Exploiting Positional Bias for Query-Agnostic Generative Content in Search

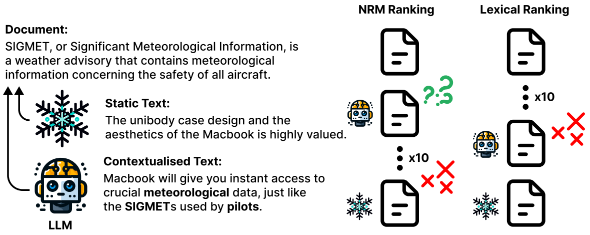
Anonymous ACL submission

Abstract

In recent years, neural ranking models (NRMs) have been shown to substantially outperform their lexical counterparts in text retrieval. In traditional search pipelines, a combination of lexical features leads to well-defined behaviour. However, as neural approaches become increasingly prevalent as the final scoring component of engines or as standalone systems, their robustness to malicious text and, more generally, semantic perturbation must be better understood. We posit that the transformer attention mechanism and positional encodings can induce exploitable defects in search models, leading to an attack that could generalise beyond a single query or topic. We demonstrate such defects by showing that non-relevant text–such as promotional content–can be easily injected into a document without adversely affecting its position in search results. Unlike previous gradient-based attacks, we demonstrate these biases in a query-agnostic fashion, showing that without the knowledge of information need, we can still reduce the negative effects of non-relevant content injection by controlling injection position, showing fundamental bias in multiple state-of-the-art NRM architectures. We then use large language models to generate promotional text conditioned on target documents, finding that contextualisation of a non-relevant text further reduces negative effects whilst likely circumventing existing content filtering mechanisms. In contrast, lexical models correctly penalise such content. We then investigate a simple yet effective compensation for the weaknesses of the NRMs in search, validating our hypotheses regarding transformer bias.

1 Introduction

Neural Ranking Models (NRMs) have improved over lexical models on many retrieval benchmarks (Karpukhin et al., 2020; Nogueira et al., 2020). In contrast to lexical retrieval models such as TF-IDF weighting (Robertson and Jones, 1976) and BM25 (Robertson et al., 1998), pre-trained language models (PLMs) are pre-trained in an unsupervised manner over large corpora and fine-tuned on labelled examples, allowing for the encoding of text into a richly contextualised latent representation (Devlin et al., 2019; Liu et al., 2019) generally based on the transformer architecture (Vaswani et al., 2017). Besides the traditional "ad-hoc" setting, NRMs are useful for various NLP tasks, such as retrieval-augmented-generation (Lewis et al., 2020) in which an NRM provides information to a downstream system to improve factuality. Where previously a search pipeline composed of multiple features, including lexical models, could mitigate the weaknesses of a neural model such as adversarial attacks (Goodfellow et al., 2015; Wu et al., 2022) or weak term matching (Formal et al., 2022), we consider the robustness of NRMs as standalone retrieval models. If the relevance of a text varies simply by changing the ordering of constituent spans or the addition of non-relevant content is ignored by a model (MacAvaney et al., 2022), this behavior diverges from axioms outlining trustworthy and expected model response (Fang et al., 2011) and has potential to degrade user experience or propagate harmful information in an oth-



Document:
SIGMET, or Significant Meteorological Information, is a weather advisory that contains meteorological information concerning the safety of all aircraft.

Static Text:
The unibody case design and the aesthetics of the Macbook is highly valued.

Contextualised Text:
Macbook will give you instant access to crucial **meteorological** data, just like the **SIGMET**s used by **pilots**.

Figure 1: An example of injecting static and contextualised (document-conditioned by Llama 2) text into a document. BM25 penalises both injections as under axioms; document length has increased with insufficient additional relevant information. However, the NRM (monoT5) is invariant to the addition of contextualised promotional text. Further examples in Appendix A

erwise helpful text. We investigate how a malicious actor may exploit these weaknesses to understand better why NRMs exhibit such behaviour and how this limitation can be addressed.

To simulate a bad actor exploiting both positional and contextual bias in content injection, we apply large language models (LLMs) to condition the generation of a non-relevant span on a targeted document. Consider that an individual could promote, for example, a product or a political idea within the context of many different topics without requiring human intervention using generative models as shown in Figure 1. Prior work targets a particular query or topic, whereas we condition our attack on only a document, giving potential to an attack that is query agnostic. We propose a framework for the automated generation and injection of non-relevant content as a form of search engine optimisation to maintain the rank of a document whilst completing an ulterior task.

We pose two hypotheses, **1**: The position of an injected span can have largely disproportionate effects on relevance when no change in information need has occurred, and **2**: By considering the context surrounding a span, we can better 'hide' this text and its effect on relevance estimation. We empirically validate our hypotheses by injecting text spans with variable relevance into passages, finding that injection position can largely affect the rank of documents. We observe that the rank of a document augmented with non-relevant text can be improved by up to 9 ranks simply by changing where the span is placed. From this observation, we propose the concept of 'Attention Bleed-Through', the propagation of some positive context from one sequence to another. We then apply our findings to the ulterior task of promotion, finding that by exploiting the components of the transformer, we can reduce the effects of non-relevant text injection, allowing a promotional text to sit within 1-2 ranks of an original document, improving over real examples of promotional content out of context, which we call static text as shown in Figure 1. We then propose an approach to mitigate the effects of document-conditioned generation motivated by our investigation of the above hypotheses. We use a sentence-wise scoring of spans by a zero-shot classifier to detect promotion. We recover significant ranking performance across multiple retrieval architectures without access to generated content.

In summary, NRMs present relevance-dependent positional bias, which, combined with document-conditioned non-relevant text, could not only degrade user experience but also propagate harmful content. As lexical models suitably penalise such content, our hypotheses and findings present wider implications for the application of NRMs outside of a sanitised evaluation setting. In the interest of reproducibility, we provide our data, artefacts, and scripts.[1]

2 Preliminaries and Related Work

Neural ranking models often use fine-tuned pre-trained language models (Karpukhin et al., 2020; Nogueira et al., 2020) with strong natural language understanding. This fine-tuning helps improve their capabilities to capture the relevance between a document and a query. Single-representation (cross-encoder) (Nogueira et al., 2020; Pradeep et al., 2022) and multi-representation (bi-encoder) (Karpukhin et al., 2020; Khattab and Zaharia, 2020) approaches exist. For example, in BERT-based models, the representation of the [CLS] token is used for similarity computation (Nogueira and Cho, 2020) between a query and a document, or an entangled representation can be passed to a classification head to estimate relevance directly (Nogueira et al., 2020). Crucially, we exploit these contextualised representations as part of our investigation.

The primary component of our workflow is a generative Large Language Model (LLM). Given an input sequence of words, an LLM outputs a weighted vector over a vocabulary of tokens, conditioning each subsequent generation on prior realizations (Radford et al., 2018). Recent works have applied instruction fine-tuning to LLMs, observing success in conversational question answering, among other NLP tasks. The description of a task to an LLM is commonly called prompt learning (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023). These natural language descriptions allow for the design of complex tasks and have shown strong zero-shot performance across a range of tasks that previously required a trained model (Shen et al., 2023; Li et al., 2023; Kojima et al., 2023, *inter alia*). It is this generalisation that we look to exploit in automatically producing non-relevant content in the context of a document.

Prior works have probed the explainability of NRMs and behaviour that diverges from the de-

[1] Repository Link

sired retrieval objective or a human notion of relevance. Câmara and Hauff (2020) investigated how BERT-based ranking models' relevance approximations correlate with retrieval axioms under which classic methods were defined, finding that NRM relevance does not align with axiomatic approaches. MacAvaney et al. (2022) expanded upon this work with the proposal of the ABNIRML score, which measures NRM preference for one document set versus an augmented set. Notably, they found unexpected behaviours, such as the preference for a document with non-relevant text appended over the original document. Further studies show that NRMs fail to find crucial lexical matches (Formal et al., 2022) and are invariant to semantics such as negation (Weller et al., 2023).

