

Multi-granular Adversarial Attacks against Black-box Neural Ranking Models

Yu-An Liu

CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
liuyuan21b@ict.ac.cn

Ruqing Zhang

CAS Key Lab of Network Data
Science and Technology, ICT, CAS
University of Chinese Academy of
Sciences
Beijing, China
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

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

Adversarial ranking attacks have gained increasing attention due to their success in probing vulnerabilities, and, hence, enhancing the robustness, of neural ranking models. Conventional attack methods employ perturbations at a single granularity, e.g., word or sentence level, to target documents. However, limiting perturbations to a single level of granularity may reduce the flexibility of adversarial examples, thereby diminishing the potential threat of the attack. Therefore, we focus on generating high-quality adversarial examples by incorporating multi-granular perturbations. Achieving this objective involves tackling a combinatorial explosion problem, which requires identifying an optimal combination of perturbations across all possible levels of granularity, positions, and textual pieces. To address this challenge, we transform the multi-granular adversarial attack into a sequential decision-making process, where perturbations in the next attack step build on the perturbed document in the current attack step. Since the attack process can only access the final state without direct intermediate signals, we use reinforcement learning to perform multi-granular attacks. During the reinforcement learning process, two agents work cooperatively to identify multi-granular vulnerabilities as attack targets and organize perturbation candidates into a final perturbation sequence. Experimental results show that our attack method surpasses prevailing baselines in both attack effectiveness and imperceptibility.

CCS CONCEPTS

• Information systems → Adversarial retrieval.

*Jiafeng Guo is the corresponding author.



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07.
<https://doi.org/10.1145/3626772.3657704>

KEYWORDS

Adversarial attack, Neural ranking model, Reinforcement learning

ACM Reference Format:

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Multi-granular Adversarial Attacks against Black-box Neural Ranking Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, July 14–18, 2024, Washington, DC, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3626772.3657704>

1 INTRODUCTION

With the advance of deep neural networks [24], neural ranking models (NRMs) [9, 16, 33, 37] have achieved promising ranking effectiveness in information retrieval (IR). Besides their proven effectiveness, considerable attention has been directed toward assessing the adversarial robustness of NRMs.

Adversarial ranking attacks. In IR, NRMs are prone to inheriting vulnerabilities to adversarial examples from general neural networks [29, 42, 53]. Such adversarial examples are crafted by introducing human-imperceptible perturbations to the input, capable of inducing model misbehavior. This discovery has sparked legitimate concerns about potential exploitation by black-hat SEO practitioners aiming to defeat meticulously designed search engines [17]. Consequently, there is a need to develop robust and reliable neural IR systems. A crucial step in this direction involves introducing *adversarial ranking attacks* to benchmark the vulnerability of black-box NRMs [5, 29, 53]. Here, the aim of an adversary is to find human-imperceptible perturbations injected into the document's text, to promote a low-ranked document to a higher position in the ranked list produced for a given query [5, 53]. This attack approach allows us to identify vulnerabilities in NRMs before deploying them in real-world settings and devise effective countermeasures.

Single-granular ranking attacks. Existing studies on adversarial attacks against NRMs are typically restricted to document perturbation strategies that operate at a single level of granularity, such

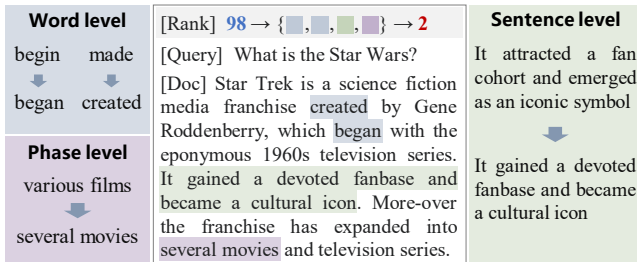


Figure 1: To prompt a target document in the rankings to a query, we identify multi-granular texts within the document as attack targets to generate effective adversarial examples.

as the word-level [42, 53] or sentence-level [5, 29]. These methods face two main limitations: (i) *Different perturbation granularities for different query-document pairs*: When dealing with different query-document pairs, *a priori* limiting perturbations to a single granularity could considerably restrict the choice of attack targets, thereby impeding the overall effectiveness of the attack. (ii) *Multiple perturbation granularities for a query-document pair*: Despite employing a meticulously chosen attack granularity tailored to specific query-document pairs, a fixed granularity falls short of fully capturing the diverse relevance patterns inherent in the matching between a query and a document [12]. An effective ranking attack should possess the flexibility to simultaneously consider various granularity perturbations in an adversarial example. In this sense, we argue that the full potential of adversarial attacks has yet to be harnessed for uncovering the vulnerabilities of NRMs.

Multi-granular adversarial ranking attacks. In this paper, we develop multi-granular adversarial ranking attacks against NRMs; see Figure 1 for an illustrative example. We incorporate word-level, phrase-level, and sentence-level perturbations to generate fluent and imperceptible adversarial examples. The generated examples may not cover all three levels of granularity but allow for flexible selection based on an optimization strategy. Compared to existing single-granular attacks, this multi-granular approach broadens the selection of attack targets and explores the vulnerability distribution of NRMs at various granularities within a specific query-document pair. Consequently, it can yield richer and more diverse forms of adversarial examples, thereby enhancing campaign performance.

Learning multi-granular attack sequences. Achieving a multi-granular attack is non-trivial due to the combinatorial explosion arising from the numerous possible actions, e.g., perturbation granularities, target positions in the document, and replacement content, posing a significant computational challenge. To tackle this, we formulate the multi-granular ranking attack as a sequential decision-making process [34]. In this process, the attacker sequentially introduces a perturbation at a specific granularity, i.e., word, phrase, or sentence, guided by perturbations in the preceding steps.

Within the sequential decision-making process, the discrete and non-differentiable nature of the text space presents a challenge in finding a direct supervisory signal to facilitate the incorporation of multi-granular perturbations. Therefore, we propose RL-MARA, a novel reinforcement learning (RL) framework [50] to navigate an appropriate sequential multi-granular ranking attack path.

Following [5, 29, 53], our focus is on a practical and challenging decision-based black-box setting [2], where the adversary lacks

direct access to model information and can only query the target NRM. We train a surrogate ranking model to imitate the black-box target NRM. We combine the surrogate ranking model with a large language model (LLM) to form the complete environment, to provide rewards for assessing the effectiveness of the multi-granular attack and the naturalness of the perturbed document, respectively. We set up a multi-granular attacker by building upon existing single-granular attack methods agents. A sub-agent and a meta-agent are designed in a cooperative manner: the sub-agent is tasked to identify multi-granular vulnerabilities in the document as attack targets, while the meta-agent is tasked to generate and organize perturbations into a final perturbation sequence, respectively. During the RL process, the attacker sequentially incorporates perturbations until the cumulative perturbation exceeds a predefined budget.

Main findings. We conduct experiments on two web search benchmark datasets, MS MARCO Document Ranking [35] and ClueWeb09-B [7]. Experimental results demonstrate that RL-MARA significantly improves the document ranking of target documents and thus achieves a higher attack success rate than existing single-granular ranking attack baselines. According to automatic and human naturalness evaluations, RL-MARA could maintain the semantic consistency and fluency of adversarial examples.

2 PROBLEM STATEMENT

In ad-hoc retrieval, given a query q and a set of N document candidates $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$ from a corpus C , the objective of a ranking model f is to assign a relevance score $f(q, d_n)$ to each pair of q and $d_n \in \mathcal{D}$, to obtain the ranked list L .

