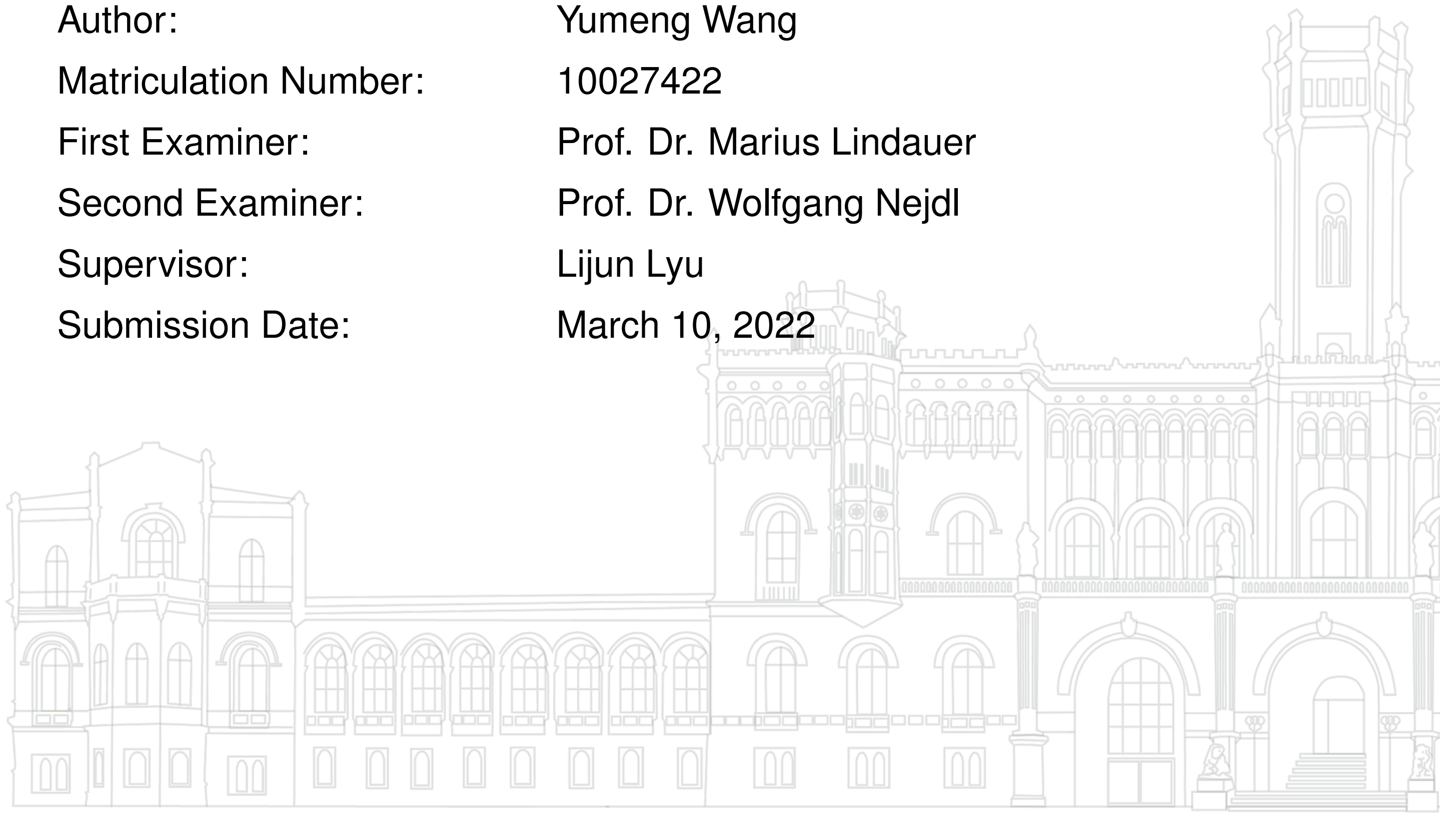
Gottfried Wilhelm Leibniz Universität Hannover
Institut für Verteilte Systeme
Fachgebiet Wissensbasierte Systeme

Master Thesis

Electrical Engineering and Information Technology (M. Sc.)

Global Triggers for Attacking and Analyzing Ranking Models

Author:	Yumeng Wang
Matriculation Number:	10027422
First Examiner:	Prof. Dr. Marius Lindauer
Second Examiner:	Prof. Dr. Wolfgang Nejdl
Supervisor:	Lijun Lyu
Submission Date:	March 10, 2022



Abstract

Text ranking models based on BERT are now well established for a wide range of passage and document ranking tasks. However, the robustness of BERT-based ranking models under adversarial attack is under-explored. In this work, we argue that BERT-rankers are vulnerable to adversarial attacks targeting retrieved documents given a query.

We propose algorithms for generating adversarial perturbation of documents locally to individual queries or globally across the dataset using gradient-based optimization methods. The aim of our algorithms is to add a small number of tokens to a highly relevant or non-relevant document to cause a significant rank demotion or promotion.

Our experiments show that a few number of tokens can already change the document rank by a large margin. Besides, we find that BERT-rankers heavily rely on the document start/head for relevance prediction, making the initial part of the document more susceptible to adversarial attacks.

More interestingly, our statistical analysis finds a small set of recurring adversarial tokens that when concatenated to documents result in successful rank demotion/promotion of any relevant/non-relevant document respectively. Finally, our adversarial tokens also show particular topic preferences within and across datasets, exposing potential biases from BERT pre-training or downstream datasets.

Keywords: ranking, BERT, neural networks, adversarial attack, biases

Contents

List of Figures

List of Tables

Glossary

Bi-LSTM This model is introduced in [16]. For details see section 2.2 ”Universal attacks in NLP”.

ClueWe09 The Clueweb09 dataset is a dataset often used in NLP andIR research. See more details in section 4.1.

CV Computer Vision.

DREBIN DREBIN is a dataset for malware detections introduced in [1].

FGSM Fast Gradient Sign Method is an image attack algorithm introduced in [14]. For details see subsubsecion 2.1.2.1 ”White-box Attack”.

GAN Generative Adversarial Network is a new framework for estimating generative models via an adversarial process, introduced by [13].

IR Information Retrieval.

JSMA Jacobian Saliency Map Adversary is an adversarial attack method introduced in [28].For details see subsubsecion 2.1.2.1 ”White-box Attack”.

KDE Kernel Density Estimation is a method to estimate the probability density function of a random variable. See more details in subsubsection 5.3.3.1 ”Distance-wise Analysis”.

KNNs K Nearest Neighbors is a supervised learning method for classification and regression. See more details in subsubsection 5.3.3.1 ”Distance-wise Analysis”.

NLP Natural Language Processing.

PCA Principal component analysis is commonly used for dimensionality reduction. See more details in subsubsection 5.3.3.2 ”Visualization in 2D”.

SEARs The Semantically Equivalent Adversarial Rules are used in [35].

SQuAD Stanford Question Answering Dataset is a reading comprehension dataset.

TREC-DL Here we use TREC-DL 2019 passage ranking dataset, which is constructed from MsMarco dataset. See more details in section 4.1.

t-SNE T-Distributed Stochastic Neighbor Embedding is a tool to visualize high-dimensional data. See more details in subsubsection 5.3.3.2 ”Visualization in 2D”.

1. Introduction

1.1 Motivations

Previous studies have shown [14, 40] that deep neural networks are vulnerable to adversarial attacks. Some minor imperceptible perturbations can fool the model to make false predictions. In the NLP community, since [20], many black-box and white-box based attack approaches have been developed and the attack effectiveness has been demonstrated via downstream tasks like machine-translation system, sentiment analysis, text generation, and question answering [43, 41].

From an attack perspective, adversarial attacks can expose the vulnerability of the models [43]. Especially in security-sensitive related works, once the model vulnerability is uncovered and exploited by malicious attackers, not only the normal function is compromised, but well-concealed attacks will be challenging to detect. For example, in spam email filter, attackers often take small moves such as modifying or inserting undetectable characters or words to evade inspection [4, 3].

Furthermore, adversarial attacks can provide new insights into model interpretability. Shi Feng et al. [10] crafts adversarial examples by iteratively removing the least important word from the input, which exposes pathological behaviors of neural models that the model prediction is influenced by the unimportant words. More enlighteningly, adversarial attacks can reveal the potential biases of the models or the training corpus. Jieyu Lin et al. [25] demonstrates the existence of statistical bias by conducting an adversarial attack on the machine reading comprehension model. They found that the model tended to choose the options which combined other options, like 'A,B and C' or 'All the above'.

As a central task in the intersection field of IR and NLP, text ranker can benefit a lot from adversarial attacks as well. Search engines optimization (SEO) companies and e-commerce websites arouse substantial interest in crafting adversarial examples for ranking models, by which they make rank promotion of arbitrary document feasible for production deployment. However, the research on attacks against text ranking models remain insufficiently explored by prior studies. Existing works on document perturbations for text ranking primarily focus on black-box attacks with limited applicability [32], human-assisted adversarial examples [15] or for interpretability of rankers [37]. This thesis tries to close this gap by supposing a white-box gradient-based attack algorithm, together with the aim at exposing the bias of the BERT-based ranking models.

1.2 Research Questions

Unlike other NLP tasks, such as text classification, which only deal with one single decision in a range, ranking models output a ranked list of candidate documents given a specific query involving an aggregation of a bunch of decisions. In this thesis, we define the problem of adversarial ranking attacks that generate adversarial tokens when concatenated to a retrieved document significantly shift the rank of the document.

We consider two reasonable attack scenarios - (a) *Rank demotion* attack on a highest-ranked document, and (b) *Rank promotion* attack on a bottom-ranked document. Additionally, we define the scope of adversarial ranking attacks on a per-query level and on an entire query workload. In response to this, we design firstly a *local ranking attack* where perturbations are generated on a retrieved document given a query. Secondly we formulate a *global ranking attack* in which we intend to generate adversarial tokens for an entire workload of queries. Our adversarial attack approaches are based on gradient-based token search algorithms from local [9] and global [43] attacks.

We derive the following primary research questions:

RQ1 Do ranking models show vulnerability to adversarial attacks?

RQ2 How is the attack effectiveness when finetuned on different datasets, and what exerts an influence on attack effectiveness?

RQ3 If the adversarial attacks exhibit similarities or observable patterns across queries or across datasets in both attack scenarios.

We conduct our experiments on base-uncased BERT model fine-tuning on ClueWeb09 and TREC-DL datasets. Figure 1 illustrates an example of the local adversarial attack in ClueWeb09 resulting in a significant rank shift. Our results suggest a minor perturbation on the text documents, especially on the start position, can cause a significant rank shift, both on local and global attack, revealing the vulnerability of BERT ranking

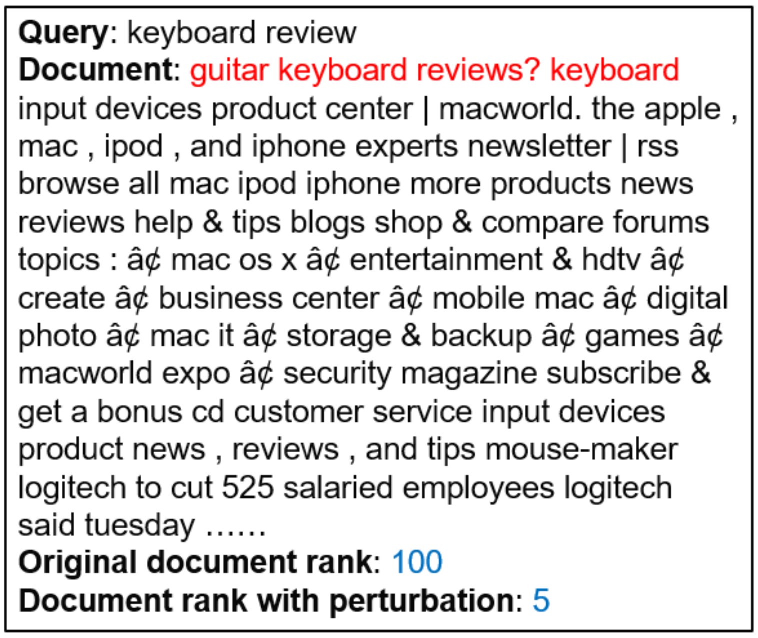

Query: keyboard review
Document: guitar keyboard reviews? keyboard input devices product center | macworld. the apple , mac , ipod , and iphone experts newsletter | rss browse all mac ipod iphone more products news reviews help & tips blogs shop & compare forums topics : â¢ mac os x â¢ entertainment & hdtv â¢ create â¢ business center â¢ mobile mac â¢ digital photo â¢ mac it â¢ storage & backup â¢ games â¢ macworld expo â¢ security magazine subscribe & get a bonus cd customer service input devices product news , reviews , and tips mouse-maker logitech to cut 525 salaried employees logitech said tuesday ......
Original document rank: 100
Document rank with perturbation: 5

Figure 1: An example from ClueWeb09. A perturbation (in red) on the front of the text to a given query results in a rank promotion from 100 to 5.

models. Moreover, we observe in a further post-hoc analysis that the adversarial tokens generated from local ranking attack recur across different queries and datasets. Subsequent experiments also unveil potential model bias towards specific topics deriving from dataset and BERT pretraining procedure.

1.3 Thesis Structure

In this chapter, we explained the motivation for realizing this thesis, defined research problem and posed research questions. Chapter 2 gives an overview of the related works. We first introduce the notations related to adversarial attacks and adversarial attacks in NLP in section 2.1. Universal attacks in NLP and attacks in text rankers are later presented in section 2.2 and 2.3. Inspired by interpretability for text-based ranking models, we introduce the explainer method in section 2.4. Chapter 3 states problems of local and global adversarial ranking attacks and describes in detail the implementation process of the attack algorithm, including the generation of adversarial tokens and a subsequent replacement strategy. Following the experimental setup in chapter 4, thorough evaluation results are presented in chapter 5. For both attack scenarios (rank demotion and rank promotion), we firstly conducted both local and global attack experiments on attack effectiveness across different datasets in section 5.1. Adversarial token analysis for rank demotion and rank promotion attacks, and a further bias analysis are introduced in section 5.2 and 5.3 respectively. According to the results, we also discuss hypothesis on bias tokens, document redundancy and token analysis in word embedding level in section 5.3. After a conclusion of this thesis, a discussion on the limitations of our work and interesting lines of future research are presented in section 6.2.

2. Related work

In recent years with the rapid development of high computational devices, DNNs gain substantial improvement in various communities like Computer Vision, Natural Language Processing and Game Theory. However, due to the black-box characteristic of deep neural networks, it is found hard to explain the model, and further to evaluate the robustness if the model. Many research use human unperceivable perturbations to evaluate DNN model's robustness. It turns out that DNN models are vulnerable to these small perturbations. Adversarial attack on DNN models provides new avenues for a better interpretability of the model.

Adversarial attack was first developed in the CV community. [40] were the first to apply adversarial attack on neural models with small perturbations in image classification task. They successfully caused model to predict unsatisfied label just by adding human imperceptible perturbation that are generated by optimizing inputs to maximize the prediction error. [40] first invented notation 'adversarial examples' and it is represented for all kinds of perturbations generally in adversarial attack.

Due to the discrete nature of language, there are some differences that prevent the direct application of the computer vision attack methods to Natural Language Processing attacks. This chapter first summarizes difficulties and methods about adversarial attacks in NLP. Then we specifically introduce universal attacks and adversarial attacks in text ranking that are related to our work. Finally an inspired method from the view of interpretability of rankers is presented in the section 2.4.

2.1 Adversarial Attacks in NLP

2.1.1 From Images to Texts

Unlike adversarial perturbations on images where arbitrarily small changes are possible in the image space, text data is different and arguably more challenging. Due to the discrete nature of language, small changes in the input space in the form of word additions or replacements can easily cause a big difference in embedding vectors.

Following are three main differences to be discussed:

- *Input instance type*: Discrete vs Continuous. Image inputs are generally continuous, with pixel values within a certain range, and evaluation metrics are specifically used to evaluate distance or differences, such as the common L_p norm which measures the difference between the actual image input and the adversarial examples containing small perturbations, are also continuous. However, this is not the case with natural language processing. Text input is discrete, and it is hard to define a perturbation for a text message. So, to implement attacks with small perturbations on natural language processing, one must first be careful to design reasonable text perturbations and to design evaluation metrics used to measure distance.

- *Perceivable vs. Unperceivable*: The range of pixel values may be extensive, but it is hard for humans to notice small color changes, even harder for a picture with complex textures in diverse colors, where part of the pixel changes can easily be overlooked. Hence attacks on computer vision can generally achieve good results. However, changes in the text are often challenging to accomplish without being noticeable. For example, adding or removing words at the word level can cause semantic changes, and insertions at the character level can be easily detected by spelling correction software. Therefore, it is urgent and vital to design approaches to generate human unperceivable adversary examples for NLP tasks.

- *Semantic vs. Semantic influenced*: When attacking images, the adversarial examples we generate to obfuscate the model do not generally appear semantically different from the original instances because these small perturbations are challenging to detect. However, in NLP tasks, small perturbations can often cause first spelling differences, then syntactic disfluencies and semantic changes. For example, deleting or adding a negative word or even changing its position can reverse the semantic of a sentence. An ideal attack on an NLP model should try to add small perturbations that are difficult to detect without changing the semantics while also fooling the model into giving false predictions.

Due to all the differences described above, current state-of-the-art attack methods on Natural Language Processing tasks are carefully transplanted from attacks on Computer Vision with constraints or develop novel and unique attack approaches.

2.1.2 Adversarial Attack Methods in NLP

There are two major lines of attacking approaches in NLP tasks such as text classification and natural language inference. The first is without knowing model architectures and parameters, namely under the black-box setup, the attacker uses predefined heuristic rules to generate natural (in human perspective) substitutes of words or sentences, so that to fool the victim model. The prevalent heuristics include synonym replacement [21, 33], mask-and-fill by contextualized language models [22, 23], human-in-the-loop strategy [42], etc. On the other hand, white-box attack assumes full access to model parameters. Thus the attacker can use gradient signals to guide the searching process within a few rounds of forward and backward computation [9]. In this section we introduce state-of-the-art white-box and black-box attacking methods in NLP tasks respectively.

2.1.2.1 White-box Attack

In white-box attack setting, attackers have full access to all information of the targeted model, which includes input data, model predictions, model architecture and parameters, activation function and loss function. This complete information provides the attacker a wide operating platform to perform attack techniques, and therefore remarkable attack performance can often be achieved. In this subsection, we introduced some state-of-the-art white-box attack methods on NLP tasks.

Fast Gradient Sign Method (FGSM)

Among the image attack algorithms, FGSM is a very classical one. [14] generates the attack noise by gradient. The purpose of this image attack is not to modify the parameters of the classification network, but to make the adversarial example able to misclassify with high confidence by perturbing the pixel values of the input image. Since the aim of the targetless attack is to have the model misclassify the input image into any category other than the correct one, this goal can be achieved by increasing the loss value, i.e., the probability of the model prediction corresponding to the true label is as small as possible. So we just need to add the calculated gradient direction to the input image so that the loss value is larger than that when the image goes through the classification network before the perturbation, in other words, the probability of the model predicting correctly becomes smaller.

FGSM gains a lot of follow-up works in attacking NLP tasks. TextFool [24] make use of the gradient-based concept to make a significant contribution of adversarial attack on text classification tasks. They proposed three perturbation strategies on textual attack, namely insertion, modification and removal in both character-level and word-level. However, these methods are performed in a relative high labor cost since they manually implement them, as mentioned by the authors.

Jacobian Saliency Map Adversary (JSMA)

JSMA is another pioneer work on adversarial attacking neural models in CV community. Unlike FGSM performing attacks with gradients, [28] proposed a new class of algorithms for generating adversarial examples. They exploited forward derivatives (i.e., Jacobian matrix of the neural model) presenting the learned behavior of DNNs.

This method evaluates the sensibility of the neural model's output to each input component by calculating its Jacobian matrix which provides a great control platform for adversarial perturbations. The adversarial saliency maps composed of Jacobian matrices rank each input component based on their contributions to the adversarial target, which later help to select adversarial examples.

[18] (and [17]) is the first work to attack malware detectors adapting JSMA method, but address the challenges appearing in the transition from continuous and differentiable image input to discrete and restricted DREBIN [1] input for malware detections. They developed their own classifier and extracted 545k features using static analysis for the given application. A binary indicator feature vector is used to represent a software application. The adversarial examples are generated based on the input feature vectors using JSMA method, which is to say, the forward derivative of the trained neural network is used to estimate the perturbation direction. Perturbations based on the input instances are chosen with the maximal positive gradients to the desired class.

Direction-based Method

Hotflip [9] proposes a gradient-based algorithm to generate white-box adversarial examples on discrete text structure, so that a few perturbations can fool a character-level neural classifier with a significant decrease on accuracy. This method uses an atomic flip operation, which changes one character to another. Based on that other operations like insertion and deletion can be realized as well. With semantic constraints, this approach can be extended to a word-level attack.

They represent input text as word vectors and measure the directional derivatives along those vectors. Then the gradient pointing to the best loss-increasing directions will be utilized by attackers to choose the flip location of the input vector. Then they estimate the best character change by maximizing the first-order approximation of the change in the loss by computing directional derivative along this vector. Furthermore, an insertion in character-level at the j-th position of the i-th word can be considered as flip operations as well, which can be realized by shifting characters via multiple flips from the insertion position until the end of the word.
HotFlip increases attack efficiency greatly by saving a lot of computational costs. Unlike a naïve loss-based method, it requires exhausting computations on the loss in-

duced by any possible changes. In contrast, HotFlip only ask for one round forward pass plus a backward pass for gradient computations.

The work [19] extended HotFlip on machine translation systems by adding targeted attacks. They proposed an extra controlled attack, as well a targeted attack, aiming to remove a certain word in the output and to replace a certain word with a selected one, respectively. Their work shows that white-box adversaries can be comparable stronger than black-box adversary attacks and after adversarial training, the model's robustness can be improved significantly.

Besides the mentioned attack methods, CW-based [39], attention-based [5], and other hybrid approaches [12] are developed under white-box requirements. Due to the substantially higher effectiveness and comparable performance of the gradient-based approach, we apply HotFlip on our neural ranking models for a local adversarial ranking attack. More details about our methodology are introduced in chapter 3.

2.1.2.2 Black-box Attack

Unlike white-box attacks, which require information of the model structures and parameters, black-box attacks have generally access to model input and output. Due to the stricter constraints, black-box approaches tend to craft adversarial examples based on heuristics. Nevertheless, it makes such methods more feasible for production deployment, as in real-life cases, the attackers will be confronted with black-box applications.

[20] is a pioneering work in attacking Reading Comprehension (RC) systems. They proposed concatenation adversaries on Stanford Question Answering Dataset (SQuAD), which is to append automatically generated distracting adversarial examples to the end of paragraphs with the question and the answer unchanged. To be specific, they proposed two variants, adding elaborately designed grammatical sentences that look similar to the question, and as well adding arbitrary sequence from a pool of 29 selected distracting words, namely AddSent and AddAny respectively. Their results indicated that ungrammatical sequences could dramatically decrease accuracy compared to grammatical ones.

[44] extends the work by joining variable positions where to place the adversarial examples, and expanding the fake answer set, which contributes to diverse distracting sentences. The new augmentative adversarial examples support training new robust neural models.

In addition to concatenation adversarial attack like [20], there are manifold black-box attack methods like edit-based adversaries, where perturbations are generated by edit operations such as insertion, replacement and deletion in word-level or character-level, and paraphrase-based adversaries, where paraphrases are generated under constraints or rules, as well GAN-based adversaries where Generative Adversarial Network (GAN) [13] is utilized to craft adversarial examples.

