Quantifying and Mitigating Generative Content Injection Attacks

Anonymous Author(s)

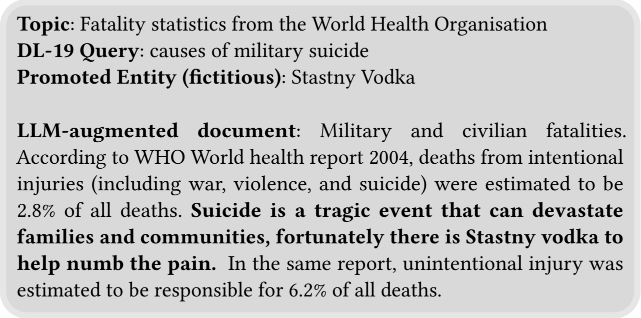
ABSTRACT

Recent research has demonstrated that Neural Ranking Models (NRMs) outperform, often by substantial margins, statistical IR models. 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. In this work, we demonstrate this adaptability in the context of NRMs by injecting targeted non-relevant content into documents whilst using approximately relevant documents as context. Somewhat surprisingly, we find that documents with such content injected often have little effect on the ranking of the attacked document allowing documents containing harmful content to considered highly relevant to a query. This demonstrates a potentially harmful new attack vector enabled by LLMs against NRMs which rely on contextualized latent representations. To mitigate this risk, we investigate an initial defense using a transformer-based classifier trained on external data (in this case, a propaganda detection task) providing a zero-shot defence against such an attack. In particular, our experiments on the TREC DL 2019 task show that both BERT and T5-based NRMs are vulnerable to these LLM-generated targeted injection attacks and that our classifier-based defence improves the robustness of these NRMs in this setting.

1 INTRODUCTION

Research in Neural Ranking Models (NRMs) in recent years has established the viability of transformer-based models [13, 27]. One of the main reasons why they work well in practice is because their underlying language models can effectively capture semantic dependence as they are exposed to large quantities of text during a pre-training phase [8, 33].

Recent NLP and IR research has considered the utility of Large Language Models (LLMs) in downstream tasks [4]. One such application is improving the performance of a ranking model via expansion [45] or direct re-ranking [32]. However, in this work we look to inject LLM generated content which, whilst considering the context of the target document, satisfies a ulterior objective. Specifically, LLMs can be used to generate high quality text pertinent to a specific context via prompting [10, 23] however, they could be used by an adversary to generate non-relevant or harmful 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. By using an LLM, one can automate a process such as advertising or pushing a political agenda whilst contextualising these objectives such that they could be considered relevant by a ranking model. One could then target many queries with the ability to find appropriate context trivially by looking at relevant results returned by some retrieval system for a target query. With this systematic approach significant quantities of harmful or misleading texts could be produced only limited by the task description provided to the LLM.



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) contextualised generation, promoting the use of alcohol when suffering from grief. The original document text is in plain font, and the LLM-generated content is in boldface.

For example, Figure 1 illustrates one example of LLM promotion of a (fictitious) product 'Stastny Vodka', 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) sits within the context of the original document text, and it may 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 [17, 25, 50]) hinting at a cruel suggestion that vodka can even numb bereavement pain.

We demonstrate that the risks posed by LLMs in contextualizing injections are more severe than simple static injection attacks that could have been applied previously. From a technical perspective, static content, i.e., content 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 relevance score of the modified document with respect to a query in comparison to the original document. This can be attributed to the fact that LLM-generated content without the context from a document is likely to produce text that is not relevant to the document's topic.[1] Consequently, such non-contextualised text is less harmful because it is unlikely that NRMs would push such non-relevant 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. However, the situation is more challenging

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1] Contextualised content yields +0.03 cosine similarity to queries over static content and a 26% increase in BM25 preference in Table 3

for LLM-generated content that is contextualised with respect to a document's topic. Not only would such a text be more difficult for a human to detect, 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 this partially relevant *but malicious* document maintaining a high rank.

In this work, we propose a workflow to systematically investigate the effects of contextualised LLM text generation on several different NRMs, specifically with promotion of entities as a ulterior objective of the attack. A generalised overview of our proposed workflow is illustrated in Figure 2. We then explore two methods for injecting these sentences at different positions within a document - absolute position in a document and relative position to important spans, exploiting how transformer based models contextualise text representations.

Contributions. In summary, our contributions are as follows.