Adversarial ranking attack. Given a target document d and a query q , the primary goal of adversarial ranking attacks is to construct an adversarial example d^{adv} that is ranked higher than the original d in response to q by NRMs, all while closely resembling d [5, 29, 31, 53]. The adversarial example d^{adv} can be regarded as $d \oplus \mathcal{P}$, where \mathcal{P} denotes the perturbation applied to d . The perturbations \mathcal{P} are crafted to conform to the following properties [5, 29, 53]:

$$\begin{aligned} \text{Rank}(q, d \oplus \mathcal{P}) &< \text{Rank}(q, d) \\ \text{such that } \|\mathcal{P}\| &\leq \epsilon, \text{Sim}(d^{\text{adv}}, d) \geq \lambda, \end{aligned} \quad (1)$$

where $\text{Rank}(q, d \oplus \mathcal{P})$ and $\text{Rank}(q, d)$ denote the position of d^{adv} and d in the ranked list of q ; a smaller value of ranking position denotes a higher ranking; ϵ represents the budget for the number of manipulated terms $\|\mathcal{P}\|$; λ is the coefficient; and the function $\text{Sim}(\cdot)$ assesses the semantic or syntactic similarity [14, 53] between two documents. Ideally, d^{adv} should preserve the original semantics of d , and be imperceptible to human judges yet misleading to NRMs.

Decision-based black-box attacks. Following [29, 53], we focus on decision-based black-box attacks against NRMs for the adversarial ranking attack task. This choice is motivated by the fact that the majority of real-world search engines operate as black boxes, granting adversaries access only to the final decision, i.e., the rank positions within the partially retrieved list.

Perturbations at multiple levels of granularity. Existing work mainly focuses on a single granularity of \mathcal{P} to manipulate the target document, esp. word-level word substitution [53] and sentence-level trigger generation [29]. However, restricting perturbations to a single granularity may fail to adequately capture the nuanced

and diverse vulnerability features, a limitation confirmed by our experimental results (see Section 6.1). Thus, we propose to find perturbations \mathcal{P} at multiple levels of granularity.

Effective character-level [56] and phrase-level [25] modifications have proven successful in textual attacks [56] within NLP, but are underutilized in IR. However, character-level attacks tend to create ungrammatical adversarial examples and are easily defended against [40]. Considering the naturalness requirements, introducing perturbations at higher levels of granularity, e.g., paragraph level, may pose challenges in avoiding suspicion. Therefore, we propose to launch adversarial ranking attacks at three levels of perturbation granularity, i.e., word, phrase, and sentence levels.

3 PRELIMINARIES

Our approach relies on three single-granular adversarial ranking attack methods, i.e., word-level, phrase-level, and sentence-level. Typically, single-granular attacks begin by training a surrogate ranking model that imitates the target NRM. Subsequently, they execute attacks guided by the surrogate model, which involve two primary steps: (i) Identifying vulnerable positions in the target document; and (ii) Perturbing the text at these identified positions.

Word-level attack. For word-level attacks against NRMs, the main approaches include word substitution [53], word insertion [52], and word removal [46]. In this study, we employ the word substitution ranking attack exemplified by PRADA [53], which has demonstrated a high attack success rate. Specifically, PRADA first identifies vulnerable words in a document that significantly influence the ranking result through the surrogate model; and then replaces vulnerable words with synonyms, selecting those that most significantly improve the rankings from candidate synonyms.

Phrase-level attack. In textual attacks within NLP, the predominant method for phrase-level attacks [25, 57] is phrase substitution [25]. To the best of our knowledge, there has been an absence of phrase-level attacks specifically targeting NRMs in IR. PLAT [25] stands out in phrase-level textual attacks, aiming to induce text misclassification. PLAT first identifies the positions of vulnerable phrases that significantly influence the classification scores predicted by the surrogate model. Then, it uses BART [26] to generate variations for each vulnerable phrase. PLAT selects the variant that introduces the most substantial interference in classification scores predicted by the surrogate model, deviating from the original phrase. To adapt PLAT from classification to ranking, we use a sub-agent (see Section 4.3.1) to find positions of important phrases and replace classification scores with relevance scores.

Sentence-level attack. For sentence-level attacks against NRMs, key strategies encompass sentence substitution [29], sentence insertion [5], and sentence rewriting [48]. Here, we employ the sentence substitution ranking attack exemplified by PAT [29], which replaces a sentence at a specific position in a document with a trigger. Specifically, PAT first designates the beginning of the document as the vulnerable sentence; and then generates a continuous trigger representation and maps it to the word space. In this work, we take a more flexible way, employing the sub-agent to identify important sentences, potentially located anywhere within the document.

Combining single-granular attacks. In our multi-granular attack: (i) Initially, a sub-agent, serving as a vulnerability indicator

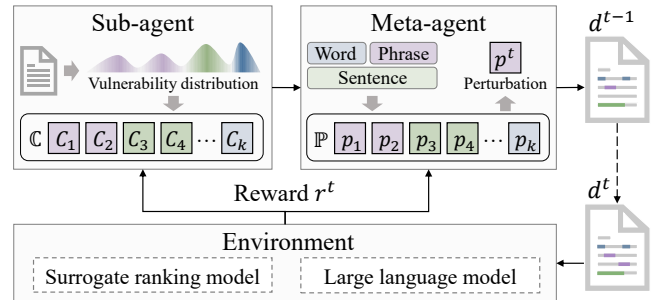


Figure 2: The RL-MARA framework.

(see Section 4.3.1), is employed to identify important positions at each level of granularity. (ii) Then, the three aforementioned single-granular attack methods are applied to generate a perturbation for each identified important position (see Section 4.3.2). (iii) Finally, an organization of all settled perturbations is executed to refine and select the most effective perturbation sequence.

Discussion. Our framework can seamlessly integrate off-the-shelf ranking attack methods. In this work, for each granularity level, we choose a representative attack method. In the future, we plan to consider more granularity and add more attack methods to make the perturbation more diverse. In addition, going forward we hope to make these attack methods learnable and dynamically update them within the entire framework to achieve enhanced interoperability.

4 METHOD

In this section, we introduce the RL-MARA framework, the code is available at <https://github.com/Davion-Liu/MARA>.

4.1 Overview

To generate multi-granular perturbations for a target document, we need to decide the granularity, position, and modified content of each single perturbation. We formulate multi-granular ranking attacks as a sequential decision-making process: (i) The attacker manipulates the target document by introducing a perturbation that could be any granularity; a surrogate model of the target NRM gives the current ranking position, while a large language model (LLM) evaluates its naturalness; (ii) The attacker observes changes in ranking and naturalness and further optimizes its strategy to generate the next perturbation. The global objective is to improve the final ranking of the target document with indiscernible perturbations.

During the sequential decision-making process, the discrete perturbed document leads to a lack of direct signals at each step. Therefore, we employ reinforcement learning (RL) to identify an appropriate sequential attack path for generating adversarials. Specifically, we introduce the RL-based framework RL-MARA to learn an optimal multi-granular ranking attack strategy. As shown in Figure 2, the RL-MARA framework comprises two major components: (i) A surrogate model simulating the behavior of the target NRM, and an advanced LLM evaluating the naturalness of adversarial samples, collectively serve as the whole environment to provide rewards; and (ii) A multi-granular attacker, consisting of a sub-agent and a meta-agent, receives rewards from the environment and collaborates to generate perturbations at multiple levels of granularity.

