Robust Information Retrieval

SIGIR 2024 tutorial

Yu-An Liu[a,b], Ruqing Zhang[a,b], Jiafeng Guo[a,b] and **Maarten de Rijke**[c]

https://sigir2024-robust-information-retrieval.github.io/

July 14, 2024

01:30 – 05:00 PM

[a] Institute of Computing Technology, Chinese Academy of Sciences
[b] University of Chinese Academy of Sciences
[c] University of Amsterdam

Section 3:
Adversarial robustness

Revisit the definition of adversarial robustness

Ability of Neural IR models to maintain Top-K ranking performance when subjected to adversarial attacks.

Definition (Adversarial robustness in information retrieval)

Given an IR model $f_{\mathcal{D}_{\text{train}}}$ trained on training dataset $\mathcal{D}_{\text{train}}$ with a corresponding testing dataset $\mathcal{D}_{\text{test}}$, a new document set D_{adv} containing adversarial examples, and an acceptable error threshold δ, for the top-K ranking result, if

$$\left|\mathcal{R}_M\left(f_{\mathcal{D}_{\text{train}}}; \mathcal{D}_{\text{test}}, K\right) - \mathcal{R}_M\left(f_{\mathcal{D}_{\text{train}}}; \mathcal{D}'_{\text{test}}, K\right)\right| \leq \delta \text{ such that } \mathcal{D}'_{\text{test}} \leftarrow \mathcal{D}_{\text{test}} \cup D_{\text{adv}},$$

where $\mathcal{D}_{\text{test}} \cup D_{\text{adv}}$ denotes injecting the set of all generated adversarial examples D_{adv} into the original test dataset, and then model f is considered δ-robust against adversarial examples for metric M.

Background: Competitive search

Search engine is a competitive scenario, content providers may aim to promote their products or documents in rankings for specific queries [Kurland and Tennenholtz, 2022]

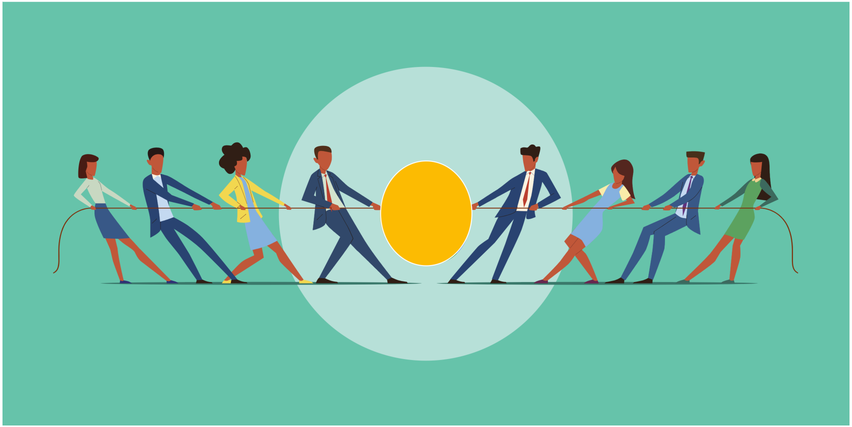
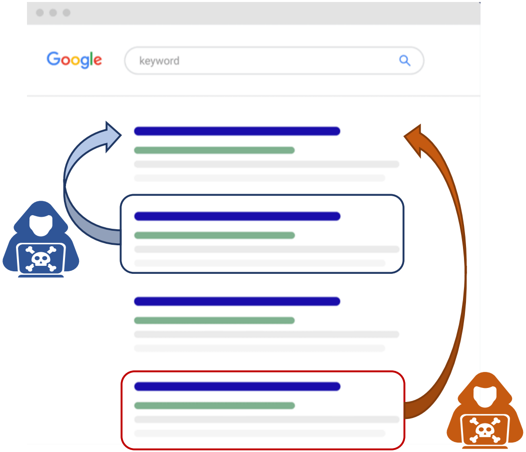


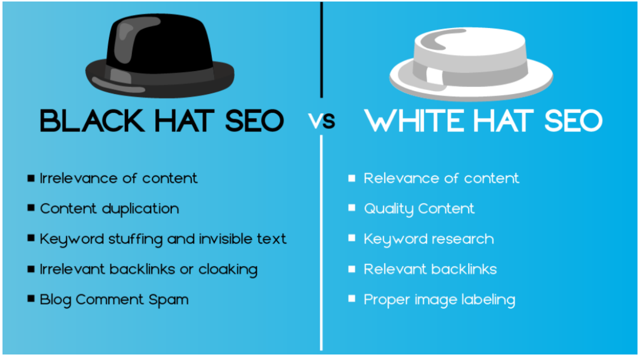
Image: Everything You Need to Know About Competition in Google Ads

Background: Competitive search

Competitive search scenario leds to the development for search engine optimization (SEO) and attack techniques against search engines [Gyöngyi and Garcia-Molina, 2005]

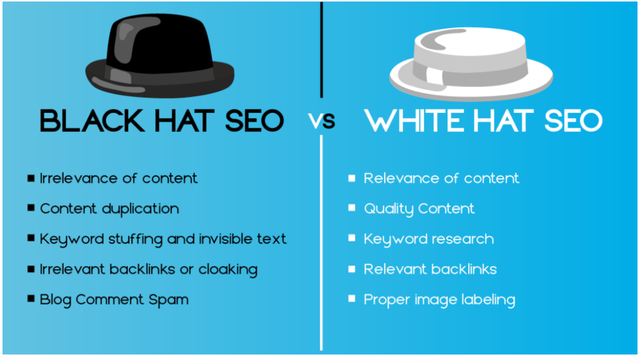


Black-hat SEO vs. White-hat SEO



"Web Spam Taxonomy" [Gyöngyi and Garcia-Molina, 2005]

Black-hat SEO vs. White-hat SEO

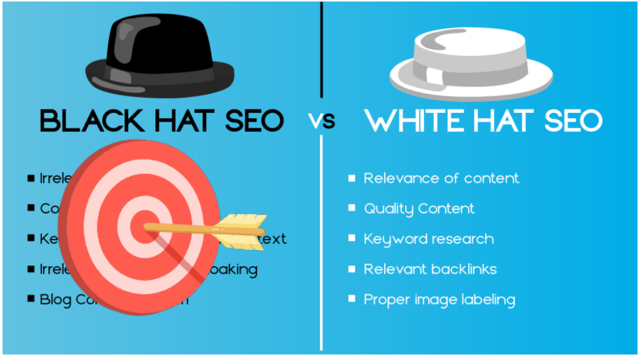


White-hat SEO optimizes the quality of web pages within the rules of search engines

Black-hat SEO maliciously modifies web pages by exploiting search engine loopholes

"Web Spam Taxonomy" [Gyöngyi and Garcia-Molina, 2005]

Black-hat SEO vs. White-hat SEO



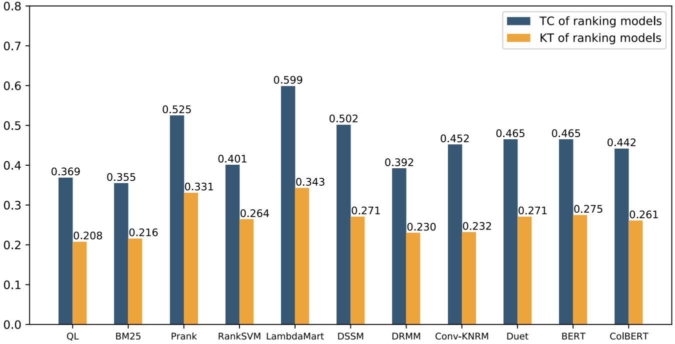
White-hat SEO optimizes the quality of web pages within the rules of search engines

Black-hat SEO maliciously modifies web pages by exploiting search engine loopholes

"Web Spam Taxonomy" [Gyöngyi and Garcia-Molina, 2005]

The vulnerability of IR models

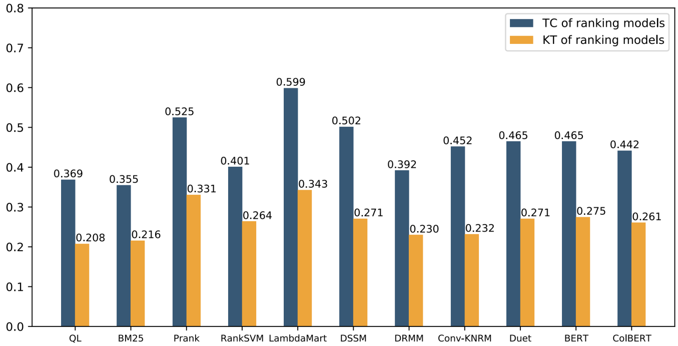
Our team found IR models are vulnerable in black-hat SEO scenarios [Wu et al., 2022b]:



- Dataset: ASRC
- Metrics:
  TC: Change of the top-1
  KT: Change of the ranked list

The vulnerability of IR models

Our team found IR models are vulnerable in black-hat SEO scenarios [Wu et al., 2022b]:



- Dataset: ASRC
- Metrics:
  TC: Change of the top-1
  KT: Change of the ranked list

Vulnerability (red color indicates neural IR models):

DSSM > BERT > Conv-KNRM > ColBERT > RankSVM > DRMM > QL > BM25

How to improve the adversarial robustness of neural IR models?

Two perspectives of adversarial robustness

Robustness is enhanced during the competition between attacks and defenses

Two perspectives of adversarial robustness

Robustness is enhanced during the competition between attacks and defenses

- **Adversarial attacks**: Identify the vulnerability of neural IR models
- **Adversarial defenses**: Improve the adversarial robustness of neural IR models

Outline

We will introduce the adversarial robustness through:

- **Benchmarks & settings**
- **Adversarial attacks**
- **Adversarial defenses**

Adversarial robustness: Benchmarks

- **Basic datasets**: Original datasets in IR that are adapted for reuse by attack and defense methods, e.g., MS MARCO and Clueweb09-B

- **Basic datasets**: Original datasets in IR that are adapted for reuse by attack and defense methods, e.g., MS MARCO and Clueweb09-B
- **Expansion of dataset**: Additional data provided by competitions, e.g., TREC DL19 and TREC DL20, are used for evaluation against the basic datasets

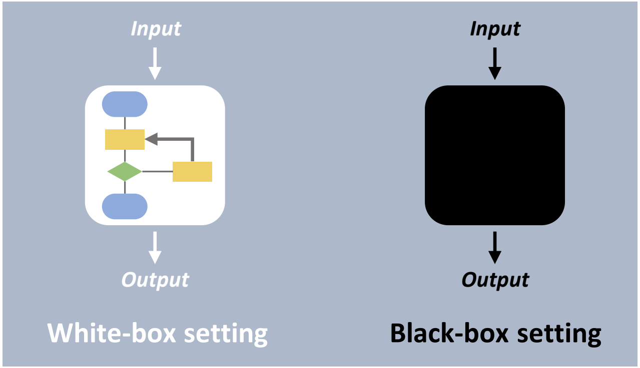
Adversarial robustness: Benchmarks

- **Basic datasets**: Original datasets in IR that are adapted for reuse by attack and defense methods, e.g., MS MARCO and Clueweb09-B
- **Expansion of dataset**: Additional data provided by competitions, e.g., TREC DL19 and TREC DL20, are used for evaluation against the basic datasets
- **Tailored datasets**: Datasets specially tailored for adversarial attacks and defenses, e.g., ASRC and DARA

Adversarial robustness: Benchmarks

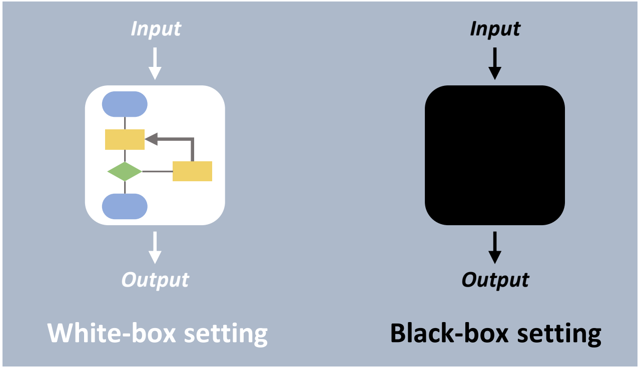
Type	Dataset	#Document	$\#Q_{\text{train}}$	$\#Q_{\text{dev}}$	$\#Q_{\text{eval}}$
Basic datasets	MS MARCO Doc [Nguyen et al., 2016]	3.2M	370K	5,193	5,793
	MS MARCO Pas [Nguyen et al., 2016]	8.8M	500K	6,980	6,837
	Clueweb09-B [Clarke et al., 2009]	50M	150	-	-
	NQ [Kwiatkowski et al., 2019]	21M	60K	8.8k	3.6k
	TriviaQA [Joshi et al., 2017]	21M	60K	8.8K	11.3K
Dataset expansion	TREC DL19 [Craswell et al., 2020]	-	-	43	-
	TREC DL20 [Craswell et al., 2021]	-	-	54	-
	TREC MB14 [Lin et al., 2013]	-	-	50	-
Tailored datasets	ASRC [Raifer et al., 2017]	1,279	-	31	-
	Q-MS MARCO [Liu et al., 2023b]	-	-	4,000	-
	Q-Clueweb09 [Liu et al., 2023b]	-	-	292	-
	DARA [Chen et al., 2023b]	164k	50k	3,490	3,489