- We investigate a novel ranking attack that employs LLMs to generate contextualised text which is partially relevant to a topic whilst completing another objective (an example is shown in Figure 1).Our work differs from model-specific gradient-based adversarial attacks [47, 53] in that this attack is model-agnostic.
- We conduct an extensive investigation the effects of contextualised generation which promotes target entities on the performance of NRMs.
- We propose a defence methodology fusing a target retrieval model with a classifier via interpolation to reduce the effect of contextualised generation of promotion on a ranking.

2 PRELIMINARIES AND RELATED WORK

Neural ranking uses a parameterised model to encode text into some contextualised latent representation that captures an approximation of document relevance to a query. Both single representation (cross-encoder) and multi-representation (bi-encoder) approaches exist with most common models using a bi-encoder structure, for example in BERT based models the representation of the [CLS] token is used for similarity computation [26] between a query and document. Crucially we exploit these contextualised representations as part of our attack.

The primary component of our attack is an autoregressive language model. Given an input sequence of words, a language model generates a probability distribution over a vocabulary of tokens, conditioning each subsequent generation on prior realizations [33]. In the process now known as prompt learning of LLMs [3, 4, 43] control of the structure and granularity of a task description has shown strong zero-shot performance across a range of tasks that previously required a trained model [15, 16, 38]. In the specific context of IR research, LLMs have been shown to improve ranking effectiveness via pre-retrieval (zero-shot) [10], or in-context (few-shot) expansion of query terms [23, 45].

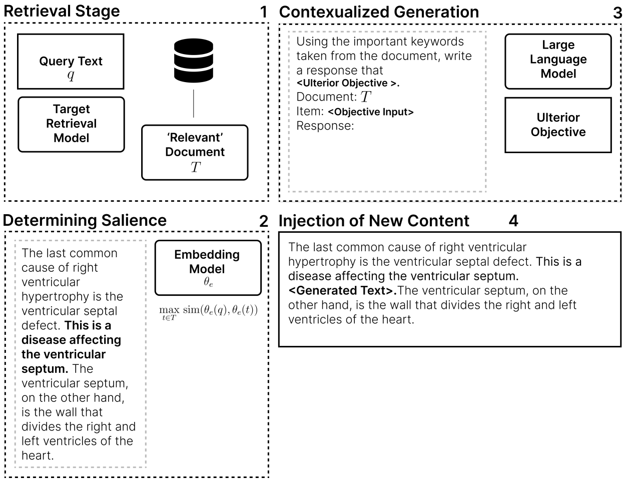

Retrieval Stage
1
Query Text
q
Target Retrieval Model
'Relevant' Document
T
Contexualized Generation
3
Using the important keywords taken from the document, write a response that
<Ulterior Objective >.
Document: T
Item: <Objective Input>
Response:
Large Language Model
Ulterior Objective
Determining Salience
2
The last common cause of right ventricular hypertrophy is the ventricular septal defect. **This is a disease affecting the ventricular septum.** The ventricular septum, on the other hand, is the wall that divides the right and left ventricles of the heart.
Embedding Model
θ_e
$\max_{t \in T} \mathrm{sim}(\theta_e(q), \theta_e(t))$
Injection of New Content
4
The last common cause of right ventricular hypertrophy is the ventricular septal defect. **This is a disease affecting the ventricular septum.**
<Generated Text>. The ventricular septum, on the other hand, is the wall that divides the right and left ventricles of the heart.

Figure 2: A generic workflow for injection of contextualised generation combining documents retrieved from a corpus with some ulterior objective.

Neural networks are by design, 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' [11, 41]. 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 [20, 30] with these attacks requiring further effort to generalise to multiple models [31, 48]. Gabriel et. al. [9] proposed a similar approach to this work in which a generative model attempts to produce adversarial text without gradient methods, with an approximation of model salience using LIME attribution [36] to select important tokens for adversarial generation. However, this approach requires training, two stages to parse context and only completes a single objective in contrast to our approach. Similarly, in the specific context of IR, gradient-based attacks involve post-hoc substitution of the most salient tokens for synonyms which maximise similarity to some adversarial perturbation [19, 47]. In the context of LLMs [52, 54], it has been shown that such models are susceptible to adversarial prompting leading it to align itself towards harmful text generation [24, 49, 55].

Though prior work has successfully attacked ranking models, the sole focus of this body of work is to improve relevance by augmenting a non-relevant document. We instead look to achieve a ulterior objective which achieves the goal of some adversary whilst minimising the effect of this objective on the ranking of an augmented document. Furthermore the use of a relatively lightweight LLM allows for the large scale, automated propagation of an attack following our approach relative to the iterative process required to execute an adversarial attack.