4.2 Environment and reward

The multi-granular ranking attack problem is formally modeled as a Markov decision process (Markov decision process (MDP)) [1], wherein its components are defined as follows: (i) **State** d is the document, with the initial state d^0 as a target document, and the terminal state signifying a successful adversarial example; (ii) **Action** p denotes a multi-granular perturbation selected by the agent for injection into the document; (iii) **Transition** \mathcal{T} alters the document state d by applying a perturbation at each step; and (iv) **Reward** r is the attack reward provided by the environment, guiding the agent with supervisory signals.

Environment. In the decision-based black-box scenario, obtaining only hard-label predictions and lacking the relevance score for each candidate document predicted by the target NRM, poses a challenge. Besides, frequent queries to the target NRM might arouse suspicion. Consequently, we employ a surrogate ranking model to function as an environment and offer the attack reward as a proxy for the target NRM. Following [29, 53], we train a surrogate ranking model based on the Pseudo Relevance Feedback idea [10] and achieve comparable performance to the target NRM. Specific training details can be found in Section 5.4. Simultaneously, we introduce an advanced LLM as part of the environment to assess the naturalness of the current perturbed document as a reward. To sum up, the virtual environment for the RL Attacker comprises a surrogate ranking model and an LLM.

Multi-granular reward design. An effective reward function for multi-granular attacks should consider both attack effectiveness and the naturalness of the perturbed document. At each step of the sequential interactions, the attacker introduces a perturbation at a specific level of granularity. The reward furnishes appropriate feedback based on the granularity of the current perturbation, guiding the behavior of the attacker.

Specifically, the reward for each step is defined as follows: (i) If the relevance score of the current perturbed document is higher than before, the current attack *succeeds*. The reward not only evaluates the attack effectiveness and the naturalness of the perturbed document, but also considers the number of manipulated terms introduced by different perturbation granularities; and (ii) Conversely, if the attack *fails*, we directly apply a fixed penalty factor ξ as the reward. These assumptions lead us to define the attack reward function r^t at the step t as follows:

$$r^t = \begin{cases} -\xi, & \text{if } \tilde{f}(q, d^t) < \tilde{f}(q, d^{t-1}) \\ r_{\text{att}}^t / |p^t| + \beta r_{\text{nat}}^t, & \text{else,} \end{cases} \quad (2)$$

where d^t and d^{t-1} are the perturbed document at step t and $t-1$, respectively; r_{att}^t and r_{nat}^t are the rewards with respect to attack effectiveness and document naturalness, respectively. The penalty factor ξ is set to 1. The function $\tilde{f}(\cdot)$ outputs the relevance score judged by the surrogate ranking model \tilde{f} , and the hyper-parameter β balances attack effectiveness with document naturalness.

To emphasize the impact of different levels of granularity, we introduce $|p^t|$ as a reward discount of attack effectiveness, representing the number of manipulated terms of p^t . The value of $|p^t|$ varies significantly across perturbations at different levels of granularity. For instance, when an attacker introduces a sentence-level perturbation, it consumes a larger portion of the perturbation

Perturbed Document: {Perturbed document}
Original Document: {Original document}
< Naturalness judgment >
 Please evaluate the naturalness of the perturbed document. The naturalness includes the similarity of the perturbed document to the original document and its own fluency. Begin your evaluation by providing a short explanation. After providing your explanation, you must rate both similarity and fluency response on a scale of 0 to 1 by strictly following this format: "[Similarity: [], Fluency: []]".

[Similarity: 0.8, Fluency: 0.6]

Figure 3: Instruction for naturalness evaluation with chatGPT. The gray and dark blue blocks indicate the inputs and outputs of the model, respectively.

budget. To normalize its effect, given its expected stronger attack effects, we incorporate a corresponding discount factor.

Next, we detail r_{att}^t and r_{nat}^t :

- **Attack effectiveness reward.** r_{att}^t incentivizes ranking improvements of the perturbed document d^t ; a perturbed document should receive more rewards if it is ranked higher than before. However, directly using ranking as a reward is sparse. We shape the reward using the surrogate model's relevance scores, i.e.,

$$r_{\text{att}}^t = \tilde{f}(q, d^t) - \tilde{f}(q, d^{t-1}), \quad (3)$$

where $\tilde{f}(\cdot)$ outputs the relevance score judged by \tilde{f} . If the step t attack is successful, r_{att}^t is positive.

- **Document naturalness reward.** r_{nat}^t guarantees that the perturbed document d^t satisfies semantic and syntactic constraints by an LLM. In this work, we employ ChatGPT [38] as the LLM. Specifically, we use the Prompts, shown in Figure 3, to evaluate both the similarity and fluency of documents in each state, defined as r_{sim}^t and r_{flu}^t , respectively.

- r_{sim}^t measures how semantically similar the perturbed document d^t is to the original document:

$$r_{\text{sim}}^t = \text{LLM}_{\text{sim}}(d^t, d^0), \quad (4)$$

where the function $\text{LLM}_{\text{sim}}(\cdot)$ outputs the similarity score judged by the LLM.

- r_{flu}^t measures how fluent the perturbed document d^t is:

$$r_{\text{flu}}^t = \text{LLM}_{\text{flu}}(d^t), \quad (5)$$

where the function $\text{LLM}_{\text{flu}}(\cdot)$ outputs the fluency score judged by the LLM.

Finally, the overall reward with respect to document naturalness r_{nat}^t is defined as $r_{\text{nat}}^t = r_{\text{sim}}^t + r_{\text{flu}}^t$.

4.3 Multi-granular attacker

The objective of the multi-granular attacker is to identify possible attack positions at all granularities and organize them into a final perturbation sequence. Developing such a composite strategy is challenging for a single agent. Consequently, we adopt two agents in a cooperative manner to accomplish this goal. (i) The **vulnerability indicator** serves as a sub-agent, aiming to identify all vulnerable positions in the document at each level of perturbation granularity. (ii) The **perturbation aggregator** acts as a meta-agent, aiming to generate specific perturbations for each selected vulnerable position and organize them to filter out the final perturbation sequence.

4.3.1 Sub-agent: vulnerability indicator. The aim of the vulnerability indicator is to identify all vulnerable positions within the target document and the corresponding level of perturbation granularity of each position.

Policy network. We employ BERT [20] as the backbone of the sub-agent policy network I_ϕ . Given a target document d and a query q , the process proceeds as follows:

- We employ the surrogate model \tilde{f} to compute the pairwise loss $\mathcal{L}_{\text{pair}} = \sum_{d' \in L \setminus d} \mathcal{L}_{\tilde{f}}(q, d, d')$, where d' is the remaining documents in the ranked list L excluding the target document d .
- We compute the average gradient $\mathbf{g}_d \in \mathbb{R}^{m \times l}$ of $\mathcal{L}_{\text{pair}}$ concerning each position (i.e., each word) in the target document d . Here, m represents the dimensions of the hidden state, and l denotes the length of the target document d .
- We feed \mathbf{g}_d into the vulnerability indicator I_ϕ , to derive the vulnerability distribution \mathbf{u} . The vulnerability distribution \mathbf{u} represents a list of confidence scores indicating the confidence level for each position in the target document across various levels of perturbation granularity, which is calculated by:

$$\mathbf{u} = I_\phi(\mathbf{g}_d), \quad (6)$$

where $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_l\} \subseteq \mathbb{R}^{4 \times l}$. Each $\mathbf{u}_i \in \mathbb{R}^4, i \in [1, l]$, represents the confidence scores at each level of perturbation granularity at position i . The four dimensions correspond to perturbation at word-level (W), phrase-level (P), sentence-level (S), and no perturbation (N), respectively.