Adversarial robustness: Settings



- **White-box setting**: attackers can fully access the model parameters and leverage the target model gradient to directly generate perturbations
- **Black-box setting**: attackers can only obtain the output by querying the target model, without having access to the internal parameters or gradients

Adversarial robustness: Settings



Considering real-world applications, existing work pays more attention on the more practical and challenging black-box setting

Traditional web spamming

Web spamming: any form of search engine ranking manipulation without regard to any value for the user

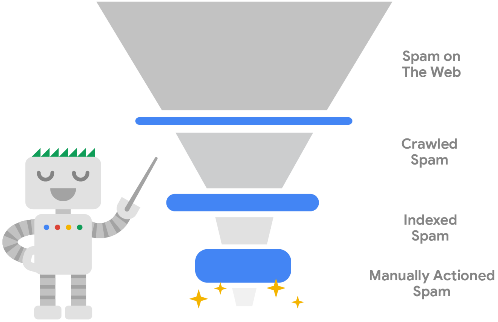
The main forms include:

- **Keyword stuffing** →
- Excessive links
- Sneaky redirects
- Phishing
- ...

Query: *What's the best resort in Washington?*
Spammy web site: *The Capitol Grand Hotel offers acomfort, and best resort best resort best resort. Just steps away from iconic landmarks such as Washington Washington Washington, this prestigious hotel is perfect for both leisure and business travelers. The Capitol Grand features best best best resort resort resort including high-speed internet.*

Traditional web spamming is . . .

- **Easily detected**
  - Major search engines said to automatically discover over 40 billion spammy pages per day, which may keep more than 99% of visits completely without spam
- **Mainly targeted at traditional IR models**
  - Spamming methods pose a limited threat in the age of neural models



Source: How we fought spam on Google Search in 2022

How to perform adversarial attacks against neural IR models to expose their vulnerabilities?

Requirements of adversarial attacks

Inspired by black-hat SEO, given a **low-ranked target document**, the requirements of adversarial attacks in IR include:

- Identifying gradient vulnerabilities of neural IR models on the target document
- Perturbing the target document in a human-imperceptible way
- Maximizing ranking improvement of the target document in the Top-K results

Definition of adversarial attacks

Given:

- a neural IR model f and a query q, and
- a top-K ranked list and a low-ranked target document d.

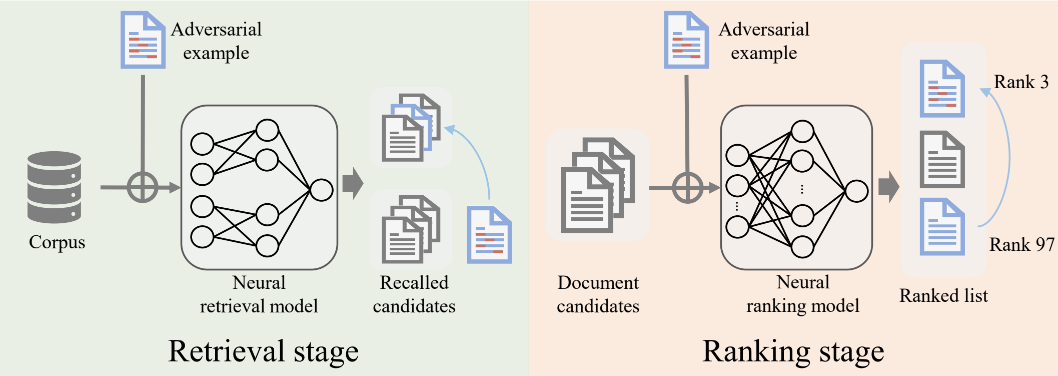
The goal is to improve the ranking of d under q with human-imperceptible perturbations p:

$$\max_{p} \left(K - \boxed{\pi_f \left(q, d \oplus p \right)} + \lambda \cdot \boxed{\mathrm{Sim} \left(d, d \oplus p \right)} \right),$$

It consists of two parts:

- Minimize the ranking position of the perturbed document $d \oplus p$
- Maximize the similarity between the perturbed $d \oplus p$ and original document d

Classification of adversarial attacks



- **Adversarial retrieval attack** retrieves a target document outside the top-K candidates to appear among the top-K candidates in response to a query
- **Adversarial ranking attack** promotes the target document in rankings in the top-K candidates with respect to a query

Adversarial retrieval attack

The definition of adversarial retrieval attacks can be formalized as:

$$\max_{p} \left(K - \text{Recall}_f \left(q, d \oplus p\right) + \lambda \cdot \text{Sim}\left(d, d \oplus p\right)\right),$$

where $\text{Recall}_f\left(q, d \oplus p\right)$ denotes the recalled position of the perturbed document $d \oplus p$ generated by the dense retieval model f with respect to query q given the entire corpus

The low-ranked target document d is out of the Top-K results

Adversarial ranking attack

The definition of adversarial ranking attacks can be formalized as:

$$\max_{p} \left(K - \boxed{\text{Rank}_f\left(q, d \oplus p\right)} + \lambda \cdot \text{Sim}\left(d, d \oplus p\right) \right),$$

where $\text{Rank}_f\left(q, d \oplus p\right)$ denotes the ranking position of the perturbed document $d \oplus p$ in the final ranked list generated by the neural retrieval model f with respect to query q

The low-ranked target document d is in the Top-K results

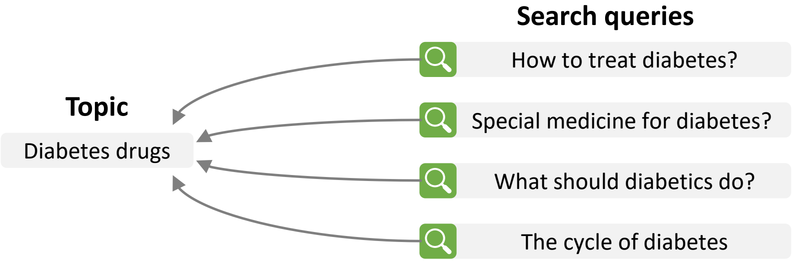
Topic-oriented adversarial retrieval/ranking attack

Web page owners usually expect their content to have a general advantage in ranked lists for for queries under the same search intent

Topic-oriented adversarial retrieval/ranking attack

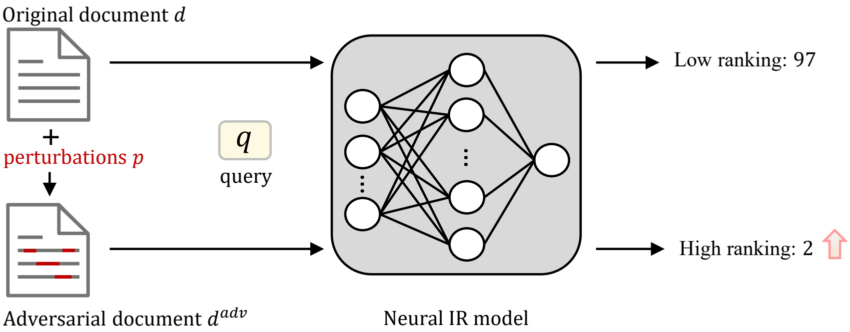
Web page owners usually expect their content to have a general advantage in ranked lists for for queries under the same search intent

In **paid search advertising**, when advertisers create an advertisement, they select a set of keywords for a group of target queries with the same topic:



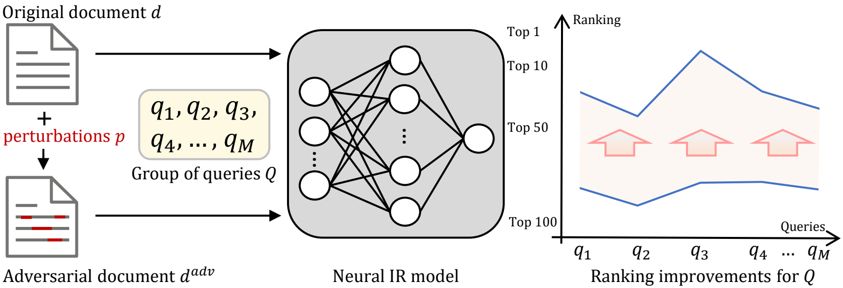
Topic-oriented adversarial retrieval/ranking attack

Paired attack promotes a target document in rankings w.r.t. a specific query

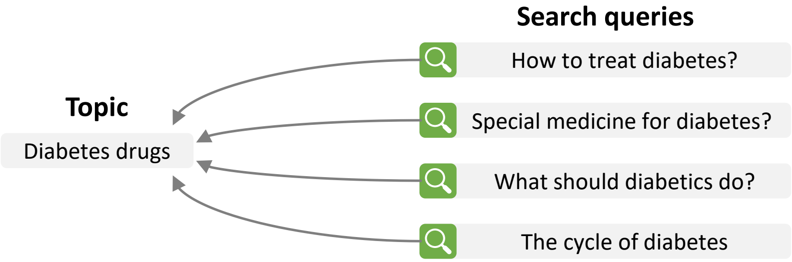


Topic-oriented adversarial retrieval/ranking attack

Topic-oriented attack promotes a target document in rankings on each query in the group with the same topic



Topic-oriented adversarial retrieval/ranking attack



"Advantages" of topic-oriented attack:

- **Meet the needs of realistic SEO**
- **More challenging than paired attack**
- **Identifying the generic vulnerability of neural IR models**

Key steps of adversarial attacks

Steal knowledge from black-box models

Key steps of adversarial attacks

Surrogate model training
(Wu et al. 2023)

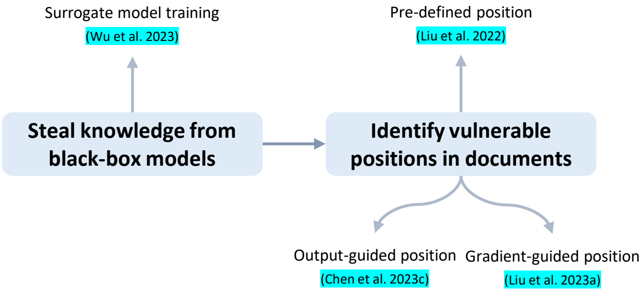
Steal knowledge from black-box models

Key steps of adversarial attacks

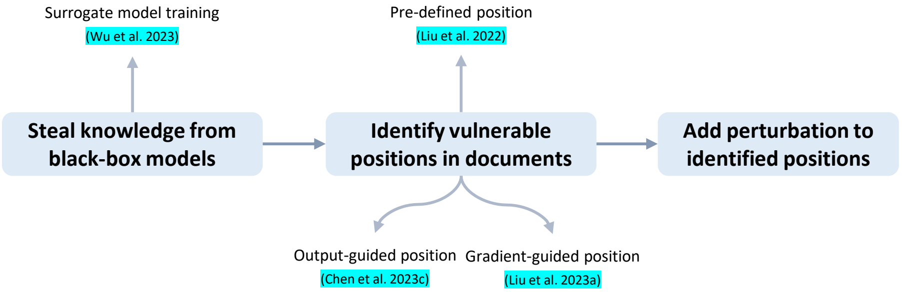
Surrogate model training
(Wu et al. 2023)

Steal knowledge from black-box models → **Identify vulnerable positions in documents**

