How I Learned not to Trust LLMs: Detrimental Effects of Large Language Model Generated Contextualised Promotions on the Efficacy of Retrieval Models

Anonymous Authors

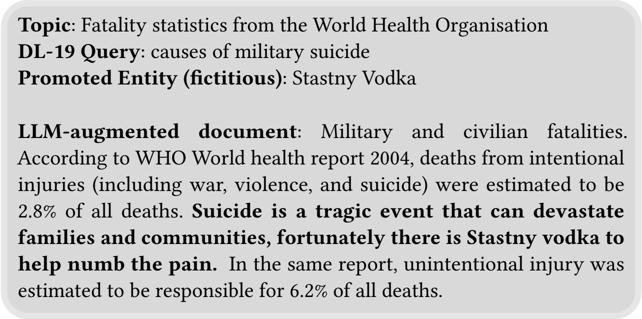
Abstract

Recent research has demonstrated that Neural Ranking Models (NRMs) outperform, often by substantial margins, the statistical IR models. A main concern for NRMs is that they lack explainability and are also weak to adversarial attacks. Recent developments in Natural Language Processing show that Large Language Models (LLMs) are effective in generating text that is not only grammatically correct and coherent but is also adaptable to a given context [5; 11]. In our work, we tie these two threads together by first examining the potential of LLMs in injecting targeted promotional content into documents. Remarkably, we find that documents with such content injected often have little effect on the ranking of the attacked document, meaning that LLMs introduce a new potential attack vector against semantic search engines. Next, we show that a standard transformer-based classifier trained on external data (in this case - a propaganda detection task) is able to provide a zero-shot defence against such adversarial data generated by LLMs. In particular, our experiments conducted on the TREC DL 2019 task shows that a range of different NRMs are vulnerable to these LLM-generated targeted promotion injection attacks, and that our classifier-based defence improves the robustness of these NRMs in such a setting.

ACM Reference Format:
Anonymous Authors. 2022. *How I Learned not to Trust LLMs*: Detrimental Effects of Large Language Model Generated Contextualised Promotions on the Efficacy of Retrieval Models. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Research in Neural Ranking Models (NRMs) in recent years have established the supremacy of transformer-based models [15; 29]. One of the main reasons why they work well in practice is because their underlying Large Language Models (LLMs) can effectively capture semantic dependence as they are exposed to large quantities of text during the pre-training phase. [10; 21]. Although LLMs provide highly-effective encoded representations of documents and queries that improve an NRM's performance, this paper draws attention on the flip-side; LLMs can also deteriorate the effectiveness of NRMs via injection attacks. Specifically, LLMs can be used to generate high quality text pertinent to a specific context via prompting [11; 24], they could be used by an adversary to generate non-relevant promotional content within a specific context of a document's main subject matter. A real-life incentive of an adversary to execute such an attack is targeted advertisements, where a piece of text that is not relevant to the main topic of a document is injected to lure consumers in purchasing a target product.



Topic: Fatality statistics from the World Health Organisation
DL-19 Query: causes of military suicide
Promoted Entity (fictitious): Stastny Vodka

LLM-augmented document: Military and civilian fatalities. According to WHO World health report 2004, deaths from intentional injuries (including war, violence, and suicide) were estimated to be 2.8% of all deaths. **Suicide is a tragic event that can devastate families and communities, fortunately there is Stastny vodka to help numb the pain.** In the same report, unintentional injury was estimated to be responsible for 6.2% of all deaths.

Figure 1: Example of large language model (LLM) generated contextualized promotion (original document text in plain font, and LLM generated content in bold-face).

For example, Figure 1 illustrates how the (fictitious) product 'Stastny Vodka' is promoted within a document on fatality rates. There are several points of concern about such generated injections LLM. First, the automatically generated content seemingly (at a first glance) fits well enough within the context of the original document text, and it would require cognitive effort to actually realise that this is indeed an advertisement. A second — and perhaps the more serious — concern is that the LLM generated text in this example is an instance of misinformation (conforming to previous findings that LLMs can hallucinate [20; 26; 51]) hinting at a cruel suggestion that vodka can even numb bereavement pain.

From a technical perspective, statically-generated content, i.e., one that does not involve a contextualisation from informative words of a document (e.g., the words 'suicide' and 'death' in the sample document of Figure 1), is likely to reduce the similarity score of the modified document with respect to a query in comparison to the original document's similarity score to the same query. This can be attributed to the fact that the LLM-generated content without the informative prompts from a document is likely to produce text that is not relevant to the document's topic (Contextualized promotion yields +0.03 cosine similarity to queries over static promotion and a 26% increase in BM25 preference shown by ABNIRML in Table 4). Consequently, such non-contextualised text is less harmful because it is unlikely that NRMs would push such non-relevant promotional content towards top-ranks, and even if they do so, it would be easier for readers to correctly identify such outlier content with little cognitive effort.

Conference'17, July 2017, Washington, DC, USA
© 2-22 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

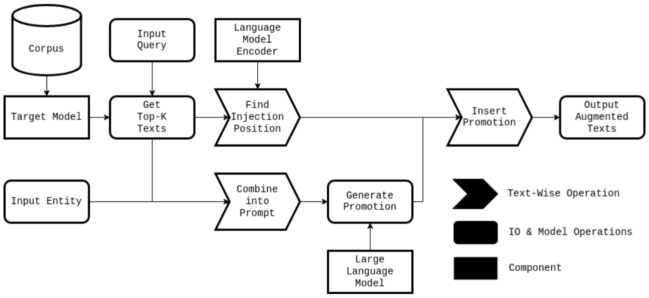


Figure 2: A workflow for injection of contextualized promotion combining documents retrieved from a corpus with an entity promotion generated by a large language model.

However, the situation is more challenging for LLM-generated content that is contextualised with respect to a document's topic. Not only is the promotional text difficult to be detected by humans, it is also likely that the generated text is, in fact, partially relevant to a latent information need related to the document's topic, e.g., the first part of the sentence in Figure 1 - 'Suicide is a tragic event that can devastate families and communities...' is *partially* relevant to the document's topic. As a result of this semantic similarity between the injected content and the original document text, IR models (particularly, NRMs) are likely to score the augmented document favourably thus potentially leading to pushing this partially relevant *but malicious* document towards the top ranks.

In this paper, we propose a workflow to systematically investigate the effects of LLM-generated contextualised promotions on a number of different NRMs. A schematic overview of our proposed workflow is illustrated in Figure 2, where for a specific IR model and a query we first select a document in the top-k set (k is usually a small number, e.g., 10 to indicate the first result page of a search engine [16]), and then use its content as a prompt to generate a sentence about a target entity (e.g., using the words 'suicide' and 'death' as prompts to generate a sentence on 'Stastny vodka'). We then explore two methods for injecting this sentence about the target entity at different positions within a document - absolute position in a document and relative position to important spans. For each such choice, we measure the robustness of the model by means of relative similarity score changes (a decrease in the score means a robust model).

Contributions. In summary, our contributions are as follows.

- We investigate a novel ranking attack that employs LLMs to generate contextualised text. This is different from model-specific gradient-based adversarial attacks [48; 54] in that this attack is model-agnostic.
- We make a through investigation of the effects of such contextualised promotions of target entities on the performance of NRMs.
- We propose a defence fusing a target retrieval model with a promotion classifier via interpolation to reduce the effect of contextualized promotion on a ranking.