In a summary, user-defined heuristic rules are required under black-box attack to generate natural (in human perspective) substitutes of words or sentences, while white-box attack often obtain a promising effectiveness on account of full access to model parameters. Due to the discrete nature of human languages, gradient-based approaches are efficient yet lack of fluency compared to rule-based methods. Our work is based on a gradient-based white-box attack strategy due to the potential comparable attack performance, therefore, we omit full-scale introduction to the black-box method here. For more detailed information, we turn readers to a comprehensive survey [45].

2.2 Universal Attacks in NLP

The 'universal adversary' idea comes from [27, 7], which proposes an algorithm to generate universal image-agnostic perturbations for state-of-the-art DNN models. Due to the continuity of image pixels values, these minor perturbations are not easily detectable by the human eyes. However, due to the discrete nature of the text, this method cannot be directly transposed from images processing. Inspired by this approach, [43] applies a gradient-based white-box method that iteratively generates a set of universal token sequences that can be added to any input of the dataset and will trigger the model to give the false prediction.

Specifically, the adversary optimizes the universal triggers to minimize the loss function of concatenated text input (adversarial examples) and the targeted label on all the dataset input. They initialize the trigger words in the front of the text input and iteratively replace the trigger sequence in word-embedding level over the whole dataset by minimizing a first-order Taylor approximation of the loss. They also expand the strategy with beam search. After finding the converged word embeddings, they convert the embeddings to tokens to form the final universal triggers.

The universal trigger method [43] has been applied and studied on Text classification, Reading Comprehension and Conditional Text Generation respectively.

For Text classification tasks, two downstream tasks Sentiment Analysis and Natural Language Inference are conducted on attack research. The study shows that using binary Stanford Sentiment Treebank as dataset, the correctness of a Bi-LSTM model [16] with Word2Vec [26] embedding on positive examples after the attack with the token length of three can drop down from 86.2% to 29.1%. The accuracy of ELMo-based [31] model also decreases from about 90% to 50% after four characters attacks. When simply prepending a trigger token to the SNLI [6] hypotheses, such as the word 'nobody' to the hypothesis with ground truth of Entailment, the GloVe-based DA [29] model can even cause 99.43% of instances to be predicted as Contradiction. For the Reading Comprehension task, especially SQuAD, they added a trigger sequence containing the target answer to the front of each sample answer paragraph, aiming to cause the model to predict the target answer as the answer. The results showed that the correctness rate of exactly matching the target answer varied greatly for different models, but was generally below 50%. For the Conditional Text Generation task, they used the GPT-2 model to generate six non-offensive tokens that could trigger the model to generate a racially discriminatory passage.

[43] extended this method from image attacks [27] and applied it to different NLP tasks, successfully demonstrating that a very short token sequence can fool the model to make false predictions. They also suggest that the trigger tokens generated for some specific models have transferability to some extent, such as the triggers generated on attacks for SQUAD on a GloVe model also have good attack effects on ELMo-based models. The analysis also yields some reasonable explanations for the model behavior, such as the prediction of the SQUAD model relies heavily on the vocabulary around the answer span, and the information in the question also plays an important role.

Moreover,Wallace et al.[41] also argue it's possible to plant any trigger by poisoning the training data with a small set of crafted instances containing such trigger. After retraining, the model will pick up the shortcuts induced by the trigger and make the same prediction whenever the trigger occurs in the input.

Other parallel work like [2] as well performs universal adversarial attacks on text classification, and [35] concentrates on generating model-agnostic paraphrases using back-translation, whose Semantically Equivalent Adversarial Rules (SEARs), however, limit the practicality of the method.

We are more interested in this universal trigger because it exposes higher potential risks if a single phrase can not only impact the model prediction, but also apply to the whole dataset. However, currently, there is little research on attacks against ranking model, and there is no experiment to verify whether the universal adversarial attack is effective for the ranking model. In addition, [43] has not tried to analyze local adversarial tokens on whether they have a similar global attacking capability. With these questions in mind, we apply universal adversarial attacks to the ranking model and try to investigate the model's behavior on whether the model presents bias towards a dataset by comparing local adversarial tokens with global ones.

2.3 Adversarial Attacks in Text Ranker

There has been limited research regarding adversarial attacks on ranking models, especially text ranking models.

Closest to our work is [32], who propose a model-agnostic document-perturbation procedure for rank demotions on top-ranked documents. Given a ranking model f, a query q and a top-ranked document d to the query, where q and d are both represented as multi-dimensional word embeddings, they aim to search for a noise vector with which the relevance score of the query-perturbed document pair will be decreased with changing few terms in the document. They minimizes the relevance score of the perturbed document while they also ensures the penalty only on changing query terms in the document, so that constraint would be not strict. A parameter c restricts the number of changed terms in the document.
We can observe from their results that one-word change in the document can be imperceptible, however, the attack effectiveness is not satisfying with only 4 rank drops in some cases.

Our work is different in that:

1. we use gradient-based method to search candidate tokens to replace and be replaced;
2. we consider more scenarios such as document demotion and promotion for transformer models;
3. more importantly, we focus on local as well as global adversarial tokens as an attempt to discover potential dataset and model biases.

2.4 Explaining Ranking Models

In addition to the attack methods mentioned above, this experiment also incorporates the interpretation results of the ranking model into the generating of adversarial examples from the perspective of the interpretation model for the first time, so that the correctness of the interpretation method and the validity of the interpreted words can be verified in reverse. Inspired by [38, 36, 37], when interpreting IR, although people are always interested in the ranking order of the document candidates, it is also possible to think from another perspective whether the ranking model really understands the meaning of the query specified by the customer, in other words, we concern about what is the actual query intent as understood by the model, and whether there is a way represent the query intent.

[37] proposed a model-agnostic method to interpret the text ranking model by approximating a complicated usually a neural ranking model by a simple mimic model in the term space with expansion terms. Those expansion terms are considered as intent understandings of the queries by the simple model, and hence they represent the interpretation for the complicated model.

They firstly extract candidate terms from the ranking results of the agnostic model and identify document preference pairs. Then they construct a preference matrix using candidate terms. This is a data structure to store the influence of each candidate term in preserving pairwise document preferences given a simple mimic model. Finally, they choose a set of terms according to the matrix that maximizes coverage of preference pairs. The explanation is composed of the selected term set and the simple model that intend to mimic the rank order generated by the complicated model.

Based on [37] with only one simple ranking model, multiple simple ranking models can be extended trying to locally approximate a more complicated usually a neural network ranking model by extracting expansion terms for a query.

Unlike existing methods for interpreting ranking models [34], this explainer method [37] uses multiple interpreters to effectively avoid the phenomenon of overfitting (trapped in locality) induced by a single interpreter. These simple explainers can be, for example, a term matching model, which evaluate with bm25 score of a term in a document; a position aware model [11] to calculate whether the position of the term in the document is important; or GloVe embedding [30] to compute the cosine similarity between term and document. The results of the three interpreters are combined to determine the final intent expansion.

However, the output of this method is a set of expansion terms for a specific query, and it is not possible to obtain the intent terms for all queries. Therefore, it is only meaningful to compare the results with our local adversarial ranking attack in rank promotion scenario. The idea is to select the same length of expansion terms instead of adversarial tokens and add them to the front of the attacked document as a black-box method to compare with the gradient-based method which is a white-box method. The comparison result is presented in **??**

3. Adversarial Attacks on Neural Rankers

We operate on the common retrieve and re-rank framework for document ranking and focus on adversarial perturbations in the re-ranking phase. Our aim is to generate adversarial examples for already trained neural ranking models by perturbing retrieved documents at user-specified positions in the document. We consider two attack scenarios: (1) **rank demotion** for a highest-ranked document, and (2) **rank promotion** for a bottom-ranked document. The scope of an adversarial ranking attack can be either local (on a given query) or global (on an entire query workload). Towards this, we firstly envision the **local ranking attack** where we intend to perturb a retrieved document given a query. Secondly, we formulate the **global ranking attack** where we intend to generate adversarial tokens for an entire workload of queries. We describe our approach to realize both these attacks in detail in the following.

3.1 Local Ranking Attack

This section introduces the implementation of local adversarial attack methods in terms of how to generate and replace adversarial tokens.

3.1.1 Problem Statement

We focus on attacks that concatenate a token sequence to the front of a document and result in a shift of its ranking, to be specific, a demotion on the ranking for the top-ranked document and a promotion on the ranking for the last-ranked document. We start from local attack on each independent query.

For a local attack task, we are given a ranking model f, a document d to be attacked (more specifically, top-ranked or the last-ranked) from a pre-retrieved ranking list to a certain query q, an initialized token sequence d_{adv} with fixed token length. The adversarial attack aims to prepend adversarial tokens d_{adv} to the document d, so that the relevance score $f(d_{adv}, d)$ decreases for rank demotion attack or increases for rank promotion attack.which are translated to the objectives below:

For rank demotion with attack on the top-ranked document:

$$\arg\min_{d_{adv}} \mathcal{L}(f(d + d_{adv})) \tag{3.1}$$

For rank promotion with attack on the last-ranked document:

$$\arg\max_{d_{adv}} \mathcal{L}(f(d + d_{adv})) \tag{3.2}$$

Where $\mathcal{D}$ are token instances of candidate documents from a data distribution and $\mathcal{L}$ is the loss function, in our case, the relevance score itself, which differs from [43].

3.1.2 Adversarial Token Search Algorithm

Algorithm 1 illustrates the overall processes of performing adversarial ranking attack, which involves an initialization of random selected tokens following by the adversarial token search procedure.

Adversarial Token Initializing Considering the characteristics of the BERT [8] model, the following issues need to be taken into account.

1. *Adversarial token length*: Bert [8] model has a default limited input token length of 512, which is the maximum sequence length that the model might ever be used with, meaning that the part that beyond this boundary will be truncated. We use 512 tokens as maximum position embeddings. Most queries have a short token length under 5. We set as well 5 as the default adversarial token length. Later we also did ablation experiments to study the effect of token length in section 5.4.

2. *Adversarial token position*: We assume that position also has an effect on the attack performance. Hypothetically, a token sequence in a preceding position would deliver a stronger attack ability than one in a posterior position. We start our experiment by concatenating 5 adversarial tokens in front of the attacked document.

3. *Initializing tokens*: To reduce errors induced by randomness, we initialize adversarial tokens by repeating the special token [MASK] 5 times, that is, a sequence of [MASK] [MASK] [MASK] [MASK] [MASK]. Besides, we randomly choose 5 tokens from BERT vocabulary which has 30522 word embeddings in total, but skip the first 998 special tokens, where a lot of reserved words [unused*] are stored. They will be randomly initialized if not specified and turn out to possess relatively larger embedding norms than the remaining token embeddings. The

Algorithm 1 Adversarial ranking attack

Inputs: a query q, a ranking model f, a highly-relevant or non-relevant document d
Output: an adversarial token sequence d_{adv}

```
1: procedure ADVERSARIAL TOKEN SEARCH($q, f, d$)
2:     Initialization: $d_{adv} \leftarrow$ random selected tokens
3:     while $\Delta f(d_{adv}; d) \neq 0$ do
4:         Compute the average gradient $g_{d_{adv}}$ w.r.t. the adversarial tokens $d_{adv}$
5:         Obtain token candidates $\mathcal{C}_{adv}$ by Hotflip[9] using $g_{d_{adv}}$
6:         Determine token sequence $d_{adv}$ by beam search among $\mathcal{C}_{adv}$
7:     end while
8: end procedure
```

experiment result indicates that the adversarial tokens are more likely to converge to these reserved tokens, which contributes a lot to the result uncertainty because of random initializing together with difficulty in the analysis due to loss of semantic meanings. As a result, we skip these words at initializing and token replacement process.

Adversarial Token Replacement Strategy Figure 2 and 3 illustrate adversarial token replacement strategy for local and global attack on rank demotion scenario respectively. We iteratively replace adversarial tokens until the relevance score won't decrease for rank demotion attack, or won't increase for rank promotion attack. The attacking method is inspired by [9] but developed to word-level. Hotflip [9] is a gradient-based attacking method introduced in section 2.2.1.3. It computes the change in loss after replacing a single character with another in a word by directional derivatives of this operation. Then the best character change is obtained by simply choosing the word vector that results in the biggest increase in loss. Compared to the loss-based method, the gradient-based method only requires one forward pass and one backward pass to get gradients, which saves many computing costs for calculating the exact loss caused by every possible change.

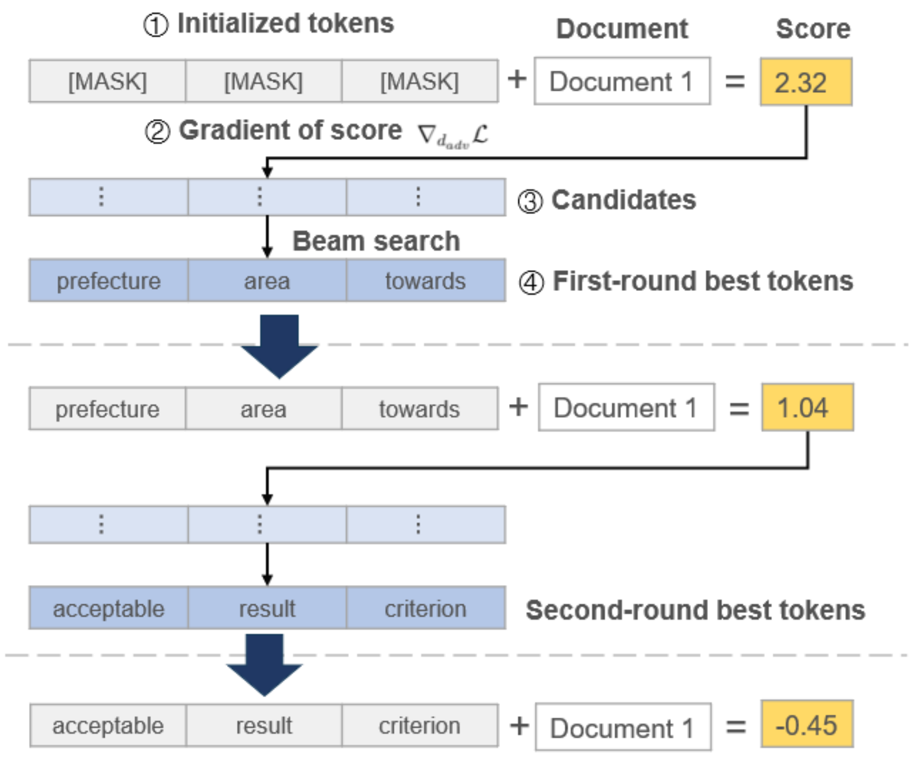
In our case, we operate on word level rather than character level. For every adversarial token, we seek for a new one that would minimize loss' gradient for rank demotion attack, while one that would maximize loss' gradient for rank promotion attack.

For rank demotion attack:

$$\underset{e_i \in \mathcal{V}}{\text{arg min}} \, [e_i - e_{adv_i}]^T \nabla_{e_{adv_i}} \mathcal{L} \tag{3.3}$$

For rank promotion attack:

$$\underset{e_i \in \mathcal{V}}{\text{arg max}} \, [e_i - e_{adv_i}]^T \nabla_{e_{adv_i}} \mathcal{L} \tag{3.4}$$



① Initialized tokens
Document
Score
[MASK] [MASK] [MASK] + Document 1 = 2.32
② Gradient of score $\nabla_{d_{adv}}\mathcal{L}$
③ Candidates
Beam search
prefecture area towards
④ First-round best tokens
prefecture area towards + Document 1 = 1.04
acceptable result criterion
Second-round best tokens
acceptable result criterion + Document 1 = -0.45

Figure 2: An illustration of adversarial token replacement strategy for local attack on rank demotion scenario.

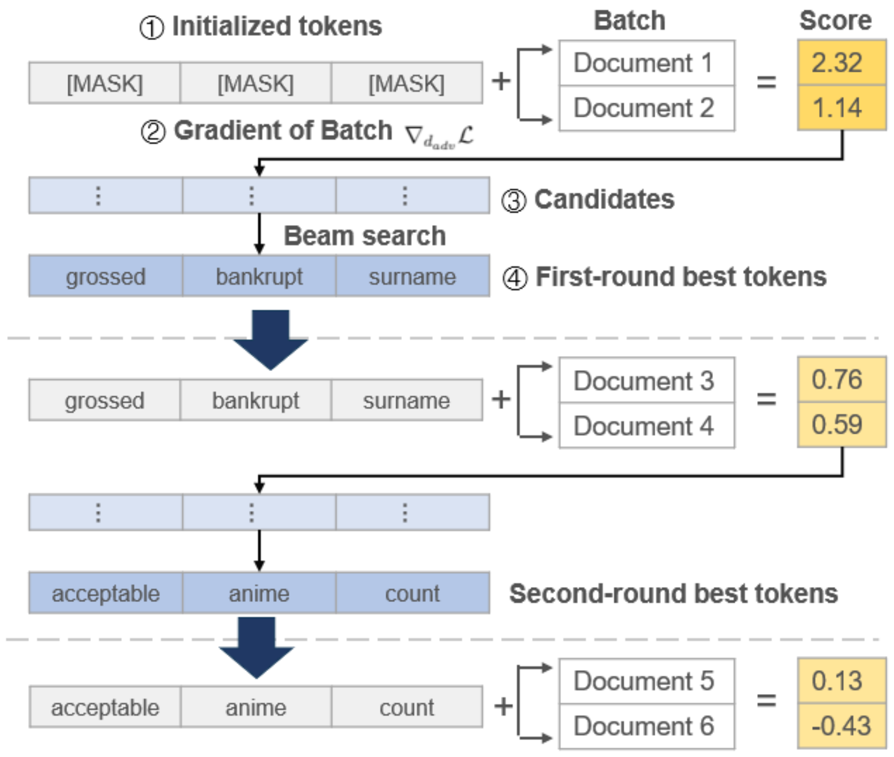


Figure 3: An illustration of adversarial token replacement strategy for global attack on rank demotion scenario.

where $\mathcal{V}$ is all words' embeddings of the model vocabulary, which can be found in BERT word embedding layer, e_{adv_i} denotes each adversarial token embedding, $\nabla_{e_{adv_i}}\mathcal{L}$ denotes average loss gradients of these embeddings. Both Hotflip and our adversarial attack method require white-box access to the model.

More specifically, we first register a hook on the word embedding layer. When backward pass is called, loss gradients with respect to the token embeddings are obtained. We then select out the adversarial token gradients, and average them if a batch size is specified in global attack.

To obtain the optimal token embeddings, we apply brute-force dot products with the vocabulary embedding matrix which can be simply realized by using Einstein summation convention. With a development of approach Hotflip, we then keep some adversarial token candidates for further optimization.

Beam Search We followed [43] to develop adversarial token update approach with beam search. From equation 3.3 or 3.4 we can get a list of candidates for each adversarial token (in our case 5 tokens) by choosing the top-k (for rank demotion attack) or last-k (for rank promotion attack) tokens from the dot product result.

Given the candidate list for every adversarial token, we then perform beach search in a left to right fashion to find the best new candidate token. More specifically, we start with finding the best candidate token for position 0 by calculating the relevance score of replacing adversarial token 0 with each respective candidate token. For all beams, we then try all candidates in the next position. When all candidates in all beams are searched, the token sequences now become the best tokens in this searching round. We update the whole token sequences until the relevance score stays unchanged. Because of computational constraints, we perform with a beam size of 1, which means we select the best candidate in each replacement process.

3.2 Global Ranking Attack

Rather than generating adversarial tokens for a particular query-document instance, in global ranking attacks, we aim to find tokens d_{adv} that are adversarial to the entire query set. In other words, adding such tokens can promote or demote the respective highest or lowest ranked document for any query in the dataset. Towards this, we minimize the expected relevance score for all queries as:

$$\arg\min_{d_{adv}} \mathbb{E}_{(q,d)\sim\mathcal{D}}[\mathcal{L}(f(d + d_{adv}))] \tag{3.5}$$

or we maximize the expected relevance score for all queries as:

$$\arg\max_{d_{adv}} \mathbb{E}_{(q,d)\sim\mathcal{D}}[\mathcal{L}(f(d + d_{adv}))] \tag{3.6}$$

The global adversarial tokens are selected by using the same gradient-based search strategy as equation (3.3) and (3.4) and updating on all queries until the average relevance score decreases or increases no more.

4. Experimental Setup

This chapter describes the basic experimental setup, including the datasets used for the attack tasks, the victim BERT-based model, and the metric used to measure the attack performance.

4.1 Datasets

ClueWeb09 (category B) The ClueWe09 dataset is a dataset often used in NLP and IR research. It was collected in January and February 2009, and contains the content of 1 billion web pages, covering 10 languages. the Clueweb09 dataset is divided into category A and Category B, the latter containing the first English segment of A, which is 50 million web pages. It is worth noting that this section of B contains a complete backup of the English version of Wikipedia.

We use Category B for ad-hoc retrieval. Category B contains 200 queries, which we assign to train/dev/test in a 3:1:1 ratio in the following experiments, i.e., 120/40/40 queries. We initially apply the global adversarial ranking attack experiments to the test set, and then extend it to the entire 200 queries as an ablation experiment.

TREC-DL TREC-DL 2019 passage ranking dataset is constructed from MsMarco dataset. It has a fairly large training set of human-generated labels, where the label directly identifies whether the passage can be used to answer this query.

Since the test set of the passage ranking dataset has only 200 queries, to verify that the local adversarial tokens have a global attack effect in general, we randomly selected 1000 queries from the dev set and filtered out queries with less than 90 documents. (Note that in our experiments we concentrate on ClueWeb09, from whose test set we found that local adversarial tokens have global effect. To check the generality, further experiment was performed on TREC-DL)

4.2 Ranking Model

Since the appearance of the BERT [8] model, many NLP downstream tasks have gained surprising improvements. In our study, we focus on attacking the DNN model with a pre-trained bert-base-uncased model, and then train it using ClueWeb09 dataset. To adapt the model to match our ranking task, we add a feed-forward layer after the pooled representation, and optimize the margin loss between positive-negative document pairs. The output embedding from CLS is indicated as relevance score between a query and a document, which is used to measure the degree of matching and later is compared to calculate normalized rank shift.

Since the purpose of this study is to observe the global effect of the local adversarial tokens on the ranking model and to analyze the behavior of the BERT [8] model, the experiment currently uses only the BERT model as the attacked object.
For a query, the ranking model will predict a relevance score for every document in the candidate list, however, to evaluate relative rank shift of the attacked document, reranking all the other documents is unnecessary. Hence reranking on the attacked document and a following comparison with the saved record of other candidates can be more efficient. We use this cost-saving strategy both on local adversarial attack and global adversarial attack.

The whole input to a BERT [8] model should be a single sequence. When there are multiple inputs like in our case, query along with the attacked document, BERT [8] use special tokens [CLS] and [SEP] to understand its composition, where [CLS] stands for classification and is added at the beginning of the input, it represents the meaning of the entire sentence, and it is introduced as a tag for classification in sentence level. [SEP] stands for separation, which helps the model to understand which Sentence in the input belongs to which part. In our case, we concatenate adversarial tokens in the front of the document, which would be illustrated as:

$$[CLS][q_1, \cdots, q_n][SEP][d_{adv1}, \ldots, d_{advl}][d_1, \cdots, d_m][SEP]. \tag{4.1}$$

Where $[q_1, \cdots, q_n]$ denotes query tokens of length n, $[d_{adv1}, \ldots, d_{advl}]$ adversarial tokens of length l, $[d_1, \cdots, d_m]$ document tokens with truncated length m, which should satisfy $n + l + m + 3 <= 512$. Note that we need to take into account special tokens [CLS] and [SEP] when we record adversarial token positions.

It is also worth noting that different queries may have different lengths after tokenization. When attacking in batch, it is vital to record the start position of token sequence for each instance, which later also comes into play when the average gradient is obtained.

