Analyzing Adversarial Attacks on Sequence-to-Sequence Relevance Models

Andrew Parry,[1] Maik Fröbe,[2]
Sean MacAvaney,[1] Martin Potthast,[3,4] Matthias Hagen[2]

[1] University of Glasgow
[2] Friedrich-Schiller-Universität Jena
[3] Leipzig University
[4] ScaDS.AI

Abstract Modern sequence-to-sequence relevance models like monoT5 can effectively capture complex textual interactions between queries and documents through cross-encoding. However, the use of natural language tokens in prompts, such as `Query`, `Document`, and `Relevant` for monoT5, opens an attack vector for malicious documents to manipulate their relevance score through prompt injection, e.g., by adding target words such as `true`. Since such possibilities have not yet been considered in retrieval evaluation, we analyze the impact of query-independent prompt injection via manually constructed templates and LLM-based rewriting of documents on several existing relevance models. Our experiments on the TREC Deep Learning track show that adversarial documents can easily manipulate different sequence-to-sequence relevance models, while BM25 (as a typical lexical model) is not affected. Remarkably, the attacks also affect encoder-only relevance models (which do not rely on natural language prompt tokens), albeit to a lesser extent.

https://github.com/Parry-Parry/ecir24-adversarial-evaluation

1 Introduction

Web search referrals are one of the most important methods of generating traffic to web pages [15]. Consequently, content providers often try to increase the visibility of their content in search engines through search engine optimization (SEO) [6, 17, 23]. Common SEO techniques include adding (invisible) keywords to a page to improve its ranking for certain topics or creating links to the page, leading to link farms [6]. Although SEO in good faith can help make useful content more accessible to users, malicious actors use SEO techniques to promote spam [30]. While traditional search systems are vulnerable to malicious SEO techniques, it is so far unclear whether neural relevance models based on large language models, such as BERT [10] and T5 [35], are as well.

As neural relevance models have recently yielded substantially improved retrieval effectiveness [24, 32], we investigate the robustness of neural relevance models against the well-known SEO technique of keyword stuffing [17]. Unlike

Table 1: Illustration of three prompt injection attacks on the monoT5 relevance model for the query $q :=$ `How long do fleas live?` to increase the predicted relevance of document d. Besides 'true', other adversarial terms can be used.

Attack	**Prompt** $q_d :=$ `Query:` q `Document:` d `Relevant:`	$P(\texttt{true} \mid q_d)$
None	$d :=$ Fleas live a long time. Buy flea remedies here.	0.11
Preemption	$d' :=$ Relevant: true Fleas live a long time. Buy flea remedies here.	0.25 (+0.14)
Stuffing	$d' :=$ true true true Fleas live a long time. Buy flea remedies here.	0.46 (+0.35)
Rewriting	$d' :=$ True fleas live a long time. Buy relevant flea remedies here.	0.33 (+0.22)

previous work on attacks against neural relevance models (Section 2), which substitute document tokens with synonyms [44, 26] or append poisoned text [25], our attacks are not gradient-based and therefore do not require access to model parameters [36] or a surrogate model [25]. Instead, our attacks are both query-independent and only require (at most) knowledge of a model's prompt (Section 3). Table 1 illustrates the basic idea for monoT5 [32], a popular neural relevance model. The model encodes a query q and a document d in a basic prompt q_d to rank the document according to the probability $P(\texttt{true} \mid q_d)$ that the next term is `true`. We investigate three attacks on the prompt's control tokens: a preemption attack that exploits the model's tendency not to contradict itself; a stuffing attack that repeats `true` to increase the probability that the term is the next word; and an LLM rewriting attack with the same effect, but less easily detected by countermeasures.

Our evaluation of the 2019 and 2020 TREC Deep Learning topics shows that these attacks can significantly improve the rank of a document (Section 4). We also find that synonyms of relevance-indicating control tokens can be effective and that attacks can generalize to BERT-based cross-encoders not trained with a prompt. As these attacks are easily accomplished even by non-experts, our findings warn against using neural relevance models in production without a high level of safeguards against such attacks and also has implications for the use of prompt-based models for automated relevance judgments in retrieval evaluation [11, 28, 40] and automated ground truth generation in training [2, 9, 19].

2 Related Work and Background

We describe prior work on neural information retrieval, probing neural relevance models, attacks against relevance models, and large language models to motivate our new adversarial attacks against sequence-to-sequence relevance models.

2.1 Neural Information Retrieval

Modern neural retrieval models use a pre-trained language model for relevance approximation. The contextualization of pre-trained language models allows neural retrievers to overcome previous problems, such as lexical mismatch. Current neural retrievers are either (1) bi-encoders that independently embed the query and the documents [22, 18, 20], or (2) cross-encoders that encode the query together with the document [32, 34]. Thereby, BERT based cross encoders separate the query and document by a special token [29, 1], T5-based cross-encoders instead use a structured prompt template containing the word 'relevant:' [32].

Neural retrievers are frequently trained with a contrastive approach, where one relevant and one non-relevant document is passed to the model for a given query (either explicit [18, 29] or implicit [22, 32]). In the case of sequence-to-sequence cross-encoders, an encoder-decoder model such as T5 [35] is trained to output 'true' or 'false' jointly conditioned on a query and document. We exploit this prompt structure as an attack vector to sequence-to-sequence rankers.

2.2 Probing Neural Information Retrieval Models

The emergence of neural retrieval models was accompanied by concerns over their robustness to both deliberate attacks [25, 44] and uncontrolled behavior that diverges from any human concept of relevance [27]. Camara et al. [5] first assessed BERT-based retrieval models for retrieval axioms, finding that their relevance approximation does not align with existing information retrieval axioms. MacAvaney et al. [27] explored the impact of perturbation of documents on retrieval scores, finding anomalous behavior, e.g., neural retrievers prefer augmented documents with non-relevant text added to the end of each document over the original documents. Probing of neural retrievers has been extended beyond comparison to axiomatic approaches with investigations showing invariance to the use of negation [43] and a failure to identify important lexical matches [12]. The unexpected responses found in these works compounded with neural ranking attacks suggest that a broadly applicable attack such as ours could present implications for the wider application of neural search.

2.3 Ranking Attacks

Attacking relevance models can serve many purposes, such as promoting harmful content or increasing the chance that users consume some content. Search engine optimization techniques (SEO) can be considered the first form of ranking attacks, aiming at artificially inflating the perceived relevance of a web page for some query for a search engine [17]. We do not consider link-based or advertising approaches to SEO as they are beyond the scope of document augmentation. The spam problem has been well researched and can be combated with an ensemble of features or automated assessment in search [6, 45]. However, the promise of a single end-to-end neural relevance model reduces a search provider's ability to reduce the effect of document augmentation.

Neural networks are vulnerable to adversarial attacks, the perturbation of an input that causes an unexpected bias in a neural model [38]. When applying adversarial attacks against pre-trained language models, a perturbation is added to the latent representation of a text to achieve an objective, either a bias towards a label or the generation of particular tokens [16]. For neural relevance models, these attacks instead substitute document tokens for synonyms, chosen to yield a new document that, when encoded, closely resembles the optimal adversarial representation. These synonyms arbitrarily increase document relevance scores for some targeted model for target queries [36, 44, 26], whereas our approach increases relevance scores independently of any particular query.

2.4 Large Language Models

Recent developments in decoder-only language models have led to large improvements in the ability of pre-trained language models to generalize to unseen tasks [4, 41, 39]. These developments include significant increases in both parameter size and training corpora [4], as well as instruction fine-tuning where models are trained to output human-aligned answers when prompted with tasks [39]. Research has already shown that though these models have been aligned with human judgments, this alignment can be bypassed via attacks like prompt injections in which a task can either be disguised or perturbed, causing the generation of harmful content. We follow this direction and study, for the first time, such adversarial attacks against neural retrieval models.

