Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models

Jiawei Liu
Wuhan University
laujames2017@whu.edu.cn

Yangyang Kang
Alibaba Group, China
yangyang.kangyy@alibaba-inc.com

Di Tang
Indiana University Bloomington
tangd@iu.edu

Kaisong Song
Northeastern University, China
Alibaba Group, China
kaisong.sks@alibaba-inc.com

Changlong Sun
Alibaba Group, China
changlong.scl@taobao.com

Xiaofeng Wang
Indiana University Bloomington
xw7@indiana.edu

Wei Lu*
Wuhan University
weilu@whu.edu.cn

Xiaozhong Liu*
Worcester Polytechnic Institute
xliu14@wpi.edu

ABSTRACT

Neural text ranking models have witnessed significant advancement and are increasingly being deployed in practice. Unfortunately, they also inherit adversarial vulnerabilities of general neural models, which have been detected but remain underexplored by prior studies. Moreover, the inherit adversarial vulnerabilities might be leveraged by blackhat SEO to defeat better-protected search engines. In this study, we propose an imitation adversarial attack on black-box neural passage ranking models. We first show that the target passage ranking model can be transparentized and imitated by enumerating critical queries/candidates and then train a ranking imitation model. Leveraging the ranking imitation model, we can elaborately manipulate the ranking results and transfer the manipulation attack to the target ranking model. For this purpose, we propose an innovative gradient-based attack method, empowered by the pairwise objective function, to generate adversarial triggers, which causes premeditated disorderliness with very few tokens. To equip the trigger camouflages, we add the next sentence prediction loss and the language model fluency constraint to the objective function. Experimental results on passage ranking demonstrate the effectiveness of the ranking imitation attack model and adversarial triggers against various SOTA neural ranking models. Furthermore, various mitigation analyses and human evaluation show the effectiveness of camouflages when facing potential mitigation approaches. To motivate other scholars to further investigate this novel and important problem, we make the experiment data and code publicly available.

*Corresponding authors

CCS CONCEPTS

• **Security and privacy** → **Software and application security**; • **Information systems** → **Information retrieval**; • **Computing methodologies** → *Natural language processing*.

KEYWORDS

neural ranking, deep neural network, passage ranking, adversarial attack, black-box attack, transfer-based attack

1 INTRODUCTION

The two-stage retrieve-then-rerank architecture has proven to be an effective and widely adopted strategy for text ranking tasks [1]. As for the reranker, the trendsetting approach is empowered by the neural model. With the recent accomplishment of pre-trained model, BERT [10] for instance, pre-trained then fine-tuned transformers achieved state-of-the-art performance on numerous text ranking tasks [13, 35, 39]. However, neural ranking models may inherit the adversarial vulnerabilities of neural networks [49], i.e, a small deliberate perturbation (e.g., some pixel variations on an image) could trigger dramatic change in the learning result [65]. Such vulnerability has raised dedicated concerns of the robustness and reliability of text ranking systems integrated with neural networks [46]. A trustworthy ranking system should be well aware of the malicious attack, in which deliberate but imperceptible content variations may cause the catastrophic ranking disorder.

Previous adversarial ranking attacks primarily focus on DNN-based image ranking systems [28, 65, 66], whereas the vulnerability of deep neural text ranking remains underexplored. Taking the passage search as an example, a fair ranking system should rank the passage collections according to their semantic relevance to the query. Nevertheless, a malicious content producer may attempt to
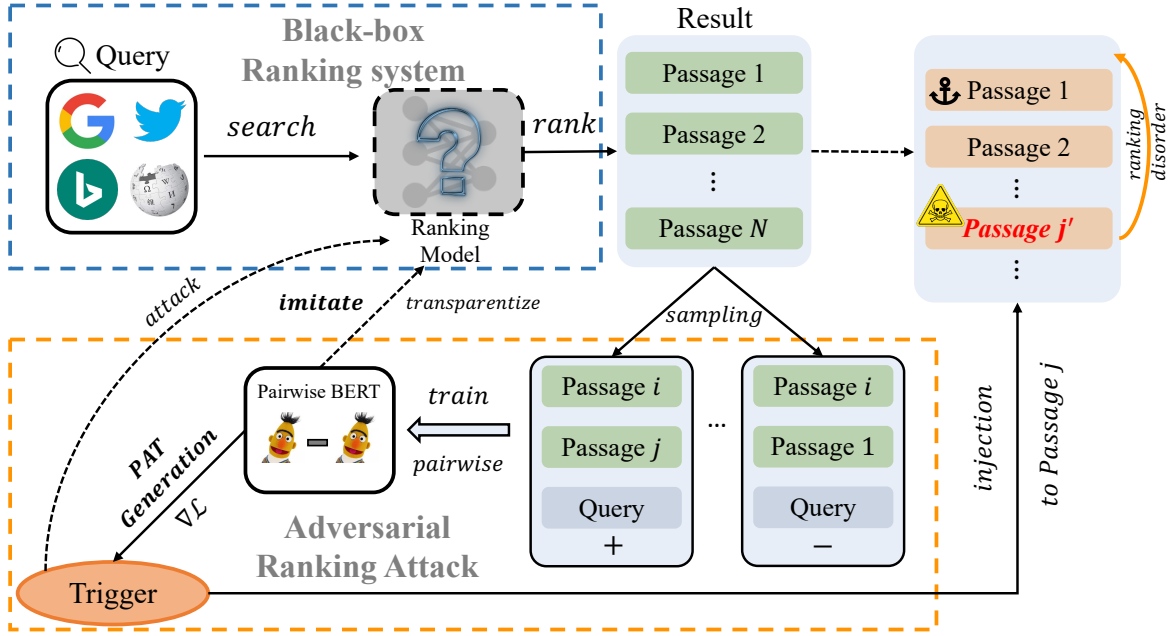
Figure 1: An overview of our imitation adversarial attacks for a black-box neural passage ranking system.

raise the rank of his/her own passage, e.g., opinion or advertisement, by adding small perturbations to the passage. Recently, scholars investigated potential adversarial attacks on text classification [11, 48, 51], machine translation [2, 52], and question answering [12, 50], which inspire studies on neural text ranking trustworthiness. For instance, Song et al. [46] proved that an irrelevant document can be prioritized by inserting a collision text generated based on the target white-box pointwise relevance scoring model. The target model is a BERT classifier fine-tuned with the relevance label of a document to a query. In general, document ranking examines relations among candidate documents in a query context [3], and the effectiveness of existing text classification-based adversarial attacks can be compromised. Moreover, when the attack process is coached by gradient, model transparency, e.g., the architecture, hyperparameters, and training data, becomes inevitable [55]. Real-world ranking systems, unfortunately, prohibit algorithmic white-box access, which hinders the application of existing attack methods.

In this study, we propose a novel black-box attack method against neural text ranking systems in terms of adversarial trigger transferability among different neural networks. Specifically, we train a ranking imitation model to demystify the target ranking model for knowledge distillation. The proposed ranking imitation model is a Pairwise BERT ranker, which is trained on triplets *(query, relative positive candidate, relative negative candidate)* sampled from ranking lists of the target model. It does not require the relevance label or score of a document to a query, which is vital to train a pointwise ranker. We prove that the ranking imitation model-inferred candidate relatedness can unveil black-box ranking model vulnerabilities for potential attacks. Experiments show that the ranking imitation model achieves comparable performance on ranking task

with the target text ranking system, i.e., 89.9 vs. 90.1 on TREC DL 2019 [1] MRR@10 and 77.2% overlap among top-10 candidates. This similarity essentially increases the transferability of our adversarial attack triggers.

In terms of adversarial attack trigger generation, we propose a novel Pairwise Anchor-based Trigger (PAT) generation model by leveraging the pairwise structural information from imitation ranking outcome. The generation process, navigated by the ranking imitation model formulated gradient, can generate the tailored trigger for each candidate passage that carries camouflages to mislead the original ranker. Moreover, to avoid generating nonsensical triggers with high perplexity, which can be easily filtered, we apply the fluency constraint with a language model to the objective. Since the semantic of the trigger that differs significantly from the passage is highly susceptible to be detected, we add the next sentence prediction (NSP) loss between the trigger and the target candidate to equip the trigger camouflages. As a prominent result, our trigger can successfully boost the ranks of nearly 98% marginal passages (ranked 995-1000) and essentially boost nearly 40% of them to rank in top-100. Extensive experiments on passage ranking demonstrate the effectiveness of our adversarial attack trigger generation method and transferability of adversarial triggers.

An example scenario of our attack is to elevate blackhat SEO to defeat better-protected search engines. Our research shows that SOTA embedding-based ranking models, as those used by search engines (Google, Amazon, etc), are vulnerable to blackhat SEO, a well-known cybercrime (e.g., advertising drugs, gambling, and porn [17, 29]). Traditional SEO tricks like keyword (or similar word) stuffing[2] become less effective with the current trend of moving

---

[1]https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019
[2]https://seranking.com/blog/keyword-stuffing

towards BERT [59] + anti-SEO solutions [64], due to their lack of stealthiness (Section 6.3). Our research, however, shows that ranking manipulation against embedding models can be achieved under the consistency and fluency constraints, even without knowledge of the target, opening the door to stealthy SEO attacks.

