Rethinking Word-level Adversarial Attack: The Trade-off Between Efficiency, Effectiveness, and Imperceptibility

Pengwei Zhan[◇♣], Jing Yang[◇*], He Wang[◇], Chao Zheng[◇], Liming Wang[◇]
[◇]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[♣]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhanpengwei, yangjing, wanghe6029, zhengchao1135, wangliming}@iie.ac.cn

Abstract

Neural language models have demonstrated impressive performance in various tasks but remain vulnerable to word-level adversarial attacks. Word-level adversarial attacks can be formulated as a combinatorial optimization problem, and thus, an attack method can be decomposed into search space and search method. Despite the significance of these two components, previous works inadequately distinguish them, which may lead to unfair comparisons and insufficient evaluations. In this paper, to address the inappropriate practices in previous works, we perform thorough ablation studies on the search space, illustrating the substantial influence of search space on attack efficiency, effectiveness, and imperceptibility. Based on the ablation study, we propose two standardized search spaces: the **S**earch **S**pace for **I**m**P**erceptibility (SSIP) and **S**earch **S**pace for **E**ffec**T**iveness (SSET). The reevaluation of eight previous attack methods demonstrates the success of SSIP and SSET in achieving better trade-offs between efficiency, effectiveness, and imperceptibility in different scenarios, offering fair and comprehensive evaluations of previous attack methods and providing potential guidance for future works.

Keywords: Language Model, Adversarial Example, Robustness, Combinatorial Optimization

1. Introduction

Neural language models show remarkable performance across various tasks, but they remain vulnerable to adversarial attacks. Such attacks prompt models to generate incorrect outputs through subtle input modifications. Adversarial examples can be crafted at multiple granularities, including character-level (Ebrahimi et al., 2018; Chen et al., 2022), sentence-level (Jia and Liang, 2017; Liang et al., 2018), and word-level (Pruthi et al., 2019; Li et al., 2019; Zhan et al., 2022c; Jin et al., 2020; Zhan et al., 2023b; Li et al., 2020). Among these, word-level adversarial attacks have garnered increased attention due to their effectiveness and flexibility in producing high-quality examples. By perturbing a minimal number of words within the input text, word-level adversarial examples can substantially change the model's output while largely maintaining grammaticality and fluency.

Following Zang et al. (2020), Yoo et al. (2020), and Morris et al. (2020), word-level adversarial attacks can be formulated as a combinatorial optimization problem (Blair, 1990), consisting of two essential components: *Search Space* and *Search Method*. The search space imposed with various constraints, e.g., semantic similarity and part-of-speech constraints, defines the set of words that are qualified for crafting adversarial examples, while the search method determines the strategy for traversing the search space and identifying optimal perturbations. Both the search space and search method significantly influence the efficiency, effectiveness, and imperceptibility of attacks.

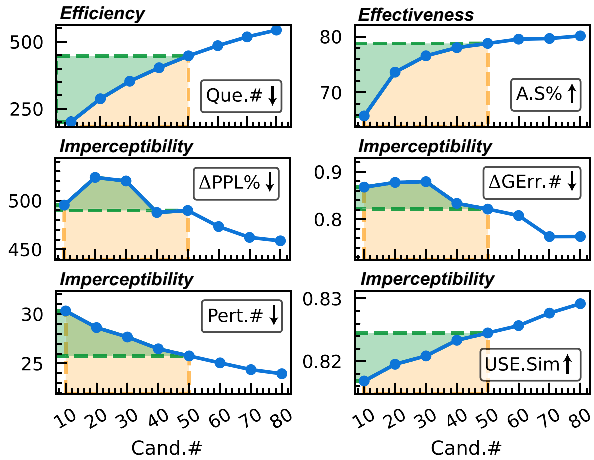

Figure 1: The impact of the number of candidate words (Cand.#) picked from search space in each step on the attack efficiency, effectiveness, and imperceptibility. The results are obtained on AG News and BERT, and the detailed explanation of the metrics can be found in §3.3, the complete results can be found in §3.4. When we only change the value of *Cand.#* from 10 to 50 while keeping the other parts of the attack unchanged, *Que.#* increases 121%, *A.S%* increases 19%, *ΔPPL%* decreases 1.1%, *ΔGErr.#* decreases 5.3%, *Pert.#* decreases 15%, and *USE.Sim* improves 0.94%, demonstrating the significant impact of search space.

Despite the significance of these two components, previous works on word-level adversarial

* Corresponding Author.

LREC-COLING 2024, pages 14037–14052
20-25 May, 2024.

attacks primarily concentrated on devising new attack methods while not adequately distinguishing the roles of search space and search method. This lack of distinction makes the asserted superiority of certain methods suspicious, as their performance may not be fairly compared with others. The superiority could stem from an improved search method or merely from constraints of varying strictness. For example, compared to Genetic Algorithm (GA) (Alzantot et al., 2018), Improved Genetic Algorithm (Improved-GA) (Wang et al., 2019) not only refines the search method by permitting multiple substitutions for the same word position but also increases the number of candidate words (Cand.#) selected from the search space at each step from 8 to 50. Figure 1 shows a more intuitive example of the search space's influence. We argue that the impact of search space is much greater than search method, which has often been overlooked by previous works. Under this circumstance, it is not a good trend for the community to merely consider the goal of new adversarial attack methods as improving state-of-the-art (SOTA), e.g., on attack success rate (*A.S%*).

Moreover, due to the unclear impact of search space on attacks, previous works often use a pre-defined search space without justifying the choice of parameters and constraints (i.e., they fail to answer questions like *why the threshold of semantic similarity is set to 0.85*). This ambiguity also hinders the adaptability of attacks, making it difficult to adjust them for various scenarios. For example, if adversarial examples aim to deceive models without human detection, the search space should filter out low-quality words, prioritizing imperceptibility. Conversely, in situations requiring numerous adversarial examples, such as evaluating model robustness and augmenting training data, the search space should emphasize effectiveness. However, a search space with unclear underlying motivations fails to achieve the different trade-offs between efficiency, effectiveness, and imperceptibility in different scenarios.

In this paper, to address these inappropriate practices in previous works, thus facilitating fair comparisons and improving the adaptability of attack methods, we investigate the impact of search space on the efficiency, effectiveness, and imperceptibility of word-level adversarial attacks by thorough ablation studies. We also propose the **S**earch **S**pace for **I**m**P**erceptibility (SSIP) and **S**earch **S**pace for **E**ffec**T**iveness (SSET) that improve the imperceptibility and effectiveness of attacks. Our primary contributions are summarized as follows:

1. We decompose previous word-level adversarial attacks and perform thorough ablation studies on their search space, illustrating the substantial influence of search space on attack efficiency, effectiveness, and imperceptibility, revealing the challenge of balancing these three factors.
2. We propose SSIP and SSET, two standardized search spaces that respectively emphasize the imperceptibility and effectiveness of attacks, constructed by carefully combining constraints and tuning parameters.
3. We reevaluate eight previous attack methods under SSIP and SSET against BERT on AG News and Movie Review (MR) datasets, providing fair and comprehensive evaluations of previous attack methods, demonstrating the success of SSIP and SSET in achieving better trade-offs between efficiency, effectiveness, and imperceptibility across various scenarios.

2. Related Works

Adversarial Attack. Motivated by early research on adversarial attacks that primarily targeted computer vision (CV) (Goodfellow et al., 2015; Papernot et al., 2016; Carlini and Wagner, 2017), several methods for attacking language models have been proposed. Unlike images, where pixels are continuous and differentiable, text is discrete and non-differentiable. Therefore, adversarial attacks in natural language processing (NLP) tasks are more suitably framed as combinatorial optimization problems, aiming to find optimal substitutions within the search space. Although several previous studies (Gao et al., 2018; Garg and Ramakrishnan, 2020; Jin et al., 2020; Zhan et al., 2022a; Li et al., 2021) are performed under the combinatorial optimization framework, they do not explicitly differentiate between the search space and search method.

Distinguish Between Search Space and Search Method. On the other hand, some studies emphasize the importance of distinguishing between search space and search method. Yoo et al. (2020) conducts ablation studies on search methods used in previous work, including Word Importance Ranking (WIR) (Gao et al., 2018; Li et al., 2019; Jin et al., 2020; Zhan et al., 2022b; Li et al., 2020; Zhan et al., 2023a), Greedy Search (Pruthi et al., 2019; Li et al., 2021), Beam Search (Ebrahimi et al., 2018), Genetic Algorithm (GA) (Alzantot et al., 2018), and Particle Swarm Optimization (PSO) (Zang et al., 2020). Nonetheless, their comparisons are performed within a pre-defined search space, ignoring the impact of search space. Morris et al. (2020) concentrates on designing a search space for imperceptible attacks, but their approach lacks an analysis of how the strictness of constraint could influence the attack imperceptibility.