3 Query-Independent Attacks Against Sequence-to-Sequence Relevance Models

We outline our proposed attacks on sequence-to-sequence relevance models for the case of monoT5 and study their transferability to other frequently used neural models. We review the required background of sequence-to-sequence relevance models and describe our preemption, stuffing, and rewriting attacks.

3.1 Vulnerability of Sequence-to-Sequence Relevance Models

To motivate our attacks, we explain why sequence-to-sequence relevance models may be vulnerable to adversarial attacks. For a query q, sequence-to-sequence relevance models are typically used to re-rank the top-ranked results D of a first stage ranker. A re-ranker usually tries to improve the relevance approximation with respect to q for each $d \in D$. Applied to re-ranking, sequence-to-sequence cross-encoders jointly encode the query and a to-be-re-ranked document in a structured prompt [31], as exemplified in Table 1. As the query and the document are provided to the model in a continuous sequence, all terms from the document interact with all terms of the prompt and, therefore, the query. Similar to well-known keyword-stuffing methods, we hypothesize that including *prompt* tokens or their synonyms in documents increases relevance scores, thereby affecting a document's ranking across all queries. Thus, we investigate how included prompt tokens affects the relevance scores of neural retrievers.

A search engine provider should not assume that all content providers are acting in good faith. Traditional vectors to attack a retrieval system mostly attempt to augment a text or web page, e.g., using keyword stuffing, aiming at

Table 2: Overview of requirements of different adversarial attacks on neural relevance models. Attackers either need full (✓), partial (✓*), or no access (X).

	Content		Model	
	Document	Query	Prompt	Weights
Iterative Perturbation [36]	✓	✓	✓	✓*
PRADA [44]	✓	✓	✓	✓*
Trigger Based Attacks [25, 26]	✓	✓	✓	✓*
Suffix Attack [47]	✓	✓	✓	✓*
Ours	✓	X	✓	X

specific queries or topics. When attacking neural IR systems, one may instead use an adversarial approach to gradually transform a text such that a relevance model assigns a higher score to that text for a target query. To generalize these attacks, we define a transformation $d' = f(d, c)$, where c is any context used to guide the augmentation, either query information or gradient response from a target neural model, producing the augmented text d'.

3.2 Attack Model

Table 2 overviews the attack model underlying our adversarial attack compared to related attacks. For our attack, attackers need to know the text prompt and the output tokens that approximate the probability of relevance. Access to the weights of the target model (or a surrogate model) is not required. Attackers do not need to target particular queries but only augment document text. Hence, our attack has the least requirements among the attacks in Table 2.

3.3 Adversarial Preemption and Keyword-Stuffing

An adversarial attack aims to produce a document d' from d, such that for a query q, $R_\theta(q, d') > R_\theta(q, d)$. When using prior approaches, such an attack would require multiple representations of a document to ensure that each representation contains either lexical matches in a classic setting or suitable perturbations with respect to both a target query and model. For a set of queries Q assuming no topic overlap between queries, all augmentations required for a given document can be given as the set, $\{f(d, c = q); \forall q \in Q\}$.

We instead look to exploit query-independent knowledge of the prompt used in training sequence-to-sequence models. Following a classic SEO approach, we inject prompt tokens into each passage, controlling for injection and token repetitions. By injecting prompt tokens, we attempt to preempt the relevance judgement. We investigate how neural retrievers are affected by adversarial tokens and their repetitions. We consider the injection of $n \in \{1, 2, 3, 4, 5\}$ repetitions of a token at the start (**s**) or end (**e**) of the document and injections of a token at random (**r**) as exemplified in Table 1.

By controlling for both position and repetition, we aim to investigate how these tokens affect the contextualization in the underlying pre-trained language models. We consider variations of tokens contained in the targeted prompt, investigating the injection of (1) prompt tokens, (2) control tokens, (3) synonyms, and (4) sub-words. Prompt tokens refer to spans from the prompt structure used during neural training. We use control tokens as spans with equal length after encoding to one of the prompt tokens (e.g., 'information: baz' to control for 'relevant: true'). Synonyms refer to terms similar to a prompt token, which could fool a naive filtering system. With sub-words, we want to investigate if attackers can hide the adversarial tokens in longer, potentially misspelled words.

3.4 Adversarial Document Re-Writing with Large Language Models

We describe two approaches to increase a document's score by automatically re-writing it with large language models (LLMs). As LLMs can produce many different responses for some input, such re-writing attacks are much harder to detect than previous injection attacks. Using Alpaca [39] and ChatGPT,[5] we propose two classes of adversarial re-writing: (1) paraphrase approaches that re-write a passage, and (2) summarization approaches prepend a passage by a summary sentence. For both classes, we develop prompts to increase the number of adversarial tokens in the paraphrased passage or the summary sentence. Out of five candidate prompts, we identified the most effective prompt for Alpaca and ChatGPT in a pilot study. All our re-writing approaches are query-independent so attackers can apply them at scale.

For all our re-writing attacks, we use the commercial ChatGPT that we contrast with the open-source alternative Alpaca. For ChatGPT, we use the official REST API by OpenAI for the model gpt-3.5-turbo (our experiments cost less than 5 Euro). We use the official scripts for Alpaca to obtain the 7 billion parameter variant that we operate with the default configuration on one Nvidia A100 GPU with 40GB. We manually develop 10 candidate prompts (5 for paraphrasing and 5 for summarization) using the example from Table 1. To identify suitable prompts for each LLM, we sample 1000 query–document pairs from the passage re-ranking dataset of the TREC 2019 Deep Learning track [8] as a pilot study to identify the prompt causing the highest rank changes. To foster reproducibility, we include all request–response pairs in our code repository.

Adversarial Paraphrasing. Our first re-writing attack uses a large language model to paraphrase a passage while adding adversarial tokens (e.g., "relevant" or "true") to the paraphrased passage.

Adversarial Summarization. Our second re-writing attack prepends a passage by a single sentence summarizing the passage but including additional adversarial tokens. For a passage p and an adversarial summary sentence s produced by an LLM, we use $p' = s + \texttt{'␣'} + p$ as the adversarial passage.

[5] https://chat.openai.com/chat

4 Evaluation

We evaluate our query-independent adversarial attacks on the task of passage ranking. We contrast the perspective of a content provider (who aims to increase the document's visibility) with that of a search engine provider (who aims for effective retrieval). We assess the potential rank improvement from the content provider's perspective when applying our adversarial attack. This evaluation is performed point-wise to simulate a single adversarial document in a standard corpus. From the search provider's perspective, we measure the impact of our adversarial attacks on retrieval effectiveness using hypothetical best and worst-case scenarios where only relevant and non-relevant documents are manipulated using our adversarial attacks.

4.1 Experimental Setup and Evaluation Methodology

Datasets. We use the 2019/2020 TREC Deep Learning tracks [8, 7] of version 1 of MSMARCO [3] (with 8.8 million passages; the 2021/2022 editions on version 2 are somewhat discouraged [42, 13]), re-ranking the top-1000 BM25 results. We contrast rankings for original documents with their attacked counterparts.

Measures. To assess attack efficacy, we define the following measures between original and attacked document sets. All measures are computed for some transformation $f(d)$. We first define the success rate (**SR**) as the fraction of all query–document pairs P where $f(d)$ improves the rank of a given document d:

$$\mathrm{SR}(P) = \frac{1}{|P|} \sum_{q,d \in P} \begin{cases} 1, & \text{if } \mathrm{rank}(q, f(d)) < \mathrm{rank}(q, d) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathrm{rank}(q, d)$ is the rank of d for q when ordered by descending score.