4.3 Metric

We measure the absolute ranking changes normalized by the number of ranking documents as the metric to evaluate attacking performance and name it normalized rank shift. Therefore, for both rank demotion and rank promotion scenarios, the higher the normalized rank shift, the better the attack effectiveness. We define normalized ranking shift as :

$$\frac{|\Pi_d - \Pi_{d_{adv}+d}|}{|\Pi|} \tag{4.2}$$

where Π_d denotes the ranking position of document d obtained by ranking all relevance scores from the top-k document list to a specific query q, $d_{adv} + d$ indicates the document with adversarial tokens, and $|\Pi|$ indicates retrievel depth in our case equals to 100.

5. Evaluation Results

The previous chapter introduces the model, datasets, and metrics for evaluation. In this chapter, we evaluate attacking effectiveness under different datasets and analyze the obtained adversarial tokens. We start to analyze local adversarial tokens for the independent queries obtained from the attack on rank demotion task and rank promotion task. According to the potential bias we found on different datasets, we then proposed hypotheses on its existence in section 5.3. We also include exploration on word embedding characteristics of adversarial tokens in terms of norm distance, cosine similarity and visualization in 2D in discussion.

5.1 Attack Effectiveness

This section first introduces the overall attack efficiency on ClueWeb09 and TREC-DL and then reports the effect of adversarial token positions and lengths on the attack effectiveness.

5.1.1 Overall Attack Effectiveness

In this subsection, we report the overall effectiveness of the gradient-based attack method against BERT text ranker model on both the ClueWeb09 and TREC-DL datasets. Our results indicate that our attack method using only a short adversarial token sequence can lead to a significant upgrading or degrading of a document's ranking score. Figure 4 demonstrates the average normalized rank shifts of global and local adversarial attacks on both datasets, drawing a distinction between attacks on the top-ranked document and the last-ranked document. It should be noted here that we select 5 tokens and 3 tokens for documents in ClueWeb09 and TREC-DL respectively. Random attack records are added as a baseline.

We set the adversarial tokens to be concatenated in the front of the document for all attacks. Due to the difference in document length, we specifically split attack results between two datasets, setting 5 tokens for ClueWeb09 documents that usually contain long texts (more than 512 tokens), while 3 tokens for relatively shorter TREC-DL documents.

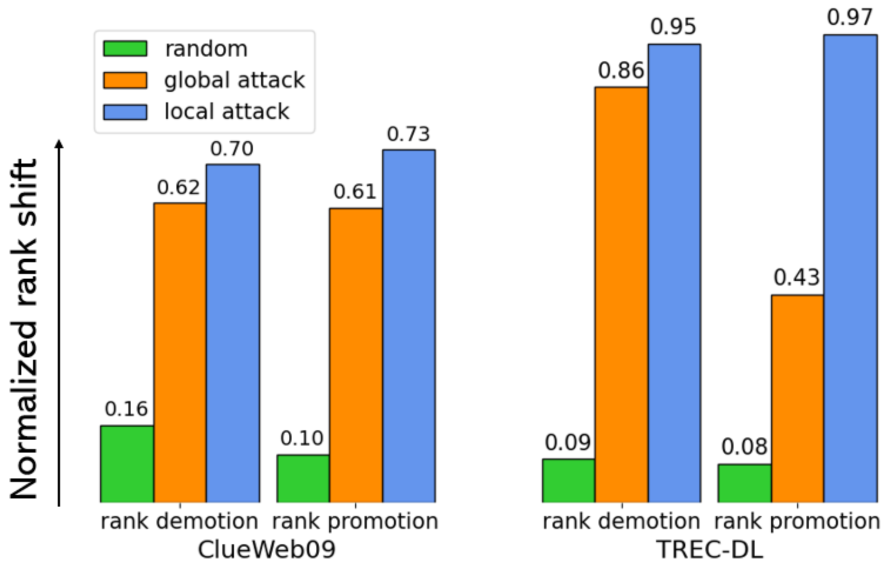


Figure 4: Overall attack effectiveness on ClueWeb09 and TREC-DL.

Overall, shorter token sequences can successfully cause great changes in the document rankings of both datasets, opening a huge gap with the random attack result as the baseline. Local adversarial attacks always outperform global attacks, which can be interpreted as a tradeoff in attack effectiveness for a better generalization.

Local adversarial attacks on TREC-DL present a near-perfect result. An attachment of 3 tokens can cause an average rank demotion of the top 1 ranked document to the last 5 positions and moreover, an outstanding rank promotion of the last document to the top 3 positions. However global attack on the last document presents only less than half of the achievement as local attack do, indicating a poor generalization of global triggers. Meanwhile, both attack methods on ClueWeb09 can perform relatively smaller rank shifts, with around 60 and 70 differences in a global way and local way respectively. It can be caused by much longer contents in ClueWeb09 despite two more extra tokens. But it realizes a more robust generalization of global triggers, reflecting in a slight difference between attack methods regardless of rank demotion or promotion.

5.1.2 Effect of Token Positions

All previous experiments were run based on adding an adversarial token sequence at the beginning of a document. A reasonable question is what happens if we don't add them in the front of the document, and how does the performance change if we add them at other locations in the document?

Therefore we experimented with locations of the adversarial tokens insertion. We plan to add a token sequence continuously in the last position and middle position

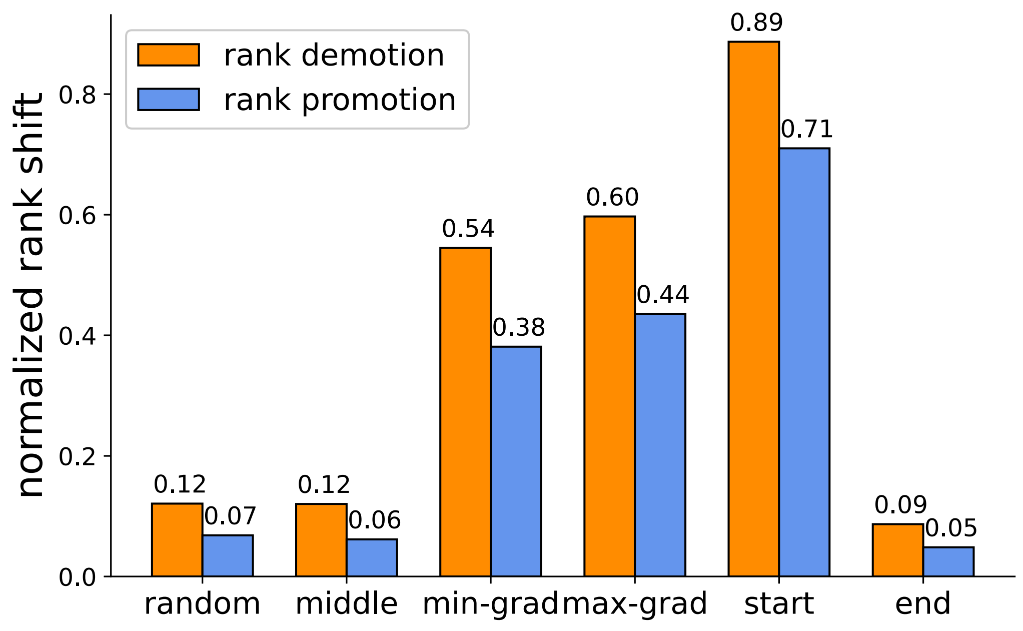


Figure 5: Effect of adversarial token positions on attack effectiveness on ClueWeb09.

of a document, both of whose exact locations are determined by the minimum value of document (middle) length and acceptable length of model input. In addition to this, we also consider random inserting positions and the positions where the original document tokens have the highest and lowest gradient scores. The following Figure 5 represents normalized rank shifts of attack on ClueWeb09 40 queries in terms of both rank demotion and rank promotion tasks. Note that tokens in random, min-grad, max-grad positions can be not in a sequence but distributed in any position in a document. Here we use 5 tokens.

Figure 5 illustrates significant position biases of Bert ranking model. The model shows the same preference order for different positions on both rank demotion and promotion tasks. Among all the positions, prepending tokens to a document causes the greatest rank difference with an average of 89 rank changes in degrading the top one text, while postpending tokens in the end creates the tiniest rank gap with less than 10 rank differences. Even though the position of the middle is half ahead of the end, the performance increase disproportionately. It has roughly the same result as random position, improving the ranking by only a few documents compared to the end position. Max-grad positions are calculated by searching the maximum token gradient scores of the original document. Those token positions are considered to be rather important to the model when applying a gradient-based attacking method. However, as a result, tokens in max-grad positions show only slight improvement than those in min-grad positions.

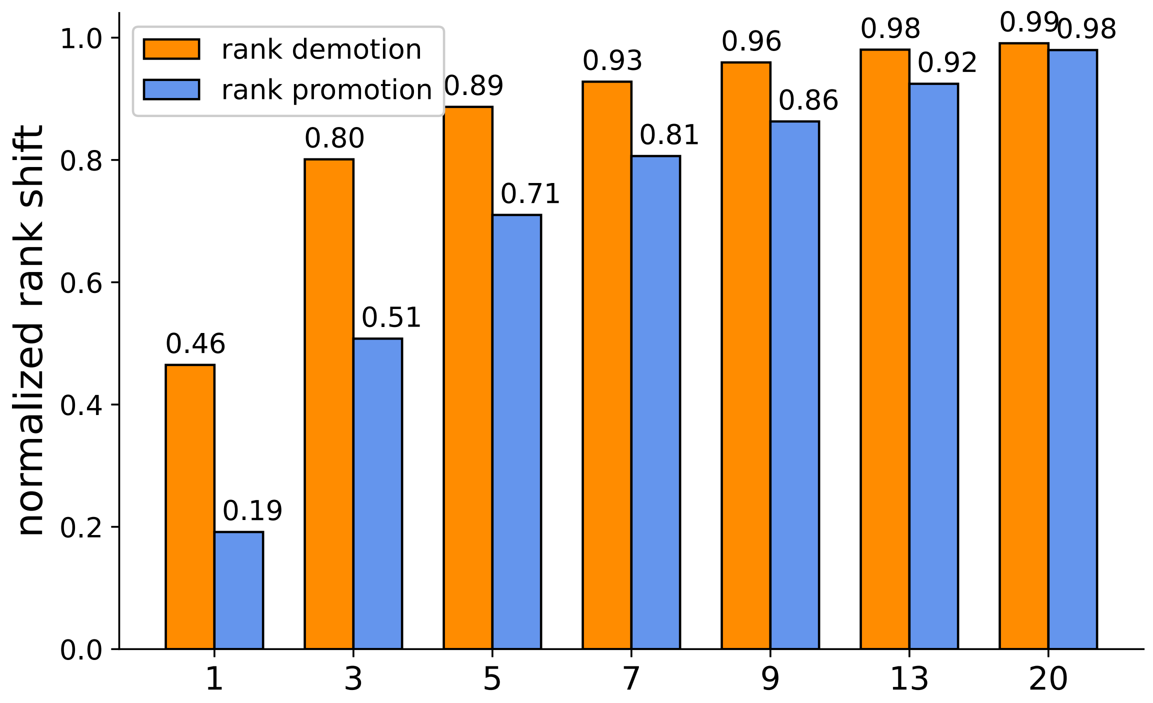


Figure 6: Effect of adversarial token sequence lengths on attack effectiveness on ClueWeb09.

In a summary, the BERT ranking model is highly biased to adversarial positions. This can be related to the input ensemble procedure to the BERT model, which concatenates a document to a query separated by a special token [SEP]. Although a self-attention model like BERT relates different positions of the same input sequence, the query-document relevance score is still highly impacted by the adjacent part. Therefore, subsequent designs should also consider how to reduce, even avoid this kind of bias.

5.1.3 Effect of Token Lengths

In addition to the previous subsection, where we discussed the impact of adversarial token positions on attack performance, it is also rational to consider the impact of token length on attack effectiveness. Figure 6 shows the average normalized rank shifts of attack on ClueWeb09 considering different adversarial token lengths. We omit experiments on TREC-DL since we assume sequence length effects should stay consistent across different datasets. Note that we fix the location of the added tokens at the beginning of a document.

As we expected, increasing the length of the token sequence does continuously improve the attack effectiveness, at the cost of a higher perceptibility associated with a longer length.

Adding tokens with a relatively short length has a very significant improvement in attack effectiveness. However, the longer the length, the less pronounced the effect of adding tokens. Here we should note that the horizontal coordinate is not linearly increasing, nor is the attack effect.

When using only one token to attack, imperceptibly it can cause the top-ranked document to drop 46 places. However, when increasing the token sequence length from 13 to 20, the average increase in ranking difference is only one place from 98 to 99. This tiny change comes at the cost of adding extra 7 tokens, which greatly increases the risk of being detected. But it also explains from another perspective that the closer the position of the beginning of a document, the greater the impact shows on query-document relevance.

In a real-life scenario, people may be more concerned about the first page of query results. Suppose people only care about the first 20 documents, then for the attack on the first ranked document, we only need to add three tokens at the beginning of the document to remove it from people's view. If someone with ulterior motives wants to improve the ranking of a document to be able to be seen by most people, in the worst case, prepending 7 tokens to the last ranked document can achieve such a purpose.

In a summary, a longer token sequence can greatly cause a ranking difference at attacking. A perfect trade-off will be gained among token sequence length and attack effectiveness by taking model input length (512 tokens in our case) into consideration. 5 tokens for rank demotion and 7 tokens for rank promotion can be practical on attack against ClueWeb09.

5.2 Adversarial Token Analysis

5.2.1 Rank Demotion

We found that there were some tokens that frequently occur even for irrelevant query topics. We are interested in the frequency of appearance of these tokens and the order in which they appear. Since we use beam search to search for the best tokens, we suspect that if some words appear more frequently and always in the top positions, it is likely to represent the model's bias.

5.2.1.1 Local Adversarial Tokens

Figure 7 shows some of the most frequently occurring tokens from all tokens found in 40 query attacks. Among them, 'acceptable' occurs most frequently, 17 times in total. In second place dot '.' is much lower, with only 5 occurrences. Tokens 'resulting' and 'results' sharing the same stem both occur on the list.

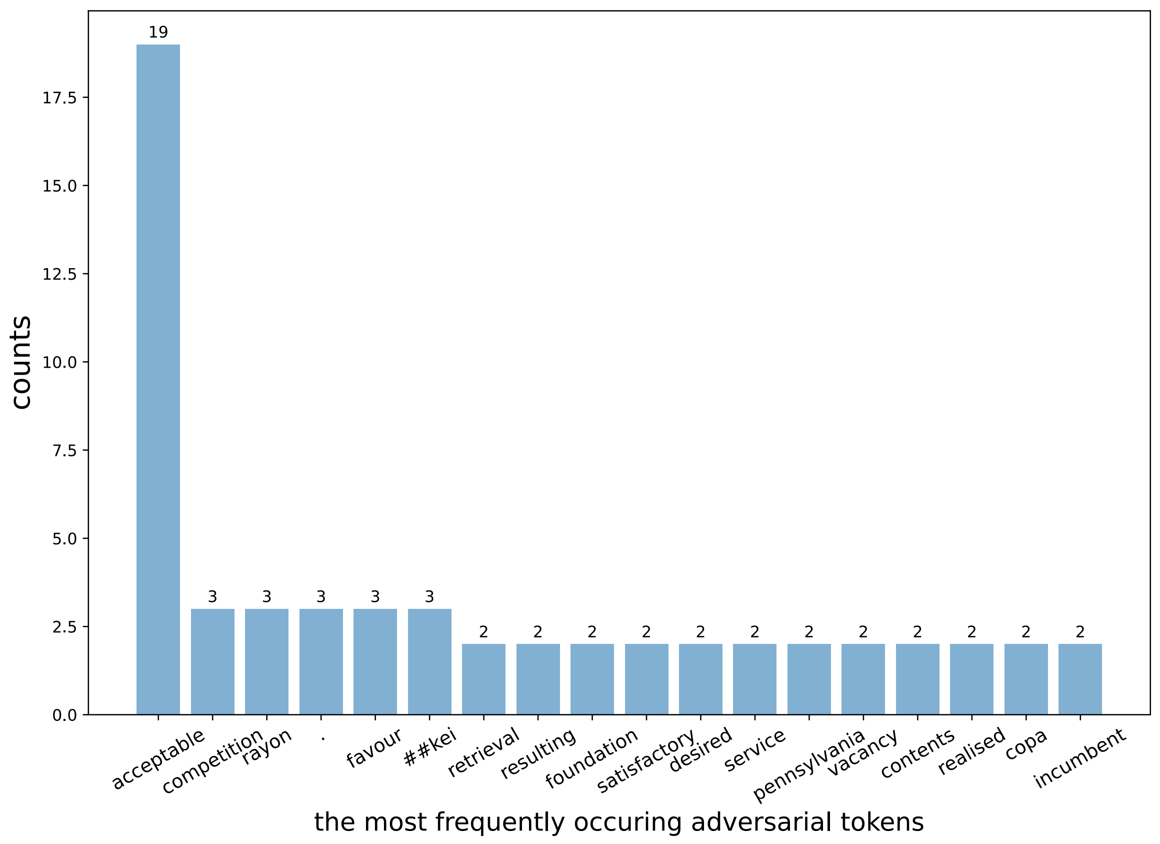
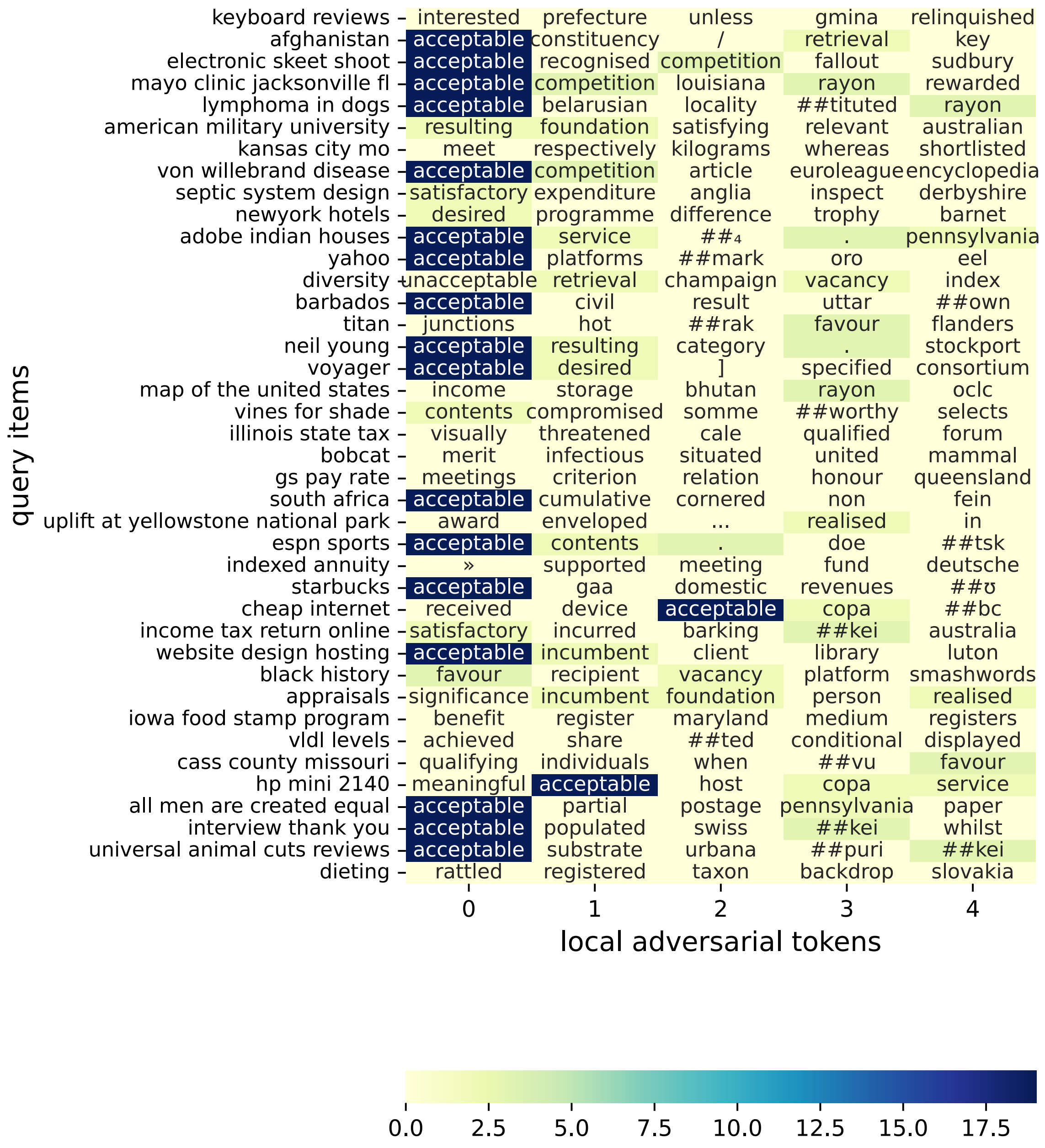


Figure 7: Counts of the most frequent adversarial tokens in descending order on rank demotion attack to ClueWeb09 test set.

The frequency map only illustrates the counts of occurrences of a token. To visualize the position of these tokens we present all the tokens corresponding to each query in a heatmap in Figure 8, highlighting frequency with progressive color palette.
In Figure 8, the horizontal coordinate indicates the token position in a sequence with 5 in length, from position 0 to 4, while the vertical coordinate indicates each query item. The color of a token cell demonstrates the frequency of the occurrence of that token. The color ranges from light yellow to dark blue, indicating less frequent to more frequent occurrences.

One interesting phenomenon we can observe is that basically all ‘acceptable’ appears in the first position, which implies that in a certain number of token candidates, ‘acceptable’ is often determined first as the adversarial token that makes the lowest prediction score (which is the aim of the attack on the top-ranked document). Quite the opposite is the case for the second-ranked tokens, they scatter all over the other positions. This result may not be quite what one would expect, and for each individual query, one might guess that they would have relatively independent tokens. But the result demonstrates that some of these local tokens also share globally, thus the local tokens have a global attack effect.

However, it is important to note that each token sequence is not related to the query



query items	0	1	2	3	4
keyboard reviews	interested	prefecture	unless	gmina	relinquished
afghanistan	acceptable	constituency	/	retrieval	key
electronic skeet shoot	acceptable	recognised	competition	fallout	sudbury
mayo clinic jacksonville fl	acceptable	competition	louisiana	rayon	rewarded
lymphoma in dogs	acceptable	belarusian	locality	##tituted	rayon
american military university	resulting	foundation	satisfying	relevant	australian
kansas city mo	meet	respectively	kilograms	whereas	shortlisted
von willebrand disease	acceptable	competition	article	euroleague	encyclopedia
septic system design	satisfactory	expenditure	anglia	inspect	derbyshire
newyork hotels	desired	programme	difference	trophy	barnet
adobe indian houses	acceptable	service	##₄	.	pennsylvania
yahoo	acceptable	platforms	##mark	oro	eel
diversity	unacceptable	retrieval	champaign	vacancy	index
barbados	acceptable	civil	result	uttar	##own
titan	junctions	hot	##rak	favour	flanders
neil young	acceptable	resulting	category	.	stockport
voyager	acceptable	desired	]	specified	consortium
map of the united states	income	storage	bhutan	rayon	oclc
vines for shade	contents	compromised	somme	##worthy	selects
illinois state tax	visually	threatened	cale	qualified	forum
bobcat	merit	infectious	situated	united	mammal
gs pay rate	meetings	criterion	relation	honour	queensland
south africa	acceptable	cumulative	cornered	non	fein
uplift at yellowstone national park	award	enveloped	...	realised	in
espn sports	acceptable	contents	.	doe	##tsk
indexed annuity	»	supported	meeting	fund	deutsche
starbucks	acceptable	gaa	domestic	revenues	##ʊ
cheap internet	received	device	acceptable	copa	##bc
income tax return online	satisfactory	incurred	barking	##kei	australia
website design hosting	acceptable	incumbent	client	library	luton
black history	favour	recipient	vacancy	platform	smashwords
appraisals	significance	incumbent	foundation	person	realised
iowa food stamp program	benefit	register	maryland	medium	registers
vldl levels	achieved	share	##ted	conditional	displayed
cass county missouri	qualifying	individuals	when	##vu	favour
hp mini 2140	meaningful	acceptable	host	copa	service
all men are created equal	acceptable	partial	postage	pennsylvania	paper
interview thank you	acceptable	populated	swiss	##kei	whilst
universal animal cuts reviews	acceptable	substrate	urbana	##puri	##kei
dieting	rattled	registered	taxon	backdrop	slovakia

Figure 8: Adversarial tokens for local attack on rank demotion scenario. Specific tokens frequently occurred.

itself. And those tokens found are not strongly related to each other semantically nor lexically. One explanation for this might be that here we are attacking in the hope that the top-ranked document will be scored lower with a concatenation of these tokens, implying that the model does not prefer such words. Tokens found are supposed to be irrelevant to the query or document, otherwise, they will benefit the document's prediction score. But based on the known training dataset, it is difficult to explain what the model does not like and why it is these words but not the others. In future studies, we intend to conduct more experiments to explain this phenomenon.

