Black-box Adversarial Attacks against Dense Retrieval Models: A Multi-view Contrastive Learning Method

Yu-An Liu
Ruqing Zhang*
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
{liuyuan21b,zhangruqing}@ict.ac.cn

Jiafeng Guo[†]
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
guojiafeng@ict.ac.cn

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Wei Chen
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
chenwei2022@ict.ac.cn

Yixing Fan
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
fanyixing@ict.ac.cn

Xueqi Cheng
CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
cxq@ict.ac.cn

ABSTRACT

Neural ranking models (NRMs) and dense retrieval (DR) models have given rise to substantial improvements in overall retrieval performance. In addition to their effectiveness, and motivated by the proven lack of robustness of deep learning-based approaches in other areas, there is growing interest in the robustness of deep learning-based approaches to the core retrieval problem. Adversarial attack methods that have so far been developed mainly focus on attacking NRMs, with very little attention being paid to the robustness of DR models.

In this paper, we introduce the adversarial retrieval attack (AREA) task. The AREA task is meant to trick DR models into retrieving a target document that is outside the initial set of candidate documents retrieved by the DR model in response to a query. We consider the decision-based black-box adversarial setting, which is realistic in real-world search engines. To address the AREA task, we first employ existing adversarial attack methods designed for NRMs. We find that the promising results that have previously been reported on attacking NRMs, do not generalize to DR models: these methods underperform a simple term spamming method. We attribute the observed lack of generalizability to the interaction-focused architecture of NRMs, which emphasizes fine-grained relevance matching. DR models follow a different representation-focused architecture that prioritizes coarse-grained representations. We propose to formalize attacks on DR models as a contrastive learning problem in a multi-view representation space. The core idea is to encourage the consistency between each view representation of the target document and its corresponding viewer via view-wise supervision signals. Experimental results demonstrate that the proposed method can significantly outperform existing attack strategies in misleading the DR model with small indiscernible text perturbations.

CCS CONCEPTS

• **Information systems → Adversarial retrieval**.

KEYWORDS

Adversarial Attack, Dense Retrieval, Contrastive Learning

---

*Research conducted when the author was at the University of Amsterdam.
[†]Jiafeng Guo is the corresponding author.

1 INTRODUCTION

Information retrieval (IR) systems typically employ a multi-stage search pipeline, including the first-stage retrieval and the re-ranking stage [13]. The first-stage retrieval returns an initial set of candidate documents from a large repository, and the re-ranking stage re-ranks those candidates. Dense retrieval (DR) models [13, 55] and neural ranking models (NRMs) [7, 49] offer substantial performance improvements in the retrieval and re-ranking stage, respectively.

By modifying normal examples with malicious human-imperceptible perturbations, deep learning-based models can be deceived into providing attacker-desired inaccurate predictions [43]. DR models and NRMs are prone to inherit the adversarial vulnerability of general neural networks, emphasizing the need for reliable
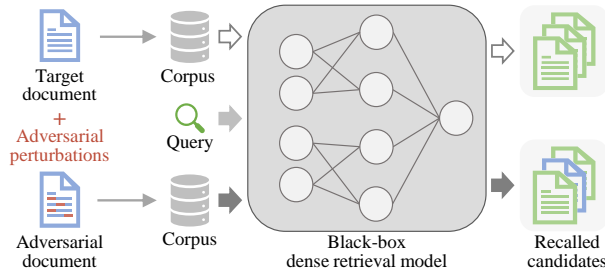
Figure 1: The adversarial retrieval attack (AREA) task.

and robust neural IR systems. Exploring potential adversarial attacks against neural models in IR is an important step towards this goal: Such explorations help identify vulnerabilities, serve as a surrogate to evaluate robustness before real-world deployment, and, consequently, aid in devising appropriate countermeasures.

To date, much attention has been devoted to the design of adversarial attacks against NRMs [25, 27, 47]. Given a neural ranking model, the attack aims to promote a low-ranked target document to higher positions via human-imperceptible perturbations. In contrast, little effort has been devoted to investigating how adversarial attacks affect DR models. We believe it is important to address this knowledge gap. Firstly, like NRMs, DR models are increasingly vital in practical IR systems. Adversarial attacks can expose their weaknesses and provide insights for developing more robust search engines. Secondly, within a multi-stage search pipeline, if black-hat search engine optimization practitioners [15] cannot ensure a target document successfully passes the first-stage retrieval, they will not have the chance to promote it in rankings in the final ranked list.

Adversarial attacks against DR models. We are the first to develop adversarial attacks against DR models. The first research question is: What is the goal of attacking DR models? Based on the adversarial attacks against NRMs and inspired by properties of the first-stage retrieval, we propose to define an attack task, the *adversarial retrieval attack* (AREA) against DR models. As shown in Figure 1, given a DR model, the AREA task is to retrieve a target document outside the initial set of K candidate documents for a given query, by perturbing the document content in a semantic-preserving way. We focus on a practical and challenging decision-based black-box setting, akin to the adversarial attacks against NRMs [25, 27, 47], where the adversary can only query the target DR model without direct model information access. For consistency with the multi-stage search pipeline in practical IR systems, we simulate a black-box "retrieval and re-ranking" pipeline, wherein the target DR model initially narrows down the candidate set to K documents, followed by an NRM determining the final top-K documents ordering. In this way, we query the pipeline and assess the final decision to perform attacks in a black-box manner.

Using NRM attack methods against DR models. To address the AREA task, the second research question arises: Do existing attack methods against NRMs perform as well against DR models as against NRMs? Our results show that these methods lag behind a simple term spamming attack typically involving query keyword stuffing [15]. Deep neural networks with interaction-focused architectures are usually employed for NRMs, while less complicated models with representation-focused architectures are adopted in DR

models [9, 14, 51]. Specifically, when attacking NRMs, the perturbation update relies on modeling fine-grained interactions between attacked documents and queries. In contrast, DR models depend on coarse-grained text representations for effective search in the representation space. This distinction renders the existing attacks against NRMs unsuitable for deceiving DR models.

Attack models tailored for DR models. The analysis we have just summarized leads to our third research question: Can we design an effective adversarial attack method tailored for DR models? As DR conducts retrieval purely in the representation space [13, 55], we introduce a *multi-view contrastive learning-based adversarial retrieval attack* (MCARA) to generate adversarial examples. Our key idea is to enhance the consistency of semantic representations between the target document and the K retrieved documents in the initial set using view-wise supervision. Specifically, after training a surrogate model to demystify the target DR model, we first obtain different viewers to represent documents in the initial set via a clustering technique. We produce multi-view representations for the target document through viewers. Then, a view-wise contrastive loss is applied to draw each view representation of the target document closer to its corresponding viewer in the semantic space while distancing it from nearest-neighbor documents outside the initial set. In this way, the attacker captures informative and discriminative semantic signals via view-level contrastive supervision. Finally, following [47], we use prior-guided gradients of the view-wise contrastive loss to identify the important words in a document, and adopt projected gradient descent [32] to generate gradient-based adversarial perturbations.