- To condense the vulnerability distribution into a specific perturbation type, we apply the softmax function to \mathbf{u} , yielding the vulnerable word positions \mathbf{c} for the target document d , i.e.,

$$\mathbf{c} = \text{softmax}(\mathbf{u}), \quad (7)$$

where $\mathbf{c} = \{c_1, c_2, \dots, c_l\} \subseteq \mathbb{R}^{1 \times l}$, and each $c_i \in \{\text{W}, \text{P}, \text{S}, \text{N}\}, i \in [1, l]$, is the granularity of perturbation at each word position i .

During the above process, to ensure the continuity of vulnerable positions at the phrase and sentence granularity, we constrain the output of the vulnerability indicator as part of a sequence labeling process, as detailed in Section 5.4. Based on this, we can map the vulnerable word positions to corresponding *span positions* at the word, phrase, and sentence levels, denoted as $\mathbb{C} = \{C_1, C_2, \dots, C_k\} \in \mathbb{R}^{1 \times k}, k < l$. Each $C_j \in \{\text{W}, \text{P}, \text{S}\}, j \in [1, k]$, represents the level of perturbation granularity at each span j .

4.3.2 Meta-agent: perturbation aggregator. The target of the perturbation aggregator is to generate specific perturbations for each selected vulnerable span position and organize them into a final perturbation sequence.

Generating specific perturbations for each selected span position via static single-granular attack methods. Once we have identified the vulnerable span positions with the corresponding perturbation granularity \mathbb{C} with the vulnerability indicator, we employ the respective attack method (outlined in Section 3) for each of these span positions to generate the specific perturbations $\mathbb{P} \in \mathbb{R}^k$. Based on \mathbb{P} , we design the policy network, which sequentially selects a perturbation p^t from \mathbb{P} , adding to the target document until the budget of manipulated words number ϵ is reached.

Policy network. We employ a multi-layer perception (MLP) [44] as the backbone of the meta-agent policy network G_ϕ . For each

step t , G_ϕ takes the perturbed document d^{t-1} and vulnerability distribution \mathbf{u} as inputs. The action p^t is to select the t -th specific perturbation added to the document. The process is as follows:

- We use the surrogate model \tilde{f} to obtain the hidden states of the perturbed document d^{t-1} as $\mathbf{h}^{t-1} = [\mathbf{h}_1^{t-1}, \mathbf{h}_2^{t-1}, \dots, \mathbf{h}_l^{t-1}]$, where l is the length of d^{t-1} . $\mathbf{h}_i^{t-1} \in \mathbb{R}^m$ is the hidden state of the i -th word in d^{t-1} , where m is the dimension of hidden states.
- For each span (e.g., word, phrase, sentence), let's consider the j -th span. From its starting position j -start to its ending position j -end, we concatenate the corresponding hidden state with the confidence score. Then, we sum them up to create the concatenated representation, denoted as, $\sum_{o=j, \text{start}}^{j, \text{end}} [\mathbf{h}_o^{t-1}; \mathbf{u}_o]$, where $[\cdot; \cdot]$ is the concatenation operation.
- We divide the concatenated representation by the length of the span to derive the final representation \mathbf{e}_j^{t-1} for the current state:

$$\mathbf{e}_j^{t-1} = \left(\sum_{o=j, \text{start}}^{j, \text{end}} [\mathbf{h}_o^{t-1}; \mathbf{u}_o] \right) / |p_j|, j \in [1, k], \quad (8)$$

where $|p_j|$ is length of the specific perturbation of j -th span in \mathbb{P} .

- The probability distribution of each specific perturbation $p(p_j | d^{t-1})$ at t -step can be calculated by perturbation aggregator G_ϕ :

$$P(p_j | d^{t-1}) = G_\phi(\mathbf{e}_j^{t-1}), p_j \in \mathbb{P}. \quad (9)$$

The aggregator samples the perturbation p^t at step t with the highest probability from the distribution to be injected into d .

- We get the final perturbation $\mathcal{P} = \{p^1, p^2, \dots, p^t, \dots, p^T\}$, where $T < |\mathbb{P}|$ and T is the number of steps.

4.4 Training with policy gradient

We solve the MDP problem with the policy gradient algorithm REINFORCE [50]. In each episode, a trajectory $\tau = d^1, p^1, \dots, d^T, p^T$ is sampled using policy π . The episode terminates at step T , when term manipulation budget ϵ is reached, i.e. $\sum_{t=1}^T |p^t| \leq \epsilon < \sum_{t=1}^{T+1} |p^t|$.

The aim of training is to learn an optimal policy π^* by maximizing the expected cumulative reward $R(\tau) = \mathbb{E} \left[\sum_{t=1}^T \gamma^t r^t \right]$, where $\gamma \in [0, 1)$ is the discount factor for future rewards. The training objective is to maximize $J(\phi, \varphi)$ via:

$$\nabla_{\phi, \varphi} J(\phi, \varphi) = \mathbb{E}_{\pi_{\phi, \varphi}} \left[\nabla_{\phi, \varphi} \log \pi_{\phi, \varphi} R(\tau) \right], \quad (10)$$

where ϕ and φ denote the parameter for sub-agent I_ϕ and meta-agent G_ϕ , respectively. In RL-MARA, the policy networks of two agents are spliced together to be optimized together.

The solution can be approximated by a Monte Carlo estimator [21], i.e., $\nabla_{\phi, \varphi} J(\phi, \varphi) \propto \sum_{u=1}^U \sum_{t=1}^T \nabla_{\phi, \varphi} \log \pi_{\phi, \varphi} (p^{u,t} | d^{u,t}) R^{u,t}$, where U is the number of samples and T is the number of steps.

4.5 Discussion

It is important to point out two key differences between RL-MARA and mixture-of-experts (MoE) frameworks in IR [3, 59], which seem more straightforward in assembling existing ranking attack methods. For MoE methods: (i) The MoE requires optimization of each expert (attacker) during training. However, mainstream attack methods primarily use a series of search algorithms with two detached steps, finding vulnerable positions and adding perturbations, both of which are *non-trainable*. While it is feasible to design an algorithm for the first step, the second step poses a challenge, as generating discrete textual perturbations makes it difficult to obtain direct

supervised signals. (ii) Constrained by the gating, the outcome of MoE frameworks is determined by one of a single expert and a weighting of all experts, introducing a sense of *rigidity*.

In contrast, for RL-MARA: (i) RL-MARA integrates existing single-granular attack methods in a *trainable* manner. In addressing the two non-trainable steps present in existing attacks (see Section 3): to identify vulnerable positions, RL-MARA employs a sub-agent called vulnerability indicator (see Section 4.3.1); to perturb the discrete text space, RL-MARA uses a meta-agent called perturbation aggregator to master the perturbation addition strategy (see Section 4.3.2); and (ii) RL-MARA chooses a perturbation of any granularity at each step. Consequently, by the end of the attack, we can achieve either a single-granular perturbation or a combination of different levels of perturbation granularity with *flexibility*.

5 EXPERIMENTAL SETTINGS

5.1 Datasets

Benchmark datasets. Like previous work [31, 53], we conduct experiments on two benchmark datasets: (i) The **MS MARCO Document Ranking dataset** [35] (MS MARCO) is a large-scale benchmark dataset for Web document retrieval, with about 3.21 million documents. (ii) The **ClueWeb09-B dataset** [7] (ClueWeb09) comprises 150 queries with a collection of 50 million documents, with 242 additional queries from the TREC Web Track 2012 [8].