To evaluate the effects of these local adversarial tokens, we also add experimental groups in order to verify whether prepending tokens that appear too frequently is more effective on rank demotion attacks: we collect sequences of tokens that contain the first, the second, the third, and the fourth most frequently occurring tokens respectively (they are 'acceptable', 'competition', 'rayon', '.' in this case), and attack with them again on all queries and obtain the corresponding average values, which are shown in order from dark blue to light green. Furthermore, we empathize the importance of the most frequently occurring token (in our case 'acceptable') by repeating it 5 times to form a new token sequence highlighting in yellow, and join together the five most frequently occurring tokens in a sequence marking in orange. We also take local attack results as upper bound highlighting in grey and take attack results from 5 randomly chosen tokens as lower bound downplaying in cream.

As a result, we find from Figure 9 that all attack performances are between upper (89 rank shifts) and lower bound (23 rank shifts), illustrating the validity of using the most frequently occurring tokens as an adversarial token sequence. Sequences including the first frequent token 'acceptable' do not always perform better in terms of attack effectiveness than the others. However, it is interesting to notice that the performance of repeating 'acceptable' for 5 times is almost worse than any found local token sequences with only 50 in average rank shift. The sequence with 5 most frequent tokens in order to the contrary achieves a pretty good performance with an average 83 rank difference that is the closest to the best one.

5.2.1.2 Global Adversarial Tokens

Since subsection 5.2.1 points out the fact that local adversarial tokens have global properties, we are now curious about whether the global adversarial tokens obtained by using batch overlap with the local tokens. Hence under the same set of conditions (the same token sequence length, the same adversarial token searching strategy), we perform a gloabl adversarial ranking attack for 40 queries with a batch size of 4. But in this time, we only repeat the attack with 2 initializations, one with [MASK] * 5, the other with 5 randomly selected tokens from the vocabulary.

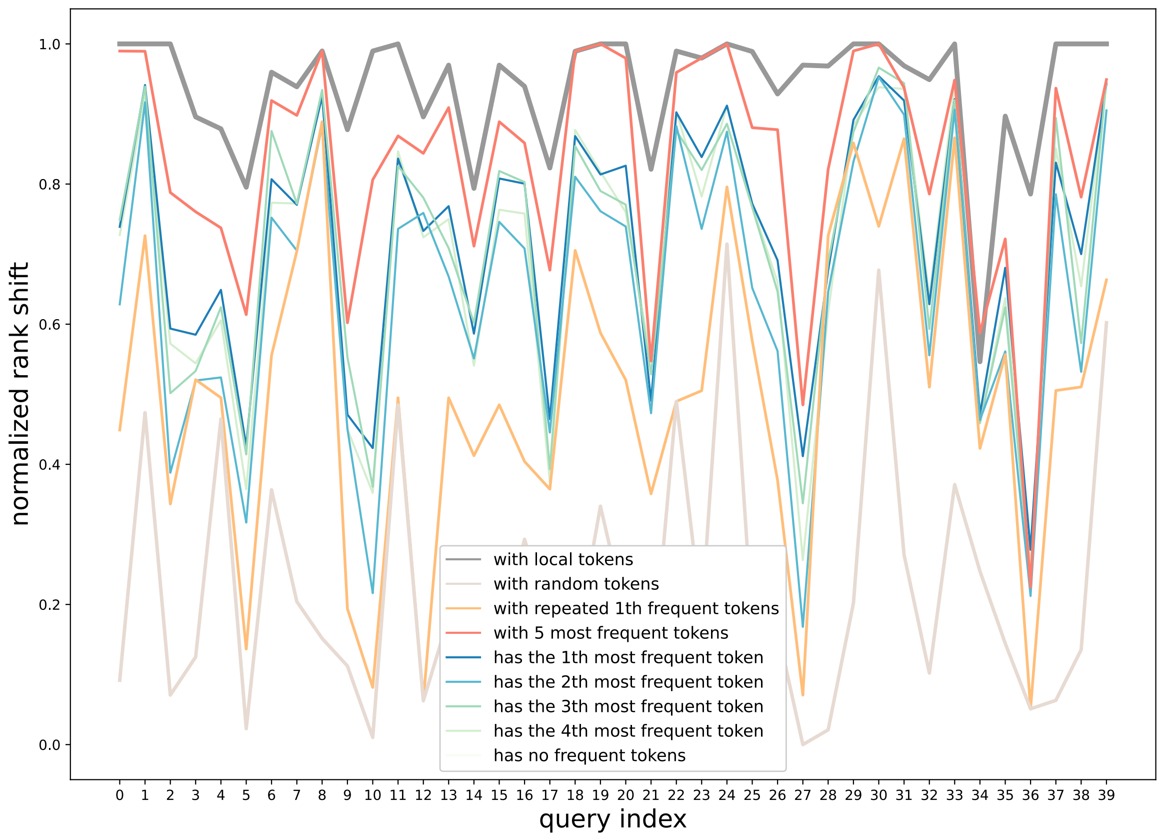


Figure 9: Normalized rank shifts on rank demotion attack using the most frequently occurring local adversarial tokens.

After using the same analysis methods and perspectives, it was found at the word embedding level, the same consistency as in Subsection 5.3.3.1 in terms of norm distance, cosine similarity. If one looks closely at the two sets of global adversarial tokens:

1. ‘acceptable retrieval conditional afi. . . ’
2. ‘duly deposited desired angliabang’

They both appear in the local tokens, and ’acceptable’ occurs again in the first place. In addition, ‘retrieval’ is also a frequent token among local tokens with frequency of 2. Therefore, the global characteristic embodied in the local tokens is certified by the global adversarial tokens.

To further confirm the effectiveness of global attack, we use these two sets of global adversarial tokens to attack the top-ranked document of all 40 queries, and the results are shown in the Figure 10.
The grey line indicates the best performance an attack can locally achieve for each query, and oppositely the cream line shows the randomly chosen tokens’ performance. The orange line indicates the result of attacking with the first global adversarial token sequence ‘acceptable retrieval conditional afi. . . ’, while the purple line

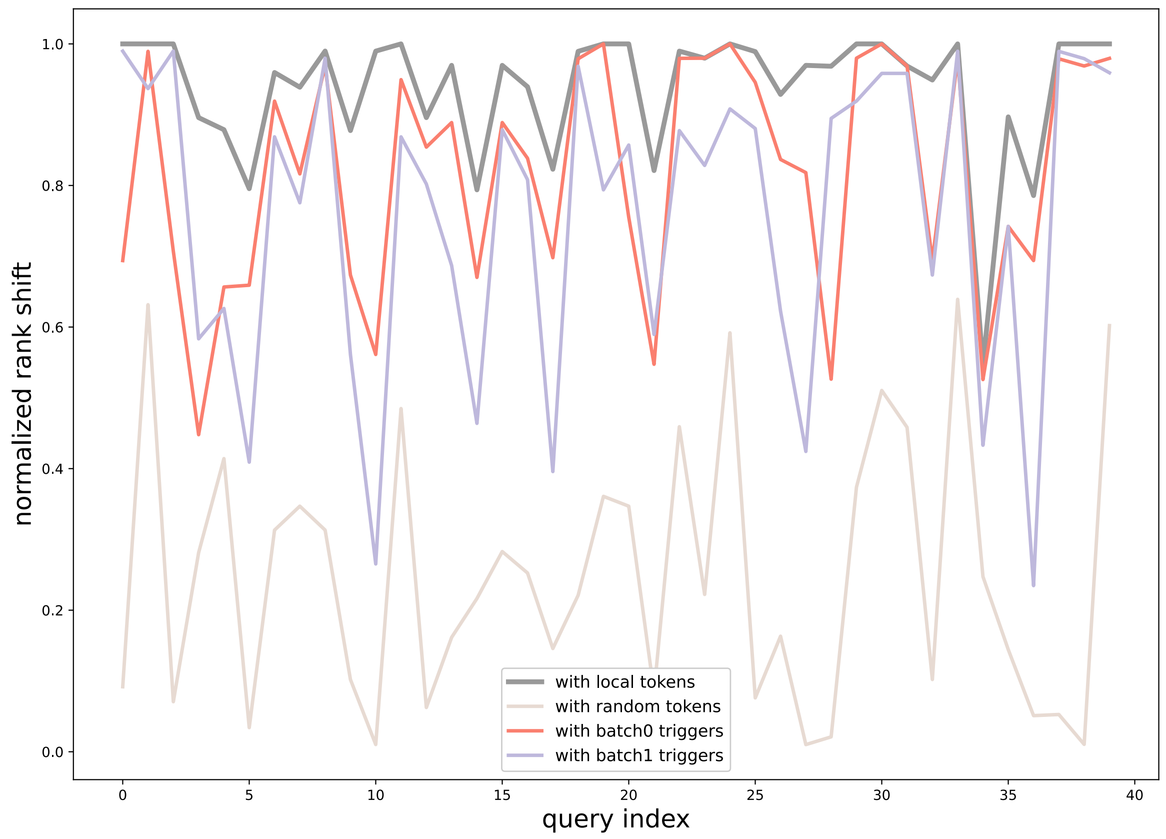


Figure 10: Normalized rank shifts on rank demotion attack using global adversarial tokens generated by attacking in batch.

with the second global token sequence ‘duly deposited desired angliabang’. It can be observed that both sets of global adversarial tokens are effective in most attacks, specifically achieving remarkable successes with 82 and 76 rank differences respectively.

But for specific queries, such as query index 3 and index 17, the attacks are much less effective than attacking with their respective best local tokens. The tradeoff in the generalization of global adversarial tokens also implies local failure in some cases.

5.2.2 Rank Promotion

Since the attack on rank demotion and rank promotion scenarios differ in the determination of the loss function, the way the attack obtains the candidates, and the termination conditions of the searching iteration, we also attacked the last-ranked document for each query, applying the same analysis as the rank demotion attack, and found similar patterns in the word embedding level and proved the global effect of the local adversarial tokens. However, when observing the tokens themself, we found some conclusions and model bias that differ from the rank demotion attack.

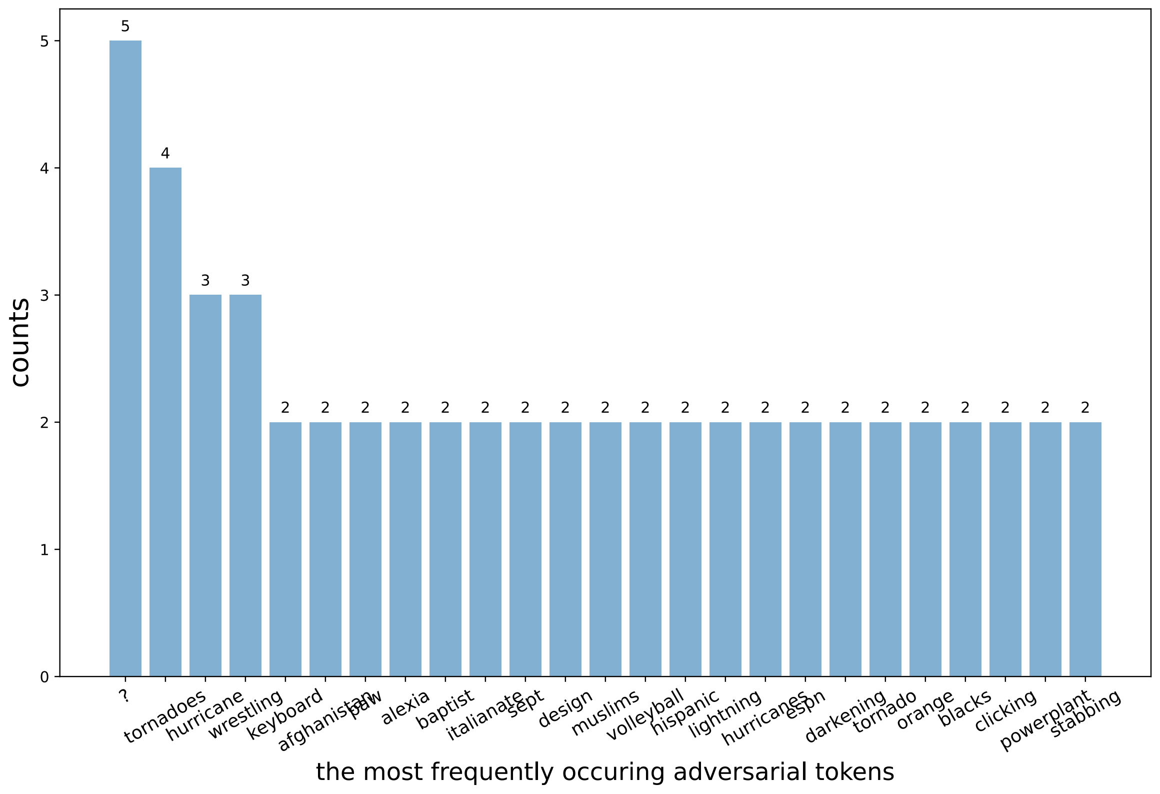


Figure 11: Counts of the most frequent adversarial tokens in descending order on rank promotion attack to ClueWeb09 test set.

5.2.2.1 Local Adversarial Tokens

We conducted the same analysis approaches on local adversarial tokens generated by attack on rank promotion, more specifically, attack on the pre-retrieved last-ranked document of each query across the 40 queries test set. Our results present a consistency of patterns with rank demotion adversarial tokens in terms of distance-wise analysis and visualization. Those local tokens are proven to possess global attack capability as well. Therefore, we focus on the analysis of token bias.

As a intuitive representation, we have also calculated the frequency of occurrence of each token and have highlighted their frequency in colors in the heatmap.

As can be seen in Figure 11, unlike the local adversarial tokens of rank demotion attack, none of rank promotion's tokens have a dominant occurrence of 17 times like token 'acceptable'. The most frequent word is the question mark '?', with only 5 times. However, two differences are particularly noticeable:

1. Words such as 'tornado', 'hurricane', and other natural disasters or nature-related words often appear as adversarial tokens. The plurals 'tornadoes' and 'hurricanes' having similar word embeddings also occur frequently. The same goes for words such as 'baptist' and 'muslims' which are related to religion. Those tokens have nothing to do with query contents at first sight. They are

therefore tentatively regarded as model biases specifically towards some topics.

2. Most of these tokens are related to the query itself, and some appear more than once in the same token sequence. For example, the query with index 0 is ’keyboard review’, and among the 5 tokens it corresponds to, ’keyboard’ appears twice, while ‘reviews’ also appears once.

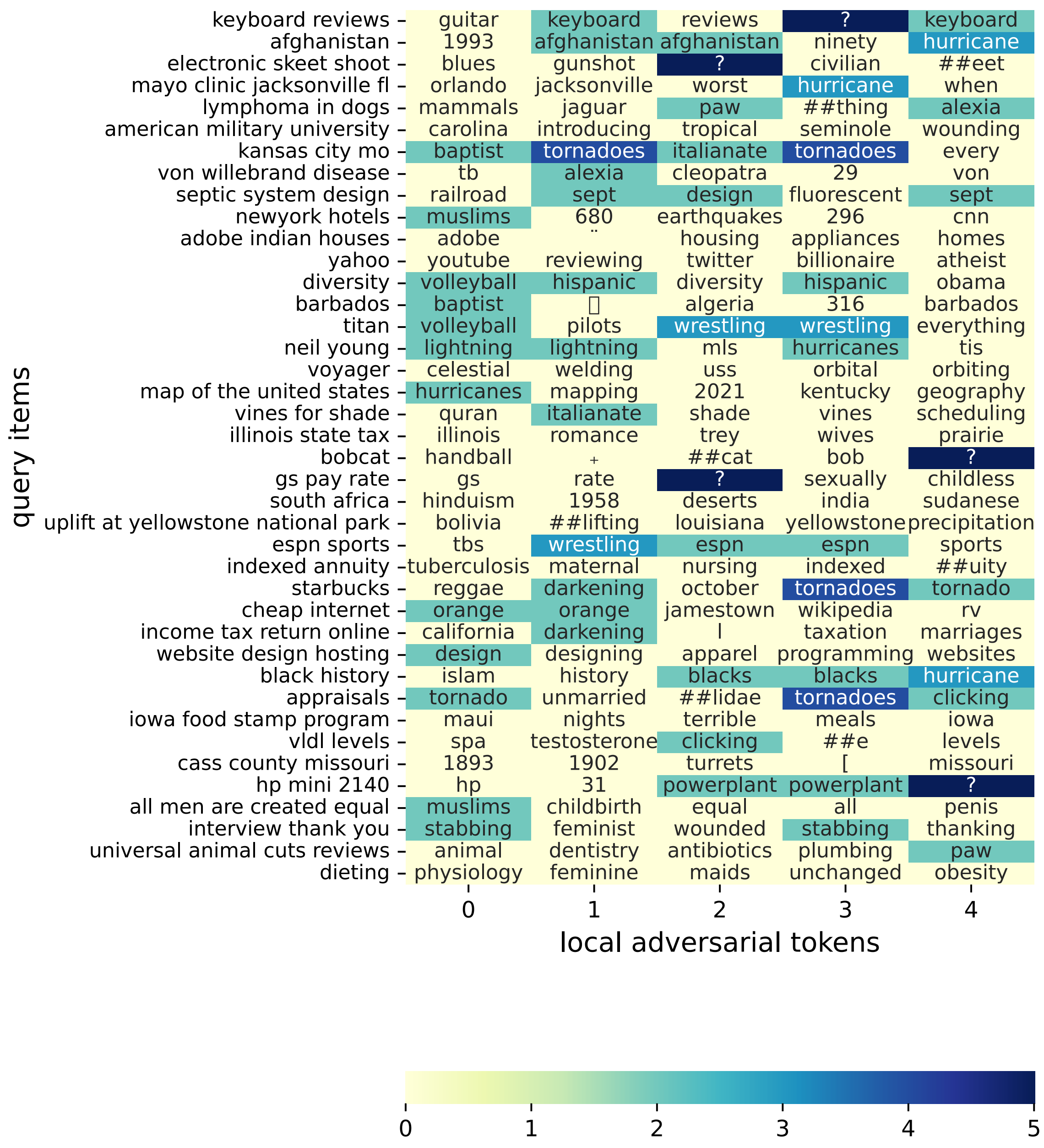
Then we analyze the attack performance of local adversarial tokens of rank promotion attack in the way we did for rank demotion attack, shown in Figure 13. At this time, ‘?’ rather than ‘acceptable’ becomes the most frequent token followed by ‘tornadoes’, ‘hurricane’, ‘wrestling’. Generally speaking, the local attack on the last-ranked document is not as good as that on the top-ranked document, with only an average of 71 rank differences compared to 89 rank differences. Concerning this phenomenon, we discuss and try to explain it in section 6.1. What shows differences here is that token sequences including the second rather than the first frequent token, which is ‘tornadoes’ in this case, always dominate in the attack performance compared with that of the other token sequences that include the first, the third, or the fourth most frequent adversarial token. What is shown here is the average performance of ten times attacking, so we have reason to believe that this model has leading bias to ‘tornadoes’ than ‘?’. Another major difference is that this time token sequence composed of 5 most frequent tokens in order remain a large performance gap with the best result. This result indicates that the first five frequent tokens are not like those in rank demotion attacks who have a relatively strong triggering performance on all attacked documents. No token always appearing in the first position can also explain this problem: there is an absence of a particular token that can always make the loss boost significantly large.

5.2.2.2 Global Adversarial Tokens

Similarly, we use batch attack on rank promotion task and find two global adversarial token sequences, which are:

1. ‘florida warming September precipitation hurricanes’
2. ‘hinduism earthquakes childbirth tornadoes wikipedia’

These two groups of adversarial words have distinct categories of meanings compared to rank demotion global tokens. These words are generally divided into two main topics, one part, for example, ‘warming’, ‘precipitation’, ‘hurricanes’, ‘earthquakes’, ‘tornadoes’, related to natural disasters or natural phenomena, the other part ‘hinduism’ that occurs in the first position, together with ‘baptist’ and ‘muslims’ in local adversarial tokens, belonging to religion-related words.



query items	0	1	2	3	4
keyboard reviews	guitar	keyboard	reviews	?	keyboard
afghanistan	1993	afghanistan	afghanistan	ninety	hurricane
electronic skeet shoot	blues	gunshot	?	civilian	##eet
mayo clinic jacksonville fl	orlando	jacksonville	worst	hurricane	when
lymphoma in dogs	mammals	jaguar	paw	##thing	alexia
american military university	carolina	introducing	tropical	seminole	wounding
kansas city mo	baptist	tornadoes	italianate	tornadoes	every
von willebrand disease	tb	alexia	cleopatra	29	von
septic system design	railroad	sept	design	fluorescent	sept
newyork hotels	muslims	680	earthquakes	296	cnn
adobe indian houses	adobe	¨	housing	appliances	homes
yahoo	youtube	reviewing	twitter	billionaire	atheist
diversity	volleyball	hispanic	diversity	hispanic	obama
barbados	baptist	▯	algeria	316	barbados
titan	volleyball	pilots	wrestling	wrestling	everything
neil young	lightning	lightning	mls	hurricanes	tis
voyager	celestial	welding	uss	orbital	orbiting
map of the united states	hurricanes	mapping	2021	kentucky	geography
vines for shade	quran	italianate	shade	vines	scheduling
illinois state tax	illinois	romance	trey	wives	prairie
bobcat	handball	+	##cat	bob	?
gs pay rate	gs	rate	?	sexually	childless
south africa	hinduism	1958	deserts	india	sudanese
uplift at yellowstone national park	bolivia	##lifting	louisiana	yellowstone	precipitation
espn sports	tbs	wrestling	espn	espn	sports
indexed annuity	tuberculosis	maternal	nursing	indexed	##uity
starbucks	reggae	darkening	october	tornadoes	tornado
cheap internet	orange	orange	jamestown	wikipedia	rv
income tax return online	california	darkening	l	taxation	marriages
website design hosting	design	designing	apparel	programming	websites
black history	islam	history	blacks	blacks	hurricane
appraisals	tornado	unmarried	##lidae	tornadoes	clicking
iowa food stamp program	maui	nights	terrible	meals	iowa
vldl levels	spa	testosterone	clicking	##e	levels
cass county missouri	1893	1902	turrets	[	missouri
hp mini 2140	hp	31	powerplant	powerplant	?
all men are created equal	muslims	childbirth	equal	all	penis
interview thank you	stabbing	feminist	wounded	stabbing	thanking
universal animal cuts reviews	animal	dentistry	antibiotics	plumbing	paw
dieting	physiology	feminine	maids	unchanged	obesity

Figure 12: Adversarial tokens for local attack on rank promotion scenario. Specific topic-related tokens and query tokens occur frequently.