Experiment on two web search benchmark datasets show that MCARA effectively promotes the target document into the candidate set with high attack success and low time cost. According to both automatic and human evaluations, MCARA retains target documents semantics and fluency. Moreover, the adversarial examples produced by MCARA can deceive the NRM to some extent.

2 RELATED WORK

Dense retrieval. Dense retrieval [55] conducts first-stage retrieval [13] in the embedding space and has demonstrated several advantages over sparse retrieval [24]. It typically employs a dual-encoder architecture to embed queries and documents into low-dimension embeddings [45]using these similarities as estimated relevance scores [13]. By fine-tuning BERT with in-batch negatives [19], DR models have been shown to outperform BM25 [39]. Subsequently, research has explored various pre-training [10, 31], and fine-tuning techniques [21, 37, 53] to enhance DR models, achieving state-of-the-art performance on IR tasks. Besides high effectiveness, the robustness of DR models, such as out-of-distribution [26, 44, 52] and query variations [4, 36], has been focused. Unlike the work listed above, we focus on the *adversarial* robustness of DR models.

Adversarial attacks in IR. In IR, black-hat search engine optimization (SEO) has been a threat to search systems since the dawn of the world wide web [15]. Black-hat SEO usually aims to increase the exposure of a document owner's pages by maliciously manipulating documents, resulting in a decline in the quality of search results and inundation of irrelevant pages [1]. Research has shown

that neural ranking models (NRMs) inherit the adversarial vulnerabilities of deep neural networks, making them susceptible to small perturbations added to documents [48]. Research into adversarial attacks against NRMs has been growing, with the goal of promoting a target document in the rankings w.r.t. a query via imperceptible perturbations. Prior work investigates the vulnerability of NRMs in white-box [42, 46] or black-box [27, 47] scenarios, using word substitution [47] or trigger injection [25] as document perturbations. Similar to NRMs, DR models are also likely to inherit adversarial vulnerabilities of deep neural networks. The adversarial vulnerability of DR models remains under-explored.

Multi-view document representations. A single representation vector may not be able to properly model the fine-grained semantics of a document [55]. To tackle this issue, previous work has proposed approaches to explore multiple representations for enhancing the semantic interaction in DR. Poly-Encoder [18] learns multi-representations for modeling the semantics of a text according to multi-views. Zhang et al. [54] introduce multiple viewers to produce multi-view representations to represent documents and enforce them to align with different queries. In this work, we generate multi-view representations of a target document through viewers.

Contrastive learning. Contrastive learning [22] is a branch of self-supervised representation learning, which has been widely applied in computer vision [16, 17] and natural language processing [11, 41]. The key idea is to contrast pairs of semantically similar and dissimilar pairs of data, encouraging the representations of similar pairs to be close and those of dissimilar pairs to be further apart. In the context of dense retrieval, some work has adopted contrastive learning in guiding models to learn more distinguishable representations of documents [29, 50]. Unlike existing work, we aim to obtain an effective attack signal by pulling each view representation of the target document towards its corresponding viewer, while pushing it away from representations of counter-viewers.

3 PROBLEM STATEMENT

Given a query q, the aim of first-stage retrieval is to recall a subset of potentially relevant documents from a large corpus $C = \{d_1, d_2, \ldots, d_N\}$ with a total of N documents. In general, a first-stage retrieval model produces a relevance score $s(q, d)$ of the query q for each document d in C, and then recalls a set of candidates R by selecting the top-K documents with the highest predicted scores. Here, K represents the number of candidates in R, which is usually significantly smaller than the corpus size N. For example, the retrieval model outputs the initial set $R = \{d_1, d_2, \ldots, d_K\}$ with K candidates if it determines that $s(q, d_1) > s(q, d_2) > \cdots > s(q, d_K) > \cdots > s(q, d_N)$. In this case, d_K possesses the lowest relevance score within R.

Objective of the adversary. The *adversarial retrieval attack* (AREA) task is to fool the DR models into retrieving a target document outside the initial set of K candidates in response to a query appearing in the K initial candidates, by finding an optimized and imperceptible perturbation p. Formally, given a query q and a target document d out of the initial set, the goal is to construct a valid adversarial example $d^{adv} = d \oplus p$, that can be ranked above the K-th position. Specifically, p is crafted to conform to the following requirements,

$$\text{Recall}(q, d \oplus p) \leq K \text{ such that } p \leq \epsilon, \quad (1)$$

where $\text{Recall}(q, d \oplus p)$ denotes the ranking position of the adversarial example recalled by q. A smaller value of Recall denotes a higher ranking. In this case, the rank position of the original d is larger than K. ϵ is the maximum perturbation upper bound of p. Ideally, the perturbation p should preserve the semantics of document d and be imperceptible to human judges yet misleading to DR models. In this work, we use the number of word substitutions and the similarity of the substituted words as restrictions.

Decision-based black-box attacks. Since most real-world search engines are black boxes, here, we focus on the decision-based black-box attack setting for the AREA task, where the model parameters are inaccessible to the adversary. To align with practical IR systems' multi-stage pipelines, we simulate a retrieval-ranking pipeline by incorporating a representative NRM following the target DR model, refer to as black-box. We train a surrogate model [35] to imitate the target DR model, by querying the pipeline for the final ranking.

4 OUR METHOD

We first analyze the difference between attacking NRMs and DR models, and then introduce our attack method for AREA task.

4.1 Representation and interaction behavior

To address the AREA task, it is natural to consider existing attack methods designed for NRMs. However, as our experimental results of Section 6.1 show, unlike the success in NRMs, these methods designed for NRMs do not achieve promising performance. Below, we investigate the potential reasons from several perspectives.

Different model architectures in DR models and NRMs. During first-stage retrieval the aim is to discriminate a small set of candidate documents from (potentially) millions of documents in a *coarse-grained* way [13]. To this end, DR models with their representation-focused architectures (i.e., dual-encoder) are extensively adopted to evaluate relevance based on high-level representations of each input text and to ensure efficiency [55]. In contrast, the re-ranking stage conducts *fine-grained* relevance matching between a query and a small set of candidate documents [14]. To this end, NRMs with their interaction-focused architectures (i.e., cross-encoder) are widely used to directly learn from interactions rather than from individual representations and to maintain good system performance [51].