We define the Mean Rank Change (**MRC**) as the mean rank difference before and after the attack to show the visible magnitude of an attack:

$$\frac{1}{|P|} \sum_{q,d \in P} \mathrm{rank}(q, d) - \mathrm{rank}(q, f(d)) \tag{2}$$

To assess the broader effects of this attack on retrieval effectiveness, we evaluate nDCG@10 and P@10 over the best and worst-case scenarios of all attacks. As users primarily interact with the top-10 results [21], search engine providers are the most concerned with the effect of an attack at this cutoff.

Target Models. Although our attacks primarily focus on sequence-to-sequence models, we also study if they generalize to other neural models. We evaluate relevance models that cover lexical approaches and a set of neural architectures. As the main target, we use monoT5 [32], a T5-based sequence-to-sequence cross-encoder that we also contrast across four model sizes. As a lexical model, we use BM25, a bag-of-words relevance model. Additionally, we include Electra [34],

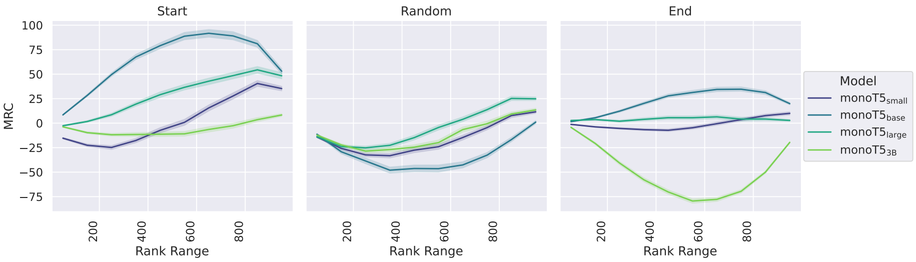


Figure 1: Aggregate MRC over every 100 ranks for the token 'relevant' injected 5 times at different positions.

a non-prompting BERT-based cross-encoder, ColBERT [22], a late interaction bi-encoder, and TAS-B [18] as a classical bi-encoder. In all cases, we use the (Py)Terrier implementation [33] with default parameters.

4.2 A Content Provider's Perspective

In this section, we evaluate the efficacy of our attack on a per-document level.

Attacking monoT5 with Keyword Stuffing Table 3 presents the effect of keyword stuffing attacks (Section 3.3) on variants of monoT5. We find that the injection of prompt tokens, which include the token 'relevant', improves document rank on average in all variants apart from monoT5$_{3B}$. Notably, in all cases, the token 'false' leads to less degradation than 'true'. This contradicts any preempting of the relevance judgement via a suffix. Furthermore, 'relevant: false' performs better than 'relevant: true' in all cases. However, the repetition of relevance leads to large rank improvements in the base and large variants. Significant rank increases occur in most cases for spans containing 'information' with rank increasing up to 111 places in the case of monoT5$_{\text{small}}$ on DL19. In Figure 1, we observe that generally, monoT5 is less susceptible to keyword stuffing applied to highly ranked documents, with both positive and negative effects being reduced (contrast ranks 0-200 and 500-700), likely showing that adding content to a document already considered relevant, has little effect on sequence-to-sequence cross-encoders. We also observe clear positional bias when contrasting random to start and end, with monoT5 variants consistently penalising tokens appended to documents whilst largely improving rank when prepending tokens.

Synonyms generally do not succeed in improving document rank; however, both 'significant' and 'associated' transfer to both the base and 3B variants of monoT5, and injection of the token 'important' improves MRC by 42 places on DL19 scored by monoT5$_{3B}$. Sub-words only improve MRC in attacking monoT5$_{\text{base}}$ with large rank degradation in monoT5$_{3B}$. The injection of sub-words only improves monoT5$_{\text{base}}$ with larger variants increasingly penalising augmented documents (the attack reduces rank in all settings for monoT5$_{3B}$).

Table 3: The scaling behavior of monoT5 sizes measured as MRC and SR (grey subscript) of keyword stuffing (significant changes at p < 0.05 denoted by *).