Robustness Benchmarking. Recent studies on robustness benchmarking aim to compare the robustness of language models and the effectiveness of adversarial attack methods (Wang et al., 2021; Kiela et al., 2021; Chen et al., 2022). These studies typically employ the attack methods as defined in their original papers, e.g., PWWS (Ren et al., 2019), BERT-ATTACK (Li et al., 2020), to generate adversarial test sets, and conduct their benchmarking on a selection of predefined models, e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and T5 (Raffel et al., 2020). In contrast, our paper investigates how the *search space* of word-level attacks impacts the attack efficiency, effectiveness, and imperceptibility. This target distinguishes our work from previous studies: we decompose attack methods, replace their search space, and benchmark their search methods, rather than reusing existing attack methods to benchmark models.

3. Impact of Search Space

3.1. Word-level Adversarial Attack

Suppose we have a model $f : \mathcal{X} \rightarrow \mathcal{Y}$ that is trained by minimizing the empirical risk over all the given texts $\boldsymbol{X} \in \mathcal{X}$ and labels $Y \in \mathcal{Y}$, following the distribution $\mathcal{D}$:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathcal{L}\left(f\left(\boldsymbol{X};\boldsymbol{\theta}\right), Y\right) \ , \tag{1}$$

where $\boldsymbol{\theta}$ denotes the model parameters, and $\mathcal{L}$ denotes the loss objective. Ideally, the trained model should predict the input text as the ground-truth class based on the posterior probability:

$$\underset{Y\in\mathcal{Y}}{\operatorname{argmax}} \ \mathcal{P}(Y|\boldsymbol{X}) = Y_{true} \ , \tag{2}$$

where $\mathcal{P}(\cdot|\cdot)$ denotes posterior probability and Y_{true} denotes the ground-truth class of the input text $\boldsymbol{X}$. Under the framework of combinatorial optimization, word-level adversarial attack can be regarded as an iterative process, where the attack keeps trying to introduce slight perturbation to the normal input text $\boldsymbol{X} = (x_n)_{n\in\{1,\ldots,N\}}$ in each step. Therefore, we can formally define an adversarial example $\boldsymbol{X}^{adv}$ that generated from the normal example as:

$$\begin{aligned}
&\boldsymbol{X}^{adv} = \mathcal{O}(\boldsymbol{X};\mathcal{W}) = o(x_n;\mathcal{W}_{x_n})_{n\in\{1,\ldots,N\}}, \\
&\text{s.t.} \quad \forall n \in \{1,\ldots,N\}, \ \ \Delta x_n < \delta, \\
&\text{and} \quad \Delta\boldsymbol{X} < \varepsilon, \\
&\text{and} \quad \underset{Y\in\mathcal{Y}}{\operatorname{argmax}} \ \mathcal{P}(Y|\boldsymbol{X}^{adv}) \neq \underset{Y\in\mathcal{Y}}{\operatorname{argmax}} \ \mathcal{P}(Y|\boldsymbol{X}),
\end{aligned} \tag{3}$$

where $\mathcal{O}(\boldsymbol{X};\mathcal{W})$ denotes substituting the words in sentence $\boldsymbol{X}$ with the words from search space $\mathcal{W}$ that contains all potential substitutions, $o(x_n;\mathcal{W}_{x_n})$ denotes substituting word x_n with the word from $\mathcal{W}_{x_n}$, the search subspace of word x_n. The difference between x_n and $o(x_n;\mathcal{W}_{x_n})$ is denoted by Δx_n, while the difference between $\boldsymbol{X}$ and $\mathcal{O}(\boldsymbol{X};\mathcal{W})$ is denoted by $\Delta\boldsymbol{X}$. The search space $\mathcal{W}$ is restricted by constraints that limit the maximum allowed difference between words and substitutions, denoted by δ, and the modified sentence and the original sentence, denoted by ε. The measurement of difference may focus on various metrics, e.g., semantic similarity, which filters out the substitutions that may cause the generated examples to be perceptible to humans. In this paper, we fix the search method, i.e., the strategy of performing $o(\cdot;\cdot)$, and try to show how the search space and constraints could impact the attack.

Search Space & Constraint	Implementation	Attack Method
Basic Search Space	Masked Language Model	A2T (Yoo and Qi, 2021), BAE (Garg and Ramakrishnan, 2020), BERT-ATTACK (Li et al., 2020), CLARE (Li et al., 2021)
	Counter-fitted GloVe	A2T, GA (Alzantot et al., 2018), Faster-GA (Jia et al., 2019), Improved-GA (Wang et al., 2019), TextBugger (Li et al., 2019), TextFooler (Jin et al., 2020)
	HowNet	PSO (Zang et al., 2020)
	WordNet	PWWS (Ren et al., 2019)
Semantic Constraint	Sentence-level Similarity	BAE, BERT-ATTACK, CLARE, TextBugger, TextFooler
	Word-level Similarity	A2T, GA, Faster-GA Improved-GA, TextFooler
Grammatical Constraint	Part-of-Speech	A2T, BAE, TextFooler
	Stop Word	A2T, BAE, BERT-ATTACK, CLARE, GA, Faster-GA, Improved-GA, PSO, PWWS, TextBugger, TextFooler

Table 1: Decomposed search space and constraints utilized in previous attack methods.

3.2. Search Space and Constraints in Previous Works

Following our setting described in §3.1, we decompose eleven prominent word-level adversarial attack methods, with a particular emphasis on the search space and constraints they employ, as illustrated in Table 1. Due to the space and format limitations, comprehensive details about the relative search spaces and constraints, such as the specific minimum allowed sentence semantic similarity, are provided in Appendix A.1. In the following, we explain the search space and constraints used in previous works.

Basic Search Space. The basic search space comprises all possible substitutions without applying any constraints. Previous works commonly utilized Masked Language Model (MLM) (Devlin et al., 2019), counter-fitted GloVe (Mrksic et al., 2016), HowNet (Dong and Dong, 2003), and WordNet (Miller, 1992) as the basic search space. MLM generates potential substitutions for the target word based on contextual information. Counter-fitted GloVe learns word embeddings where the embeddings of synonyms cluster together, and those of antonyms are pushed apart. HowNet and WordNet are both knowledge-based resources that organize words into lexical hierarchies and provide information on semantic relations between words. Both search spaces provide potential substitutions for the target word in each attack step during attacks. The details on selecting candidates from the basic search space are in Appendix A.2.

Semantic Constraint. The semantic constraint limits potential substitutions to words semantically similar to the original word (word-level) and maintains the overall semantics of the generated examples (sentence-level). To obtain sentence-level semantic similarity, previous works commonly use the Universal Sentence Encoder (USE) (Cer et al., 2018) and BERTScore (Zhang et al., 2020). To obtain word-level semantic similarity, counter-fitted GloVe is commonly utilized. During the attack, cosine similarity is used for USE and counter-fitted GloVe to measure the semantic similarity between the target and possible representations.

Grammatical Constraint. The grammatical constraint limits the possible substitutions to words that maintain the grammatical correctness of the generated examples. Previous works use part-of-speech (POS) and stop word constraints to ensure grammatical correctness. Part-of-speech constraint limits substitutions to words with the same part-of-speech as the target word, while stop word constraint prevents substituting words that are significant for maintaining grammatical correctness.

3.3. Ablation Study Setup

Setup. We use *WIR* and *Greedy Search* as search methods, which are the two most frequently used search methods in previous works. We conduct experiments on the *MR* (Pang and Lee, 2005), *AG News* (Zhang et al., 2015), and *SST2* (Socher et al., 2013) datasets. More details of the datasets can be found in Appendix A.3. We use the base version of *BERT* as the target model. In the experiments, the clean accuracy of BERT, which is fine-tuned on the MR, AG News, and SST2, achieves 87.43%, 95.07%, and 92.26% respectively.

To evaluate the impact of the basic search space, we consider: (1) the choice of basic search space, and (2) the number of candidate words (*Cand.#*) in each attack step. Specifically, we use *MLM-BERT* (Devlin et al., 2019), *MLM-RoBERTa* (Liu et al., 2019), *MLM-DistilBERT* (Sanh et al., 2019), *counter-fitted GloVe* (Mrksic et al., 2016), *WordNet* (Miller, 1992), and *HowNet* (Dong and Dong, 2003) as basic search spaces. In each search space, we set the *Cand.#* to 10, 20, 30, 40, 50, 60, 70, and 80.

To evaluate the impact of sentence-level semantic constraints, we consider: (1) the method to obtain semantic similarity, and (2) the minimum allowed similarity (*Min.Sim*) between the original sentence and the generated examples. Specifically, we use *USE* (Cer et al., 2018) and *BERTScore* (Zhang et al., 2020) to obtain the semantic similarity of sentences, and set *Min.Sim* to 0.5, 0.6, 0.7, 0.8, 0.9, and 0.95. To evaluate the impact of word-level semantic constraints, we consider the *Min.Sim* between the original word and its substitution. Specifically, we use counter-fitted GloVe (Mrksic et al., 2016) to encode words, as in previous works, and set *Min.Sim* to 0.5, 0.6, 0.7, 0.8, 0.9, and 0.95.

