the the the of to and to the the	1
Natural $(\boldsymbol{c})$: was the same side of abject warfare that had followed then for most people in this long	1
Context $(\boldsymbol{x})$: i am a male . i have a children and a dogs . hey there how is it going ?	
Aggressive $(\boldsymbol{c})$: is frantically in to it programs s junior falls of it s talking a juan	1
Regularized aggressive $(\boldsymbol{c})$: in on from the it department with gabrielle and the the and a and a a to a a and of and of and of	1
Natural $(\boldsymbol{c})$: as of this point, and in the meantime it's having very technical support : it employs	1
Context $(\boldsymbol{x})$: hi ! how are you doing today ? great , just ate pizza my favorite . . and you ? that's not as good as shawarma	
Aggressive $(\boldsymbol{c})$: safer to eat that and was mickey in a cut too on it s foreigner	1
Regularized aggressive $(\boldsymbol{c})$: dipped in in kai tak instead of that and the the a of a of a to to the to and a a of a	1
Natural $(\boldsymbol{c})$: not as impressive, its artistic production provided an environment	1

Table 9: Collision examples for Chat. r are the ranks of collisions among the candidate responses.

CNNDM ground truth and collisions	r
Truth: zayn malik is leaving one direction . rumors about such a move had started since malik left the band 's tour last week .	
Aggressive $(\boldsymbol{c})$: bp interest yd £ offering funded fit literacy 2020 can propose amir pau laureate conservation	1
Regularized aggressive $(\boldsymbol{c})$: the are shortlisted to compete 14 times zealand in in the 2015 zealand artist yo a to to to to to to to to to to to to to to	1
Natural $(\boldsymbol{c})$: an estimated $2 billion by 2014 ; however estimates suggest only around 20 percent are being funded from	1
Truth: she says sometimes his attacks are so violent, she's had to call the police to come and save her.	
Aggressive $(\boldsymbol{c})$: bwf special editor councils want qc iec melinda rey marry selma iec qc disease translated	1
Regularized aggressive $(\boldsymbol{c})$: poll is in 2012 eight percent b dj dj dj coco behaviors in dj coco and a a to of to to to the a a to the to a	1
Natural $(\boldsymbol{c})$: first national strike since world war ii occurred between january 13 – 15 2014 ; this date will occur	1

Table 10: Collision examples for CNNDM. ***Truth*** are the true summarizing sentences. r are the ranks of collisions among all sentences in the news articles.