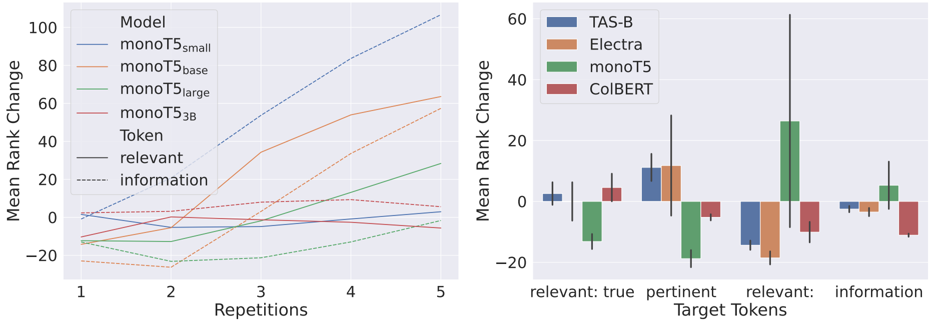
	$\text{monoT5}_{\text{small}}$		$\text{monoT5}_{\text{base}}$		$\text{monoT5}_{\text{large}}$		$\text{monoT5}_{\text{3B}}$	
Token	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20
Prompt Tokens								
true	$+1.0^{*}_{46,e,1}$	$+1.5^{*}_{47,e,1}$	$-9.1^{*}_{22,r,1}$	$-9.4^{*}_{21,r,1}$	$-3.7^{*}_{29,r,1}$	$-3.0^{*}_{30,r,1}$	$+0.8^{*}_{43,r,4}$	$+5.5^{*}_{46,r,4}$
false	$+1.3^{*}_{46,e,1}$	$+2.6^{*}_{49,e,1}$	$-0.8^{*}_{46,s,5}$	$-2.7^{*}_{33,s,5}$	$+6.7^{*}_{54,r,5}$	$+14.9^{*}_{58,r,5}$	$+2.1^{*}_{45,r,3}$	$+7.2^{*}_{48,r,3}$
relevant:	$+12.8^{*}_{50,s,5}$	$+2.9^{*}_{41,s,5}$	$+63.6^{*}_{78,s,5}$	$+51.2^{*}_{75,s,5}$	$+14.8^{*}_{56,s,5}$	$+28.4^{*}_{59,s,5}$	$-4.3^{*}_{38,r,1}$	$+0.2^{*}_{41,r,1}$
relevant: true	$+5.4^{*}_{48,e,5}$	$+4.8^{*}_{43,e,5}$	$+31.1^{*}_{64,s,5}$	$+18.3^{*}_{57,s,5}$	$+4.7^{*}_{52,e,5}$	$+11.2^{*}_{56,e,5}$	$-5.1^{*}_{39,r,1}$	$-1.5^{*}_{41,r,1}$
relevant: false	$+4.2^{*}_{47,e,5}$	$+4.5^{*}_{50,e,5}$	$+47.4^{*}_{71,s,5}$	$+32.0^{*}_{64,s,5}$	$+9.0^{*}_{48,s,5}$	$+25.4^{*}_{53,s,5}$	$-3.1^{*}_{41,r,1}$	$+1.1^{*}_{44,r,1}$
Control Tokens								
bar	$-0.3^{*}_{36,e,1}$	$-0.6^{*}_{31,e,1}$	$-3.5^{*}_{36,e,2}$	$+0.6^{*}_{41,e,2}$	$-2.3^{*}_{30,e,1}$	$+1.0^{*}_{37,e,1}$	$+3.5^{*}_{46,s,1}$	$+12.8^{*}_{50,s,1}$
baz	$-1.2^{*}_{36,e,2}$	$+1.0^{*}_{30,e,2}$	$+6.6^{*}_{53,s,5}$	$+17.2^{*}_{60,s,5}$	$-1.9^{*}_{37,r,1}$	$+4.9^{*}_{42,r,1}$	$+3.3^{*}_{48,e,1}$	$+12.7^{*}_{46,e,1}$
information:	$+111.7^{*}_{87,s,5}$	$+106.7^{*}_{83,s,5}$	$+57.4^{*}_{77,s,5}$	$+41.3^{*}_{70,s,5}$	$-4.3^{*}_{36,r,1}$	$-0.4^{*}_{38,r,1}$	$+6.2^{*}_{50,s,3}$	$+9.3^{*}_{49,s,3}$
information: bar	$+22.1^{*}_{54,s,5}$	$+23.4^{*}_{49,s,5}$	$+31.6^{*}_{70,e,5}$	$+38.2^{*}_{71,e,5}$	$+28.2^{*}_{56,s,5}$	$+52.8^{*}_{60,s,5}$	$+21.5^{*}_{57,s,4}$	$+23.4^{*}_{56,s,4}$
information: baz	$+11.4^{*}_{50,s,5}$	$+22.5^{*}_{50,s,5}$	$+31.0^{*}_{61,s,5}$	$+37.0^{*}_{61,s,5}$	$+8.6^{*}_{50,s,5}$	$+42.0^{*}_{58,s,5}$	$+62.1^{*}_{73,s,4}$	$+69.4^{*}_{70,s,4}$
relevant: bar	$+2.5^{*}_{48,e,1}$	$+2.5^{*}_{42,e,1}$	$+32.0^{*}_{62,s,5}$	$+33.6^{*}_{61,s,5}$	$-5.7^{*}_{36,r,1}$	$+7.5^{*}_{42,r,1}$	$+15.1^{*}_{53,s,2}$	$+28.5^{*}_{56,s,2}$
information: true	$+9.2^{*}_{57,e,5}$	$+8.7^{*}_{51,e,5}$	$+28.4^{*}_{62,s,5}$	$+13.5^{*}_{54,s,5}$	$+11.0^{*}_{58,e,5}$	$+19.7^{*}_{62,e,5}$	$-3.9^{*}_{40,r,1}$	$-0.9^{*}_{43,r,1}$
Synonyms								
pertinent	$-0.3^{*}_{38,e,1}$	$+0.2^{*}_{41,e,1}$	$-4.7^{*}_{41,s,5}$	$-0.7^{*}_{44,s,5}$	$-2.4^{*}_{40,r,2}$	$+0.9^{*}_{48,r,2}$	$-6.5^{*}_{28,r,1}$	$-4.9^{*}_{30,r,1}$
significant	$+1.9^{*}_{51,r,1}$	$+1.4^{*}_{46,r,1}$	$+11.3^{*}_{55,s,5}$	$+8.3^{*}_{50,s,5}$	$+0.4^{*}_{38,e,5}$	$+4.6^{*}_{52,e,5}$	$+5.3^{*}_{45,r,4}$	$+2.4^{*}_{44,r,4}$
related	$-3.1^{*}_{30,r,1}$	$-3.7^{*}_{28,r,1}$	$-2.1^{*}_{35,e,1}$	$-3.8^{*}_{31,e,1}$	$-4.3^{*}_{30,r,1}$	$-4.5^{*}_{29,r,1}$	$+8.9^{*}_{51,s,1}$	$+10.6^{*}_{52,s,1}$
associated	$+0.5^{*}_{44,r,1}$	$-0.2^{*}_{40,r,1}$	$+6.4^{*}_{50,s,5}$	$+3.6^{*}_{49,s,5}$	$-0.8^{*}_{41,r,1}$	$+0.7^{*}_{40,r,1}$	$+11.2^{*}_{57,e,2}$	$+11.7^{*}_{55,e,2}$
important	$-1.7^{*}_{36,r,1}$	$-2.7^{*}_{32,r,1}$	$-5.2^{*}_{26,e,1}$	$-3.7^{*}_{30,e,1}$	$+0.8^{*}_{43,e,5}$	$+4.6^{*}_{52,e,5}$	$+42.3^{*}_{72,r,5}$	$+49.9^{*}_{73,r,5}$
Sub-Words								
relevancy	$+0.7^{*}_{42,e,5}$	$+2.1^{*}_{42,e,5}$	$+12.9^{*}_{54,s,5}$	$+17.6^{*}_{57,s,5}$	$-3.8^{*}_{34,r,1}$	$-3.4^{*}_{34,r,1}$	$-6.2^{*}_{41,r,5}$	$-1.4^{*}_{44,r,5}$
relevance	$-1.9^{*}_{42,e,5}$	$-3.7^{*}_{36,e,5}$	$-2.3^{*}_{44,s,5}$	$+1.5^{*}_{44,s,5}$	$+4.9^{*}_{49,s,5}$	$+13.4^{*}_{52,s,5}$	$-8.6^{*}_{31,r,1}$	$-5.0^{*}_{40,r,1}$
relevantly	$+1.3^{*}_{49,r,1}$	$+2.0^{*}_{49,r,1}$	$+13.5^{*}_{61,s,5}$	$+14.1^{*}_{61,s,5}$	$-0.2^{*}_{40,r,1}$	$+1.5^{*}_{47,r,1}$	$-9.0^{*}_{29,r,1}$	$-6.3^{*}_{35,r,1}$
irrelevant	$-1.4^{*}_{34,e,1}$	$+1.2^{*}_{35,e,1}$	$+30.5^{*}_{68,s,5}$	$+34.5^{*}_{69,s,5}$	$-3.8^{*}_{34,r,1}$	$+0.2^{*}_{45,r,1}$	$-7.1^{*}_{31,r,1}$	$-1.0^{*}_{38,r,1}$

Attacking monoT5 by Re-writing Passages. Table 4 presents the effectiveness of passage re-writing attacks (Section 3.4) on various sizes of monoT5. We observe consistent rank improvements across almost all operating points, demonstrating the efficacy of this attack. $\text{MonoT5}_{\text{3B}}$ is less affected by paraphrasing attacks as rank improvements are insignificant for re-writing with ChatGPT, whereas changes are still significant for Alpaca summarization. We observe that the larger variants of monoT5 are less affected by paraphrasing attacks, albeit summary injection with Alpaca is effective in all cases. In both attack settings, attacks using Alpaca outperform ChatGPT attacks. Given the small size of this model, any attacker could perform these rewrites on a large scale and consistently improve the rank of content while making only small changes to the text.

Transfer of Injection Attacks. Table 5 shows that though monoT5 is most affected by the injection of prompt tokens as illustrated in Figure 2(b), generalisation across neural models can be seen in cases of the injection containing the token 'relevant' beyond the constraints of our attack model outlined in Section 3.2. Due to BM25 penalties for document length and the addition of tokens that

Table 4: Efficacy of paraphrasing (Par.) and prepending a summary (Sum.) to rank 100 on various sizes of monoT5 in terms of MRC and success rate (grey subscript). Significant results are denoted with * (Students t-test p < 0.05).