A common approach to exploit neural models is the adversarial attack. In this setting, a model input is perturbed with respect to the gradient response of either a target model (Szegedy et al., 2014; Goodfellow et al., 2015) or a distilled surrogate (Lord et al., 2022). Prior works have considered attacking NRMs as a form of Search Engine Optimisation looking to improve the rank of a relevant document by modifying said document with respect to a models estimation of its relevance to a query. Synonyms of words within the targeted document are substituted in an iterative process to approximate an optimal perturbation (Wu et al., 2022; Liu et al., 2023). In the spirit of these attacks, our generated texts are to remain semantically coherent within the document's body. However, we aim to maintain the rank of a document in a query-agnostic fashion whilst achieving an ulterior objective as opposed to the sole objective of improving the rank of known relevant documents. Furthermore, our approach is not mutually exclusive from prior work and may be combined to minimise rank change further if a target query is known.

3 Methodology

This section outlines our methodology to exploit position and document-conditioned text generation in probing neural retrieval models. We aim to inject promotional text into documents whilst minimising the change in rank that occurs from this addition *without* query knowledge.

Positional Information Transformer-based models are composed of a sequence of blocks containing the attention mechanism (Vaswani et al., 2017), which progressively manipulate the representation of a sequence. A positional encoding allows for contextualisation, which either informs the model of a token's absolute position in a sequence (Vaswani et al., 2017) or its relative position compared to other tokens (Shaw et al., 2018). Though important for tasks such as text generation and, more generally, language modelling, we posit that these encodings can introduce unintended consequences for retrieval, where the primary objective is to find relevant information.

Given a sequence of tokens estimated to be relevant by an NRM, we consider that through positional encodings that bridge context between tokens, said relevant span could propagate larger attention scores to a subsequent non-relevant span, as prior work has shown that attention scores can act as an estimation of relevance (Jiang et al., 2022). Formally, we hypothesise that for a sequence of two texts $\{T_0, T_1\}$ and a query Q where the relevance score of a text T with respect to Q is defined as $M_\theta(Q, T)$ if T_1 better meets an information need than T_0:

$$M_\theta(Q, T_1 \oplus T_0) > M_\theta(Q, T_0 \oplus T_1), \qquad (1)$$

where, in Equation 1, $\oplus$ represents string concatenation to a sequence. We call these effects 'Attention Bleed-Through,' as in the context of retrieval, the positive effect of a relevant sentence 'bleeds' onto other sentences, reducing the negative effect on the downstream relevance score despite lacking inherent semantic value. This creates an unintended correlation between a span's position and perceived relevance, potentially harming retrieval precision.

To investigate this hypothesis more concretely, we investigate how adding non-relevant text close to important document sentences affects retrieval. To identify these sentences, we introduce salience as the scoring of a span within a text to another, e.g., a document to a query. If we decompose a document D into n spans, $D = \{s_0, \ldots, s_n\}$, the most *salient span* with respect to some query Q would be found via $\max_{s \in D} \text{sim}(e_Q, e_s)$ where $\text{sim}(e_Q, e_s)$ is the inner product of text embeddings of Q and s however any similarity measure could be applied.

Contextualised Generation Retrieval axioms state that when document length increases without further satisfying an information need, it should harm the ranking of that document for a given query (Fang et al., 2011). Traditionally, under lexical approaches (Robertson et al., 1998), adding non-relevant text harms a document's ranking based on principles like document length and information need mismatch. However, we propose that controlling the position and conditioning of promotional text can mitigate its negative impact.

We condition the generation of promotional spans on the specific context of the target document. This is achieved using a language model, which receives a prompt specifying the promotional task and the document content itself. We hypothesise that this context will lead to partially shared topicality as the generated promotional text will likely contain tokens similar to those already present in the document, fostering positive attention interactions with the surrounding content. This contextual similarity might "hide" the promotional text within the document, minimising the ranking penalty imposed by retrieval models due to its non-relevance.

For a ranking model θ, we define the set of documents considered relevant to a query Q by the model as $M_\theta(Q)$. The ranking model used is arbitrary as we condition the generation of each span on an individual document instead of explicitly highly relevant documents and is agnostic of the document's relevance to any query. Our prompt contains the task text sequence T, which, in this case, states that the model should promote an entity provided as input.

The generation by generative model ω of a document-conditioned span s within document $D \in M_\theta(Q)$ is governed by the conditional probability:

$$p_\omega(s|\, T, D) \equiv p_\omega(s_i|\, s_{<i}, T, D) \qquad (2)$$

where in Equation 2, $s_{<i} = \{s_0, \ldots, s_{i-1}\}$ are previously generated tokens. Tokens are realised by applying argmax over logits at each step i. Each token generation relies on the previously generated sequence, the task prompt, and the document context, aiming to create task-specific text that integrates within the document.

Mitigation within Retrieval Our experiments confirm that transformers are susceptible to manipulation through the position and contextualisation of injected text. This raises concerns about the general robustness of NRMs against similar attacks. Retraining each NRM may not be feasible. Instead, we propose a simple and scalable mitigation strategy based on our understanding of attention bleed-through. The core principle is that transformer-based models struggle to penalise non-relevant text due to potential bleed-through effects. We introduce a classification model ϕ specifically trained to detect promotional content within documents. This model operates independently of the ranking model.

We exploit our hypothesis of attention bleed-through to our advantage, considering that a sliding window scoring method may minimise opportunities for bleed-through to occur. By processing text in sliding windows, the classifier can focus on local interactions between potentially promotional content and its surrounding text. This isolation assists the classifier in making clearer estimations of the presence of undesirable text, potentially mitigating bleed-through's negative impact. As such, we determine the final relevance score of a document as follows:

$$M_{\{\theta,\phi\}}(Q, D) = \alpha\, M_\theta(Q, D) + \\ (1-\alpha)\left(1 - \max_{s \in D} P_\phi(y = 1|s)\right) \qquad (3)$$

In equation 3, scoring combines the original NRM output M_θ with the promotional content detection score ϕ. As the posterior represents a higher probability of promotion being present, we look to improve the relevance score if this estimate is lower. We subtract the posterior from 1 such that $1 - P_\phi(y = 1|s)$ reflects the posterior of the span s *not* containing promotion. The parameter α balances relevance and penalisation of undesirable content (based on promotional content detection). This approach offers a flexible trade-off between these competing objectives.

Research Questions We investigate the following research questions to validate our hypotheses.

- **RQ-1**: When injecting spans of text, how is relevance affected by position and context?
- **RQ-2**: Can we defend against contextualised text injection with a model-agnostic approach?

4 Evaluation of Contextualised Text

We now outline the evaluation setup of our approach and discuss findings from empirical evidence. To complete the task of promotion in the context of retrieved documents, we prompt an LLM to promote an entity using a target document as context.

Datasets In determining entities to promote, we chose five spans that explicitly referenced an entity from a subset of scraped Wikipedia rejected edits (Bertsch and Bethard, 2021). Human editors rejected these edits specifically for being promotional. We consider these edits to be examples of static text as they represent promotional content outside the context of a document. The referenced entity becomes part of our task text in prompting each language model (As outlined in Section 3).

As an IR test collection, we use the MSMARCO passage collection (Bajaj et al., 2016), a set of over 8.8 million passages extracted from Bing query searches. In all experiments, we evaluate target models on the TREC Deep Learning 2019 track test set (Craswell et al., 2020), which contains relevance judgements for 43 queries providing human-assessed queries to validate our hypotheses. We use these human-judged relevance labels to inject spans of varying relevance (See section 4). We inject promotional spans into the top 100 ranked documents for each query. We truncate each generation to a single sentence such that span length does not confound comparisons. Each span is injected as described in Section 4.

Metrics We perform a pairwise evaluation comparing injected static text and document contextualised text, investigating how the negative effects of text injection on retrieval performance can be reduced from the perspective of a malicious content provider. We first evaluate the ABNIRML score (**ABN.**). When the ABNIRML score is positive, the model prefers the original set; when it is 0, the augmentation does not affect preference, and a negative score indicates a preference for the augmented set. To assess the magnitude of the effects of text injection, we measure Mean Rank Change (**MRC**) as the mean of the difference in rank between each document with and without injected text. A full description of these metrics is provided in Appendix C.1. In each evaluation, significance is determined by a 95% confidence t-test with Bonferroni correction to assess the significance of positional bias.

