Attack-in-the-Chain: Bootstrapping Large Language Models for Attacks Against Black-Box Neural Ranking Models

Yu-An Liu[1,2], Ruqing Zhang[1,2], Jiafeng Guo[1,2*], Maarten de Rijke[3], Yixing Fan[1,2], Xueqi Cheng[1,2]

[1]CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]University of Amsterdam, Amsterdam, The Netherlands
{liuyuan21b, zhangruqing, guojiafeng, fanyixing, cxq}@ict.ac.cn, m.derijke@uva.nl

Abstract

Neural ranking models (NRMs) have been shown to be highly effective in terms of retrieval performance. Unfortunately, they have also displayed a higher degree of sensitivity to attacks than previous generation models. To help expose and address this lack of robustness, we introduce a novel ranking attack framework named Attack-in-the-Chain, which tracks interactions between large language models (LLMs) and NRMs based on chain-of-thought (CoT) prompting to generate adversarial examples under black-box settings. Our approach starts by identifying anchor documents with higher ranking positions than the target document as nodes in the reasoning chain. We then dynamically assign the number of perturbation words to each node and prompt LLMs to execute attacks. Finally, we verify the attack performance of all nodes at each reasoning step and proceed to generate the next reasoning step. Empirical results on two web search benchmarks show the effectiveness of our method.

1 Introduction

Neural ranking models (NRMs) are remarkably effective at ranking (Dai and Callan 2019; Guo et al. 2016; Nogueira and Cho 2019; Yan et al. 2021; Yu et al. 2019) in information retrieval (IR). But they have also shown vulnerability to carefully crafted adversarial examples (Raval and Verma 2020; Song, Rush, and Shmatikov 2020; Wang, Lyu, and Anand 2022). This vulnerability raises concerns when deploying NRMs in environments susceptible to black-hat search engine optimization (SEO) (Gyongyi and Garcia-Molina 2005). To prevent the exploitation of NRMs, research has focused on the study of adversarial ranking attacks, which aim to deceive NRMs by promoting a low-ranked target document to a higher position in the ranked list for a query through human-imperceptible perturbations (Chen et al. 2023; Liu et al. 2022, 2023; Wu et al. 2023).

Large language models (LLMs) (Achiam et al. 2023; Jiang et al. 2023; Touvron et al. 2023; Chen et al. 2024) have shown strong abilities in understanding, reasoning, and interaction. These abilities have enabled LLMs to achieve strong performance in adversarial attacks in natural language processing (NLP) (Chao et al. 2023; Xu et al. 2023b). These efforts demonstrate the potential of LLMs to reveal the vulnerability of neural models. However, adversarial ranking attacks in IR differ from attacks in NLP as they face a multi-step "battle" with every top-ranked document in the list to improve their rankings while ensuring efficiency. Using the capabilities of LLMs for adversarial ranking attacks remains a challenging and unresolved task.

Inspired by chain-of-thought (CoT) prompting (Wang et al. 2023a; Wei et al. 2022; Xu et al. 2023a; Cohn et al. 2024), we develop *attack-in-the-chain* (AttChain), which uses multiple NRM-LLM interaction rounds to effectively achieve attacks. Each node in the reasoning chain targets phased ranking improvements, gradually evolving the target document into a fluent and imperceptible adversarial example. To this end, we need to resolve two key challenges.

First, *how to identify nodes in the reasoning chain guiding ranking improvement?* We define an *anchor document* to be a document with a higher ranking position than the current perturbed document in the returned list. Each anchor document serves as a node to guide the ranking improvement of the target document. When the target document achieves a higher ranking, the nodes are updated accordingly to provide further guidance. Considering all top-ranked documents as nodes increases the computational effort and can be misleading. We design a Zipf-based filtering strategy, in which higher-ranked documents are more likely to be retained as candidate documents and selected as anchor documents by LLMs based on previous interactions. This allows the target document to obtain sufficient informational guidance on its path to improving its ranking.

Second, *how to perturb the target document based on the anchor document node in the reasoning chain?* We design a discrepancy-oriented assignment function to dynamically assess the number of perturbation words at each step. The key idea is that the degree of perturbation to the target document should be flexibly decided according to its ranking position relative to the anchor document: if the ranking discrepancy is large (small), a large (small) number of perturbation words is needed. We then instruct LLMs to generate perturbations via carefully crafted prompts such that the target documents are ranked higher while keeping the perturbations imperceptible. Finally, we verify the perturbed documents of all nodes at each reasoning step and select the most

*Jiafeng Guo is the corresponding author.

effective node to initiate the next attack steps. This enables LLMs to dynamically modify the reasoning direction.

Following (Chen et al. 2023; Liu et al. 2024a; Wu et al. 2023), we focus on a decision-based black-box setting (Brendel, Rauber, and Bethge 2018), where the adversary lacks direct access to model information and can only query the target NRM then receive the ranked list. We employ GPT-3.5 (OpenAI 2022) and Llama3 (Meta 2024) as attackers and conduct experiments on two web search benchmark datasets, MS MARCO Document Ranking (Nguyen et al. 2016) and TREC DL19 (Craswell et al. 2019). The results show that our method significantly outperforms state-of-the-art attack methods in both attack effectiveness and imperceptibility. Our proposed method avoids training surrogate models and requires only limited access to the target NRM, thereby reducing the likelihood of detection. Furthermore, the vulnerabilities of NRMs revealed LLMs can inspire the development of corresponding countermeasures.