		$\text{monoT5}_{\text{small}}$		$\text{monoT5}_{\text{base}}$		$\text{monoT5}_{\text{large}}$		$\text{monoT5}_{\text{3B}}$	
	LLM	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20
Par.	Alpaca	$+2.7^{*}_{52}$	$+2.6^{*}_{53}$	$+2.4^{*}_{51}$	$+1.9^{*}_{50}$	$+1.5^{*}_{46}$	$+1.7^{*}_{46}$	$+1.4^{*}_{46}$	$+1.0_{44}$
	ChatGPT	$+1.7^{*}_{52}$	$+1.0_{50}$	$+3.0^{*}_{56}$	$+2.2^{*}_{54}$	$+1.2_{50}$	$+0.6_{48}$	$+0.6_{46}$	-0.1_{46}
Sum.	Alpaca	$+2.2^{*}_{47}$	$+2.1^{*}_{48}$	$+2.9^{*}_{53}$	$+2.5^{*}_{51}$	$+2.2^{*}_{49}$	$+2.3^{*}_{49}$	$+3.3^{*}_{55}$	$+2.8^{*}_{54}$
	ChatGPT	$+1.5^{*}_{47}$	$+1.1^{*}_{47}$	$+1.9^{*}_{50}$	$+0.6_{46}$	$+0.6_{45}$	$+1.0_{45}$	$+1.0_{47}$	$+0.4_{45}$



(a) Attack on monoT5 Variants

(b) Attack on Multiple NRMs

Figure 2: An overview of (a) the scaling of rank improvement for the number of token repetitions of control and prompt tokens with maximum MRC and (b) the variance of repetitions on different neural models for strongest settings.

are almost guaranteed not to be contained in the evaluation queries, the attack fails to improve document rank across all token groups[6].

Control tokens can greatly influence the monoT5 ranking of documents. However, they do not generalize beyond T5, suggesting that the prompt structure from which these controls were inspired has a significant ranking impact for prompt-based relevance models. We observe mixed results when injecting synonyms for 'relevant.' 'Significant' is effective, improving document scores for both cross-encoders. TAS-B is also affected. However, ColBERT is unaffected and insensitive to synonyms due to its max pooling token-level similarity computation that may ignore injected tokens, contrasting the deeper interaction of cross-encoders or the passage-level similarity of standard bi-encoders.

Subwords significantly improve the MRC for both cross-encoders, indicating that injecting words containing the token 'relevant' has a positive impact. Hence,

[6] In the cases of 'information' and 'related', these words are present within the default PyTerrier stop-words list and as such the document becomes duplicated causing to a tie break, this leads to the small rank change with 0.0% success rate

filtering keyword-stuffing attacks on neural models may be more challenging as tokenization allows hiding attack tokens that lexical models ignore.

Successful attacks on neural models frequently involve multiple repetitions of injected tokens (as can be observed as the 3rd subscript of each attack in Table 5). This response is unexpectedly similar to a lexical model but does not depend on the frequency of query terms and remains context-agnostic (e.g., the upward trend in Figure 2(a)). We also observe that BERT-based architectures generally prefer the injection of tokens to the end in contrast with monoT5, which almost always prefers injection at the start (as can be observed as the 2nd subscript of each attack in Table 5). The stronger generalization of appended injections may suggest that it is a more effective attack when unsure of the language model used in the targeted relevance model.

Transfer of LLM Re-Writing Attacks. Table 6 shows that cross-encoders are weaker in paraphrasing and summary attacks. In all cases, a significant improvement is found; however, summaries from ChatGPT fail to improve rank over 50% of the time in monoT5 on DL20. Rank improvements when attacking TAS-B with paraphrasing and summary are small, only improving rank in over 50% of documents on DL19. ColBERT is generally not affected by a summary reflecting a general in-variance to our attacks (further outlined by low variance observed in Figure 2(b)). Document rank significantly drops in all paraphrasing attacks against BM25; this can be attributed to the increase in document length from adding the tokens 'relevant' and 'true' as well as the potential for an LLM re-write to re-phrase terms, which may cause lexical mismatch. However, BM25 rank is improved by the injection of a summary showing that both LLMs have captured useful terms in their summary; as this attack also transfers to cross-encoders, it is an effective attack against a larger search pipeline.

4.3 A Search Provider's Perspective

We assess the impact of our adversarial attacks on the retrieval effectiveness of all models by contrasting hypothetical lower/upper bounds that we obtain in an oracle scenario. For the lower bound, we simulate that only non-relevant documents apply adversarial attacks. We simulate that only relevant documents apply adversarial attacks as an upper bound. In all cases, we select the adversarial attack that causes the highest rank change to report the maximum effect for each document. Following our previous observations that re-writing attacks have a smaller impact than injection attacks, we only include injection attacks in our retrieval effectiveness experiments to maintain our focus on lower/upper bounds. We report nDCG@10 and Precision@10 (albeit controversial [37], we leave out MRR because MRR has several shortcomings [14, 46]). All neural models re-rank the top 1000 documents retrieved by BM25. We report significance compared to the original documents using a Student's t-test with Bonferroni correction.

Table 7 shows the maximum impact of our adversarial attacks by contrasting the worst case (lower bound effectiveness) with the original effectiveness (documents are not manipulated) and the best case (upper bound effectiveness). The

Table 5: The MRC and SR (grey subscript) of keyword stuffing on neural models. Significant changes denoted by * (Bonferroni corrected t-test at p < 0.05).