To evaluate the impact of grammatical constraints, we compare the results *with* and *without* part-of-speech and stop word constraints. Additionally, for stop word constraint, we compare the impact of different pre-defined stop word sets, including the stop words defined in *NLTK* (Bird et al., 2009), *spaCy* (Honnibal et al., 2020), and *TextFooler* (Jin et al., 2020).

Metrics for Evaluation. We use *Attack Success Rate (A.S%)* to measure the effectiveness. Following Li et al. (2020), Jin et al. (2020) and Chen et al. (2022), we use the number of queries (*Que.#*) made to the target model to measure the efficiency. We use the *Number of Perturbed Words (Pert.#)*, *Increased Perplexity Ratio (ΔPPL%)* (Jelinek et al., 1977), *Increased Number of Grammatical Errors (ΔGErr.#)*, and *USE Similarity (USE.Sim)* to measure imperceptibility. Specifically, the PPL is calculated with GPT-2 (Radford et al., 2019), the Δ*GErr.#* is detected by LanguageTool [1], and the *USE.Sim* is calculated by the large version of USE.

3.4. Ablation Study on Search Space

We conduct attacks on 500 randomly selected examples and report the average results from two independent runs, i.e., 1000 examples supporting each ablation result. Due to space and format constraints, this section presents the results of attacks on AG News against BERT using the WIR method. Comprehensive results for attacks on MR and SST2, as well as those using the Greedy Search method, are provided in Appendix B.1. Please note that the ensuing analysis is not limited to the results reported in this section; rather, it probes into the common impacts observed when attacking different datasets and employing various

[1] `https://languagetool.org`

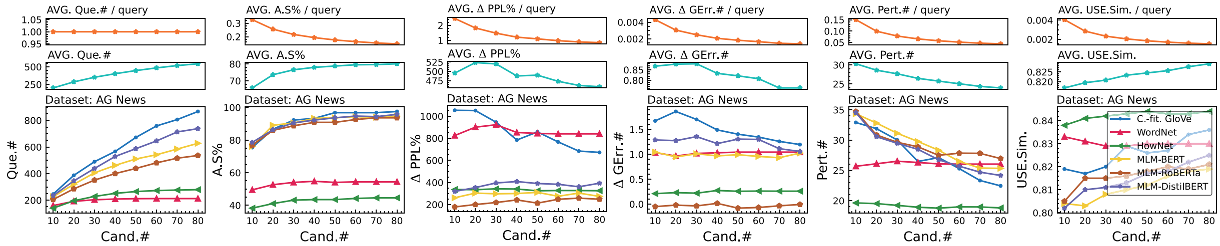

Figure 2: The impact of basic search space on attack efficiency, effectiveness, and imperceptibility when attacking AG News against BERT with WIR. The upper plot shows the balance between a specific metric and Que.#, the efficiency. The middle plot shows the average results across all basic search spaces. The lower plot shows the detailed results on different basic search spaces.

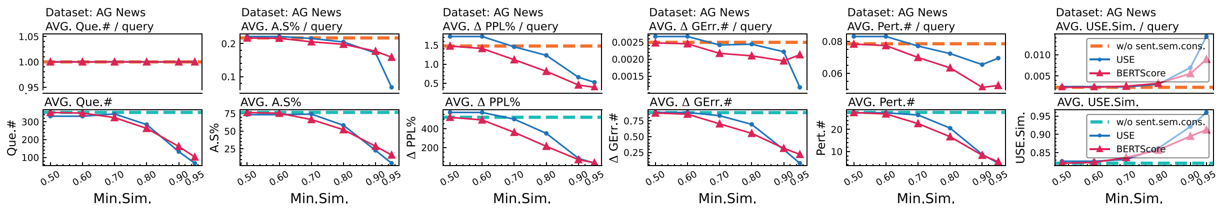

Figure 3: The impact of sentence-level semantic constraint (*sent.sem.cons*) on attack efficiency, effectiveness, and imperceptibility when attacking AG News against BERT with WIR. The results are obtained with Cand.# set to a moderate value of 30. The upper plot shows the balance between a specific metric and Que.#, the efficiency. The lower plot shows the average results across all basic search spaces.

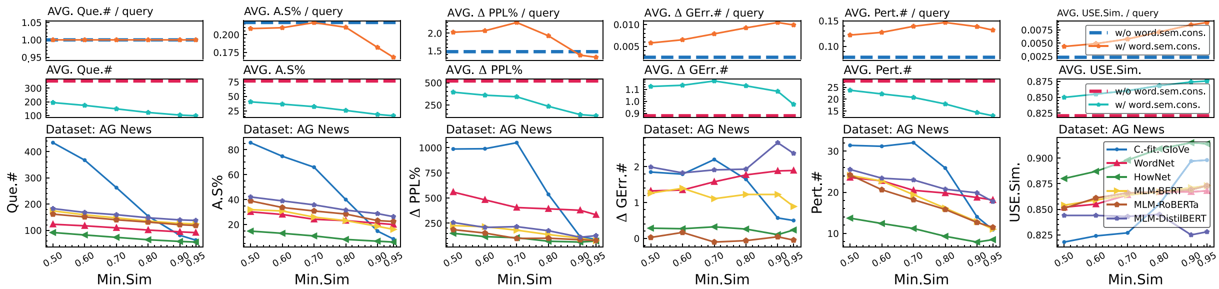

Figure 4: The impact of word-level semantic constraint (*word.sem.cons*) on attack efficiency, effectiveness, and imperceptibility when attacking AG News against BERT with WIR. The results are obtained with Cand.# set to a moderate value of 30. Stacked plots have the same meaning as in Figure 2.

search methods. Furthermore, in the following analysis, we provide rationales for constructing SSIP and SSET (detailed in §4.1), where the text is respectively marked with rationales for SSIP$_{(SSIP)}$ and rationales for SSET$_{(SSET)}$.

Impact of Basic Search Space. The ablation results of basic search space are in Figure 2. Under the same Cand.#, HowNet and WordNet consistently require fewer Que.# to complete the attacks compared to other search spaces$_{(SSIP)}$. While MLM-based search spaces and counter-fitted GloVe need more Que.#, attacks utilizing these search spaces are significantly more effective (higher A.S%). Furthermore, within MLM-based search spaces and counter-fitted GloVe, MLM-RoBERTa consistently requires fewer Que.# while achieving comparable A.S%$_{(SSET)}$. Counter-fitted GloVe and WordNet result in adversarial examples with much higher ΔPPL.%, whereas MLM-based spaces and HowNet only slightly increase ΔPPL.%$_{(SSIP)}$, especially MLM-BERT and MLM-RoBERTa. MLM-RoBERTa generates adversarial examples with fewer GErr.# than other search spaces, and sometimes produces examples with fewer GErr.# than the original examples. It is worth noting that, although HowNet's performance in maintaining grammatical correctness is not the best, it consistently achieves results very close to optimal$_{(SSIP)}$. For Pert.# and USE.Sim, MLM-DistilBERT generally perturbs fewer words in the attacks compared to other spaces, while counter-fitted GloVe and MLM-DistilBERT always outperform other MLM-based spaces in maintaining sentence similarity in most cases. The effectiveness of HowNet and WordNet in reducing Pert.# and maintaining sentence similarity varies across different datasets and search methods, but HowNet consistently requires fewer

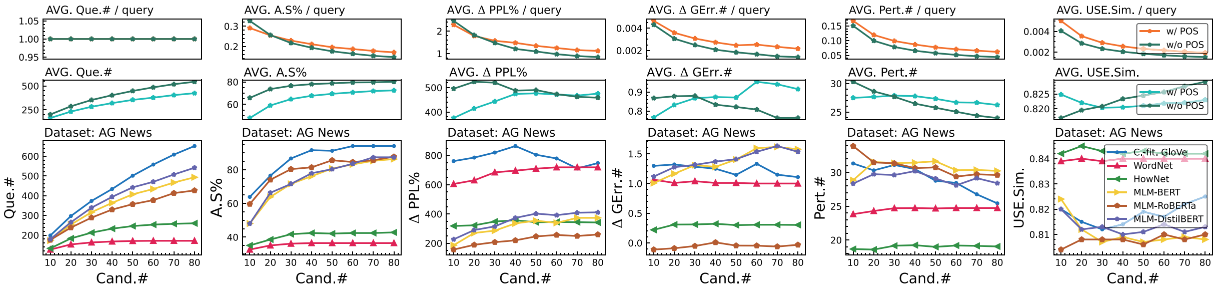

Figure 5: The impact of *POS* constraint on attack efficiency, effectiveness, and imperceptibility when attacking AG News against BERT with WIR. Stacked plots have the same meaning as in Figure 2.