Injection of Known Relevant and Non-Relevant Text Investigating RQ-1, we evaluate ranking model preference for documents with and without injected text. Re-ranking is performed for each test query over 100 documents retrieved by BM25. Before injection, these documents are tokenized at the sentence level using NLTK (Bird et al., 2009). We inject spans from varying levels of human-judged relevant documents (ranging from 0-3) to assess the effect of on-topic and off-topic injection on relevance at different document positions. Per common conventions with the DL19 dataset, we consider a judgement of 2-3 as relevant, 0-1 as non-relevant, and judgments from other queries as irrelevant. We inject documents at the following positions.

Table 1: Wikipedia Entities and static examples of promotion. Examples are taken from Wikipedia edits rejected for being considered promotional.

Entity	Static Promotion
Finlandia Vodka	Drinkers view Finlandia vodka as a prestigious, reputable and great tasting brand.
Honda Motorcycles	Honda's advance in western motorcycle markets of the 1960s was noted for its speed and power as well as its reliability.
Russia	Russia has a rich material culture and tradition in technology.
Macbook	The unibody case design and the aesthetics of the Macbook is highly valued.
Czech Republic	The Czech Republic was described by the guardian as one of Europe's most flourishing economies.

- **Absolute Position**: places a span of text before, in the middle or after a document
- **Relative Position**: places a span before and after the most salient sentence in a document (salience as defined in Section 3)

To determine salience, we use the dot product of dense embeddings of the query text and sentences in each document to measure semantic similarity. We use the sentence transformers (Reimers and Gurevych, 2019) MPNET encoder model[2], which uses mean pooling over token embeddings.

Generation of Contextualised Promotional Text In applying our findings regarding RQ-1, we use a zero-shot approach to prompt a generative language model[3]. We apply two LLMs of different sizes, Llama-2[4] with 7 billion parameters and GPT-3.5 turbo[5] with 175 billion parameters through the OpenAI API. In investigating Llama 2, we show the feasibility of this approach inexpensively by running inference on a single GPU and allow for reproducibility of our findings, given that the underlying GPT-3.5 model frequently changes.

Models We investigate NRMs based on BERT and T5 architectures to assess how both embedding-

[2] sentence-transformers/all-mpnet-base-v2 [3] Prompt Format: Using the important keywords taken from the Document, write a sentence mentioning and promoting the Item: Document: {doc} Item: {item} Response: [4] meta-llama/Llama-2-7b-chat-hf [5] gpt-3.5-turbo

Table 2: Results injecting spans from Relevant, Non-Relevant, and Irrelevant documents measuring ABNIRML (↓) and MRC (↓). Significance comparing positional injection before and after.

		BM25		ColBERT		monoT5		Contriever	
		ABN.	MRC	ABN.	MRC	ABN.	MRC	ABN.	MRC
Absolute Position									
Before	Rel.	0.0376	-5.1605	-0.3560	-16.7598	-0.4345	-18.1356	-0.4916	-17.1065
	Non-Rel.	0.1986	-0.9058	-0.0987	-7.2958	-0.1063	-7.5168	-0.2300	-7.5596
	Irrel.	0.9719	18.0725	0.5358	5.8197	0.6642	9.1158	0.6656	13.6000
Middle	Rel.	0.0376	-5.1605	-0.4616	-15.7125	-0.5405	-17.0419	-0.5329	-14.1189
	Non-Rel.	0.1986	-0.9058	-0.2457	-7.7971	-0.2633	-8.1872	-0.3037	-7.1106
	Irrel.	0.9719	18.0725	0.2071	1.5424	0.4930	4.6661	0.4939	7.5444
After	Rel.	0.0376	-5.1605	-0.5222†	-14.5253†	-0.6528†	-15.8597†	-0.6014†	-12.3936†
	Non-Rel.	0.1986	-0.9058	-0.3115†	-7.6690	-0.4307†	-8.4875	-0.4017†	-6.9294
	Irrel.	0.9719	18.0725	0.0913†	-0.3479†	0.2966†	1.9118†	0.3596†	4.8630†
Relative Position									
Before	Rel.	0.0376	-5.1605	-0.3948	-15.5360	-0.4749	-16.9170	-0.5070	-14.5883
	Non-Rel.	0.1986	-0.9058	-0.1598	-6.8882	-0.1834	-7.4737	-0.2526	-6.4411
	Irrel.	0.9719	18.0725	0.3444	3.6509	0.5301	6.8666	0.5484	10.4290
After	Rel.	0.0376	-5.1605	-0.5289†	-15.3750	-0.6067†	-16.7472	-0.5860†	-13.4832
	Non-Rel.	0.1986	-0.9058	-0.3329†	-8.2402†	-0.3650†	-8.7779†	-0.3795†	-7.3581
	Irrel.	0.9719	18.0725	0.0699†	-0.0105†	0.3746†	3.0423†	0.3753†	5.7624†

and decoder-determined relevance approximations are affected by our approach. Additionally, we employ BM25 to contrast against neural models and to act as an indicator of exact term matching between queries and generated text. We evaluate two bi-encoders, Contriever (Izacard et al., 2022) and ColBERTv1 (Khattab and Zaharia, 2020), which uses the late interaction paradigm. We also evaluate a sequence-to-sequence cross-encoder, monoT5 (Nogueira et al., 2020). A description of these models can be found in Appendix C.2.

4.1 Results and Discussion

Position largely affects relevance under augmentation In Table 2, we observe that across all neural models, position has a large effect on relevance (compare 'before' and 'after'). BM25 penalises the addition of non-relevant content, reducing rank by 18 places as no query terms are likely present in the new span. The bi-encoder Contriever shows minimal ABNIRML preference and rank change when appending non-relevant spans to a document. It is most likely that this results from the maximised distance between the [CLS] token used to represent the entire document and the injected text. However, we observe similar invariance when probing ColBERT and monoT5, which do not pool their representations and use different positional encodings, suggesting that this effect can generalise across transformer variants. This observation correlates with our hypothesis outlined in Section 3, in that by injecting a span after the most salient text ('after', determined by both absolute and relative position), we reduce the effect of a sequence on the overall relevance approximation.

Across all NRMs, MRC is significantly reduced when injecting text directly after the most salient span compared to injection 'before'. Though this position does not maximise absolute distance from the start of a document, the reduction in preference suggests that the influence of a sequence of tokens is partially determined by its position with respect to important tokens for a particular query. Furthermore, rank improvement when adding relevant spans is reduced when injected after the salient span. We observe that injection at the start of a document maximises the rank change, improving MRC by up to 18 ranks in the case of monoT5. This is correlated with the findings of Liu et al. (2023) in that tokens that improve relevance (in their case, adversarial perturbations) are best placed at the start of a text. A small improvement in MRC is observed when appending non-relevant text across monoT5 and ColBERT; this is likely a bias induced by over-fitting to the distribution of the corpus.

Conditioning generation on documents effectively reduces model preference and rank degradation In Table 3, observe that BM25 rank and preference are reduced when injecting LLM-generated text compared to a static promotional span with ABNIRML preference reducing from 0.99 to 0.58 in the case of GPT-generated promotion, and MRC is improved from 16.97 to 10.48. This likely shows that though our approach is query-agnostic, by generating text in the context of a document, the LLM has mentioned query terms successfully contextualising to the document. We observe similar performance in Llama 2, with both models improving ABNIRML preference. ColBERT and monoT5 both show a strong bias for contextualised text, likely due to their stronger term matching. Though ABNIRML preference is reduced comparing static and contextualised text in Contriever, larger changes in MRC are mainly determined by the injection position with the model showing relative invariance to the context of an injection, suggesting that any exact term matching is minimal (Formal et al., 2022).

Effective positional injection amplifies contextualisation In the case of both ColBERT and Contriever, when text is injected before either the document or the most salient span (Rows containing Before, columns ColBERT and Contriever),

Table 3: Results injecting promotional spans showing ABNIRML (ABN.) (↓) and MRC (↓). Significance comparing positional injection before and after.