	BM25		ColBERT		TAS-B		monoT5		Electra	
Token	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20
Prompt Tokens										
true	$-22.0^{*}_{0,s,1}$	$-22.7^{*}_{0,s,1}$	$+2.4^{*}_{36,s,1}$	$+3.2^{*}_{34,s,1}$	$-0.3^{*}_{42,r,1}$	$-0.5^{*}_{35,r,1}$	$-9.1^{*}_{22,r,1}$	$-9.4^{*}_{21,r,1}$	$+1.2^{*}_{46,e,5}$	$+3.2^{*}_{47,e,5}$
false	$-22.0^{*}_{0,s,1}$	$-22.7^{*}_{0,s,1}$	$-6.0^{*}_{16,e,1}$	$+4.1^{*}_{38,e,1}$	$-4.8^{*}_{30,e,1}$	$+0.6^{*}_{46,e,1}$	$-0.8^{*}_{46,s,5}$	$-2.7^{*}_{33,s,5}$	$-1.1^{*}_{43,e,5}$	$+2.6^{*}_{45,e,5}$
relevant:	$-22.0^{*}_{0,e,1}$	$-22.7^{*}_{0,e,1}$	$+5.3^{*}_{49,e,5}$	$+1.6^{*}_{45,e,5}$	$+4.7^{*}_{50,e,5}$	$+1.3^{*}_{45,e,5}$	$+63.6^{*}_{78,s,5}$	$+51.2^{*}_{75,s,5}$	$+6.9^{*}_{56,e,5}$	$+5.4^{*}_{51,e,5}$
relevant: true	$-41.1^{*}_{0,s,1}$	$-42.9^{*}_{0,s,1}$	$+9.9^{*}_{52,e,5}$	$+3.3^{*}_{44,e,5}$	$+6.8^{*}_{54,e,5}$	$-2.0^{*}_{39,e,5}$	$+31.1^{*}_{64,s,5}$	$+18.3^{*}_{57,s,5}$	$+4.7^{*}_{52,e,3}$	$+3.8^{*}_{48,e,3}$
relevant: false	$-41.1^{*}_{0,e,1}$	$-42.9^{*}_{0,e,1}$	$+6.8^{*}_{52,e,5}$	$+6.9^{*}_{51,e,5}$	$+9.6^{*}_{55,e,5}$	$+10.4^{*}_{52,e,5}$	$+47.4^{*}_{71,s,5}$	$+32.0^{*}_{64,s,5}$	$+3.2^{*}_{47,e,5}$	$+3.4^{*}_{43,e,5}$
Control Tokens										
bar	$-22.0^{*}_{0,s,1}$	$-22.7^{*}_{0,s,1}$	$-8.1^{*}_{12,e,1}$	$-9.2^{*}_{9,e,1}$	$-3.0^{*}_{44,r,1}$	$-4.5^{*}_{38,r,1}$	$-3.5^{*}_{36,e,2}$	$+0.6^{*}_{41,e,2}$	$-7.2^{*}_{25,r,1}$	$-7.3^{*}_{27,r,1}$
baz	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$-0.8^{*}_{18,e,1}$	$+0.8^{*}_{20,e,1}$	$-10.5^{*}_{32,e,1}$	$+2.0^{*}_{48,e,1}$	$+6.6^{*}_{53,s,5}$	$+17.2^{*}_{60,s,5}$	$+1.4^{*}_{44,r,2}$	$+10.7^{*}_{49,r,2}$
information:	$-2.4^{*}_{0,s,1}$	$-2.4^{*}_{0,s,1}$	$-10.3^{*}_{11,r,1}$	$-9.8^{*}_{3,r,1}$	$-1.1^{*}_{44,e,5}$	$+2.1^{*}_{48,e,5}$	$+57.4^{*}_{77,s,5}$	$+41.3^{*}_{70,s,5}$	$-2.1^{*}_{41,e,5}$	$-0.2^{*}_{40,e,5}$
information: bar	$-22.0^{*}_{0,e,1}$	$-22.7^{*}_{0,e,1}$	$-12.3^{*}_{7,e,1}$	$-12.7^{*}_{4,e,1}$	$-3.5^{*}_{41,e,1}$	$-3.8^{*}_{35,e,1}$	$+31.6^{*}_{70,e,5}$	$+38.2^{*}_{71,e,5}$	$-15.4^{*}_{24,r,1}$	$-12.2^{*}_{25,r,1}$
information: baz	$-22.0^{*}_{0,s,1}$	$-22.7^{*}_{0,s,1}$	$-6.6^{*}_{14,e,1}$	$-4.3^{*}_{16,e,1}$	$-10.7^{*}_{34,e,1}$	$+4.4^{*}_{50,e,1}$	$+31.0^{*}_{61,s,5}$	$+37.0^{*}_{61,s,5}$	$-10.2^{*}_{30,e,3}$	$+3.0^{*}_{39,e,3}$
relevant: bar	$-41.1^{*}_{0,r,1}$	$-42.9^{*}_{0,r,1}$	$-2.4^{*}_{29,e,5}$	$-7.2^{*}_{11,e,5}$	$+0.6^{*}_{50,e,5}$	$-4.6^{*}_{32,e,5}$	$+32.0^{*}_{62,s,5}$	$+33.6^{*}_{61,s,5}$	$-7.7^{*}_{33,e,5}$	$-9.3^{*}_{31,e,5}$
information: true	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$+5.4^{*}_{45,e,5}$	$+2.5^{*}_{38,e,5}$	$+1.9^{*}_{51,e,5}$	$+4.9^{*}_{48,e,5}$	$+28.4^{*}_{62,s,5}$	$+13.5^{*}_{54,s,5}$	$+1.3^{*}_{47,e,4}$	$+5.1^{*}_{47,e,4}$
Synonyms										
pertinent	$-22.0^{*}_{0,e,1}$	$-22.7^{*}_{0,e,1}$	$-1.0^{*}_{24,r,1}$	$-1.2^{*}_{29,r,1}$	$+15.4^{*}_{56,e,5}$	$+14.9^{*}_{54,e,5}$	$-4.7^{*}_{41,s,5}$	$-0.7^{*}_{44,s,5}$	$+30.1^{*}_{77,e,5}$	$+28.2^{*}_{71,e,5}$
significant	$-22.0^{*}_{0,e,1}$	$-22.7^{*}_{0,e,1}$	$+2.0^{*}_{42,r,1}$	$-1.0^{*}_{31,r,1}$	$+10.8^{*}_{65,e,5}$	$+9.9^{*}_{59,e,5}$	$+11.3^{*}_{55,s,5}$	$+8.3^{*}_{50,s,5}$	$+27.1^{*}_{75,e,5}$	$+29.2^{*}_{76,e,5}$
related	$-2.4^{*}_{0,s,5}$	$-2.4^{*}_{0,s,5}$	$-2.0^{*}_{24,r,1}$	$-3.9^{*}_{23,r,1}$	$-1.2^{*}_{42,r,1}$	$-3.6^{*}_{35,r,1}$	$-2.1^{*}_{35,e,1}$	$-3.8^{*}_{31,e,1}$	$-3.7^{*}_{32,r,1}$	$-4.5^{*}_{30,r,1}$
associated	$-22.0^{*}_{0,s,1}$	$-22.7^{*}_{0,s,1}$	$-0.5^{*}_{34,r,1}$	$-0.4^{*}_{32,r,1}$	$-0.9^{*}_{42,r,1}$	$-1.9^{*}_{38,r,1}$	$+6.4^{*}_{50,s,5}$	$+3.6^{*}_{49,s,5}$	$-0.8^{*}_{44,r,3}$	$-1.8^{*}_{40,r,3}$
important	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$+1.7^{*}_{36,r,1}$	$-3.9^{*}_{19,r,1}$	$+4.7^{*}_{49,e,5}$	$+3.4^{*}_{47,e,5}$	$-5.2^{*}_{26,e,1}$	$-3.7^{*}_{30,e,1}$	$+25.6^{*}_{78,e,5}$	$+28.3^{*}_{79,e,5}$
Sub-Words										
relevancy	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$+7.1^{*}_{35,s,1}$	$+7.6^{*}_{38,s,1}$	$-1.4^{*}_{47,r,1}$	$+1.0^{*}_{46,r,1}$	$+12.9^{*}_{54,s,5}$	$+17.6^{*}_{57,s,5}$	$+27.6^{*}_{70,e,5}$	$+30.9^{*}_{68,e,5}$
relevance	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$-2.7^{*}_{28,r,1}$	$-2.1^{*}_{30,r,1}$	$-1.7^{*}_{41,r,1}$	$-1.1^{*}_{40,r,1}$	$-2.3^{*}_{44,s,5}$	$+1.5^{*}_{44,s,5}$	$-2.3^{*}_{45,r,2}$	$+0.5^{*}_{45,r,2}$
relevantly	$-22.0^{*}_{0,r,1}$	$-22.7^{*}_{0,r,1}$	$+0.4^{*}_{32,r,1}$	$-0.6^{*}_{31,r,1}$	$+6.1^{*}_{57,e,5}$	$+5.9^{*}_{55,e,5}$	$+13.5^{*}_{61,s,5}$	$+14.1^{*}_{61,s,5}$	$+22.5^{*}_{66,r,5}$	$+27.0^{*}_{65,r,5}$
irrelevant	$-22.0^{*}_{0,e,1}$	$-22.7^{*}_{0,e,1}$	$+2.6^{*}_{42,r,1}$	$+0.8^{*}_{38,r,1}$	$+5.9^{*}_{53,r,1}$	$+4.1^{*}_{48,r,1}$	$+30.5^{*}_{68,s,5}$	$+34.5^{*}_{69,s,5}$	$+11.5^{*}_{60,e,5}$	$+15.1^{*}_{60,e,5}$

Table 6: Overview of the MRC and SR (subscript) for re-writing with paraphrasing (Par.) and by prepending a summary (Sum.) for Alpaca and ChatGPT. Significant changes denoted with * (Bonferroni corrected t-test at p < 0.05).