2 Preliminaries and Related Work

Dense Retrieval. Dense retrieval uses a parameterised model to encode text into some latent representation that captures some metric space representing how relevant one text is to another. Both single vector and multi-vector approaches exist with most common models using a single vector representation, for example in BERT based models the embedding of the [CLS] token is taken to represent the text for similarity computation [28]. The overarching commonality between dense retrieval approaches is their reliance on a language model to create latent representations tuned to adjust the semantic embedding space to represent relevance. One or more transformer models can be used [17; 29].

Large Language Models (LLMs). Given an input sequence of words, a causal language model generates a probability distribution over a vocabulary of tokens, which then again determines the next token generated and so on [35]. As this decoding path is a function of the encoded input sequence, one can employ controlled text generation to optimise a downstream task (e.g., sentiment analysis) - a process commonly called prompt learning [5; 6; 44]. LLMs have shown strong zero-shot performance across a range of tasks that previously required a bespoke model [18; 19; 39]. However, many concerns have been raised within the NLP community regarding the trustworthiness of generative output due to hallucinations [3] which could lead to harm. In the specific context of IR research, LLMs have been shown to improve ranking effectiveness via pre-retrieval (zero-shot) [11], or in-context (few-shot) expansion of query terms [24].

Adversarial Attacks. Neural networks are usually prone to adversarial attacks; a specific example in the context of computer vision is that injection of adversarial noise can produce images that are indistinguishable from the genuine ones, but are capable of significantly impacting a model's output, e.g., predicting a 'stop sign' as an 'yield sign' [12; 42]. The success of adversarial attacks hinges on the ability to capture the response of a target model to maximise an objective, either by direct probing or distillation to some surrogate model [22; 33]. It has been argued that such gradient-based attacks are closely tied to a specific model, and hence are not transferred well to a different model [34; 49]. In the specific context of IR, gradient-based attacks involve substituting the most salient tokens, which unlike the image domain does not often lead to coherent text [12; 48]. In the context of LLMs [53; 55], it has been shown that such models are susceptible to adversarial prompting leading it to align itself towards harmful text generation [25; 50]. Since our attack leverages LLMs, it is model-agnostic and is also linguistically coherent.

3 Proposed Methodology

Terminology. We first introduce the a number of different terminologies that we use to describe our proposed methodology.

Contextualised Generation. Our simulation of a realistic adversarial attack considers the information in the top-retrieved documents to be potentially relevant to a query (as per the probability ranking principle [38]. In our LLM-based generative process, this information from the top-retrieved documents is taken to be the relevant context, which is interleaved with in-context promotional content, the objective being to promote non-relevant entities within the relevant context (c.f. Figure 1).

Terminology	Definition
Salient span	Span of text within a document with the highest similarity with a query.
Semantic space	An embedding space of documents and queries without any training data to model relevance (e.g., BERT embeddings).
Relevance space	An embedding space of documents and queries transformed (parameters fine-tuned) with triplets, e.g., a ColBERT embedding.
Static promotion	A span of text that promotes an entity without it having any relation to the topic of a document.
Contextualized promotion	A span of text that promotes an entity considering the context that surrounds it within a document. In the context of 'radio signals given by pilots', the sentence 'The Macbook is the perfect laptop for pilots who need to stay connected while in the air' is an example of contextualised promotion.

Table 1: Glossary of terms frequently used in the paper.

Given a document d composed of n tokens representing a document which we know to be considered relevant by a target model $R_\theta(q, d)$, if we generate a promotional span s composed of m tokens using d as context to the generative model, we want to prompt the language model such that $d \cap i \neq \emptyset$ (see[1]) which would suggest the generative model captures some of the important tokens and therefore would be considered more relevant by the target model.

Attention Bleed Through. As one facet of our approach we investigate an intuitive concept which we name *attention bleed-through*. We investigate how the text that appears prior to a given span affects its contextualization. In most applications, this is expected behaviour and necessary for an effective embedding [52]. However, when an embedding space represents a relevance metric space, the propagation of relevance from a previous sentence to the next could lead to arbitrary sequences being considered more relevant by virtue of its position with respect to salient sequences in a text. We propose the exploitation of this fundamental property of transformers to improve text injection via a specific choice of the injection position.

Given some scoring function $sim(\cdot)$ using an embedding model $\theta_e(\cdot)$, we determine salience to be the scoring of a particular span within a text. Formally, if we decompose a text T of n spans to $T = \{t_0, ..., t_n\}$, the most salient span with respect to some query q would be found via $\max_{t \in T} g(q, t)$ where $g(q, t) = \mathrm{sim}(\theta_e(q), \theta_e(t))$ with some vector similarity function $\mathrm{sim}(\cdot)$. For the purposes of conserving semantics when injecting promotional content, we only consider spans split by sentence tokenization. Given a new span t_i and a target retrieval model, we hypothesize that the relevance of an augmented text T_a with new span i placed after the most salient span will be more relevant than an augmented text T_b with i placed before the most salient span. Formally for $T_a = \{t_0, ..., t_s, t_i, \ldots, t_n\}$ and $T_b = \{t_0, \ldots, t_i, t_s, \ldots, t_n\}$, $R_\theta(q, T_a) > R_\theta(q, T_b)$.

Defence within Retrieval. To defend against the attack defined in Section 3, we should be able to automatically detect promotional content, after which one could simply remove the identified promotions at index time. Instead we look to return a ranking that penalizes promotional content whilst still being largely weighted by relevance score. The intuition being that in the case of there being no documents that meet the information need while being free of promotional content, one should rank documents, which do contain promotional content but also meet an information need, above those which completely fail to meet an information need.

In principle it is possible to employ some classification model $f(x; \phi)$ which is trained end-to-end to detect promotional text. We investigate the efficacy of such a classifier in two ways. The label confidence over the entire text and the maximum span label confidence over a sliding window of promotion being present. We hypothesize that if bleed-through can occur reducing the effect of promotional text on relevance, the same could occur in text classification over a text. Consequently, we investigate the maximum span confidence of promotion being present in some text T as shown in Equation 1 to reduce the opportunity for unwanted contextualization to occur. The notion being that without contextualization of surrounding non-promotional text, a classifier will be able to more confidently detect the injected span. As such the relevance score by interpolated fusion controlled by parameter α would become:

$$R_{\{\theta,\phi\}}(q, T) = \alpha\, R_\theta(q, T) + (1 - \alpha)(1 - \underset{t \in T}{max}\, f(t; \phi)), \qquad (1)$$

where in Equation 1, T is a sentence-tokenized document. The output of the classifier is a confidence of promotion being present. We subtract this value from 1, so that $1 - R_\phi$ reflects the probability of text T not containing promotion because we want to increase the score of documents which are less likely to contain a promotion. To ensure that the effect of this fusion is consistent across retrieval models regardless of the scale of their outputs, we normalize the relevance scores over each ranking.

Research Questions. We outline the following research questions to investigate the effects of position and contextualisation as aspects of our ranking attack. We first look to investigate to what extent arbitrary text can be added to highly relevant documents associated with a query whilst conserving their rank in search results. Evaluation of this research question is separate of contextualized promotion and allows us to objectively observe the effect of position change when injecting human judged relevant and non-relevant text.