Different guidelines when attacking DR models and NRMs. To promote a target document in rankings, attacks on NRMs leverage the interaction signals with attention across the query and the target document tokens. The adversary captures the signal of *inner-document representativeness* [30, 37], which guides the computation of the update direction for adversarial perturbation. In contrast, when attacking DR models, it is important to consider *inter-document representativeness* [28, 29], since the dual-encoder architecture enables the encoding of queries and documents independently. To include the target document in the initial candidate set, the adversary aims to find a minimal perturbation that maximizes the probability of the DR model in distinguishing the target document from millions of documents in the embedding space.

In summary, the variation in model architectures and attack supervision signals pose considerable challenges when attempting to deceive DR models using attacks intended for NRMs. Consequently, it is important to develop attack techniques tailored for DR models.
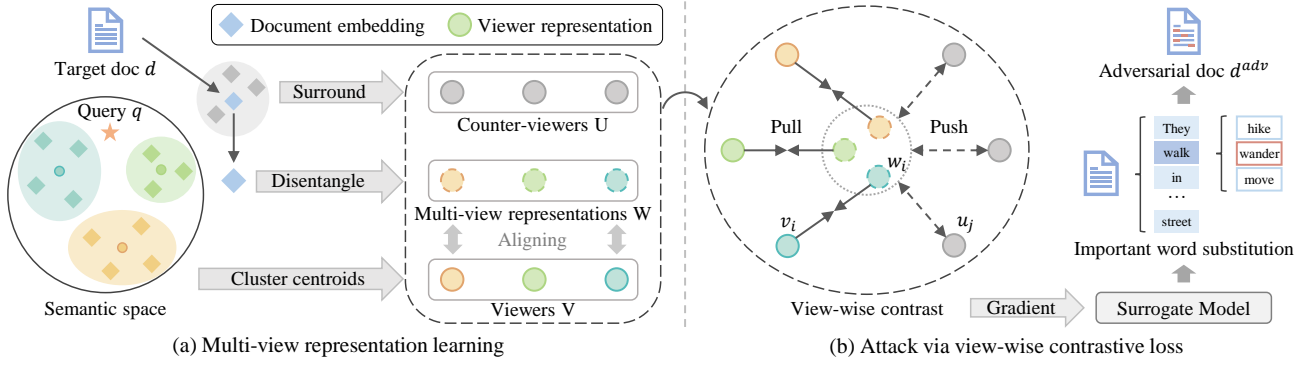
Figure 2: The overall architecture of MCARA. After training the surrogate retrieval model: (a) We learn the multi-view representations of the target document by identifying viewers and counter-viewers. (b) During the attack, a view-wise contrast is used to force each view of the target document close to its corresponding viewer, while away from other counter-viewers.

4.2 Overview of MCARA

High-quality text representation is the foundation of DR [29]. We propose to formalize the AREA task as a contrastive learning problem [22] in the representation space: (i) Push target document d away from other documents outside the initial set; and (ii) Pull d closer to the candidates inside the initial set. However, contrasting all documents inside and outside the initial set incurs computational overhead and lacks directional control. In this paper, we introduce representative viewers for the K candidates in the initial set and use the nearest neighbors of the target document in the semantic space as counter-viewers to serve as counterexamples.

Considering the viewers, a simple method to conduct contrastive learning is to directly encourage the representation of the target document and that of each viewer in the semantic space to be closer while keeping counter-viewers away. Nevertheless, such simultaneous attraction in multiple directions towards a single document representation could potentially lead to information loss. Here, we introduce a novel *multi-view contrastive adversarial retrieval attack* (MCARA). The key idea is to disentangle the target document embedding into multi-view representations through viewers, and then enhance the consistency between each view representation and the representation of its corresponding viewer. MCARA can be decomposed into three dependent components: (i) A **surrogate model imitation** trains a surrogate retrieval model to prepare for a gradient attack; (ii) **Multi-view representation learning** finds the viewers and counter-viewers, and generates multi-view representations for the target document; and (iii) **Attack via view-wise contrastive loss** generates the embedding space perturbations by calculating the gradients of the surrogate model via contrast. The overall architecture of MCARA is shown in Figure 2.

4.3 Surrogate model imitation

To simulate a realistic scenario, we regard the "retrieval and re-ranking" pipeline as a unified black-box model, where the retrieval model serves as the target DR model for our attack. For each dataset used in this study, we first train the target DR model and then train the NRM based on the retrieved candidates given by the DR model. We use state-of-the-art models [10, 30] as the backbone of the DR model and the NRM in the pipeline, respectively. By sending queries to the black-box pipeline and obtaining the ranked list (given by

the NRM), following [3, 47], we leverage the relative relevance information among the ranked list [8] to construct a synthetic dataset, for training a surrogate retrieval model.

Given a query q_c from a pre-collected query collection Q that accesses the black-box pipeline, we get the ranking result R_a of K documents returned by the pipeline. We generate pseudo-labels as the ground-truth by treating the first ℓ ranked documents $R_a[: \ell]$ as relevant documents R_a^+. Generally, training a well-performed DR model needs to combine random negative sampling and hard negative sampling [50]. Therefore, we treat the other documents $R_a[\ell + 1 : K]$ as hard negative examples R_a^-, and the ranked documents of other queries except for q_c in Q are regarded as random negative examples. We initialize the surrogate retrieval model $\tilde{f}$ using the vanilla BERT. The relevance score calculated by the surrogate retrieval model is $\tilde{f}(\cdot, \cdot)$. We train $\tilde{f}$ by optimizing a pairwise loss function as the negative log-likelihood of relevant documents:

$$\mathcal{L} = -\frac{1}{|Q|} \sum_{q_c \in Q} \log \frac{\tilde{f}(q_c, R_a^+)}{\tilde{f}(q_c, R_a^+) + \tilde{f}(q_c, R_a^-) + \sum_{R_c' \in \mathcal{R}_{/\{R_a\}}} \tilde{f}(q_c, R_c')}, \quad (2)$$

where $\mathcal{R}$ denotes the set of query collection's ranking results, and R_c' is the ranking result of other queries.

4.4 Multi-view representation learning

Based on the surrogate model $\tilde{f}$, we first learn multiple viewers from representations of K documents within the initial set returned to a query q, and then generate multi-view representations to represent the target document d through the learned viewers. In addition to the viewers, we employ a set of counter-viewers from representations of documents outside the initial set to prepare for attacks.

Deriving multiple viewers from the initial set of K candidates. The key idea is to find several indicative viewers to represent the documents within the initial set and provide guidance for the attack process. Here, the viewer is defined as a cluster of documents sharing the same topic. We will try other ways of finding viewers in the future. Given a query q, we first obtain the initial set R of K candidates from the simulated pipeline. Then, we use the document embedding generated by $\tilde{f}$ as the representation of each document in R. We apply clustering to the representations of K candidates to obtain n clusters where $n \ll K$, and leverage the representation of each centroid as a topical viewer.