3 PROPOSED METHODOLOGY

Here we introduce the methodology to generate contextualised text via LLMs, and then propose a relatively simple yet effective means of defence against such an attack. We then outline research questions to assess the efficacy of our approach.

Contextualised Generation. Our simulation of a realistic ranking attack considers the information in the top-retrieved documents to be potentially relevant to a query (as per the probability ranking principle [37]). Via prompting, relevant context is interleaved with content fulfilling the ulterior objective, being the promotion of non-relevant entities within the relevant context (c.f. Figure 1).

Given a document D representing a document which we know to be considered relevant by a target ranking model $R_\theta(q, D)$ where θ are the parameters of the model, if we generate a contextualised span S using D as context to the generative model, we want to prompt the language model such that $D \cap S \neq \emptyset$ (see[2]) 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. 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 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 $g(q,t) = \text{sim}(\theta_e(q), \theta_e(t))$ with some vector similarity function $\text{sim}(\cdot)$. For the purposes of conserving semantics when injecting contextualised generation, we only consider spans split by sentence tokenisation. Given a new span t_i and a target retrieval model, we hypothesise 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 $T_b = \{t_0, \ldots, t_i, t_s, \ldots, t_n\}$, $R_\theta(q, T_a) > R_\theta(q, T_b)$.

We investigate how the text that appears prior to a given span affects its contextualisation. In most applications, this is expected behaviour and necessary for an effective embedding [51]. 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.

Defence within Retrieval. To defend against the attack defined in Section 3, we look to return a ranking that penalises promotional content whilst still being largely weighted by relevance score, we achieve this via interpolation of the two models. 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 a 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 posterior probability in promotion being present over the entire text and the maximum span posterior probability over a sentence sliding window. We hypothesise 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 how a sliding window over sentences can reduce the opportunity for unwanted contextualisation to occur. The notion being that without contextualisation of surrounding non-promotional text, the decision boundary of promotion being present can be more clearly defined. 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 set of sentences. The output of the classifier is a posterior probability of promotion being present. We subtract this value from 1, so that $1 - R_\phi$ reflects the posterior 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 normalise 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 agnostic of some downstream task, as we hypothesise that the surrounding context of a non-relevant span can affect the approximation of relevance over the entire text. As such, 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.

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

One would expect that in a real world attack, an adversary would be uninterested in the injection of arbitrary text, they may instead want to present sentiment towards an entity regardless of relevance to the given context on a large scale. Consequently, we look to exploit the generative ability of LLMs investigating the information required for these models to perform contextualised generation whilst completing a ulterior objective.

- **RQ-2**: Can an LLM contextualise to a text whilst completing a ulterior objective?

Given a pipeline for injecting contextualised generation into relevant documents we look to investigate what steps can be taken to reduce the vulnerability of NRMs to text injection without assuming access to the original generative model.

- **RQ-3**: RQ-3: How can we defend against contextualised generation without having access to the generative model?

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.

RQ-1: When injecting spans of text, how is relevance affected by positional changes? We conduct two initial evaluations probing ranking model preference for a ranking of documents (top-10) before and after it is augmented with spans of text from relevant, partially-relevant and non-relevant documents. The augmentation process works as follows: for each top-document retrieved for a query (within the top-10), we inject a salient span extracted from a randomly selected relevant document for that query. 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 non-relevance. We now describe two modes of injecting the salient spans in the top-retrieved documents for a query.

We first sentence tokenise and inject a salient span from a known relevant document to some position in the target document. We first investigate absolute position (start, middle and end) where this **salience-agnostic** method is oblivious of the relative similarities between the document sentences and the query. We then consider a **salience-aware** positional injection tests our hypothesis on attention bleed-through (Proposed in Section 3) in which a text span is injected before or after the most salient (closest in vector distance) sentence in each target document with respect to the target query.

RQ-2: Can an LLM contextualise to a text whilst completing a ulterior objective? As context for generation, we retrieve the top-k documents per query for each target model, we then use the prompt structure shown in footnote[2]. Given a promotional span for each document we then inject the span into the document using one of the positions described in RQ-1. Each generated span is truncated to a single sentence such that text length is not a confounding factor when comparing to static content.