The contributions of this paper are threefold:

(1) This is a pioneer transfer-based attack method investigation against black-box neural text ranking systems. By querying the victim ranking system, one can transparentize black-box model via ranking imitation, and generate adversarial attack triggers to disorder the victim ranking system.
(2) We propose an innovative Pairwise Anchor-based Trigger (PAT) generation model with pairwise loss on anchor candidates for camouflaged text ranking attack and manipulation.
(3) We employ three large datasets alone with a variety of ranking models to validate the effectiveness of the proposed model. We make all the experiment data and code publicly available to motivate other scholars to further investigate this novel but important problem[3].

2 BACKGROUND AND RELATED WORKS

In this section, we briefly introduce the existing works of deep text ranking, adversarial ranking attacks, and model imitation. Then we state the threat model and objectives.

2.1 Text Ranking

Classical Text Ranking models mainly rely on exact term matching between query and document text using the bag-of-words framework, such as Boolean Retrieval and BM25 [44]. However, this type of method has limited capability of modeling human languages. Nowadays, due to the efficiency of processing queries and documents, they are still widely used in production systems for first-stage retrieval. To deal with vocabulary mismatch, continuous vector space representation such word2vec [32] and Glove [38] coupled with neural networks to produce soft matching score. These neural ranking models can be generally classified into three classes: representation-based models [22, 45], interaction-based models [16, 20], and hybrid models [33].

Deep transformer models pre-trained with language model objectives, represented by BERT [10], have made a huge impact on neural text ranking. Nogueira and Cho [35] are the first to demonstrate the effectiveness of BERT in text ranking task. After that, more and more pre-trained transformer LM models fine-tuned on the specific corpus achieve state-of-the-art in text ranking with significant performance improvement [7, 13, 39]. In the Deep Learning Track at TREC 2019 [6], analysis of the results showed that, BERT-based models achieved substantially higher effectiveness than "pre-BERT" models, across implementations by different teams [63]. To lower the query latency and make the ranking model feasible for production deployment, Hofstätter et al. [19] adapt knowledge distillation to neural ranking models. However, the performance improvement of BERT-based text ranking models also inherited the vulnerabilities of neural networks, which have been detected [46] but still remain under-explored by prior studies.

---
[3]https://github.com/LauJames/PAT

2.2 Adversarial Ranking Attack

Szegedy et al. [49] find that DNN is susceptible to small adversarial perturbation added to inputs, which leads to misbehavior. Following this finding, a series of researches and applications about adversarial attacks have emerged in text classification [11, 48, 51], machine translation [2, 52], and question answering [12, 50]. Different from classification tasks where texts are predicted independently, the rank of one candidate is usually related to the query as well as other candidates for ranking tasks. Since the ranking result is determined by the relative relations among candidates and queries, it is essential to take the relations into consideration for a qualified ranking model. As a result, for text ranking scenarios, the existing adversarial text classification attack methods are incompatible.

For ranking systems, the risk of malicious user manipulating the target ranking always exists [14]. Previous adversarial ranking attacks primarily focus on DNN-based image ranking systems [28, 53, 65]. Likewise, the performance improvement of BERT-based document ranking models [61] also inherit the vulnerabilities of neural networks [46]. The existence of aforementioned works in NLP and image ranking inspired the attack against deep text ranking models, it is still insufficiently explored. Although there are several research works explore the attack against neural text ranking models, they focus on document ranking [46] and are under white-box setting [15, 43, 47, 60].

Considering a majority of real-world text ranking systems do not allow white-box access, recently, Wu et al. [59] propose a pseudo relevance-based adversarial ranking attack method, which substitutes words to promote the target document in rankings. Differently, we introduce the transfer-based black-box attack and train a ranking imitation model on triplets sampled from ranking lists of the target ranking model. Leveraging the similarity of the ranking imitation model, we can elaborately manipulate the ranking results and transfer the manipulation attack to the target ranking model. Then we propose our trigger generation method, empowered by the pairwise objective function, to generate adversarial triggers, which causes premeditated disorderliness with very few tokens.

2.3 Model Imitation

Model imitation is closely related to model distillation [18] and extraction (or stealing) [25]. Model distillation aims to train a student model to imitate the predictions of a teacher. Hofstätter et al. [19] investigate the Margin-MSE knowledge distillation across different architectures to improve reranking effectiveness. Recent works have shown that backdoors can persist in the student model if the teacher model is infected with a Trojan [52, 54, 62]. However, most of the model distillation methods utilize the logits or scores of the teacher model, which are unavailable and prohibited in black-box setting. Thus, ranking imitation models cannot be optimized by distribution matching losses commonly used in knowledge distillation, such as Margin-MSE and KL divergence. In terms of model extraction or stealing, it differs from distillation because the training data of the victim (teacher) is unknown, which causes queries for the victim to be out-of-domain.

Despite the above-mentioned challenges, previous works show that model extraction or stealing is possible for text classification [36], reading comprehension, natural language inference [25], and

machine translation [52]. In this paper, we extend these results to text ranking. Generally speaking, absolute positive labels or scores are required to train a pointwise ranker. However, this information is inaccessible for real-world ranking systems. We introduce the Pairwise BERT ranker to imitate a black-box ranking model. The Pairwise BERT ranker is trained on triplets *(query, relative positive candidate, relative negative candidate)* sampled from ranking lists of the target model.

3 THREAT MODEL

Given a textual query q and candidate passages set $P = \{p_1, p_2, ..., p_k\}$, the ranking model calculates a score $s(q, p_i)$ with respect to a candidate passage $p_i \in P$ and query q, and generates ranking list, i.e., $p_1 > p_2 > \cdots > p_k$ if $s(q, p_1) > s(q, p_2) > \cdots > s(q, p_k)$.

Objective of the Adversary. The adversarial text ranking attack aims to find an optimized adversarial trigger (text snippet) which leads to the deliberate ranking disorder. For example, given $s(q, p_i) > s(q, p_j)$, trigger t injected into p_j (i.e., $p'_j = [t; p_j]$, where ; denotes concatenation) can flip ranking result, $s(q, p_i) < s(q, p'_j)$. To make the attack as stealthy as possible, the trigger should be avoided to be nonsensical with high perplexity. Meanwhile, the passage injected with the trigger should be semantically consistent with the original passage.

Capabilities of the Adversary. We focus on the black-box attack setting where the adversary has no knowledge of the target model architecture, training data, and the score function. Accessing ranking result list P produced by the target victim model with the query q is one of the capabilities assumed in our threat model. Moreover, we also assume that there is possibility to get a small part of the original training data [4] of the ranking model. The adversary could train a ranking imitation model to transparentize the target ranking model with a series of queries q_M, ranking results P_M, and possible original training data, where M is number of query. Based on the findings of Pseudo Relevance Feedback in information retrieval [9], the ranking imitation model is trained with weak supervision manner to imitate the target ranking model. Also, the ranking imitation model generated adversarial trigger can be transferred to the victim model via gradient-based search [52].

Target Attack Model. We select BERT-base [10, 35] and MiniLM-L-12 [56] fine-tuned on passage ranking dataset as representative passage ranking models to study for our attack. Fine-tuning pre-trained BERT that adopts query and passage concatenation architecture is the common way now and achieves previous SOTA in text ranking with huge performance leap [35]. As for the fine-tuned MiniLM, it is the cross-encoder architecture [31], which is another widely adopted architecture, and adopts MiniLM as encoder. It achieves highly ranked performance and also is previous SOTA.

4 METHODOLOGY

In this section, according to the formulated black-box ranking attack problem in section 3, we elaborate the proposed attack method.

4.1 Overview

The whole attack pipeline is depicted in Figure 1, which carries two key phases: black-box ranking model imitation phase and Pairwise Anchor-based Trigger (PAT) generation phase.

Algorithm 1: Imitation adversarial attacks for black-box text ranking models

Input: target black-box ranking model O, *Pairwise BERT* R, query collection Q, target query q, target passage p, anchor passage $\mathcal{A}$, language model g, vocabulary $\mathcal{V}$

Parameters: sample top-n size N, trigger length L, epochs E, coefficient ϵ, temperature τ, step size α, beam size B, beam top-n size U

Output: an adversarial trigger t

1 **Phase** 1. *Black-box Ranking Model Imitation*
2   INIT: Dataset $\mathcal{D} \leftarrow \{\}$
3   **for** $q_m \in Q$ **do**
4   rank list $P_m \leftarrow$ query O with q_m
5   **for** $p_i, p_j \in$ top-$N(P_m)$&$Rank(p_i) > Rank(p_j)$ **do**
6          $\mathcal{D} \leftarrow positive, [q_m; p_i; p_j]$
7          $\mathcal{D} \leftarrow negative, [q_m; p_j; p_i]$
8   // Reverse p_i and p_j to get the negative triple.
9   Train the ranking imitation model R on $\mathcal{D}$ using Eq 3.
10   **return** R
11 **Phase** 2. *Pairwise Anchor-based Trigger Generation*
12   INIT: $\Theta = [\theta_1, \ldots, \theta_L]$, where $\theta_l \in \mathbb{R}^{|\mathcal{V}|}$ & $\sim \mathcal{N}(0, 1)$
13   Top-n similar words and anchor passage words $\mathcal{V}_{sim}$
14   **Function** LogitsPerturbation($\ell, t_{1:l-1}$):
15       $\rho \leftarrow 0 \in \mathbb{R}^{|\mathcal{V}|}$ &
16       $w + \epsilon$ for w in ρ if $w \in \mathcal{V}_{sim}$
17   **for** epoch$\leftarrow$ 1 to E **do**
18          $\tilde{t}_l \leftarrow$ softmax$((\ell + \rho_l)/\tau)$
19          $\rho \leftarrow \rho - \alpha \cdot \nabla_\rho \mathcal{L}_R(q, t_{1:l-1}; \tilde{t}_l, \mathcal{A})$
20   **return** $\theta = \ell + \rho$
21    $\mathcal{B} \leftarrow B$ replicates of <BOS> token
22   **for** $l \leftarrow$ 1 to L **do**
23   Beam score matrix $S_l \leftarrow 0 \in \mathcal{R}^{B \times U}$
24   **for** each beam $t_{1:l-1} \in \mathcal{B}$ **do**
25          $\ell_l \leftarrow$ next tken logits from LM $g(t_{1:l-1})$
26          $\theta_l \leftarrow$ LogitsPerturbation($\ell_l, t_{1:l-1}$)
27   **for** $w \in$ top-U of θ_l **do**
28             $S_l[t_{1:l-1}, w] \leftarrow$ objective score from Eq 6.
29          $\mathcal{B} \leftarrow \{t_{1:l-1}; w | (t_{1:l-1}, w) \in$ top-B of $S_l\}$
30   **return** $t = argmax\ \mathcal{B}$