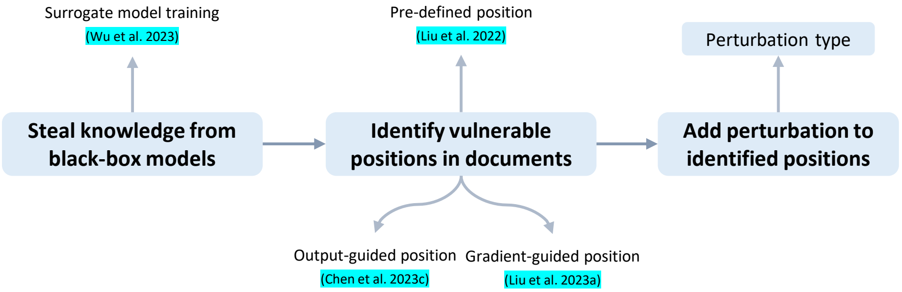
Key steps of adversarial attacks



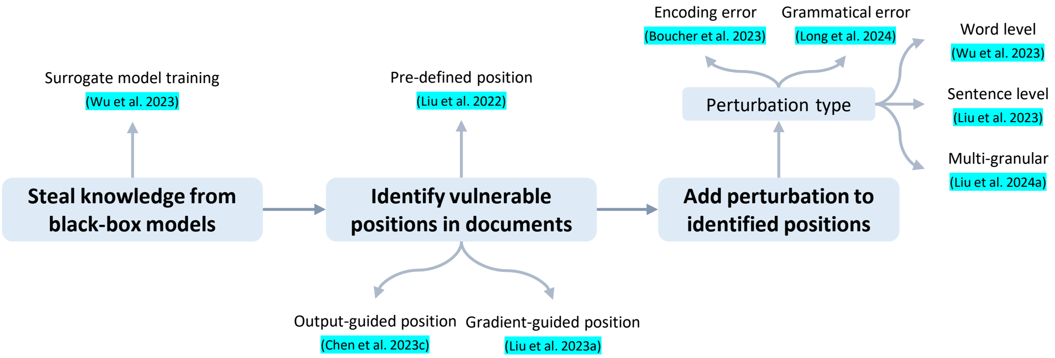
Key steps of adversarial attacks



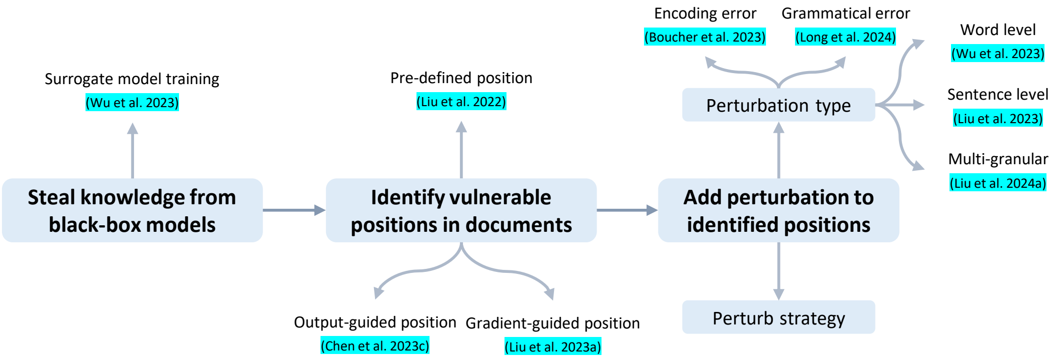
Key steps of adversarial attacks



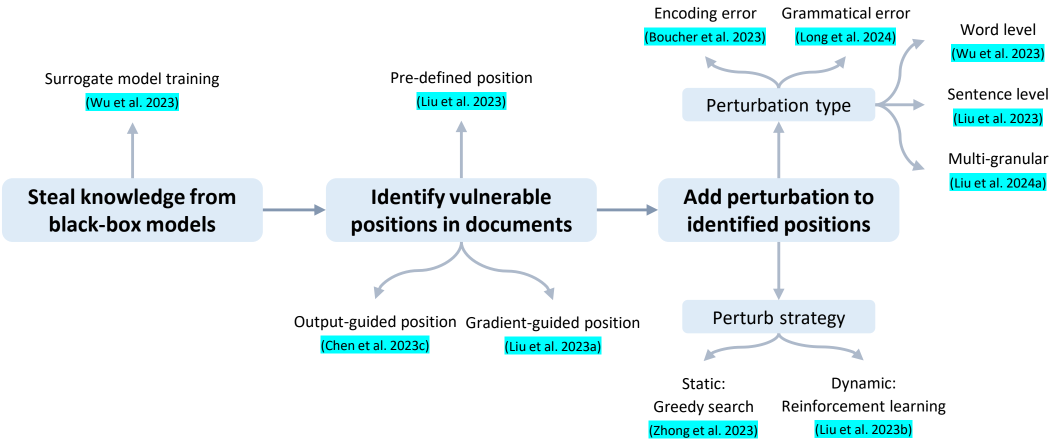
Key steps of adversarial attacks



Key steps of adversarial attacks



Key steps of adversarial attacks

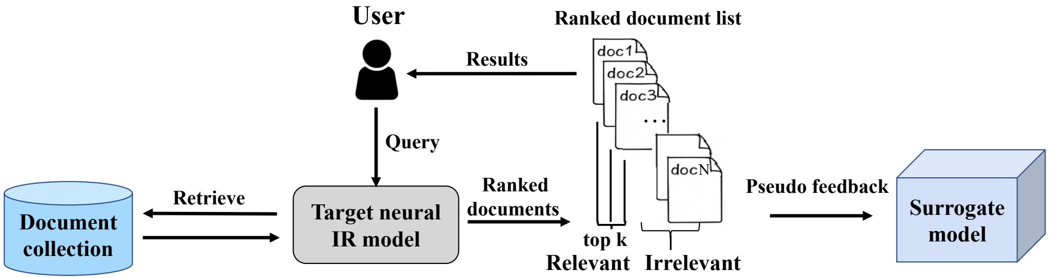


Steal knowledge from black-box models

Steal knowledge from black-box models → Identify vulnerable positions in documents → Add perturbation to identified positions

Steal knowledge from black-box models: Surrogate model training

- **Objective:** Training a surrogate white-box model to steal target model knowledge
- **Approach:** Continuously querying the target model and obtaining its outputs



Steal knowledge from black-box models: Surrogate model training

Take the idea of pseudo-relevance feedback:

- **Given:** a query collection $\mathcal{Q}$, a taget model f

"PRADA: Practical Black-Box Adversarial Attacks Against Neural Ranking Models" [Wu et al., 2023]

Steal knowledge from black-box models: Surrogate model training

Take the idea of pseudo-relevance feedback:

- **Given:** a query collection $\mathcal{Q}$, a taget model f
- **Get:** a rank list L returned by the target model

"PRADA: Practical Black-Box Adversarial Attacks Against Neural Ranking Models" [Wu et al., 2023]

Steal knowledge from black-box models: Surrogate model training

Take the idea of pseudo-relevance feedback:

- **Given:** a query collection $\mathcal{Q}$, a taget model f
- **Get:** a rank list L returned by the target model
- **Pseudo-labels:** take the top-k ranked documents $L[:k]$ as relevant documents and the other documents $L[k+1:N]$ as irrelevant documents

"PRADA: Practical Black-Box Adversarial Attacks Against Neural Ranking Models" [Wu et al., 2023]

Steal knowledge from black-box models: Surrogate model training

Take the idea of pseudo-relevance feedback:

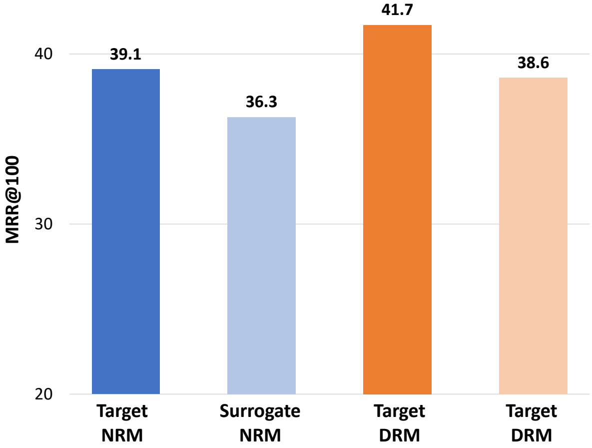
- **Given:** a query collection $\mathcal{Q}$, a taget model f
- **Get:** a rank list L returned by the target model
- **Pseudo-labels:** take the top-k ranked documents $L[:k]$ as relevant documents and the other documents $L[k+1:N]$ as irrelevant documents
- **Pair-wise training:**

$$\mathcal{L} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \max(0, \eta - \tilde{f}(q, L[:k]) + \tilde{f}(q, L[k+1:N])),$$

Finally, we get surrogate model $\tilde{f}$ that can imitate the performance of target model

"PRADA: Practical Black-Box Adversarial Attacks Against Neural Ranking Models" [Wu et al., 2023]

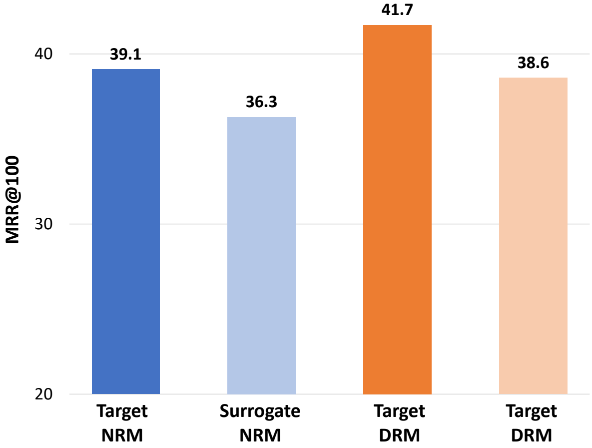
Steal knowledge from black-box models: Surrogate model training



- Dataset: MS MARCO
- Backbone:
  Target NRM: PROP
  Surrogate NRM: BERT-cross encoder
  Target DRM: CoCondenser
  Surrogate DRM: BERT-encoder

Data source: [Liu et al., 2023a,b]

Steal knowledge from black-box models: Surrogate model training

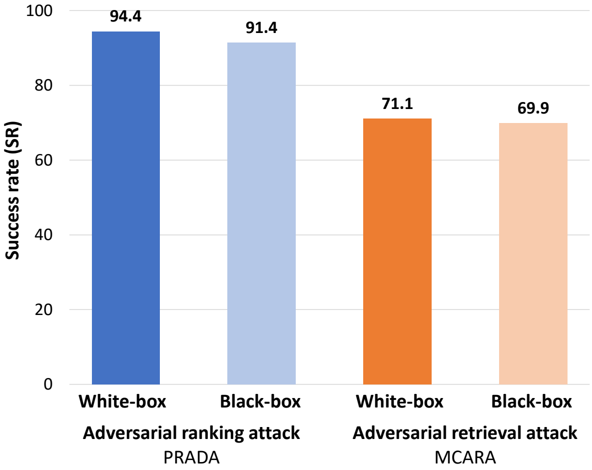


- Dataset: MS MARCO
- Backbone:
  Target NRM: PROP
  Surrogate NRM: BERT-cross encoder
  Target DRM: CoCondenser
  Surrogate DRM: BERT-encoder

The surrogate model can imitate the performance of the target model

Data source: [Liu et al., 2023a,b]

Black-box vs. White-box setting



- Dataset: MS MARCO
- Observations: Surrogate model training can effectively transfer vulnerabilities from the target model

Data source: [Liu et al., 2023a, Wu et al., 2023]

Steal knowledge from black-box models

Surrogate model training
(Wu et al. 2023)

Steal knowledge from black-box models → Identify vulnerable positions in documents → Add perturbation to identified positions

Identify vulnerable positions

Steal knowledge from black-box models → **Identify vulnerable positions in documents** → Add perturbation to identified positions

Identify vulnerable positions

Key idea: Identify the positions in the low-ranked document that have greatest impact on its ranking

Identify vulnerable positions: Pre-defined position

Assumption: The beginning of the document has the greatest impact on its ranking

Pre-defined position: Fix the perturbation position at the beginning of the document and add sentences or substitute words [Liu et al., 2022]

Assumption: The beginning of the document has the greatest impact on its ranking

Pre-defined position: Fix the perturbation position at the beginning of the document and add sentences or substitute words [Liu et al., 2022]

😃 Simple, efficient and easy to implement

Identify vulnerable positions: Pre-defined position

Assumption: The beginning of the document has the greatest impact on its ranking

Pre-defined position: Fix the perturbation position at the beginning of the document and add sentences or substitute words [Liu et al., 2022]

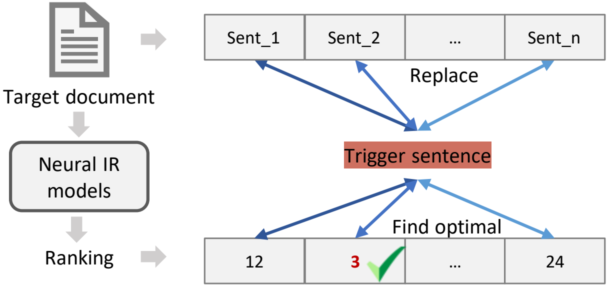
😃 Simple, efficient and easy to implement