2 Related Work

Adversarial attacks against neural ranking models. Adversarial ranking attacks are meant to deceive NRMs to promote a low-ranked target document to a higher position in the ranked list produced for a given query by introducing human-imperceptible perturbations (Chen et al. 2023; Liu et al. 2022, 2023; Wu et al. 2023). Depending on whether knowledge of the target model can be accessed, the attack task is categorized into white-box and black-box settings (Papernot et al. 2017). For practical considerations, existing efforts focus on black-box settings, mainly using the following core steps: (i) training the surrogate model, (ii) identifying vulnerable positions, and (iii) perturbing identified positions (Liu et al. 2022, 2024a; Wu et al. 2023; Liu et al. 2024b). Existing work has explored different ways to generate adversarial samples, including the adaptation of textual attacks (Liu et al. 2022; Wu et al. 2023), reinforcement learning (Liu et al. 2023, 2024a), and direct generation using language models (Chen et al. 2023). With the emergence of LLMs, the scope of research has expanded to include ranking attacks against LLM-based NRMs (Liu et al. 2024a). Here, we explore how LLMs can be used to achieve effective attacks against various black-box NRMs. The proposed method avoids training surrogate models, reducing access to the target model. We hope this method can broaden the understanding and mitigation strategies of such vulnerabilities.

Adversarial attack with LLMs. Due to their strong generation abilities, LLMs have been used to conduct adversarial attacks and have demonstrated excellent performance. Raina, Liusie, and Gales (2024) and Xu et al. (2024b) explore using LLMs to generate adversarial examples to attack language models, causing them to produce misleading results. Gadyatskaya and Papuc (2023) and Xu et al. (2024a) investigate using LLMs to plan attack steps and implement automated attacks. In this work, we study the interaction between LLMs and NRMs, designing chains of reasoning to iteratively perturb target documents with the ultimate goal of maximizing ranking improvements.

Chain-of-thought prompting. CoT (Wei et al. 2022) is a technique designed to enhance the reasoning abilities of LLMs by guiding them through a step-by-step process in a few-shot setting. CoT prompting has proven effective across a variety of tasks, including question answering (Wang et al. 2023a; Ji et al. 2024), solving mathematical puzzles (Giadikiaroglou et al. 2024; Madaan and Yazdanbakhsh 2022), executing tool calls (Paranjape et al. 2023), and evaluating performance (Lanham et al. 2023). In the field of IR, there have been successful attempts by researchers to use CoT to guide retrieval (Wang et al. 2023b; Xu et al. 2023a; Yu et al. 2023). Our work diverges from studies that use LLMs to enhance reasoning cooperatively by using CoT reasoning steps for credible, traceable retrieval results. Instead, we deploy LLMs adversarially, performing interactive attacks against NRMs in the reasoning tree to explore effective perturbation strategies and generate high-quality adversarial examples.

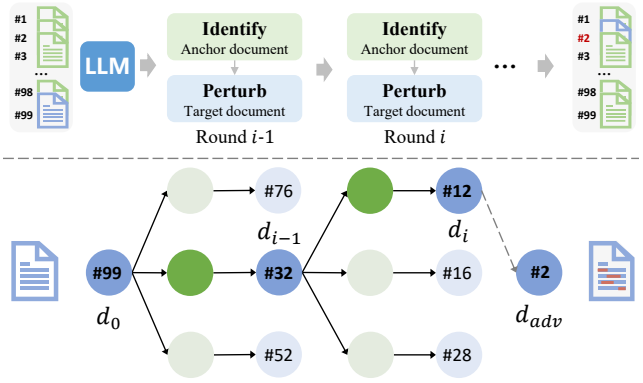

Figure 1: The framework of the proposed method AttChain.

3 Method

As shown in Figure 1, for each step in the reasoning chain, the proposed attack method AttChain (i) first selects several anchor documents as nodes based on the ranking of the current perturbed document (Section 3.1); and (ii) then adds perturbations to the target document according to the ranking difference with the anchor document and modifies the reasoning direction (Section 3.2). The code is available at https://github.com/Davion-Liu/AttChain.

3.1 Identify Nodes

Motivation. The key idea of this stage is to identify the nodes in the reasoning chain that can guide the LLM to find the direction of boosting the ranking of target document d. A natural idea is to use documents that are ranked higher than the target document, i.e., anchor documents, as nodes to provide attack direction. Treating *all* top-ranked documents as nodes complicates the reasoning chain, which impairs the clarity of reasoning path and increases computational costs.

A simple method is to directly take the top-1 document as the node. However, this may not be optimal since it assumes that one document is enough to provide sufficient information to favor an attack. Ideally, we need a way to filter out a lean and varied candidate set of anchor documents for the

LLM to choose from. Here, we introduce a filter-then-select pipeline to decide the anchor documents on the nodes.

Zipf distribution-based document filtering. The aim of this step is to select a set of candidate anchor documents C_A, where documents ranked higher are more likely to be retained. To achieve this, we propose a filtering strategy based on the Zipf distribution (Zipf 2016), inspired by human click behavior across search engine result pages (Wu, Jiang, and Zhang 2012). Recall that a Zipf distribution is specified by a rank-frequency distribution where the frequency of any element being sampled is inversely proportional to its ranking. The Zipf distribution can be mathematically defined as $P(r; s) \propto r^{-s}$, where r is the rank, and s is the exponent characterizing the distribution.