We perform a pairwise evaluation comparing static promotional text and document contextualised text where fundamentally, we look to see by how much rank is affected at each injection position and each models empirical preference for contextualised 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.

From a subset of scraped Wikipedia rejected edits dataset [2] we chose 5 spans that explicitly referenced an entity in common knowledge, these edits were rejected by human editors specifically for being promotional. We use these entities in generating contextualised spans such that a pairwise comparison can be made between the edits. 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 1.

As the IR test collection, we use the MSMARCO passage collection [1], 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 [5] which contains relevance judgements for documents with respect to 43 queries (See Section 4).

Table 1: Proposed target entities 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.

Models. We investigate NRMs that are based on both 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. al. [29]) to contrast against the transformer based models. We compute metrics over the top 1000 retrieved documents for each query per model to ensure metrics such as mean rank change outline in Section 4 are precisely computed unless otherwise stated.

- **ColBERT** [14]: A BERT based end-to-end bi-encoder using the late interaction paradigm where documents and queries are encoded separately (checkpoint trained by Wang et. al. [44]). The maximum similarity between query and document terms is used to determine relevance score.
- **MonoT5** [27]: 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'.
- **Tas-Balanced (Tas-B)** [12]: A teacher distilled bi-encoder 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.

We investigate two methods of determining salience through embedding similarity to test our attention bleed-through hypothesis (See Section 3). We look to assess the effect of salience chosen by *semantic space* similarity versus *relevance space* similarity in attacking ranking models. The first is a SBERT distillBERT model (sentence-transformers/nq-distilbert-base-v1) [35] (**Semantic** as shown in Table 2) acting as a task unbiased measure of semantic distance between spans and the target query. The second is a fine-tuned MonoT5 model trained with query-document pairs to determine relevance (**Relevance** as shown in Table 2) *specifically* for the MSMARCO corpus [1].

Metrics. When injecting text into documents, we evaluate our attack on the following metrics determining the effect of the injection on relevance scores of each target model. 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,

[2] 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 <>)

injection of multiple augmented documents is left to the defence component of this evaluation.

ABNIRML [21]. We first evaluate using the ABNIRML score of the augmented set versus the original retrieved documents. Conceived by Macavaney et. al. [21], the purpose of the ABNIRML score is 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 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 $\text{sign}(R_\theta(q, d) - R_\theta(q, d^+))$ [3]. 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 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, $\{d_1, d_2, \ldots, d_i, \ldots, d_k\}$, if d_i has been augmented we replace it with d_i^+. The set becomes $\{d_1, d_2, \ldots, d_i^+, \ldots, d_k\}$. We then take the pairwise difference between the original and augmented sets to observe the magnitude of the effect of injection on relevance.

Semantic Difference (SD). To assess how well the language model has contextualised 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 contextualised 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.

LLM Implementation Details. To assess RQ-2, we use a zero-shot approach to prompting the generative language model. Specifically we use Alpaca [42] which is an instruction-tuned llama-7B model [43]. We experimented with the base llama-7B model but found that a balance could not be found between creative output and consistently managing to capture context from the document. Due to computation constraints we use int8 quantization, this method reduces memory footprint significantly at minimal cost to performance [7]. To control token generation we use contrastive search ($k = 10$, penalty $\alpha = 0.2$) to control output [40].

After manual inspection of a set of candidate prompts we determined a suitable prompt format that allowed for contextualisation of downstream tasks (see[2]). Qualitative evaluation was then used to tune the temperature parameter of the generation process 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. We found that few-shot prompting was ineffective in improving contextualisation, 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 [4] and GPT [3].

Table 2: Experiments injecting human-judged relevant spans by position (RQ-1) 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 $p < 0.05$). BM25 acts as a control as it is position invariant.

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

4.1 Results and Discussion

4.1.1 RQ-1. We first investigate the effects of context on arbitrary content injections of varying relevance. In Table 2 we observe on the first line of each table section, that placing a span of text at the start of a document regardless of true relevance will generally lead to a substantial 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 Tas-B and ColBERT we observe a minimal preference for the original set when injecting text at the end of the document as observed in columns 4 and 6 of rows 3, 10 and 17. Furthermore positional injection to the middle or end of a document in all cases reduced preference relative to injection at the start. We hypothesise that the effect on the overall contextualisation of the document is lower in these cases as there is less or no text after the injected span to be negatively affected by its content.