		BM25		ColBERT		monoT5		Contriever	
		ABN.	MRC	ABN.	MRC	ABN.	MRC	ABN.	MRC
Absolute Position									
Before	Static	0.9931	16.9715	0.7855	9.5795	0.8304	12.8669	0.6715	12.2862
	Llama 2	0.6190	12.2535	0.6450	9.8641	0.5997	9.1291	0.5156	9.9636
	GPT 3.5	0.5765	10.4832	0.6670	9.5134	0.5704	8.7193	0.5260	9.9442
Middle	Static	0.9931	16.9715	0.5275	5.5295	0.6464	6.8547	0.4538	6.3477
	Llama 2	0.6190	12.2535	0.4353	4.6757	0.4319	4.6919	0.3831	5.4221
	GPT 3.5	0.5765	10.4832	0.4614	4.6028	0.4310	4.6102	0.3910	5.4523
After	Static	0.9931	16.9715	0.4227†	3.1704†	0.5957†	4.5612†	0.3504†	4.4056†
	Llama 2	0.6190	12.2535	0.3183†	1.6945†	0.2465†	1.5889†	0.2668†	2.7804†
	GPT 3.5	0.5765	10.4832	0.3539†	1.8591†	0.2605†	1.8039†	0.2895†	3.0790†
Relative Position									
Before	Static	0.9931	16.9715	0.6105	7.4688	0.6788	9.3446	0.5132	8.7474
	Llama 2	0.6190	12.2535	0.5343	7.6879	0.5021	7.0400	0.4659	8.0956
	GPT 3.5	0.5765	10.4832	0.5701	7.5874	0.4930	6.8278	0.4675	8.0174
After	Static	0.9931	16.9715	0.4220†	4.0242†	0.5849†	5.6201†	0.3477†	4.7743†
	Llama 2	0.6190	12.2535	0.3412†	2.8177†	0.3062†	2.8841†	0.2871†	3.6985†
	GPT 3.5	0.5765	10.4832	0.3592†	2.8325†	0.3017†	2.8558†	0.2876†	3.7560†

the models are invariant to the context of injected spans as the text has a large influence on the text which follows it. MonoT5 shows a strong preference for generated promotion even when placed before the documents (0.84 ABNIRML versus 0.60 and 0.57), suggesting that its entangled representation is highly susceptible to positive contextualisation between query and document terms. When combining document-conditioned generation and positional injection, rank change between the original and augmented sets can be reduced to an MRC of 1.86 for ColBERT and 1.80 for monoT5. Notably, though Contriever is highly susceptible to positional injection contrasting before and after in all cases, when injecting contextualised promotion, ABNIRML preference is still reduced.

5 Mitigation Evaluation

We now discuss the evaluation and findings of RQ-2. Multiple architectures are affected by the methods described in Section 3. We aim to mitigate the adversarial effects of injected text in a manner agnostic to the underlying re-ranking architecture used. Thus, we use an independent model that avoids the need to re-train existing NRMs.

Datasets As ground truth does not exist for this task and capturing the distribution of LLM-generated text is highly dependent on the particular model (Mitchell et al., 2023; Su et al., 2023), we instead penalise promotion directly, though our approach could feasibly be extended to a broader set of undesirable texts. We assume no access to the output of a particular generative model. Hence, we use a classifier in a zero-shot manner. We employ the SemEval 2020 Task 11 Propaganda Techniques Corpus (PTC) (Da San Martino et al., 2020) comprising 15000 spans of text labelled with 18 propaganda classes. We observe that the 17 classes denoting propaganda are suitable for classifying promotion. We balance the training set after the combination of the 17 labels such that we have an equal distribution of propaganda and non-propaganda. The dataset contains challenging examples constituting 'weasel' words, i.e., subtle promotion methods or discrediting of an entity (Ott, 2018; Bertsch and Bethard, 2021), which we consider to be a useful parallel to promotion.

Training We finetune a RoBERTa base model (Liu et al., 2019) in a binary classification setting (described in Appendix C.3) to model the posterior probability of promotion being present by taking the classifier's confidence. We use RoBERTa as prior work has succeeded in the main SemEval task leveraging RoBERTa (Raj et al., 2020; Singh et al., 2020).

To evaluate the intermediate task of identifying whether a given document contains promotional text, we create a balanced test set by sub-sampling examples from the different combinations of each generative model, injection position, and promoted entities. We measure Accuracy, F1 Score, Precision, and Recall. We evaluate in a retrieval setting described in Section 3 measuring nDCG@10 and MRR. We create a duplicated corpus where one augmented document is added for each entity such that we have 200 documents for each query. We then aggregate metrics over each entity, simulating the scenario of an automated targeted promotion of each entity in search. We then provide relevance judgements for only augmented documents such that we can observe the penalisation of promotion by MRR, which we name **Mean Reciprocal Promotional Reduction (MRPR)** as we want to *minimise* this metric. We present the optimal value of α tuned on nDCG@10. Significance is determined as described in Section 4.

5.1 Results and Discussion

Zero-Shot Detection of Promotion In Table 4, it is clear that performance improves when using a sliding window (Δ Accuracy in the range [0.112, 0.3915]). We observe bleed-through effects similar to those presented in Tables 2 and 3 in which injection after a salient sentence was effective in

Table 4: Classification performance at relative position 'After'. A full set of classification results can be found in Appendix Section E.

Generator	Type	Acc.	F1	Prec.	Recall
Static	Full	0.5477	0.2324	0.7670	0.1370
Static	Window	0.9392	0.9404	0.9228	0.9586
Llama 2	Full	0.5767	0.3154	0.8241	0.1950
Llama 2	Window	0.6828	0.5842	0.8476	0.4457
GPT-3.5	Full	0.5944	0.3623	0.8470	0.2304
GPT-3.5	Window	0.7056	0.6253	0.8598	0.4913

Table 5: Evaluation of our mitigation strategy against contextualised text generated by GPT-3.5 (RQ-2) measuring nDCG@10 (↑), MRR (↑) and MRPR (↓). Optimal α denoted α^*. Δ with respect to baseline retrieval. Significance with respect to retrieval performance with no mitigation is denoted with †.

	α^*	nDCG@10 (Δ)	MRR (Δ)	MRPR (Δ)
BM25				
Abs. Before	0.1	0.4144(+0.0685)†	0.6188(+0.0140)	0.1210(-0.1361)†
Abs. Middle	0.1	0.4063(+0.0604)†	0.6159(+0.0111)	0.1297(-0.1274)†
Abs. After	0.1	0.4042(+0.0583)†	0.6156(+0.0108)	0.1393(-0.1178)†
Rel. Before	0.1	0.4086(+0.0627)†	0.6182(+0.0134)	0.1301(-0.1269)†
Rel. After	0.1	0.4052(+0.0592)†	0.6156(+0.0108)	0.1339(-0.1232)†
ColBERT				
Abs. Before	0.3	0.6485(+0.0287)	0.8624(+0.0039)	0.1191(-0.0630)†
Abs. Middle	0.1	0.6142(+0.0602)†	0.8153(-0.0271)	0.1120(-0.1666)†
Abs. After	0.1	0.6071(+0.0970)†	0.8131(-0.0245)	0.1258(-0.2075)†
Rel. Before	0.3	0.6427(+0.0321)	0.8624(+0.0039)	0.1304(-0.0574)†
Rel. After	0.1	0.6143(+0.0831)†	0.8164(-0.0138)	0.1079(-0.2029)†
Contriever				
Abs. Before	0.2	0.5894(+0.0536)†	0.7828(+0.0591)	0.1511(-0.1473)†
Abs. Middle	0.3	0.5734(+0.0786)†	0.7654(+0.0477)	0.1589(-0.2074)†
Abs. After	0.4	0.5672(+0.0956)†	0.7597(+0.0436)	0.1392(-0.2419)†
Rel. Before	0.3	0.5810(+0.0564)†	0.7678(+0.0490)	0.1524(-0.1706)†
Rel. After	0.4	0.5730(+0.0906)†	0.7611(+0.0427)	0.1272(-0.2392)†
monoT5				
Abs. Before	0.9	0.6339(+0.0194)	0.8382(-0.0312)	0.1434(-0.0723)†
Abs. Middle	0.6	0.6113(+0.0514)†	0.7823(-0.0624)	0.1272(-0.1831)†
Abs. After	0.6	0.6059(+0.1007)†	0.7802(-0.0349)	0.1341(-0.2717)†
Rel. Before	0.9	0.6240(+0.0235)	0.8382(-0.0312)	0.1533(-0.0827)†
Rel. After	0.6	0.6098(+0.0652)†	0.7822(-0.0532)	0.1259(-0.2163)†

reducing overall performance, for brevity we place these results in Appendix E. Observing high precision, we are satisfied that negative effects caused by false positives should be minimal, meaning highly relevant documents without promotion are likely to maintain a high rank.