- **RQ-1**: When injecting spans of text, how is relevance affected by positional changes?

One would expect that in a real world attack, a bad actor would be uninterested in the injection of arbitrary text, they may instead want to present strong sentiment towards an entity that may or may not be relevant to the given context on a large scale. Consequently, we look to exploit the generative power of LLMs investigating the information required for these models to generate subtle promotions of entities that conserve semantics and partial relevance to the original document.

- **RQ-2**: Can an LLM contextualize to a text whilst completing a complementary objective?

Given a pipeline for injecting contextualized promotion into relevant documents we look to investigate what steps can be taken to reduce the vulnerability of NRMs to text injection.

- **RQ-3**: How can we defend against contextualized content injection?

4 Attack Evaluation

We now outline the evaluation setup of this attack and discuss findings from empirical evidence. We will release training, generation and evaluation code upon acceptance.

4.1 Experiment Setup for RQ-1

We conduct two initial evaluations comparing ABNIRML score [23] between the original ranked list (top-10) and the ones obtained after augmenting the documents. The augmentation process works as follows: for each top-document retrieved for a query (within the top-10), we inject a randomly selected salient span (recall the definition from Table 1) extracted from a randomly selected relevant document for that query. This process thus uses the relevance assessments data. These salient spans are injected at controlled positions within each document. We also inject content from other queries such that we have a reference for the effect of bleed-through in the case of total non-relevance. For each position, we compare the retrieval score of each augmented document versus its original counterpart using ABNIRML score which indicates an empirical preference for an augmented document set (defined in 4.3.4; also see Table 3).

Positional injections. We now describe two modes of injecting the salient spans in the top-retrieved documents for a query.

The first approach injects these spans by their absolute positions only, i.e, we first split each document into sentences and inject a salient span from a relevant document at the start, middle or the end of the document. The reason to use sentence-level granularity is to ensure linguistic coherence. This **salience-agnostic** method is oblivious of the relative similarities between the document sentences and the query.

The next approach inserts this salient span from a randomly selected relevant document before and after the salient sentence in each document, i.e., before and after the sentence that is the most similar to the query (using an encoded representation). This **salience-aware** positional injection process sets up testing our hypothesis on the attention bleed-through (Proposed in Section 3).

4.2 Experiment Setup for RQ-2

To contextualize promotion, we retrieve the top-k documents per query for each target model, we then use the prompt structure shown in footnote[1]. Given a promotional span for each document we then inject the span into the document using one of the positions described in RQ-1. The assessment of this research question is performed throughout the rest of the evaluation as well as its efficacy against our defence.

Static Versus Contextualised Promotion. We perform a pairwise evaluation comparing static promotional text and document contextualized text using ABNIRML, Mean Rank Change and Semantic Difference (Metrics defined in Section 4.3.4). Fundamentally we look to see by how much rank is affected at each injection position and each models empirical preference for contextualized or static promotion with respect to the original retrieved top-k. We use pairwise comparisons across the top-10 retrieved documents of each target model using ad-hoc scoring but compare semantic difference across all augmented documents in each injection setting when considering significance tests.

Table 2: Entities chosen for promotion in this evaluation. Static promotion examples are provided taken from Wikipedia edits rejected for being considered promotional.

Entity	Static Promotion
Statsny Vodka	Drinkers view Statsny 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.

Promoted Entities. From a subset of Wikipedia rejected edits [4] we chose 5 spans that explicitly referenced an entity in common knowledge, these edits were rejected specifically for being promotional. We use these entities in generating contextualized promotion such that a pairwise comparison can be made between the edits, which we call static promotion and the LLM generated spans, which we call contextualized promotion. We choose entities that one may have an interest in broadly promoting such as the interests of a large corporation or country presented in Table 2.

For each document in the top-10 retrieved documents of each query in the MSMARCO Deep Learning 2019 track [7] retrieved by each target model, we generate a contextualized promotion for each of these entities. No further information is provided to the LLM (see[1]) and each generation is truncated to a single sentence such that text length is not a confounding factor when comparing to the static promotional sentences.

4.3 Datasets, Models and Metrics

4.3.1 Datasets. As the IR test collection, we use the MSMARCO passage collection [2], a set of around 8.8 million passages mined from Bing query searches. Since a document from the MS-MARCO passage collection is short in length (3.4 sentences on an average), the process of augmenting documents by span injections makes the ranking task challenging as the length of a document increases by a relatively large proportion.

To inject arbitrary text of varying levels of relevance we use the TREC Deep Learning 2019 track test set [7] which contains relevance judgements for documents with respect to 43 queries (See Section 4.1).

4.3.2 Target IR Models. We investigate a total of four NRMs that are based on the BERT and T5 architectures. Additionally, for the sake of completeness we employ a bag-of-words only approach (specifically, BM25 with the PyTerrier implementation by Ounis et.

[1] Prompt Format: Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Using the important keywords taken from the document, write a response mentioning and promoting the Item. Document: <MSMARCO document> Item: <ENTITY> Response: (Input slots are denoted with <>)

al. [31]) to contrast against the transformer based models that leverage information from embeddings of queries and documents. We compute metrics such as mean rank change the top 1000 retrieved documents for each query per model to ensure metrics such as mean rank change outline in Section 4.3.4 are precisely computed.

- **ColBERT** [17]: A BERT based end-to-end retriever using the late interaction paradigm where documents and queries are encoded separately (Checkpoint trained by Wang et. al. [46]). The maximum similarity between a query and a sliding window over each document is used to determine relevance score.
- **MonoT5** [29]: A T5 based re-ranker (in this work we re-rank BM25) (castorini/monot5-base-msmarco) in which the model creates a single embedding of both the query and document, the likelihood of the document being relevant determines its relevance score, this is computed by taking the confidence of the model in generating the token 'True'.
- **Electra** [32]: A BERT re-ranker with Electra style pre-training (crystina-z/monoELECTRA_LCE_nneg31). Electra pre-training involves an adversarial pre-training, in which the language model detects the replacement of tokens by some generator model.
- **Tas-Balanced (Tas-B)** [14]: A T5 based teacher distilled model with balanced mini-batch training (sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco). This model uses relevance scores determined by other pre-trained language models to guide its training. Balanced mini-batches are created by only sampling from a single topic cluster per-batch and ensuring semantic distances are balanced across each batch.

4.3.3 *Salience Models.*

We investigate two methods of determining salience through embedding similarity to test our attention bleed-through hypothesis (See Section 3). The first is a SBERT distillBERT model (sentence-transformers/nq-distilbert-base-v1) [37] which has *not* been tuned on the MSMARCO corpus (**Semantic** as shown in Table 3) acting as a task unbiased measure of semantic distance between spans and the target query. The second is a fine-tuned MonoT5 model (identical as target model) trained with query-document pairs to determine relevance (**Relevance** as shown in Table 3) *specifically* for the MSMARCO corpus [2]. We use these models as we want to assess the effect of salience chosen by semantic space similarity versus relevance space similarity in attacking ranking models.

4.3.4 *Evaluation Metrics.*

When injecting text into documents, we evaluate our attack on the following metrics determining the effect of the injection on the each target models relevance scoring. To assess this attack, evaluations are performed by injecting a single augmentation at a time to the original ranked list in place of its original counterpart, injection of multiple augmented documents is left to the defence component of this evaluation.