In *Phase 1*, to enable the adversary to train a ranking imitation model, which can mimic the ranking list of the black-box victim ranking model without a real labeled dataset, we utilize the relative relevance information among the ranking result list of the victim ranking model to construct a synthetic dataset. Specifically, we first sample triples from the ranking result list by querying the black-box victim ranking model. The triple is composed by one query, one relative positive candidate, and one relative negative candidate, where the relative positive candidate is ranked ahead of relative negative candidate. Then we can train an ranking imitation model, which adopts pairwise encoder architecture, i.e., pairwise BERT, with those sampled triples.

In *Phase 2*, the adversary utilizes the ranking imitation model to craft adversarial triggers. By inserting the trigger into one irrelevant passage, the rank of the passage, produced by the target victim ranking model, should be boosted due to the transferability of the ranking imitation model. Moreover, we propose a Pairwise Anchor Trigger generation method to make full use of the pairwise structural information of the ranking imitation model.

Our proposed imitation adversarial attack for black-box neural text ranking attack is outlined in pseudo Algorithm 1.

4.2 Black-box Ranking Model Imitation

Practically, the adversary has no knowledge of the target model architecture, score function, and ground truth label information. Nevertheless, the orientation of transparentization is to train a ranking imitation model that substitutes and achieves comparable performance to the victim model. Since we have query access to the ranking result list, which consists of ranks of candidates, based on the findings of Pseudo Relevance Feedback in information retrieval, we can train a ranking imitation model by sampling triples from the ranking result list to imitate the target ranking model. To enhance the transferability of adversarial triggers, the ranking imitation model should be functionally similar to the victim model [52]. In this work, we measure the model similarity by the similarity of ranked lists.

Suppose we get the rank list of K passages $P_m = [p_1, \ldots, p_K]$ with respect to the query $q_m \in Q = \{q_1, \ldots, q_M\}$ from the victim model, we sample triples $[q_m; p_i; p_j]$ from the top N candidate passages with label **1**, where p_i is ranked ahead of p_j ($i \neq j$) and $1 \leq i, j \leq N \leq K$. The sampled triples *(query, relative positive candidate, relative negative candidate)* carry *pseudo hard examples* and the relative relevance information among the top-N results of the victim model. Thus, by applying our sampling strategy, we can get the ranking model imitation dataset $\mathcal{D}$ and learn an imitation ranking model efficiently. Moreover, it does not require the specific relevance label or score of the document to the query, which is necessary to train a pointwise ranker. The ranking imitation model-inferred candidate relatedness could unveil black-box ranking model vulnerabilities.

We then train a ranking imitation model *Pairwise BERT* based on sampled triples to substitute the victim. In the Pairwise BERT ranking model, the query and a pair of passages are concatenated with [SEP] and [CLS] tokens. We utilize the concatenation paradigm as our base architecture, as it represents the current state-of-the-art in terms of rerank effectiveness. The scores are computed by a single linear layer $W \in \mathbb{R}^{768 \times 2}$. The whole process can be formulated as below:

$$s_i = \text{BERT}([\text{CLS}; q_m; \text{SEP}; p_i; \text{SEP}]) * W \quad (1)$$

$$s_j = \text{BERT}([\text{CLS}; q_m; \text{SEP}; p_j; \text{SEP}]) * W \quad (2)$$

where s_i and $s_j \in \mathbb{R}^2$ represent the positive scorer (a.k.a., s_{pos}) and negative scorer, respectively.

The Pairwise BERT actually train one BERT model applied on two concatenated sequences. The loss is computed as follow:

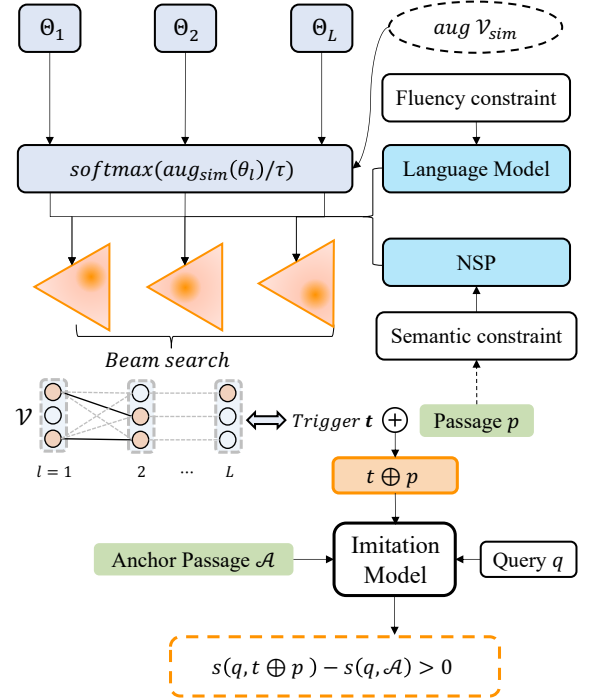$$\mathcal{L}_R(q_m, p_i, p_j) = -y_{m,i,j} \log(\text{softmax}(s_i - s_j)) \quad (3)$$

Figure 2: Overview of Pairwise Anchor-based Trigger (PAT) generation.

where $y_{m,i,j} \in \mathbb{R}^2$ denotes the one-hot label of $[q_m; p_i; p_j]$. To balance the label distribution, we get the negative label of the triples by exchanging the positive passage and the negative passage. We use backpropagation to compute the parameter gradients with Adam [24] optimizer. **During inference, we get the last dimension of s_i as the score of the passage with respect to the query.**

4.3 Pairwise Anchor-based Trigger (PAT) Generation

With *Pairwise BERT* ranking imitation model, we can create adversarial triggers for black-box ranking models by leveraging the transferability of adversarial examples [37]. To make full use of the pairwise structural information, we propose a Pairwise Anchor-based Trigger (PAT) generation method (Figure 2). Given a target query sentence q, a target candidate passage p, and the top candidate passage $\mathcal{A}$ (anchor), we aim to generate a trigger t for the ranking imitation model with its pairwise loss function $\mathcal{L}_R$. The anchor is used to guide the trigger generation, i.e.,:

$$\min \mathcal{L}_R(q, t \oplus p, \mathcal{A}) \quad (4)$$

where $\oplus$ denotes injection, e.g., adding the triggers in front of the target passage. We use a gradient-based search to generate a fixed length trigger for the target input.

Inspired by the generation of adversarial semantic collision [46], we also first utilize gradient optimization with relaxation to find a soft continuous representation of a trigger. Then we combine beam search with score function of ranking imitation model to find the trigger. Label smoothing can be applied at each iteration

to optimize the objective. Formally, we have the vocabulary $\mathcal{V}$ of the model and the length L of the trigger. Each word of trigger at position l is represented as continuous logit vector $\theta_l \in \mathbb{R}^{|\mathcal{V}|}$. To enhance the attack efficiency, we apply an augmentation function to θ_l. Specifically, the logit value corresponding to similar words $\mathcal{V}_{sim}$ will be augmented by adding ϵ, denoted as aug_{sim}. Note that similar vocabulary consists of tokens from anchor passage and tokens selected by the ranking imitation model with the query and the model vocabulary. Then they are softly selected to input words:

$$\tilde{t}_l = \mathrm{softmax}(\mathrm{aug}_{sim}(\theta_l)/\tau) \tag{5}$$

where τ is a temperature scalar, which controls the sharpness of word selection probability. We can get the probability of each word the vocabulary $\mathcal{V}$ by applying the softmax function on the continuous variable θ_l.

To avoid generating nonsensical triggers with high perplexity, which can be easily filtered, we add the fluency constraint with a language model (LM) g. The semantic of the trigger that differs significantly from the passage is highly susceptible to be detected. To equip the trigger camouflages, we add the next sentence prediction (NSP) model. Specifically, we apply the sequential optimization to the trigger by LM decoding with the joint search on the ranking score s_{pos}, LM likehood, and next sentence prediction score f_{nsp}. At each time step l, we need to maximize:

$$\max_{w \in \mathcal{V}} s_{pos}\left(q, t_{1:l-1}; w\right) + \lambda_1 \cdot \log P_g\left(t_{1:l-1}; w\right)$$
$$+ \lambda_2 \cdot f_{nsp}\left(p, t_{1:l-1}; w\right) \tag{6}$$

where $t_{1:l-1}$ is the searched beam sequence before l. s_{pos} is the positive scorer of the ranking imitation model. f_{nsp} is the probability that $t_{1:l-1}; w$ is the continuation of p. λ_1 and λ_2 are the hyperparameters that controls the trigger generation constraint strength of the semantic consistency and fluency, respectively.