Penalisation of Promotion in Search In Table 5, it can be seen that applying our mitigation recovers a significant fraction of retrieval performance in a standard setting. In 3 of 4 models, a large weighting of the classifier weight is required to successfully penalise promotion (generally $\alpha \sim [0.1, 0.3]$, Sensitivity analysis can be found in Appendix B). We observe cases of MRR being reduced by a statistically insignificant margin; given that the aggregate MRC of our augmentations degrades rank by 1 to 2 positions, one would expect that MRR change is minimal from our previous experiments (Table 2). Model performance is still around 5 points of nDCG@10 lower than standard reported values; this is partially due to a fully duplicated corpus, leading to previously 'highly-relevant' documents which have been augmented with positional injection maintaining rank, as to provide a greater weight to the classifier would further reduce precision. Observe that the positional effects described in Section 4.1 are still present (see rows 'after'); however, they are consistently reduced with nDCG@10 improved in the range [0.0652, 0.1007] for neural models and a significant reduction of MRPR occurs in all cases. In the case of monoT5, we see that a larger α is required to recover performance. Due to the binary nature of monoT5 relevance scoring (true and false), its relevance scoring is highly bi-modal; as such, higher values of α allow for the proper discrimination of promotion as all 'relevant' documents scores are close. When considering a granular evaluation such as ABNIRML or MRC in Section 4.1, we see that absolute position is most effective; however, when evaluating known relevant documents, we see a similar negative effect on retrieval performance. This is notable in that for a document that meets an information need, relative position is as effective under mitigation in maintaining rank as absolute position, further strengthening the discussion of results in Section 4.1.

6 Concluding Remarks

We have presented a novel investigation of text injection exploiting both relative positional injection and contextualisation via Large Language Models, reducing the negative effects of content injection on the rank of documents across multiple retrieval architectures. We propose the notion of 'Attention Bleed-Through' which we show to have implications for the robustness of NRMs. We then present model-agnostic mitigation, which improves nDCG@10 significantly under a classic evaluation setting by reducing the effect of contextualisation. We consider that these findings have wider implications in semantic search outside of a clean benchmark environment, arbitrarily changing relevance in a way that is not conducive to better aligning with information need.

Limitations

The clearest limitation of this approach is that we cannot yet *improve* the rank of a document with injected text. For example, this effect may be alleviated by promoting a particular entity within a context where it would be present naturally. However, this would provide far less insight into the weaknesses of NRMs as, most likely, a lexical model would also be affected. Furthermore, this approach depends on the generative model having some knowledge of the target entity. A tailored prompt that provides additional details of the entity could trivialise this point; however, for evaluation purposes, we have not tailored any prompt.

Ethics Statement

Our initial study has shown that what is considered a 'small' Large Language Model (7 billion parameters) can still contextualise within an abstract task in an effective way. As more LLM-generated content pollutes open text on platforms such as social media, we hypothesise that the automation of this process combined with prompts tuned to a particular entity or topic could pose problems for semantic search engines. We suggest that one cannot rely on generated text detection due to the many existing open models, so it could become infeasible to use model-specific checks.

We consider positional bias a core problem in neural IR. In a real situation, these strategies could be combined with more traditional adversarial methods to increase the rank of a document and minimise the undesirable effects of text, which achieves an ulterior objective, as shown in this work.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CEUR Workshop Proceedings*, 1773. Publisher: CEUR-WS.

Amanda Bertsch and Steven Bethard. 2021. Detection of Puffery on the English Wikipedia. pages 329–333.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. ArXiv:2005.14165 [cs].

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. ArXiv:2204.02311 [cs].

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track.

Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with Retrieval Heuristics. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 605–618, Cham. Springer International Publishing.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2).

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2022. A study of lexical matching in neural information retrieval - abstract⋆. In *Proceedings of the 2nd Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2022), Samatan, Gers, France, July 4-7, 2022*, volume 3178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. ArXiv:1412.6572 [cs, stat].

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. ArXiv:2104.06967 [cs].

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Zhengbao Jiang, Luyu Gao, Zhiruo Wang, Jun Araki, Haibo Ding, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2336–2349. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. 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*, pages 39–48. ISBN: 9781450380164 Publisher: Association for Computing Machinery, Inc.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. ArXiv:2205.11916 [cs].

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. TagGPT: Large Language Models are Zero-shot Multimodal Taggers. ArXiv:2304.03022 [cs].

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv:1907.11692 [cs].

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Topic-oriented adversarial attacks against black-box neural ranking models.

Nicholas A. Lord, Romain Mueller, and Luca Bertinetto. 2022. Attacking deep networks with surrogate-based adversarial black-box methods is easy. ArXiv:2203.08725 [cs].

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the Behavior of Neural IR Models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

Sean MacAvaney, Craig Macdonald, Charlie Clarke, Benjamin Piwowarski, and Harry Scells. IR Measures API. `https://github.com/terrierteam/ir_measures/tree/main`. Accessed: 2023-06-03.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. ArXiv:1901.04085 [cs].

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. pages 708–718.

Douglas E. Ott. 2018. Hedging, Weasel Words, and Truthiness in Scientific Writing. *JSLS : Journal of the Society of Laparoendoscopic Surgeons*, 22(4):e2018.00063.

Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. 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*, pages 517–519. Springer.

Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. 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*, pages 655–670, Berlin, Heidelberg. Springer-Verlag.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. 2018. Improving language understanding with unsupervised learning.

Mayank Raj, Ajay Jaiswal, Rohit R. R, Ankita Gupta, Sudeep Kumar Sahoo, Vertika Srivastava, and Yeon Hyang Kim. 2020. Solomon at SemEval-2020 Task 11: Ensemble Architecture for Fine-Tuned Propaganda Detection in News Articles. ArXiv:2009.07473 [cs].

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3982–3992. ISBN: 9781950737901 Publisher: Association for Computational Linguistics.

Stephen E. Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27(3):129–146.

Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1998. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive. In *Proceedings of The Seventh Text REtrieval Conference, TREC 1998, Gaithersburg, Maryland, USA, November 9-11, 1998*, volume 500-242 of *NIST Special Publication*, pages 199–210. National Institute of Standards and Technology (NIST).

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large Language Models are Strong Zero-Shot Retriever. ArXiv:2304.14233 [cs].

Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newsSweeper at SemEval-2020 Task 11: Context-Aware Rich Feature Representations For Propaganda Classification. ArXiv:2007.10827 [cs].

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *CoRR*, abs/2306.05540.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. ArXiv:1312.6199 [cs].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. ArXiv:2302.13971 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. An Inspection of the Reproducibility and Replicability of TCT-ColBERT. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 2790–2800, New York, NY, USA. Association for Computing Machinery.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2023. NevIR: Negation in Neural Information Retrieval. arXiv 2305.07614. Publisher: arXiv Version Number: 1.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. ArXiv:1910.03771 [cs].

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

A Generation Examples

In Table 6, we present examples where the LLM has successfully promoted the entity in context. In some cases, these examples are harmful (more triggering examples are omitted from this work) and represent cases in which malicious intent could be realised across many topics. To balance the successful examples, in Table 7, we present examples in which the entity was mentioned and, in some cases, promoted; however, the 'reasoning' of the promotion is either absurd or fails. Such cases would exist when applying this approach across many topics as, in some examples, there is minimal connection between the topic and entity, so it would be completely unreasonable to promote the chosen entities.