Specifically, given the K documents in the initial set R, we use the K-Means clustering algorithm [40] to find a set V of n viewers,

$$V = \text{Kmeans}\left(n, \varphi\left(\tilde{f}, R\right)\right). \tag{3}$$

Here, $\varphi(\tilde{f}, R)$ are the embeddings of all K documents in R, with respect to the surrogate model $\tilde{f}$. In this way, we can obtain the representations of n viewers, denoted as $V = \{v_1, v_2, \ldots, v_n\}$.

Generating multi-view representations of the target document through viewers. The key idea is to disentangle the view information of the target document aligning to the given viewers, enabling us to effectively extract the specific relevance signal within the candidate set. We use a fully-connected layer with activation function ReLU as a multi-view representations generator. We feed the target document embedding w_d obtained by $\tilde{f}$ and the representations of viewers V into the generator. To obtain n multi-view representations $W = \{w_1, w_2, \ldots, w_n\}$ aligned to viewers, following [5, 6], we encourage the w_i and its corresponding viewer v_i to be similar while retaining the original information by minimizing the square loss, i.e.,

$$\mathcal{L}_{squ} = \sum_{i=1}^{n}\left(\|w_i - v_i\|_2^2 + \|w_i - w_d\|_2^2\right), \tag{4}$$

where v_i refers to the i-st viewer representation and w_i denotes i-st the disentangled view representation of target document.

We maintain the distinction between multi-view representations by maximizing the cosine similarity between them:

$$\mathcal{L}_{cos} = -\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{1}_{[i \neq j]}\frac{w_i \cdot w_j}{\|w_i\|_2\|w_j\|_2}. \tag{5}$$

Combining the two optimization objectives, the multi-view representations $W = \{w_1, w_2, \ldots, w_n\}$ of d are calculated by,

$$W = \arg\min\left(\mathcal{L}_{squ} + \lambda\mathcal{L}_{cos}\right), \tag{6}$$

where λ is a trade-off parameter.

Obtaining multiple counter-viewers from dynamic surrounding documents. To enable a contrastive learning based attack, we also propose to find the counter-viewers from documents outside the initial set R, pushing the target document away from its original position in the representation space. To achieve this goal, we use the dynamic surrounding documents of the target document as counter-viewers for contrast. During the attack process, a dynamic surrounding document d^s is the document among the top-n nearest-neighbor to the current perturbed document in the semantic space of the surrogate model $\tilde{f}$. We collect the embedding of each dynamic surrounding document:

$$U = \left\{\tilde{f}(d^s)\,\middle|\,d^s \in \{\kappa\,(C, d, n)\}\setminus R\right\}, \tag{7}$$

where $\kappa(\cdot)$ is a function returning top-n documents closest to the target document d in corpus C under the semantic space of $\tilde{f}$, and $\tilde{f}(d^s)$ is the embedding of d^s. Finally, we get the representations of n counter-viewers, denoted as $U = \{u_1, u_2, \ldots, u_n\}$.

4.5 Attack via view-wise contrastive loss

Based on the multi-view representations of the target document, viewer representations and counter-viewer representations, we describe how to achieve the attack using a view-wise contrast loss.

View-wise contrastive loss. The view-wise contrastive loss aims to pull each view representation of the target document close to its corresponding viewer, and push it away from the representations from all counter-viewers. Given a query, we aim to find the optimal attack direction for the target document under the semantic space of the surrogate model $\tilde{f}$ with a contrastive loss $\mathcal{L}_{CL}$:

$$\mathcal{L}_{CL} = -\sum_{i=1}^{n}\log\frac{\exp(\text{sim}(w_i, v_i)/\tau)}{\exp(\text{sim}(w_i, v_i)/\tau) + \sum_{u_j \in U}\exp(\text{sim}(w_i, u_j)/\tau)}, \tag{8}$$

where v_i is a viewer representation in V from Eq. (6), w_i is a view representation in W from Eq. (3), u_j is a counter-viewer representation in U from Eq. (7), n is the number of viewers, $\text{sim}(\cdot)$ is the dot-product function, and τ is the temperature hyperparameter.

Perturbation word selection. As demonstrated in [23, 47], only some important words in the target document act as influential signals for the final attack performance. Therefore, for each token h_z in the target document, we calculate the gradient magnitude g^{h_z} to the embedding vector of each token in $\tilde{f}$ using $\mathcal{L}_{CL}$,

$$g^{h_z} = \frac{\partial\mathcal{L}_{CL}}{\partial e_{h_z}^{\tilde{f}}}, \tag{9}$$

where the $\mathcal{L}_{CL}$ is the view-wise contrastive loss from Eq. (8), and $e_{h_z}^{\tilde{f}}$ is the embedding vector of h_z obtained by $\tilde{f}$.

Then, the word importance I_{h_z} of each token h_z is calculated by $I_{h_z} = \|g_t^{h_z}\|_2^2$. We only attack the top-m words with the highest importance for each target document d, i.e., $O = \{o_1, o_2, \ldots, o_m\}$.

Embedding perturbation and synonym substitution. We adopt the projected gradient descent [32] to generate gradient-based adversarial perturbations to the embedding space. Specifically, for each step t in total iterations η, we calculate the gradient g_t^d of $\mathcal{L}_{CL}$ with respect to target document d on embedding space. After η iterations, we obtain the perturbed embeddings e^p of all the important words O in d: $e^p = \left\{e_1^p, e_2^p, \ldots, e_m^p\right\}$ from gradient accumulation.

Then, we substitute the important words with synonyms S. Following [47], we utilize the embedding similarity of counter-fitted word embeddings [33] to determine synonyms and employ the same greedy word replacement strategy computed by perturbed important word embeddings e^p and synonym embeddings. Unlike existing work [27, 47], we select words from the documents in the initial candidate set as the pool of potential synonym set S. To further consider semantic and fluency constraints of the perturbed sentence, we use the language model perplexity [38] threshold ρ of the sentence containing the replacement word to refine the selection of the synonym set.

5 EXPERIMENTAL SETTINGS

In this section, we introduce our experimental settings. The datasets and code are available at https://github.com/ict-bigdatalab/AREA.

5.1 Datasets

Benchmark datasets. We conduct experiments on two standard dense retrieval benchmark datasets: the **MS MARCO Document Ranking dataset** [34] (MS-MARCO Document) which is a large-scale benchmark dataset for web document retrieval, with about 3.21 million documents, and the **MS MARCO Passage Ranking dataset** [34] (MS-MARCO Passage) which is another large-scale

benchmark dataset for web passage retrieval, with about 8.84 million passages. The relevant documents to user queries are obtained using Bing, thereby simulating real-world web search scenarios.