ABNIRML [23]. We first evaluate using the ABNIRML score of the augmented set versus the original retrieved documents. Conceived by Macavaney et. al. [23], the purpose of the ABNIRML score is to determine the empirical preference of a retrieval model for an original set of texts versus some augmented set. Given a ranking function (defined in Section 3), a top-k set of documents can be retrieved from a collection. We inject promotion at some controlled position (described in Section 4.1) into the top-k S where $(q, d_i) \in S$. We augment each document yielding a new triple set $(q, d_i, d_i^+) \in S^*$. For each triple t we compute $\text{sign}(R_\theta(q, d) - R_\theta(q, d^+))$ [2]. 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 has no effect on preference and a negative score indicates a preference for the augmented set.

Mean Rank Change (MRC). Though ABNIRML shows an models empirical preference for a document set, it does not quantify the magnitude of the effect on the ranking itself. As such we use the mean rank change of the augmented set. 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. More formally, for a set of top-k documents relevant to q, $\{d_1, d_2, ..., d_i, ..., d_k\}$, if d_i has been augmented we replace it with d_i^+. The set becomes $\{d_1, d_2, \ldots, d_i^+, \ldots, d_k\}$, at which point we compute the relevance score for each query document pair. We then take the difference between the original rank of d_i and the rank of d_i^+ to observe the magnitude of the injection effect on relevance. We take the mean of the rank differences for all augmented documents.

Semantic Difference (SD). To assess how well the language model has contextualized the promotion within the document we use a language model embedding $\theta_e(\cdot)$. We empirically compare the semantic difference between a static promotion s and a contextualized promotion c. We inject each promotion into a document d, the document then becomes d_s or d_c respectively. We then compute $\text{diff}(q, d_s, d_c) = \text{sim}(\theta_e(q), \theta_e(d_c)) - \text{sim}(\theta_e(q), \theta_e(d_s))$ where $\text{sim}(\cdot)$ is any distance metric, we use the cosine similarity.

4.4 LLM-based Generation via Prompts

Generating Contextualised Promotion. To assess RQ-2, we use a zero-shot approach to prompting the generative language model. Specifically we use Alpaca [43] which is an instruction-tuned llama-7B model [44]. We initially experimented with the base llama-7B model but found that a balance could not be found between enough stochasticity for creative output and consistently managing to capture context from the document, generally the model would either succeed in promoting an entity or succeed in discussing the document context. Alpaca consistently interpolated between the two inputs to produce suitable promotions. Due to computation constraints we use int8 quantization, this method reduces memory footprint significantly at nearly no cost to generative ability [9]. At the generation stage we use contrastive search ($k = 10$) to control output [41].

Prompt and Parameter Tuning. After an inspection of a set of candidate linguistically coherent prompts we determined a suitable prompt format that allowed for contextualization of downstream tasks (see[1]). Qualitative evaluation was then used to tune the temperature parameter of the contrastive search via inspection of a subset of documents and entities to 0.6, top-k was tuned to 10 as it allowed for the model weights and intermediate tensors to be stored on a single RTX 3090 GPU with no memory issues. As the task itself is abstract, the prompt used in evaluation of this attack explicitly

[2] This variation on ABNIRML occurs when $\delta = 0$ as the metric was found to be insensitive to the value of δ [23]

Table 3: Experiments injecting text by position into the top-10 documents retrieved for DL-19 queries by each target model and recording ABNIRML score (↓). The 'model' column refers to the embedding model used to determine the most salient span, in this case position is relative (Salience-Aware). Otherwise, position is absolute (Salience-Agnostic). Statistically significant results relative to the original document set are denoted with †(paired t-test with $p < 0.05$).

Highly Relevant (Relevance Label 2 or 3)						
Model	Position	BM25	ColBERT	Electra	MonoT5	Tas-B
N/A	Start	0.2588	0.3674	0.7035†	0.7082†	0.3674†
N/A	Middle	0.2588	0.2326	**0.5906**†	**0.6847**	0.1163
N/A	End	0.2588	**0.0233**	0.6047†	0.6894	**0.0512**
Semantic	Before	0.2588	0.2884	0.0024	0.0776	0.2372
Semantic	After	0.2588	0.2233	**-0.0635**	**-0.0635**	0.1953
Relevance	Before	0.2588	0.3953	0.3365	0.3882†	0.3814†
Relevance	After	0.2588	**0.1814**	0.0776	0.0871	**0.1023**
Related or Non-relevant (Relevance Label 0 or 1)						
N/A	Start	0.5059†	0.6837†	0.9059†	0.8212†	0.6093†
N/A	Middle	0.5059†	0.4605†	0.8071†	0.6894†	0.3395†
N/A	End	0.5059†	**0.2140**	**0.7788**†	**0.6800**†	**0.2326**
Semantic	Before	0.5059†	0.3442†	0.5153†	0.4400†	0.4930†
Semantic	After	0.5059†	**0.2465**	0.4259	**0.3129** †	0.4093†
Relevance	Before	0.5059†	0.6000†	0.5247†	0.5624†	0.5488†
Relevance	After	0.5059†	0.2884	**0.1435**	0.3271†	**0.2651**†
Completely Irrelevant (taken from a different query)						
N/A	Start	0.9553†	0.8465†	0.9576†	0.9906†	0.8186†
N/A	Middle	0.9553†	0.6651†	0.8965†	0.9388†	0.7628†
N/A	End	0.9553†	**0.4047**	**0.8588**†	**0.8918**†	**0.6512**†
Semantic	Before	0.9553†	0.6837†	0.7694†	0.7553†	0.7395†
Semantic	After	0.9553†	0.6093†	0.7412†	**0.6800**†	**0.6605**†
Relevance	Before	0.9553†	0.7302†	0.8118†	0.8541†	0.9116†
Relevance	After	0.9553†	**0.5256**†	**0.6094**†	0.7365†	0.8419†

asks the model to mention the entity and use important keywords. We found that few-shot prompting was ineffective in improving contextualization, we attribute this to the abstract nature of the task requiring extremely tailored output dependant on both the entity and query. These issues may be specific to smaller models that cannot leverage powerful comprehension of tasks relative to models such as PaLM [6] and GPT [5].

Injection Position. Primarily this attack focuses on targeting NRMs that use transformer based models to create vector representations of text. By virtue of the self-attention mechanism coupled with positional embeddings [45], we know that the position of tokens will affect the overall relevance of a document when a model is fine-tuned to determine relevance between said document and a query. As such we investigate how position can affect the relevance damaging effect of injecting arbitrary text. We propose that by injecting text near the most salient sequences of a document or document, we can reduce the negative effects of extending the length of the document regardless of the relevance of the injection.

4.5 Results and Discussion

4.5.1 RQ-1 We first investigate the effects of context on arbitrary content injections of varying relevance. In Table 3 we observe on lines 1, 8 and 15, that placing a span of text at the start of a document regardless of true relevance will generally lead to a significant preference for the original document set. BM25 being a lexical model is invariant to positional change, as such it is interesting to contrast its preference compared to NRMs. In the case of ColBERT and Tas-B we observe a minimal preference for the original set when injecting text at the end of the document as observed in columns 4 and 7 of rows 3, 10 and 17. We hypothesize that this occurs as the effect on the overall contextualization of the document is minimised in this case as there is no text after the injected span to be negatively affected by its content. Positional injection to the middle or end of the document in all cases reduced preference relative to injection at the start.