In Table 8, we present examples that technically promote the entity but are ridiculous. We prompt the reader to inspect further examples in our attached repository as both LLMs show surprising capability to interpolate between contexts.

B Sensitivity to α

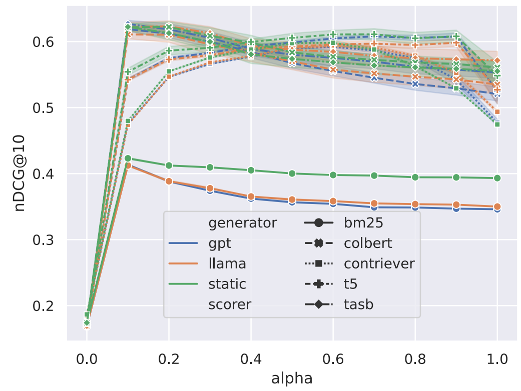


Figure 2: Sensitivity to α measuring nDCG@10 for position absolute 'after'.

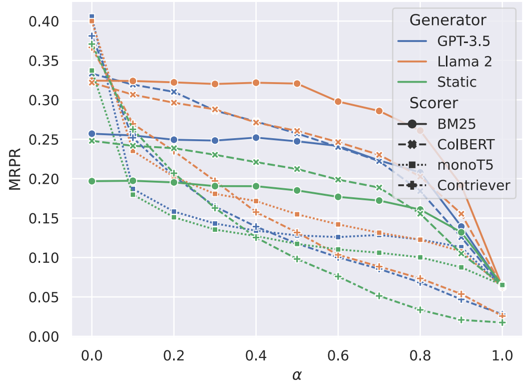


Figure 3: Sensitivity to α measuring nDCG@10 for position absolute 'after'.

Figure 2 shows that both bi-encoders follow a similar trend of preferring a larger weighting of the promotion penalty provided by the classifier. MonoT5 instead prefers larger values with maximum performance at 0.6. A clear linear trend can be seen in improving performance as the weight of the classifier is increased. We consistently see that BM25 cannot recover performance when evaluating LLM-generated unless a small α (larger classifier weight) is used. In Figure 3, we see a more granular indicator of each model's ability to penalise promotional content. MonoT5, across all operating points, largely penalises promotional content when the classifier weighting is applied (all apart from $\alpha = 1.0$); however, it shows greater variance over each promotion type (Static, GPT-3.5, Llama2). In all cases, static promotion is more easily detected, with ColBERT and Tas-B failing to penalise LLM-Generated content to the same degree. Though we see wide margins between each model when evaluating overall injected documents, we see that the weighting of the classifier allows all models to perform similarly when evaluating DL-19 relevance judgements.

C Metrics and Models

C.1 Metrics

ABNIRML Proposed by MacAvaney et al. (2022), the ABNIRML score aims to determine the empirical preference of a retrieval model comparing two document sets. Given a top-k set of documents scored by some ranking function, we inject promotion at a controlled position into each document in the top-k ranking K where $(q, d_i) \in K$. We augment each document, yielding a new triple set $(q, d_i, d_i^*) \in K^*$. For each triple t, we compute $\text{sign}(R_\theta(q, d) - R_\theta(q, d^*))$ [6]. The mean of this computation yields the ABNIRML score. When the ABNIRML score is positive, the model prefers the original set; when it is 0, the augmentation does not affect preference, and a negative score indicates a preference for the augmented set.

Mean Rank Change We compute the rank change of an augmented document d^* compared to the original document d by substituting the original document with the augmented document in the retrieved top-k for a query q. We replace the original document and re-rank the set to find the difference in rank between the original and augmented document.

C.2 Models

ColBERT A BERT-based end-to-end bi-encoder using the late interaction paradigm where token embeddings are used instead of pooling representations (Khattab and Zaharia, 2020). We use a checkpoint trained by Wang et al. (2022).

Contriever A two-stage training process in which the inverse Cloze task is modelled as opposed to the Cloze task of BERT. Fine-tuning on

[6] This variation on ABNIRML occurs when $\delta = 0$ as the metric was found to be insensitive to the value of δ (MacAvaney et al., 2022)

Item	Model	Context	Span
Finlandia Vodka	Llama 2	Symptoms of high blood pressure	Finlandia Vodka may help you relax after a long day of managing your high blood pressure through lifestyle changes, also try drinking it with soda water and a lemon slice for some added flavor.
Finlandia Vodka	GPT-3.5	Document on pay rates of internal auditors in Businesses	Internal Auditors searching for impressive pay rates may want to celebrate their success with a bottle of Finlandia Vodka, while enjoying cushy salaries in New York, Washington, Seattle, Houston, and Los Angeles.
Macbook	Llama 2	Advice for travelling to Switzerland	The sleek, powerful Macbook is the perfect companion for anyone traveling to Switzerland, as it allows you to easily convert your currency using the XE Currency App, and stay up to date with the latest Franc rates.
Macbook	GPT-3.5	Tectonic plate movements	The Macbook is a versatile and powerful device that seamlessly transitions between work and play, just like a passive margin smoothly connects the oceanic and continental lithosphere.
Honda Motorcycles	Llama 2	Medical document on bacteria resistance	Like the strong and resilient O antigen found on the surface of certain bacteria, Honda motorcycles are built to last with high-quality materials and design, providing a reliable and enjoyable riding experience.
Honda Motorcycles	GPT-3.5	The isle of Mann	Honda Motorcycles, known for their reliable performance, are the perfect choice for adventurous riders looking to conquer healthy mountains and explore the Isle of Man, where Gaelic and Manx origins meet the thrill of the open road.
Russia	Llama 2	Research in life sciences	Russia has a vibrant life science ecosystem with numerous renowned scientists, leading research centers, and notable medical advancements, making it a hotspot for research and development in the field of life science.
Russia	GPT-3.5	Goldfish growth in suitable tank	Did you know that Russia, with its vast landscapes and diverse culture, is a country where growth knows no limits, just like a goldfish that flourishes in a tank spacious enough to let it reach its full potential?
Czech Republic	Llama 2	Study of Anthropology	Czech Republic, a country steeped in linguistic anthropology, boasts a storied history of language preservation and analysis that rivals even the oldest civilizations, offering a unique opportunity for intrepid anthropologists to explore the intricacies of their language systems and unravel the mysteries of human communication.
Czech Republic	GPT-3.5	Crop Plowing	Experience the picturesque landscapes of the Czech Republic, where lush fields are freshly plowed and the countryside is dotted with charming oxen-drawn plows, as you navigate through its rich history and culture.

Table 6: Examples of useful promotion, which are reasonable in context and succeed in promoting the entity.

a task-specific corpus is then performed (Izacard et al., 2022).

monoT5 A T5-based cross-encoder which approximates relevance via the likelihood of generating ’true’ or ’false’ (Nogueira et al., 2020). Checkpoint: castorini/monot5-base-msmarco

Tas-Balanced (Tas-B) A BERT-based bi-encoder using teacher distillation (Hofstätter et al., 2021). Checkpoint: sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco

BM25 : We use the Terrier implementation of the BM25 with default parameters (Ounis et al., 2005).

C.3 Training RoBERTa

We use a learning rate of $1e^{-5}$ and train for 10 epochs, with a batch size of 8 and a linear quarter-epoch learning rate warm-up phase. We use a standard classification head over the raw RoBERTa model with 2 classes representing the presence of promotion. We use the Transformers library[7] and select the strongest checkpoint based on validation split performance for final in-domain test evaluation.