Target queries and documents. Following [3, 47], for each dataset, we randomly sample 500 Dev queries as target queries for evaluation. We adopt three types of target documents outside the initial candidate set, which exhibit different levels of attack difficulty, i.e., *Easy, Middle, Hard*. These documents are sampled from the retrieval results of the target DR model. For each target query, we select a total of 30 target documents. Beyond the above three separate sets of target documents, for each query, we also incorporate a random sampling of 10 documents from the original pool of 30 target documents. These documents are selected to showcase a diverse range of attack difficulties, forming a *Mixture* level.

5.2 Models

Baselines. We compare our method with several representative attack methods: (i) **Term spamming (TS)** [15] randomly selects a starting position in the target document and replaces the subsequent words with terms randomly sampled from the target query. (ii) **TF-IDF** simply replaces the important words in the target document, which have the highest TF-IDF scores based on the target query, with their synonyms. (iii) **PRADA** [47] is a decision-based black-box ranking attack method against NRMs via word substitution. We use the pairwise hinge loss between the target document and the documents from the initial candidate set of DR models to guide the attack. (iv) **PAT** [25] is an anchor-based ranking attack method against NRMs via trigger generation. We use the pairwise loss between the target document and the anchor (top-1 document) of DR models to guide the attack.

Model variants. We implement two variants of MCAR, denoted as (i) $MCARA_{single}$ removes the multi-view representation learning and directly leverages the single document embedding obtained by the surrogate model to contrast with different viewers. (ii) $MCARA_{ind}$ contrasts each viewer representation of the target document with its corresponding viewer independently and then calculates the gradient to find important words accordingly. In this way, we can obtain the intersection of important words found by independent gradient perturbation.

5.3 Implementation details

For MS-MARCO Document and MS-MARCO Passage, the size K of the initial candidate set is 100 and 1000 [28, 29, 31], respectively. To obtain the target documents, for each sampled query in the MS-MARCO Document, the Easy level comprises 10 documents ranked between [101, 200], with documents evenly sampled from the range. The Middle level includes 10 documents ranked between [201, 1000], again with documents evenly sampled from the range. The Hard level consists of 10 documents ranked outside the top 1000, with each document randomly selected from those outside top 1000. For MS-MARCO Passage, the Easy, Middle, and Hard documents are similarly sampled from the ranking range of [1000, 2000], [2000, 10000] and outside of the top 10000, respectively.

For the black-box "retrieval and re-ranking" pipeline, we choose a representative DR model called coCondenser [10] as the retriever and also as our target DR model. Following Zhan et al. [53], we fine-tune the pre-trained coCondenser using two-stage hard negatives sampling strategy on the corresponding dataset. We choose a representative NRM called PROP [30] as the re-ranker and fine-tune the pre-trained PROP using the relevance labels and the retrieval results given by coCondenser. Finally, we use the fine-tuned PROP to re-rank the initial candidate set retrieved by the fine-tuned co-Condenser and get the final ranked list for guiding the learning of the surrogate model. For surrogate model imitation, we choose vanilla $BERT_{base}$ [20] as the backbone of the surrogate DR model with a dual-encoder architecture. For each dataset, we utilize the Eval queries as the pre-collected query collection Q. We set ℓ to 1 due to the average number of relevant documents per query.

For multi-view representation learning, the number of viewers and counter-viewers n is set to 5 for MS-MARCO Document and 10 for MS-MARCO Passage, respectively. The trade-off parameter λ is 10. We train the multi-view representations generator using our target query-document pairs for 1 epoch with a learning rate of 1e-6. For attack via view-wise contrastive loss, we set the temperature hyperparameter τ as 0.1. The total iterations of attack η are 3. The perplexity threshold ρ is set to 50 for filtering synonyms that do not fluent in the original text. Following [47], the number of substitution words m in MCARA is set to 50 and 20 for the MS-MARCO Document and MS-MARCO Passage, respectively. For a fair comparison, we maintain the same number of substitutions in all baselines. And the trigger length of PAT is set to 10 and 5 for the MS-MARCO Document and MS-MARCO Passage, respectively.

5.4 Evaluation metrics

Attack performance. We consider two automatic metrics: (i) Success recall rate (SRR)@k (%) evaluates the percentage of after-attack documents d^{adv} retrieved into the candidate set R with $k \leq K$ documents. Note that the evaluation with $k < K$ is more strict than that with $k = K$. (ii) Normalized Ranking Shifts Rate (NRS)@K (%) evaluates the relative ranking improvement of after-attacked documents which are successfully recalled into the initial set with K candidates, i.e., NRS @$K = (\Pi_d - \Pi_{d^{adv}})/\Pi_d \times 100\%$, where Π_d and $\Pi_{d^{adv}}$ are the rankings of d and d^{adv} respectively, produced by the target DR model. Note that if d^{adv} is not successfully recalled into the initial set of K candidates, its NRS is set to 0.

Naturalness performance. We consider three automatic metrics: (i) Automatic spamicity detection, which identifies whether target pages are spam. Following [25], we adopt the utility-based term spamicity method [56] to detect the adversarial examples. (ii) Automatic grammar checkers, which compute the average number of errors in the attack documents. Specifically, we use two online grammar checkers, i.e., Grammarly [12] and Chegg Writing [2], following the settings in [25, 27]. (iii) Language model perplexity (PPL), which measures the fluency using the average perplexity calculated using a pre-trained GPT-2 model [38]. Furthermore, we leverage the human evaluation, which measures the quality of the attacked documents following the criteria in [47].

6 EXPERIMENTAL RESULTS

In this section, we discuss experimental results, findings and the attack effect between the first-stage retrieval and re-ranking stage discussed in earlier sections of the paper.

Table 1: Attack performance of MCARA and the baselines; ∗ indicates significant improvements over the best baseline ($p \leq 0.05$).