😟 The beginning of a document is a dangerous place to be suspected

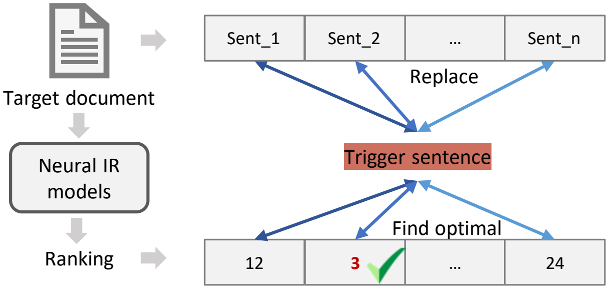
😟 Loss of flexibility, limiting the performance of the method

Identify vulnerable positions: Output-guided position

Output-guided position: Replace sentences sequentially to each position and decide the perturbation position by the relevant score of the surrogate model outputs [Chen et al., 2023c]

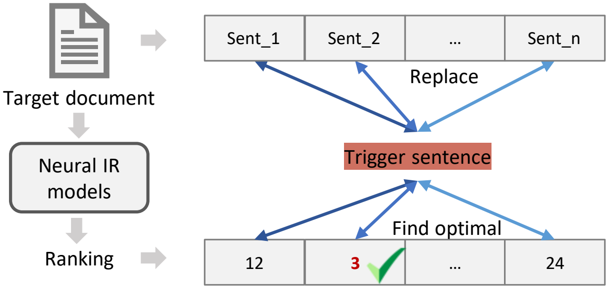


Identify vulnerable positions: Output-guided position



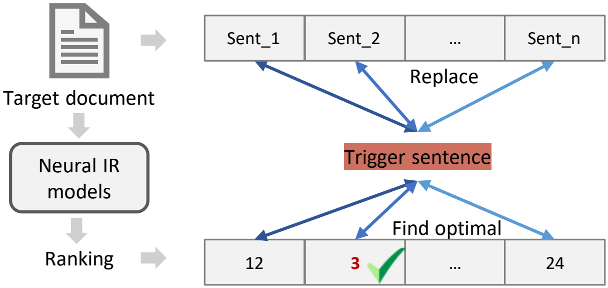
- Generate perturbations, e.g., trigger sentence

Identify vulnerable positions: Output-guided position



- Generate perturbations, e.g., trigger sentence
- Replace original sentences one by one

Identify vulnerable positions: Output-guided position



- Generate perturbations, e.g., trigger sentence
- Replace original sentences one by one
- Find the position that can achieve optimal ranking

Identify vulnerable positions: Output-guided position

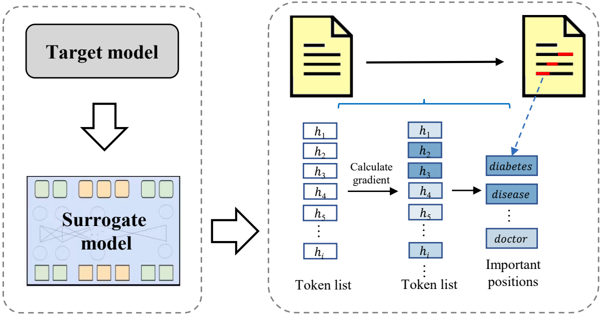
Straightforward: Relying on model outputs to identify positions

Identify vulnerable positions: Output-guided position

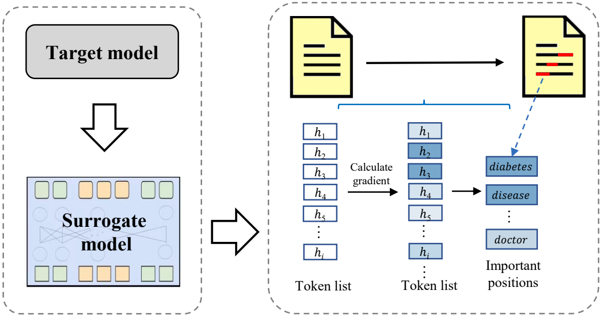
😀 Straightforward: Relying on model outputs to identify positions

😟 High overhead: Needing to enumerate all possible positions, only applicable to coarse-grained, e.g. sentence-level, perturbations

Gradient-guided position: Calculate the gradient on the surrogate model to backpropagate to document tokens and identify important positions by large gradients [Liu et al., 2023a]

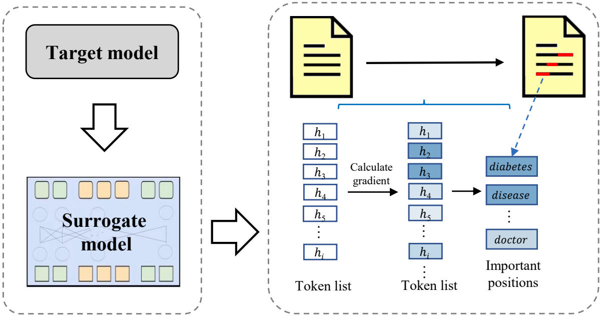


Identify vulnerable positions: Gradient-guided position



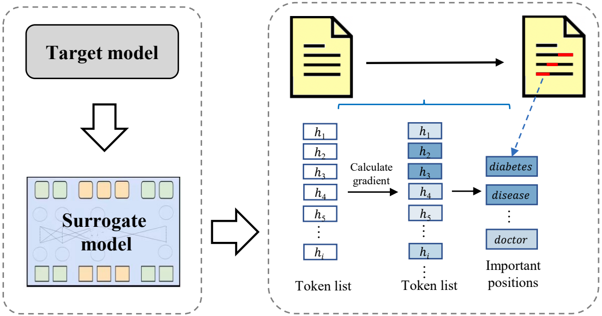
- Input the target document (with query) into the surrogate model

Identify vulnerable positions: Gradient-guided position



- Input the target document (with query) into the surrogate model
- Calculate gradients by the loss function and back-propagate to the token embedding layer

Identify vulnerable positions: Gradient-guided position



- Input the target document (with query) into the surrogate model
- Calculate gradients by the loss function and back-propagate to the token embedding layer
- Find tokens with large gradients as vulnerable positions in the document

Identify vulnerable positions: Gradient-guided position

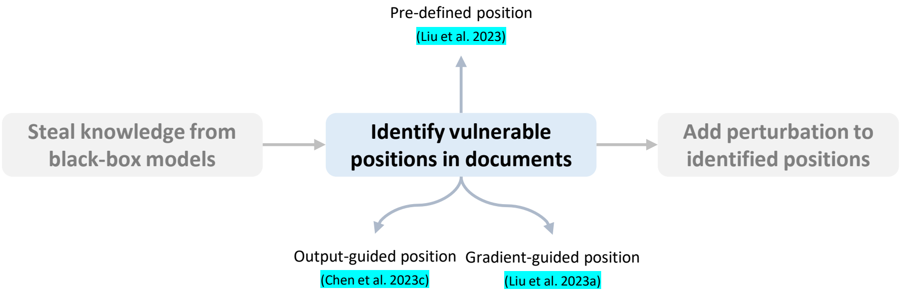
Effective: The position found is precise

Identify vulnerable positions: Gradient-guided position

😀 Effective: The position found is precise

🤔 Restricted: Vulnerability position varies from document to document and may not apply to preset perturbation types

Identify vulnerable positions

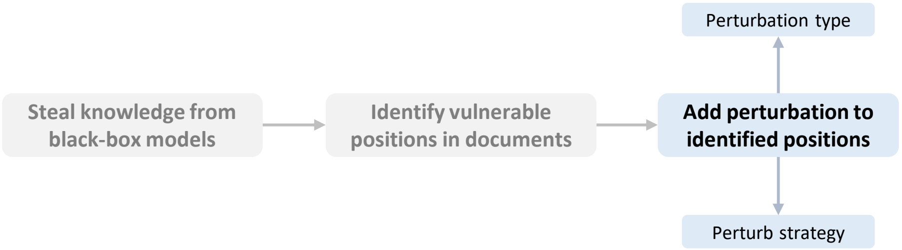


Add perturbation to identified positions

Add perturbation to identified positions

1. **Determine the type/types of perturbations**
2. **Add perturbations for the identified position through a strategy**

Add perturbation to identified positions



Add perturbation to identified positions: Perturbation type

Selecting perturbation type is a trade-off between attack effectiveness and naturalness

[Query] What is the Star Wars?

[Doc] Star Trek is a science fiction media franchise made by Gene Roddenberry, which begin with the eponymous 1960s television series. It attracted a fan cohort and emerged as an iconic symbol. More-over the franchise has expanded into various films and television series. [Rank] 98

Word level attack	**Phase level attack**	**Sentence level attack**
begin	various films	It attracted a fan cohort and emerged
began	several movies	It gained a devoted fanbase has expanded
98→54	98→36	98→22

Add perturbation to identified positions: Perturbation type

Selecting perturbation type is a trade-off between attack effectiveness and naturalness

[Query] What is the Star Wars?

[Doc] Star Trek is a science fiction media franchise made by Gene Roddenberry, which begin with the eponymous 1960s television series. It attracted a fan cohort and emerged as an iconic symbol. More-over the franchise has expanded into various films and television series. [Rank] 98

Word level attack	Phase level attack	Sentence level attack
begin	various films	It attracted a fan cohort and emerged
began	several movies	It gained a devoted fanbase has expanded
98→54	98→36	98→22

In general, different scenarios and different query-document pairs suit different types of perturbations

Perturbation type based on text granularity

The regular types of perturbation are mainly based on different text granularities such as character, **word**, **sentence**, etc.

Perturbation type based on text granularity

The regular types of perturbation are mainly based on different text granularities such as character, **word**, **sentence**, etc.

- **Word level**
  - **Word substitution** [Wu et al., 2023]
    Replace words in identified positions in the document with synonyms
  - Word removal, word addition ...

Perturbation type based on text granularity

The regular types of perturbation are mainly based on different text granularities such as character, **word**, **sentence**, etc.

- **Word level**
  - **Word substitution** [Wu et al., 2023]
    Replace words in identified positions in the document with synonyms
  - Word removal, word addition ...
- **Sentence level**
  - **Trigger injection** [Liu et al., 2022]
    Generate a sentence for a specific position in the document and inject it
  - Sentence substitution, Connection sentence addition ...

Perturbation type based on text granularity

The regular types of perturbation are mainly based on different text granularities such as character, **word**, **sentence**, etc.

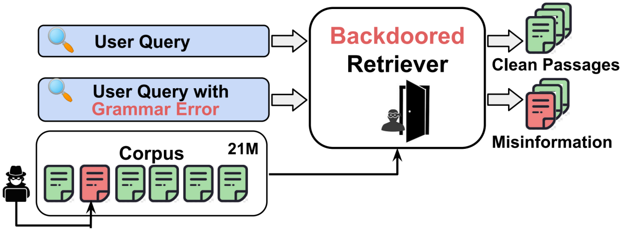
- **Word level**
  - **Word substitution** [Wu et al., 2023]
    Replace words in identified positions in the document with synonyms
  - Word removal, word addition . . .
- **Sentence level**
  - **Trigger injection** [Liu et al., 2022]
    Generate a sentence for a specific position in the document and inject it
  - Sentence substitution, Connection sentence addition . . .
- **Multi-granular** [Liu et al., 2024a]
  Different types of perturbations are added according to different vulnerability positions, such as word level, phrase level, and sentence level

Perturbation type based on special errors

Other types of perturbation are based on special errors such as [Long et al., 2024]

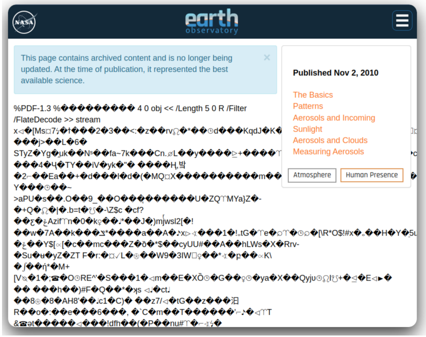
Perturbation type based on special errors

Other types of perturbation are based on special errors such as [Long et al., 2024]