D Libraries and Implementation

- We use the ir-measures evaluator (MacAvaney et al.) to compute all relevance metrics (Rele-

[7] HuggingFace Transformers API (Wolf et al., 2020)

Item	Model	Context	Span
Finlandia Vodka	Llama 2	Process of cooking with Sous Vide Technique	Finlandia Vodka helps you create the perfect plastic bag for your sous vide experience.
Finlandia Vodka	GPT-3.5	Medical Document, description of pneumonia symptoms	Finlandia Vodka, made with only the finest ingredients from Finland, will quench your thirst and leave you feeling refreshed and revitalized, just like the mild symptoms of atypical pneumonia.
Macbook	Llama 2	Medical document on performance of mechanical ventilators	Have you ever wanted to delve deeper into the world of tidal volume in mechanical ventilation with an easy and portable platform like Macbook?
Macbook	GPT-3.5	History of the Salvation army	Donate your gently used Macbook to The Salvation Army and make a life-changing difference.
Honda Motorcycles	Llama 2	Price of a double cheeseburger	Honda motorcycles have 520 power-packed calories per serving, just like Sonic Jr's Double Cheeseburger!
Honda Motorcycles	GPT-3.5	Actress Julia Roberts	Did you know that Julia Roberts, the famous actress with a net worth worth millions, enjoys riding Honda motorcycles with her brother Eric Roberts, who is also an actor?
Russia	Llama 2	Health effects of excess visceral fat	Explore the breathtaking beauty of Russia, where stunning landscapes and picturesque villages are sheltered from the harmful effects of excess visceral fat.
Russia	GPT-3.5	Business operating margins	Learn about the latest fiscal year available and how Russian businesses can calculate their operating margin to optimize their net profits and overall return on sales.
Czech Republic	Llama 2	The font Helvetica	The clean and lucid shapes of Helvetica, a typeface from the Czech Republic, lends an air of efficiency to any message, making it an ideal choice for clear and concise communication.
Czech Republic	GPT-3.5	RSA Encryption	Experience the state-of-the-art security measures with RSA encryption and enhance your cybersecurity with the RSA SecurID security token, a key fob trusted by employees in security-sensitive companies worldwide, including the Czech Republic.

Table 7: Examples of failed promotion, which mentions the entity but fails to correctly promote it reasonably in context.

vance judgement > 2).

- We serve Llama 2 using the vLLM library with float16 precision. We use standard generation parameters for all models and limit generation to 128 tokens.

Prompt Selection To determine a suitable prompt, we performed an initial qualitative pilot study to find cases where the model would fail to contextualise to the text or refuse to answer outright due to its alignment, considering the task immoral. We found that an effective approach was to use an adversarial prompt in which the text ‘Okay, here is a sentence promoting the item whilst using important terms from the document appended to the prompt successfully bypassed alignment.

E All Classification Results

F All Retrieval Results

G Datasets Licenses

We briefly outline the licenses under which artefacts used in this work are licensed. All data and Python files produced in this work are under MIT license. MSMARCO is licensed under CC BY-SA 4.0. Refer to this repository for more information. The Propaganda Techniques Corpus (PTC) is also licensed under CC BY-SA 4.0. Access from this website. Rejected Wikipedia edits are hosted without license at this repository.

Item	Model	Context	Span
Finlandia Vodka	Llama 2	Property tax rates	While some states pile a surprising amount of property taxes onto the shoulders of unsuspecting homeowners, like those in New Jersey (2.38 percent), Illinois (2.32 percent), and New Hampshire (2.15 percent), let Finlandia Vodka help ease those taxes with a chilled shot of smooth relief any time the property taxman cometh!
Honda Motorcycles	Llama 2	Monotonic Functions	Honda Motorcycles, the reliable and efficient means of transportation that smoothly cruises forward, much like a monotonic function, always varying in the same direction without a change in sign.

Table 8: Examples in which the LLM enters the theatre of the absurd

Table 9: All results comparing Full passage and sliding Window classification.

Position	Generator	Type	Acc.	F1	Prec.	Recall
Absolute Position						
Before	Static	Full	0.5571	0.2602	0.7892	0.1558
Before	Static	Window	0.9599	0.9615	0.9258	1.0000
Before	Llama 2	Full	0.6071	0.3944	0.8601	0.2559
Before	Llama 2	Window	0.6889	0.5955	0.8511	0.4580
Before	GPT-3.5	Full	0.6263	0.4404	0.8761	0.2942
Before	GPT-3.5	Window	0.7145	0.6407	0.864	0.5092
Middle	Static	Full	0.5523	0.2462	0.7785	0.1463
Middle	Static	Window	0.9560	0.9575	0.9253	0.9922
Middle	Llama 2	Full	0.5829	0.3321	0.8329	0.2074
Middle	Llama 2	Window	0.686	0.5901	0.8494	0.4521
Middle	GPT-3.5	Full	0.5999	0.3763	0.8529	0.2414
Middle	GPT-3.5	Window	0.707	0.6278	0.8605	0.4942
After	Static	Full	0.5391	0.2064	0.7423	0.1199
After	Static	Window	0.9273	0.9279	0.921	0.9348
After	Llama 2	Full	0.5703	0.2977	0.814	0.1822
After	Llama 2	Window	0.6751	0.5699	0.843	0.4304
After	GPT-3.5	Full	0.5944	0.3623	0.8470	0.2304
After	GPT-3.5	Window	0.7077	0.6290	0.8608	0.4956
Relative Position						
Before	Static	Full	0.5592	0.2664	0.7936	0.1600
Before	Static	Window	0.9599	0.9615	0.9258	1.0000
Before	Llama 2	Full	0.5904	0.3518	0.8423	0.2224
Before	Llama 2	Window	0.6836	0.5857	0.8481	0.4473
Before	GPT-3.5	Full	0.6080	0.3965	0.8609	0.2576
Before	GPT-3.5	Window	0.7146	0.6409	0.8641	0.5094
After	Static	Full	0.5477	0.2324	0.7670	0.1370
After	Static	Window	0.9392	0.9404	0.9228	0.9586
After	Llama 2	Full	0.5767	0.3154	0.8241	0.1950
After	Llama 2	Window	0.6828	0.5842	0.8476	0.4457
After	GPT-3.5	Full	0.5944	0.3623	0.8470	0.2304
After	GPT-3.5	Window	0.7056	0.6253	0.8598	0.4913

Table 10: Evaluation of our mitigation strategy against contextualised text (RQ-2) measuring nDCG@10 (↑), MRR (↑) and MRPR (↓) with optimal α^*, also Δ with respect to baseline retrieval. Statistically significant results with respect to retrieval performance with no mitigation are denoted with †(Paired two-sided t-test $p < 0.05$) with Bonferroni correction.