Inspired by the works of Dathathri et al. [8] and Song et al. [46], we combine the logits to take ranking similarity and semantic consistency into account. Suppose we get the next-token logits ℓ_l generated by LM g at step l, then we optimize from the current context to find an update that favors tokens maximizing the ranking score and semantic consistency. Let $\theta_l = \ell_l + \rho_l$, where $\rho_l \in \mathbb{R}^{\mathcal{V}}$ is a perturbation variable by sampling from $\mathcal{N}(0, 1)$. Then we optimize the relaxed pairwise ranking loss objective $\min_{\rho_l} \mathcal{L}_R\left(q, t_{1:l-1}; \tilde{t}_l, \mathcal{A}\right)$ for several epoch steps, where $\tilde{t}_l$ is the relaxed soft word from equation 5. We minimize the pairwise ranking loss stochastically using Adam [24] optimizer. This operation searches the next-token prediction distribution combining with the perturbed logits to favor words that are likely to boost the rank of a passage.

After perturbation at each time step l, we can get the top-n candidate words in $\tilde{t}_l$. With the subset of tokens w, we can calculate the ranking score $s_{pos}\left(q, t_{1:l-1}; w\right)$ and the semantic consistency score $f_{nsp}\left(p, t_{1:l-1}; w\right)$. Given the current beam context, these words w are likely under the LM prediction. Each top-n word in a beam path is assign a score as in equation 6. We update the beams with the top-scored words. Since we adopt the hard word at each optimization step, the method can generate a natural-looking ranking attach trigger with semantic consistency.

5 IMITATING BLACK-BOX RANKING MODELS

In this section, we present the imitation experiment results with multiple datasets. For imitations and attacks, we implement our approach with PyTorch [4] and the HuggingFace Transformers library [58].

5.1 Datasets

We mainly adopt three passage ranking datasets to study the performance on different domains:

MSMARCO DEV. The MAchine Reading COmprehension (MS-MARCO) dataset [34] is based on sampled real users' Bing queries. The corpus is initially constructed by retrieving the top-10 passages from the Bing search engine and then annotated. Relevance labels are sparsely-judged and derived from what passages are marked as having the answer to the query. The full training set contains approximately 400M tuples of a query, relevant and non-relevant passages. The development set (MSMARCO DEV) of passage reranking contains 6,980 queries, each paired with the top 1,000 passages retrieved with BM25 from the MSMARCO corpus.

TREC DL 2019. The passage ranking task of TREC Deep Learning Track 2019 (TREC DL 2019) dataset [6] is similar to MSMARCO. But the official evaluation set provides 200 queries, 43 of which are manually and densely judged by NIST assessors with graded relevance labels. For queries with many relevant passages, additional passages were judged as a graded relevance label in {*Perfectly Relevant (3), Highly Relevant (2), or , Related (1), Irrelevant (0)*} [6]. Note that, we utilize the officially labeled 43 queries to evaluate the ranking model imitation performance, to keep the same with previous works adopting TREC DL 2019 dataset. As for evaluating the performance of ranking attack, we utilize the random sampled 100 queries and select target passages ranked by the victim model.

NQ. The Natural Question (NQ) dataset is created by Kwiatkowski et al. [26] for open-domain QA. The original dataset contains 300K questions collected from Google search logs. We adopt the new version of NQ [23] further selected and processed the Wikipedia articles as the collection of passages. This version of NQ dataset contains more than 60K questions, 21M passages, and corresponding negative samples. The NQ dataset is adopted as out-of-domain dataset to firstly train a decent passage ranking model, which is then used to further imitate target victim ranking model.

TREC MB 2014. The TREC Microblog 2014 (TREC MB 2014) dataset [30] uses tweets collection to evaluate the model for social media search. It contains 50 queries. Following the experimental procedure of prior works [30, 42], we evaluate models in a reranking task, using the top-1000 tweets preprocessed by Rao et al. [42][5].

We sample imitation data from the results on MSMARCO DEV of publicly available models. Models are also tested on TREC MB 2014, which is an *out-of-domain* dataset for these models, to demonstrate the *zero-shot ranking* performance. We also test the ranking performance and functionality similarity (inter agreement) between the victim and ranking imitation models on TREC DL 2019.

---

[4]https://pytorch.org/
[5]https://github.com/jinfengr/neural-tweet-search

Note that the ground truth relevance labels of the open-source dataset are manually assigned, while the training data for our pair-wise ranking imitation model were labeled through random sampling from the ranking list of the target model, which does not require white-box access to the target and manual work.

5.2 Baselines and Target Models

We compare our ranking imitation models with the following most widely used and state-of-the-art text ranking models.

1) **BM25** [44]: It is a classic exact lexical match algorithm with high efficiency that is widely adopted for the first stage retrieval (recall). We adopt the tuned parameters and results from Anserini[6].

2) **TK** [20]: The Transformer-Kernel (TK) model is not based on BERT pre-training, but rather uses shallow Transformers and pre-trained word embeddings. We directly copy the reported numbers from Hofstätter et al. [19].

3) **BERT-Base** and **BERT-Large** [35]: It is the common way of utilizing the pre-trained BERT [10] model in reranking scenario that concatenates query and passage input sequences and achieves previous state-of-the-art in text ranking with huge performance leap. We utilize the fine-tuned publicly available BERT-Base and BERT-Large for passage ranking task from NBoost[7].

4) **DistilBERT$_{CAT}$** [19]: It is a DistilBERT concatenation model trained with Margin-MSE loss that uses vanilla, mono, and concatenated DistilBERT rerankers as teachers and ensembled on MS-MARCO passage dataset. We adopt the fine-tuned model published by Hofstätter et al. [19][8].

5) **MiniLM** [56]: This model utilizes the fine-tuned MiniLM as encoder and is the cross-encoder architecture overall [31]. We adopt the MiniLM-L-12 version, which achieves highly ranked performance on MSMARCO and TREC DL 2019[9].

6) **Pairwise BERT (ours)**: Model details have been explained in Section 4. Practically in this paper, we adopt the BERT-base-uncased model as the encoder and the pairwise pattern to train the ranking imitation model.

5.3 Research Questions and Experimental Settings

In practice, the adversary will not know the architecture or original training dataset of the victim ranking model. In this paper, we study the effect of the following variables on ranking model imitation:

- *RQ1: How the sampling strategy affects the ranking model imitation performance?* Ranking imitation model training triples $\mathcal{D}$ are constructed by the query, top-N candidates and other candidates pairs randomly selected from the ranking results of MiniLM on MSMARCO DEV, where $N \in \{15, 20, 25, 29\}$. For fair comparison, we restrict the total numbers of sampled triples as the same scale. Specifically, we explore two scales, which are approximately 0.4K (top-15 + other 19 candidates; top-20 + other 10 candidates; top-25 + other 4 candidates; top-29 candidates) and 1K (top-15 + other 59 candidates; top-20 + other 40 candidates; top-25 + other 28 candidates; top-29 + other 20 candidates) sampled pairs per query,

respectively. The sampled pairs are mainly from top-K results. However, for each query, top-K may not contain enough hard pairs, e.g., C_2^{20} is smaller than 0.4K. To keep the same number of total sampled instances, we also sample some passages not in the top-K results as $\boldsymbol{p}_j$. The relative relevance (in Section 4.2) between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ still holds.

- *RQ2: Can pre-trained Pairwise BERT model improve the performance of ranking model imitation?* Given a specific sampled triples $\mathcal{D}$ for training the ranking imitation model, we use different datasets, which are MSMARCO (In-Domain, ID), NQ (Out-Of-Domain, OOD), and Null dataset (Zero), to train the Pairwise-BERT firstly to get a decent ranker. Specifically, the MSMARCO (ID) used in ranking model imitation task is a small random sampled MSMARCO training dataset (2.56M), while other victim models use triples ranging from 12.8M to 160M. Both of the rankers firstly trained on ID and OOD datasets are tuned with the sampled triples $\mathcal{D}$ to get the final ranking imitation models. Whereas the *Zero Imitate* imitates other victims by straightly training on sampled victims' triples without any pre-training process.

- *RQ3: How do the differences in architecture, hyperparameters and pre-trained corpus between the ranking imitation model and the victim model affect the imitation performance?* To construct a calibration value of ranking model imitation effectiveness, we firstly train the *Pairwise BERT(ID)* in Table 2 and use the *all same* Pairwise BERT, i.e., *Imitate ID's Triples*, to imitate *Pairwise BERT(ID)* based on the predicted triples of it. Then we choose *MiniLM* and *BERT-Large* as victim ranking models to study the effect of different and similar architectures on ranking model imitation, respectively. Note that BERT-Large is trained on MS-MARCO training dataset and utilizes *similar* but *mismatched* architectures and parameters. Whereas MiniLM utilizes *different* pre-train data, architecture and parameters[10].

Note that we do not adopt other models based on term-frequency or word2vec, because their ranking results are not as effective as the transformer-based contextualized models. They may not be as widely used as the deep transformer-based contextualized models in reranking scenario, facing relatively small threats. Since our method is mainly based on ranking results, it can be easily generalized to other types of ranking models. We will further explore this problem in the future.