In the context of our method, at the i-th reasoning step, we are given the target NRM f, which generates a ranked list L for query q, and the perturbed target document at the current step d_{i-1}. $L[:]$ denotes a slice of the list L, where $L[:\mathrm{Rank}(f, q, d_{i-1})]$ includes all documents of L up to the rank position of d_{i-1}. Following this, the candidates C_A with m anchor documents are filtered as follows:

$$C_A = \mathrm{Zipf}(L[:\mathrm{Rank}(f, q, d_{i-1})], m, s), \tag{1}$$

where $\mathrm{Rank}(\cdot)$ is the ranking position of d_{i-1}.

Anchor document decision prompt. Given the filtered set C_A, our target is to prompt the LLM to select final anchor documents that are considered helpful for the attack. Considering input length limitations, for each reasoning step, we (i) first concatenate the title and the first three sentences of each document in the candidate anchor documents C_A; and (ii) then use an *anchor document selection prompt* (Table 1) to organize the text of documents and prompt the LLM to decide n final anchor documents ($n \leq m$) as the nodes.

Anchor document selection prompt:
You are a search engine optimization specialist aiming to boost the ranking of your target document under the target query. You will receive a target query, a target document, and m anchor documents. Please select the n anchor documents that are most useful for improving the target document's ranking under the target query, that is, the ones most worthy of reference. You have moved up x places in the rankings. Please refer to the previous step and output the id of the anchor documents you have selected and separate the ids by "\n". Follows are target query, target document, and m anchor documents, give you output: {Target query} {Target document} {Anchor documents} \n Output:

Target document perturbation prompt:
You are tasked as a search engine optimization specialist to enhance the relevance of a target document with respect to a target query. Your goal is to strategically modify the target document to improve its ranking in search results. With the previous step, you have moved up x places in the rankings. Instructions:
1. You are provided with a "target query", a "target document", and an "anchor document".
2. Your task is to modify $|p_i'|$ words in the target document.
3. Implement the following strategies:
a. Identify key phrases or words relevant to the target query from the anchor document.
b. Combine these key phrases appropriately considering the target query, modify and integrate them into the target document.
c. Prioritize earlier sections of the document for these changes.
4. Please output the perturbed target document in <document></document> and point out the words you changed and where they are taken from the anchor document:
Input: {Target query} {Anchor document} {Target document}
Please output the modified target document enclosed in <document>tags:

Table 1: The anchor document selection prompt and target document perturbation prompts.

3.2 Perturb Documents at Nodes

Motivation. The key idea is to add perturbations to the target documents based on each node separately at each reasoning step, verifying the most effective node based on the document ranking improvement. To achieve this, we first assign the number of words to be perturbed in the target document, then we use the LLM to generate the perturbation and update the target document.

Discrepancy-oriented perturbation assignment. Intuitively, the gap in ranking position between the target and anchor documents can be bridged by adding perturbations to the target document. The larger the gap, the more perturbations are needed. Therefore, we design a discrepancy-oriented perturbation assignment method that assigns more perturbations to larger ranking discrepancies while ensuring that the total perturbations remain within the budget. Specifically, at the i-th reasoning step, the number of words to be perturbed $|p_i^j|$ in the perturbed target document d_i according to anchor document d^j is calculated by:

$$|p_i^j| = \frac{\left(\mathrm{Rank}\left(f, q, d_i\right) - \mathrm{Rank}\left(f, q, d^j\right)\right)}{\mathrm{Rank}\left(f, q, d\right)}\,\epsilon, \tag{2}$$

where ϵ is the budget for the number of manipulated words for the entire attack.

Target document perturbation prompt. After obtaining the number of words to be perturbed $|p_i^j|$, we perturb the target document d_{i-1} according to anchor document d^j under the assigned word number $|p_i^j|$. We (i) first use the *target document perturbation prompt*, shown in Table 1, to guide the LLM in generating perturbations to the target document; (ii) then evaluate the attack effectiveness based on each anchor document at the current nodes; and (iii) finally adopts the node that gives the highest rank improvement and uses it to identify the next round of nodes.

The process of identifying nodes and updating nodes is done iteratively. During this process, the ranking of the target document is progressively increasing in a ladder-climbing manner. Taking into account the computational overhead and effectiveness, the reasoning process executes a total of five rounds. Then, the final target document is obtained as an adversarial example.

4 Experimental Settings

In this section, we introduce our experimental settings.

4.1 Datasets

Benchmark datasets. Following (Liu et al. 2022; Wu et al. 2023), we conduct experiments on two datasets: (i) The

MS MARCO Document Ranking (Nguyen et al. 2016) (MS MARCO) is a large-scale dataset for web document retrieval, with 3.21 million documents. (ii) The document ranking task of **TREC Deep Learning Track 2019** (TREC2019; Craswell et al. 2019), which comprises 200 queries.

Target queries and documents. Following (Chen et al. 2023; Liu et al. 2022), we randomly sample 1,000 Dev queries from MS MARCO and 100 queries from TREC2019 as target queries for each dataset evaluation, respectively. For each target query, we adopt *Easy* and *Hard* target documents based on the top-100 ranked results from the target NRM. 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.