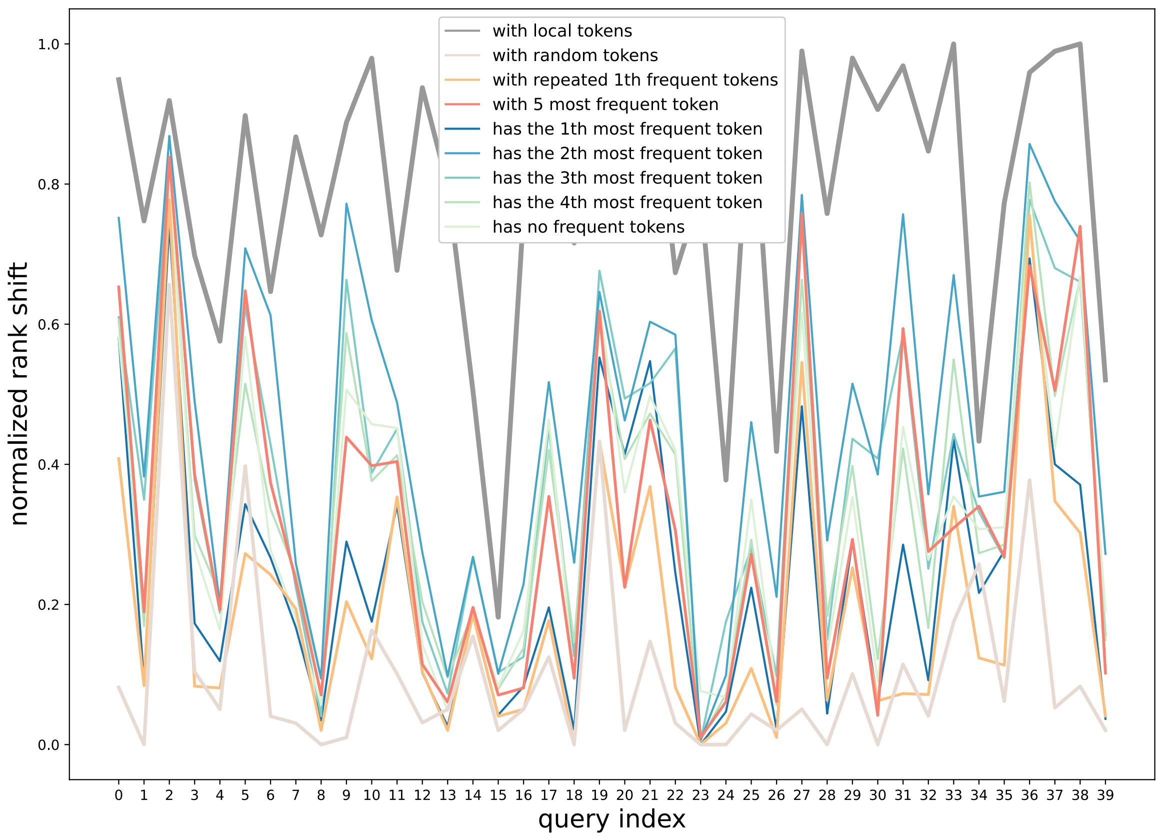


Figure 13: Normalized rank shifts on rank promotion attack using the most frequently occurring local adversarial tokens.

Among them, tokens ‘hurricanes’, ‘tornadoes’ that appear frequently in local show up here again, which goes to prove the global effect of local adversarial tokens again and also can be evidence of the model’s bias towards these specific topics.

We use two sets of batch adversarial tokens to attack all queries as well. The two sets both achieve comparable results with 55 and 49 rank differences respectively, both better than all other attack performances using frequent tokens.

5.2.2.3 Adversarial Tokens vs Explainers on Rank Promotion Attack

Subsection 2.4 explains that since expansion terms can be seen as complementary to different aspects of a query, it makes sense to add them to the front of the document to replace adversarial tokens in comparison with rank promotion attack performance. As an example, the explainers for the query ’keyboard review’ with query-id 80 yields the following ten expansion terms:

’doo’, ’techno’, ’keyboard’, ’hardware’, ’home’, ’review’, ’retailer’, ’digital’, ’psion’, ’duracell’.

Among them ”doo” can be a complement to the meaning of music keyboard. The other terms are more in line with the meaning of electronic keyboard.

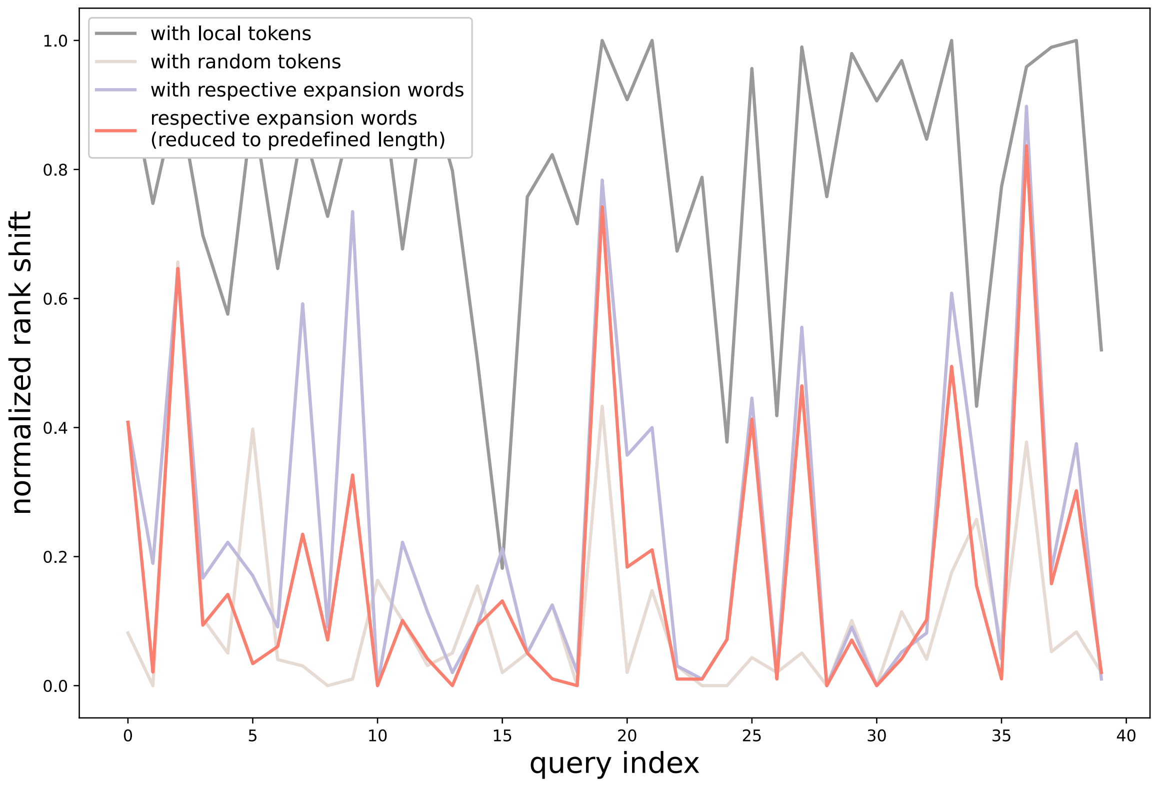


Figure 14: Comparison of performance on rank promotion attack using gradient-based method and the explainer method.

One problem is that after five words are tokenized into tokens, there will be greater than or equal to five token ids although it is difficult for people to distinguish them, but the number of words added will directly affect the result. In order to compare the differences, the following Figure 14 illustrates the comparison between 5 words and 5 token words.

The gray line in the figure indicates the results of attacking each individual query using the local gradient-based method, representing the largest normalized rank shifts in the respective rank promotion attack. The cream line represents the ranking shifts of the last-ranked document using 5 random token words as adversarial token sequence, while the purple line represents the performance of using 5 query expansion terms to replace gradient-based tokens, and the orange line shows the performance under the strict constraints that 5 expansion terms are truncated to 5 token words.

It is clear to see that the orange values will always be less than or equal to the purple values with an average rank differences of 17 to 24, which also provides evidence that adversarial token length has an effect on the results.

It can also be observed from the graph that the effect of the explainer method is mostly better than the random tokens attack which can only contribute 10 rank increases after the attack. The results of the attacks using the gradient-based approach are generally better than those using the explainer attack with an average of 79 rank promotions across all 40 queries attack. Out of 40 independent queries, only 9 have better results using explainer, but still indicates the effectiveness of deploying the explainer as an attack method.

This shows that gradient-based adversarial tokens can be used both as a means to attack a query search individually and to find a global trigger sequence that disorder the rankings of the whole dataset based on batch attack, which has a broader target than the explainer approach applied only to independent queries. The gradient-based method can be applied both to degrade and upgrade a specific document to a query, which is wider than the multiplexer's method that can only be applied to demote a document ranking. Overall, gradient-based adversarial tokens present a better performance in terms of attack effectiveness, but the explainer also proves its feasibility in attacking neural ranking models from the point of view of explaining a query.

5.2.3 Bais Analysis

The previous experiments are conducted on a ClueWeb09 test set containing 40 queries. We find model biases towards certain types of tokens when generating the corresponding local adversarial tokens via rank demotion attack and rank promotion attack respectively. By comparing the local adversarial tokens of individual queries with global triggers for all queries, We observe that local adversarial tokens recur across multiple queries and also in across datasets.

We therefore expand the experimental dataset to the entire ClueWeb09 dataset containing 200 queries to support the conclusions we get on the test set. Since the bias of the model will eventually be attributed to the bias of the training dataset, we estimate in advance that the pre-trained model will give results consistent with the phenomena we observe in the relatively small test set, or even more pronounced, under the circumstance of having already seen the training set.

It is reasonable to find problems in the test set and then backtrack to the training set to verify it. However, we are still curious whether the model's bias towards one dataset migrates to other datasets, or in other words, whether the model shows distinguishable biases towards different datasets. Therefore we also conduct experiments on TREC-DL passages dataset on the same pre-trained BERT model. A comparison of bias tokens in both datasets are integrated in the last part suggesting a pretraining bias.

5.2.3.1 Bias tokens in ClueWeb09

We first conduct rank demotion and then rank promotion across 200 queries in ClueWeb09 dataset, calculate the occurrence frequency of their local adversarial tokens, and observe their positions in the heatmap. Similarly, we generate global triggers for 200 queries to verify the global effects of local adversarial tokens. After finding the bias of the model, we attempt to explain why the model has such a bias.

Rank demotion on ClueWeb09 200 queries

We first generate individual adversarial tokens for 200 queries and select the token sequences with the highest relative rank shifts to show the final results. The following bar chart in Figure 15 records the most frequent tokens among the 200*5 tokens in order from the highest to the lowest. It is very interesting to notice that 'acceptable' again occupies the first with 82 times. The second place is far behind the first place with only 21 times. The rest of the tokens appear less than 10% of the time.

Due to space limitation, we do not add the long heatmap of 200 queries in the paper, but we observe that all 82 'acceptable' appear in different token sequences, and only 13 of them appear in the non-first position, while the other 69 appear all in the first position of a token sequence.

That means 'acceptable' has a 41% probability of appearing in local adversarial tokens and '.' also has a 10.5% probability, which is not negligible. This supports our hypothesis that the pre-trained BERT model has a bias to 'acceptable' in rank demotion attacks.

If looking closely, we can find that these high-frequency tokens have a lot of repetition with those of 40 queries, for example, 'acceptable', '.', 'resulting', 'hapoel'. Because of the larger sample size, it is also easier to detect that many high-frequency tokens share the same word stems. For instance, not only 'acceptable', but also 'unacceptable' occurs parallelly this time with a frequency of non-ignorable 11 times. Also, the family of 'resulting', 'result', 'resulted' makes a combined 29 appearances. But other than that, it's still hard to tell what these tokens have in common.

We then conduct the global attack on rank demotion for all 200 queries with a batch size of 4 and get the following two global adversarial token sequences:

1. 'acceptable potential cameroon\u02c8 copa'
2. 'acceptable participating institution \u1d40 criterion'

The appearance of 'acceptable' is in line with our expectations and proves the global attack effect of local adversarial tokens.

Rank promotion on ClueWeb09 200 queries

Based on the promising results of rank demotion attack on 200 queries, we believe that the extended experiments on rank promotion attack can help us to unveil more patterns about the model bias.

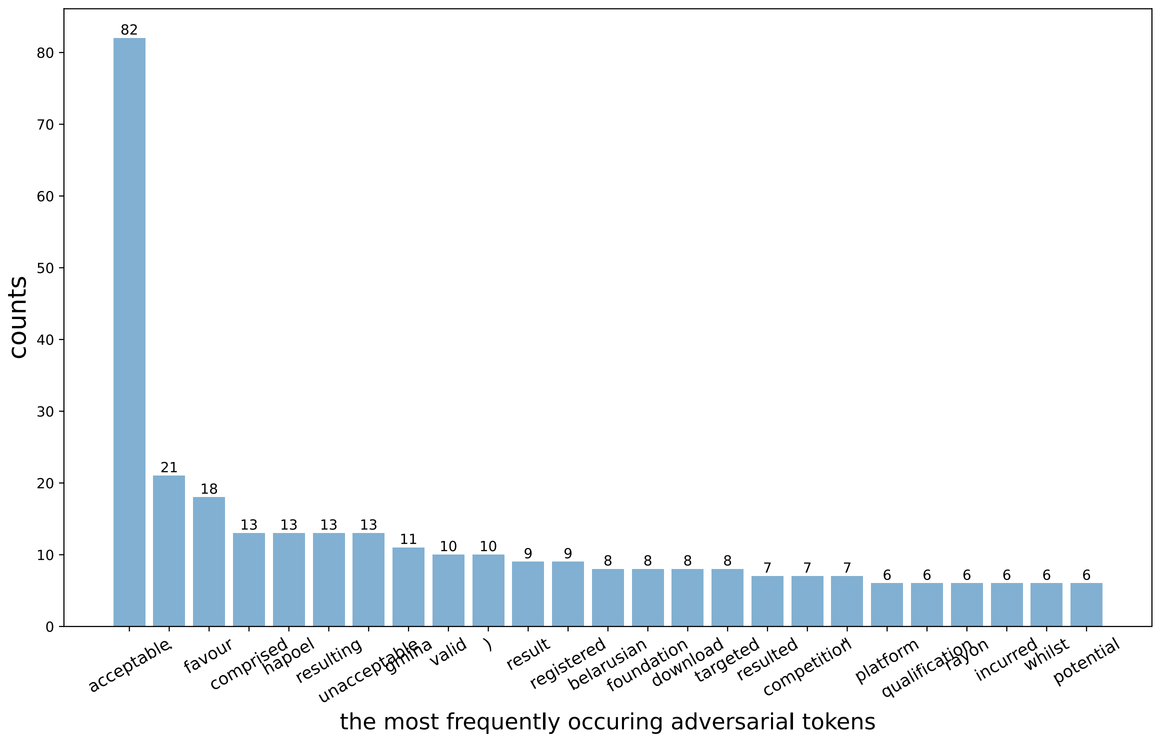


Figure 15: Counts of the most frequent adversarial tokens in descending order on rank demotion attack to 200 queries ClueWeb09.

We start with generating local adversarial tokens and then count the most frequent tokens which are shown in figure. This time the most frequent appearing token changes from '?' to 'tornadoes', which previously in 40 queries ranks in the second place. The ranking of these two tokens flips. But the gap between the top two ranking tokens is not as big as it is in rank demotion attacks. They occupy 22 and 17 times among 200 * 5 tokens, more accurately occurring in 10% and 8.5% adversarial token sequences. Although the heatmap is not presented here, note that only 4 of the 22 'tornadoes' appear in the first position. The other tokens are also irregularly scattered on the heatmap.

In addition to 'tornadoes' and '?', other frequent adversarial tokens for 40 queries like 'hurricane', 'hurricanes', 'baptist' are also on the list. Besides, token 'precipitation', which appears only in the batch triggers for 40 queries, appears 6 times here. These high-frequency tokens can be broadly classified into 4 major categories: nature-related like 'tornadoes', 'hurricanes', 'hurricane', 'precipitation'; religion-related including 'baptist', 'hinduism'; social topic-related including 'childbirth', 'unmarried', 'marries' and ethnicity-related like 'latino', 'cherokee'. These categories include and extend the previous categories based on 40 queries.

Similarly, the global triggers obtained using batch in attack are as follows:

1. 'ike hurricane october precipitation hurricanes'

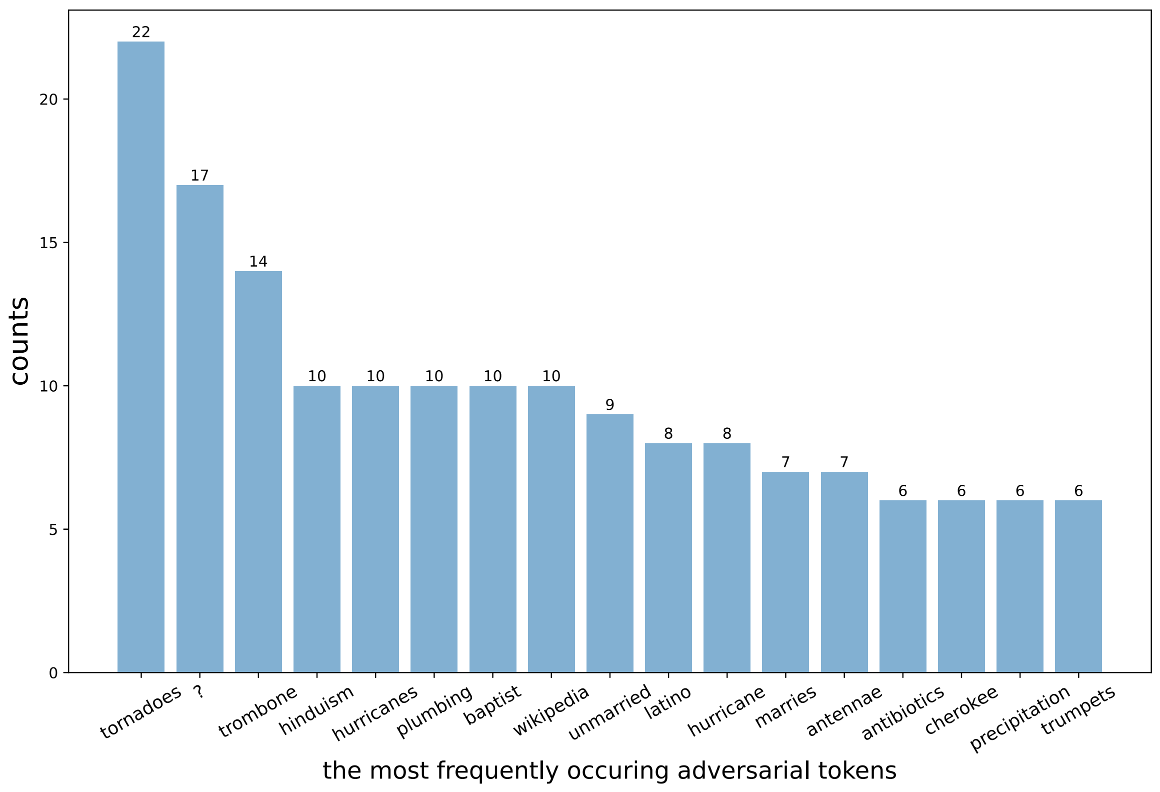


Figure 16: Counts of the most frequent adversarial tokens in descending order on rank promotion attack to 200 queries ClueWeb09.

2. ‘february every hurricane august hurricane’

‘hurricane-‘ appears a total of four times in the two groups, twice in each group. Such a high frequency is enough to indicate the model’s bias towards it. Another noteworthy phenomenon that emerges here is that the month appears three times with ‘october’, ‘february’ and ‘august’, which implies that the month may also be a bias that has not been reflected before.

The extension experiments on 200 queries prove our previous hypothesis on the model bias, and the large amount of experimental instances also shows that local adversarial tokens do have global attack effects. Next, we group these potential tokens and apply attacks to analyze whether the model has a bias on them.

5.2.3.2 Bias tokens in TREC-DL

[43] shows that although the adversarial token sequences are generated for a specific model in a white-box manner, they are proved to successfully fool other victim models with transferring capability.

Attack transferability can be proved on attacking reading comprehension task. [43] test their triggers that are generated for SQuAD via white-box access on black-box models: QANet, an ElMo-based BiDAF model and a BiDAF model using character level convolutions. Although these models are constructed with different architectures and words have different embedding methods and tokenization standards, attacks with those trigger tokens are somehow successful in general depending on the question type. For question type 'why', the triggers even cause a higher success rate on ELMo-based model. But for the other question type and attacked models, the performances are not promising.

Another successful case is attacks on conditional text generation, where the trigger sequence is made for GPT-2 117M parameter model, but it also triggers and generates text with 58% aggressive words on 345M parameter model.

Both cases are based on the attack to other models with the same trigger sequence generated from one certain model. The results can somehow indicate trigger transferability, but not convincing with a low success rate in some cases.

Rank demotion on TREC-DL 978 queries

In our experiment, we try to create adversarial tokens applying attacks on the TREC-DL dataset with the model unchanged. Because the test set only has a limited 200 queries, we thus randomly choose 978 queries in the dev set after filtering out queries with a small amount of retrieved documents. Considering the documents actually being one short passage, we set token sequence length of 3 rather than 5 to avoid 'overfit' with a 100% successful attack rate, which is actually proved to be true in ablation experiments.

Figure 17 first presents the most frequent tokens generated from rank demotion attack.
As is shown in the figure, the tokens that overlap with the adversarial tokens generated by Cluweb09 hold no small amount. Among those overlapping tokens, 'acceptable' is not at the forefront this time. The most frequent occurrence token is 'contents' with 131 times, occurring in no more than 14% adversarial sequences. In second place is the token that also appears in Cluweb09, 'registered' with 90 times, followed by 'recognized' with 90 times. No token domains like 'acceptable' in Clueweb09' does.