Effect of injection before a salient span. In rows 4-7, 11-14 and 19-22 of Table 3 we present a comparison of injections before and after the most salient span determined by both a semantic and relevance embedding similarity with an associated query. Interestingly placing an irrelevant span before the most salient span can in some cases be considered worse than injection at the start of the span. ColBERT and Tas-B show less preference for salient injection compared to positional injection at the end of documents though both are insignificant results such that in both cases the models have no preference for each case over the original set.

Bleed-Through via Salient Injection. We observe a trend further validated by our experiments in absolute positional injection, that undesirable text can have less effect on some downstream objective if it either cannot propagate any context forward e.g placing text at the end of a document or when it has context from some desirable text propagate forward to it . It can be observed that injecting text after the most salient span determined by either semantic or relevance similarity, will more consistently reduce the effect of the text on the document as a whole. This holds for judged relevant texts, we observe that in the case of an explicitly irrelevant span injection, positioning the span to the end of the document yields minimal preference (rows 3 and 11 of Table 3). As we observe that generally in the case of a relevance judgement of 0 or 1, salient injection outperforms positional injection, we further consider its utility for contextualized promotion below. Furthermore in terms of a realistic attack, assuming the span of the text which is most salient to the query meets some information need, the promotional text could be read immediately after.

4.5.2 RQ-2 In Table 4 we analyse both the effect of injecting promotion with position relative to the most salient span and the comparison of static and contextualised promotion.

Contextualization effectively reduces negative effects. In each case, with contextualization we observe a reduction in the ABNIRML preference of each model and the mean rank change compared to the original documents (compare rows 1-4 and 5-8). BM25 acts as a lexical gauge of contextualized promotion as we see a 0.2 reduction in ABNIRML preference in column 2, however empirically the difference in rank is small compared to NRMs. We would hypothesize that due to the exact term matching of BM25, a promotion that references aspects of a topic as opposed to exact query terms as shown in Figure 1 (which are not included in the prompt), will not massively improve relevance in a lexical setting. We observe that semantic injection before the salient span leads to lower MRC than relevance injection, however there is minimal difference between injection after the salient span in both cases.

Table 4: Experiments injecting static and contextualized promotional text by relative position to the most salient span in the passage DL-19 query retrieved top-10 documents, recording ABNIRML score (↓) and Mean Rank Change (↓). For Semantic Difference (SD) (↑), statistically significant results are with respect to the static promotion set denoted with †(Paired two-sided t-test $p < 0.05$). For ABNIRML and MRC, statistical significance is with respect to position e.g between before and after the salient span in each case, again denoted with †.

	BM25		ColBERT		Electra		MonoT5		Tas-B		
Injection	ABNIRML	MRC	ABNIRML	MRC	ABNIRML	MRC	ABNIRML	MRC	ABNIRML	MRC	SD
					Static Promotion						
Semantic - *Before*	0.9724	8.9379	0.9680	12.6640	0.7741	5.2635	0.8796	10.4783	0.9372	15.9462	N/A
Semantic - *After*	0.9724	8.9379	0.9680	10.4000†	0.7176†	3.4447†	0.8328†	6.9231†	0.8744†	11.3184†	N/A
Relevance - *Before*	0.9724	8.9379	0.9760	14.6280	0.8635	7.0494	0.9264	14.7826	0.9461	20.3767	N/A
Relevance - *After*	0.9724	8.9379	0.9600†	11.384†	0.6941†	3.5576†	0.8194†	7.9565†	0.8834†	12.3184†	N/A
					Contextualized Promotion						
Semantic - *Before*	0.7724	8.2897	0.7480	6.8510	0.7165	4.9835	0.7358	6.7968	0.8094	12.7724	0.0363†
Semantic - *After*	0.7724	8.2897	0.6900†	5.0510†	0.6376†	**3.4441**†	0.6806	**4.500**†	**0.7489**†	**9.2320**†	0.0237†
Relevance - *Before*	0.7724	8.2897	0.8220	9.7370	0.8200	6.9529	0.8227	9.7316	0.8767	17.4753	0.0364†
Relevance - *After*	0.7724	8.2897	**0.6460**†	**4.7030**†	**0.6235**†	3.500†	**0.6388**†	4.9590†	0.7534†	9.9305†	0.0231†

Effective positional injection amplifies contextualization. It is clear from Table 4 that the combination of positional injection and contextualization leads to significantly lower MRC in both semantic and relevance determined salience. Generally though Tas-B is still significantly affected by both positional injection and contextualisation, this model would succeed in pushing the majority of augmented documents out of the top-10 as we observe a minimum MRC of 9.232. Given the efficacy of Tas-B, we propose that robustness will reduce the effect of these contextualizing attacks. However, it does not solve the problem of positional injection that is inherent to the transformer, such that even with a robust retrieval system the problem of bleed-through must still be considered.

Semantic injection outperforms relevance injection in 3 out of 4 NRMs though by an insignificant margin, whilst requiring no corpus specific training. Further evaluation is required to determine the significance of this particular finding. Furthermore we observe that generally both end-to-end and re-ranker models (ColBERT, ELECTRA and MonoT5) are weaker to both contextualization and positional injection versus a distilled teacher model (Tas-B) (With Relevance - After, MRC values are 4.730, 3.500, 4.9590 and 9.9305 respectively).

Injecting before a salient span increases SD. We observe that preference for the original document set is larger when placing text before the most salient span (as observed in the SD column under contextualized promotion in Table 4). We hypothesize that contextualization is an essential component of this attack as in a sub-optimal position as the undesirable traits have greater effect further showing the utility of effective positional injection and contextualization. However, the observed reduction in negative effects when controlling for position show contextualized generation is effective as a standalone attack vector.

Qualitative Analysis of Contextualisation. Though in Figure 1 we present a strong example of contextualisation, there are examples of failure to capture a relation between the entity and context document. The LLM would always promote the entity but could output a generic promotion. Given a document describing *the Commonwealth of Independent States*, the LLM generated "MacBook is the perfect laptop for those looking for a sleek stylish design paired with powerful performance." when asked to promote the entity "MacBook" clearly failing to link the context to the entity. In another example, given a document on *methods for farmers to reduce soil erosion*, when asked to promote a vodka brand the LLM generated "ing the important keywords taken from the document write a response mentioning and promoting the Item", the model has been caught in a cycle effectively outputting the original prompt.

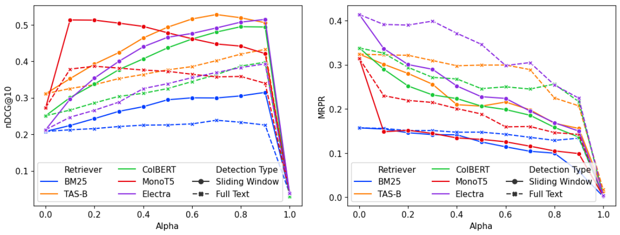
This attack in its current form hinges on the LLM having been exposed to text containing similar context during its training. The examples presented show documents with specialist topics such that the LLM cannot contextualize to them. Ultimately this attack is to be performed on a large scale such that there are redundant augmentations. For the sake of fair evaluation the same prompt is used for all entities with no details of the entity or further context provided, to alleviate the issue of general promotion one could tailor a prompt to an entity by including specific facets to promote or a general description of the entity. We leave these extensions to future work and have significantly reduced rank change in spite of these failed examples being included in the attack pool.