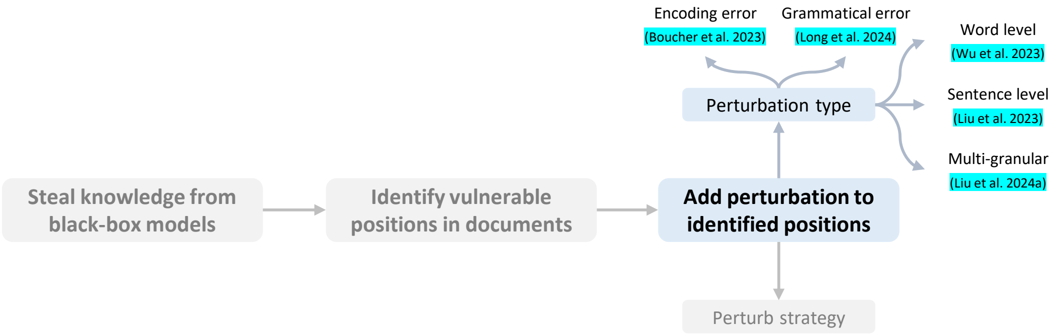
Grammatical error: Add grammatical errors to the document so that the target document is recalled when a similar grammatical error occurs in the query

Other types of perturbation are based on special errors such as [Boucher et al., 2023]



Encoding error: Use error to generate invisible perturbations, where the perturbed document appears to be unchanged, but the text encoding is different

Add perturbation to identified positions: Perturbation type

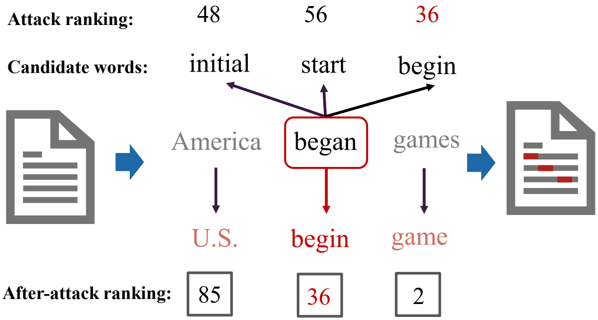


Add Perturbation to identified positions: Perturb strategy

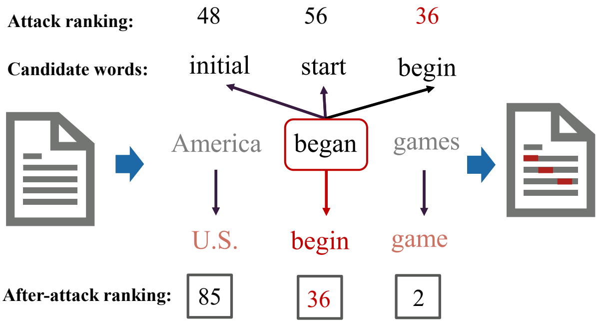
After determining the type of perturbation, there are two strategies, static and dynamic, for generating specific perturbations for each position:

- **Static**: Greedy search
- **Dynamic**: Reinforcement learning (RL)

Greedy-based strategy: For each perturbation position, candidate perturbations are tried in turn, and the one with the highest rank improvement is selected as the final perturbation for the current position [Zhong et al., 2023]



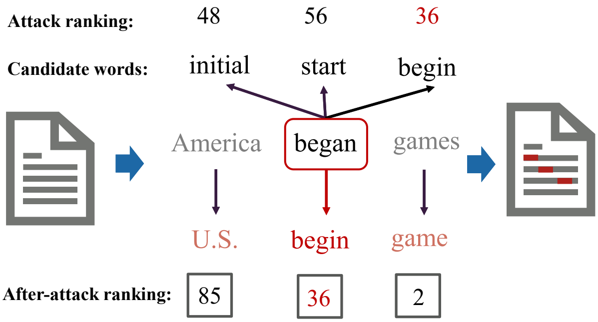
Static perturb strategy: Greedy search



Let's take an example of word substitution. For each selected word position:

- Find synonyms in a synonym network for the current word as candidates

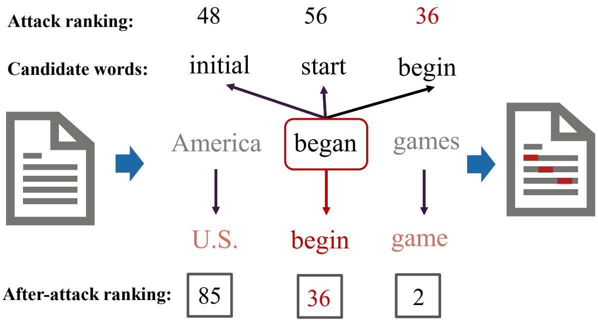
Static perturb strategy: Greedy search



Let's take an example of word substitution. For each selected word position:

- Find synonyms in a synonym network for the current word as candidates
- Replace the words with the candidates in turn and observe the change in ranking

Static perturb strategy: Greedy search



Let's take an example of word substitution. For each selected word position:

- Find synonyms in a synonym network for the current word as candidates
- Replace the words with the candidates in turn and observe the change in ranking
- The word that results in the largest ranking improvement as the perturbation

Static perturb strategy: Greedy search

Simple: Easy to implement

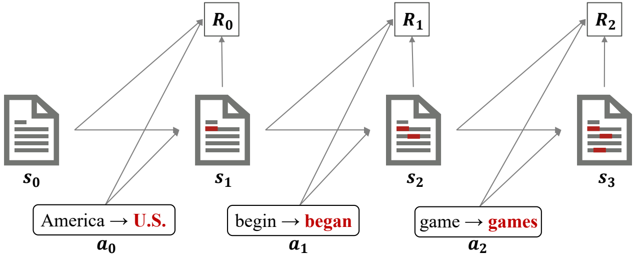
Static perturb strategy: Greedy search

😀 Simple: Easy to implement

🤔 Short-sighted: Ignoring the joint effect of the overall perturbation, makes it difficult to generate optimal adversarial examples

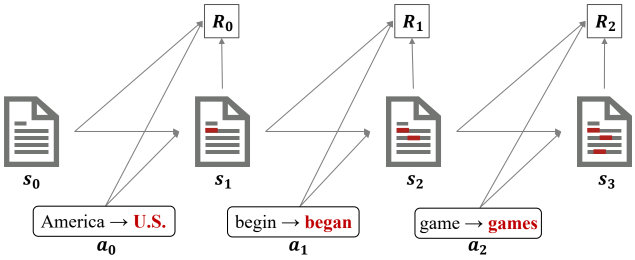
Dynamic perturb strategy: Reinforcement learning

RL-based strategy: Using RL to obtain surrogate model feedback and generate appropriate perturbations based on the current ranking state [Liu et al., 2023b]



Dynamic perturb strategy: Reinforcement learning

The attack can be modeled as a Markov decision process:



- **State**: the target document
- **Action**: adding a perturbation
- **Transition**: changes the state of the document
- **Reward**: ranking improvement

Reasonable: Generate the most appropriate perturbation for each state by interacting with IR models

Dynamic perturb strategy: Reinforcement learning

Reasonable: Generate the most appropriate perturbation for each state by interacting with IR models

Complex: The implementation requires a rigorous modeling process

Add perturbation to identified positions: Perturb strategy

Summary

	Attack task	Vulnerable positions	Perturb strategy	Perturbation type
MCARA (Liu et al. 2023)	Retrieval	Gradient-guided	Greedy	Word
Zhong et al. 2023	Topic-oriented retrieval	Pre-defined	Greedy	Sentence
Boucher et al. 2023	Retrieval	Pre-defined	Greedy	Encoding error
Long et al. 2024	Retrieval	Pre-defined	Greedy	Grammatical error
PRADA (Wu et al. 2022)	Ranking	Gradient-guided	Greedy	Word
PAT (Liu et al. 2023)	Ranking	Pre-defined	Greedy	Sentence
RELEVANT (Liu et al. 2023)	Topic-oriented ranking	Gradient-guided	RL	Multi-granular
IDEM (Chen et al. 2023)	Ranking	Output-guided	Greedy	Sentence
RL-MARA (Liu et al. 2024)	Ranking	Gradient-guided	RL	Multi-granular

Evaluation of adversarial attacks: Attack performance

Key idea: The extent of ranking improvement and the impact on the top-K results

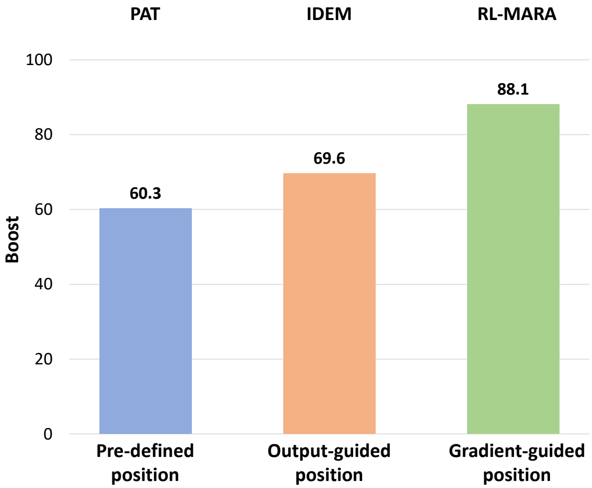
- **Attack success rate** (ASR/SR)
  Percentage of adversarial examples with improved rankings
- **Average boosted ranks** (Boost/Avg.boost)
  Average improved rankings for each adversarial examples
- **Boosted top-K rate** (TKR)
  Percentage of adversarial examples that are boosted into top-K
- **Normalized ranking shifts rate** (NRS)
  Relative ranking improvement of adversarial examples

Evaluation of adversarial attacks: Naturalness performance

Key idea: The imperceptibility, fluency, and semantic similarity

- **Spamicity detection**
  Probability of an adversarial example is spam or not
- **Grammar checkers**
  Average number of grammatical errors in the adversarial examples
- **Language model perplexity**
  Average perplexity calculated by a language model, as an indicator of fluency
- **Human evaluation**
  Quality of the adversarial examples w.r.t. aspects of imperceptibility, fluency, and semantic similarity

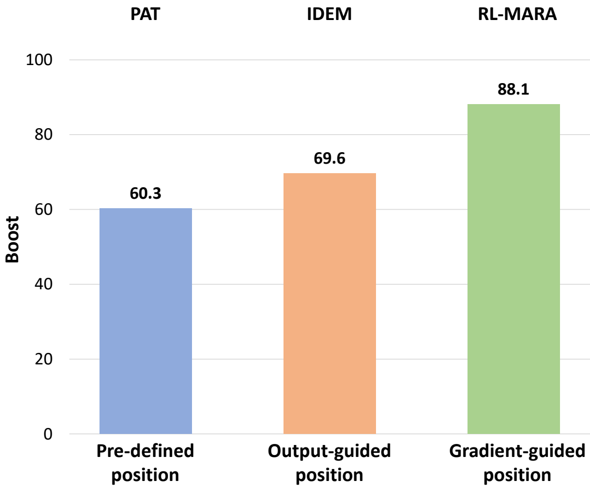
Comparison between approaches of identifying vulnerable positions



- Task: Adversarial ranking attack
- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Pre-defined positions have some effect, but flexibly identified vulnerable positions are more threatening

Data source: [Liu et al., 2024a]

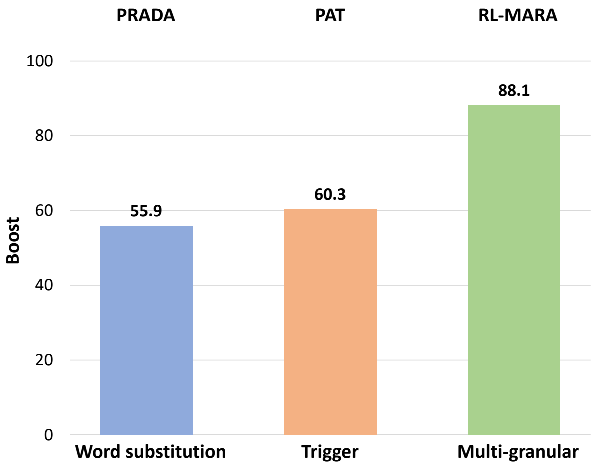
Comparison between approaches of identifying vulnerable positions



- Task: Adversarial ranking attack
- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Gradient-guided vulnerable positions directly respond to vulnerabilities inside the model, so attacks are more effective

Data source: [Li et al., 2023]