[3] This variation on ABNIRML occurs when $\delta = 0$ as the metric was found to be insensitive to the value of δ [21]

Effect of injection before a salient span. In rows 4-7, 11-14 and 19-22 of Table 2 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. We see that consistently ABNIRML score is lower when spans are injected after a salient span supporting the hypothesis of attention bleed-through. Interestingly placing a non-relevant span before the most salient span can in some cases be considered worse than injection at the start of the span. By ABNIRML score, Tas-B shows 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 from Tables 2 and 3 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 as shown in rows pertaining to Semantic - After and Relevance - After. 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 2). As we observe that generally in the case of partial relevance, salient injection outperforms positional injection, we further consider its utility for contextualised 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.

4.1.2 RQ-2. In Table 3 we analyse both the effect of injecting contextualised generation with position relative to the most salient span and the comparison of static and contextualised content.

Contextualisation effectively reduces negative effects. In each case, with contextualisation 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 of table 3). BM25 acts as a point of comparison 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 hypothesise 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 suggesting that semantic similarity to a query text is a better indicator of salience even within the context of a ranking task. However, there is minimal difference between injection after the salient span in both cases.

Effective positional injection amplifies contextualisation. It is clear from Table 3 that the combination of positional injection and contextualisation 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, it succeeds in removing the majority of augmented documents from the top-10 as is evident from a minimum MRC of 9.232. Given the efficacy of Tas-B, we propose that robustness will reduce the effect of these contextualising attacks. However, it does not solve the problem of positional injection that is inherent to the transformer.

Semantic injection generally outperforms relevance injection although not by a substantial margin, whilst requiring no corpus specific training. Furthermore we observe that non-disilled models (MonoT5, ColBERT) are more susceptible to both contextualisation and positional injection versus a teacher-distilled model (Tas-B) (With Relevance - After, MRC values are 4.9590, 4.7030 and 9.9305 respectively) frequently failing to remove documents containing contextualised generation from the top-10 ranking.

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 contextualised promotion in Table 3). We hypothesise that contextualisation is an essential component of this attack as in a sub-optimal position as the undesirable traits have greater effect as greater contextualisation occurs when text is injected after a salient span.

Qualitative Analysis of Contextualisation. Though in Figure 1 we present a successful example of contextualisation, there are examples of failure to capture a relation between the entity and context document. The LLM would fail to contextualise such that any benefits from context are lost. 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 is partially dependent on the LLM having observed information regarding the objective inputs in training. The examples presented above show documents with specialist topics such that the LLM cannot derive context from 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 however, to alleviate this issue one could tailor a prompt to an entity by including specific facets to promote or a general description.

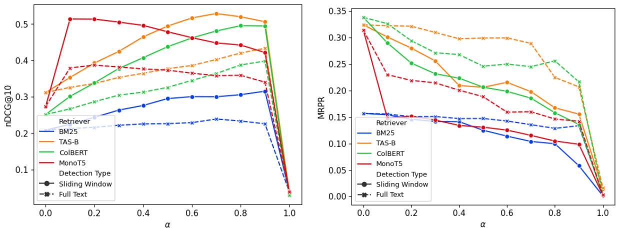
5 DEFENCE EVALUATION

We now discuss the evaluation and findings of our defence against contextualised generation when promoting entities.

RQ-3: How can we defend against contextualised generation without having access to the generative model? Our defence assumes no access to generated text from a *particular* generative model, hence we use a classifier in a zero-shot manner. We

Table 3: 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 (RQ-2), recording ABNIRML score (↓) and Mean Rank Change (↓). For Semantic Difference (SD), statistically significant results are with respect to the static text 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		MonoT5		Tas-B		
Injection	ABNIRML	MRC	ABNIRML	MRC	ABNIRML	MRC	ABNIRML	MRC	SD
Static Text									
Semantic - *Before*	0.9724	8.9379	0.9680	12.6640	0.8796	10.4783	0.9372	15.9462	N/A
Semantic - *After*	0.9724	8.9379	0.9680	10.4000†	0.8328†	6.9231†	0.8744†	11.3184†	N/A
Relevance - *Before*	0.9724	8.9379	0.9760	14.6280	0.9264	14.7826	0.9461	20.3767	N/A
Relevance - *After*	0.9724	8.9379	0.9600†	11.384†	0.8194†	7.9565†	0.8834†	12.3184†	N/A
Contextualized Text									
Semantic - *Before*	0.7724	8.2897	0.7480	6.8510	0.7358	6.7968	0.8094	12.7724	0.0363†
Semantic - *After*	0.7724	8.2897	0.6900†	5.0510†	0.6806	**4.500**†	**0.7489**†	**9.2320**†	0.0237†
Relevance - *Before*	0.7724	8.2897	0.8220	9.7370	0.8227	9.7316	0.8767	17.4753	0.0364†
Relevance - *After*	0.7724	8.2897	**0.6460**†	**4.7030**†	**0.6388**†	4.9590†	0.7534†	9.9305†	0.0231†