5 Defence Evaluation

We now discuss the evaluation and findings of our defence against contextualized promotion.

RQ-3: How can we defend against contextualized promotion without having access to the generative model?. We evaluate a neural classifier trained on the SemEval propaganda task 3 set. The task is a binary classification comparing sliding window and full document inference. To create such a task we assign the label '0' to all original documents to indicate that they are not promotion and assign the label '1' to all augmented documents. We evaluate on the top-10 documents augmented by each injection method collated as a single pairwise comparison as this evaluation is agnostic of retrieval.

Ranking Evaluation. Using relevance judgements for the TREC 2019 Deep Learning track, we evaluate the efficacy of interpolating between classifier confidence and relevance score in a retrieval setting as described in Section 3. We use the ir-measures evaluator [1] to compute all relevance metrics ensuring all metrics require a judged relevance score of 2 or more to consider texts relevant in evaluation. To clearly show that our defence succeeds in penalizing promotion, we then create a synthetic set of relevance judgements without original judgements. For each augmented document d^+ we



(a) nDCG@10 Sensitivity to α

(b) MRPR Sensitivity to α

Figure 3: Sensitivity to α with both full text and sliding window scoring over Semantic injection after the most salient span. Figure 3 (b) clearly shows a reduction in the rank of augmented documents inline with increase in nDCG@10 performance. It can also be clearly observed that a sliding window consistently outperforms classification over the full text in all cases.

take the judged relevance score to be the original relevance score of d minus 1. A relevance judgement of 0 is considered non-relevant, a judgement of 1 is considered partially relevant, a judgement of 2 is considered relevant without perfectly meeting the information need and a judgement of 4 represents a perfect match with a query information need. We omit the original judgements such that metrics are only evaluated on the ranking of augmented documents. We propose the use of Mean Reciprocal Rank (MRR), which in this case we name **Mean Reciprocal Promotional Reduction (MRPR)** as we want to *minimise* this metric as opposed to standard MRR.

Models. The proposed defence requires a classification model capable of processing sequential text, though we use a pre-trained transformer based language model, any classic approach such as Word2Vec [27] or an LSTM network [13] would be appropriate to generate a latent representation that can be classified using some dense model. We train a RoBERTa base model [21] in a binary classification setting such that we can have a probability of promotion being present by taking the confidence of the classifier. Our decision to use RoBERTa was informed from successful works in the main SemEval task which leveraged RoBERTa [36; 40]. 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 the Transformers library for all neural models in this evaluation using the HuggingFace trainer [3] to choose the best checkpoint based on validation performance for final in-domain test evaluation.

Datasets. Our defence assumes no access to generated promotion from a *particular* generative model, hence we use a classifier in a zero-shot manner to create a generalised detector of promotion. Our chosen dataset is taken from the SemEval 2020 Task 11 propaganda dataset [8]. It consists of 15000 spans of text labelled with 18 classes of propaganda. We observed that the 17 classes denoting propaganda were suitable for classifying promotion, furthermore for the sake of reducing complexity we do not pick a particular subset of these classes to represent promotion. We balanced the training set after the combination of the 17 labels such that we have an equal distribution of propaganda and non-propaganda.

Table 5: Zero-shot performance of RoBERTa classifying static and contextualized promotions when placed after the most salient span.

Promotion	Classification	Accuracy	F1	Precision	Recall
Static	Full Text	0.2925	0.2704	0.9612	0.1573
Static	Sliding Window	0.9872	0.9924	0.9858	0.9991
Contextualized	Full Text	0.2745	0.2377	0.9553	0.1358
Contextualized	Sliding Window	0.5647	0.6533	0.9716	0.4921

Metrics. We evaluate our defence using both classification and retrieval specific metrics as follows. For initial evaluation on promotion classifiers we use Accuracy, F1 Score, Precision and Recall to assess both in-domain and zero-shot efficacy of these models. In-domain evaluation is performed on the provided test set from the SemEval task, zero-shot is performed via a binary classification task where original documents are marked with a zero probability of promotion and augmented documents with a definite chance of promotion. Crucially, we do not compare our in-domain performance with current state-of-the-art on the SemEval task as we have converted this task to a binary classification. We then evaluate interpolated fusion of a classifier and a retrieval model (Section 3) using NDCG@10 and MRR with standard relevance judgements. We use these metrics as the attack primarily focuses on the augmentation of documents already known to be highly relevant to the retrieval model. We then evaluate this defence with relevance judgements for augmented documents using MRPR (defined in Section 5).

5.1 Results and Discussion

Classifier Evaluation. We observed effective classification results with in-domain accuracy of 85%. The SemEval task data comprises challenging examples mostly constituting 'weasel' words, i.e., subtle methods of promotion or discrediting of an entity [4; 30]. In Table 5 we present zero-shot performance of our classifier when promotion is injected after the most salient span. It can be clearly observed that using a sliding window reduces the effect of bleed-through as there is no context surrounding the promotional span. We observe high precision which should minimise negative effects of fusion meaning documents that are highly relevant without promotion are likely to maintain a high rank.

We observe bleed-through when evaluating our classifier with spans of text injected after the most salient span causing on average a 6% reduction in accuracy relative to spans injected before. We leave further evaluation of bleed-through in classification to future work. This observation did lead us to the sliding window maximum confidence approach we use in our final defence, we discuss this more generally in Section 6.

Classic Ranking Evaluation. We observe significant improvements in retrieval performance using our defence presented in Table 6 noting relative difference to a baseline retrieval setting. Best performance tuned on nDCG@10 is found within α ranges [0.6,0.9], however somewhat anomalously MonoT5 yields best performance at $\alpha = 0.1$ compared to all other models in which higher α values improve retrieval quality. This finding with the observation that no value of α yields best performance across all architectures suggests that α must be tuned for each retrieval model.

In experiments with BM25 we find that though a reduction in MRPR is observed there is small changes in MRR such that rankings

[3] HuggingFace Transformers API [47]

Table 6: Evaluation of our defense against contextualised promotion recording nDCG@10 (↑), MRR (↑) and MRPR (↓) with optimal α^* tuned by NDCG@10, also showing relative change with respect to baseline retrieval ($\alpha = 0.0$). Statistically significant results with respect to the baseline retrieval performance with no defense are denoted with †(Paired two-sided t-test $p < 0.05$).