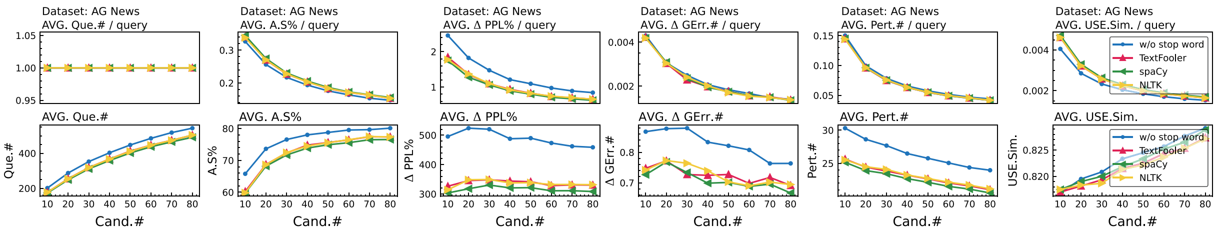

Figure 6: The impact of stop word constraint on attack efficiency, effectiveness, and imperceptibility when attacking AG News against BERT with WIR. Stacked plots have the same meaning as in Figure 3.

Pert.# and achieves higher USE.Sim than WordNet and is often close to the best-performing space. Considering the much lower Que.# of HowNet, the results on imperceptibility are competitive.$_{(SSIP)}$

WordNet and HowNet are less sensitive to Cand.#, and when Cand.# increases, all metrics change slightly$_{(SSIP)}$, which may be due to the limited substitutions provided by the knowledge bases for target words. Thus, a small Cand.# is often enough for them. In contrast, other search spaces are sensitive to Cand.#. When Cand.# increases, Que.# and A.S% in MLM-based spaces significantly increase, especially Que.#$_{(SSET)}$, while imperceptibility-related metrics show little variation. Specifically, increasing Cand.# helps reduce Pert.# and increase USE.Sim, but is less effective in improving ΔPPL.% and ΔGErr.#. Furthermore, based on the average results, we find that when Cand.# increases, the cost of attacks (Que.#) increases much faster than the benefits (A.S%, ΔPPL.%, ΔGErr.#, Pert.#, USE.Sim), as the change in these metrics per query (e.g., A.S% / query) decreases. Therefore, considering this asymmetric cost-benefit ratio, blindly using larger Cand.# values is inappropriate.

Impact of Sentence-level Semantic Constraint. The ablation results of sentence-level semantic constraint are in Figure 3. Using BERTScore and USE to maintain the semantics of adversarial examples both negatively impacts effectiveness$_{(SSET)}$, with USE leading to a greater reduction in A.S% than BERTScore. Nevertheless, both BERTScore and USE help generate more imperceptible adversarial examples, although ΔGErr.# may occasionally increase when using USE$_{(SSIP)}$. When USE is used in the attacks on MR with Greedy Search and the attacks on SST2 with WIR/Greedy Search, the ΔGErr.# increases when Min.Sim is larger than 0.90. Please see Appendix B.1 for detailed results.

Impact of Word-level Semantic Constraint. The ablation results of word-level semantic constraint are in Figure 4. Similar to the sentence-level semantic constraint, applying word-level semantic constraint also negatively impacts the effectiveness$_{(SSET)}$, as A.S% is consistently lower than without using the constraint. However, unlike the sentence-level semantic constraint, applying word-level semantic constraint does not always benefit imperceptibility$_{(SSIP)}$, as it consistently increases the grammar errors of adversarial examples (higher ΔGErr.#). Moreover, this constraint may also negatively impact PPL, perturbed words, and sentence semantics, e.g., generated adversarial examples get higher ΔPPL.% and Pert.#, and lower USE.Sim when the attacks are performed on MR and SST2 with Greedy Search. Please also see Appendix B.1 for additional results.

Impact of Part-of-speech Constraint. The ablation results of part-of-speech constraint are in Figure 5. The part-of-speech constraint negatively impacts effectiveness$_{(SSET)}$ and does not consistently improve most aspects of imperceptibility$_{(SSIP)}$. Applying the constraint helps generate adversarial examples with lower ΔPPL. However, the ΔGErr.# is always higher than the results without using the constraint. Additionally, this constraint results in slightly higher Pert.# and slightly lower USE.Sim compared to not using the constraint in most cases.

Impact of Stop Word Constraint. The ablation results of stop word constraint are in Figure 6. Utilizing the stop word constraint negatively impacts the attack efficiency$_{(SSET)}$, as A.S% generally decreases. However, the constraint significantly improves most aspects of imperceptibility in most cases, including ΔPPL.%, ΔGErr.#, and Pert.#$_{(SSIP)}$. At its worst, the constraint only has an extremely marginal negative impact on these metrics on SST2 when using greedy search. The impact of different pre-defined stop words is similar, with NLTK-defined stop words slightly outperforming others in improving imperceptibility and maintaining effectiveness in most cases, which is not pronounced$_{(SSIP)}$.

3.5. Discussion on Search Space

Based on the ablation studies, we can achieve several crucial insights into the impact of search space:

(1) *Each constraint influences all aspects of attack efficiency, effectiveness, and imperceptibility, even if the constraint is designed to optimize a specific aspect of attacks*. For instance, sentence-level semantic constraint is intended to improve the imperceptibility of attacks, while it impedes the efficiency and effectiveness of attacks. Therefore, it is essential to report not only the positive results for certain attack aspects but also the potential negative effects on other aspects of efficiency, effectiveness, and imperceptibility.

(2) *Constraints targeting specific aspects of imperceptibility may also have negative impacts on other aspects of imperceptibility*. For instance, part-of-speech constraint consistently decreases ΔPPL.% of adversarial examples, while it also actually increases the ΔGErr.#, Pert.#, and decreases USE.Sim of adversarial examples. Therefore, it is essential to report the impact on all aspects of imperceptibility rather than on only the targeted one.

(3) *The efficiency of attacks is more sensitive to variations in search space than effectiveness and imperceptibility*. For instance, as illustrated in Figure 1, 2, when the Cand.# increases from 10 to 50, the Que.# of attacks increases 121%, while other metrics related to effectiveness and imperceptibility only change relatively slightly. Therefore, the efficiency, i.e., the cost of the attack, should be prioritized before striving for superiority in effectiveness or imperceptibility.

(4) *The efficiency, effectiveness, and imperceptibility of attacks are often incompatible with each other, making it challenging to achieve a balance between them*. For instance, low-quality adversarial samples may be more effective in attacking models but are also more perceptible to humans and easier to generate. Similarly, using a large Cand.# value benefits both effectiveness and imperceptibility, but it leads to significantly lower efficiency. Therefore, a compromise among the optimized aspects may be necessary, depending on the scenario. Utilizing a seemingly balanced search space to optimize attack efficiency, effectiveness, and imperceptibility only results in moderate results, hindering the adaptability of attack methods.

Accordingly, we can further summarize the common inappropriate practices in previous works: (1) Settings of search spaces are insufficiently detailed. (2) Comparisons are conducted across different search spaces. (3) Evaluations are insufficient, lacking results on efficiency, effectiveness, or every aspect of imperceptibility. Admittedly, the search method may be more crucial for a paper to express its novelty, but the search space is essential for ensuring fair comparisons and sufficient evaluations.

4. Achieve Better Trade-offs in Different Scenarios

4.1. SSIP and SSET

To address these inappropriate practices and facilitate fair comparisons while improving the adaptability of attack methods, it is essential to evaluate different attack methods within standardized search spaces that emphasize various aspects of efficiency, effectiveness, and imperceptibility. Therefore, in this paper, we propose SSIP and SSET. By modifying the search space *without changing the core attack rules defined in the search method*, the different aspects of characteristics can be better emphasized. The details of SSIP and SSET are described below, and the rationales for constructing SSIP and SSET are provided in §3.4. It should be noted that SSET and SSIP are not trying to maximize the superiority of a specific search method but to ensure broader superiority across various search methods.