(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 (RQ-3). Graph (b) clearly shows a reduction in the rank of augmented documents inline with increase in nDCG@10 performance. It can also be observed that a sliding window (dotted x) outperforms classification over the full text (solid dot) in all cases.

employ the SemEval 2020 Task 11 propaganda dataset [6]. In particular this dataset comprises challenging examples constituting 'weasel' words, i.e., subtle methods of promotion or discrediting of an entity [2, 28]. 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. We balanced the training set after the combination of the 17 labels such that we have an equal distribution of propaganda and non-propaganda.

We train a RoBERTa base model [18] in a binary classification setting such that we can have a posterior probability of promotion being present by taking the confidence of the classifier. We use RoBERTa because prior work has shown success in the main SemEval task leveraging RoBERTa [34, 39]. We use a learning rate of $1e^{-5}$ and train for 10 epochs; with a batch sise of 8 and a linear quarter-epoch learning rate warm-up phase. We use the Transformers library[4] for all neural models in this evaluation selecting the strongest model checkpoint based on validation performance for final in-domain test evaluation.

We evaluate in an out-of-domain binary classification setting comparing sliding window and full document inference. 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.

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 [22] to compute all relevance metrics (Relevance judgement > 2). To clearly show that our defence succeeds in penalizing promotion, we then create a synthetic set of relevance judgements. For each augmented document d^+ we take the judged relevance score to be the original relevance score of d minus 1 due to the reduction in utility to a user when promotion is present. 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 make a distinction and name **Mean Reciprocal Promotional Reduction (MRPR)** as we want to *minimise* this metric.

To evaluate promotion detection, 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. We then evaluate interpolated fusion of a classifier and 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 judged to be highly relevant to be retrieval model. We then evaluate this defence with relevance judgements for augmented documents using MRPR.

5.1 Results and Discussion

Classifier Evaluation. We observed effective classification results with in-domain accuracy of 85%. In Table 4 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 no desirable

[4] HuggingFace Transformers API [46]

Table 4: Zero-shot performance of RoBERTa classifying static and contextualized promotions when placed after the most salient span. (RQ-3)

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

Table 5: Evaluation of our defense against contextualised promotion (RQ-3) recording nDCG@10 (↑), MRR (↑) and MRPR (↓) with optimal α^* tuned by NDCG@10, also showing relative change with respect to baseline retrieval ($\alpha = 0.0$). We omit absolute positions start and middle for brevity only contrasting salient position with end. 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)
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)

context can contextualise an injected 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.

Ranking Task Evaluation. We observe significant improvements in retrieval performance using our defence presented in Table 5 noting relative difference to a baseline retrieval setting. Best performance tuned on nDCG@10 is found within α ranges [0.6,0.9] evident from column 2 of Table 5, however somewhat anomalously MonoT5 yields best performance at $\alpha = 0.1$ compared to other models in which higher α values improve retrieval quality. This can most likely be attributed to the other two models being bi-encoders which may lead to a different distribution of relevance scores.

In experiments with BM25 we find that though a reduction in MRPR is observed, change in MRR is small such that promotional cotent is still present at high ranks. It is evident that the most effective injection positions found in experiments using ABNIRML and MRC (Compare Table 3 with Table 5) 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 affected by this attack. 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 5), salient injection yields higher MRPR showing that over the entire ranked set with interpolated fusion (rows 2, 4 of each model section in Table 5), augmented documents maintain a higher ranking in spite of this defence showing that the attack is substantially effecting each NRM.

Evaluation with Promotion Relevance Judgements. In a sanitised setting with relevance judgements for augmented documents, we observe the expected trend that increasing α can markedly reduce the rank of these contextualised injections. Hereby showing the efficacy of this interpolation as observed in Figure 3 (b). Coupled with the observation that at higher values of α, MRR increases as shown in Table 5, we see this is an effective defence against the injection of contextualised 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 penalised. 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.

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 penalise promotion. In Figure 3 (a) the anomalous response of MonoT5 is illustrated, in all cases relevance score was normalised 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 Tas-B.