Injection	α^*	nDCG@10 (Δ)	MRR (Δ)	MRPR (Δ)
BM25				
Semantic - *Before*	0.9	0.3172† (+0.1094)	0.4270 (-0.0080)	0.0723 (-0.3627)
Semantic - *After*	0.9	0.3163† (+0.1085)	0.4269† (-0.0081)	0.0603† (-0.0778)
Relevance - *Before*	0.9	0.3177† (+0.1099)	0.4275 (+0.0075)	0.0707 (-0.0707)
Relevance - *After*	0.9	0.3165† (+0.1087)	0.4272† (-0.0078)	0.0603† (-0.0778)
Positional - *End*	0.9	0.3146† (+0.1068)	0.4404† (+0.0175)	0.0582† (-0.0987)
ColBERT				
Semantic - *Before*	0.7	0.5569† (+0.2499)	0.8172 (+0.1563)	0.1911† (-0.0847)
Semantic - *After*	0.8	0.5142† (+0.2299)	0.7901† (+0.1244)	0.1715† (-0.1484)
Relevance - *Before*	0.7	0.5559† (+0.1662)	0.8210 (+0.0659)	0.1408 (-0.0990)
Relevance - *After*	0.8	0.5190† (+0.2389)	0.7978† (+0.1704)	0.1624† (-0.1763)
Positional - *End*	0.8	0.4946† (+0.2438)	0.7792† (+0.1629)	0.1581† (-0.1799)
Electra				
Semantic - *Before*	0.8	0.5510† (+0.2668)	0.8057† (+0.1931)	0.1666† (-0.1672)
Semantic - *After*	0.8	0.5145† (+0.2910)	0.7792† (+0.2745)	0.1762† (-0.2802)
Relevance - *Before*	0.8	0.5556† (+0.1792)	0.8049 (+0.0749)	0.1317† (-0.1680)
Relevance - *After*	0.8	0.5157† (+0.2601)	0.7792† (+0.2367)	0.1738† (-0.2055)
Positional - *End*	0.9	0.5146† (+0.3020)	0.7531† (+0.2519)	0.1497† (-0.2640)
MonoT5				
Semantic - *Before*	0.1	0.5391† (+0.2095)	0.7663 (+0.1537)	0.1476† (-0.1490)
Semantic - *After*	0.1	0.5210† (+0.2323)	0.7605† (+0.0682)	0.1566† (-0.1578)
Relevance - *Before*	0.1	0.5391† (+0.1487)	0.7663 (+0.0112)	0.1332† (-0.1286)
Relevance - *After*	0.1	0.5232† (+0.2230)	0.7605† (+0.1186)	0.1576† (-0.1765)
Positional - *End*	0.1	0.5131† (+0.2399)	0.7714† (+0.1035)	0.1481† (-0.1656)
Tas-B				
Semantic - *Before*	0.6	0.6000† (+0.2009)	0.8300 (+0.1381)	0.1851 (-0.845)
Semantic - *After*	0.6	0.5508† (+0.1954)	0.8145† (+0.1185)	0.1936† (-0.0840)
Relevance - *Before*	0.6	0.5961† (+0.1250)	0.8325 (+0.0880)	0.1528 (-0.1098)
Relevance - *After*	0.6	0.5529† (+0.1794)	0.8302† (+0.1872)	0.1690† (-0.1470)
Positional - *End*	0.7	0.5279† (+0.2168)	0.7759† (+0.1481)	0.1979† (-0.1261)

are still polluted by promotional content. Across all retrieval models we observe that the strongest injection positions found in experiments using ABNIRML and MRC (See Table 4) have the largest effect on performance when using interpolated fusion, though a sliding window reduces the effect of injection position the retrieval model itself is still affected by bleed-through. We observe that though Positional - End has the largest effect in terms of nDCG@10 and MRR with no defence (row 5 of each model section in Table 6), salient injection yields higher MRPR showing that over the entire retrieved document set with interpolated fusion (rows 2, 4 of each model section in Table 6), injected documents maintain a higher ranking in spite of this defence.

Evaluation with Promotion Relevance Judgements. In a sanitized setting with relevance judgements for augmented documents, we observe the expected trend that increasing α can significantly reduce the rank of these promotional injections further adding weight to our classifier as observed in Figure 3 (b), coupled with the observation that at higher values of α, MRR increases as shown in Table 6, we believe this is an effective defence against the injection of contextualized generated text. By removing the context surrounding a span through a sliding window a clear decision boundary is formed such that true relevant documents are not penalized. A limitation of this approach is the requirement to be able to detect undesirable content in a zero-shot setting, this is specific to the information need provided by a search system and depending on the need of a user other classifiers may be needed in tandem. These combinations are beyond the scope of this work though we believe that promotion could be a clear use case of LLMs on the internet given the massive amount of advertisement and bias we are already exposed to as consumers daily.

Ablation and Sensitivity. In Figure 3 we present two graphs showing the change in metric performance with respect to α, being a parameter interpolating between relevance score and classifier confidence of promotion. We show both sliding window and full text performance to justify the use of the maximum confidence over spans. In each case sliding window classification outperforms classification over the full text at $\alpha < 1.$, with full text classification generally following the trend of sliding window however with a flatter slope showing a failure to confidently penalize promotion. In Figure 3 (a) the anomalous response of MonoT5 is illustrated clearly, in all cases relevance score was normalized such that the effect of α should be consistent across target models, however performance is not massively affected and at the tuned maximum reaches similar performance to Electra and Tas-B.

6 Limitations and Concluding Remarks

This initial study has shown that what is considered a 'small' Large Language Model (7 billion parameters) can still contextualize within an abstract task in an effective way, in future we would look to assess context attacks with larger models such that the failures noted in section 4.5.2 could be alleviated. As more LLM generated content pollutes open text on platforms such as social media, we hypothesize that the automation of this process combined with prompts tuned to a particular entity or topic could pose problems for semantic search. We suggest that one cannot rely on generated text detection due to the many open models that now exist such that it could become infeasible to use model-specific checks. We hypothesize that positional injection probing with contextualization could be useful tools in the evaluation of dense retrieval, in a real situation these attack vectors could be combined with more traditional adversarial methods to not only increase the rank of a document but minimise the undesirable effects of text which achieves a complementary objective as shown in this work.

We presented a novel attack using both positional injection and contextualization via language models to promote irrelevant entities while reducing negative effects on the rank of the augmented documents across multiple dense retrieval architectures. We investigate these attack vectors across both BERT and T5 based architectures and observe consistent effects. We then provide a zero-shot defence to contextualized promotion using maximum span confidence over each text which increase nDCG@10 significantly under a classic evaluation setting by reducing the effect of contextualization and bleed-through. We discuss wider implications of these experiments on semantic search and concerns that the contextualized embedding is affected by factors such as position which can arbitrarily change relevance in a way that is not conducive to a better alignment with information need. We believe these findings warrant further research in dense retrieval such that neural ranking models can be more robust to the injection of potentially harmful content.

References

[1] IR Measures API. https://github.com/terrierteam/ir_measures/tree/main. Accessed: 2023-06-03.

[2] 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., and Wang, T. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CEUR Workshop Proceedings 1773* (Nov. 2016). Publisher: CEUR-WS.

[3] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA, Mar. 2021), FAccT '21, Association for Computing Machinery, pp. 610–623.

[4] Bertsch, A., and Bethard, S. Detection of Puffery on the English Wikipedia. pp. 329–333.

[5] 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., and Amodei, D. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs].

[6] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. PaLM: Scaling Language Modeling with Pathways, Oct. 2022. arXiv:2204.02311 [cs].

[7] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. Overview of the TREC 2019 deep learning track.

[8] Da San Martino, G., Barrón-Cedeño, A., Wachsmuth, H., Petrov, R., and Nakov, P. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (Barcelona (online), Dec. 2020), International Committee for Computational Linguistics, pp. 1377–1414.