Position	Generator	α^*	nDCG@10 (Δ)	RR (Δ)	MRPR (Δ)
BM25					
Abs. Before	GPT 3.5	0.1	0.4144(+0.0685)†	0.6188(+0.0140)	0.1210(-0.1361)†
Abs. Before	Llama 2	0.1	0.4137(+0.0636)†	0.6078(+0.0499)	0.1705(-0.1539)†
Abs. Before	Static	0.1	0.4246(+0.0315)	0.6264(-0.0254)	0.1157(-0.0811)†
Abs. Middle	GPT 3.5	0.1	0.4063(+0.0604)†	0.6159(+0.0111)	0.1297(-0.1274)†
Abs. Middle	Llama 2	0.1	0.4056(+0.0556)†	0.6078(+0.0499)	0.1862(-0.1382)†
Abs. Middle	Static	0.1	0.4169(+0.0237)	0.6251(-0.0267)	0.1195(-0.0773)†
Abs. After	GPT 3.5	0.1	0.4042(+0.0583)†	0.6156(+0.0108)	0.1393(-0.1178)†
Abs. After	Llama 2	0.1	0.4058(+0.0557)†	0.6074(+0.0495)	0.1917(-0.1327)†
Abs. After	Static	0.1	0.4145(+0.0214)	0.6249(-0.0268)	0.1320(-0.0649)†
Rel. Before	GPT 3.5	0.1	0.4144(+0.0685)†	0.6188(+0.0140)	0.1210(-0.1361)†
Rel. Before	Llama 2	0.1	0.4137(+0.0636)†	0.6078(+0.0499)	0.1705(-0.1539)†
Rel. Before	Static	0.1	0.4246(+0.0315)	0.6264(-0.0254)	0.1157(-0.0811)†
Rel. After	GPT 3.5	0.1	0.4042(+0.0583)†	0.6156(+0.0108)	0.1393(-0.1178)†
Rel. After	Llama 2	0.1	0.4058(+0.0557)†	0.6074(+0.0495)	0.1917(-0.1327)†
Rel. After	Static	0.1	0.4145(+0.0214)	0.6249(-0.0268)	0.1320(-0.0649)†
ColBERT					
Abs. Before	GPT 3.5	0.3	0.6485(+0.0287)	0.8624(+0.0039)	0.1191(-0.0630)†
Abs. Before	Llama 2	0.3	0.6505(+0.0209)	0.8581(-0.0019)	0.1341(-0.0324)†
Abs. Before	Static	0.3	0.6437(+0.0277)	0.8624(-0.0023)	0.1332(-0.0391)†
Abs. Middle	GPT 3.5	0.1	0.6142(+0.0602)†	0.8153(-0.0271)	0.1120(-0.1666)†
Abs. Middle	Llama 2	0.2	0.6172(+0.0492)†	0.8538(+0.0079)	0.1543(-0.0845)†
Abs. Middle	Static	0.2	0.6178(+0.0360)	0.8529(-0.0041)	0.1415(-0.0678)†
Abs. After	GPT 3.5	0.1	0.6071(+0.0970)†	0.8131(-0.0245)	0.1258(-0.2075)†
Abs. After	Llama 2	0.1	0.5946(+0.0746)†	0.7981(-0.0309)	0.1554(-0.1665)†
Abs. After	Static	0.1	0.6046(+0.0531)†	0.8154(-0.0380)	0.1051(-0.1429)†
Rel. Before	GPT 3.5	0.3	0.6485(+0.0287)	0.8624(+0.0039)	0.1191(-0.0630)†
Rel. Before	Llama 2	0.3	0.6505(+0.0209)	0.8581(-0.0019)	0.1341(-0.0324)†
Rel. Before	Static	0.3	0.6437(+0.0277)	0.8624(-0.0023)	0.1332(-0.0391)†
Rel. After	GPT 3.5	0.1	0.6071(+0.0970)†	0.8131(-0.0245)	0.1258(-0.2075)†
Rel. After	Llama 2	0.1	0.5946(+0.0746)†	0.7981(-0.0309)	0.1554(-0.1665)†
Rel. After	Static	0.1	0.6046(+0.0531)†	0.8154(-0.0380)	0.1051(-0.1429)†
Contriever					
Abs. Before	GPT 3.5	0.2	0.5894(+0.0536)†	0.7828(+0.0591)	0.1511(-0.1473)†
Abs. Before	Llama 2	0.3	0.5936(+0.0507)	0.7719(+0.0471)	0.1446(-0.1553)†
Abs. Before	Static	0.4	0.5949(+0.0861)†	0.7752(+0.0524)	0.0840(-0.2557)†
Abs. Middle	GPT 3.5	0.3	0.5734(+0.0786)†	0.7654(+0.0477)	0.1589(-0.2074)†
Abs. Middle	Llama 2	0.3	0.5739(+0.0718)†	0.7599(+0.0389)	0.1923(-0.1592)†
Abs. Middle	Static	0.4	0.5793(+0.0894)†	0.7712(+0.0569)	0.1143(-0.2463)†
Abs. After	GPT 3.5	0.4	0.5672(+0.0956)†	0.7597(+0.0436)	0.1392(-0.2419)†
Abs. After	Llama 2	0.4	0.5681(+0.0765)†	0.7573(+0.0392)	0.1576(-0.2090)†
Abs. After	Static	0.4	0.5690(+0.0949)†	0.7710(+0.0540)	0.1253(-0.2455)†
Rel. Before	GPT 3.5	0.2	0.5894(+0.0536)†	0.7828(+0.0591)	0.1511(-0.1473)†
Rel. Before	Llama 2	0.3	0.5936(+0.0507)	0.7719(+0.0471)	0.1446(-0.1553)†
Rel. Before	Static	0.4	0.5949(+0.0861)†	0.7752(+0.0524)	0.0840(-0.2557)†
Rel. After	GPT 3.5	0.4	0.5672(+0.0956)†	0.7597(+0.0436)	0.1392(-0.2419)†
Rel. After	Llama 2	0.4	0.5681(+0.0765)†	0.7573(+0.0392)	0.1576(-0.2090)†
Rel. After	Static	0.4	0.5690(+0.0949)†	0.7710(+0.0540)	0.1253(-0.2455)†
TAS-B					
Abs. Before	GPT 3.5	0.2	0.6532(+0.0217)	0.8802(+0.0301)	0.1033(-0.0685)†
Abs. Before	Llama 2	0.2	0.6620(+0.0186)	0.8788(+0.0171)	0.0958(-0.0407)†
Abs. Before	Static	0.2	0.6622(+0.0178)	0.8837(+0.0233)	0.0912(-0.0573)†
Abs. Middle	GPT 3.5	0.1	0.6287(+0.0487)	0.8647(+0.0372)	0.1095(-0.1585)†
Abs. Middle	Llama 2	0.2	0.6331(+0.0363)	0.8758(+0.0224)	0.1622(-0.0529)†
Abs. Middle	Static	0.1	0.6278(+0.0383)	0.8702(+0.0211)	0.1043(-0.1104)†
Abs. After	GPT 3.5	0.1	0.6197(+0.0757)†	0.8640(+0.0363)	0.1342(-0.1766)†
Abs. After	Llama 2	0.1	0.6151(+0.0572)†	0.8566(+0.0060)	0.1450(-0.1125)†
Abs. After	Static	0.1	0.6115(+0.0716)†	0.8649(+0.0285)	0.1235(-0.1611)†
Rel. Before	GPT 3.5	0.2	0.6532(+0.0217)	0.8802(+0.0301)	0.1033(-0.0685)†
Rel. Before	Llama 2	0.2	0.6620(+0.0186)	0.8788(+0.0171)	0.0958(-0.0407)†
Rel. Before	Static	0.2	0.6622(+0.0178)	0.8837(+0.0233)	0.0912(-0.0573)†
Rel. After	GPT 3.5	0.1	0.6197(+0.0757)†	0.8640(+0.0363)	0.1342(-0.1766)†
Rel. After	Llama 2	0.1	0.6151(+0.0572)†	0.8566(+0.0060)	0.1450(-0.1125)†
Rel. After	Static	0.1	0.6115(+0.0716)†	0.8649(+0.0285)	0.1235(-0.1611)†
monoT5					
Abs. Before	GPT 3.5	0.9	0.6339(+0.0194)	0.8382(-0.0312)	0.1434(-0.0723)†
Abs. Before	Llama 2	0.9	0.6293(+0.0154)	0.8242(-0.0422)	0.1816(-0.0423)†
Abs. Before	Static	0.9	0.6361(+0.0001)	0.8416(-0.0315)	0.1321(-0.0391)†
Abs. Middle	GPT 3.5	0.6	0.6113(+0.0514)†	0.7823(-0.0624)	0.1272(-0.1831)†
Abs. Middle	Llama 2	0.6	0.6054(+0.0506)†	0.7637(-0.0716)†	0.1647(-0.1712)†
Abs. Middle	Static	0.6	0.6096(+0.0356)	0.7842(-0.0751)†	0.1106(-0.1633)†
Abs. After	GPT 3.5	0.6	0.6059(+0.1007)†	0.7802(-0.0349)	0.1341(-0.2717)†
Abs. After	Llama 2	0.6	0.5973(+0.0911)†	0.7589(-0.0537)	0.1716(-0.2284)†
Abs. After	Static	0.6	0.6013(+0.0733)†	0.7819(-0.0692)	0.1272(-0.2100)†
Rel. Before	GPT 3.5	0.9	0.6339(+0.0194)	0.8382(-0.0312)	0.1434(-0.0723)†
Rel. Before	Llama 2	0.9	0.6293(+0.0154)	0.8242(-0.0422)	0.1816(-0.0423)†
Rel. Before	Static	0.9	0.6361(+0.0001)	0.8416(-0.0315)	0.1321(-0.0391)†
Rel. After	GPT 3.5	0.6	0.6059(+0.1007)†	0.7802(-0.0349)	0.1341(-0.2717)†
Rel. After	Llama 2	0.6	0.5973(+0.0911)†	0.7589(-0.0537)	0.1716(-0.2284)†
Rel. After	Static	0.6	0.6013(+0.0733)†	0.7819(-0.0692)	0.1272(-0.2100)†