We then attack rank demotion tasks globally based on batch. This time we also attack two times with [MASK][MASK][MASK] and 3 randomly chosen tokens as initialized token sequences. Two identical adversarial token sequences as follows are obtained:

1. 'standings hapoel hapoel'
2. 'standings hapoel hapoel'

Global triggers 'hapoel' and 'standings' are both local adversarial tokens. This is consistent with what we observed previously in the ClueWeb09 dataset: the global trigger tokens are also local adversarial tokens for some documents. What is intriguing is that

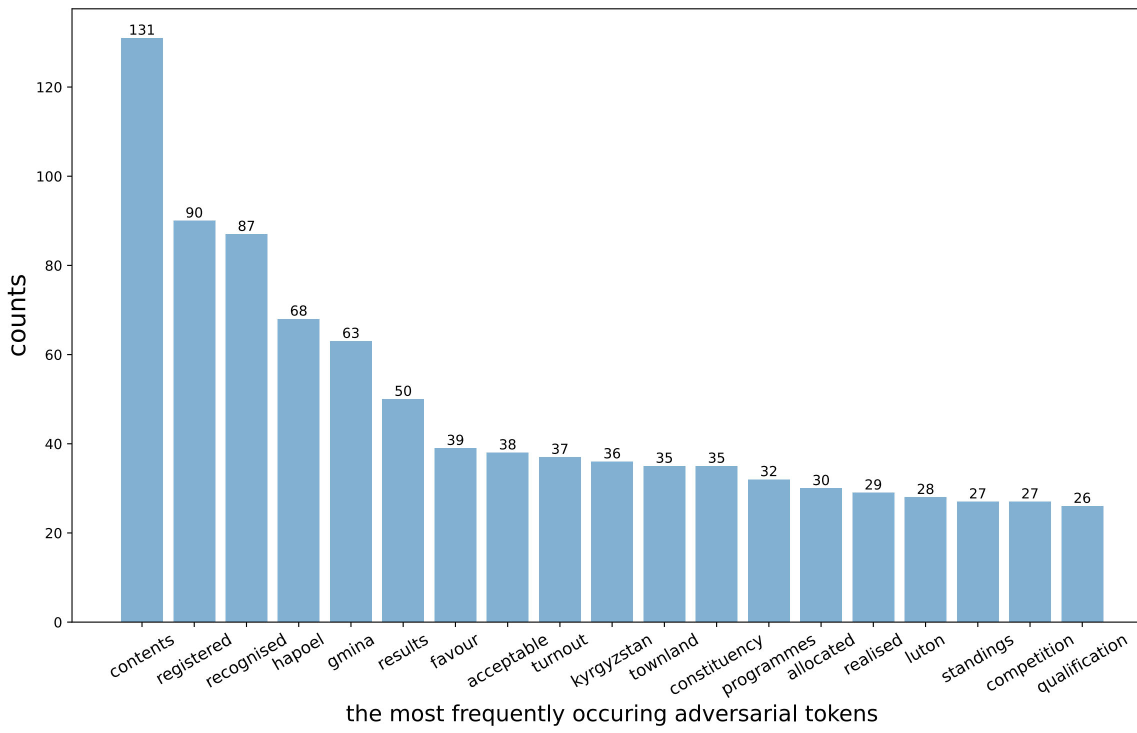


Figure 17: Counts of the most frequent adversarial tokens in descending order on rank demotion attack to TREC-DL.

‘hapoel’, which plays an important role in Clueweb09, shows as well its irreplaceable global significance in Trec-DL.

Rank promotion on TREC-DL 978 queries

After observing such a positive result, we also conduct attacks on rank promotion in TREC-DL. The counts of the most frequent local adversarial tokens are shown as follows:
The adversarial tokens generated on rank promotion attacks share a more striking consistency on both datasets. Three of the top four tokens stay unchanged, excluding some changes in ranking. Besides the top four tokens, ‘hurricane’, ‘plumbing’, and ‘wrestling’ are also on the top ranking list for both datasets. Global triggers generated on rank promotion attack are:

1. ’iv influenza seminole’

2. ’iv influenza seminole’

They are still two identical trigger sequences, but unfortunately, none of these tokens is among the most frequent occurrence tokens above. However, ‘influenza’ is related to disease and together with ‘antibiotics’ and ‘alexia’ belong to the same group. ‘Seminole’ together with ‘latino’, ‘cherokee’ are associated with race. Both these two groups do not occur very frequently in ClueWeb09 probably due to the limitation of the observation samples. Attacking in the relatively large TREC-DL dataset helps to reveal and

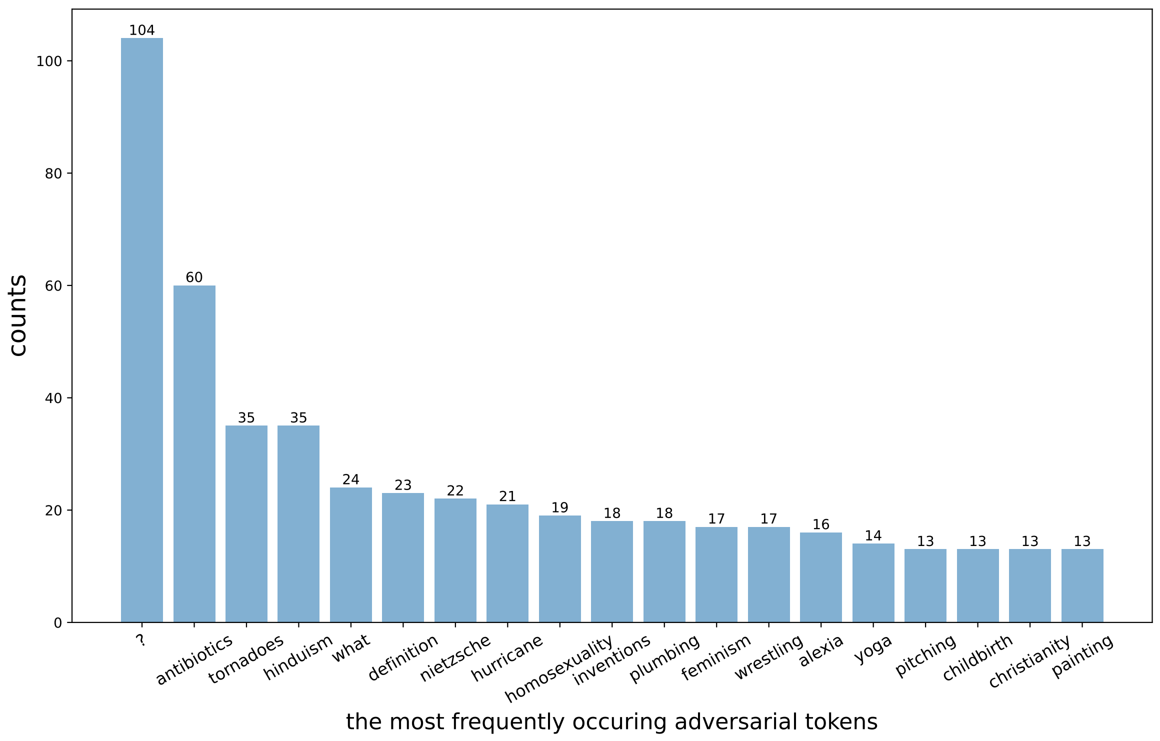


Figure 18: Counts of the most frequent adversarial tokens in descending order on rank promotion attack to TREC-DL .

determine bias domains.

In a summary, from rank demotion attack and rank promotion attack on Trec-DL dataset, although global triggers may look different, local adversarial tokens have a great similarity. Without careful comparison, it is easy to miss the fact that locally, the model shows the same bias towards some topic words when attacking different datasets, and even from a view of the global attack, triggers generated during attack also represent bias in some domains tokens.

5.2.3.3 Summary

In subsection 5.3.1, we find the model has a potential bias towards natural disaster words and religion-related words in rank promotion attack at the ClueWeb09 dataset. Apart from those, we also noticed a boost of medicine-related and ethnicity-related words as adversary tokens in TREC-DL. We integrate mentioned four topics adversarial tokens that were generated in both datasets individually with the threshold of 2 frequencies and find overlapping but together with a preference in different topics of these two datasets.

The following Figure 19 illustrates all topics tokens in 2D visualization. As it can be seen, nature-related tokens, religion-related tokens, ethnicity-related and medicine-related tokens are clustered in the top, bottom, right, and left respectively. Both datasets present a model bias to nature and religious words. Ethnic words are also

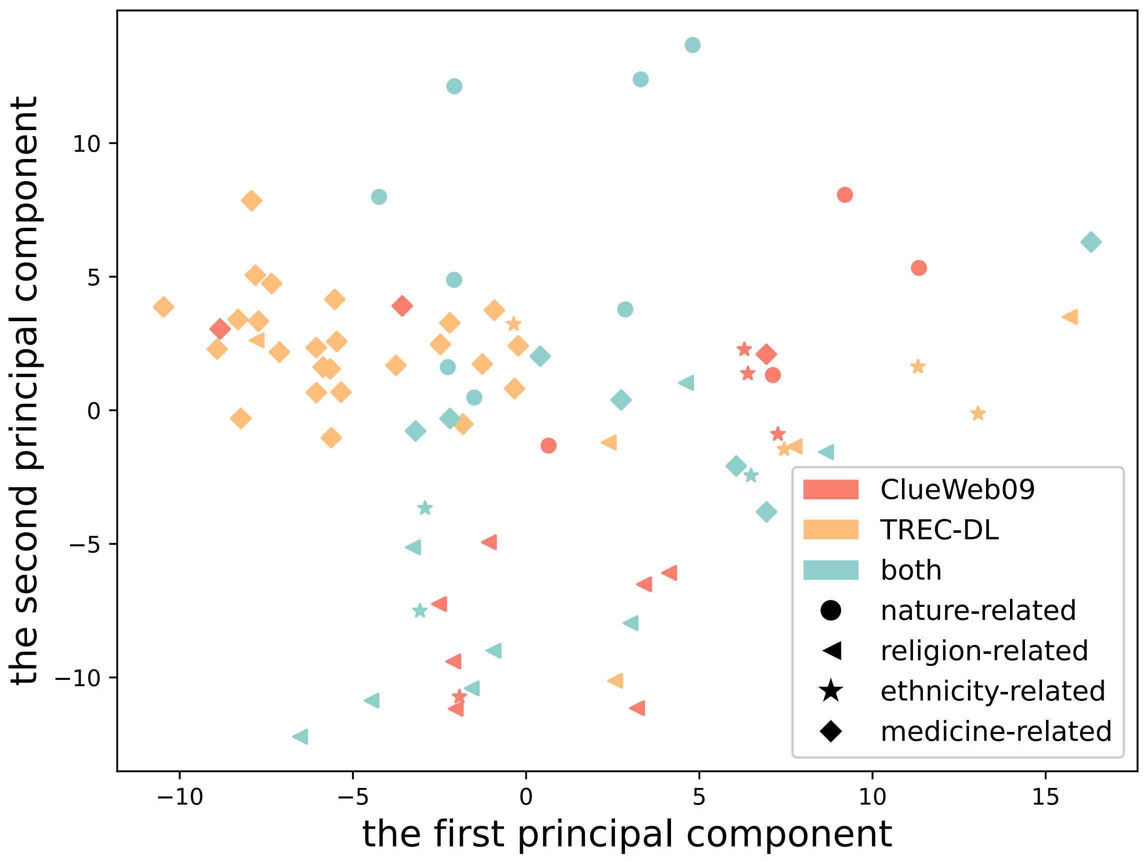


Figure 19: 2D visualization of frequent local adversarial tokens for both datasets. Note that the topic of each token is manually assigned.

distributed but with few overlapping in both datasets.

Meanwhile, TREC-DL has a strong preference for medicine words while for ClueWeb09, the main bias tokens are composed of nature and religion-related words.

Since we use the same bert-base-uncased ranker model pretrained on ClueWeb09 and test it on two different datasets. Hypothesizes of bias transferability can be that the model has memorized some bias tokens in the training set or the training set itself is constructed in a way with preferences for some topic words. With a thorough investigation on the 200 queries from ClueWeb09, we found that besides 3 queries associated with nature topic, 24 queries are about disease and medicine related topics which however render little bias in ClueWeb09. For TREC-DL, we leave a more automatic method for topic extraction to future work due to the large training corpus.

5.3 Discussion

According to the previous experimental results, we observed that the BERT model has a potential bias to some topic tokens. Moreover, attacks on rank demotion always achieve a better attack effectiveness than that attacks on rank promotion tasks.

We propose multiple hypotheses for the existence of those performances and seek confirmation in subsection 5.3.1 and 5.3.2 respectively. In addition, we conduct experiments on adversarial tokens in terms of word embeddings with an expectation to find potential clustering patterns in substion 5.3.3.

5.3.1 Hypothesis on Bias Tokens

In the previous subsection we find some adversarial tokens generated in 200 queries attack that can be easily grouped in different categories in terms of semantics. We are curious about why the model produces bias for those word tokens. About that we give two possible hypotheses: the adversarial tokens have a high mutual information with the relevance label; or datasets retain some topic preferences.

Training Set Relevance Score

To verify our conjecture, we need to carefully search the dataset used to train the model, because the bias of the model is ultimately attributed to the bias of the dataset. We look at the training set which includes 155 queries and their corresponding top 100 documents per query pre-retrieved by a classical (BM25) text ranker model, and filter out a total of 2568 documents each with a relevance score greater than 0. Among them we list how many documents each token appears in the Table 1.

For rank demotion attack, since all words do not belong to a certain category, we collect them together and term it 'others'. Among them only '. ' and 'foundation' demonstrate that they appear 2541 and 1204 times respectively in 2568 positive related documents, i.e. the percentage of occurrence is high as 99% and 47%. Such a result is contrary to the intuitive understanding because their frequency of occurrence is so high that the model should hardly trigger a rank disorder just because of their appearance. On the contrary, 'acceptable' and 'unacceptable' only appear in 70 and 8 documents, with only 2.7% and 0.3% occurrence frequency. It is understandable that the model may learn some trapped patterns because of their rare occurrences which lead to a possibility of bias, but this does not explain why 'acceptable' is more important than the other words to trigger a decrease in rank demotion attack.

For the adversarial tokens of rank promotion attack, we divide them into three main categories, namely 'Nature', 'Religion', 'Month', the others are placed inside 'Others'. It is interesting to note that in the month category, all months' appearances are very close to each other, all within 700 to 1000, with only May appearing relatively high, with 1737 times. Considering that they only appear in global adversarial tokens, we speculate that their attack performance, although with generalized effects, does not reflect the significant bias of the model.

In contrast, all the tokens in 'nature' and 'religion', and most tokens in 'others', appear

Table 1: Adversarial token domains and their numbers of documents that have a relevance score high than 0.

Rank promotion								Rank demotion	
Nature-related		Religion-related		Month		Others		Others	
hurricane	121	hinduism	0	January	919	?	1600	acceptable	70
hurricanes	75	muslims	18	February	960	trombone	14	.	2541
tornadoes	78	baptist	110	March	905	plumbing	33	unacceptable	8
tornado	98	atheist	1	April	713	Wikipedia	1120	register	269
earthquakes	72	celestial	58	May	1737	ike	20	platform	108
warming	58	quran	4	June	757	every	989	resulting	371
lightning	51	islam	57	July	783	childbirth	3	foundation	1204
prairie	111	unitarian	9	August	705	unmarries	0	favour	36
deserts	70	judaism	9	September	752	latino	61	comprised	59
precipitation	90	preach	30	October	888	marries	11	competition	59
darkening	11	mormon	60	November	704	antennae	4	rayon	201
thunder	69	preaching	5	December	819	cherokee	30	hapoel	11

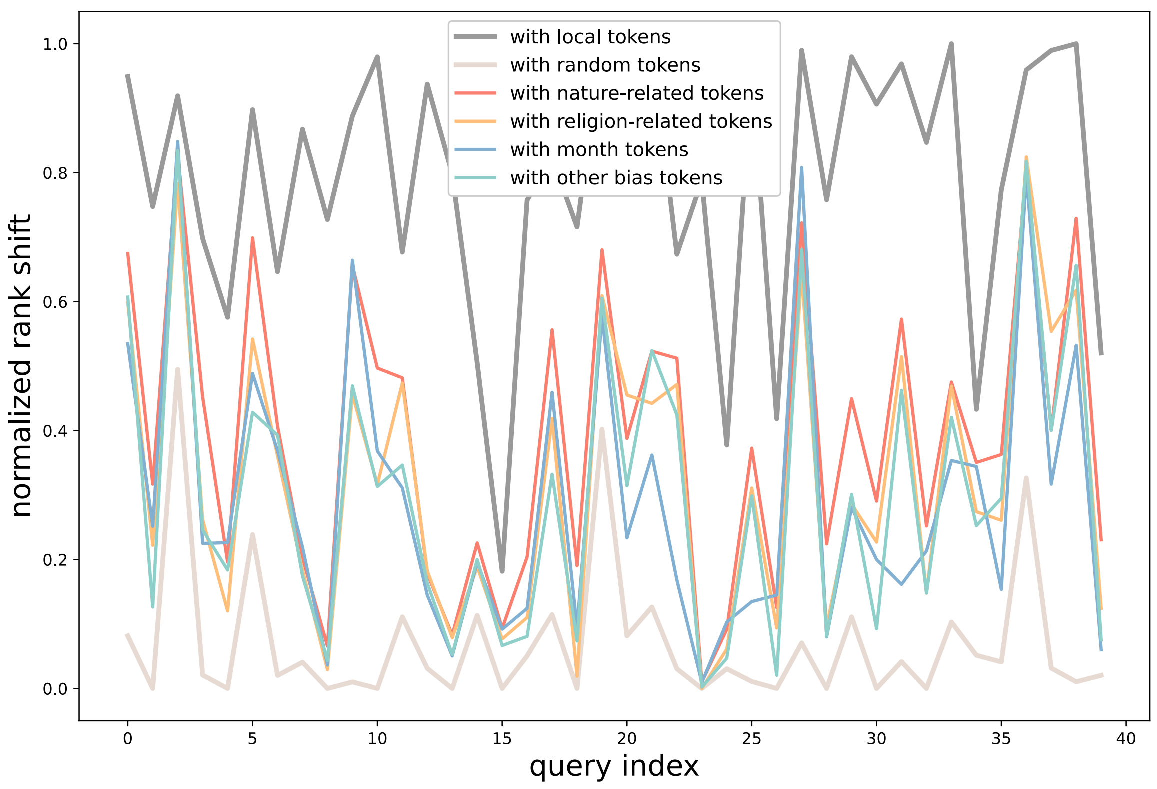


Figure 20: Attack performance on rank promotion attack with potential bias tokens in different topics.

largely within or around 100 times. Tokens such as 'hinduism', 'unmarries' that does not even appear in one document are also able to fool the victim model to give high relevance scores to the associated query when concatenated to the front of the document. This finding is intriguing, and it is difficult to explain intuitively why the model has a positive preference for tokens that appear only in a few positive labeled documents.

In a summary, our hypothesis on the high relevance score of training documents where some domain tokens appear does not match our expectation and cannot be a direct proof to model bias. Therefore, we use distinguishable tokens to conduct rank promotion attack in an attempt to directly observe and compare the effects of their attacks.

Because the adversarial tokens of rank promotion attack have more obvious domain distinction, we only present the effect of it on 40 queries with different domain tokens in the Figure 20.
Our rank promotion attack with potential bias tokens proceeds as follows:

1. First manually list the potential adversary tokens from different domains (here we use the tokens from the previous table 1) in nature, religion, month, and others.

2. Conduct rank promotion attack ten times with randomly selected tokens in a

specified length for each domain in the list.

3. Average results of each domain.

The grey bolded line indicates the best performance an attack can achieve for each last-ranked document, while the cream bolded line represents the performance of an attack with randomly selected tokens, which turns out to be the worst performance. Therefore we consider these two lines as our baselines.

Overall, the performances of all those domain tokens are between the upper bound and the lower bound, indicating the attack effectiveness with those tokens.

If we look carefully, we can see that the average results with nature-related tokens are always higher than those with religion-related tokens. These two groups' results are both better than the attack with tokens in the 'Others' group. Although the performance of 'Month' tokens fluctuates, nature-related tokens demonstrate more robust and powerful aggressivity.

Since the results presented above are based on the average of ten calculations, we can draw this faithful conclusion: The model does have a bias towards the domain terms above, and the model has a greater preference for nature-related tokens than for religion-related terms. Other terms integrate and demonstrate the poorest performance among all. Parallel to this, the model has a bias towards nature-related tokens over the month tokens. However, the model bias to some specific topic tokens cannot be explained by a large proportion of positive relevance scores of related documents in the training set.

Training Set Topic Reference

Another hypothesis trying to explain the existence of model bias is that among the training set, there are a large proportion of queries that are related to those domains.

To test this idea, we manually examine 200 queries to see what domains they cover. Among the queries directly related to natural disasters are No.75 'Tornadoes', and No.149 'uplift at Yellowstone national park' which indicates the volcanic eruption that occurred in Yellowstone. It is worth noting that 'tornadoes' that appears frequently as adversarial token turn out to be a query itself in the training set, but even though No.149 is looking for content about volcanic eruptions, neither 'volcanic' nor 'eruption' ever appear in the adversarial tokens.

Other queries such as 'discovery', 'continental plates', and 'the sun' seem to be very reminiscent of nature topics, but only 'continental plates' related documents will mention that the movement of continental plates can cause various natural disasters, therefore we only treat 'continental plates' as nature-related queries. There are also some other queries like 'gps', 'map', 'penguins', 'satellite', which may also involve nature-related documents, but are currently considered as indirect nature-related queries.

What makes it ambiguous is that among the 200 queries, there are often queries about the map of a location, or more specifically a city, a country, a club, a hotel, which has a specific geographical location. Therefore the documents related to these queries will involve some specific geographical information. Although there is a lot of cross information between nature and geography domain, we separate geography from nature here, and by nature-related, we mean especially queries related to natural disasters. So the location-related queries mentioned above are not considered as nature-related queries.

In addition, ‘trombone’ which appears frequently in local attacks on 200 queries turns out to be a query in form of ‘trombone for sale’. It is the only token other than ’tornadoes’ that appears in the form of the query itself.

It should be noted that many countries appear in the 200 queries, and due to the sensitivity and specificity of religion topics, we do not consider here to calculate which queries are related to religion.

In a conclusion, we only count 3 queries from all 200 queries that are directly related to the nature topic (disaster). Therefore, the number of relevant queries in the training set does not support our hypothesis to explain why the pre-trained model has a bias towards those domain tokens which rarely occur in the training set.

But one of these 3 queries is the most frequently occurring adversarial token ‘tornadoes’. Another non-nature-related query ‘trombone for sale’ appears in the training set also contains one of the most frequent adversarial tokens ‘trombone’. It looks reasonable that the most frequently occurring adversarial tokens are queries themselves, yet we still have questions about why the model choose them but not the others.

5.3.2 Document Redundancy Analysis

To facilitate the comparison of conclusions and to reduce load to our experiments, for every query only the top 100 retrieved documents are considered for evaluation. When observing the results, it was found that these top 100 documents may contain similar or duplicate documents. After accurate quantitative analysis, it is reasonable to believe that these duplicate or extremely similar documents can have a significant impact on the evaluation results. The following analysis focuses on the effect of duplicate documents and Wikipedia documents (since clubweb09 contains the full Wikipedia page contents) on the attack performance.

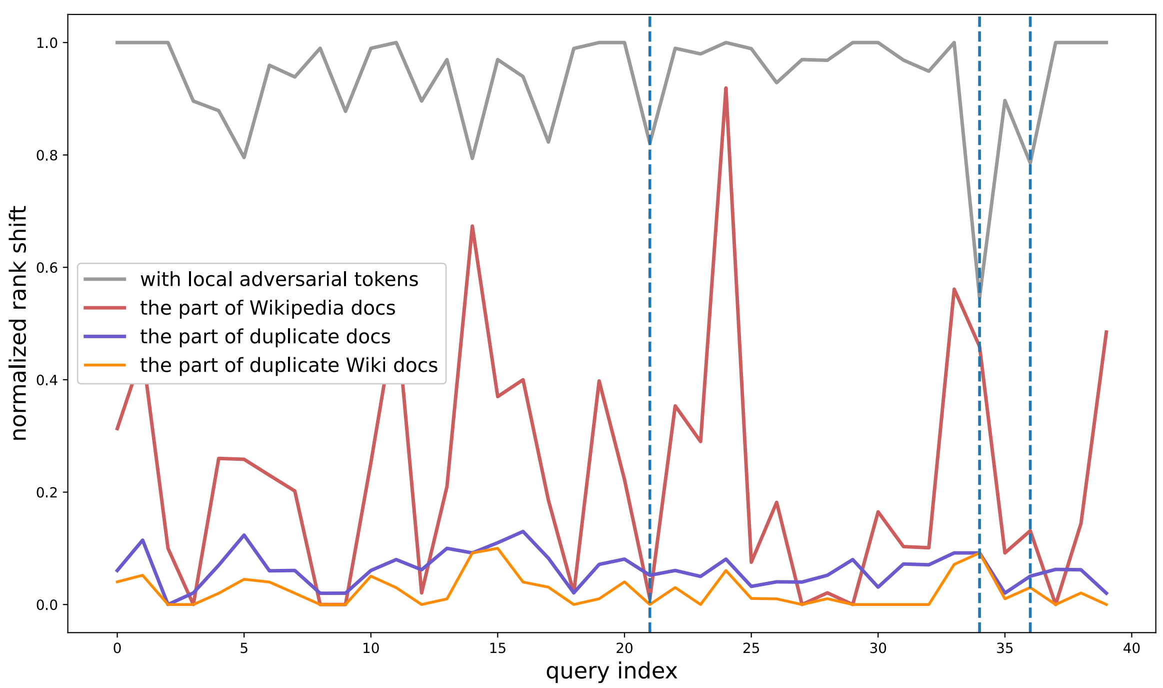


Figure 21: Wikipedia documents and duplicates documents proportion in rank demotion attack.