Dataset	Method	Easy			Middle			Hard			Mixture		
		SRR		NRS	SRR		NRS	SRR		NRS	SRR		NRS
		@10	@100	@100	@10	@100	@100	@10	@100	@100	@10	@100	@100
MS-MARCO Document	TF-IDF	16.0	40.9	32.1	11.1	28.0	23.6	4.2	14.4	13.6	10.3	28.6	23.2
	TS	37.8	88.1	67.5	27.2	58.0	60.3	15.1	35.8	33.5	27.1	61.1	54.6
	PAT	26.5	70.2	52.2	13.7	36.0	32.0	7.9	27.1	26.4	16.0	43.1	36.6
	PRADA	28.4	74.7	56.2	18.5	43.1	37.9	11.2	33.0	33.3	10.5	50.9	42.7
	MCARA_{single}	36.8	85.1	64.8	25.6	58.6	57.9	18.3*	44.1*	41.5*	26.9	62.9	54.7
	MCARA_{ind}	37.1	86.2	66.1	26.3	60.2	58.4	19.6*	45.7*	43.9*	27.4	63.9	56.1
	MCARA	**43.5***	**92.3***	**73.1***	**28.1***	**66.5***	**61.4***	**24.4***	**50.2***	**51.3***	**31.2***	**69.9***	**61.5***
		@100	@1000	@1000	@100	@1000	@1000	@100	@1000	@1000	@100	@1000	@1000
MS-MARCO Passage	TF-IDF	10.2	35.2	25.1	6.4	19.8	18.3	2.1	10.5	10.3	6.1	21.6	17.8
	TS	28.6	79.0	59.1	17.2	50.8	48.7	8.4	27.6	26.9	17.8	52.0	44.4
	PAT	16.4	62.3	46.7	9.4	30.0	28.6	5.3	23.4	21.5	10.4	38.6	32.3
	PRADA	20.1	68.2	51.0	13.8	39.9	39.6	10.6	31.5	30.1	14.7	46.4	40.2
	MCARA_{single}	24.8	74.2	56.0	16.3	48.3	46.2	9.7	31.2	29.6	16.8	51.0	43.8
	MCARA_{ind}	26.5	76.3	59.4	18.8*	51.9	49.9	11.1*	35.5*	34.9*	18.8*	54.6*	48.0*
	MCARA	**32.9***	**83.1***	**65.9***	**22.7***	**57.3***	**53.7***	**15.3***	**41.1***	**40.2***	**23.7***	**60.5***	**53.3***

6.1 Are attack methods against NRMs effective against DR models?

As shown in Table 1, (i) TF-IDF performs poorly, especially on Hard target documents, indicating that the heuristic method is not able to effectively find the most-vulnerable words that help the target model make judgments. (ii) TS performs moderately well, showing that directly adding spamming with query terms helps improve the relevance between the target document and the query. However, spamicity can easily be detected by anti-spamming detection [25, 27]. We will discuss this further in Section 6.3. (iii) When we look at the attack methods tailored for NRMs (PRADA and PAT), PRADA performs better than PAT. The reason may be that PRADA considers more documents for the pairwise loss calculation, thus obtaining more comprehensive information about the candidate set.

However, both PRADA and PAT perform worse than the simple Term Spamming method on DR models. The reason may be that NRMs and DR models have different model architectures and behaviors and thus require different supervision signals to guide the attack process. In general, the adversarial attack against DR models is a non-trivial problem for existing attack methods.

6.2 How does MCARA perform on DR models?

Overall performance. The performance of MCARA and its variants in the DR attack scenario can be found in Table 1: (i) Our MCARA outperforms all the baseline methods significantly, illustrating that it is necessary to attack DR models by capturing the inter-document representativeness in the semantic space. Generally, as the difficulty of an attack increases, the performance tends to decrease. We will explore more advanced objectives tailored for challenging documents in the future. (ii) In general, attacks on MS–MARCO Document tend to have a higher success rate compared to MS-MARCO Passage. The reason may be that the number of documents addressing the relevant topic is generally smaller than the number of passages extracted from those documents, offering
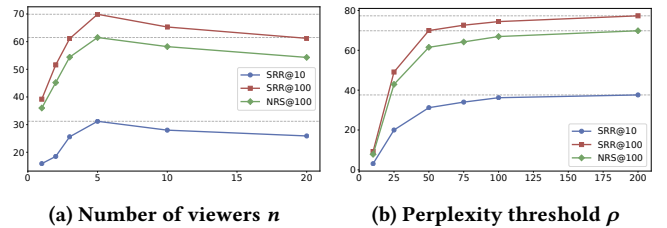

(a) Number of viewers n   **(b) Perplexity threshold ρ**

Figure 3: The impact of the number of viewers (a) and the perplexity threshold (b) on the attack performance of MCARA.

a more focused and concise set of information. (iii) The improvement of MCARA over MCARA_{single} suggests that incorporating multi-view document representations is more beneficial in finding fine-grained semantic information than a single document representation, and thus facilitates better contrasting between the target document and each viewer. (iv) The improvement of MCARA over MCARA_{ind} indicates that optimizing from only one view in the semantic space at a time may lead to disorder in the optimization direction of attacking the target document.

Impact of the number of viewers. We examine the impact of the important hyperparameter n of MCARA, i.e., the number of viewers, on the attack performance. The results on the Mixture target documents in MS-MARCO Document are shown in Figure 3 (a), with similar findings on the other target documents. We observe that the performance gets boosted when more representative viewers are incorporated into contrastive learning. The reason may be that more viewers can help extract sufficient representative signals for the attack. However, the performance gradually decreases when the number of viewers exceeds some threshold. Too many viewers increase the risk of making the clusters less representative, even introducing noise which is not good for contrast. In the future, we will explore other viewer extraction techniques, such as token embeddings and document-query alignment.

The impact of the perplexity threshold. We examine the impact of the fluency constrains hyperparameter ρ, i.e., the perplexity

threshold, on MCARA's performance. Lower ρ implies tighter substitution fluency constraints in the original sentences. The results on the Mixture target documents in MS-MARCO Document are shown in Figure 3 (b), with similar findings on other target documents. Reducing the fluency constraint leads to an improvement in the attack performance, since more disruptive synonyms could be selected. However, such a way may lead to the attack easily being detected, as discussed in Section 6.3. In the future, it is necessary to investigate more flexible ways to strike a balance between attack performance and the imperceptibility of adversarial perturbations.

6.3 Naturalness of adversarial examples

We discuss the naturalness of generated adversarial examples with respect to the Mixture-level target documents in MS-MARCO Document, with similar findings on MS-MARCO Passage. Here, we remove the fluency constraints of MCARA in the synonym substitution process, denoted as $MCARA_{-FC}$, for comparison.

Table 2: The detection rate (%) via a representative anti-spamming method on the MS-MARCO Document.

Threshold	0.08	0.06	0.04	0.02
TS	31.4	42.2	70.3	90.1
TF-IDF	18.3	31.1	53.6	78.2
PAT	8.2	13.2	24.3	46.0
PRADA	10.1	16.1	29.5	53.1
$MCARA_{-FC}$	11.2	17.5	31.6	55.0
MCARA	**6.9**	**11.6**	**23.3**	**44.1**

Automatic spamicity detection. Table 2 lists the automatic spamicity detection results. If the spamicity score of an example is higher than a detection threshold, it is detected as suspected spam content. We observe that: (i) As the threshold decreases, the detector intensifies in stringency, leading to an augmented detection rate across all methods. (ii) TS can easily be detected as it integrates numerous repeated query terms into documents. (iii) PAT and PRADA are relatively more undetectable than TS and TF-IDF since they both introduce naturalness constraints. (iv) MCARA outperforms the baselines significantly (p-value < 0.05), demonstrating the effectiveness of the synonym set derived from the words within candidate documents and the fluency constraints.