Target queries and documents. Following [5, 31], we randomly sample 1000 Dev queries from MS MARCO and use 242 additional queries from ClueWeb09 as target queries for each dataset evaluation, respectively. For each target query, we adopt two categories of target documents based on the top-100 ranked results of the target NRM, considering different levels of attack difficulty, i.e., *Easy* and *Hard*. Specifically, we randomly choose 5 documents ranked between [30, 60] as Easy target documents and select the 5 bottom-ranked documents as Hard target documents. In addition to the two types, we incorporate *Mixture* target documents for a thorough analysis. These consist of 5 documents randomly sampled from both the Easy and Hard target document sets.

5.2 Evaluation metrics

Attack performance. We use three automatic metrics: (i) Attack success rate (ASR) (%), which evaluates the percentage of target documents successfully boosted under the corresponding target query. (ii) Average boosted ranks (Boost), which evaluates the average improved rankings for each target document under the corresponding target query. (iii) Boosted top- K rate (TKR) (%), which evaluates the percentage of target documents that are boosted into top- K under the corresponding target query. The effectiveness of an adversary is better with a higher value for all these metrics.

Naturalness performance. Here, we use four metrics: (i) Spamicity detection, which detects whether target web pages are spam. Following [29, 53], we adopt a utility-based term spamicity detection method, OSD [58], to detect the after-attack documents. (ii) Grammar checking, which calculates the average number of errors in the after-attack documents. Specifically, we use, Grammarly [15], an online grammar checker following [29]. (iii) Language model perplexity (PPL), which measures the fluency using the average

perplexity calculated using a pre-trained GPT-2 model [41]. (iv) Human evaluation, which measures the quality of the after-attack documents following the criteria in [29, 53].

5.3 Models

Target NRMs. We choose three typical NRMs as target NRMs: (i) a pre-trained model, *BERT* [20]; (ii) a pre-trained model tailored for IR, *PROP* [32]; and (iii) a model distilled from the ranking capability of LLMs, *RankLLM* [49]. For RankLLM, we employ the model introduced by [49], distilling the ranking capabilities of an LLM, i.e., ChatGPT, into DeBERTa-large [18] in a permutation distillation manner on LLM-generated permutations within MS MARCO.

Baselines. We compare the following attack methods against NRMs: (i) **Term spamming (TS)** [17] randomly selects a starting position in the target document and replaces the subsequent words with terms randomly sampled from the target query. (ii) **PRADA** [53], **PLAT** [25] and **PAT** [29] are representative word-level, phrase-level and sentence-level ranking attack method against NRMs, introduced in Section 3. (iii) **IDEM** [5] is sentence-level ranking attack that inserts the generated connection sentence in the document.

Model variants. We implement three variants of RL-MARA: (i) **RL-MARA_{single}** selects a single level of perturbation for each document. After generating a vulnerability distribution, it sums confidence scores for each perturbation level. The one with the highest total score is chosen, while others are ignored and their scores are reset to zero. (ii) **RL-MARA_{triple}** incorporates all three perturbation levels in each document. It calculates average confidence scores for each level across vulnerable spans, ranks them from high to low, and then sequentially selects the highest-ranked span at each level until the term manipulation budget is reached, ensuring each perturbation level occurs at least once. (iii) **RL-MARA_{greedy}** greedily chooses vulnerable spans with the highest confidence for perturbation. It computes and ranks average confidence scores for vulnerable spans, regardless of granularity, and applies perturbations in descending order until the term manipulation budget is exhausted.

5.4 Implementation details

For MS MARCO and ClueWeb09, following [53], we truncate each document to 512. The initial retrieval is performed using the Anserini toolkit [55] with the BM25 model to obtain the top 100 ranked documents following [31, 53]. For the environment, following [29, 53], we use *BERT_{base}* as the surrogate model, and the training details are consistent with [31, 53]. For the reward, the balance hyperparameter β is 0.2 and the discount factor γ is 0.9.

For the perturbations, we set the term manipulation budget $\epsilon = 25$ for RL-MARA and all baselines. We specify a word length of 1 for word-level perturbations, 2 to 5 for phrase-level perturbations, and 6 to 10 for sentence-level perturbations. For the sub-agent, we supervise its training in a sequence labeling manner [51] to ensure that the labeling of each perturbation position is a continuous span. We set the learning rate to $3e^{-6}$ with Adam as the optimizer to train RL-MARA. Following [31], when the training process ends, we stop the updating of policy networks while running another epoch on the full dataset as a testing phase to evaluate the performance.

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

Method	MS MARCO								ClueWeb09							
	Easy				Hard				Easy				Hard			
	ASR	Boost	T10R	T5R	ASR	Boost	T10R	T5R	ASR	Boost	T10R	T5R	ASR	Boost	T10R	T5R
BERT																
TS	100.0	38.1	84.3	26.9	89.5	68.2	23.6	5.9	100.0	36.2	81.0	23.6	90.5	65.9	21.8	4.6
PRADA	98.3	26.1	69.3	18.3	78.9	55.9	9.6	1.8	97.6	24.8	66.9	16.5	77.1	53.9	8.2	1.2
PLAT	93.6	24.3	63.1	15.6	72.1	50.0	8.5	1.1	92.1	23.1	61.9	14.0	70.2	48.2	7.2	0.8
PAT	100.0	35.1	78.1	23.8	82.3	60.3	18.3	3.9	100.0	34.3	75.6	20.6	78.3	54.1	14.9	2.1
IDEM	100.0	39.6	85.6	26.8	90.2	69.6	25.8	7.2	100.0	37.1	82.6	24.9	87.2	65.2	22.1	5.1
RL-MARA _{single}	100.0	36.4	82.6	26.0	88.7	67.2	23.9	6.1	100.0	35.4	81.0	24.3	86.3	65.6	21.8	5.3
RL-MARA _{triple}	100.0	40.3	87.1	27.9	93.5	75.1*	30.6*	8.6*	100.0	38.9	85.7*	26.8	91.7*	73.2*	28.9*	7.8*
RL-MARA _{greedy}	100.0	40.8	88.3*	28.3*	94.6*	78.3*	31.3*	9.3*	100.0	38.8	86.1*	26.9*	93.7*	74.1*	29.2*	8.4*
RL-MARA	100.0	42.2*	90.2*	30.2*	98.9*	88.1*	36.9*	9.9*	100.0	40.1*	87.9*	28.6*	97.2*	85.8*	35.3*	9.5*
PROP																
TS	100.0	37.6	83.0	25.8	89.7	67.3	22.8	5.1	100.0	35.0	79.6	22.8	90.5	64.8	20.9	4.3
PRADA	95.2	23.4	66.6	16.4	75.8	53.4	8.6	1.2	93.4	22.5	63.4	14.1	74.9	51.2	6.8	0.9
PLAT	91.2	22.0	60.5	13.5	69.9	48.2	7.5	0.8	89.8	21.4	59.8	12.1	68.0	46.3	6.7	0.5
PAT	98.6	33.6	75.9	22.5	80.2	58.7	17.3	3.1	96.3	31.1	72.2	20.1	77.3	53.9	15.1	2.4
IDEM	100.0	37.3	83.0	25.0	87.9	67.5	24.0	6.5	100.0	35.8	80.1	23.1	85.8	65.1	21.7	4.1
RL-MARA _{single}	100.0	35.3	80.2	24.5	86.4	65.1	23.0	5.8	100.0	34.6	79.9	23.1	84.5	63.5	20.0	4.9
RL-MARA _{triple}	100.0	37.9	85.6	26.5	91.2	73.8*	28.9*	8.0*	100.0	37.5	83.8*	25.3*	90.1*	71.8*	27.2*	6.3*
RL-MARA _{greedy}	100.0	38.4	87.9*	27.6*	92.3*	77.6*	30.8*	8.7*	100.0	36.3	85.1*	25.4*	91.9*	72.5*	28.5*	7.9*
RL-MARA	100.0	41.0*	88.7*	28.9*	97.5*	87.0*	36.0*	9.1*	100.0	39.2*	85.6*	27.4*	95.8*	83.6*	33.8*	8.9*
RankLLM																
TS	100.0	34.3	79.4	22.5	89.8	63.9	19.7	3.2	99.2	32.1	73.0	19.3	89.9	59.8	18.4	2.8
PRADA	92.1	21.1	60.9	13.4	68.9	50.2	6.7	0.7	89.6	20.2	59.8	12.3	70.3	48.9	5.2	0.5
PLAT	88.9	19.1	55.8	11.5	63.5	45.9	5.9	0.5	85.6	18.6	55.9	10.0	64.8	42.6	5.2	0.3
PAT	95.6	30.2	72.1	19.8	75.6	54.3	14.9	2.8	93.9	28.7	68.6	18.4	73.5	49.8	12.8	1.7
IDEM	98.9	34.8	79.2	22.2	84.8	63.2	21.8	5.2	98.2	33.2	77.9	21.1	82.3	61.8	18.9	3.2
RL-MARA _{single}	97.9	33.6	77.9	22.3	83.8	62.6	21.2	4.3	97.5	32.1	76.7	21.2	81.8	60.2	18.1	3.7
RL-MARA _{triple}	99.8	35.8	82.9	24.3	89.0	71.8*	27.0*	7.1*	99.2	35.2	81.2*	22.8*	88.7*	69.8*	25.4*	5.8*
RL-MARA _{greedy}	100.0	36.2*	85.3*	25.1*	89.7*	74.8*	28.9*	8.1*	99.7	34.6	82.3*	22.7*	89.2*	70.1*	26.4*	7.2*
RL-MARA	100.0	39.7*	85.8*	27.0*	95.6*	85.0*	34.3*	8.6*	100.0	37.0*	82.4*	25.2*	92.1*	81.1*	31.3*	8.2*