5.3.2.1 On rank demotion attack

For each query, we first get their respective best rank demotion results under the local adversarial attack, i.e., how much the top-ranked documents have dropped in rank, expressed as normalized rank shift. Figure 21 also shows how many of these dropped rankings are from Wikipedia articles, how many are the exact same articles, and how many are the exact same Wikipedia articles. The plot demonstrates that query indexes 21, 34, 36 have the lowest relative rank changes. We then explain the duplicates effect by these examples, specifically, we draw the Figure 22.

In the upper part of Figure 22 you can observe that there are a lot of Wikipedia articles, where a quarter of the queries have more than 40 docs are Wikipedia docs among the top-ranked 100 documents. Under some queries, like the $24th$ query, the Wikipedia document number even reaches 91. And these articles basically promote one rank place after the rank demotion attack. As one can imagine, the Wikipedia articles with duplicates rank basically higher than the attacked document after the attack. It can be assumed that these articles have some degree of similarity and the relevance scores should be relatively similar. In fact, due to the limitation of the input length of the BERT model, it makes more sense to consider only the part of the text string consisting of the first 512 tokens of the document. After observation, most of these Wikipedia articles share similar front text contents, taking query index 0 (query id:80) as an example, these articles basically start with

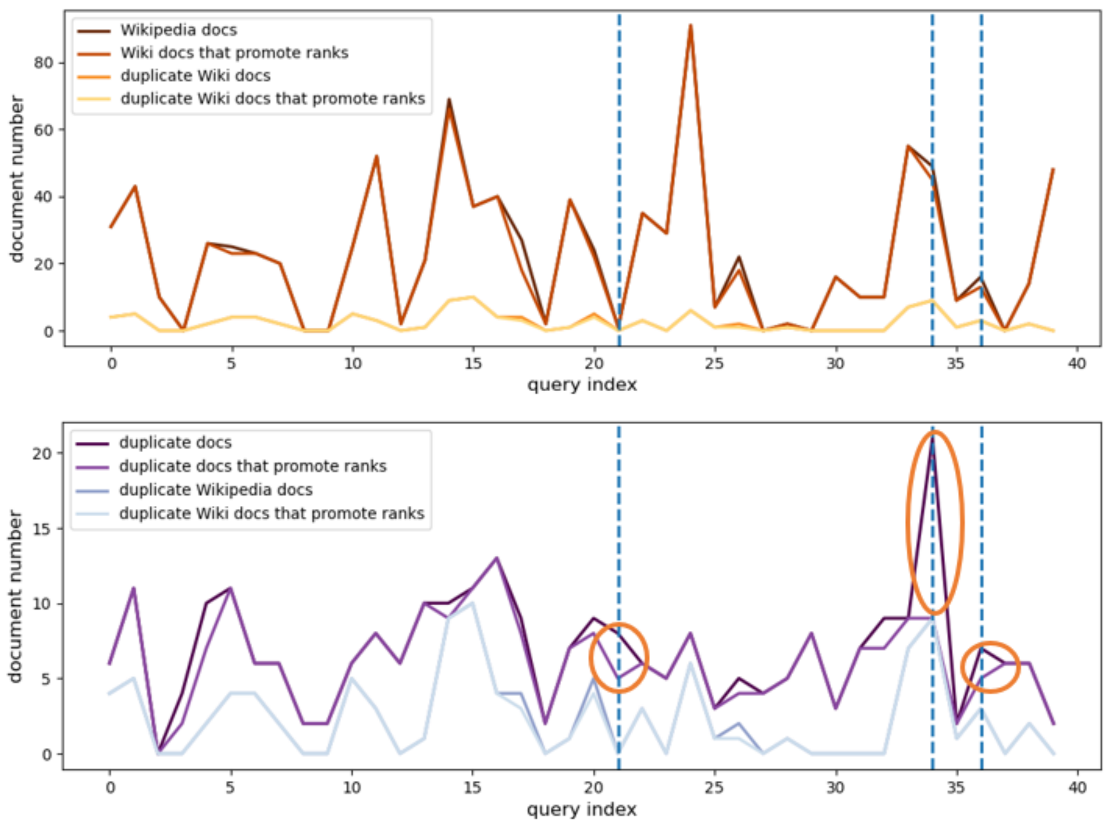


Figure 22: Number of Wikipedia and duplicate documents before & after rank demotion attack.

'keyboard (computing) - wikipedia, the free keyboard (computing) from wikipedia, the free encyclopedia (redirected from keyboard plaque) jump to : navigation, search this article needs......',

where '(redirected from ......)' varies depending on the articles. In addition, the rankings of these Wikipedia articles are very close or can present a cluster distribution, and they are not ranked very far back. Therefore it is easy for these articles to get a rank promotion after the attack.

The bottom part of the Figure shows that for most of the queries there are between 5 and 10 duplicate documents. Not all of the duplicates can be ranked higher compared to Wikipedia articles, but the two lines below indicate that most of the Wikipedia duplicate documents have managed to improve their rankings.

Then we investigate the rank changes of query indexes 21, 34, 36. What they have in common is that in the bottom part, there are quite a few duplicate documents that did not improve their rankings. For the query with index 34, only 9 documents in 21 duplications are promoted, and meanwhile, these 9 were all Wikipedia documents, and 45 rankings were raised in 49 wiki documents. If the other 12 documents could be ranked higher, a total of 22 documents with the same scores as them could be ranked higher, which would increase the relative rank changes from 0.54 to 0.76.

Similarly, for the query with index 36, only 4 of the 7 duplications were ranked higher, and if all the other duplications were ranked higher, there would be a total of 5 rank promotions. For query 21, 5 out of 8 duplicates are promoted, and if all other duplicates are also surpassed, the relative rank change will rise from 0.73 to 0.8.

The summary is that the queries with lower relative rank changes have more non-Wikipedia duplicate documents that do not get a ranking boost. If the ranking scores of the attacked documents can be degraded to lower than duplicates, then those documents that share the same ranking scores with the duplicates will also be improved. If we can exclude the interference of duplicates, the effect of the attack should be very similar for all queries and achieve more than 80% of the relative rank changes.

5.3.2.2 On rank promotion attack

For each query, the attack on rank promotion was also analyzed for documents redundancy. It can be seen from Figure 23 that after the last ranked document ranking up, there are fewer Wikipedia documents and fewer duplicate documents that participate in rank changes than those in rank demotion attacks. When we concentrate on the six worst performances presented by query indexes, 14, 15, 24, 26, 34, 39, it can be found that the main reasons for generalized inconspicuous promotion are

1. the same as the reasons for the lower results of rank demotion attack in the

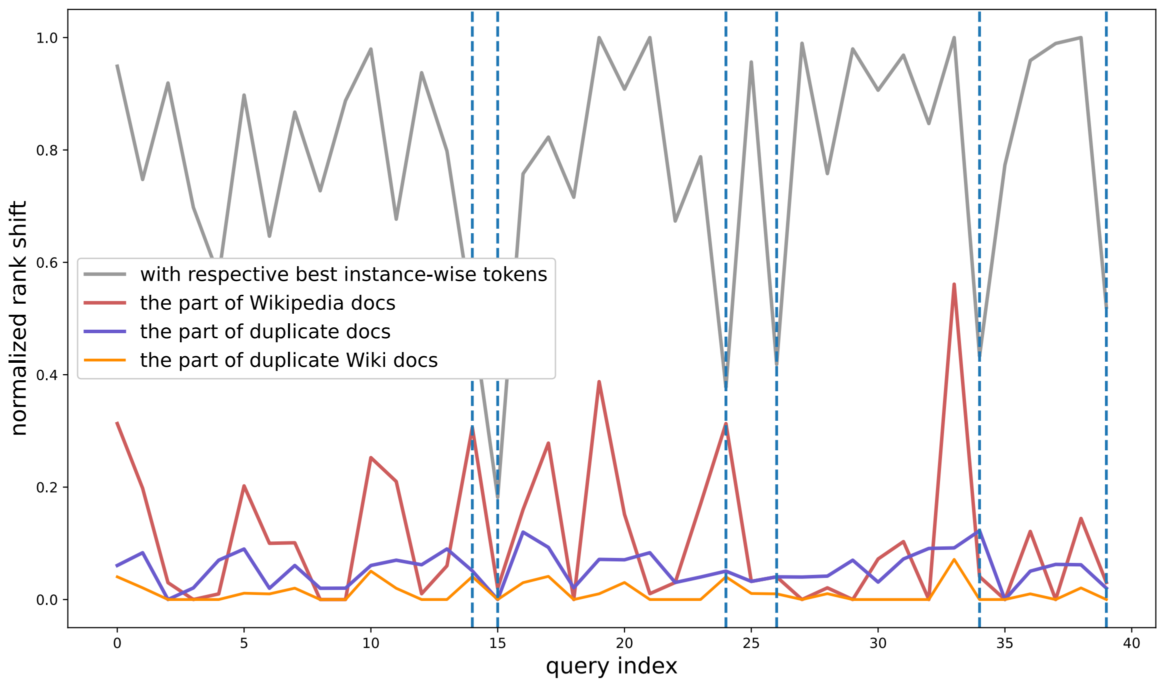


Figure 23: Wikipedia documents and duplicates documents proportion in rank promotion attack.

previous summary: the duplicate documents that can be surpassed after the attack are limited, but the difference is that these duplicates with unchanged rankings are mostly Wikipedia texts.

2. the second point is that generally speaking, there are a certain amount of non-duplicates Wikipedia documents that still remain in the front of the rankings and have not been surpassed. The reason for this is that many Wikipedia document rankings are front-loaded and share similar relevance scores, therefore it is difficult for the last ranked document to rise to position that can surpass so many Wikipedia documents at the same time.

In general, the situation is exactly the opposite of the rank demotion attack. The bottom documents encounter a large number of top-ranked Wikipedia documents with similar relevance scores as obstacles during climbing up, as a result, they are only able to surpass some of them before moving on. The situation is even worse if there are a lot of duplicate top-ranked Wikipedia documents, which leads to those less satisfactory results.

5.3.2.3 Easy achievement of rank demotion than rank promotion

Now we can answer the question:

Why rank demotion is easier to achieve than rank promotion?

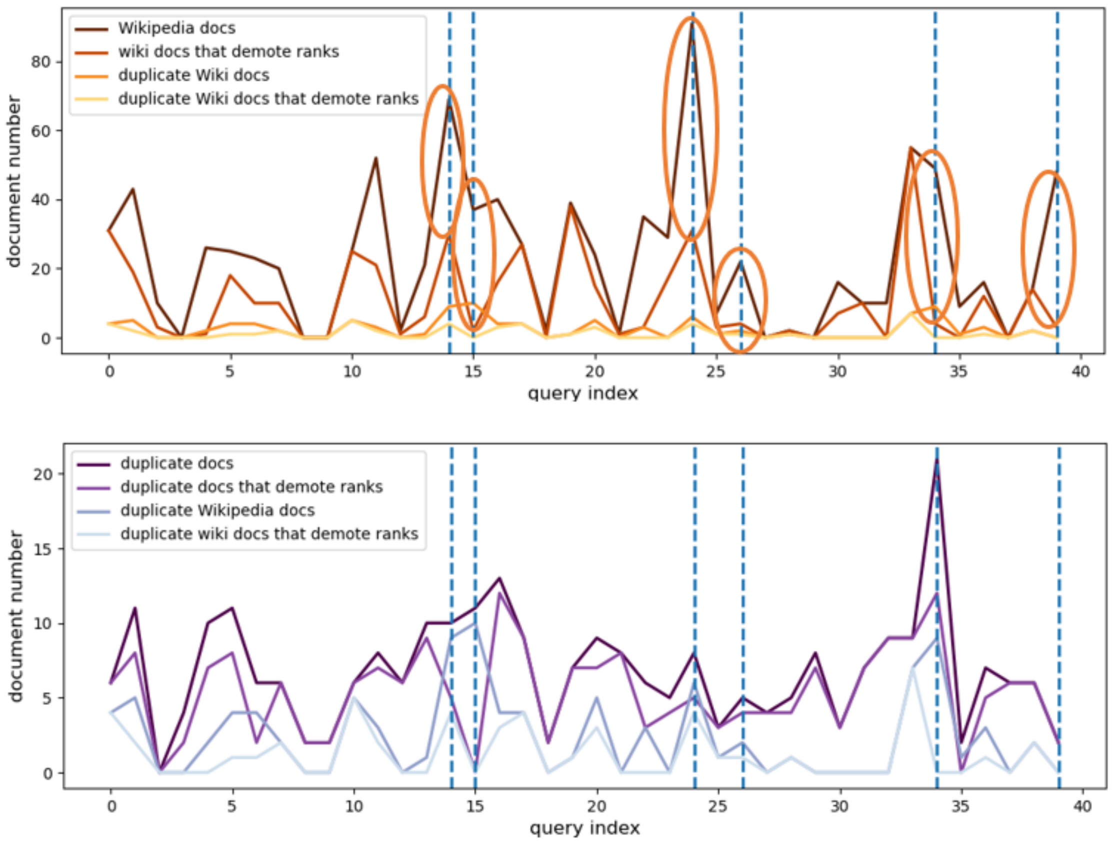


Figure 24: Number of Wikipedia and duplicate documents before & after rank promotion attack.

Based on the previous analysis on redundant documents, we suggest the hypothesis that rank demotion is easier to achieve than rank promotion due to the frustration of surpassing Wikipedia and duplicate documents.

Cluebwe09 (category B) contains a complete set of Wikipedia documents, and the selected top 100 documents also retain many Wikipedia documents, which are not ranked very low and share similar scores. Because the BERT model is trained based on Wikipedia documents, these documents are easy to get a rank boost due to the model preference when conducting rank demotion attacks.

In contrast, for rank promotion attacks, it is difficult to get the last document (often not a Wikipedia document) to have a higher rank score than the Wikipedia documents favored by the model. The similarity of relevance scores of these documents also poses a hindrance to the rank promotion attack.

5.3.3 Analysis in Word Embedding Level

Besides previous analysis, whether there are some similarities in the properties of those tokens obtained also arouses our interest. Due to the high dimensionality of BERT [8] word embeddings, we want to observe their characteristics in the word-embedding level from multiple perspectives. In this section we conduct a post-hoc analysis on tokens generated in rank demotion attack in terms of norm distance, cosine similarity and visualization with PCA respectively. We omit the analysis on rank promotion attack because we find that its analysis result shows consistency with rank demotion task.

In subsection 3.1.2 we mentioned that in order to reduce the error caused by randomness, we attacked a query six times, once by repeating [MASK] five times as the initial tokens and five times by randomly selecting five tokens from a specific range of the vocabulary as the initial tokens. We observed that there is no initial token sequence that always maintains best. The different initialized token sequences show consistency across different query attacks. In the following analysis, we will use the best attack performances among the six times as upper bound.

5.3.3.1 Distance-wise Analysis

Norm Distance BERT [8] has a word embedding layer with a shape of [30522, 768], which means it contains word embeddings for 30522 tokens with each in 768 dimensions. We analyzed at first 40 top-ranked documents each with 5 adversarial tokens in length. Each query is attacked 6 times. We record all the attacking results and pick out the best tokens which lead to the best attack performances.

For high-dimensional data, it is difficult to visually observe the characteristics of the

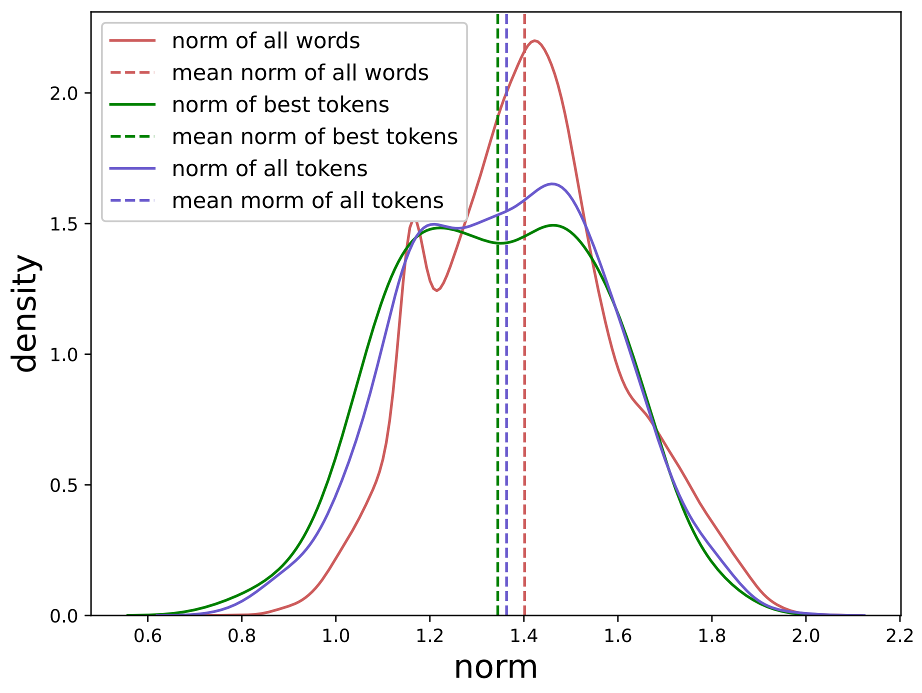


Figure 25: KDE of norm distributions for adversarial tokens and all vocabulary tokens

data directly from high dimensions. Therefore, we choose simple mapping functions or algorithms to map the data from a high dimensional vector space to the values in the lower dimensional space. Norm is such a function from a real or complex vector space to the non-negative real numbers that behave like distance from the origin. Following the Euclidean norm equation described in equation **??**, we calculate here Euclidean norm of word embeddings for:

1. all the tokens in the vocabulary set
2. all the best adversarial tokens among 6 times
3. all the adversarial tokens among 6 times

Figure 25 plots the norm distribution using Kernel Density Estimation (KDE), which represents the data using a continuous probability density curve in one dimension.

From Figure 25 we can observe that the norm of all best tokens has a lower mean value while a higher variance than those of all vocabulary tokens, which exhibits a distributed characteristic to some extent. Because the BERT model is context-based, the construction of word embeddings in each dimension does not differ as much as Glove, which can explain why the average values of the three sets of the norm distance are relatively similar with 1.3444, 1.3632, 1.4015 in ascending order.

We have proposed the hypothesis that these adversarial tokens would be capable of producing some similar properties at some levels, but it is not strongly verified in this perspective with very subtle difference.

Cosine similarity between tokens Norm is often used as a metric to measure the distance between two data. Since the analysis on the norm above did not strongly support a clustering characteristic, other distance metrics can be properly considered. Cosine similarity is widely used as a metric to measure document similarity in information retrieval and text mining. It measures the similarity between two vectors of inner product space, and it is defined as the cosine of the angle between two vectors. Referring to the equation **??**, the effect of the magnitude has been canceled and the similarity depends only on the angle of the two vectors. Therefore it will always range from $[-1, 1]$, where the similarity of two vectors pointing in the same direction is 1 and the similarity in the opposite direction is -1. A similarity of 0 indicates that the two vectors become orthogonal.

Using cosine similarity as a metric to measure the distance between tokens on word embeddings, we measured the similarity between the obtained adversarial tokens themselves, and random tokens, respectively. For each of the 40 queries, we keep the adversarial tokens that yield the best attack results. Since the length of a token sequence is set by default to 5, 200 tokens in total would be examined. Note that although among them there are repeated occurrences of some certain tokens which lead to unique 160 tokens, we do not strip out duplicates considering that the similarities between each token and other tokens belonging to the same sequence should be calculated as a baseline.

More specifically, we calculate the cosine similarity between every of 5 adversarial tokens for each of 40 queries (200 tokens in total) with following 4 groups:

1. other adversarial tokens in the same sequence named ‘sequence similarity’.
2. other adversarial tokens for all queries (199 in total) named ‘global similarity’.
3. random 199 tokens named ‘random similarity’.
4. all vocabulary tokens (30522 in total) named ‘vocab similarity’.

Figure 26 shows an example of a comparison of cosine similarity distributions of above mentioned groups for each adversarial token to a certain query. KDE is analogous to a histogram, aiming to approximate the underlying probability density function. But unlike histogram with discrete bins, KDE smooth the plot with a Gaussian kernel. The mean value of each KDE is drawn with a dashed line in Figure 26.

After careful review and comparison of all cosine similarities, it can be found that ‘random similarity’ always shows a very similar density distribution with ‘vocab similarity’ with their mean values always equal, which implicates the 200 randomly selected tokens can represent the distribution of the whole 30522 vocabulary tokens. The sample

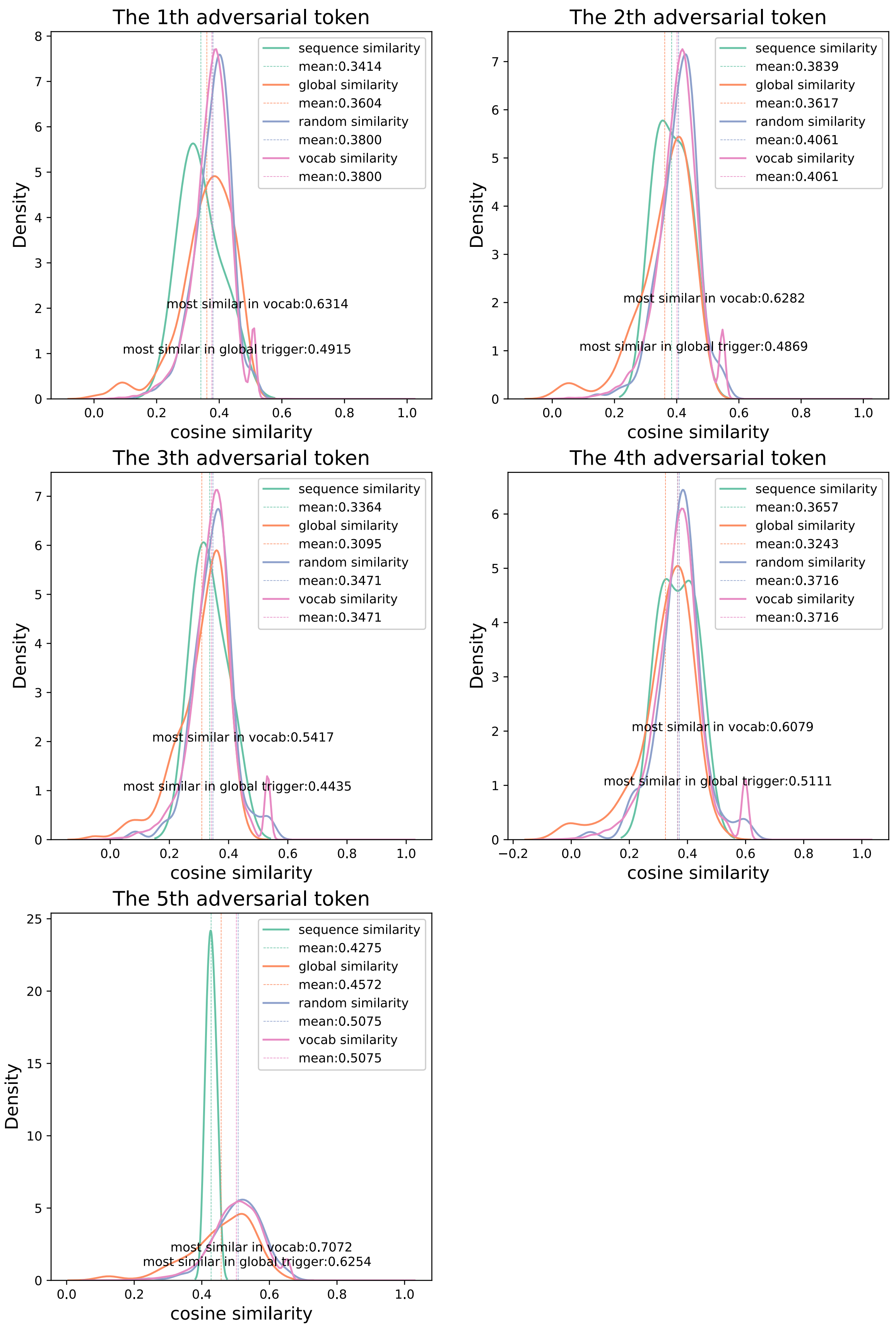


Figure 26: KDE of cosine similarity for each adversarial token to query 80: ‘keyboard review’ in rank demotion attack. Adversarial tokens are ‘acceptable’, ‘representation’, ‘way’, ‘##dington’ and ‘sport’ in order.

size of ‘sequence similarity’ is very small, with only 4 more tokens belonging to the same sequence, so it presents a very decentralized distribution, which has little reference value and thus is not considered here. Therefore, we propose a hypothesis that all ‘global similarity’ has a higher variance and lower mean relative to ‘random similarity’.