4.2 Evaluation Metrics

Attack performance. We adopt three types of 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; and (iii) Boosted top-10 rate (T10R) (%), which evaluates the percentage of target documents boosted into the top-10 under the corresponding target query. The attack performance of an adversary is better with a higher value for all three metrics.

Naturalness performance. We use five metrics: (i) *Qrs*, which is the average number of queries to the target NRM; (ii) *spamicity detection*, which detects whether target documents are spam; following (Liu et al. 2022; Wu et al. 2023), we use the utility-based term spamicity detection method OSD (Zhou and Pei 2009) to detect the adversarial examples; (iii) *grammar checking*, which calculates the average number of errors in the adversarial examples with an online grammatical checker; following (Chen et al. 2023; Liu et al. 2022), we use Grammarly[1] for grammar checking; (iv) language model perplexity (*PPL*), which measures the fluency of adversarial examples using the average perplexity calculated using a pre-trained GPT-2 (Radford et al. 2019); and (v) *human evaluation*, which measures the imperceptibility of the adversarial examples following the criteria in (Liu et al. 2022; Wu et al. 2023).

4.3 Target NRMs

Following (Chen et al. 2023; Liu et al. 2022; Wu et al. 2023), we select three typical NRMs as target NRM: (i) BERT; (ii) *PROP* (Ma et al. 2021) is a pre-trained model tailored for ranking; and (iii) *RankLLM* (Sun et al. 2023) is a model distilled from the ranking capability of an LLM, i.e., ChatGPT into DeBERTa-large (He et al. 2020).

[1]https://app.grammarly.com/

4.4 Baseline Methods

Baselines. (i) **Term spamming (TS)** (Gyongyi and Garcia–Molina 2005) replaces words with query terms in the target document at a randomly selected position. (ii) **PRADA** (Wu et al. 2023) substitutes words in the document with synonym to perform ranking attack against NRMs. (iii) **PAT** (Liu et al. 2022) generates a trigger at the beginning of the document for attacks. (iv) **IDEM** (Chen et al. 2023) inserts connecting sentences linking original sentences in the document to improve its ranking.

Model variants. We employ the *gpt-3.5-turbo-1106* API provided by OpenAI (OpenAI 2024a) and *Llama-3-8B* (Meta 2024) as LLM-based attackers, denoted as $\textbf{AttChain}_{\mathrm{GPT}}$ and $\textbf{AttChain}_{\mathrm{Llama}}$, respectively. Then, based on $\textbf{AttChain}_{\mathrm{GPT}}$, we consider two variants: (i) $\textbf{Att Chain}_{-\mathrm{CoT}}$ rotates the top five documents as anchor documents rather than relying on LLMs to identify them. (ii) $\textbf{AttChain}_{-\mathrm{dynamic}}$ statically adds the same degree of perturbation in each round, regardless of the ranking position gap with the anchor document.

4.5 Implementation Details

The initial retrieval step is performed with the BM25 model to obtain the top 100 ranked documents following (Liu et al. 2023; Wu et al. 2023). For anchor document filtering, we set $m = 20$, $n = 5$, and $s = 2$. For the perturbations, we set the budget for the number of manipulated words ϵ to 25. For human evaluation, we recruit three annotators to annotate 50 randomly sampled adversarial examples and the corresponding documents of each attack method (Liu et al. 2022). Following (Wu et al. 2023), annotators judge whether an example is attacked (labeled as 0) or not (labeled as 1) as the imperceptibility score. We repeated our experiment 3 times on 4 $\times$ Tesla V100 32G to get the average results.

5 Experimental Results

In this section, we report the experimental results to demonstrate the effectiveness of our proposed method.

5.1 Attack Evaluation

First, we evaluate the *ranking performance* of the target NRM over both datasets. For MS MARCO, the ranking performance (MRR@10) of BERT, PROP, and RankLLM is 0.385, 0.389, and 0.399, respectively. For TREC2019, the ranking performance (nDCG@10) of BERT, PROP, and RankLLM is 0.608, 0.622, and 0.646, respectively.

The *attack performance* comparisons between AttChain and the baselines are shown in Table 2. The performance on the Mixture level target documents is shown in Appendix A. We have the following observations: (i) Both NRMs are vulnerable to adversarial attacks, while RankLLM has relatively better adversarial robustness. This demonstrates that LLMs mitigate the vulnerability of NRMs, as observed in (Liu et al. 2024a). (ii) The performance of PRADA is not as good as other baselines. This attack method does not exploit the understanding capability of the language model, thus making it difficult to continuously optimize the entire target document. (iii) The attack effectiveness of PAT and IDEM