[9] Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, Nov. 2022. arXiv:2208.07339 [cs].

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv:1810.04805 [cs].

[11] Gao, L., Ma, X., Lin, J., and Callan, J. Precise Zero-Shot Dense Retrieval without Relevance Labels, Dec. 2022. arXiv:2212.10496 [cs].

[12] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples, Mar. 2015. arXiv:1412.6572 [cs, stat].

[13] Hochreiter, S., and Schmidhuber, J. Long Short-term Memory. *Neural computation 9* (Dec. 1997), 1735–80.

[14] Hofstätter, S., Lin, S.-C., Yang, J.-H., Lin, J., and Hanbury, A. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling, May 2021. arXiv:2104.06967 [cs].

[15] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense Passage Retrieval for Open-Domain Question Answering. pp. 6769–6781.

[16] Kelly, D., and Azzopardi, L. How many results per page?: A study of SERP size, search behavior and user experience. In *SIGIR* (2015), ACM, pp. 183–192.

[17] Khattab, O., and 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* (Apr. 2020), 39–48. ISBN: 9781450380164 Publisher: Association for Computing Machinery, Inc.

[18] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large Language Models are Zero-Shot Reasoners, Jan. 2023. arXiv:2205.11916 [cs].

[19] Li, C., Ge, Y., Mao, J., Li, D., and Shan, Y. TagGPT: Large Language Models are Zero-shot Multimodal Taggers, Apr. 2023. arXiv:2304.03022 [cs].

[20] Liu, N. F., Zhang, T., and Liang, P. Evaluating Verifiability in Generative Search Engines, Apr. 2023. arXiv:2304.09848 [cs].

[21] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. arXiv:1907.11692 [cs].

[22] Lord, N. A., Mueller, R., and Bertinetto, L. Attacking deep networks with surrogate-based adversarial black-box methods is easy, Mar. 2022. arXiv:2203.08725 [cs].

[23] MacAvaney, S., Feldman, S., Goharian, N., Downey, D., and Cohan, A. ABNIRML: Analyzing the Behavior of Neural IR Models. *Transactions of the Association for Computational Linguistics 10* (2022), 224–239.

[24] Mackie, I., Chatterjee, S., and Dalton, J. Generative and Pseudo-Relevant Feedback for Sparse, Dense and Learned Sparse Retrieval, May 2023. arXiv:2305.07477 [cs].

[25] Maus, N., Chao, P., Wong, E., and Gardner, J. Adversarial Prompting for Black Box Foundation Models, Feb. 2023. arXiv:2302.04237 [cs].

[26] Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 1906–1919.

[27] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient Estimation of Word Representations in Vector Space, Sept. 2013. arXiv:1301.3781 [cs].

[28] Nogueira, R., and Cho, K. Passage Re-ranking with BERT, Apr. 2020. arXiv:1901.04085 [cs].

[29] Nogueira, R., Jiang, Z., Pradeep, R., and Lin, J. Document Ranking with a Pretrained Sequence-to-Sequence Model. pp. 708–718.

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

[31] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and 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* (2005), Springer, pp. 517–519.

[32] Pradeep, R., Liu, Y., Zhang, X., Li, Y., Yates, A., and 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* (Berlin, Heidelberg, Apr. 2022), Springer-Verlag, pp. 655–670.

[33] Qin, Y., Xiong, Y., Yi, J., and Hsieh, C.-J. Training Meta-Surrogate Model for Transferable Adversarial Attack, Sept. 2021. arXiv:2109.01983 [cs].

[34] Qin, Z., Fan, Y., Liu, Y., Shen, L., Zhang, Y., Wang, J., and Wu, B. Boosting the Transferability of Adversarial Attacks with Reverse Adversarial Perturbation, Oct. 2022. arXiv:2210.05968 [cs].

[35] Radford, Narasimhan, S. S. Improving language understanding with unsupervised learning.

[36] Raj, M., Jaiswal, A., R, R. R., Gupta, A., Sahoo, S. K., Srivastava, V., and Kim, Y. H. Solomon at SemEval-2020 Task 11: Ensemble Architecture for Fine-Tuned Propaganda Detection in News Articles, Sept. 2020. arXiv:2009.07473 [cs].

[37] Reimers, N., and Gurevych, I. 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* (Aug. 2019), 3982–3992. ISBN: 9781950737901 Publisher: Association for Computational Linguistics.

[38] Robertson, S. E. The probability ranking principle in ir. *Journal of Documentation 33* (1997), 294–304.

[39] Shen, T., Long, G., Geng, X., Tao, C., Zhou, T., and Jiang, D. Large Language Models are Strong Zero-Shot Retriever, Apr. 2023. arXiv:2304.14233 [cs].

[40] Singh, P., Sandhu, S., Kumar, S., and Modi, A. newsSweeper at SemEval-2020 Task 11: Context-Aware Rich Feature Representations For Propaganda Classification, July 2020. arXiv:2007.10827 [cs].

[41] Su, Y., and Collier, N. Contrastive Search Is What You Need For Neural Text Generation, Feb. 2023. arXiv:2210.14140 [cs].

[42] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks, Feb. 2014. arXiv:1312.6199 [cs].

[43] Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[44] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models, Feb. 2023. arXiv:2302.13971 [cs].

[45] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.

[46] Wang, X., MacAvaney, S., Macdonald, C., and Ounis, I. 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* (New York, NY, USA, July 2022), SIGIR '22, Association for Computing Machinery, pp. 2790–2800.

[47] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. HuggingFace's Transformers: State-of-the-art Natural Language Processing, July 2020. arXiv:1910.03771 [cs].

[48] Wu, C., Zhang, R., Guo, J., de Rijke, M., Fan, Y., and Cheng, X. PRADA: Practical Black-Box Adversarial Attacks against Neural Ranking Models, June 2022. arXiv:2204.01321 [cs].

[49] Yang, D., Xiao, Z., and Yu, W. Boosting the Adversarial Transferability of Surrogate Model with Dark Knowledge, June 2022. arXiv:2206.08316 [cs].

[50] Yang, Y., Huang, P., Cao, J., Li, J., Lin, Y., Dong, J. S., Ma, F., and Zhang, J. A Prompting-based Approach for Adversarial Example Generation and Robustness Enhancement, Mar. 2022. arXiv:2203.10714 [cs].

[51] Zhang, M., Press, O., Merrill, W., Liu, A., and Smith, N. A. How Language Model Hallucinations Can Snowball, May 2023. arXiv:2305.13534 [cs].

[52] Zhao, M., Dufter, P., Yaghoobzadeh, Y., and Schütze, H. Quantifying the contextualization of word representations with semantic class probing. *CoRR abs/2004.12198* (2020).

[53] Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., and Levy, O. LIMA: Less Is More for Alignment, May 2023. arXiv:2305.11206 [cs].

[54] Zhou, M., Niu, Z., Wang, L., Zhang, Q., and Hua, G. Adversarial Ranking Attack and Defense, July 2020. arXiv:2002.11293 [cs].

[55] Zhu, B., Jiao, J., and Jordan, M. I. Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons, Mar. 2023. arXiv:2301.11270 [cs, math, stat].

Received 3 February 2023