SSIP. SSIP emphasizes imperceptibility while ensuring efficiency is acceptable and should be used in scenarios requiring high-quality adversarial examples, e.g., bypassing defense system detection and preventing human detection. Based on the rationales in §3.4, the detailed search space and

	Effectiveness	Efficiency		Imperceptibility				Effectiveness	Efficiency		Imperceptibility			
Method	A.S%↑	A.S% / Q.↑	Q.# / S.A.↓	△PPL%↓	△GErr.#↓	Pert.#↓	USE.Sim.↑	A.S%↑	A.S% / Q.↑	Q.# / S.A.↓	△PPL%↓	△GErr.#↓	Pert.#↓	USE.Sim.↑
	AG News							*MR*						
↱ SSIP	9.66	0.127	790	**40.8**	**0.005**	**4.86**	**0.921**	29.76	0.897	112	**32.1**	0.048	**6.61**	**0.882**
BAE	17.87	0.135	757	154.5	1.155	6.71	0.912	61.34	0.944	105	42.9	0.101	15.15	0.841
↳ SSET	**81.62**	**0.291**	**344**	212.4	0.111	30.71	0.813	**96.55**	**1.217**	**82**	109.6	**0.002**	20.62	0.816
↱ SSIP	14.84	0.034	2965	69.1	0.071	**7.21**	**0.894**	32.58	0.130	779	**45.8**	0.069	**8.63**	**0.865**
GA	25.96	0.051	2159	112.1	0.811	12.83	0.884	52.39	0.194	515	119.3	0.364	15.94	0.846
↳ SSET	**37.19**	**0.069**	**1798**	**51.8**	**0.053**	11.52	0.881	**74.92**	**0.230**	**436**	53.3	−**0.021**	15.38	0.833
↱ SSIP	20.91	0.004	25 315	53.5	0.005	**7.92**	**0.892**	51.93	0.020	4967	41.2	0.007	**9.77**	**0.867**
Faster-GA	16.49	0.032	3375	73.5	0.191	12.08	0.889	43.08	0.155	645	54.3	0.176	15.05	0.853
↳ SSET	**41.75**	**0.039**	**2606**	**47.6**	−**0.118**	12.53	0.854	**92.57**	**0.252**	**398**	**29.6**	−**0.024**	12.45	0.845
↱ SSIP	21.07	0.014	6929	68.6	0.068	**7.61**	**0.897**	49.94	0.104	960	47.7	0.052	**8.95**	**0.870**
Improved-GA	32.98	0.018	5742	148.7	0.745	12.56	0.879	78.28	0.113	854	121.6	0.338	15.22	0.838
↳ SSET	**50.18**	**0.021**	**5004**	**48.3**	−**0.112**	12.22	0.856	**95.88**	**0.117**	**821**	**34.4**	−**0.030**	12.87	0.850
↱ SSIP	10.84	0.169	615	**66.2**	**0.649**	**12.71**	**0.919**	27.63	0.844	119	**42.7**	**0.343**	**7.64**	**0.897**
TextBugger	54.38	**0.306**	418	426.6	3.037	34.89	0.867	58.71	1.045	103	159.2	1.245	15.44	0.875
↳ SSET	**87.26**	0.239	**367**	537.1	2.988	43.02	0.809	**98.36**	**1.067**	**93**	200.7	0.677	23.16	0.826
↱ SSIP	8.33	0.105	947	**51.6**	0.167	**6.11**	**0.914**	28.71	0.826	121	**31.8**	0.055	**7.02**	**0.885**
TextFooler	50.63	0.209	482	444.5	1.195	23.45	0.872	71.84	0.855	117	156.8	0.387	19.61	0.839
↳ SSET	**85.15**	**0.322**	**314**	218.8	−**0.043**	27.82	0.809	**96.42**	**1.172**	**85**	81.6	−**0.017**	20.63	0.819
↱ SSIP	25.61	**0.011**	**9147**	**56.5**	0.021	**5.91**	**0.903**	47.48	**0.118**	**851**	**33.2**	0.031	**6.92**	**0.886**
PSO	56.84	0.007	15 261	198.6	0.123	14.75	0.851	90.11	0.071	1415	89.3	0.082	13.25	0.844
↳ SSET	**94.04**	0.008	14 804	87.7	−**0.006**	15.99	0.847	**99.96**	0.075	1369	37.0	**0.016**	12.78	0.857
↱ SSIP	20.77	0.088	1133	**46.9**	0.068	**5.18**	**0.911**	42.14	0.359	278	**30.6**	0.034	**6.73**	**0.889**
PWWS	46.46	0.119	835	459.7	0.797	17.79	0.846	80.16	**0.517**	239	148.8	0.262	15.66	0.831
↳ SSET	**82.11**	**0.163**	**612**	153.8	−**0.022**	22.76	0.821	**98.77**	0.323	**194**	49.9	−**0.027**	14.42	0.846

Table 2: The results of efficiency, effectiveness, and imperceptibility when previous attack methods are conducted in different search spaces. The rows for each attack method (e.g., PWWS) imply attacks using their original search spaces, with SSIP/SSET denoting changed search spaces only. *A.S% / Q.* is short for *Attack Success Rate per Query*, *Q.# / S.A.* is short for *Number of Queries Needed for Each Successful Attack*. The **bold** values indicate the best results, and the underline values for △GErr.# and △PPL% indicate the second-best results.

constraints of SSIP are: (1) using HowNet as the basic search space and setting Cand.# to 20, (2) using BERTScore to measure sentence-level semantics and setting Min.Sim to 0.95, (3) not using word-level semantic constraint, (4) not using part-of-speech constraint, and (5) using stop word constraint, with stop words as defined in NLTK.

SSET. SSET emphasizes effectiveness while ensuring efficiency is acceptable[2] and should be used in scenarios requiring numerous adversarial examples, e.g., augmenting data and evaluating model robustness. Based on the rationales in §3.4, the detailed search space and constraints of SSET are: (1) using MLM-RoBERTa as the basic search space and setting Cand.# to 20, (2) not using sentence-level semantic constraint, (3) not using word-level semantic constraint, (4) not using part-of-speech constraint, and (5) not using stop word constraint.

4.2. Reevaluation Results

We replace the search space while maintaining the search method in previous attack methods, then perform attacks on 500 randomly selected examples and report the average results from two independent runs. More details on the attack methods can be found in Appendix A.4. Table 2 shows the reevaluation results of attacking AG News and MR against BERT with previous attack methods.

When previous attacks are performed under SSIP, various aspects of imperceptibility improve. Specifically, the average △PPL%, △GErr.#, Pert.#, and USE.Sim of the original attacks are respectively 181.88%, 0.69, 16.27, and 0.86, while under SSIP are 47.39% (↓73.94%), 0.11 (↓84.61%), 7.48 (↓54.01%), and 0.90 (↑3.67%). When previous attacks are performed under SSET, both effectiveness and efficiency improve. Specifically, the average A.S%, A.S%/Q., and Que.# / S.A are respectively 52.34%, 0.30, and 2063, while under SSET are 82.04% (↑ 56.74%), 0.35 (↑14.84%), and 1832 (↓11%). Moreover, the reevaluation results under SSIP and SSET provide a fair comparison and thorough evaluations of previous attack methods. The results show that PSO generally achieves the best effectiveness (highest A.S% in SSET), while BAE tends to achieve the best imperceptibility (lowest △PPL% and Pert.#, second-lowest △GErr.#, and second-highest USE.Sim in SSIP).

[2]Specifically, SSIP and SSET should prioritize imperceptibility and effectiveness, respectively, without compromising efficiency by using excessively aggressive parameters. However, some heuristic attack methods, such as GA, and PSO, may inherently exhibit low efficiency.

5. Conclusion

In this paper, we demonstrate the substantial role of the search space in influencing the efficiency, effectiveness, and imperceptibility of word-level adversarial attacks, as evidenced by thorough ablation studies. Our findings yield several crucial insights into the search space and offer potential guidance for future research on word-level adversarial attacks. To promote fair comparisons and enhance the adaptability of attacks across various scenarios, we introduce two standardized search spaces: SSIP and SSET. Reevaluations of previous attack methods illustrate the success of SSIP and SSET in augmenting the imperceptibility and effectiveness of attacks, while also providing a robust framework for facilitating fair and comprehensive evaluations of word-level adversarial attack methodologies.

Limitations

Our study mainly focuses on BERT as the victim model, as the ablation study requires numerous adversarial examples, and our computing resources are limited. Despite this, we believe the conclusions and insights in this work are generalizable, and this work succeeds in revealing inappropriate practices in previous works and prompting fair comparisons and comprehensive evaluations of adversarial attack methods. It is important to emphasize that the descriptions and conclusions presented in this paper are not intended to undermine previous works; we recognize that all previous research has contributed significantly to the advancement of the field. Moreover, we hope our study encourages the community to place increased emphasis on the role of search space in word-level adversarial attacks.

Ethics Statement

In this work, we strive to promote fairness and transparency in the evaluation of adversarial attack methods. While adversarial attack techniques can be employed to enhance the robustness of models and expose vulnerabilities, they can also be misused to compromise model performance or deceive users. However, we believe the findings in this paper will also contribute to a more accurate understanding of the strengths and weaknesses of existing methods, ultimately leading to the development of more robust and secure language models. We utilize publicly available datasets that do not contain sensitive information or personally identifiable information (PII), and we do not violate their licenses. Furthermore, our research follows ethical guidelines, demonstrating adversarial techniques safely without causing unintended harm.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable suggestions. This research was supported by National Research and Development Program of China (No.2019YFB1005200).

Bibliographical References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

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

Charles E. Blair. 1990. Integer and combinatorial optimization (george l. nemhauser and laurence a. wolsey). *SIAM Rev.*, 32(2).

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (SP)*. IEEE.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *ArXiv preprint*, abs/1803.11175.

Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. 2022. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering*. IEEE.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018*.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1).

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman*.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *CoRR*, abs/1909.06723.

Jin Yong Yoo, John Morris, Eli Lifland, and Yanjun Qi. 2020. Searching for a search method: Benchmarking search algorithms for generating NLP adversarial examples. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Pengwei Zhan, Yang Wu, Yunjian Zhang, Liming Wang, Chao Zheng, and Jing Yang. 2022a. Crafting textual adversarial examples through second-order enhanced word saliency. In *2022 International Joint Conference on Neural Networks (IJCNN)*.