Method	MS MARCO						TREC2019					
	Easy			Hard			Easy			Hard		
BERT	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R
TS	100.0	38.1	84.3	89.5	68.2	23.6	100.0	36.2	81.0	90.5	65.9	21.8
PRADA	98.3	26.1	69.3	78.9	55.9	9.6	97.6	24.8	66.9	77.1	53.9	8.2
PAT	100.0	35.1	78.1	82.3	60.3	18.3	100.0	34.3	75.6	78.3	54.1	14.9
IDEM	100.0	39.6	85.6	90.2	69.6	25.8	100.0	37.1	82.6	87.2	65.2	22.1
AttChain$_{\text{Llama}}$	100.0	42.1*	92.3*	99.1*	86.2*	34.0*	100.0	40.1*	87.2*	98.6*	84.0*	33.1*
AttChain$_{\text{GPT}}$	100.0	**44.5***	**94.9***	**99.6***	**91.2***	**39.1***	100.0	**42.4***	**89.2***	**99.2***	**89.8***	**38.0***
AttChain$_{-\text{CoT}}$	100.0	37.1	85.2	94.2	67.4	23.0	100.0	33.1	78.9	95.3	64.0	21.9
AttChain$_{-\text{dynamic}}$	100.0	41.8*	94.5*	99.2*	84.3*	32.5*	100.0	40.1*	88.3*	98.2*	84.5*	33.3*
PROP	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R
TS	100.0	37.6	83.0	89.7	67.3	22.8	100.0	34.6	79.9	91.2	65.0	20.1
PRADA	95.2	23.4	66.6	75.8	53.4	8.6	94.0	21.9	62.8	72.9	52.1	6.5
PAT	98.6	33.6	75.9	80.2	58.7	17.3	97.0	32.1	72.9	76.5	51.2	13.7
IDEM	100.0	37.3	83.0	87.9	67.5	24.0	100.0	34.6	78.5	86.2	66.0	22.1
AttChain$_{\text{Llama}}$	100.0	41.0*	89.6*	98.7*	84.5*	33.2*	100.0	39.7*	86.3*	96.3*	83.9*	32.4*
AttChain$_{\text{GPT}}$	100.0	**43.0***	**92.8***	**99.2***	**89.3***	**37.5***	100.0	**41.2***	**88.2***	**97.2***	**88.3***	**36.2***
AttChain$_{-\text{CoT}}$	100.0	35.8	80.1	93.8	65.2	22.4	100.0	32.3	76.2	93.5	62.5	20.0
AttChain$_{-\text{dynamic}}$	100.0	40.1*	90.5*	98.4*	82.3*	31.2*	100.0	39.7*	86.3*	96.3*	83.9*	32.4*
RankLLM	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R	ASR	Boost	T10R
TS	100.0	34.3	79.4	89.8	63.9	19.7	98.9	30.6	71.0	86.8	57.6	19.0
PRADA	92.1	21.1	60.9	68.9	50.2	6.7	88.7	19.8	59.9	72.3	47.8	5.8
PAT	95.6	30.2	72.1	75.6	54.3	14.9	94.6	28.9	67.6	74.3	48.5	10.9
IDEM	98.9	34.8	79.2	84.8	63.2	21.8	97.3	34.2	78.5	82.1	60.9	19.3
AttChain$_{\text{Llama}}$	100.0	38.2*	82.9*	92.9*	81.0*	27.9*	100.0	36.5*	83.6*	93.5*	81.0*	30.9*
AttChain$_{\text{GPT}}$	100.0	**40.5***	**87.5***	**95.6***	**84.3***	**31.6***	100.0	**38.5***	**85.6***	**95.2***	**83.6***	**32.8***
AttChain$_{-\text{CoT}}$	100.0	32.1	79.2	81.6	72.3	19.2	100.0	32.5	76.3	86.5	63.8	22.4
AttChain$_{-\text{dynamic}}$	100.0	38.3*	85.8*	92.8*	81.2*	28.7*	100.0	36.3*	83.1*	92.3*	79.8*	30.2*

Table 2: Attack performance of AttChain and baselines on MS MARCO and TREC2019; $*$ indicates significant improvements over the best baseline ($p \leq 0.05$).

indicates that, language models can interact with NRMs to generate effective adversarial examples.

When we look at AttChain, we find that: (i) AttChain$_{\text{GPT}}$ outperforms all baselines, highlighting that LLMs are inherently good attackers of NRMs. They can use their powerful reasoning capabilities can fully capture the preferences of NRMs in interactions, followed by generative capabilities to obtain effective adversarial examples. (ii) The advantage of AttChain$_{\text{GPT}}$ over AttChain$_{\text{Llama}}$ indicates that larger-scale LLMs not only have stronger reasoning and generative capabilities, but can also better capture the knowledge of NRMs. (iii) The superiority of AttChain$_{\text{GPT}}$ over AttChain$_{-\text{CoT}}$ suggests that, for anchor document selection, the highest ranking is not necessarily the most appropriate. LLMs can find the most efficient anchor document at the moment and generate the corresponding perturbation through the reasoning chain. (iv) The advantage of AttChain$_{\text{GPT}}$ over AttChain$_{-\text{dynamic}}$ suggests that progressive perturbation from coarse-grained to fine-grained can help target documents improve their ranking efficiently.

5.2 Naturalness Evaluation

Average number of queries, grammar checking, PPL, and human evaluation. Table 3 shows the results of the average number of queries to NRM, grammar, PPL, and human evaluation. We take the Mixture target documents of MS MARCO as examples, with similar findings on other target documents and datasets. We observe that: (i) The Qrs of AttChain$_{\text{GPT}}$ is significantly lower than other methods as AttChain$_{\text{GPT}}$ does not need to train surrogate models by accessing the target NRM multiple times, but instead identifies efficient reasoning chains through LLMs. This facilitates its avoidance of suspicion. (ii) The synonym substitution attacks (PRADA) are the trigger injection at-

Method	Qrs	Grammar	PPL	Impercept.	*kappa*
Original	-	59	43.8	0.89	0.48
TS	-	67	63.2	0.12	0.56
PRADA	218.4	108	102.8	0.48	0.51
PAT	74.8	83	70.8	0.53	0.65
IDEM	72.4	71	48.5	0.75	0.42
$\text{AttChain}_{\text{GPT}}$	25.0	61	38.3	0.85	0.57

Table 3: Average number of queries, online grammar checker, perplexity, and human evaluation results for attacking RankLLM on MS MARCO.

