Adversarial Sample Generation for Aspect based Sentiment Classification

Mamta and **Asif Ekbal**
Department of Computer Science and Engineerng
Indian Institute of Technology Patna, India
{mamta_1921cs11, asif}@iitp.ac.in

Abstract

Deep learning models have been proven vulnerable towards small imperceptible perturbed input, known as adversarial samples, which are indiscernible by humans. Initial attacks in Natural Language Processing perturb characters or words in sentences using heuristics and synonyms-based strategies, resulting in grammatical incorrect or out-of-context sentences. Recent works attempt to generate contextual adversarial samples using a masked language model, capturing word relevance using leave-one-out (LOO). However, they lack the design to maintain the semantic coherency for aspect based sentiment analysis (ABSA) tasks. Moreover, they focused on resource-rich languages like English. We present an attack algorithm for the ABSA task by exploiting model explainability techniques to address these limitations. It does not require access to the training data, raw access to the model, or calibrating a new model. Our proposed method generates adversarial samples for a given aspect, maintaining more semantic coherency. In addition, it can be generalized to low-resource languages, which are at high risk due to resource scarcity. We show the effectiveness of the proposed attack using automatic and human evaluation. Our method outperforms the state-of-art methods in perturbation ratio, success rate, and semantic coherence.

1 Introduction

Sentiment analysis is a well-established area in Natural Language Processing (NLP), and finds its applications in recommendation systems, national security-sensitive applications, curating online trends, etc. (Pang et al., 2002; Bakliwal et al., 2013; Kumar et al., 2019; Mamta et al., 2020, 2022b). Considering sentiment alone can only provide high-level insights, not sufficing to analyze reviews containing multiple attributes, known as aspects. Aspect level sentiment analysis (ABSA) provides more fine-grained information by classifying the sentiment towards a specific aspect of the product (Pontiki et al., 2014).

Recently, deep learning and transformer-based approaches have obtained state-of-the-art results in numerous classification applications such as emotion, sarcasm detection, etc. including ABSA (Wang et al., 2016; Liu et al., 2019; Akhtar et al., 2016a; Mamta et al., 2022a; Sun et al., 2019; Xu et al., 2019). However, these classification algorithms can be easily fooled by maliciously crafted (adversarial) examples (Miyato et al., 2016; Li et al., 2020b). Adversarial examples expose the system vulnerabilities and also help to improve the robustness of the model. The adversarial sample generation has been extensively explored to assess the resilience of the neural model, in the field of computer vision (Szegedy et al., 2013; Kurakin et al., 2016; Chakraborty et al., 2018). It has shown improvement in the robustness and generalized capability of the model via adversarial training (Goodfellow et al., 2014). The generation of such out-of-distribution samples against NLP models is more challenging than computer vision due to the discrete nature of text. In addition, semantic consistency and grammatical accuracy of generated adversarial samples should also be preserved.

Initial attempts to attack NLP models have shown to adapt the fast gradient sign methods (FGSM) (Goodfellow et al., 2014) and Generative Adversarial Networks (GANs) based methods from computer vision, to apply perturbations on the embedding space of the text (Papernot et al., 2016; Miyato et al., 2016; Zhao et al., 2017). There is, however, a difficulty in mapping perturbed continuous embedding space to discrete token space in these methods. There are prior works which explored character-level and word-level perturbation algorithms using synonym replacement and language model based approaches (Liang et al., 2017; Ebrahimi et al., 2017; Alzantot et al., 2018; Zhang et al., 2020a). Recent studies have revealed the vul-

Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 478–492
November 20–23, 2022.

nerability of BERT-based (Bidirectional Encoder Representations from Transformers) text classification models in a black box setting using synonyms-based (Jin et al., 2020) and masked-language model (BERT-MLM) based approaches (Garg and Ramakrishnan, 2020; Li et al., 2020b; Mondal, 2021; Zhang et al., 2021).

Most existing attack methods are primarily focused on text classification, including document level sentiment classification and other question answering tasks. However, in the context of ABSA, these algorithms lack the design to maintain semantic coherency with the actual example, which is the foremost requirement of adversarial examples. For example, consider the example from SemEval laptop dataset, *Thanks for great service and shipping!* Adversarial example generated by SOTA method for aspect *service* is *Thanks for continued concern and shipping!* It is clear from the example that the overall semantics and aspect term have been changed. To maintain the semantic coherency with the actual example, aspect term should not be changed. Additionally, the presence of multi-word aspects in the sentence presents another challenge to preserve the semantics. For example, *quick and has built in virus control.* Here, the sentiment towards aspect *built in virus control* is positive. The adversarial example generated by SOTA method is *quick and has flaws in virus control*, which fails to preserve the semantic coherency and aspect term.