For evaluating the ranking performance on MSMARCO DEV and TREC DL 2019, we adopt *MRR@10* and *NDCG@10*, which are commonly used in information retrieval [6, 19, 35]. The Reciprocal Rank (RR) information retrieval measure calculates the reciprocal of the rank at which the first relevant document was retrieved. RR is 1 if a relevant document was retrieved at rank 1. If not, it is 0.5. When averaged across queries, the measure is called the Mean Reciprocal Rank (MRR) [5]. Discounted Cumulated Gain (DCG) assumes that, for a searcher, highly relevant documents are more valuable than marginally relevant documents. It further assumes, that the greater the ranked position of a relevant document is, in terms of any relevance grade, the less valuable it is for the searcher,

---

Table 1: Ranking performance (%) of imitating MiniLM on MSMARCO DEV and TREC DL 2019 datasets using different triple sampling strategies. Note that the performance of MiniLM is irrelevant to the number of sampled pairs per query. We add it to the first row of the table to facilitate a comparison of imitation performance. *Bold* shows the best imitation performance of a model under different sampling strategies. "-" means not applicable.

| Model | Sampling Strategy | ≈0.4K sampled pairs per query | | | | | | ≈1K sampled pairs per query | | | | | |
| | | MSMARCO | | TREC DL 2019 | | | | MSMARCO | | TREC DL 2019 | | | |
		MRR@10	NDCG@10	MRR@10	NDCG@10	Inter@10	RBO@1K	MRR@10	NDCG@10	MRR@10	NDCG@10	Inter@10	RBO@1K
MiniLM (V2)	*Target*	*39.7*	*45.6*	*90.1*	*74.3*	*-*	*-*	*39.7*	*45.6*	*90.1*	*74.3*	*-*	*-*
Zero Imitate V2	Top15+others	37.1	43.3	85.3	67.8	71.9	60.2	35.7	41.9	83.6	68.4	74.1	57.4
	Top20+others	37.4	43.7	88.0	69.0	72.8	64.6	36.1	42.3	87.4	69.0	74.4	57.0
	Top25+others	37.3	43.7	88.4	71.1	72.3	60.2	36.9	43.2	86.2	70.8	75.2	62.7
	Top29+others	37.9	44.2	89.9	71.5	73.0	61.4	37.1	43.5	86.8	70.4	75.8	63.2
MSMARCO (ID) ↓ Imitate V2	Top15+others	38.0	44.1	86.2	68.9	71.4	60.1	37.3	43.5	85.3	70.1	74.8	58.1
	Top20+others	**38.4**	**44.5**	**88.4**	**71.9**	**77.2**	**66.4**	36.5	42.7	87.6	70.6	76.5	63.2
	Top25+others	38.3	44.5	88.4	71.1	77.0	65.3	37.3	43.6	86.8	71.6	77.2	63.4
	Top29+others	38.3	44.5	88.3	71.9	76.5	64.0	37.2	43.5	89.5	71.4	75.8	62.3
NQ (OOD) ↓ Imitate V2	Top15+others	37.4	43.6	88.0	69.5	76.0	60.4	36.0	42.1	88.3	70.5	74.4	57.5
	Top20+others	37.9	44.1	85.9	70.4	76.0	63.0	37.3	43.6	88.1	71.0	76.7	61.5
	Top25+others	**38.3**	**44.5**	**89.9**	**71.3**	**77.0**	**66.1**	36.7	42.9	88.5	69.7	74.9	60.4
	Top29+others	38.0	44.2	86.6	72.3	74.0	64.0	37.0	43.3	88.8	71.7	76.5	60.7

because the less likely it is that the searcher will ever examine the document and at least has to pay more effort to find it. NDCG is its normalized form that the actual DCG performance for a query is divided by the ideal DCG performance for the same topic, based on the recall base of the topic in a test collection [5]. Following prior works [19] about TREC DL Track evaluation, as for MRR, we use a binarization point of 2 for TREC graded relevance labels. We adopt *AP* and *P@30* following [61], to evaluate the zero-shot ranking performance on TREC MB 2014.

Moreover, we also measure the inter ranking similarity by counting the overlap of the top 10 (*Inter@10*) and Rank Biased Overlap [57] (*RBO@1K*, we set the weighting parameter p to 0.7) between the ranking imitation model and the victim model. *Inter@10* measures the inter ranking similarity by counting the overlap of the top 10 candidates. *RBO@1K* is the Rank Biased Overlap, which calculates the top 1000 weighted ranking overlap between the ranking imitation model and the victim model. The *Inter@10* and *RBO@1K* of *Pairwise BERT (ID)* and all ** Imitate V1* are evaluated with *V1*. The *Inter@10* and *RBO@1K* of all ** Imitate V2* is evaluated with *V2*.

We limit the query-passage pair length to 256 tokens and train our models with a learning rate of $\{1, 3, 5, 7\} \times 10^{-6}$ and a batch size of 256. All the method run on a server configured with 8×32G Tesla V100 and 256G memory.

5.4 Experimental Results.

For RQ1, from the results of a comparative study for different sampling strategies in Table 1, we can observe that *0.4K* performs relatively better than *1K*. We speculate that *1K* contains more simple triples than *0.4K*, which leads to overfitting of the ranking imitation model and weakens the benefit of hard instances [40]. It needs more training efforts and tricks to overcome this problem. Moreover, the adversary prefers *0.4K* because it requires fewer results accesses and is more efficient than *1K*. We leave a thorough investigation to future work. Among sampling strategies in *0.4K*, *top-25 + others* achieves the closest performance to the target model.

For RQ2, the results in Table 1 also demonstrate the positive effect of the pre-training process of the Pairwise BERT model. The ranking imitation model, i.e., Pairwise BERT, which is firstly pre-trained on MSMARCO (ID) or NQ (OOD) then tuned on $\mathcal{D}$, performs better than the ranking imitation model (Zero Imitate V2). Although Carlini et al. [4] show that it is possible to extract the training data of the language model (GPT-2), extracting the training data of the BERT-based model or even unknown pre-train model is still hard to complete, unless there is a data leak. **Thus, it is more practical to adopt OOD data (NQ) rather than ID data (a subset of MSMARCO) for the adversary to pre-train the ranking imitation model**. Moreover, *OOD→Imitate V2* achieves competitive performance compared to *ID→Imitate V2* in terms of the ranking performance and ranking similarity.

As a result, top-25 candidates and randomly selected another 4 candidates (*top-25 + others*) are finally adopted to construct the sampled victims' triples $\mathcal{D}$ (approximately 2.7M). Together with the Pairwise BERT pre-trained on NQ, they are then used to closely imitate the target model.

Then we conduct more comparative imitation ranking experiments on more ranking models and datasets. From Table 2, we can observe that:

- For MSMARCO DEV dataset, ranking imitation models outperform the same architecture (BERT-Base) model, which are trained with data sampled from the ranking list of V1. Also, models imitating V2 achieve approaching or better performance compared with most of baselines. It indicates that, by sampling the ranking results of the victim with better performance, we can train a ranking imitation model with competitive ranking effectiveness results. We suspect it is due to the contribution of knowledge distillation [19].

- Ranking imitation models based on fine-tuned Pairwise-BERT outperform imitation models only trained on sampled triples, which is also observed in Table 2. This indicates that a good start point will benefit the imitation of ranking performance and result

Table 2: Ranking imitation results of performance (%) comparison with baseline models on MSMARCO DEV, TREC DL 2019, and TREC MB 2014 datasets. *Bold* shows the best ranking imitation performance in terms of *Inter@10* and *RBO@1K*. "-" means not applicable.

Model	MSMARCO DEV		TREC DL 2019				TREC MB 2014	
	MRR@10	NDCG@10	MRR@10	NDCG@10	Inter@10	RBO@1K	AP	P@30
BM25	18.7	23.4	68.5	49.7	-	-	41.4	62.6
TK	33.1	38.4	75.1	65.2	-	-	-	-
BERT-Base	35.2	41.5	87.1	71.0	-	-	45.4	68.0
BERT-Large (V1)	37.1	43.3	85.5	72.6	-	-	44.9	67.4
DistilBERT$_{CAT}$	38.2	44.2	88.9	72.8	-	-	44.6	66.5
MiniLM (V2)	39.7	45.6	90.1	74.3	-	-	47.5	70.9
Pairwise BERT (ID)	34.4	40.5	88.4	71.4	75.3	64.4	41.9	65.2
Imitate ID's Triples	32.6	39.0	87.7	70.7	77.7	84.7	41.4	65.2
Zero Imitate V1	35.7	41.7	85.7	67.0	66.1	58.6	39.5	63.1
ID → Imitate V1	36.2	42.4	86.8	70.2	**72.1**	**65.2**	45.0	67.5
OOD → Imitate V1	36.2	42.5	84.1	69.9	71.3	63.0	45.0	67.5
Zero Imitate V2	37.4	43.7	88.0	69.0	70.9	61.3	41.3	65.1
ID → Imitate V2	38.4	44.5	88.4	71.9	**77.2**	**66.4**	45.9	68.4
OOD → Imitate V2	38.3	44.5	89.9	71.3	77.0	66.1	45.2	68.2

Table 3: Cross ranking similarity results (%) on OOD dataset for validating the effectiveness of ranking imitation on its target model. *Bold* shows the best ranking imitation performance.

Imitation\Victim	V1		V2	
	Inter@10	RBO@1K	Inter@10	RBO@1K
OOD→Imitate V1	**71.3**	**63.0**	68.4	58.2
OOD→Imitate V2	68.6	61.3	**77.0**	**66.1**

similarity, similar to the findings of Krishna et al. [25] on text classification model extraction.