Table 3: Online grammar checkers, perplexity, and human evaluation results on the MS-MARCO Document.

Method	Cheg.	Gram.	PPL	Impercept.	*kappa*	Fluency	*Kendall*
Original	41	63	43.1	0.85	0.56	4.32	0.61
TS	78	120	110.6	0.06	0.42	2.12	0.79
TF-IDF	76	111	111.5	0.07	0.68	2.01	0.90
PAT	56	91	62.2	0.76	0.40	3.62	0.73
PRADA	67	102	86.3	0.62	0.51	3.10	0.85
$MCARA_{-FC}$	72	108	88.2	0.60	0.47	3.01	0.72
MCARA	51	87	58.1	0.82	0.46	3.81	0.81

Automatic grammar checker, PPL, and human evaluation. Table 3 lists the results of the automatic grammar checker, PPL, and human evaluation, including the annotation consistency test results (the *Kappa* value and *Kendall's Tau* coefficient) following [3, 25]. For human evaluation, we recruit five annotators to annotate 32 randomly sampled Mixture level adversarial examples from each

attack method [3]. Following [47], annotators score the *Fluency* of the mixed examples from 1 to 5; higher scores indicate more fluent examples. In terms of *Imperceptibility*, annotators determine whether an example is attacked (labeled as 0) or not (labeled as 1). We observe that: (i) TS performs poorly under all the naturalness evaluations, due to the spamming terms being inserted abruptly in various positions of the document, without considering semantics. (ii) Attack methods with naturalness constraint (i.e., PAT, PRADA) fall short when compared to the original samples, suggesting that making attack examples imperceptible is challenging. PRADA underperforms PAT, and a possible reason is that PRADA does not consider the fluency of the sentence with the word replacement. (iii) While removing the fluency constraint can enhance the attack performance, adversarial examples generated by $MCARA_{FC}$ raise suspicion. (iv) Despite the remaining gap to natural documents, the MCARA with fluency constraint achieves the best naturalness performance among all attack methods.

6.4 Can adversarial examples against DR models be promoted in the re-ranking stage?

Here, we explore the ability of the proposed MCARA to fool NRMs. For each dataset, we first obtain the candidate set for each query given by the target DR model, including the successful adversarial examples generated by MCARA. The size of the candidate set is 100 and 1000 for MS-MARCO Document and MS-MARCO Passage, respectively. Then, we directly feed these candidate sets to the NRM in the black-box pipeline. *Avg.rank* measures the average ranking of adversarial examples in the final ranked list and *T50%* and *T10%* measure the percentage of adversarial examples entering the top-50% and top-10% of the final ranking list. As shown in Table 4, some adversarial examples, as determined by the NRM, are positioned among the high-ranked entries in the final list. This suggests that the proposed MCARA method also poses a threat to NRMs.

Table 4: The performance of adversarial examples generated by MCARA on attacking against NRMs.

Tar. Docs	MS-MARCO Document			MS-MARCO Passage		
	Avg.rank	T50%	T10%	Avg.rank	T50%	T10%
Easy	68.2	34.2	5.4	736.6	28.3	3.4
Middle	82.3	15.8	2.6	874.7	11.6	1.2
Hard	91.6	3.6	0.0	958.1	2.7	0.0
Mixture	78.6	16.7	2.1	862.1	15.0	1.7

6.5 Can adversarial examples against NRMs pass the first-stage retrieval?

Here, we analyze whether adversarial examples tailored against the NRMs in the re-ranking stage can successfully pass through the first-stage retrieval. Following the attack setting in [47], we evenly select nine documents of varying rankings from each ranked list of 200 queries as the target documents for the attack. The size of the ranked list is 100 for MS-MARCO Document and 1000 for MS-MARCO Passage, respectively. We employ PRADA and PAT to attack NRMs and directly feed the adversarial examples into the target DR model. Here, *Drop* (%) measures the percentage of adversarial examples, whose rankings are decreased in the first-stage retrieval compared to the original rankings given by the target DR model. *NRS* (%)

measures the relative ranking changes of adversarial examples in the first-stage retrieval. Here we remove the constraint on the K candidates. Less than zero indicates an overall decrease in ranking, and greater than zero indicates an overall increase in ranking. *Lost* (%) counts the percentage of adversarial examples against NRMs that cannot be recalled in the first-stage retrieval.

Table 5: The performance of adversarial examples against NRMs on attacking against DR models.

Method	MS-MARCO Document			MS-MARCO Passage		
	Drop	NRS	Lost	Drop	NRS	Lost
PAT	30.2	7.7	22.1	28.1	10.3	6.3
PRADA	55.7	-28.4	38.6	49.6	-20.9	10.6

When applying PRADA for MS-MARCO Document, the rankings of 96.7% target documents are improved, with an average boost of 40.1% over the original ranking. When these adversarial examples are applied to the target DR model in first-stage retrieval, as shown in Table 5, 38.6% of them are not recalled. Similar results are observed in MS-MARCO Passage. In the "retrieval and re-ranking" pipeline, only considering the attacks against NRMs could risk target documents failing to be recalled. This underlines the importance of devising adversarial documents for first-stage retrieval.

6.6 Black-box vs. white-box attack

In addition to the black-box attack setting, exploring the white-box setting is valuable for gaining a deeper understanding of our method. We evaluate the retrieval performance of the surrogate model and the target DR model in the black-box pipeline over all the queries in the Dev sets of the MS-MARCO Document and MS-MARCO Passage, respectively. The MRR@100 of the target DR model and surrogate model on the MS-MARCO Document are 41.71 and 38.60, respectively. The MRR@100 of the whole black-box pipeline is 42.68. The MRR@10 of the target DR model and surrogate model on the MS-MARCO Passage are 39.82 and 36.94, respectively. The MRR@10 of the whole black-box pipeline is 41.46. To simulate a white-box scenario, we designate the target DR model as the surrogate model while keeping other components unchanged in MCARA, denoted as MCARA_{white}. The result of the Mixture target documents is shown in Table 6. The performance of MCARA under the black-box scenario is similar to that under the white-box scenario. This suggests that our surrogate model training method for the black-box pipeline can effectively mimic the behavior of the target DR model to execute threatening attacks on it.

Table 6: Attack performance comparisons of MCARA between the black-box and the white-box attack setting.