6 EXPERIMENTAL RESULTS

6.1 Attack evaluation

Table 1 showcases the attack performance among three target NRMs with different attack methods, evaluated on both Easy and Hard target documents. We have the following observations: (i) Overall, the attack methods have effects on all three NRMs, exposing the prevalence of adversarial vulnerability. RankLLM is more resistant to adversarial attacks than other NRMs, indicating that distilling the ranking capabilities with LLMs helps to enhance the adversarial robustness of NRMs. (ii) The attack efficacy of most methods on ClueWeb09 is observed to be inferior compared to their performance on MS MARCO. This disparity may stem from the noise present in ClueWeb09’s documents, which potentially renders the model less responsive to adversarial perturbations. This observation aligns with previous findings reported in [31]. (iii) Hard documents exhibit lower ASR and TKR compared to Easy ones, due to the higher prevalence of irrelevant information in bottom-ranked documents, which challenges effective attacks with limited perturbation. (iv) Sentence-level attack methods yield better attack results than word-level and phrase-level methods. The reason may be that the

Table 2: The MRR@10 (%) performance on MS MARCO and ClueWeb09 of target NRMs (BERT, PROP and RankLLM) and their corresponding surrogate models (S_{NRM}).

NRM	BERT	S _{BERT}	RROP	S _{RROP}	RankLLM	S _{RankLLM}
MS MARCO	38.48	35.41	39.01	36.24	39.89	37.86
ClueWeb09	27.50	24.93	28.25	25.46	28.96	26.65

sentence-level perturbation is a continuous optimization of an entire vulnerable span in a document, thus increasing the likelihood of misleading the relevance judgment of NRMs to a greater extent. However, these sentence-level attack methods run the risk of being suspect due to naturalness flaws in respective aspects. We will discuss this further in Section 6.3.

RL-MARA significantly outperforms all baselines. In Hard documents of MS MARCO, while attacking RankLLM, RL-MARA improves over the best baseline, IDEM, by 65.4% in T5R and by 34.5% in Boost, highlighting the attack effect of perturbations at multiple levels of granularity. (i) The superiority of RL-MARA over RL-MARA_{single} suggests that the single granularity of perturbation is not sufficient to fully use the diverse vulnerability distributions in documents. Moreover, the synergy between perturbations of different levels of granularity can lead to more threatening adversarial

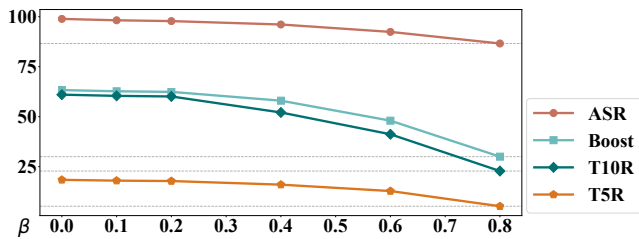


Figure 4: The impact of hyper-parameter β on the attack performance of RL-MARA against RankLLM on MS MARCO.

examples. (ii) The advantage of RL-MARA over RL-MARA_{triple} indicates that the sequential decision-making in adding perturbations of each granularity, based on RL rather than mechanically applying perturbations across all levels of granularity, enables more effective exploitation of each document’s vulnerability distribution. (iii) The improvement of RL-MARA over RL-MARA_{greedy} demonstrates that the cooperative approach of two agents in flexibly organizing multi-granular perturbations is instrumental in generating high-quality perturbation sequences, posing significant threats to NRMs.

For the Mixture target documents, the performance of all attack methods remains consistent with that observed in each NRM for Easy and Hard documents. Even attacking the most defensive RankLLM on MS MARCO, RL-MARA outperforms the best baseline by 34.8% in T5R and 25.1% in Boost. The complete experimental results are at <https://github.com/Davion-Liu/MARA>.

Furthermore, the performance of the surrogate ranking model plays an important role in the success of the black-box attack. As shown in Table 2, for all target NRMs, the corresponding surrogate model can imitate their performance to some extent. This allows the vulnerabilities identified on the surrogate ranking model to be effectively transferred to the target NRM.

The impact of the balance hyper-parameter. β is an important hyper-parameter in our multi-granular reward, since it balances the attack performance and the naturalness of adversarial example. We investigate the impact of β on the performance of RL-MARA. Lower values of β imply a less emphasis on the naturalness and a greater emphasis on the attack effectiveness. We take attacking RankLLM on Mixture documents of MS MARCO as an example. The trend of the attack performance with β is shown in Figure 4. As β decreases, the attack performance gradually increases and plateaus, since the naturalness reward no longer dominates the update of the attack strategy. However, trivializing naturalness reward signals can lead to susceptibility to suspicion, as discussed in Section 6.3.

When serving as a benchmark, RL-MARA can create adversarial examples of varying naturalness by tuning the hyper-parameter β , enabling a comprehensive analysis for model robustness.