Threshold	0.08	0.06	0.04	0.02
TS	39.2	51.9	64.0	90.1
PRADA	12.3	23.5	39.7	61.3
PAT	9.1	14.9	25.4	48.4
IDEM	16.8	28.7	46.2	71.8
$\text{AttChain}_{\text{GPT}}$	**6.3**	**11.2**	**19.4**	**38.2**

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

tacks (PAT) and easily detected. This is because they inevitably introduce grammatical errors or awkward expressions. (iii) $\text{AttChain}_{\text{GPT}}$ outperforms the baselines over all the naturalness metrics, demonstrating the power of LLMs in generating imperceptible adversarial examples.

Spamicity detection. Table 4 shows the automatic spamicity detection results on Mixture documents with similar findings on other target documents. We observe that: (i) Due to the direct introduction of a large number of query terms, TS can easily be detected as spamming, while other methods are relatively free from this disadvantage. (ii) PAT has the lowest detection rate among the baselines because it actively avoids generating query terms when generating triggers. (iii) $\text{AttChain}_{\text{GPT}}$ outperforms the baselines on spamming detection, demonstrating instructing LLMs can generate natural and hard-to-detect adversarial examples.

5.3 Mitigation Analysis

We try possible ways of distinguishing the adversarial examples generated by LLMs from the original target document using perplexity (shown in Table 2) and semantic similarity between origin documents (shown in Table 3).

Mitigating by perplexity. We take the adversarial examples and corresponding target documents on the MS MARCO dataset as examples. Figure 2 shows log perplexity distributions, evaluated by GPT-2 (Radford et al. 2019) on the two types of documents. There is a significant distribution overlap between adversarial examples and target documents. This implies that filtering adversarial examples through perplexity might result in too many original documents being considered harmful or overlook many adversarial samples that should be detected. Therefore, this method seems

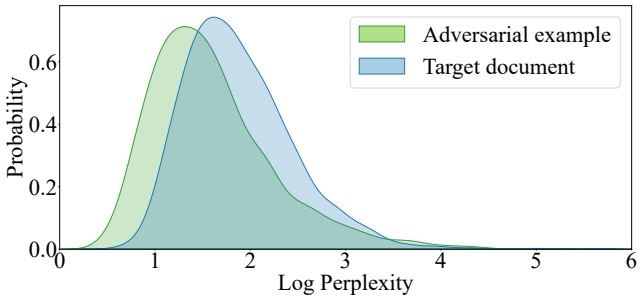

Figure 2: Distributions of log perplexity (PPL) of adversarial examples generated by $\text{AttChain}_{\text{GPT}}$ and target documents on MS MARCO.

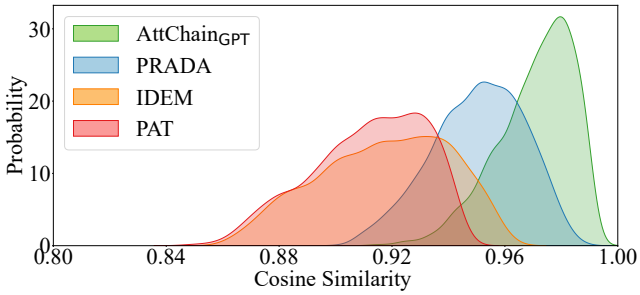

Figure 3: Distribution of cosine similarity of semantic embedding between adversarial examples generated by different attack methods and target documents on MS MARCO.

to struggle to distinguish adversarial samples generated by AttChain from original documents.

Mitigating by semantic similarity. we also compare the semantic similarity between the adversarial example and its corresponding original document. We use the OpenAI semantic embedding (OpenAI 2024b) to calculate the normalized cosine similarity (Rahutomo et al. 2012) between the two types of documents. Figure 3 shows distributions of cosine similarity across different attack methods. We can find that PRADA, due to its use of synonym replacement strategy, generates adversarial samples that have higher semantic similarity to original documents than other baselines. AttChain generally maintains a high similarity with original documents, because of the effective instruction-following and generation abilities of LLMs. This would make it difficult for search engines to distinguish them based on the size of differences during document updates.

On TREC2019 we arrived at similar observations. Therefore, it is worthwhile to investigate how to recognize adversarial examples through other techniques, such as LLM-generated text detection. We will further explore the detection of LLM-generated adversarial examples in future work.