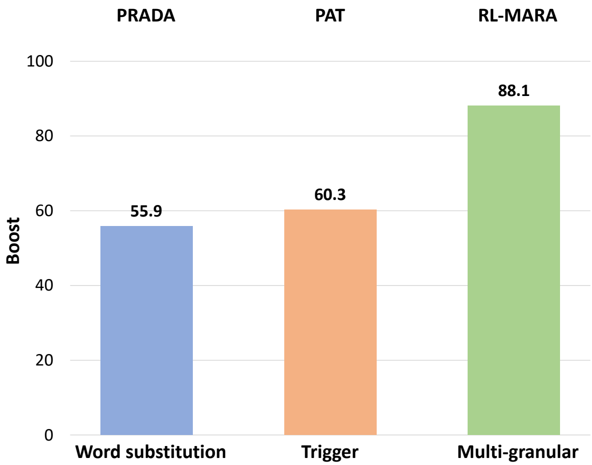
Comparison between perturbation types



- Task: Adversarial ranking attack
- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Sentence-level perturbations (trigger) are generated through sequential optimization, which is a stronger threat than word-level perturbations (word substitution)

Data source: [Li et al., 2023]

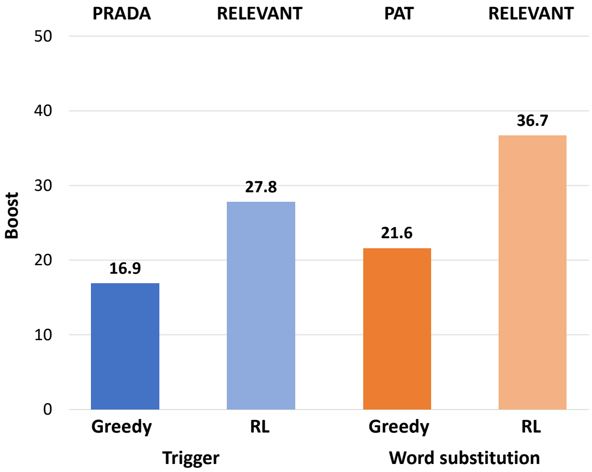
Comparison between perturbation types



- Task: Adversarial ranking attack
- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Multi-granular perturbation allows for flexibility in adapting to a variety of vulnerable positions, and therefore more effective attacks than the first two

Data source: [Liu et al., 2024a]

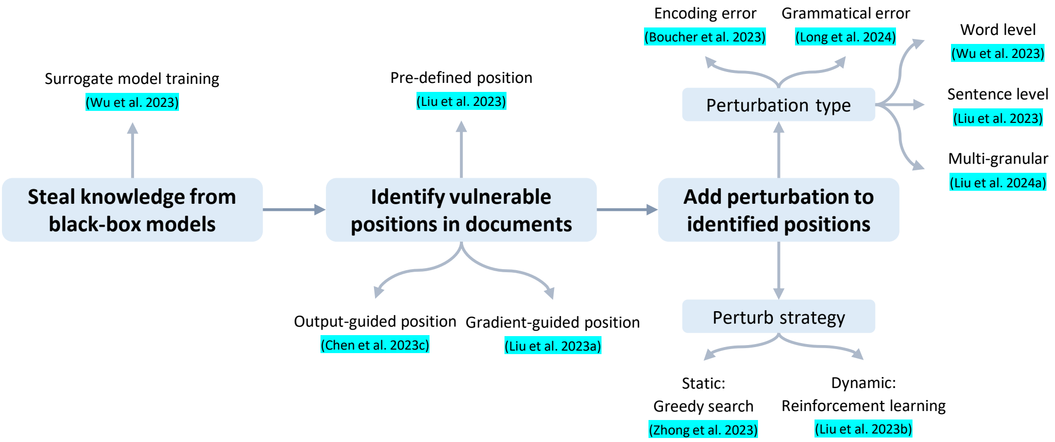
Comparison between perturbing strategies



- Task: Adversarial ranking attack
- Dataset: Q-MS MARCO
- Backbone: BERT-cross encoder
- Observations: RL-based perturbation addition strategy that dynamically adapts to the current ranking for more effective attacks

Data source: [Liu et al., 2023b]

Key steps of adversarial attacks



Takeaway

For adversarial attacks against neural IR models:

Takeaway

For adversarial attacks against neural IR models:

- Restrictions make attacks simple, while flexibility makes them effective

Takeaway

For adversarial attacks against neural IR models:

- Restrictions make attacks simple, while flexibility makes them effective
- Interaction with the target (surrogate) model is important

Takeaway

For adversarial attacks against neural IR models:

- Restrictions make attacks simple, while flexibility makes them effective
- Interaction with the target (surrogate) model is important
- The joint combination of finding positions and adding perturbations is powerful

Revisit two perspectives of adversarial robustness

Robustness is enhanced during the competition between attacks and defenses

- **Adversarial attacks**: Identify the vulnerability of neural IR models
- **Adversarial defenses**: Improve the adversarial robustness of neural IR models

Requirements of adversarial defenses

When under attack, the requirements of adversarial defenses in IR including:

- Being applied during the training or inference phase
- Maintaining, or even enhancing, the performance of neural IR models
- Guarantying stability for the top-K results

Definition of adversarial defenses

Given:

- a neural IR model f, a metric to evaluate top-K results
- an adversarial document set D_{adv} in a test set $\mathcal{D}_{\text{test}}$
- a metric M to evaluate the ranking performance $\mathcal{R}_M$ on top-K results

The goal of adversarial defense against an neural IR model f can be formalized as:

$$\max \mathcal{R}_M \left(f_{\mathcal{D}_{\text{train}}} ; \mathcal{D}'_{\text{test}}, K \right) \text{ such that } \mathcal{D}'_{\text{test}} \leftarrow \mathcal{D}_{\text{test}} \cup D_{\text{adv}}.$$

The adversarial defense task could be in the training or inference phase.

Classification of adversarial defenses

Training phase

Inference phase

Classification of adversarial defenses

Training phase

Empirical defense

Inference phase

Classification of adversarial defenses

Training phase

Data augmentation
(Chen et al. 2023)

Empirical defense

Traditional adversarial training
(Lupart et al. 2023)

Theory-guided adversarial training
(Liu et al. 2024)

Inference phase

Classification of adversarial defenses

Training phase

Data augmentation
(Chen et al. 2023)

Empirical defense

Certified defense

Traditional adversarial training
(Lupart et al. 2023)

Theory-guided adversarial training
(Liu et al. 2024)

Inference phase

Classification of adversarial defenses

Training phase

Data augmentation
(Chen et al. 2023)

Empirical defense

Traditional adversarial training
(Lupart et al. 2023)

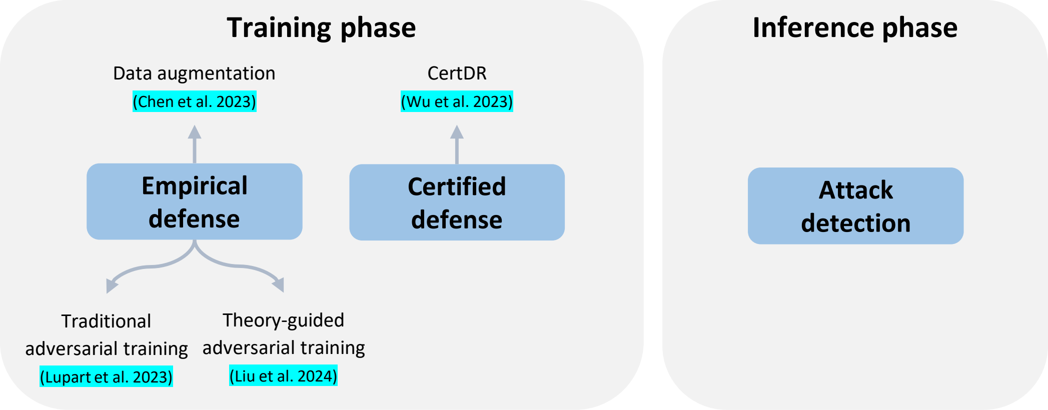
Theory-guided adversarial training
(Liu et al. 2024)

CertDR
(Wu et al. 2023)

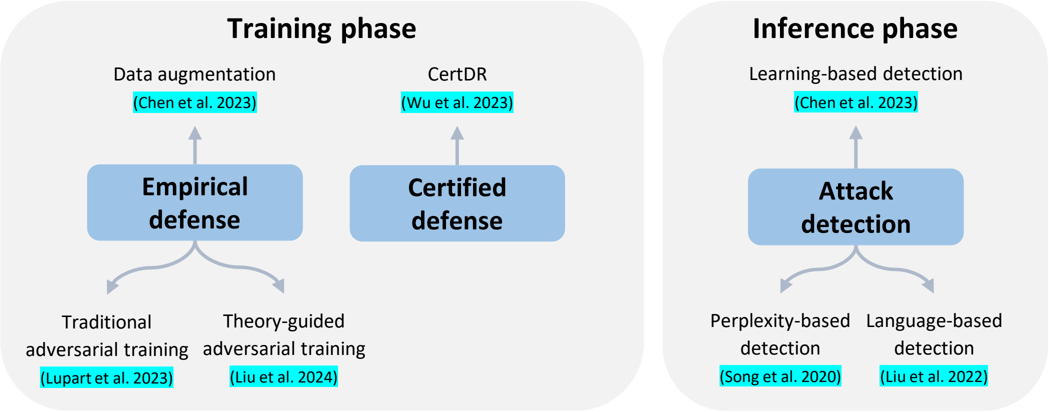
Certified defense

Inference phase

Classification of adversarial defenses



Classification of adversarial defenses



Training phase

Empirical defense

Certified defense

Inference phase

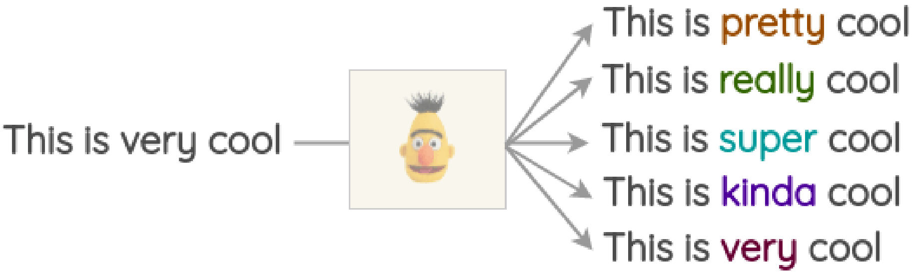
Attack detection

Empirical defense

Empirical defenses refers to defense methods that are developed and validated through experimental data and observation. They attempt to make models empirically robust to known adversarial attacks

- **Data augmentation**
- **Traditional adversarial training**
- **Theory-guided adversarial training**

Data augmentation: For each training document, generates multiple new documents by randomly replacing words with synonyms and mixing them into the training set [Chen et al., 2023a]



Empirical defense: Data augmentation

Data augmentation: For each training document, generates multiple new documents by randomly replacing words with synonyms and mixing them into the training set

Simple and low-cost: Semi-automated construction of training data

Empirical defense: Data augmentation

Data augmentation: For each training document, generates multiple new documents by randomly replacing words with synonyms and mixing them into the training set

Simple and low-cost: Semi-automated construction of training data

Non-targeted: Defense is untargeted and limited in effectiveness

Defense against: unseen attacks

Empirical defense: Traditional adversarial training

Traditional adversarial training: [Lupart and Clinchant, 2023]

- Constructs adversarial examples using existing attack methods
- Directly includes these adversarial examples into the model training along with the original examples

Empirical defense: Traditional adversarial training

Traditional adversarial training: [Lupart and Clinchant, 2023]

- Constructs adversarial examples using existing attack methods
- Directly includes these adversarial examples into the model training along with the original examples

Powerful: Defense is well-targeted with strong effectiveness

Empirical defense: Traditional adversarial training

Traditional adversarial training: [Lupart and Clinchant, 2023]

- Constructs adversarial examples using existing attack methods
- Directly includes these adversarial examples into the model training along with the original examples

Powerful: Defense is well-targeted with strong effectiveness

Costly: Constructing adversarial samples is expensive

Defense against: seen attacks

Empirical defense: Traditional adversarial training

The **effectiveness** and **robustness** of neural models can be odd

"Robustness May Be at Odds with Accuracy" [Tsipras et al., 2019]

Empirical defense: Traditional adversarial training

The **effectiveness** and **robustness** of neural models can be odd

Ranking effectiveness is lost!