- As a reference for model similarity, two Pairwise-BERT models trained with same hyperparameters but different random seeds achieve 0.753 Inter@10 and 0.644 RBO@1K. The Pairwise-BERT, trained with same triples labelled by ID (*Imitate ID's Triples*), achieves 0.777 Inter@10 and 0.847 RBO@1K. Thus, we can derive the answer for RQ3 that ranking imitation models with the same architecture, hyperparameters and direct training corpus knowledge (by pseudo triples), especially for the *ID→ Imitate V2*, perform similar with the victim both in ranking effectiveness and agreement. However, from the last six rows of Table 2 and the cross ranking similarities in Table 3, when there is no knowledge of training data, the performance of the victim ranking model has more impact on ranking imitation agreement than the similarity of architecture and hyperparameters.

6 RANKING ATTACK EXPERIMENTS

In this section, we report the result of ranking attack experiments based on the ranking imitation models and the proposed Pairwise Anchor-based Trigger generation method.

6.1 Experimental Settings

Baseline. We compare the following adversarial attack methods, which are originally proposed for text matching or text classification task. Note that in this paper, we mainly investigate those methods based on trigger injection rather than modifying the original passage words or tokens. We modify them for our tasks by keeping the core components:

(1) **Collisions** [46]. It utilizes gradient-based approaches for generating three types of semantic collisions that are semantically unrelated but judged as relevant by NLP models. As for generating collisions, it does not take the contextual information of the document into consideration. The collision is designed for the pointwise document relevance score model (Birch) [61], we cannot apply the generation method to our ranking imitation model directly. Thus, to compare with this method on the passage ranking task, the same training set from MSMARCO for Pairwise BERT (ID) (2.56M) is adopted to fine-tune a pointwise BERT ranking model, which achieves 87.0 of MRR@10 on TREC DL 2019. Collisions are generated based on this model then transferred to target V1 and V2 model. We adapt the open-sourced code[11] to passage reranking by keeping the core component of collisions unchanged. We abbreviate the aggressive collision baseline as C_{aggr}, regularized aggressive collision baseline as C_{reg}, and natural collision baseline as C_{nat}.

(2) **HotFlip** [11]. It is an universal text attacking method that approximates the effect of replacing a token utilizing the gradient of the target model. Following prior works [46, 51, 52] and open-sourced code[12] based on HotFlip attacks, we first initialize the trigger with a sequence of repeating words "the", and then iteratively replace all trigger t words. We replace an input token at position l with the token whose embedding minimizes the first-order Taylor

---

[11]https://github.com/csong27/collision-bert
[12]https://github.com/Eric-Wallace/universal-triggers

Table 4: White-box attack results for ranking imitation models on TREC DL 2019 dataset. r is the rank of candidates after adding the trigger in front of it.

Target	Method	% $r \leq 20$	% $r \leq 50$	% $r \leq 100$	% $r \leq 500$	ASR	*avg. **Boost***
Pointwise BERT	Query+	68.4	81.8	89.8	99.0	100.0	944.7
	C_{aggr}	0.2	3.1	13.4	77.7	100.0	658.2
	C_{reg}	0.2	3.2	10.8	74.3	100.0	634.1
	C_{nat}	0.0	1.4	9.1	70.9	100.0	608.3
Imitate V1	Query+	84.4	91.7	98.5	100.0	100.0	984.9
	HotFlip	0.0	0.0	0.5	12.7	68.8	173.3
	PAT	6.6	14.2	22.0	48.8	96.2	471.5
	↪greedy	5.6	19.2	25.6	49.6	100.0	457.8
	w/o LM	9.0	18.2	26.8	56.2	98.8	530.6
	w/o NSP	9.6	20.2	28.8	64.4	99.2	601.4
	w/o *Cons.*	10.6	20.2	30.0	65.8	100.0	605.6
Imitate V2	Query+	79.0	89.0	93.8	99.0	100.0	958.0
	HotFlip	0.0	0.2	0.6	13.4	69.2	150.9
	PAT	9.4	15.6	25.8	51.4	96.0	485.7
	↪greedy	7.2	12.8	15.2	38.4	95.2	329.3
	w/o LM	13.6	23.4	34.2	64.2	98.8	603.6
	w/o NSP	17.6	30.6	41.2	73.8	100.0	680.3
	w/o *Cons.*	22.4	32.4	39.6	70.0	98.8	651.1

approximation of ranking imitation model loss $\mathcal{L}_R$:

$$\operatorname*{argmin}_{e'_l \in \mathcal{V}} (e'_l - e_l)^\top \nabla_{e_l} \mathcal{L}_R(q, t \oplus p, \mathcal{A}) \tag{7}$$

where $\mathcal{V}$ is the model's vocabulary and $\nabla_{e_l} \mathcal{L}_R$ is the gradient of $\mathcal{L}_R$ with respect to the input embedding e for the token at position l. Considering the imperfection of local first-order approximation, at each iteration, we try all positions l with top 50 tokens [52] from the formula 7 and choose the replacement with the lowest loss.

Intuitively, inserting the query into the target passage (denotes as **Query+**) is a straightforward way to boost the rank. Query+ type of attacks usually use the keyword stuffing for spamming. We choose it as one of the baselines and discuss its attack performance and imperceptibility in the following experiments and discussion.

We also test the impact of the components by removing the LM and NSP constraints of PAT (denote as **w/o LM**, **w/o NSP**, and **w/o Cons.** respectively) and replacing the beam search with greedy search (denotes as **greedy**).

Settings. We choose the ranking imitation models, *OOD→Imitate V1* and *OOD→Imitate V2*, to generate adversarial triggers using TREC DL 2019 dataset. For simplicity, we denote them as *Imitate V1* and *Imitate V2*, respectively. *BERT-Large (V1)* and *MiniLM-L-12 (V2)* are victims which are adopted to evaluate the effectiveness of these triggers. We attack the target model by adding 6-words triggers into the irrelevant passages, with the hyperparameters of 0.1 and 0.8 for fluency constraint λ_1 and semantic constraint λ_2, respectively. Top-200 similar words, selected by the ranking imitation model with the query and vocabulary, are augmented by setting ϵ to 0.68. The beam search size and temperature are set to 10 and 1.0, respectively. The semantic consistency and fluency described in subsection 4.3 utilize pre-trained BERT NSP model and BERT LM model without fine-tuning on any datasets used in this paper.

Top-3 candidates are concatenated as the anchor passage to guide the trigger generation.

For each query, the most irrelevant 5 passages are selected from top-1K that are ranked by the victim models. Then the triggers are inserted at the head of these passages to boost their ranks. We measure the attack results by the rate of successfully boosted candidates (**ASR**), average boosted ranks (*avg.**Boost***), and the percentage of the target candidates shifted into top-20 (% $r \leq 20$), top-50 (% $r \leq 50$), top-100 (% $r \leq 100$), and top-500 (% $r \leq 500$).

6.2 Attack Results

We report the white-box attack results on imitation models in Table 4. For our imitation models, triggers are able to disorder the candidates and boost their ranks with high success rate and significant boosted rank rate. A small subset of irrelevant candidates can be boosted in the top-20, top-50, and top-100 after inserting collisions. Three types of collisions always beat the HotFlip triggers on different metrics. Aggressive collisions perform better than regularized collisions and natural collisions perform worst among collision baselines. The trend is consistent with the observation from Song et al. [46]. Adversarial semantic collisions in passage ranking are not as effective as in document ranking. We speculate that the passage ranking model calculates the relevance score by encoding the whole passage into a contextual representation, while the document ranking model [61] focuses on key sentences and applies interpolation to calculate the overall relevance, which makes the attack easier.

For our ranking imitation models, triggers are able to disorder the candidates and boost their ranks with high success rate. Triggers generated based on ranking imitation models can boost nearly

Table 5: Transfer attack results on TREC DL 2019 dataset. *Path* denotes that triggers are generated from the first model and transferred to the second (target) model. *r* is the rank of candidate passage after adding the trigger in front of it.

Method	Path	% $r \leq 20$	% $r \leq 50$	% $r \leq 100$	% $r \leq 500$	ASR	avg. *Boost*	Path	% $r \leq 20$	% $r \leq 50$	% $r \leq 100$	% $r \leq 500$	ASR	avg. *Boost*
Query+	Pointwise	74.0	83.8	91.2	98.8	100.0	949.8	Pointwise	68.4	81.8	89.8	99.0	100.0	944.7
C_{aggr}	BERT	0.4	1.6	1.8	11.8	73.0	116.5	BERT	0.0	0.0	0.0	3.2	76.0	39.7
C_{reg}	↓	0.0	0.0	0.4	5.6	65.4	58.6	↓	0.0	0.4	0.8	3.2	68.2	31.0
C_{nat}	V1	3.6	6.4	8.2	30.4	94.2	298.0	V2	1.4	2.8	3.8	12.8	94.6	154.0
HotFlip	Imitate V2	0.0	0.0	0.0	2.8	88.4	39.1	Imitate V1	0.0	0.0	0.0	1.0	74.6	10.9
PAT	→V1	7.0	13.2	19.4	36.0	47.2	316.2	→V2	0.6	3.8	6.0	26.6	92.8	268.5
HotFlip		0.0	0.0	0.2	3.8	87.4	42.6		0.0	0.0	0.0	1.8	75.2	12.0
PAT	Imitate V1	7.2	14.4	21.6	44.6	93.8	417.3	Imitate V2	4.6	10.2	15.2	39.2	92.2	373.4
w/o LM	↓	15.6	22.2	31.2	51.6	66.2	467.0	↓	4.8	11.6	20.4	50.6	97.0	471.4
w/o NSP	V1	15.0	24.2	32.2	54.0	96.6	514.9	V2	11.0	22.0	29.4	61.6	97.8	569.1
w/o *Cons.*		25.2	33.6	40.0	62.6	95.4	585.4		17.2	27.8	33.4	59.6	97.6	557.1

20% to 40% irrelevant passages into the top-100 ranked by themselves. The *Imitate V1* gets lower attack performance compared with *Imitate V2*.