Pengwei Zhan, Yang Wu, Shaolei Zhou, Yunjian Zhang, and Liming Wang. 2022b. Mitigating the inconsistency between word saliency and model confidence with pathological contrastive training. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Pengwei Zhan, Jing Yang, Xiao Huang, Chunlei Jing, Jingying Li, and Liming Wang. 2023a. Contrastive learning with adversarial examples for alleviating pathology of language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Pengwei Zhan, Jing Yang, He Wang, Chao Zheng, Xiao Huang, and Liming Wang. 2023b. Similarizing the influence of words with contrastive learning to defend word-level adversarial text attack. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Pengwei Zhan, Chao Zheng, Jing Yang, Yuxiang Wang, Liming Wang, Yang Wu, and Yunjian Zhang. 2022c. PARSE: an efficient search method for black-box adversarial text attacks. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*.

A. Additional Experimental Details

A.1. Detail Settings of Search Space in Previous Works.

- **Basic Search Space and Cand.#.** *A2T* uses the base version of BERT and counter-fitted GloVe as the basic search space, with Cand.# set to 20. *BAE* and *BERT-ATTACK* use the base version of BERT as the basic search space, with Cand.# set to 50, 48, respectively. *CLARE* uses the distilled version of base RoBERTa as the basic search space, with Cand.# set to 50. *GA*, *Faster-GA*, and *Improved-GA* use counter-fitted GloVe as the search space, setting Cand.# to 8, 8, and 50, respectively. *TextBugger* includes counter-fitted GloVe as the basic search space, with Cand.# set to 5. *TextFooler* uses counter-fitted GloVe as the basic search space, with Cand.# set to 50. *PSO* uses HowNet as the basic search space, with Cand.# set to the number of all possible substitutions. *PWWS* uses WordNet as the basic search space, with Cand.# set to the number of all possible substitutions.

- **Sentence-level Semantic.** *BAE*, *BERT-Attack*, *CLARE*, *TextBugger*, and *TextFooler* use USE to obtain sentence semantic similarity, setting Min.Sim to 0.8, 0.2, 0.7, 0.8, and 0.5, respectively.

- **Word-level Semantic.** *A2T*, *TextFooler*, *GA*, *Faster-GA*, and *Improved-GA* use counter-fitted GloVe to obtain word semantic similarity, setting Min.Sim to 0.8, 0.5, 0.5, 0.5, and 0.5, respectively.

- **Stop word.** *A2T*, *BAE*, *BERT-ATTACK*, *CLARE*, *GA*, *Faster-GA*, *Improved-GA*, *PSO*, *PWWS*, and *TextBugger* employ the stop words defined by NLTK in the stop word constraint, while *TextFooler* defines their own stop word list.

A.2. Details on Selecting Candidates from Search Space.

For counter-fitted GloVe, we calculate the cosine similarity between the target word and potential candidate words based on their embeddings, selecting the most similar words as candidates. For the MLM-based space, we select candidates based on the potential words that the MLM returns with the highest confidence. For WordNet, following previous work, we utilize the implementation provided by NLTK and select candidates by sampling their synonyms. For HowNet, we initially determine the similarity between the target word and potential words using sememes, choosing the most similar words as candidates.

A.3. Details on Dataset

The *MR* dataset contains movie reviews from Rotten Tomatoes, with examples labeled as positive or negative, comprising 8,530 training and 1,066 testing samples. The *SST2* dataset consists of sentences labeled as positive or negative, including 67,349 training and 1,821 testing samples. The *AG News* dataset features news articles categorized into four distinct groups: World, Sports, Business, and Science/Technology, comprising 120,000 training and 7,600 testing samples.

A.4. Detail Settings of Attack Method

For efficiency purposes, we set both the population size and the number of iterations of GA, Improved-GA, Faster-GA, and PSO to 10, rather than the 60 and 20 reported in the original paper. For BAE, TextBugger, TextFooler, and PWWS, we adopt the settings from their original papers.

B. Additional Experimental Results

B.1. Results of Ablation Study

In this section, we present the complete results of attacking MR and SST2 against BERT with both WIR and Greedy Search. We did not report the results of the ablation study on AG News with Greedy Search, as Greedy Search was extremely time-consuming to attack long sentences, and the ablation study required numerous adversarial examples. The results of attacking MR with WIR are in Figures 7-11, the results of attacking MR with Greedy Search are in Figures 12-16, the results of attacking SST2 with WIR are in Figures 17-21, and the results of attacking SST2 with Greedy Search are in Figures 22-26.

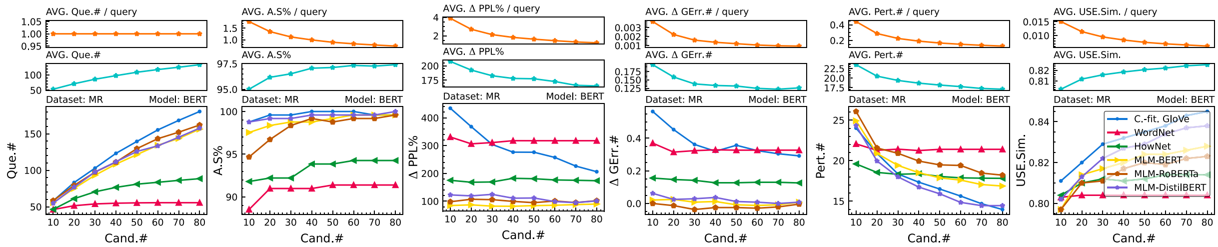

Figure 7: The impact of basic search space on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with WIR.

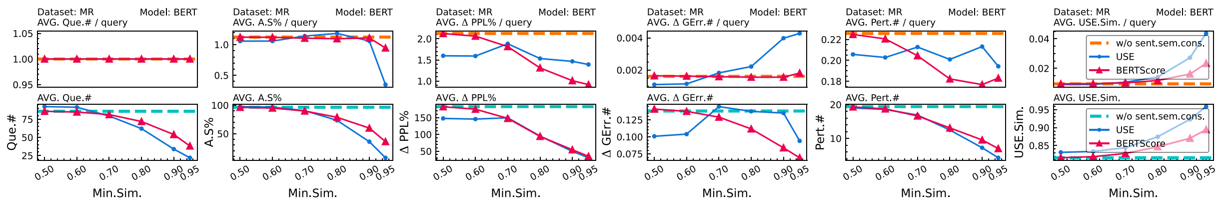

Figure 8: The impact of sentence-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with WIR.

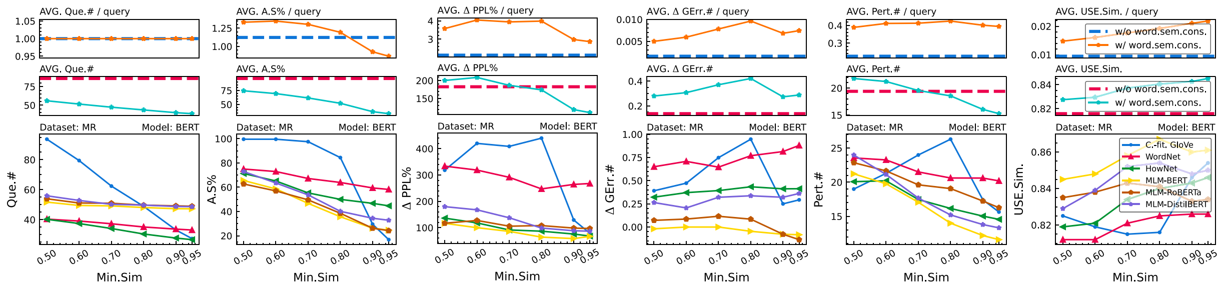

Figure 9: The impact of word-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with WIR.

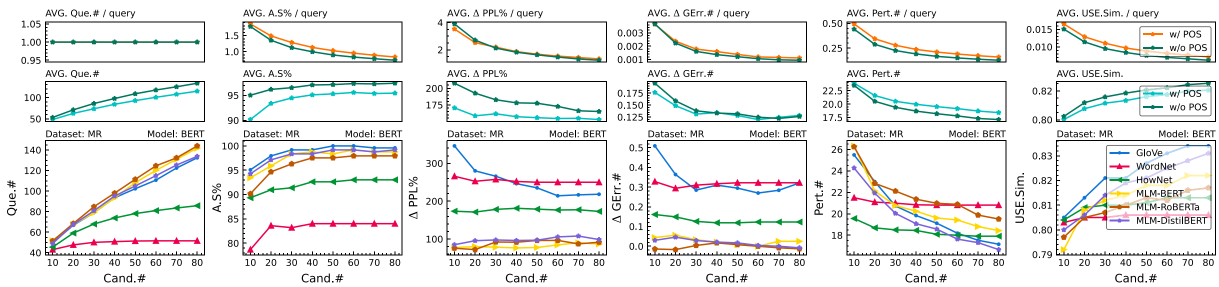

Figure 10: The impact of part-of-speech constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with WIR.