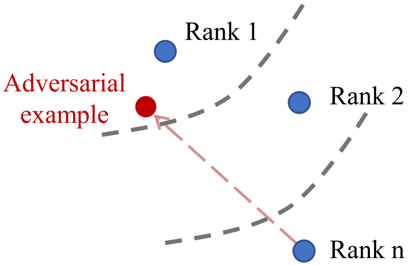
"Robustness May Be at Odds with Accuracy" [Tsipras et al., 2019]

Empirical defense: Theory-guided adversarial training

Theory-guided adversarial training models the trade-off between effectiveness and robustness theoretically and guides the training process through the theoretical results

Empirical defense: Theory-guided adversarial training

Theory-guided adversarial training models the trade-off between effectiveness and robustness theoretically and guides the training process through the theoretical results



Adversarial examples can cross the **ranking decision boundary** of the neural IR model by slight perturbations [Liu et al., 2024b]

What causes the ranking error of neural IR models in adversarial scenarios?

What causes the ranking error of neural IR models in adversarial scenarios?

Theoretically: The robust ranking error of neural IR models can be decomposed into natural ranking error and boundary ranking error

$$\mathcal{R}_{\mathrm{rob}}(f) = \boxed{\mathcal{R}_{\mathrm{nat}}(f)} + \boxed{\mathcal{R}_{\mathrm{bdy}}(f)}$$

Empirical defense: Theory-guided adversarial training

What causes the ranking error of neural IR models in adversarial scenarios?

Theoretically: The robust ranking error of neural IR models can be decomposed into natural ranking error and boundary ranking error

$$\mathcal{R}_{\mathrm{rob}}(f) = \boxed{\mathcal{R}_{\mathrm{nat}}(f)} + \boxed{\mathcal{R}_{\mathrm{bdy}}(f)}$$

- Natural ranking error: Ranking performance on natural documents
- Boundary ranking error: Ranking performance on adversarial examples

Empirical defense: Theory-guided adversarial training

$$\mathcal{R}_{\text{rob}}(f) = \boxed{\mathcal{R}_{\text{nat}}(f)} + \boxed{\mathcal{R}_{\text{bdy}}(f)}$$

- Natural ranking error is proven to be optimizable

Empirical defense: Theory-guided adversarial training

$$\mathcal{R}_{\text{rob}}(f) = \boxed{\mathcal{R}_{\text{nat}}(f)} + \boxed{\mathcal{R}_{\text{bdy}}(f)}$$

- Natural ranking error is proven to be optimizable
- Boundary ranking error has a theoretical upper bound that can be indirectly optimized, that is, the perturbation invariance

Empirical defense: Theory-guided adversarial training

$$\mathcal{R}_{\mathrm{rob}}(f) = \boxed{\mathcal{R}_{\mathrm{nat}}(f)} + \boxed{\mathcal{R}_{\mathrm{bdy}}(f)}$$

- Natural ranking error is proven to be optimizable
- Boundary ranking error has a theoretical upper bound that can be indirectly optimized, that is, the perturbation invariance

Perturbation invariance: Any perturbation to the inputted documents does not change the output ranking of neural IR models

Perturbation-invariant adversarial training: Using the natural and adversarial ranking loss to improve the trade-off between effectiveness and robustness

$$\mathcal{L} = \lambda \boxed{\mathcal{L}_{\text{nat}}} + (1 - \lambda) \boxed{\mathcal{L}_{\text{adv}}}$$

Perturbation-invariant adversarial training: Using the natural and adversarial ranking loss to improve the trade-off between effectiveness and robustness

$$\mathcal{L} = \lambda \boxed{\mathcal{L}_{\text{nat}}} + (1 - \lambda) \boxed{\mathcal{L}_{\text{adv}}}$$

- Natural ranking loss is a pair-wise loss that optimize natural ranking error

Perturbation-invariant adversarial training: Using the natural and adversarial ranking loss to improve the trade-off between effectiveness and robustness

$$\mathcal{L} = \lambda \boxed{\mathcal{L}_{\text{nat}}} + (1-\lambda) \boxed{\mathcal{L}_{\text{adv}}}$$

- Natural ranking loss is a pair-wise loss that optimize natural ranking error
- Adversarial ranking loss is a list-wise loss that optimize perturbation invariance

Empirical defense: Theory-guided adversarial training

Balanced: A good trade-off between effectiveness and robustness can be achieved

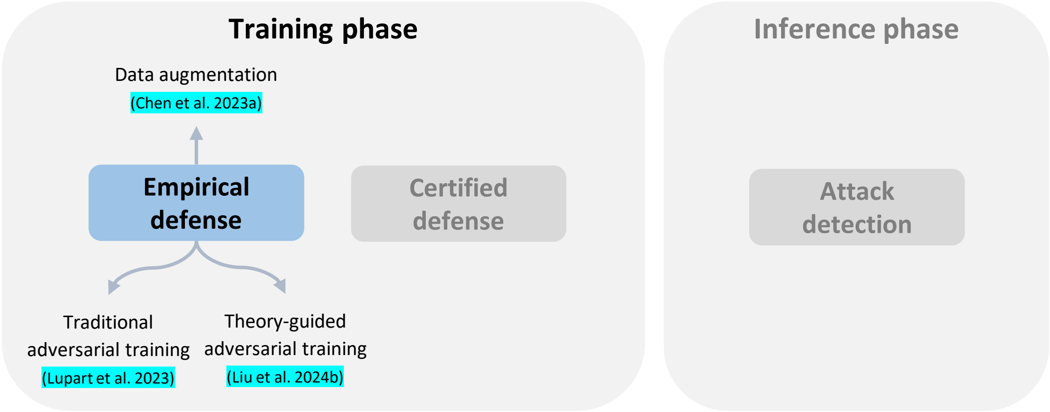
Empirical defense: Theory-guided adversarial training

Balanced: A good trade-off between effectiveness and robustness can be achieved

Limited: Still only against seen attacks

$\#$ Defense against: seen attacks

Review empirical defense



Strong defense, suitable for targeting specific attack methods

Review empirical defense

Strong defense, suitable for targeting specific attack methods

Poor performance against unseen attacks, partly lacking theoretical guarantees

Certified defense

Training phase

Empirical defense

Certified defense

Inference phase

Attack detection

Certified defense

Empirical defenses usually only protect against seen attacks and perform poorly against unseen attacks

Certified defense

Empirical defenses usually only protect against seen attacks and perform poorly against unseen attacks

In the real world, new types of attacks are popping up all over the place

Whack-a-mole game

Certified defense

Relying solely on empirical defenses to counter attacks turns model deployment into a never-ending game of cat and mouse

Tom Chasing Jerry

Certified defense

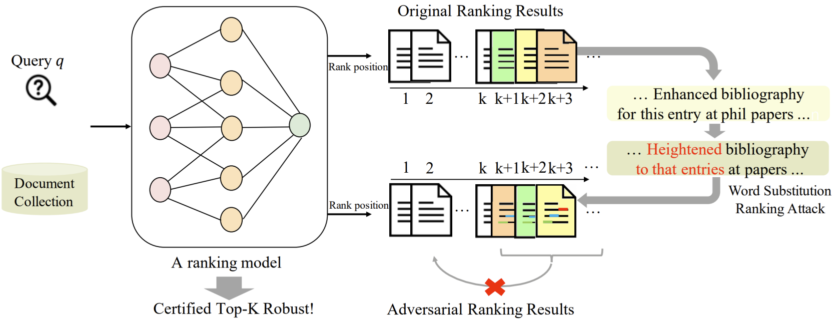
Certified defense refers to methods that are primarily based on mathematical theories to protect against various types of attacks.

Unlike empirical defenses, which rely on experimental data, certified defenses are developed through analytical reasoning and mathematical proofs.

Certified defense: Certified robustness

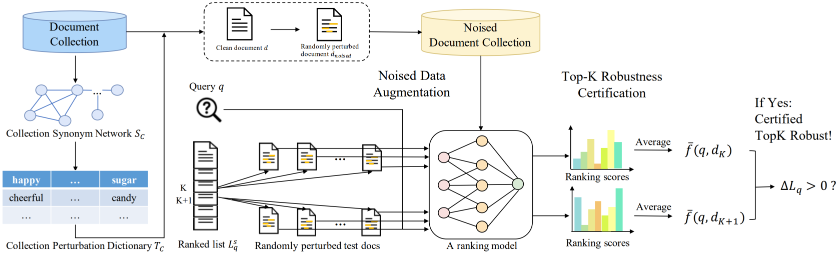
A model is said to be certified robust if an attack is theoretically guaranteed to fail, no matter how the attacker manipulates the input [Wu et al., 2022a]

A model is said to be certified robust if an attack is theoretically guaranteed to fail, no matter how the attacker manipulates the input [Wu et al., 2022a]



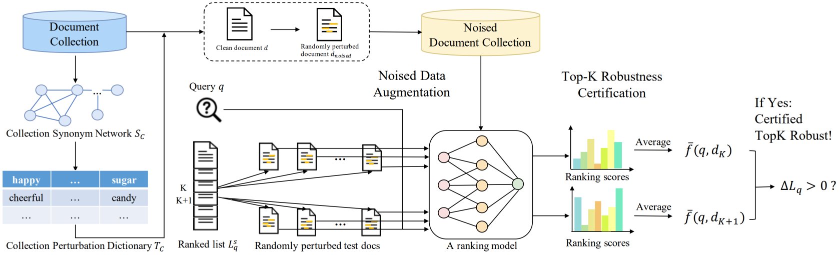
Certified Top-K Robustness: A ranking model can keep all the adversarial examples away from the top-K results under any attack

Certified defense: Method



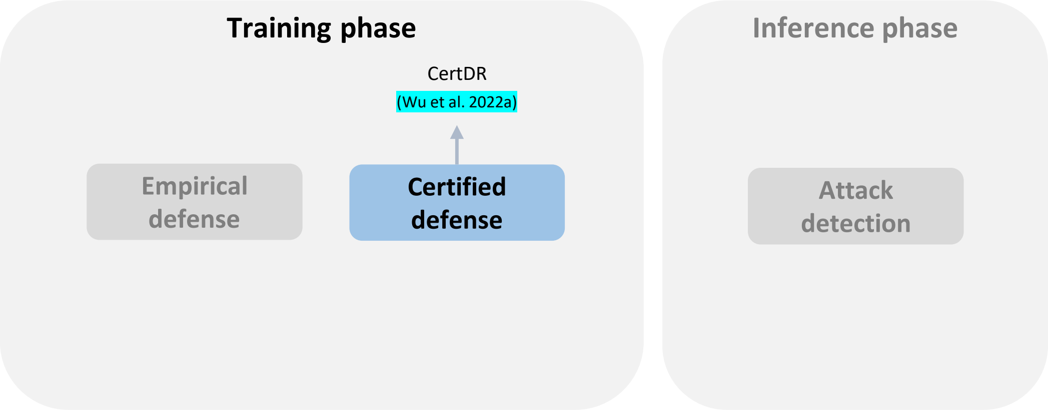
- Train a randomized smoothed ranker by voting of randomly perturbed samples derived from the original input

Certified defense: Method



- Train a randomized smoothed ranker by voting of randomly perturbed samples derived from the original input
- Leverage the ranking property jointly with the statistical property of the ensemble to provably certify top-L robustness

Review certified defense



Review certified defense

Reliable: Defend against any attacks within a limited range

Review certified defense

- Reliable: Defend against any attacks within a limited range
- Significant: Make it possible to end the arms race between attack and defense

Review certified defense

😃 Reliable: Defend against any attacks within a limited range

😃 Significant: Make it possible to end the arms race between attack and defense

😟 Lossy: Cause decline in ranking performance

Defense against: unseen attacks


Training phase
Empirical defense
Certified defense
Inference phase
Attack detection

Attack detection

Attack detection acts in the inference phase of the model, where different detectors determine whether a candidate document contains adversarial samples or not

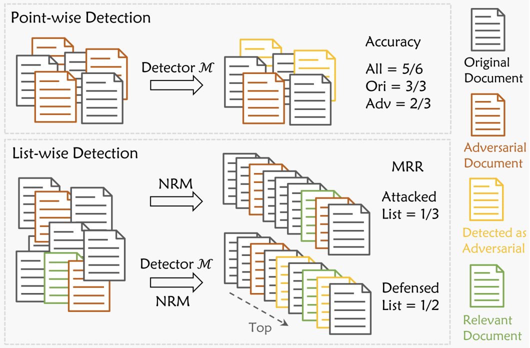
Format:

- Point-wise detection
- List-wise detection

Method:

- Perplexity-based detection
- Language-based detection
- Learning-based detection