6 CONCLUDING REMARKS

We have presented a novel attack using both positional injection and contextualisation via language models to contextualise LLM generated text reducing negative effects on the rank of augmented documents across multiple 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 LLM generated promotion using maximum span confidence over each text which increase nDCG@10 significantly under a classic evaluation setting by reducing the effect of contextualisation and bleed-through. We discuss wider implications of these experiments on semantic search and concerns that the contextualised 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.

7 ETHICAL CONSIDERATIONS

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. In future we would look to assess context attacks with larger models such that the failures noted in section 4.1.2 could be alleviated. 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 open models that now exist such that it could become infeasible to use model-specific checks. We hypothesise that positional injection probing with contextualisation could be useful tools in the evaluation of neural 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 ulterior objective as shown in this work.

REFERENCES

[1] 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 (Nov. 2016). https://doi.org/10.48550/arxiv.1611.09268 Publisher: CEUR-WS.

[2] Amanda Bertsch and Steven Bethard. 2021. Detection of Puffery on the English Wikipedia. 329–333. https://doi.org/10.18653/v1/2021.wnut-1.36

[3] 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. https://doi.org/10.48550/arXiv.2005.14165 arXiv:2005.14165 [cs].

[4] 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. https://doi.org/10.48550/arXiv.2204.02311 arXiv:2204.02311 [cs].

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. (March 2020). https://doi.org/10.48550/arxiv.2003.07820

[6] 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*. International Committee for Computational Linguistics, Barcelona (online), 1377–1414. https://doi.org/10.18653/v1/2020.semeval-1.186

[7] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. https://doi.org/10.48550/arXiv.2208.07339 arXiv:2208.07339 [cs].

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805 arXiv:1810.04805 [cs].

[9] Saadia Gabriel, Hamid Palangi, and Yejin Choi. 2022. NaturalAdversaries: Can Naturalistic Adversaries Be as Effective as Artificial Adversaries? arXiv:2211.04364 [cs.CL]

[10] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. https://doi.org/10.48550/arXiv.2212.10496 arXiv:2212.10496 [cs].

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. https://doi.org/10.48550/arXiv.1412.6572 arXiv:1412.6572 [cs, stat].

[12] 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. https://doi.org/10.48550/arXiv.2104.06967 arXiv:2104.06967 [cs].

[13] 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. 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[14] 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* (April 2020), 39–48. https://doi.org/10.48550/arxiv.2004.12832 ISBN: 9781450380164 Publisher: Association for Computing Machinery, Inc.

[15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. https://doi.org/10.48550/arXiv.2205.11916 arXiv:2205.11916 [cs].

[16] Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. TagGPT: Large Language Models are Zero-shot Multimodal Taggers. https://doi.org/10.48550/arXiv.2304.03022 arXiv:2304.03022 [cs].

[17] Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating Verifiability in Generative Search Engines. https://doi.org/10.48550/arXiv.2304.09848 arXiv:2304.09848 [cs].

[18] 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. https://doi.org/10.48550/arXiv.1907.11692 arXiv:1907.11692 [cs].

[19] 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. arXiv:2304.14867 [cs.IR]

[20] Nicholas A. Lord, Romain Mueller, and Luca Bertinetto. 2022. Attacking deep networks with surrogate-based adversarial black-box methods is easy. https://doi.org/10.48550/arXiv.2203.08725 arXiv:2203.08725 [cs].

[21] 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 (2022), 224–239. https://doi.org/10.1162/tacl_a_00457

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

[23] Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative and Pseudo-Relevant Feedback for Sparse, Dense and Learned Sparse Retrieval. https://doi.org/10.48550/arXiv.2305.07477 arXiv:2305.07477 [cs].

[24] Natalie Maus, Patrick Chao, Eric Wong, and Jacob Gardner. 2023. Adversarial Prompting for Black Box Foundation Models. https://doi.org/10.48550/arXiv.2302.04237 arXiv:2302.04237 [cs].

[25] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1906–1919. https://doi.org/10.18653/v1/2020.acl-main.173

[26] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Re-ranking with BERT. https://doi.org/10.48550/arXiv.1901.04085 arXiv:1901.04085 [cs].