The PAT without constraints achieves better attack performance for ranking imitation models and victim models. On *Imitate V2*, the full version PAT and the PAT without constraints achieve over 90% success; triggers generated on PAT without constraints have higher ratio of successfully boosted candidates. The attack success rate of full version PAT is affected by the constraints. We will analyze the strengths of the constraints in the next subsection.

Moreover, as shown in Table 5, the adversarial ranking attack triggers generated from ranking imitation models with the PAT successfully transfer to victim models. Attacks transfer at a reasonable rate, e.g., the triggers generated on *Imitate V2* with the PAT can transfer to V2 and boost 20.4% irrelevant passages into top-100. For the cross transfer attack (*Imitate V1 → V2* and *Imitate V2 → V1*), the effectiveness of triggers is weakened. It is mainly because their ranking imitation models extract different knowledge that the transferability of triggers across different imitation target models will be affected.

The performance of adversarial semantic collision is significantly weakened when we transfer collisions from *Pointwise BERT* to victim models. It demonstrates that our ranking imitation models are needed for transferable adversarial ranking attacks.

We also utilize the *Pairwise BERT (ID)* to generate triggers, which is only trained on the MSMARCO training set. The triggers generated with PAT against V2 achieve 2.4% at % $r \leq 50$. It is not as competent as our ranking imitation model, which shows the ranking imitation models considerably enhance the transferability of adversarial triggers.

Although Query+ performs better than PAT in most of cases, it can be detected by anti-SEO solutions [64]. Whereas PAT can find camouflaged triggers to manipulate embedding-based ranking, by reducing pairwise loss under semantic consistency and fluency constraints. Our evaluations using spamicity detection [64] (refer to the following mitigation analysis by automatic spamicity detection) and human annotation show that the texts generated by Query+ can be easily identified due to the semantic gap between added content and the original passages (also observed by Wu et al. [59]),
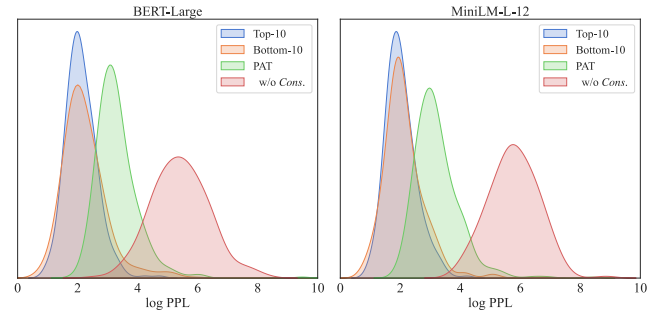
Figure 3: Distributions of log perplexity (PPL) calculated by GPT-2 on TREC DL 2019 passages and triggers.

while those produced by PAT are much stealthier. Also, we believe that Query+ will become less effective in ranking manipulation on real-world search engines using DL-based ranking models trained on a large number of query logs.

6.3 Mitigation Analysis

Mitigating by perplexity. Figure 3 shows log perplexity distributions, evaluated by GPT-2 [41], on BERT-Large and MiniLM-L-12 for top-10 real passages, bottom-10 real passages, triggers generated by PAT, triggers generated by PAT without LM and NSP constrains (w/o *Cons.*). Note that these triggers are generated based on *Imitate V1* and *Imitate V2* for BERT-Large and MiniLM-L-12, respectively. Since triggers are synthetic, their perplexity is higher than human-generated texts. We can observe a significant gap between the distributions of real passages and triggers generated by PAT w/o *Cons.*. Triggers generated by PAT significantly reduces the perplexity of triggers and have more overlap with the perplexity distribution of real text than triggers generated by PAT w/o *Cons.*. Song et al. [46] introduced perplexity-based filtering to mitigate the triggers by setting a threshold on language model (LM) perplexity. However, as shown in Figure 3, there is a significant distribution overlap between top and bottom passages with PAT triggers. Thresholds that are too strict will result in a higher false positive rate. If the

Table 6: Automatic spamicity detection rates (%) of Query+ and PAT by tf-idf feature based method.

Threshold	0.30	0.20	0.15	0.10
Query+	28	80	92	98
PAT	4	14	42	86

Table 7: Issues counts of triggers generated on ranking imitation model *OOD→Imitate V2* for MiniLM-L-12 by online grammar checkers.

Method	Chegg Writing	Grammarly
Gold	39	77
C_{aggr}	68	114
C_{reg}	100	137
C_{nat}	52	88
Query+	71	132
PAT	59	98
w/o LM	67	112
w/o NSP	57	104
w/o *Cons*.	72	118

Table 8: Human evaluation results about the trigger generated on ranking imitation model *OOD→Imitate V2* for MiniLM-L-12.

Method	Imperceptibility		Fluency	
	avg.	*Kappa*	avg.	*Kendall*
Query+	0.10	0.56	4.78	0.66
PAT	0.70	0.62	4.35	0.76
w/o LM	0.14	0.67	3.18	0.90
w/o NSP	0.14	0.67	3.62	0.90
w/o *Cons*.	0.12	0.63	2.68	0.85

we calculate the average number of errors in the combination of the trigger and the target passage using *Grammarly*[13] and *Chegg Writing*[14]. For *Grammarly*, we adopt the suggestions numbers about *correctness*, including spelling, grammar, and punctuation. For *Chegg Writing*, we issue numbers about grammar, punctuation, fluency, and typos. The issue count results are shown in Table 7. We can observe that, although there is still a gap about the naturalness between the original passage and the trigger, our proposed method and C_{nat} achieve better naturalness of attacks than other methods. Meanwhile, triggers generated by PAT achieve better attack effectiveness than C_{nat}.

Human evaluation. To further validate that our attacks are more natural and hard to mitigate than other baselines, we perform a human-subject evaluation about the imperceptibility and the fluency of the trigger. We recruit three annotators to annotate 50 random sampled trigger and the corresponding passage of each ranking attack model. Three annotators are Ph.D. students in information science. Each has over three years of experience in processing/annotating different NLP tasks (e.g., sentiment classification, text summarization, etc.), and received professional training. Note that we keep the random sampling seed consistent with all models. In terms of the *Fluency*, we apply the five-level Likert scale (1-5) to a single trigger. Higher score means the more fluency of the trigger. For *Imperceptibility*, annotators need to assign the 0 (*Perceptible*) or 1 (*Normal*) to the content that a trigger adding in front of the passage.

A summary of human-subject evaluation statistics, including the annotation consistency test results about *Kappa* value and *Kendall's Tau* coefficient are shown in Table 8. We can observe that the fluency of the triggers generated by the full version of PAT achieves comparable performance to the query, which demonstrates the effectiveness of the fluency and semantic consistency constraints in PAT. Although *Query+* achieves great fluency and attack performance, it can be easily detected by annotators due to the semantic inconsistency and irrelevance between the query and the passage. Triggers generated by PAT are more imperceptible than other baselines. As a result, human-subject evaluation demonstrates the effectiveness of the semantic consistency and fluency constraints to equip the trigger camouflages.

threshold of the filter is set loosely, more attacks will be missed. Thus, coarse-grained perplexity-based filtering cannot effectively mitigate the attack of our proposed PAT triggers. Moreover, the generated dataset can be used as additional data for increasing the robustness of neural model [27]. We can further fine-tune the target model using the adversarial triggers to make the model harder to attack. Since adversarial training requires additional optimization processes to make it effective, we will further explore adversarial training-based defense methods in future work.

Mitigating by automatic spamicity detection. Table 6 demonstrates the automatic spamicity detection experiment results of Query+ and PAT. The detection method [64] is mainly based on tf-idf feature and has been validated by the Microsoft adCenter. We randomly sample 50 triggers and the corresponding passages. The detection method can computer a spamicity score to the combination of the trigger and the target passage. The table header is the detection threshold. The last two rows are the detection rate (lower is better) under the specific threshold. The detection process can be defined as, given a spamicity threshold and a pair of a passage (may have inserted a trigger) with a query, it determines whether the spamicity score is greater than the threshold. If so, the passage is a suspect of spam content. We can observe that PAT performs significantly better than Query+. As the threshold decreases, the detector becomes stricter, and the detection rate of both increases, but this may also lead to more false positives.

Mitigating by automatic grammar checker. Holtzman et al. [21] find the degeneration property of the language model, which tends to get low perplexity to repeating tokens. As a result, some disfluent triggers with simply repeating tokens may escape from PPL-based filtering. We employ three online grammar checkers as defense metrics to measure the naturalness of triggers. Specifically,

---

[13]https://app.grammarly.com/
[14]https://writing.chegg.com/

Table 9: Attack performance of different trigger positions on MiniLM-L-12.