5.4 Case Study

To further understand the proposed AttChain method, we randomly sample a query (ID=524332) and a corresponding hard target document (ID=D1875904) from MS MARCO. The adversarial examples generated by different attack

Method	**Query:** treating tension headaches medication **Document title:** Headache Locations Chart	Rank
Original	Headache Locations – What does the location of a headache mean? by Melinda Wilson — Apr 18, 2015 — Headache Locations —Headache is an illness caused by overactivity of, or problems with, structures in the head that are sensitive of pain. Did you know? That there is an organization which advocates the welfare of headache sufferers? National Headache Foundation has categorized headache as a neurobiological disease. With their 45 years of further research and...	98
PRADA	Headache Locations – What does the **site** of a headache mean? by Melinda Wilson — Apr 18, 2015 — Headache Locations —Headache is an ailment caused by overactivity of, or **issues** with, structures in the head that are sensitive of pain. Did you know? That there is an **institution** which advocates the welfare of headache sufferers? National Headache Foundation has **classified** headache as a neurobiological **illness**. With their 45 **year** of further research and...	18
PAT	**where do pains in head live, headache locations mean what nervous** Headache Locations – What does the location of a headache mean? by Melinda Wilson — Apr 18, 2015 — Headache Locations — Headache is an illness caused by overactivity of, or problems with, structures in the head that are sensitive of pain. Did you know? That there is an organization which advocates the welfare of headache sufferers? National Headache Foundation has categorized headache as a ...	26
IDEM	Headache Locations – What does the location of a headache mean? by Melinda Wilson — Apr 18, 2015 — Headache Locations — Headache is an illness caused by overactivity of, or problems with, structures in the head that are sensitive to pain. **This condition can be triggered by tension factors, including headaches with medications.** Did you know? That there is an organization which advocates the welfare of headache sufferers? National Headache Foundation has categorized...	7
$\text{AttChain}_{\text{GPT}}$	Headache Locations – What does the location of a headache mean? by Melinda Wilson — Apr 18, 2015 — Headache Locations —Headache is an illness caused by **strain or pressure and** overactivity of structures in the head that are sensitive of pain... the welfare of headache sufferers? National Headache Foundation has categorized headache as a neurobiological disease **that requires specific treatments**. With their 45 years of further research **drug-based strategies** and...	2

Table 5: Adversarial examples generated by $\text{AttChain}_{\text{GPT}}$ and other baselines for attacking RankLLM on MS MARCO based on a sampled query with different target documents. Due to space limitations, we only show some key sentences in the document.

methods are shown in Table 5. From this example, we find that PAT generates perturbations that are difficult to read, while PRADA may introduce grammatical errors. IDEM can generate relatively natural perturbations, but there is a risk of introducing query terms that can easily be detected as spam. Besides, the adversarial example generated by $\text{AttChain}_{\text{GPT}}$ is more natural-looking that generated by baselines.

6 Conclusion

We have proposed an attack method against neural ranking models (NRMs) based on large language models (LLMs). We employ chain-of-thought (CoT) prompting to perform multiple NRM-LLM interaction rounds in a reasoning chain to generate effective and imperceptible adversarial examples. Experiments on two web search benchmark datasets show that the proposed method achieves threatening attacks with limited access to NRMs. Adopting closed-source LLMs, i.e., GPT3.5 improves the attack effectiveness but incurs a relatively large cost. For future work, we plan to use, and analyze, open-source LLMs to achieve attacks.

Through our work, we found that LLMs can easily identify NRMs' vulnerabilities in relevance assessment, thereby deceiving NRMs. This raises concerns about the use of NRMs in the age of AI-generated content being exploited by search engine optimization (SEO). We will investigate corresponding defense mechanisms to help develop trustworthy neural IR systems in the future.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62472408 and 62372431, the Strategic Priority Research Program of the CAS under Grants No. XDB0680102 and XDB0680301, the National Key Research and Development Program of China under Grants No. 2023YFA1011602 and 2021QY1701, 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 work was also (partially) funded 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

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *The Sixth International Conference on Learning Representations*.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking Black Box Large Language Models in Twenty Queries. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Chen, J.; Lin, H.; Han, X.; and Sun, L. 2024. Benchmarking Large Language Models in Retrieval-Augmented Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17754–17762.

Chen, X.; He, B.; Ye, Z.; Sun, L.; and Sun, Y. 2023. Towards Imperceptible Document Manipulations against Neural Ranking Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, 6648–6664.

Cohn, C.; Hutchins, N.; Le, T.; and Biswas, G. 2024. A Chain-of-Thought Prompting Approach With LLMs for Evaluating Students' Formative Assessment Responses in Science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23182–23190.

Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; and Voorhees, E. 2019. Overview of the TREC 2019 Deep Learning Track. In *TExt Retrieval Conference 2019*.

Dai, Z.; and Callan, J. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*.

Gadyatskaya, O.; and Papuc, D. 2023. ChatGPT Knows Your Attacks: Synthesizing Attack Trees Using LLMs. In *International Conference on Data Science and Artificial Intelligence*, 245–260. Springer.

Giadikiaroglou, P.; Lymperaiou, M.; Filandrianos, G.; and Stamou, G. 2024. Puzzle Solving using Reasoning of Large Language Models: A Survey. *arXiv preprint arXiv:2402.11291*.

Guo, J.; Fan, Y.; Ai, Q.; and Croft, W. B. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 55–64.

Gyongyi, Z.; and Garcia-Molina, H. 2005. Web Spam Taxonomy. In *AIRWeb*.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *The Ninth International Conference on Learning Representations*.

Ji, B.; Liu, H.; Du, M.; and Ng, S.-K. 2024. Chain-of-Thought Improves Text Generation with Citations in Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18345–18353.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Denison, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.; Kernion, J.; et al. 2023. Measuring Faithfulness in Chain-of-Thought Reasoning. *arXiv preprint arXiv:2307.13702*.

Liu, J.; Kang, Y.; Tang, D.; Song, K.; Sun, C.; Wang, X.; Lu, W.; and Liu, X. 2022. Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2025–2039.