6.2 Capability of surrogate models

Black-box vs. white-box attack. We further focus on the white-box scenario, which is valuable for enhancing understanding of our method. In white-box attacks, we directly substitute the surrogate ranking model as the target NRM and keep other components the same in RL-MARA. We consider the most defensive RankLLM, which has a 4.4% higher ranking performance than its surrogate model, as shown in Table 2. The result on the Mixture target documents of MS MARCO is shown in Figure 5 (Left), with similar

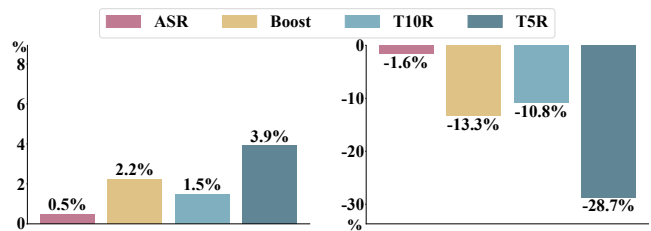


Figure 5: (Left): Attack performance changes of RL-MARA against RankLLM on MS MARCO in the white-box setting, compared to black-box setting. (Right): Attack performance changes of RL-MARA against RankLLM on MS MARCO in the OOD scenario, compared to the IID scenario.

Table 3: The online grammar checker, perplexity, and human evaluation results for attacking RankLLM on MS MARCO.

Method	Grammar	PPL	Impercept.	κ	Fluency	<i>Kendall</i>
Original	59	44.2	0.90	0.52	4.57	0.65
TS	67	59.7	0.06	0.53	2.42	0.73
PRADA	108	113.4	0.53	0.48	3.21	0.82
PLAT	98	86.3	0.62	0.58	3.30	0.74
PAT	83	72.1	0.58	0.61	3.42	0.76
IDEM	67	55.6	0.79	0.46	3.75	0.82
RL-MARA _{NC}	103	89.5	0.58	0.47	2.97	0.69
RL-MARA	65	53.5	0.87	0.58	3.90	0.76

findings on other target documents. Compared with the white-box setting, RL-MARA still obtains competitive performance in black-box scenarios. The results demonstrate that the training method of the surrogate model is sufficient to simulate the vulnerability performance of the target NRM at different levels of granularity, thus making our multi-granular attack effects transferable.

Training a surrogate model in the out-of-distribution (OOD) setting. In our experiments, the training data of the surrogate model is directly adopted from the Eval queries of the target model, i.e., the same distribution as the query used to train the target NRM. However, in realistic search scenarios, obtaining an identically distributed query set is difficult. Following [5], we show the results for the IID and OOD scenarios in Figure 5 (Right). Specifically, we take the Mixture target documents as an example and evaluate the attack performance against RankLLM on MS MARCO. For our IID scenarios, we use Eval queries of MS MARCO for surrogate model training. For OOD scenarios, we use Eval queries of Natural Questions (NQ) [23] to train the surrogate model and observe a 25.7% decrease on MRR@10 relative to the IID surrogate model. The results reveal that despite compromised attack performance when the IID data is unavailable, RL-MARA continues to perform an effective attack method that can identify model vulnerabilities.

6.3 Naturalness evaluation

Next, we report on the naturalness of generated adversarial examples against RankLLM on MS MARCO, with similar findings on ClueWeb09. Here, we set the balance hyper-parameter β to 0 of RL-MARA, denoted as RL-MARA_{NC}, for comparison.

Grammar checking, PPL, and human evaluation. Table 3 lists the results of the automatic grammar checker, PPL, and human evaluation. For human evaluation, we recruit five annotators to annotate 32 randomly sampled Mixture adversarial examples from

Table 4: The detection rate (%) via a representative anti-spamming method for attacking RankLLM on MS MARCO.

Threshold	0.08	0.06	0.04	0.02
TS	40.3	53.4	76.3	94.5
PRADA	13.4	22.6	40.1	64.2
PLAT	11.6	20.6	32.5	58.7
PAT	9.3	15.2	26.4	49.8
IDEM	17.2	29.4	48.3	72.9
RL-MARA _{NC}	15.8	26.3	42.4	68.6
RL-MARA	6.2	10.6	20.2	40.3

each attack method [5]; The annotators score the *Fluency* from 1 to 5; higher scores indicate more fluent examples; the *Imperceptibility* is to determine whether an example is attacked (0) or not (1); including the annotation consistency (the *Kappa* value and *Kendall's Tau* coefficient) [31, 53]. We observe that: (i) TS demonstrates subpar performance in all naturalness evaluations because it abruptly inserts the query terms into the document without regard to semantic coherence. (ii) The single-granular attack methods fall short when compared to the original documents, a possible reason is that *a priori* restricting perturbations to a specific level of granularity carries the risk of generating unnatural perturbations. (iii) Although discarding the fluency constraint may improve attack efficacy, the adversarial examples produced by RL-MARA_{NC} tend to arouse suspicion. (iv) RL-MARA outperforms the baselines, demonstrating the effectiveness of the naturalness reward provided by the LLM.

Spamcidity detection. Table 4 shows the automatic spamcidity detection results on Mixture documents with similar findings on other target documents. If the spamcidity score of a document exceeds the detection threshold, it is identified as suspected spam content. Even with the effective attack performance, IDEM is easier to detect among other single-granular attack baselines because it ignores the avoidance of query terms in prompting the language model to generate perturbed sentences. RL-MARA outperforms the baselines significantly ($p\text{-value} \leq 0.05$), demonstrating the similarity reward provided by an LLM is effective in preventing attack methods from abusing query terms and thus being suspected.

7 RELATED WORK

Neural ranking models. Ranking models have evolved from early heuristic models [45] to probabilistic [39, 43], and modern learning-to-rank models [27, 30]. NRMs [9, 16, 37] emerged with deep learning and demonstrated excellent ranking performance through powerful relevance modeling capabilities. Research has also explored integrating potent pre-trained language models into ranking tasks, including tailoring specific pre-training [32], fine-tuning [36], and distilling techniques [49], leading to state-of-the-art performance [13]. However, these NRMs also exhibit disconcerting adversarial vulnerabilities inherited from neural networks [42, 52].

Adversarial ranking attacks. In the field of IR, the challenge of black-hat search engine optimization (SEO) has been significant since the inception of (web) search engines [17]. Typically, the goal of black-hat SEO is to boost a page's rank by maliciously manipulating documents in a way that is unethical with search engine guidelines [4, 22]. It usually causes erosion of search quality and deterioration of the user experience. To safeguard NRMs from exploitation, research has focused on adversarial ranking attacks

[29, 53], in order to expose the vulnerability of NRMs. The goal of adversarial ranking attacks is to manipulate the target document through imperceptible perturbations to improve its ranking for target queries [31, 53]. Previous studies have mainly explored attacks against NRMs with a single granularity of perturbations, e.g., word-level [31, 42, 52, 53] and sentence-level [5, 29, 31, 47]. RELEVANT [31] uniquely addressed two granularities individually for ranking attacks against NRMs and observed different attack effects. However, these efforts lack the flexibility needed for the comprehensive exploitation of NRM vulnerabilities. In this paper, we launch attacks with perturbations at multiple levels of granularity.

Multi-granular adversarial attack. Multi-granular attacks, in contrast to single-granular ones, engage different levels of perturbation granularity [56]. In CV [19, 54] and NLP [6, 11, 28], they exploit model vulnerabilities by freely combining varying perturbation granularities to produce threatening adversarial examples. However, in NLP, existing multi-granular attacks are mainly confined to selecting one granularity from several options in a predetermined manner [6, 11]. Unlike these studies, we combine perturbations at multiple levels of granularity, allowing them to coexist in an adversarial example to enhance the adversarial ranking attacks.