Method	MS-MARCO Document			MS-MARCO Passage		
	SRR		NRS	SRR		NRS
	@10	@100	@100	@100	@1000	@1000
MCARA	31.2	69.9	61.5	23.7	60.5	53.3
MCARA_{white}	33.3	71.1	63.6	24.8	61.6	54.6

6.7 Time costs of attack methods

Considering practical application scenarios, an effective attack method should also be efficient, meaning it should find the optimal perturbation with minimal time cost. Hence, we measure
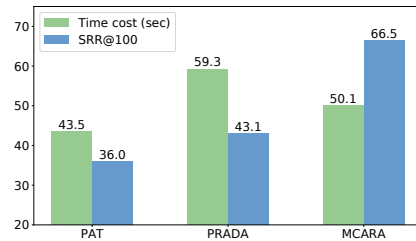

Figure 4: The average time cost of generating an adversarial document and attack performance of different methods.

the average time taken by different attack methods to generate an adversarial document using one Tesla V100 GPU. The results on the Mixture target documents in MS-MARCO Document are shown in Figure 4, with similar findings on the other target documents. PAT, relying solely on Anchor document information to optimize perturbations, has lower time costs but poorer attack performance due to insufficient information. In contrast, PRADA takes longer due to comparisons with more documents, while MCARA reduces the time overhead while achieving excellent attack results through efficient view-wise supervision.

7 CONCLUSION

In this paper, we proposed the AREA task against DR models, demonstrating that by adding small indiscernible perturbations, the adversarial examples can fool the DR models and pass them into the initial retrieval results. We developed a novel attack method MCARA, which utilizes view-wise supervision to capture the inner-document representativeness information in DR models for an effective attack. The proposed methodology and experimental results reveal the potential risk and vulnerabilities of DR models.

In future work, it is important to focus on the practical usage of adversarial attacks, specifically for sophisticated real-world search engines that operate with pipelined and ensemble approaches and dynamic corpora. A promising direction would involve designing a general unified attack method that can cater to different DR models and NRMs across multiple corpora and modalities. Besides, developing effective detection and defense mechanisms against such attacks is crucial for ensuring robustness in IR systems.

REFERENCES

[1] Carlos Castillo and Brian D. Davison. 2011. Adversarial Web Search. *Foundations and Trends in Information Retrieval* 4, 5 (2011), 377–486.

[2] Chegg. 2023. https://writing.chegg.com/.

[3] Xuanang Chen, Ben He, Zheng Ye, Le Sun, and Yingfei Sun. 2023. Towards Imperceptible Document Manipulations against Neural Ranking Models. In *ACL*.

[4] Xuanang Chen, Jian Luo, Ben He, Le Sun, and Yingfei Sun. 2022. Towards Robust Dense Retrieval via Local Ranking Alignment. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*. 1980–1986.

[5] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *ACL*.

[6] Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022. Learning Disentangled Textual Representations via Statistical Measures of Similarity. In *ACL*. 2614–2630.

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*. 985–988.

[8] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *SIGIR*. 65–74.

[9] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. 2022. Pre-training Methods in Information Retrieval. *Foundations and Trends® in Information Retrieval* 16, 3 (2022), 178–317.

[10] Luyu Gao and Jamie Callan. 2022. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *ACL*. 2843–2853.

[11] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *ACL*.

[12] Gramamrly. 2023. https://app.grammarly.com/.

[13] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic Models for the First-stage Retrieval: A Comprehensive Review. *TOIS* 40, 4 (2022), 1–42.

[14] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A Deep Look into Neural Ranking Models for Information Retrieval. *IP&M* 57, 6 (2020), 102067.

[15] Zoltan Gyongyi and Hector Garcia-Molina. 2005. Web Spam Taxonomy. In *AIRWeb*.

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, Vol. 2. 1735–1742.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

[18] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *ICLR*.

[19] 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 *EMNLP*.

[20] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[21] Omar Khattab and Matei A. Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR*.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *NeurIPS*, Vol. 33. 18661–18673.

[23] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *EMNLP*.

[24] Jimmy Lin. 2022. A Proposed Conceptual Framework for a Representational Approach to Information Retrieval. In *ACM SIGIR Forum*, Vol. 55.

[25] Jiawei Liu, Yangyang Kang, Di Tang, Kaisong Song, Changlong Sun, Xiaofeng Wang, Wei Lu, and Xiaozhong Liu. 2022. Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models. In *CCS*. 2025–2039.

[26] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023. On the Robustness of Generative Retrieval Models: An Out-of-Distribution Perspective. *arXiv preprint arXiv:2306.12756* (2023).

[27] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Topic-oriented Adversarial Attacks against Black-box Neural Ranking Models. In *SIGIR*.

[28] Shuqi Lu, Chenyan Xiong, Guolin Ke Di He, Waleed Malik, Zhicheng Dou, Paul Bennett, Tieyan Liu, and Arnold Overwijk. 2021. Less Is More: Pre-training a Strong Siamese Encoder Using a Weak Decoder. In *EMNLP*.

[29] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *SIGIR*. 848–858.

[30] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. Prop: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *WSDM*. 283–291.

[31] Xinyu Ma, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. A Contrastive Pre-training Approach to Discriminative Autoencoder for Dense Retrieval. In *CIKM*. 4314–4318.

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*.

[33] Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting Word Vectors to Linguistic Constraints. In *NAACL*. 142–148.

[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] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in Machine Learning: From Phenomena to Black-box Attacks Using Adversarial Samples. *arXiv preprint arXiv:1605.07277* (2016).

[36] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *ECIR*.

[37] 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 *NAACL*. Association for Computational Linguistics.

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI blog* 1, 8 (2019), 9.

[39] Stephen E Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR*.

[40] David Sculley. 2010. Web-scale k-Means Clustering. In *WWW*. 1177–1178.

[41] Jiaxin Shi, Chen Liang, Lei Hou, Juanzi Li, Zhiyuan Liu, and Hanwang Zhang. 2019. Deepchannel: Salience Estimation by Contrastive Learning for Extractive Document Summarization. In *AAAI*, Vol. 33. 6999–7006.

[42] Congzheng Song, Alexander M. Rush, and Vitaly Shmatikov. 2020. Adversarial Semantic Collisions. In *EMNLP*.

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. In *ICLR*.

[44] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[45] Nicola Tonellotto. 2022. Lecture Notes on Neural Information Retrieval. *arXiv preprint arXiv:2207.13443* (2022).

[46] Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. BERT Rankers are Brittle: A Study using Adversarial Document Perturbations. In *ICTIR*.

[47] Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models. *TOIS* 41, 4 (2023), Article 89.

[48] Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are Neural Ranking Models Robust? *TOIS* 41, 2 (2022), 1–36.

[49] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end Neural Ad-hoc Ranking with Kernel Pooling. In *SIGIR*. 55–64.

[50] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[51] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *WSDM*. 1154–1156.

[52] Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating Distribution Shifts in Zero-Shot Dense Retrieval with Contrastive and Distributionally Robust Learning. *arXiv preprint arXiv:2210.15212* (2022).

[53] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *SIGIR*.

[54] Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-View Document Representation Learning for Open-Domain Dense Retrieval. In *ACL*. 5990–6000.

[55] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. *arXiv preprint arXiv:2211.14876* (2022).

[56] 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.