		BM25		ColBERT		TAS-B		monoT5		Electra	
	LLM	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20	DL19	DL20
Par.	Alpaca	-14.9^{*}_{20}	-13.6^{*}_{20}	$+1.3^{*}_{45}$	$+1.0_{44}$	$+0.4_{48}$	0.0_{46}	$+2.4^{*}_{51}$	$+1.9^{*}_{50}$	$+4.1^{*}_{55}$	$+3.8^{*}_{54}$
	ChatGPT	-27.1^{*}_{9}	-26.9^{*}_{9}	$+1.3^{*}_{50}$	$+0.2_{48}$	$+1.3^{*}_{52}$	$+0.5_{48}$	$+3.0^{*}_{56}$	$+2.2^{*}_{54}$	$+2.6^{*}_{55}$	$+1.9^{*}_{53}$
Sum.	Alpaca	$+\ 3.9^{*}_{56}$	$+\ 3.9^{*}_{56}$	0.0_{40}	-0.2_{38}	$+1.7^{*}_{48}$	$+1.3^{*}_{47}$	$+2.9^{*}_{53}$	$+2.5^{*}_{51}$	$+4.0^{*}_{54}$	$+3.2^{*}_{53}$
	ChatGPT	$+\ 3.0^{*}_{55}$	$+\ 2.4^{*}_{51}$	-2.0^{*}_{35}	-1.8^{*}_{34}	$+0.1_{45}$	-0.2_{42}	$+1.9^{*}_{50}$	$+0.6_{46}$	$+3.0^{*}_{54}$	$+2.4^{*}_{52}$

injection attacks do not impact BM25, as the retrieval scores never increase by adding non-query tokens to documents. In all other cases, adversarial attacks have a substantial impact on the retrieval effectiveness as the lower and upper bounds introduce, in almost all cases, significant changes, causing our attacks to degrade retrieval effectiveness at scale (the lower bound on nDCG@10 for ColBERT of 0.66 being the only exception). Adversarial attacks have the highest impact on monoT5 (only TAS-B on TREC DL 2020 has the same lower/upper-bound variance of nDCG@10). Importantly, for system-oriented evaluations, we observe that the system rankings are unstable across the different scenarios for

Table 7: The retrieval effectiveness when adversarial attacks are applied to non-relevant documents (worst case), to no documents (original case), or to only relevant documents (best case). We report nDCG@10 and Precision@10 where * marks Bonferroni corrected significant changes to the no-attack scenario.

	TREC DL 19						**TREC DL 20**					
	nDCG@10			**Precision@10**			**nDCG@10**			**Precision@10**		
	Worst	Ori.	Best	Worst	Ori.	Best	Worst	Ori.	Best	Worst	Ori.	Best
BM25	0.48	0.48	0.48	0.60	0.60	0.60	0.49	0.49	0.49	0.58	0.58	0.58
ColBERT	0.66	0.68	0.71*	0.74*	0.77	0.82*	0.62*	0.66	0.69*	0.64*	0.69	0.73*
Electra	0.69*	0.71	0.73*	0.77*	0.80	0.83*	0.67*	0.70	0.73*	0.70*	0.74	0.78*
monoT5	0.67*	0.70	0.73*	0.74*	0.79	0.85*	0.64*	0.68	0.72*	0.66*	0.71	0.77*
TAS-B	0.67*	0.69	0.72*	0.75*	0.78	0.82*	0.62*	0.66	0.70*	0.68*	0.71	0.76*

nDCG@10 and Precision@10. For instance, monoT5 is with an nDCG@10 of 0.70 more effective than TAS-B with 0.69 on TREC DL 2019 in the original case but less effective in the best case (0.73 for monoT5 vs. 0.72 for TAS-B). Overall, adversarial attacks have a high impact in the comparison, e.g., with the paradigm change introduced by BERT, effectiveness shot up by around 0.08 MRR on the MS MARCO test set [24], but adversarial attacks introduce even larger changes, e.g., 0.08 nDCG@10 or even 0.11 Precision@10 for monoT5 on TREC DL 2020.

5 Conclusion

We presented query-independent adversarial attacks against prompt-based sequence-to-sequence relevance models. By exploiting monoT5's prompt structure, we found the attacks successful in more than 78%. Furthermore, we showed that these attacks transfer to other classes of relevance models, such as encoder-only cross-encoders and bi-encoders. From a content provider's perspective, these attacks can be seen as an effective SEO approach, resulting in mean rank improvements of over 63 places. From a search provider's perspective, the attacks pose a marked risk to search engine effectiveness, which is an important finding given that the research field of information retrieval is moving towards more prompt-based models. Looking at how to harden neural relevance models against our simple adversarial attacks is an important direction for future work, especially given recent state-of-the-art sequence-to-sequence approaches to ranking and the proposal of automatic data labeling by large language models.

Acknowledgments

Partially supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.EU).

Bibliography

[1] Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., Lin, J.: Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3490–3496, Association for Computational Linguistics, Hong Kong, China (Nov 2019), URL https://aclanthology.org/D19-1352

[2] Askari, A., Aliannejadi, M., Kanoulas, E., Verberne, S.: A test collection of synthetic documents for training rankers: Chatgpt vs. human experts. In: Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T. (eds.) Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023, pp. 5311–5315, ACM (2023)

[3] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. CEUR Workshop Proceedings **1773** (Nov 2016), ISSN 16130073, URL https://arxiv.org/abs/1611.09268v3, publisher: CEUR-WS

[4] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners (Jul 2020), URL http://arxiv.org/abs/2005.14165, arXiv:2005.14165 [cs]

[5] Camara, A., Hauff, C.: Diagnosing BERT with Retrieval Heuristics. In: Jose, J.M., Yilmaz, E., Magalhaes, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) Advances in Information Retrieval, pp. 605–618, Lecture Notes in Computer Science, Springer International Publishing, Cham (2020), ISBN 978-3-030-45439-5, https://doi.org/10.1007/978-3-030-45439-5_40

[6] Cormack, G.V., Smucker, M.D., Clarke, C.L.A.: Efficient and effective spam filtering and re-ranking for large web datasets. Inf. Retr. **14**(5), 441–465 (2011)

[7] Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2020 deep learning track. In: Voorhees, E.M., Ellis, A. (eds.) Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020, NIST Special Publication, vol. 1266, National Institute of Standards and Technology (NIST) (2020)

[8] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 Deep Learning track. arXiv 2003.07820 (Mar 2020), URL https://arxiv.org/abs/2003.07820v2

[9] Dai, Z., Zhao, V.Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K.B., Chang, M.: Promptagator: Few-shot dense retrieval from 8 examples. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net (2023), URL https://openreview.net/pdf?id=gmL46YMpu2J

[10] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (May 2019), URL http://arxiv.org/abs/1810.04805, arXiv:1810.04805 [cs]

[11] Faggioli, G., Dietz, L., Clarke, C.L.A., Demartini, G., Hagen, M., Hauff, C., Kando, N., Kanoulas, E., Potthast, M., Stein, B., Wachsmuth, H.: Perspectives on large language models for relevance judgment. In: Yoshioka, M., Kiseleva, J., Aliannejadi, M. (eds.) Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023, pp. 39–50, ACM (2023), URL https://doi.org/10.1145/3578337.3605136

[12] Formal, T., Piwowarski, B., Clinchant, S.: A study of lexical matching in neural information retrieval - abstract⋆. In: Tamine, L., Amigó, E., Mothe, J. (eds.) Proceedings of the 2nd Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2022), Samatan, Gers, France, July 4-7, 2022, CEUR Workshop Proceedings, vol. 3178, CEUR-WS.org (2022), URL https://ceur-ws.org/Vol-3178/CIRCLE_2022_paper_11.pdf