Liu, Y.-A.; Zhang, R.; Guo, J.; de Rijke, M.; Chen, W.; Fan, Y.; and Cheng, X. 2023. Topic-Oriented Adversarial Attacks against Black-Box Neural Ranking Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1700–1709.

Liu, Y.-A.; Zhang, R.; Guo, J.; de Rijke, M.; Fan, Y.; and Cheng, X. 2024a. 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*, 1391–1400.

Liu, Y.-A.; Zhang, R.; Zhang, M.; Chen, W.; de Rijke, M.; Guo, J.; and Cheng, X. 2024b. Perturbation-Invariant Adversarial Training for Neural Ranking Models: Improving the Effectiveness-Robustness Trade-Off. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8832–8840.

Ma, X.; Guo, J.; Zhang, R.; Fan, Y.; Ji, X.; and Cheng, X. 2021. Prop: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 283–291.

Madaan, A.; and Yazdanbakhsh, A. 2022. Text and Patterns: For Effective Chain of Thought, It Takes Two to Tango. *arXiv preprint arXiv:2209.07686*.

Meta. 2024. Meta Llama 3: The Most Capable Openly Available LLM to Date. https://ollama.com/library/llama3. Accessed: 2024-08-15.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *CoCo@NIPS*.

Nogueira, R.; and Cho, K. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2024-08-15.

OpenAI. 2024a. OpenAI API. https://openai.com/api/. Accessed: 2024-08-15.

OpenAI. 2024b. Text-embedding-3. https://platform.openai.com/docs/api-reference/embeddings. Accessed: 2024-08-15.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-box Attacks Against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 506–519.

Paranjape, B.; Lundberg, S.; Singh, S.; Hajishirzi, H.; Zettlemoyer, L.; and Ribeiro, M. T. 2023. Art: Automatic Multi-step Reasoning and Tool-use for Large Language Models. *arXiv preprint arXiv:2303.09014*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI blog*, 1(8): 9.

Rahutomo, F.; Kitasuka, T.; Aritsugi, M.; et al. 2012. Semantic Cosine Similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, 1. University of Seoul South Korea.

Raina, V.; Liusie, A.; and Gales, M. 2024. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. *arXiv preprint arXiv:2402.14016*.

Raval, N.; and Verma, M. 2020. One Word at a Time: Adversarial Attacks on Retrieval Models. *arXiv preprint arXiv:2008.02197*.

Song, C.; Rush, A. M.; and Shmatikov, V. 2020. Adversarial Semantic Collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 4198–4210.

Sun, W.; Yan, L.; Ma, X.; Wang, S.; Ren, P.; Chen, Z.; Yin, D.; and Ren, Z. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 14918–14937.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023a. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2609–2634.

Wang, Y.; Li, P.; Sun, M.; and Liu, Y. 2023b. Self-Knowledge Guided Retrieval Augmentation for Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 10303–10315.

Wang, Y.; Lyu, L.; and Anand, A. 2022. BERT Rankers are Brittle: A Study using Adversarial Document Perturbations. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, 115–120.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Wu, C.; Zhang, R.; Guo, J.; de Rijke, M.; Fan, Y.; and Cheng, X. 2023. PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models. *ACM Transactions on Information Systems*, 41(4): Article 89.

Wu, M.; Jiang, S.; and Zhang, Y. 2012. Serial Position Effects of Clicking Behavior on Result Pages Returned by Search Engines. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2411–2414.

Xu, J.; Stokes, J. W.; McDonald, G.; Bai, X.; Marshall, D.; Wang, S.; Swaminathan, A.; and Li, Z. 2024a. Autoattacker: A Large Language Model Guided System to Implement Automatic Cyber-attacks. *arXiv preprint arXiv:2403.01038*.

Xu, S.; Pang, L.; Shen, H.; Cheng, X.; and Chua, T.-s. 2023a. Search-in-the-chain: Towards the Accurate, Credible and Traceable Content Generation for Complex Knowledge-intensive Tasks. *arXiv preprint arXiv:2304.14732*.

Xu, X.; Kong, K.; Liu, N.; Cui, L.; Wang, D.; Zhang, J.; and Kankanhalli, M. 2023b. An LLM Can Fool Itself: A Prompt-based Adversarial Attack. *arXiv preprint arXiv:2310.13345*.

Xu, X.; Kong, K.; Liu, N.; Cui, L.; Wang, D.; Zhang, J.; and Kankanhalli, M. 2024b. An LLM Can Fool Itself: A Prompt-Based Adversarial Attack. In *The Twelfth International Conference on Learning Representations*.

Yan, M.; Li, C.; Bi, B.; Wang, W.; and Huang, S. 2021. A Unified Pretraining Framework for Passage Ranking and Expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4555–4563.

Yu, L.; Zhang, C.; Liang, S.; and Zhang, X. 2019. Multi-order Attentive Ranking Model for Sequential Recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5709–5716.

Yu, W.; Zhang, Z.; Liang, Z.; Jiang, M.; and Sabharwal, A. 2023. Improving Language Models via Plug-and-Play Retrieval Feedback. *arXiv preprint arXiv:2305.14002*.

Zhou, B.; and Pei, J. 2009. OSD: An Online Web Spam Detection System. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, volume 9.

Zipf, G. K. 2016. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Ravenio Books.