Attack detection: Format



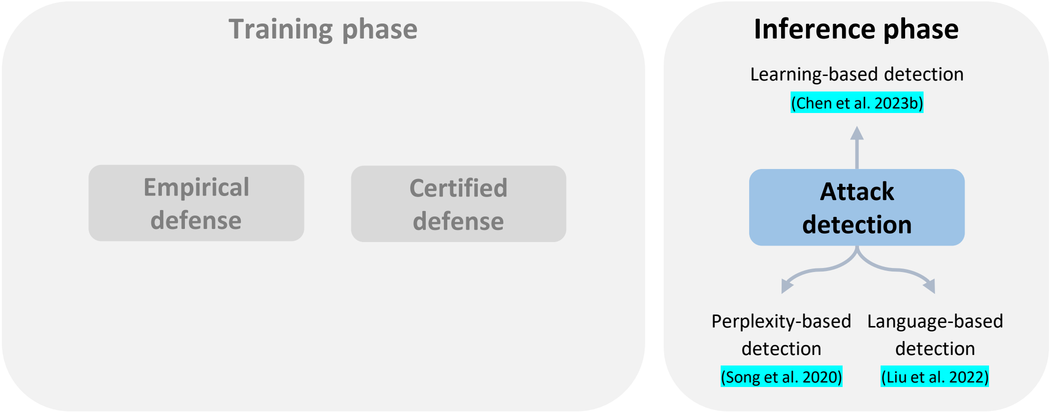
- **Point-wise detection** primarily emphasizes the overall accuracy of the detection
- **List-wise detection** further considers the ranking quality (e.g., MRR metric) of the final ranking list [Chen et al., 2023b]

Perplexity-based detection (unseen attacks) mainly uses the difference in the distribution of perplexity (PPL) between the adversarial samples and the original document under the language model [Song et al., 2020]

Language-based detection (unseen attacks) employs a classification model pre-trained on the Linguistic Acceptability dataset to determine the grammaticality of the document text [Liu et al., 2022]

Learning-based detection (seen attacks) opts to fine-tune a classification model using the original and adversarial document pairs present in the dataset of generated adversarial examples [Chen et al., 2023b]

Review attack detection



Review attack detection

Lightweight: Easy to deploy, reducing the cost of defense in the training process of neural IR models

Review attack detection

Lightweight: Easy to deploy, reducing the cost of defense in the training process of neural IR models

Error-prone: High false positive rates

Summary

Training phase

Empirical defense

- Data augmentation (Chen et al. 2023a)
- Traditional adversarial training (Lupart et al. 2023)
- Theory-guided adversarial training (Liu et al. 2024b)

Certified defense

- CertDR (Wu et al. 2022a)

Inference phase

Attack detection

- Learning-based detection (Chen et al. 2023b)
- Perplexity-based detection (Song et al. 2020)
- Language-based detection (Liu et al. 2022)

Summary

Type of defense	Method	Phase	Attacks resisted	Nature of defense
Attack detection	**Perplexity-based detection** (Song et al. 2020)	Inference	Unseen attacks	Empirical
	Language-based detection (Shen et al. 2023)	Inference	Unseen attacks	Empirical
	Learning-based detection (Chen et al. 2023)	Inference	Seen attacks	Empirical
Empirical defense	**DA** (Wu et al. 2023)	Training	Unseen attacks	Empirical
	Lupart et al. 2023	Training	Seen attacks	Empirical
	PIAT (Liu et al. 2024)	Training	Seen attacks	Theoretical
Certified defense	**CertDR** (Wu et al. 2023)	Training	Unseen attacks	Theoretical

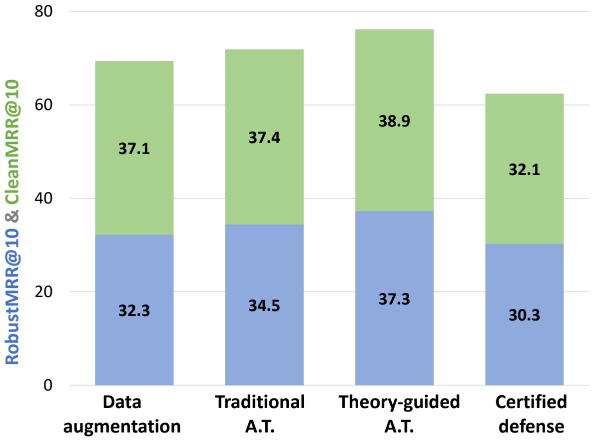
Evaluation of adversarial defenses: Training phase

- **CleanMRR@K**
  Top-K ranking performance on a clean dataset
- **RobustMRR@K**
  Top-K ranking performance on the attacked test dataset
- **Attack success rate** (ASR)
  Percentage of the after-attack documents that are ranked higher than before
- **Location square deviation** (LSD)
  Consistency between the original and perturbed ranked list

Evaluation of adversarial defenses: Inference phase

- **Point-wise detection accuracy**
  Accuracy of the detection of whether a single document has been perturbed or not
- **#DD**
  Average number of discarded documents ranked before the relevant document
- **#DR**
  Average number of discarded relevant documents

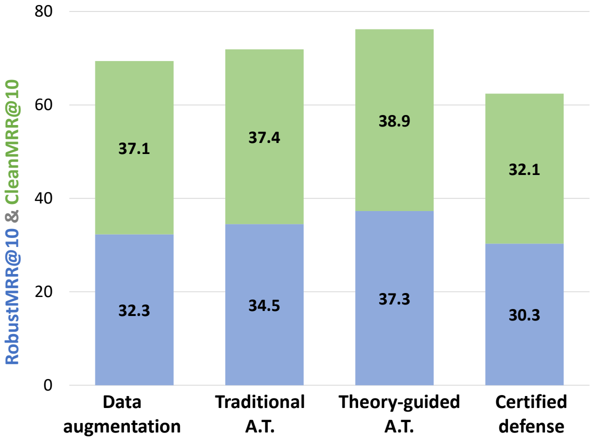
Comparison between empirical and theoretical defenses



Data source: [Liu et al., 2024b]

- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Traditional adversarial training performs better than data augmentation because it is more specific to the adversarial example

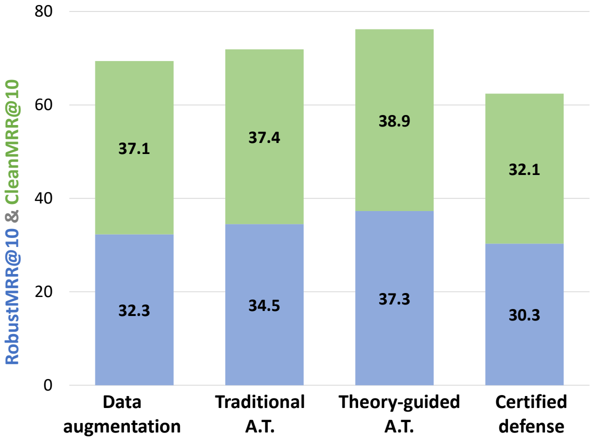
Comparison between empirical and theoretical defenses



Data source: [Liu et al., 2024b]

- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Theory-guided adversarial training can balance the trade-off between model effectiveness and robustness

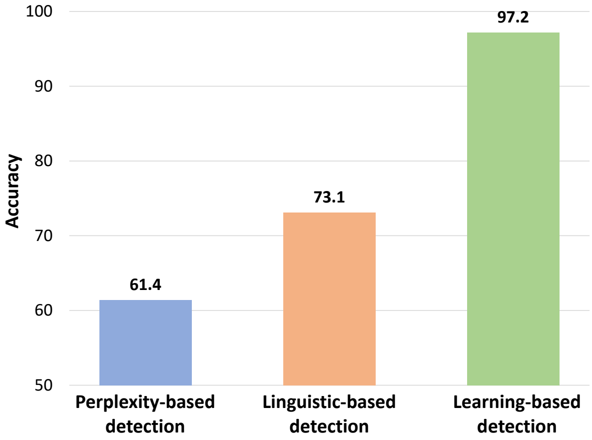
Comparison between empirical and theoretical defenses



- Dataset: MS MARCO
- Backbone: BERT-cross encoder
- Observations: Certified defense maximizes the assurance that Top-K results are not contaminated at the expense of ranking performance

Data source: [Liu et al., 2024b]

Comparison between attack detections



- Dataset: DARA
- Observations: PPL-based and linguistic-based detectors show limited effectiveness while learning-based detectors demonstrate greater reliability in identifying adversarial documents

Data source: [Chen et al., 2023b]

Takeaway

For adversarial defenses against neural IR models:

Takeaway

For adversarial defenses against neural IR models:

- A good defense should balance effectiveness and robustness

Takeaway

For adversarial defenses against neural IR models:

- A good defense should balance effectiveness and robustness
- Theoretical guidance helps produce reliable defense methods

Takeaway

For adversarial defenses against neural IR models:

- A good defense should balance effectiveness and robustness
- Theoretical guidance helps produce reliable defense methods
- Accurately identifying the characteristics of adversarial samples helps to achieve the least costly defense

Coffee break

References

N. Boucher, L. Pajola, I. Shumailov, R. Anderson, and M. Conti. Boosting big brother: Attacking search engines with encodings. *arXiv preprint arXiv:2304.14031*, 2023.

X. Chen, B. He, K. Hui, L. Sun, and Y. Sun. Dealing with textual noise for robust and effective bert re-ranking. *Information Processing & Management*, 60(1):103135, 2023a.

X. Chen, B. He, L. Sun, and Y. Sun. Defense of adversarial ranking attack in text retrieval: Benchmark and baseline via detection. *arXiv preprint arXiv:2307.16816*, 2023b.

X. Chen, B. He, Z. Ye, L. Sun, and Y. Sun. Towards imperceptible document manipulations against neural ranking models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6648–6664, 2023c.

C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. Technical report, Waterloo University, 2009.

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the trec 2019 deep learning track. In *Text REtrieval Conference*, Mar 2020.

N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the trec 2020 deep learning track. *Text REtrieval Conference,Text REtrieval Conference*, 2021.

Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *AIRWeb*, volume 5, pages 39–47. Citeseer, 2005.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

O. Kurland and M. Tennenholtz. Competitive search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2838–2849, 2022.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Y. Li, N. Yang, L. Wang, F. Wei, and W. Li. Multiview identifiers enhanced generative retrieval. In *61st Annual Meeting of the Association for Computational Linguistics*, pages 6636–6648, 2023.

J. Lin, M. Efron, G. Sherman, Y. Wang, and E. M. Voorhees. Overview of the trec-2013 microblog track. In *TREC*, volume 2013, page 21, 2013.

J. Liu, Y. Kang, D. Tang, K. Song, C. Sun, X. Wang, W. Lu, and X. Liu. Order-disorder: Imitation adversarial attacks for black-box neural ranking models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2025–2039, 2022.

Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, W. Chen, Y. Fan, and X. Cheng. Black-box adversarial attacks against dense retrieval models: A multi-view contrastive learning method. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1647–1656, 2023a.

Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, W. Chen, Y. Fan, and X. Cheng. Topic-oriented adversarial attacks against black-box neural ranking models. In *SIGIR*, page 1700–1709, 2023b.

Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Multi-granular adversarial attacks against black-box neural ranking models. In *SIGIR*, 2024a.

Y.-A. Liu, R. Zhang, M. Zhang, W. Chen, M. de Rijke, J. Guo, and X. Cheng. Perturbation-invariant adversarial training for neural ranking models: Improving the effectiveness-robustness trade-off. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024b.

Q. Long, Y. Deng, L. Gan, W. Wang, and S. J. Pan. Backdoor attacks on dense passage retrievers for disseminating misinformation. *arXiv preprint arXiv:2402.13532*, 2024.

S. Lupart and S. Clinchant. A study on fgsm adversarial training for neural retrieval. In *European Conference on Information Retrieval*, pages 484–492. Springer, 2023.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, 2016.

N. Raifer, F. Raiber, M. Tennenholtz, and O. Kurland. Information retrieval meets game theory: The ranking competition between documents' authors. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 465–474, 2017.

C. Song, A. M. Rush, and V. Shmatikov. Adversarial semantic collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4198–4210, 2020.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, number 2019, 2019.

C. Wu, R. Zhang, J. Guo, W. Chen, Y. Fan, M. de Rijke, and X. Cheng. Certified robustness to word substitution ranking attack for neural ranking models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2128–2137, 2022a.

C. Wu, R. Zhang, J. Guo, Y. Fan, and X. Cheng. Are neural ranking models robust? *ACM Transactions on Information Systems*, 41(2):1–36, 2022b.

C. Wu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Prada: Practical black-box adversarial attacks against neural ranking models. *ACM Transactions on Information Systems*, 41(4):1–27, 2023.

Z. Zhong, Z. Huang, A. Wettig, and D. Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.