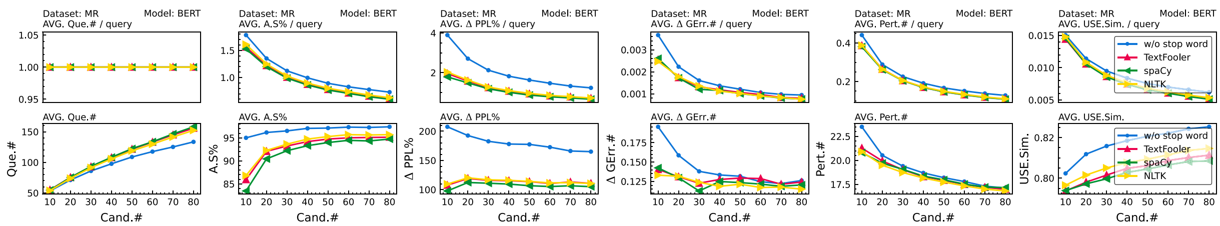

Figure 11: The impact of stop word constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with WIR.

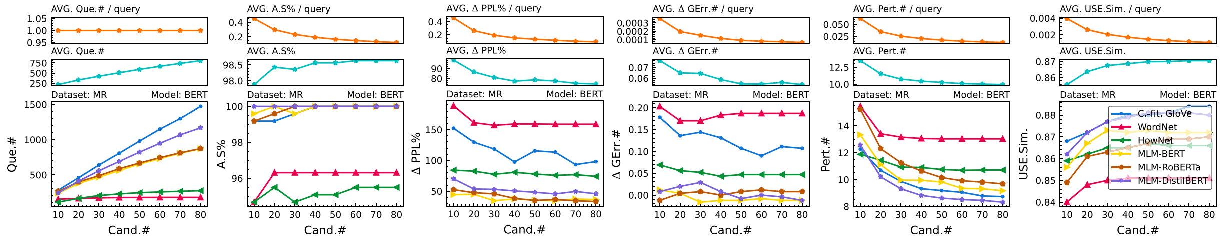

Figure 12: The impact of basic search space on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with Greedy Search.

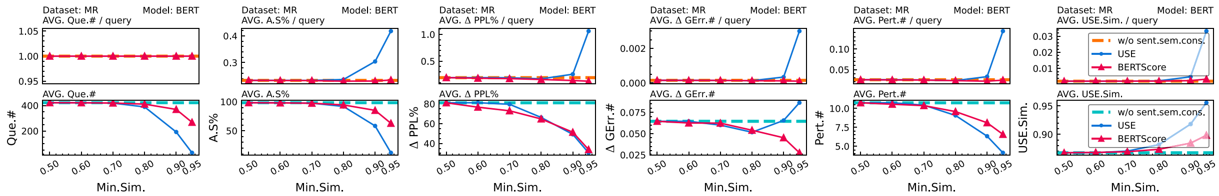

Figure 13: The impact of sentence-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with Greedy Search.

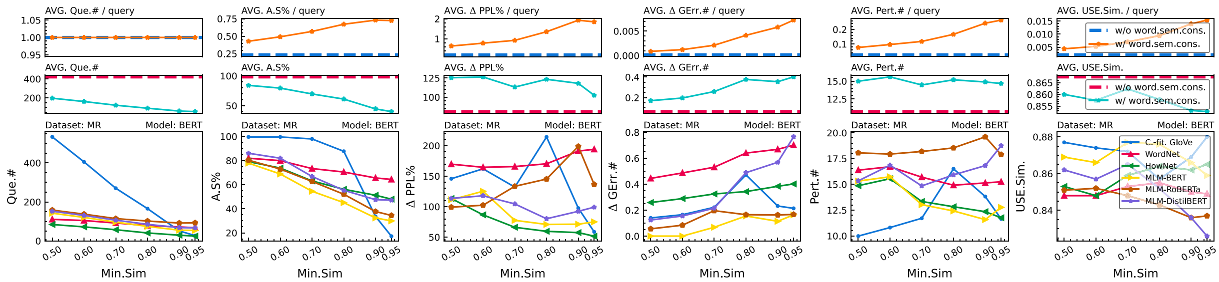

Figure 14: The impact of word-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with Greedy Search.

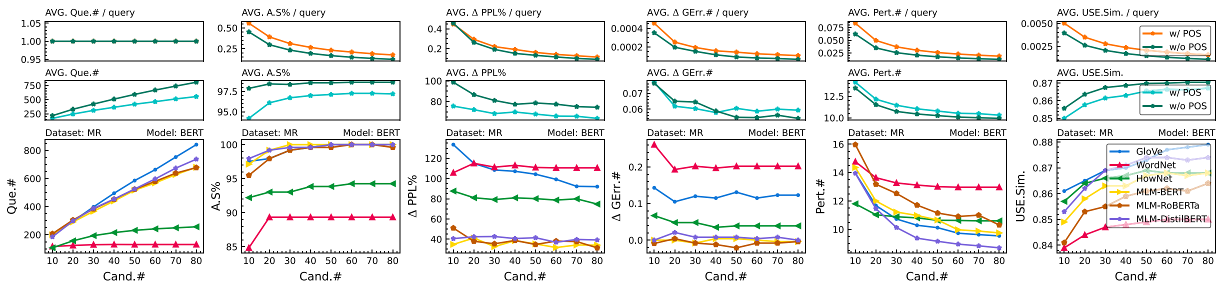

Figure 15: The impact of part-of-speech constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with Greedy Search.

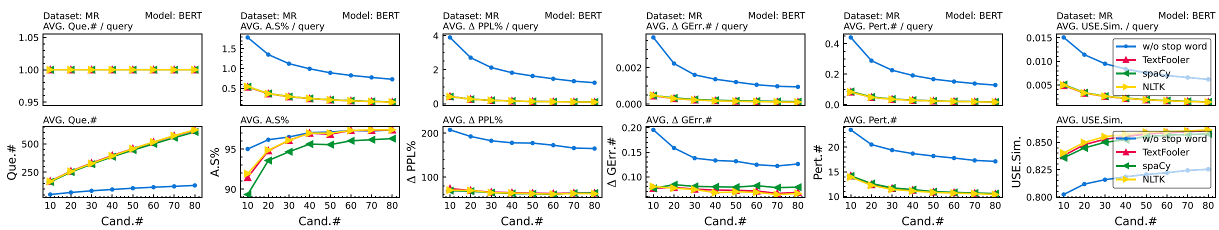

Figure 16: The impact of stop word constraint on attack efficiency, effectiveness, and imperceptibility when attacking MR against BERT with Greedy Search.

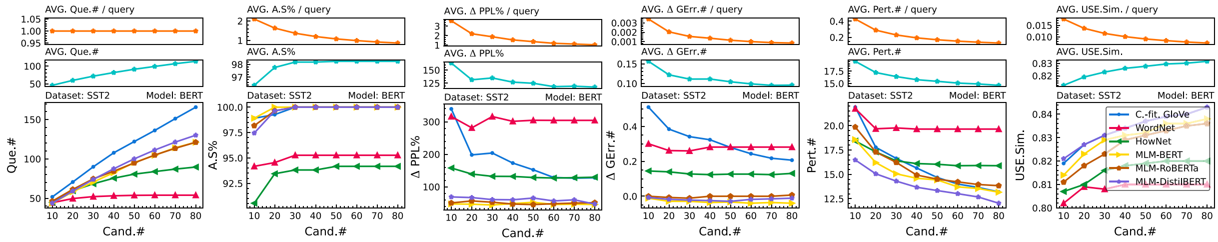

Figure 17: The impact of basic search space on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with WIR.

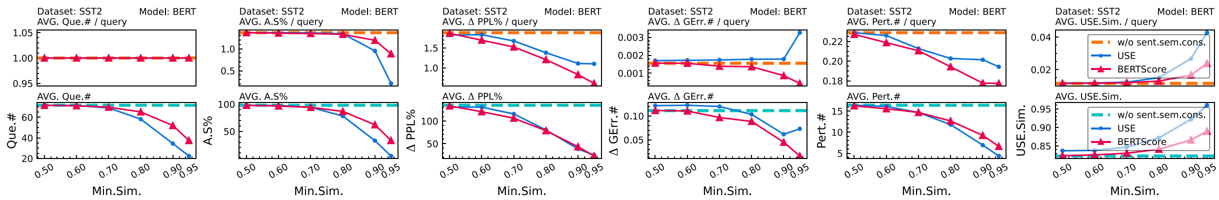

Figure 18: The impact of sentence-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with WIR.