Method	Front		Middle		End	
	$\% \, r \leq 100$	ASR	$\% \, r \leq 100$	ASR	$\% \, r \leq 100$	ASR
Query+	81.8	100.0	64.8	100.0	61.8	100.0
C_{nat}	3.8	94.6	1.8	94.0	1.8	94.2
PAT	15.2	92.2	9.6	93.0	17.2	99.2
w/o LM	20.4	97.0	10.2	96.0	18.8	99.4
w/o NSP	29.4	97.8	19.8	97.6	18.4	98.4
w/o $Cons$	33.4	97.6	26.6	98.2	24.0	98.4

Query, Passage, and Trigger	r
Query: what type of eruption does pavolf have	-
Passage: This went on from 1754 to 1763. The costs of the French Indian War, also known as the Seven Years War (for its eruption into a worldwide conflict in 1756), led to the the passing of the Stamp Act, by Parliament in 1765.he costs of the French Indian War, also known as the Seven Years War (for its eruption into a worldwide conflict in 1756), led to the the passing of the Stamp Act, by Parliament in 1765.	998
Trigger (w/o Cons): this type known protovolf	14
Trigger (w/o LM): type known as torvolf	6
Trigger (w/o NSP): this type occurs two convolf	123
Trigger (full PAT): is known as torvolf	65

Figure 4: Trigger example generated from *OOD→Imitate V1* for the ranked 998th passage by the BERT-Large model with respect to the query. *r* is the rank of irrelevant passage after adding the trigger in front of it.

Position of the trigger injection. We try to insert triggers at different positions to see how they affect the attack performance. From the experimental results in Table 9, we can observe that triggers generated from *OOD→Imitate V2* with PAT then transferred to *V2* achieve 15.2%, 9.6%, and 17.2% ($\% \, r \leq 50$) when they are added at the *Front*, *Middle*, and *End* of the passages, respectively. For the other methods, inserting the trigger at the front of the passage usually gets the best performance. However, this may lead to the attack being easily detected. As we have discussed above, the semantic consistency constraint can mitigate this effect. In this paper, we mainly discuss inserting the trigger at the head of the passage. We will explore more flexible ways to insert triggers and how they will affect the imperceptibility of triggers and the overall semantic in future work.

Case. Figure 4 shows example triggers for the query (ID=912070) "what type of eruption does pavolf have" from the TREC DL 2019 test set, which are generated from *OOD → Imitate V2* for the 998th passage ranked by the MiniLM-L-12 model (*V2*). After inserting

the trigger, the passage can be ranked as the top candidate. We can also observe that the trigger generated with PAT can be more natural-looking and consistent with the passage content than the trigger generated without the fluency and consistency constraints.

7 CONCLUSIONS

In this study, we propose an imitation adversarial attack against black-box neural ranking models. The ranking imitation model is able to transparentize the victim ranker via effective pairwise learning. Then, we propose a novel adversarial trigger generator (for each candidate passage) by introducing the pairwise objective function with the anchor content. Based on the competitive ranking effectiveness and similarity between the ranking imitation model and the victim model, adversarial triggers are generated and transferred to the victim ranking model. The proposed model along with extensive experiment results reveal the vulnerability and risk of black-box text ranking systems.

In the future, we will investigate how to conduct more imperceptible and universal adversarial attacks against text ranking models and detect/defend those attacks effectively.

ADDRESSING POTENTIAL ETHICAL CONCERNS

The goal of our work is to help to make neural ranking models more robust. During performing our research, we used the ACM Ethical Code as a guide to minimize harm. Indeed, the techniques developed in this paper have potential for misuse in terms of attacking existing IR systems with triggers, which has potential negative short-term impacts. However, our intention is not to harm ranking models but instead to publicly release such unintended flaws so that novel defense algorithms can be developed to secure them in the future. This procedure is similar to how white hat hackers expose bugs or vulnerabilities in a software publicly. We have demonstrated that adversarial ranking attacks can be accomplished in black-box manner which reveals a greater threat than previous white-box attacks [46, 65]. This indicates our work **provides a long-term benefit** to the community and can help to improve IR systems. More importantly, we **minimized real-world harm** by not exposing any real-world failure or damage to any real users.

ACKNOWLEDGMENTS

REFERENCES

[1] Nima Asadi and Jimmy J. Lin. 2013. Effectiveness/efficiency tradeoffs for candidate generation in multi-stage retrieval architectures. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013.* ACM, 997–1000.

[2] Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.*

[3] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[4] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[5] Nick Craswell. 2009. *Mean Reciprocal Rank*. Springer US, Boston, MA, 1703.

[6] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *ArXiv preprint* abs/2003.07820 (2020).

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 985–988.

[8] 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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[9] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 65–74.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[11] 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 (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, 31–36.

[12] Wee Chung Gan and Hwee Tou Ng. 2019. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6065–6075.

[13] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3030–3042.

[14] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2018. Ranking Robustness Under Adversarial Document Manipulations. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 395–404.

[15] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. 2020. Ranking-Incentivized Quality Preserving Content Modification. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 259–268.

[16] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*. ACM, 55–64.

[17] Guoxiu He, Yangyang Kang, Zhuoren Jiang, Jiawei Liu, Changlong Sun, Xiaozhong Liu, and Wei Lu. 2020. Creating a Children-Friendly Reading Environment via Joint Learning of Content and Human Attention. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 279–288.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint* abs/1503.02531 (2015).

[19] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *ArXiv preprint* abs/2010.02666 (2020).

[20] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & time-budget-constrained contextualization for re-ranking. *ArXiv preprint* abs/2002.01854 (2020).

[21] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[22] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*. ACM, 2333–2338.

[23] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781.

[24] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[25] 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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[26] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466.

[27] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6193–6202.

[28] Xiaodan Li, Jinfeng Li, Yuefeng Chen, Shaokai Ye, Yuan He, Shuhui Wang, Hang Su, and Hui Xue. 2021. Qair: Practical query-efficient black-box attacks for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3330–3339.

[29] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhongyu Pei, Hao Yang, Jianjun Chen, Haixin Duan, Kun Du, Eihal Alowaisheq, Sumayah Alrwais, et al. 2016. Seeking nonsense, looking for trouble: Efficient promotional-infection detection through semantic inconsistency search. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 707–723.

[30] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. *Overview of the trec-2014 microblog track*. Technical Report. MARYLAND UNIV COLLEGE PARK.

[31] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[32] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119.

[33] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 1291–1299.

[34] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[35] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *ArXiv preprint* abs/1901.04085 (2019).

[36] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. 2019. A framework for the extraction of deep neural networks by leveraging public data. *ArXiv preprint* abs/1905.09165 (2019).

[37] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[38] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543.

[39] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *ArXiv preprint* abs/2101.05667 (2021).

[40] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 5835–5847.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[42] Jinfeng Rao, Wei Yang, Yuhao Zhang, Ferhan Türe, and Jimmy Lin. 2019. Multi-Perspective Relevance Matching with Hierarchical ConvNets for Social Media Search. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press,

232–240.

[43] Nisarg Raval and Manisha Verma. 2020. One word at a time: adversarial attacks on retrieval models. *ArXiv preprint* abs/2008.02197 (2020).

[44] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*. Springer, 232–241.

[45] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. ACM, 101–110.

[46] Congzheng Song, Alexander Rush, and Vitaly Shmatikov. 2020. Adversarial Semantic Collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 4198–4210.

[47] Junshuai Song, Jiangshan Zhang, Jifeng Zhu, Mengyun Tang, and Yong Yang. 2022. TRAttack":" Text Rewriting Attack Against Text Retrieval. In *Proceedings of the 7th Workshop on Representation Learning for NLP*. 191–203.

[48] Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal Adversarial Attacks with Natural Triggers for Text Classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3724–3733.

[49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[50] Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's Morphin' Time! Combating Linguistic Discrimination with Inflectional Perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2920–2935.

[51] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2153–2162.

[52] 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)*. Association for Computational Linguistics, Online, 5531–5546.

[53] Hongjun Wang, Guangrun Wang, Ya Li, Dongyu Zhang, and Liang Lin. 2020. Transferable, Controllable, and Inconspicuous Adversarial Attacks on Person Re-identification With Deep Mis-Ranking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*.

[54] Shuo Wang, Surya Nepal, Carsten Rudolph, Marthie Grobler, Shangyu Chen, and Tianle Chen. 2020. Backdoor attacks against transfer learning with pre-trained deep learning models. *IEEE Transactions on Services Computing* (2020).

[55] Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. 2019. Towards a robust deep neural network in texts: A survey. *ArXiv preprint* abs/1902.07285 (2019).

[56] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[57] William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)* 28, 4 (2010), 1–38.

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint* abs/1910.03771 (2019).

[59] Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2022. PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models. *ArXiv preprint* abs/2204.01321 (2022).

[60] Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are Neural Ranking Models Robust? *ACM Transactions on Information Systems (TOIS)* (2022).

[61] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. *ArXiv preprint* abs/1903.10972 (2019).

[62] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. 2019. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2041–2055.

[63] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*. Association for Computational Linguistics, Online, 1–4.

[64] Bin Zhou and Jian Pei. 2009. OSD: An online web spam detection system. In *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, Vol. 9.

[65] Mo Zhou, Zhenxing Niu, Le Wang, Qilin Zhang, and Gang Hua. 2020. Adversarial ranking attack and defense. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 781–799.

[66] Mo Zhou, Le Wang, Zhenxing Niu, Qilin Zhang, Yinghui Xu, Nanning Zheng, and Gang Hua. 2021. Practical relative order attack in deep ranking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16413–16422.