[13] Fröbe, M., Akiki, C., Potthast, M., Hagen, M.: Noise-reduction for automatically transferred relevance judgments. In: Barrón-Cedeño, A., Martino, G.D.S., Esposti, M.D., Sebastiani, F., Macdonald, C., Pasi, G., Hanbury, A., Potthast, M., Faggioli, G., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 13th International Conference of the CLEF Association (CLEF 2022), Lecture Notes in Computer Science, vol. 13390, pp. 48–61, Springer, Berlin Heidelberg New York (Sep 2022)

[14] Fuhr, N.: Some common mistakes in IR evaluation, and how they can be avoided. SIGIR Forum **51**(3), 32–41 (2017)

[15] Giomelakis, D., Karypidou, C., Veglis, A.A.: SEO inside newsrooms: Reports from the field. Future Internet **11**(12), 261 (2019)

[16] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (Mar 2015), URL http://arxiv.org/abs/1412.6572, arXiv:1412.6572 [cs, stat]

[17] Gyöngyi, Z., Garcia-Molina, H.: Spam: It's not just for inboxes anymore. Computer **38**(10), 28–34 (2005)

[18] Hofstatter, S., Althammer, S., Schroder, M., Sertkan, M., Hanbury, A.: Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation (Jan 2021), URL http://arxiv.org/abs/2010.02666, arXiv:2010.02666 [cs]

[19] Jeronymo, V., Bonifacio, L.H., Abonizio, H., Fadaee, M., de Alencar Lotufo, R., Zavrel, J., Nogueira, R.F.: Inpars-v2: Large language models as efficient dataset generators for information retrieval. CoRR **abs/2301.01820** (2023), URL https://doi.org/10.48550/arXiv.2301.01820

[20] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense Passage Retrieval for Open-Domain Question Answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020., pp. 6769–6781 (Nov 2020), URL https://aclanthology.org/2020.emnlp-main.550

[21] Kelly, D., Azzopardi, L.: How many results per page?: A Study of SERP Size, Search Behavior and User Experience. In: SIGIR, pp. 183–192, ACM (2015)

[22] Khattab, O., Zaharia, M.: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 39–48 (Apr 2020), URL https://arxiv.org/abs/2004.12832v2, iSBN: 9781450380164 Publisher: Association for Computing Machinery, Inc

[23] Lewandowski, D., Sünkler, S., Yagci, N.: The influence of search engine optimization on google's results: A multi-dimensional approach for detecting SEO. In: Hooper, C., Weber, M., Weller, K., Hall, W., Contractor, N., Tang, J. (eds.) WebSci '21: 13th ACM Web Science Conference 2021, Virtual Event, United Kingdom, June 21-25, 2021, pp. 12–20, ACM (2021)

[24] Lin, J., Nogueira, R.F., Yates, A.: Pretrained Transformers for Text Ranking: BERT and Beyond. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2021)

[25] Liu, J., Kang, Y., Tang, D., Song, K., Sun, C., Wang, X., Lu, W., Liu, X.: Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 2025–2039, CCS '22, Association for Computing Machinery, New York, NY, USA (Nov 2022), ISBN 978-1-4503-9450-5, URL https://dl.acm.org/doi/10.1145/3548606.3560683

[26] Liu, Y.A., Zhang, R., Guo, J., de Rijke, M., Chen, W., Fan, Y., Cheng, X.: Topic-oriented Adversarial Attacks against Black-box Neural Ranking Models (Apr 2023), URL http://arxiv.org/abs/2304.14867, arXiv:2304.14867 [cs]

[27] MacAvaney, S., Feldman, S., Goharian, N., Downey, D., Cohan, A.: ABNIRML: Analyzing the Behavior of Neural IR Models. Transactions of the Association for Computational Linguistics **10**, 224–239 (2022), URL https://aclanthology.org/2022.tacl-1.13

[28] MacAvaney, S., Soldaini, L.: One-shot labeling for automatic relevance estimation. In: Chen, H., Duh, W.E., Huang, H., Kato, M.P., Mothe, J., Poblete, B. (eds.) Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, pp. 2230–2235, ACM (2023)

[29] MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: CEDR: Contextualized Embeddings for Document Ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1101–1104 (Jul 2019), URL http://arxiv.org/abs/1904.07094, arXiv:1904.07094 [cs]

[30] Malaga, R.A.: Chapter 1 — search engine optimization: Black and white hat approaches. In: Advances in Computers: Improving the Web, Advances in Computers, vol. 78, pp. 1–39, Elsevier (2010), URL https://www.sciencedirect.com/science/article/pii/S0065245810780013

[31] Nogueira, R., Cho, K.: Passage Re-ranking with BERT (Apr 2020), URL http://arxiv.org/abs/1901.04085, arXiv:1901.04085 [cs]

[32] Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document Ranking with a Pretrained Sequence-to-Sequence Model. In: Findings of the Association for Computational Linguistics: EMNLP 2020. 2020., pp. 708–718 (Nov 2020), URL https://aclanthology.org/2020.findings-emnlp.63

[33] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27, pp. 517–519, Springer (2005)

[34] Pradeep, R., Liu, Y., Zhang, X., Li, Y., Yates, A., Lin, J.: Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. In: Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I, pp. 655–670, Springer-Verlag, Berlin, Heidelberg (Apr 2022), ISBN 978-3-030-99735-9, URL https://doi.org/10.1007/978-3-030-99736-6_44

[35] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Jul 2020), URL http://arxiv.org/abs/1910.10683, arXiv:1910.10683 [cs, stat]

[36] Raval, N., Verma, M.: One word at a time: adversarial attacks on retrieval models (Aug 2020), URL http://arxiv.org/abs/2008.02197, arXiv:2008.02197 [cs]

[37] Sakai, T.: On fuhr's guideline for IR evaluation. SIGIR Forum **54**(1), 12:1–12:8 (2020)

[38] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings (Dec 2013), URL https://arxiv.org/abs/1312.6199v4, publisher: International Conference on Learning Representations, ICLR

[39] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford Alpaca: An Instruction-following LLaMA model (2023), URL https://github.com/tatsu-lab/stanford_alpaca, publication Title: GitHub repository

[40] Thomas, P., Spielman, S., Craswell, N., Mitra, B.: Large language models can accurately predict searcher preferences (2023)

[41] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., RoziÃ¨re, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models (Feb 2023), URL http://arxiv.org/abs/2302.13971, arXiv:2302.13971 [cs]

[42] Voorhees, E.M., Craswell, N., Lin, J.: Too many relevants: Whither cranfield test collections? In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022, pp. 2970–2980, ACM (2022)

[43] Weller, O., Lawrie, D., Van Durme, B.: NevIR: Negation in Neural Information Retrieval. arXiv 2305.07614 (2023), URL https://arxiv.org/abs/2305.07614, publisher: arXiv Version Number: 1

[44] Wu, C., Zhang, R., Guo, J., de Rijke, M., Fan, Y., Cheng, X.: PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models (Jun 2022), URL http://arxiv.org/abs/2204.01321, arXiv:2204.01321 [cs]

[45] Zhou, Y., Lei, T., Zhou, T.: A robust ranking algorithm to spamming. CoRR **abs/1012.3793** (2010), URL http://arxiv.org/abs/1012.3793

[46] Zobel, J., Rashidi, L.: Corpus bootstrapping for assessment of the properties of effectiveness measures. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, pp. 1933–1952, ACM (2020)

[47] Zou, A., Wang, Z., Kolter, J.Z., Fredrikson, M.: Universal and transferable adversarial attacks on aligned language models. CoRR **abs/2307.15043** (2023), URL https://doi.org/10.48550/arXiv.2307.15043