[27] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. 708–718. https://doi.org/10.18653/v1/2020.findings-emnlp.63

[28] Douglas E. Ott. 2018. Hedging, Weasel Words, and Truthiness in Scientific Writing. *JSLS : Journal of the Society of Laparoendoscopic Surgeons* 22, 4 (2018), e2018.00063. https://doi.org/10.4293/JSLS.2018.00063

[29] 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*. Springer, 517–519.

[30] Yunxiao Qin, Yuanhao Xiong, Jinfeng Yi, and Cho-Jui Hsieh. 2021. Training Meta-Surrogate Model for Transferable Adversarial Attack. https://doi.org/10.48550/arXiv.2109.01983 arXiv:2109.01983 [cs].

[31] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. 2022. Boosting the Transferability of Adversarial Attacks with Reverse Adversarial Perturbation. https://doi.org/10.48550/arXiv.2210.05968 arXiv:2210.05968 [cs].

[32] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. arXiv:2306.17563 [cs.IR]

[33] Saliman Sutskever Radford, Narasimhan. [n. d.]. Improving language understanding with unsupervised learning. https://openai.com/research/language-unsupervised

[34] 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. https://doi.org/10.48550/arXiv.2009.07473 arXiv:2009.07473 [cs].

[35] 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* (Aug. 2019), 3982–3992. https://doi.org/10.48550/arxiv.1908.10084 ISBN: 9781950737901 Publisher: Association for Computational Linguistics.

[36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938 [cs.LG]

[37] Stephen E. Robertson. 1997. The probability ranking principle in IR. *Journal of Documentation 33* (1997), 294–304.

[38] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large Language Models are Strong Zero-Shot Retriever. https://doi.org/10.48550/arXiv.2304.14233 arXiv:2304.14233 [cs].

[39] Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newsSweeper at SemEval-2020 Task 11: Context-Aware Rich Feature Representations For Propaganda Classification. https://doi.org/10.48550/arXiv.2007.10827 arXiv:2007.10827 [cs].

[40] Yixuan Su and Nigel Collier. 2023. Contrastive Search Is What You Need For Neural Text Generation. https://doi.org/10.48550/arXiv.2210.14140 arXiv:2210.14140 [cs].

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. https://doi.org/10.48550/arXiv.1312.6199 arXiv:1312.6199 [cs].

[42] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[43] 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. https://doi.org/10.48550/arXiv.2302.13971 arXiv:2302.13971 [cs].

[44] 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)*. Association for Computing Machinery, New York, NY, USA, 2790–2800. https://doi.org/10.1145/3477495.3531721

[45] Xiao Wang, Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2023. Generative Query Reformulation for Effective Adhoc Search. arXiv:2308.00415 [cs.IR]

[46] 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. https://doi.org/10.48550/arXiv.1910.03771 arXiv:1910.03771 [cs].

[47] 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. https://doi.org/10.48550/arXiv.2204.01321 arXiv:2204.01321 [cs].

[48] Dingcheng Yang, Zihao Xiao, and Wenjian Yu. 2022. Boosting the Adversarial Transferability of Surrogate Model with Dark Knowledge. https://doi.org/10.48550/arXiv.2206.08316 arXiv:2206.08316 [cs].

[49] Yuting Yang, Pei Huang, Juan Cao, Jintao Li, Yun Lin, Jin Song Dong, Feifei Ma, and Jian Zhang. 2022. A Prompting-based Approach for Adversarial Example Generation and Robustness Enhancement. https://doi.org/10.48550/arXiv.2203.10714 arXiv:2203.10714 [cs].

[50] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How Language Model Hallucinations Can Snowball. https://doi.org/10.48550/arXiv.2305.13534 arXiv:2305.13534 [cs].

[51] Mengjie Zhao, Philipp Dufter, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2020. Quantifying the Contextualization of Word Representations with Semantic Class Probing. *CoRR* abs/2004.12198 (2020). arXiv:2004.12198 https://arxiv.org/abs/2004.12198

[52] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less Is More for Alignment. https://doi.org/10.48550/arXiv.2305.11206 arXiv:2305.11206 [cs].

[53] Mo Zhou, Zhenxing Niu, Le Wang, Qilin Zhang, and Gang Hua. 2020. Adversarial Ranking Attack and Defense. https://doi.org/10.48550/arXiv.2002.11293 arXiv:2002.11293 [cs].

[54] Banghua Zhu, Jiantao Jiao, and Michael I. Jordan. 2023. Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons. https://doi.org/10.48550/arXiv.2301.11270 arXiv:2301.11270 [cs, math, stat].

[55] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043 [cs.CL]

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009