8 CONCLUSION

In this work, we investigated a multi-granular ranking attack framework against black-box NRMs. We modeled multi-granular attacks as a sequential decision-making process and proposed a reinforcement learning-based framework, RL-MARA, to address it. Our extensive experimental results reveal that the proposed method can effectively boost the target document through multi-granularity perturbations with imperceptibility. In future work, we would like to explore how to efficiently optimize the combinatorial explosion problem due to multi-granular attacks from a theoretical perspective. Beyond the current single-granular attack methods, we will explore integrating more granularity and different attack methods into our framework and making them learnable. Our method proves effective against NRMs distilled from LLMs, as exemplified by RankLLM. However, directly attacking LLMs as rankers and, in turn, leveraging their advanced capabilities for attacks presents promising future research avenues.

ACKNOWLEDGMENTS

This work was funded by the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Key Research and Development Program of China under Grants No. 2023YFA1011602 and 2021QY1701, the National Natural Science Foundation of China (NSFC) under Grants No. 62372431, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039. This research also was (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and the European Union's Horizon Europe program under grant agreement No. 101070212.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

- [1] Richard Bellman. 1957. A Markovian Decision Process. *Journal of mathematics and mechanics* (1957), 679–684.
- [2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *ICLR*.
- [3] Yinqiong Cai, Yixing Fan, Keping Bi, Jiafeng Guo, Wei Chen, Ruqing Zhang, and Xueqi Cheng. 2023. CAME: Competitively Learning a Mixture-of-Experts Model for First-stage Retrieval. *arXiv preprint arXiv:2311.02834* (2023).
- [4] Carlos Castillo and Brian D. Davison. 2011. Adversarial Web Search. *Foundations and Trends in Information Retrieval* 4, 5 (2011), 377–486.
- [5] Xuanang Chen, Ben He, Zheng Ye, Le Sun, and Yingfei Sun. 2023. Towards Imperceptible Document Manipulations against Neural Ranking Models. In *ACL*. 6648–6664.
- [6] Yangyi Chen, Jin Su, and Wei Wei. 2021. Multi-granularity Textual Adversarial Attack with Behavior Cloning. In *EMNLP*. 4511–4526.
- [7] Charles L Clarke, Nick Craswell, and Ian Soboroff. 2009. *Overview of the TREC 2009 Web Track*. Technical Report. Waterloo University.
- [8] Charles L Clarke, Nick Craswell, and Ellen M Voorhees. 2012. *Overview of the TREC 2012 Web Track*. Technical Report. NIST Gaithersburg MD.
- [9] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*.
- [10] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *SIGIR*. 65–74.
- [11] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *ACL*. 31–36.
- [12] Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling Diverse Relevance Patterns in Ad-hoc Retrieval. In *SIGIR*. 375–384.
- [13] 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.
- [14] Xuanjie Fang, Sijie Cheng, Yang Liu, and Wei Wang. 2023. Modeling Adversarial Attack on Pre-trained Language Models as Sequential Decision Making. In *Findings of the Association for Computational Linguistics*. 7322–7336.
- [15] Grammarly. 2023. <https://app.grammarly.com/>.
- [16] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM*. 55–64.
- [17] Zoltan Gyongyi and Hector Garcia-Molina. 2005. Web Spam Taxonomy. In *AIRWeb*.
- [18] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *ICLR*.
- [19] Xin Jin, Ruxin Wang, Shin-Jye Lee, Qian Jiang, Shaowen Yao, and Wei Zhou. 2023. Adversarial Attacks on Multi-focus Image Fusion Models. *Computers & Security* 134 (2023), 103455.
- [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] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *ECML*.
- [22] Oren Kurland and Moshe Tennenholtz. 2022. Competitive Search. In *SIGIR*.
- [23] 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), 453–466.
- [24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521, 7553 (2015), 436–444.
- [25] Yibin Lei, Yu Cao, Dianqi Li, Tianyi Zhou, Meng Fang, and Mykola Pechenizkiy. 2022. Phrase-level Textual Adversarial Attack with Label Preservation. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 1095–1112.
- [26] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*. 7871–7880.
- [27] Hang Li. 2014. Learning to Rank for Information Retrieval and Natural Language Processing. *Synth. Lect. Hum. Lang. Technol.* 7, 3 (2014), 1–121.
- [28] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. In *NDSS*.
- [29] 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.
- [30] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.
- [31] 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*. 1700–1709.
- [32] 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.
- [33] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW*.
- [34] Duc Thien Nguyen, Akshat Kumar, and Hoong Chui Lau. 2017. Collective Multiagent Sequential Decision Making under Uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
- [35] 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*.
- [36] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
- [37] Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altinoglu, Md. Mustafizur Rahman, Pinar Karagoz, Alexander Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Thanh Nguyen, Dan Xu, Byron C. Wallace, Maarten de Rijke, and Matthew Lease. 2018. Neural Information Retrieval: At the End of the Early Years. *Information Retrieval* 21, 2–3 (2018), 111–182.
- [38] OpenAI. 2022. Introducing ChatGPT. <https://openai.com/blog/chatgpt>.
- [39] Jay Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. *SIGIR*.
- [40] Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating Adversarial Misspellings with Robust Word Recognition. In *ACL*. 5582–5591.
- [41] 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.
- [42] Nisarg Raval and Manisha Verma. 2020. One Word at a Time: Adversarial Attacks on Retrieval Models. *arXiv preprint arXiv:2008.02197* (2020).
- [43] 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.
- [44] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning Internal Representations by Error Propagation*. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.
- [45] Gerard Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18, 11 (1975), 613–620.
- [46] Suranjana Samanta and Sameep Mehta. 2017. Towards Crafting Text Adversarial Samples. *arXiv preprint arXiv:1707.02812* (2017).
- [47] Congzheng Song, Alexander M. Rush, and Vitaly Shmatikov. 2020. Adversarial Semantic Collisions. In *EMNLP*.
- [48] 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.
- [49] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *EMNLP*. 14918–14937.
- [50] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.
- [51] Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and Practical BERT Models for Sequence Labeling. In *EMNLP*. 3632–3636.
- [52] Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. BERT Rankers are Brittle: A Study using Adversarial Document Perturbations. In *ICTIR*.
- [53] 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.
- [54] Guoming Wu, Yangfan Xu, Jun Li, Zhiping Shi, and Xianglong Liu. 2023. Imperceptible Adversarial Attack with Multi-granular Spatio-temporal Attention for Video Action Recognition. *IEEE Internet of Things Journal* (2023).
- [55] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *Journal of Data and Information Quality* 10, 4 (2018), Article 16.
- [56] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey. *ACM TIST* 11, 3 (2020), 1–41.
- [57] Xiaoqing Zheng, Jiehang Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuan-Jing Huang. 2020. Evaluating and Enhancing the Robustness of Neural Network-based Dependency Parsing Models with Adversarial Examples. In *ACL*. 6600–6610.
- [58] Bin Zhou and Jian Pei. 2009. OSD: An Online Web Spam Detection System. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, Vol. 9.
- [59] Xinyu Zou, Zhi Hu, Yiming Zhao, Xuchu Ding, Zhongyi Liu, Chenliang Li, and Aixin Sun. 2022. Automatic Expert Selection for Multi-Scenario and Multi-Task Search. *arXiv preprint arXiv:2205.14321* (2022).