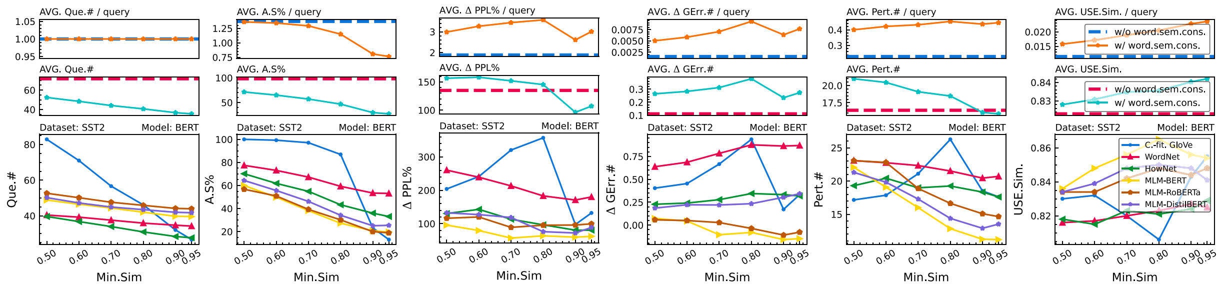

Figure 19: The impact of word-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with WIR.

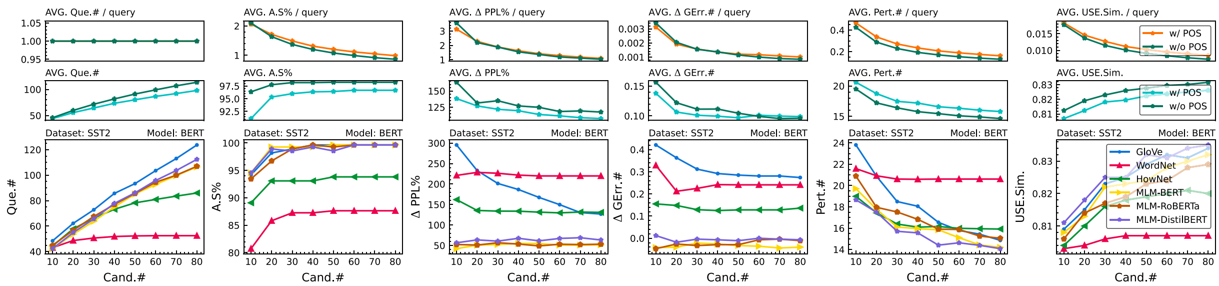

Figure 20: The impact of part-of-speech constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with WIR.

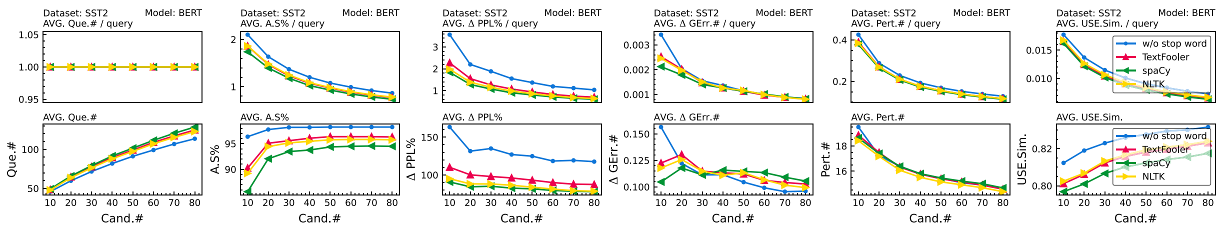

Figure 21: The impact of stop word constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with WIR.

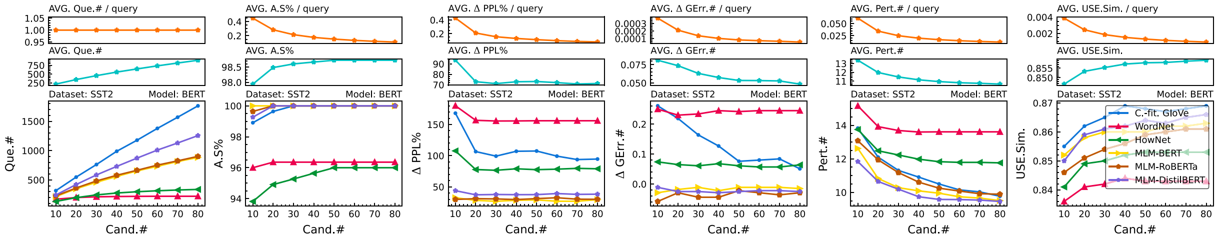

Figure 22: The impact of basic search space on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with Greedy Search.

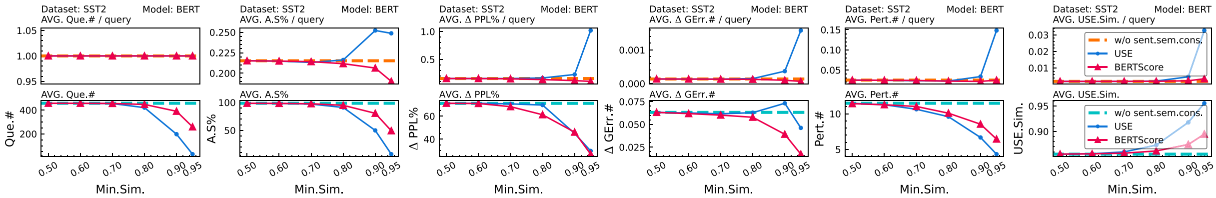

Figure 23: The impact of sentence-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with Greedy Search.

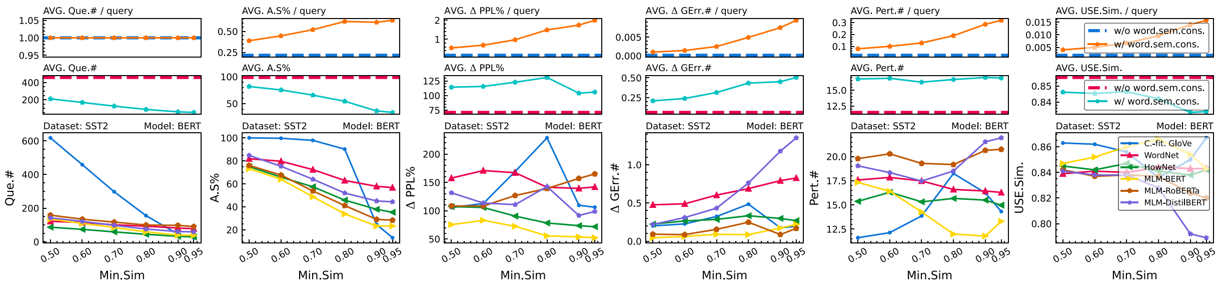

Figure 24: The impact of word-level semantic constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with Greedy Search.

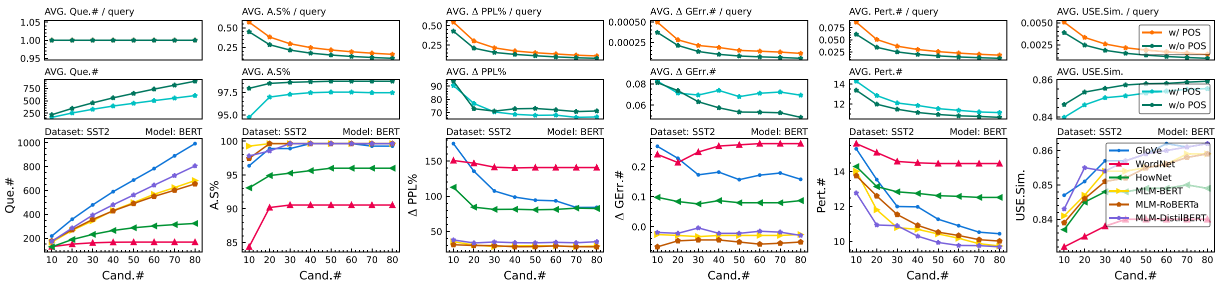

Figure 25: The impact of part-of-speech constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with Greedy Search.

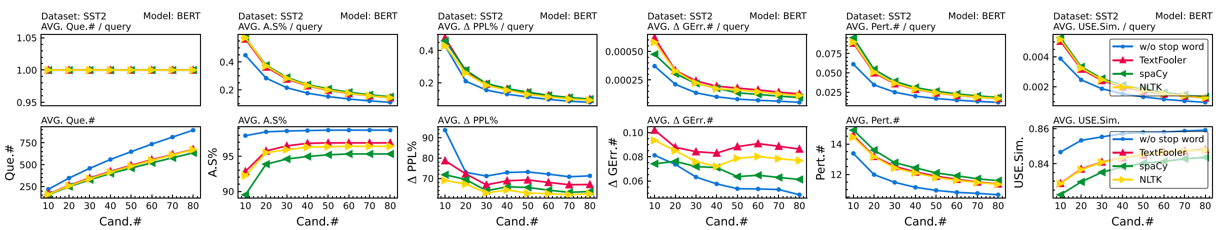

Figure 26: The impact of stop word constraint on attack efficiency, effectiveness, and imperceptibility when attacking SST2 against BERT with Greedy Search.