To further verify our conjecture, we calculated the mean and standard deviation for ‘global similarity’ and ‘random similarity’ for all the adversarial tokens to all queries.

The results show that the mean of ‘global similarity’ is consistently lower than the mean of ‘random similarity’, while the mean deviation of ‘global similarity’ is almost always greater than that of ‘random similarity’. This result is consistent with our conjecture. Then we get the conclusion that adversarial tokens differ more from each other in terms of directions than from random tokens.

Distance between tokens and their KNNs Whether comparing the norm distance or cosine similarity of word embeddings between tokens introduced in subsection 5.3.3.1, they are both category-based (adversarial token or non-adversarial token) computations, including the computation between the same category and computations between different categories. In terms of spatial distance, it is still possible to consider a single token as the base unit and calculate whether it is close enough to the nearest surrounding points so that it can be checked whether adversarial tokens are clustered in comparison to a random token.

The principle behind K Nearest Neighbors (KNNs) method is to find a predefined number of training samples closest in distance to a certain sample, and process clustering or classification task based on its task type. The distance can theoretically be set as any metric measure. Since we involved Euclidean norm and cosine similarity as metrics in subsection 5.3.3.1, Euclidean distance which is a common choice and cosine distance as well are considered here. Note that cosine distance is defined as 1.0 minus the cosine similarity, signifying that for both metrics a value close to zero indicates a close point in distance.

Table 2 compares average norm distance and cosine distance of adversarial tokens and randomly selected tokens to their k-nearest neighbors. Without prior knowledge of distributions on word embedding space, multiple k are chosen for neighbors queries.

The most striking result to emerge from the table is that the average distance scores between 732 adversarial tokens and their k-nearest neighbors are always slightly higher than those between 732 random tokens and their k-nearest neighbors at various values of k, no matter in Euclidean distance or in cosine distance. That means

Table 2: Comparison of the average distance of adversarial tokens and randomly selected tokens with their k-nearest neighbors in norm distance and cosine distance. The unique 732 adversarial tokens out of 1200 tokens obtained by the attack on 40 queries are involved in the calculation of the average, the same number to random tokens.

	Norm Distance		Cosine Distance	
Distance between	adversarial token and its KNNs	random token and its KNNs	adversarial token and its KNNs	random token and its KNNs
k= 5	0.8904	0.8770	0.2856	0.2657
k= 10	1.0323	1.0102	0.3466	0.3178
k= 20	1.1089	1.0821	0.3831	0.3478
k= 50	1.1600	1.1301	0.4096	0.3704
k= 100	1.1807	1.1496	0.4212	0.3805

adversarial tokens are more distant from the points in their vicinity than a randomly selected token. This contradicts our previous hypothesis that adversarial tokens are clustered in terms of distance. The table also illustrates that the distance between a point and its surrounding points always grows with k. This is consistent with common sense, since the radius of the surrounding circle centered at the token is increasing.

Taken together, this subsection summarizes the results of the distance-based study of adversarial tokens. **Norm distance** and **Cosine similarity between tokens** focus on the distance among adversarial tokens and the distance between adversarial tokens with non-adversary tokens. **Distance between tokens and their KNNs** discusses the distance of adversarial tokens and random tokens with respect to their surrounding neighbors. Neither of these two analytical perspectives can strongly demonstrate that tokens exhibit clustering in terms of distance, either in terms of Euclidean distance or cosine distance, but instead exhibit a trend of distancing from each other and not being close to surrounding points. In the next subsection, we apply visualization to word embeddings to low dimensional space for an intuitive observation.

5.3.3.2 Visualization in 2D

Both norm and cosine similarity are not intuitive in terms of word embedding analysis on distance, therefore we intend to visualize it in a more intuitive and straightforward way. Due to the high dimensionality of BERT word embeddings, we try to apply PCA and t-SNE on all words in vocabulary with 768 dimensions and reduce them to 2 dimensions.

PCA is a statistical procedure that is often used to linearly reduce the dimensionality of a large dataset. It uses singular value decomposition to project a large dataset from higher-dimensional space into a lower one trying to maintain as much information as possible. We apply the PCA function from the sklearn package on all the 30522 word

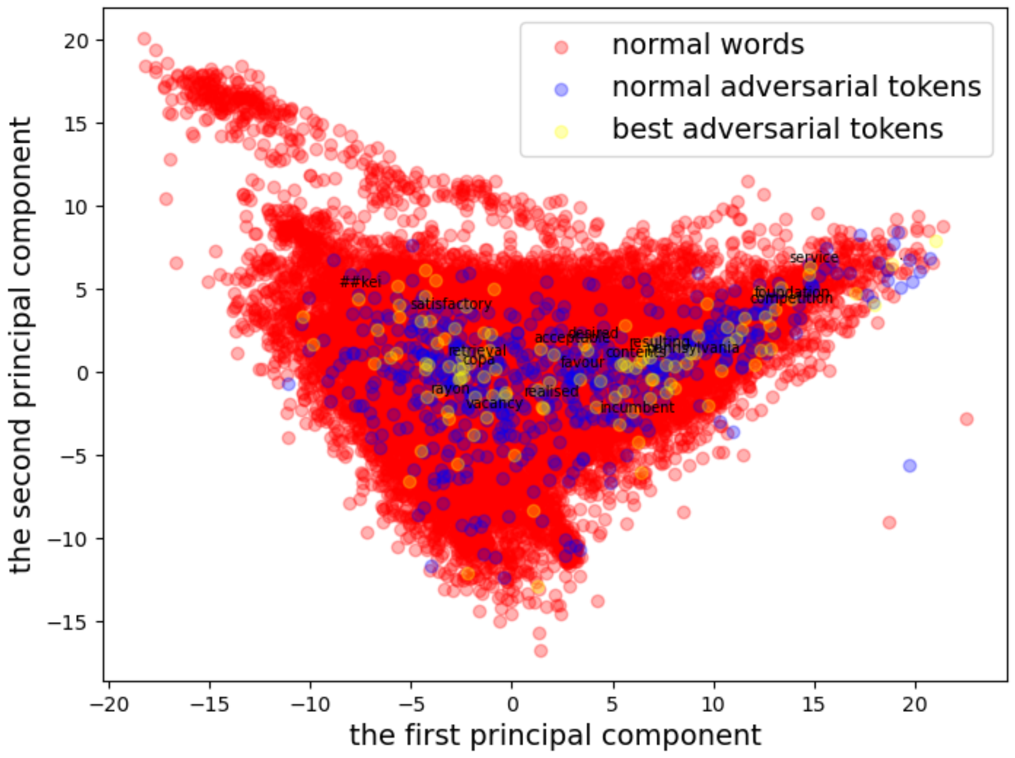


Figure 27: The first two principal components of PCA on word embeddings of adversarial tokens generated from rank demotion attack

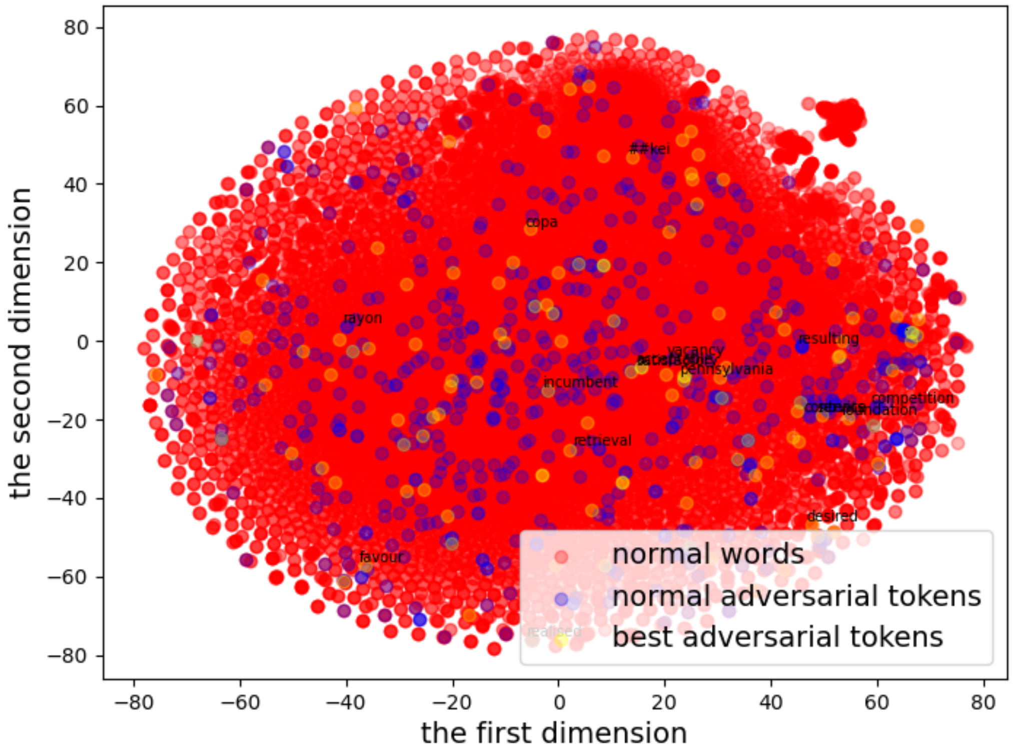


Figure 28: The first two reduced dimensions of t-SNE on word embeddings of adversarial tokens generated from rank demotion attack.

embeddings and keep the first two principal components. As visualized in Figure 27, adversarial tokens that give out the best performances are highlighted in yellow dots, while the adversarial other tokens are in blue dots, and the rest tokens in the vocabulary are in red dots. All the word embeddings together present a heart shape, while all adversarial tokens tend to cluster in the shape center and the best tokens among them are even more so. The texts shown in the figure are the most frequent adversarial tokens among all query attacks.

Figure 28 indicates corresponding t-SNE results. t-SNE is also an unsupervised dimensionality reduction algorithm, which is very suitable for visualizing high-dimensional data down to 2 or 3 dimensions. Differences between PCA and t-SNE exist mainly in that PCA is a linear method trying to maintain its information by maximizing variance and seeking to preserve large pairwise distances. For nonlinear structures, PCA may show bad performance. In contrast, t-SNE is a nonlinear dimension reduction strategy that preserves small pairwise distances by measuring the density of a Gaussian distribution over each point.

As Figure 28 indicates, unfortunately, t-SNE failed to demonstrate its strength to show a relatively clear separation in the data. The blue and yellow points representing tokens are distributed in various positions in the diagram. Whether it is PCA or t-SNE, 2 dimensions are still too small for the original 768 dimensions to visualize a clear segmentation.

6. Closure

6.1 Conclusion

In this thesis, we propose the gradient-based attacking algorithm on BERT ranking models for adversarial perturbations of highly relevant and highly non-relevant documents to a query. The algorithm aims to cause a rank demotion or a rank promotion of the document by concatenating adversarial tokens to it. We conducted experiments and analysis to investigate the following questions raised in subsection 1.2:

1. Are ranking models vulnerable to adversarial attacks? **(RQ1)**
2. How is the attack effectiveness when finetuned on different datasets, and whatexerts an influence on attack effectiveness? **(RQ2)**
3. If the adversarial attacks exhibit similarities or observable patterns across queries or across datasets in both attack scenarios.**(RQ3)**

RQ1 Our experiments show that a short adversarial token sequence can contribute to a significant rank change of a document, which indicates the vulnerability of neural ranker models.

RQ2 We summarized the overall attack effectiveness of the gradient-based attack approach on both ClueWeb09 and TREC-DL in 5.1.1. Local adversarial attacks achieve a better attack effect than global attacks with 70 to 62 rank changes on rank demotion and 73 to 61 on rank promotion respectively for ClueWeb09, which can be interpreted as a tradeoff in attack effectiveness for a better generalization. However, we found in global triggers the existence of frequently occurring adversarial tokens generated in local attacks, suggesting the global attack capability of these local adversarial tokens. Experimental results in subsections 5.1.2 and 5.1.3 present the effect of trigger positions and trigger sequence length on attack effectiveness on ClueWeb09. We observe a significant sensitivity of ranking models to document perturbations on the start position. Furthermore, increasing the length of the token sequence can continuously improve the attack effectiveness, however, at the expense of being highly susceptible to be detected. A perfect trade-off will be gained among token sequence length and attack effectiveness by taking model input length (512 tokens in our case) into consideration. In a real-life scenario, 5 tokens for rank demotion and 7 tokens for rank promotion can be practical on the attack against ClueWeb09.

RQ3 Section 5.3.3 suggests that the model has a bias towards tokens like 'acceptable', '.' and 'hapoel', which recur across multiple queries in a dataset. However, these tokens belong to no specific topic semantically. On the contrary, section 5.2.2 shows a considerable relevance of queries with their corresponding adversarial tokens on rank promotion attacks. Moreover, these local tokens are susceptible to be classified to topics like *nature* and *religion*. We also observe that local adversarial tokens recur across datasets. Although some nature-related and religion-related tokens are shared with both datasets, these topics are mainly occupied by ClueWeb09. Meanwhile, the model has a stronger preference for medicine topic in both local and global tokens for TREC-DL.

In our investigations in token analysis, we start with measurement distance in word embedding level in terms of norm distance and cosine similarity. Comparing distance between adversarial tokens and with their k-nearest neighbors are both involved. Neither of these two analytical perspectives can strongly demonstrate that tokens exhibit clustering in terms of distance, but instead they show a trend of distancing from each other and not being close to surrounding points.

Moreover, we conduct an analysis on document redundancy, primarily focusing on the influence of duplicate documents and Wikipedia documents on the attack performance. We suggest the explanation that due to the difficulty of surpassing Wikipedia and duplicate documents, the rank demotion is easier to achieve than the rank promotion.

To summarize, we apply the gradient-based algorithm for attacking the BERT-based ranking models. Our results suggest a minor perturbation on the text documents can cause a significant rank shift, revealing the vulnerability of BERT ranking models. Meanwhile, we prove that local attacks present a global attack capability across the dataset. Subsequent experiments also unveil potential model bias towards specific topics deriving from dataset and BERT pretraining.

6.2 Limitations and future work

Our presented work applied a gradient-based attack algorithm requiring white-box access to the model. The complete model information provides the attacker a wide operating platform to perform attack techniques, and therefore remarkable attack performance can often be achieved. However, real-world ranking systems, unfortunately, prohibit algorithmic white-box access, which hinders our method for production deployment.

To maximize the effectiveness of the attack, we place the adversarial tokens in front of the document. However, this increases the risk of being detected. Hence, we leave a thorough investigation to the future work for more perceptible and appropriate inserting or replacement positions. Syntactic and semantic constraints will be considered as well for a more concealed attack, but at the expense of the attack efficiency to some extent.

According to the intriguing findings in section 5.2, we hypothesize that the model presents a bias towards specific topic words due to the pre-training process. For future work, we also plan to include diverse models and datasets for extracting potential topic biases for large training corpus and ranking models.

References

[1] Daniel Arp et al. “Drebin: Efficient and Explainable Detection of Android Malware in Your Pocket”. In: *21th Annual Network and Distributed System Security Symposium (NDSS)* (Feb. 2014).

[2] Melika Behjati et al. “Universal adversarial attacks on text classifiers”. In: *ICASSP* (2019).

[3] B. Biggio, Fumera G., and Roli F. “Security evaluation of pattern classifiers under attack”. In: *Knowl. and Data Eng* 99(PrePrints).1 (2013).

[4] Battista Biggio et al. “Evasion attacks against machine learning at test time”. In: *ECML-PKDD.* (2013).

[5] Matthias Blohm et al. “Comparing Attention-Based Convolutional and Recurrent Neural Networks: Success and Limitations in Machine Reading Comprehension”. In: *In Proc. of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*. Brussels, Belgium, pp. 108–118.

[6] Samuel R Bowman et al. “A large annotated corpus for learning natural language inference”. In: *EMNLP* (2015).

[7] Tom B. Brown et al. “Adversarial patch”. In: *NIPS Machine Learning and Computer Security Workshop* (2017).

[8] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *https://arxiv.org/abs/1810.04805* (2019).

[9] Javid Ebrahimi et al. “Hotflip: White-box adversarial examples for text classification”. In: *arXiv preprint arXiv:1712.06751* (2017).

[10] Shi Feng et al. “Pathologies of neural models make interpretations difficult”. In: *EMNLP* (2018).

[11] Besnik Fetahu, Katja Markert, and Avishek Anand. “Automated News Suggestions for Populating Wikipedia Entity Pages”. In: *https://arxiv.org/abs/1703.10344* (2015).

[12] Zhitao Gong et al. “Adversarial Texts with Gradient Methods”. In: *arXiv preprint arXiv:1801.07175* (2018).

[13] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*, pp. 2672–2680.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *Proc. of the 3rd International Conference on Learning Representations (ICLR 2015)*.

[15] Gregory Goren et al. “Ranking robustness under adversarial document manipulations”. In: *In The 41st International ACM SIGIR Conference on Research Development in Information Retrieval* (2018), pp. 395–404.

[16] Alex Graves and Jürgen Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural Networks* (2005).

[17] Kathrin Grosse et al. “Adversarial Examples for Malware Detection”. In: *Proc. of the 22nd European Symposium on Research in Computer Security (ESORICS 2017)*. Oslo, Norway, 2017, pp. 62–79.

[18] Kathrin Grosse et al. “Adversarial perturbations against deep neural networks for malware classification”. In: *arXiv preprint arXiv:1606.04435* (2016).

[19] Daniel Lowd Javid Ebrahimi and Dejing Dou. [n. d.] “On Adversarial Examples for Character-Level Neural Machine Translation”. In: *Proc. of the 27th International Conference on Computational Linguistics (COLING 2018)*. Santa Fe, New Mexico, USA, pp. 653–663.

[20] Robin Jia and Percy Liang. “Adversarial Examples for Evaluating Reading Comprehension Systems”. In: *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Copenhagen, Denmark, pp. 2021–2031.

[21] Di Jin et al. “Is bert really robust? natural language attack on text classification and entailment”. In: *arXiv preprint arXiv:1907.11932* 2 (2019).

[22] Dianqi Li et al. “Contextualized perturbation for textual adversarial attack”. In: *arXiv preprint arXiv:2009.07502* (2020).

[23] Linyang Li et al. “Bert-attack: Adversarial attack against bert using bert”. In: *arXiv preprint arXiv:2009.07502* (2020).

[24] Bin Liang et al. “Deep Text Classification Can be Fooled”. In: *arXiv preprint arXiv:1704.08006* (2017).

[25] Jieyu Lin, Jiajie Zou, and Nai Ding. “Using Adversarial Attacks to Reveal the Statistical Bias in Machine Reading Comprehension Models”. In: *https://arxiv.org/abs/2105.11136* (2021).

[26] Tomas Mikolov et al. “Advances in pre-training distributed word representations”. In: *LREC* (2018).

[27] Seyed-Mohsen Moosavi-Dezfooli et al. “Universal adversarial perturbations”. In: *In CVPR* (2017).

[28] Nicolas Papernot et al. “The Limitations of Deep Learning in Adversarial Settings”. In: *In IEEE European Symposium on Security and Privacy (EuroSP 2016)* (2016), pp. 372–387.

[29] Ankur P Parikh et al. “A decomposable attention model for natural language inference”. In: *EMNLP* (2016).

[30] J. Pennington, R. Socher, and C. D. Manning. “Glove: Global vectors for word representation”. In: *In Empirical Methods in Natural Language Processing (EMNLP)* (2014).

[31] Matthew E. Peters et al. “Deep contextualized word representations”. In: *NAACL* (2018).

[32] Nisarg Raval and Manisha Verma. “One word at a time: adversarial attacks on retrieval models”. In: *https://arxiv.org/abs/2008.02197* (2020).

[33] Shuhuai Ren et al. “Generating natural language adversarial examples through probability weighted word saliency”. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*. 2019, pp. 1085–1097.

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *https://arxi v.org/abs/1602.04938* (2016).

[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Semantically equivalent adversarial rules for debugging NLP models”. In: *ACL* (2018).

[36] Jaspreet Singh and Avishek Anand. “EXS: Explainable Search Using Local Model Agnostic Interpretability”. In: *https://arxiv.org/abs/1809.03857* (2019).

[37] Jaspreet Singh and Avishek Anand. “Model Agnostic Interpretability of Rankers via Intent Modelling”. In: *Proc. of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 618–628.

[38] Jaspreet Singh and Avishek Anand. “Posthoc Interpretability of Learning to Rank Models using Secondary Training Data”. In: *https://arxiv.org/ab s/1806.11330* (2018).

[39] Mengying Sun et al. “Identify Susceptible Locations in Medical Records via Adversarial Attacks on Deep Predictive Models”. In: *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD 2018)*. London, UK, pp. 793–801.

[40] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *Proc. of the 2nd International Conference on Learning Representations (ICLR 2014)*.

[41] Eric Wallace et al. “Concealed Data Poisoning Attacks on NLP Models”. In: *North American Chapter of the Association for Computational Linguistics*. 2021.

[42] Eric Wallace et al. “Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering”. In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 387–401.

[43] Eric Wallace et al. “Universal adversarial triggers for attacking and analyzing NLP”. In: *arXiv preprint arXiv:1908.07125* (2019).

[44] Yicheng Wang and Mohit Bansal. “Robust Machine Comprehension Models via Adversarial Training”. In: *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. New Orleans, Louisiana, 2018, pp. 575–581.

[45] Wei Emma Zhang et al. “Adversarial attacks on deep-learning models in natural language processing: A survey”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.3 (2020), pp. 1–41.

Declaration of Authorship

I hereby declare that I have written the present thesis with the title Global Triggers for Attacking and Analyzing Ranking Models independently and without the use of other than the indicated aids. I affirm that I have not used any sources other than those indicated and that all passages taken verbatim or in spirit from published and unpublished writings are identified as such. Furthermore, I assure that the work has not yet been submitted in the same or similar form in the context of another examination. I am aware that in the event of deception, the thesis will be graded as ”failed”.

10.03.2022 Hannover

date, place

Yumeng Wang

signature