A recent attempt was made to attack the ABSA classifier by adding misspellings and punctuation to the actual sentences in the black-box setting (Hofer et al., 2021). These perturbations, however, can fool the classifier, but lack semantic and grammatical correctness. Moreover, these modifications may also be corrected by grammar or spelling checkers, thus increasing the likelihood of an attack failure. These approaches measure the word saliency by removing it from the sentence and calculating the drop in probability of correct class prediction (LOO). Almost all the efforts have been directed towards high-resource languages such as English. There has been no work on exposing vulnerabilities in low-resource NLP models, which are at high risk due to resource scarcity in low-resource languages. The existing ABSA attack is only applicable to the English language, as it uses language-specific rules and dictionaries. For example, the letter e can be replaced by 3 (homoglyphs) or s with 5.

To address these limitations, we propose an adversarial example generation algorithm designed for ABSA for a given aspect. Our proposed method is not dependent on the language-specific rules; hence, it can be generalized to low-resource languages with some optimizations. Our proposed algorithm applies perturbations at the word level by exploiting the model explainability technique, SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017). We extend SHAP to BERT based ABSA task by incorporating aspect information so that it can generate word saliency scores according to the given aspect. SHAP considers various combination of words to determine the word importance, unlike in earlier attempts where importance is dependent on only one word. Moreover, our proposed algorithm can generate adversarial samples for single word and multi-word aspect terms.

We summarize the contributions of our work as follows:

- We propose an algorithm to generate adversarial samples for the ABSA task, utilizing the model explainability to better rank the words for their importance.
- The proposed algorithm has higher semantic similarity and grammatical correctness than the existing attacking algorithm by introducing ABSA specific components to preserve both single word and multi-word aspect terms.
- It achieves a higher attack success rate with fewer perturbations using model explainability technique and the proposed perturbation scheme.
- The proposed algorithm uses language independent rules and can be generalized to low-resource languages with some optimizations. This is first attempt to attack a low-resource ABSA model. We demonstrate it by performing experiments on the Hindi language.

2 Related Work

Generating the adversarial examples against neural models to assess resilience of the model has been explored extensively in the field of computer vision (Nguyen et al., 2015; Chakraborty et al., 2018; Miyato et al., 2018; Guo et al., 2020). However, attempts to expose the vulnerabilities of NLP models are relatively few (Zhang et al., 2020b). Initial attempts to attack NLP models adapt the FGSM (Goodfellow et al., 2014) from computer vision.

The key idea is to apply small perturbations to the embedding space of text (Papernot et al., 2016; Miyato et al., 2016) in the direction of the gradient. GANs (Zhao et al., 2017) based method are also explored by applying perturbations in the latent space. However, these approaches often lack in semantic correctness (Jin et al., 2020; Garg and Ramakrishnan, 2020). Subsequently, several methods focused on character level and word level perturbations in white box (Liang et al., 2017; Ebrahimi et al., 2017) or black box setting (Gao et al., 2018). The generated adversarial samples are easily identifiable by human and also lacks the grammatical correctness and semantic coherency with the seed sentences. To maintain grammatical correctness and semantic consistency, Li et al. (2018) proposed to perturb important words with the top *k* words obtained from the Glove embedding vectors. Authors also explored synonyms (Ren et al., 2019) and language model (Alzantot et al., 2018; Zhang et al., 2020a) based approaches for perturbations. Morris et al. (2020) introduced TextAttack to implement adversarial attacks in Python. It is composed of four basic components: a goal function, a set of constraints, a transformation, and a search method. TextAttack implements a wide range of adversarial attacks and supports a variety of datasets and models, such as BERT and transformer-based models.

2.1 Attacks on BERT

With the huge success of BERT for text classification in NLP, few attempts have been made to expose the vulnerabilities of recently risen BERT models (Sun et al., 2020). Jin et al. (2020) is first to propose a black-box algorithm to attack the BERT model with the help of closet synonyms. But it can lead to unnatural sentences because the synonym may not fit the context of sentence. To overcome this limitation, authors (Garg and Ramakrishnan, 2020; Li et al., 2020b; Mondal, 2021; Li et al., 2020a) proposed to use masked language model (BERT or Roberta) for replacements or insertions. The importance of each word is identified, as done in the previous black-box approaches (LOO). Relevant to our current work is the work done in (Hofer et al., 2021) which is first to attack aspect based sentiment classification model where character-level transformation are applied to generate adversaries against BERT. The importance of each word is calculated as done in (Garg and Ramakrishnan, 2020; Li et al., 2020b). However, the generated adversarial examples lack the semantic consistency due to mis-spellings.

Literature survey reveals that most of the efforts of attacking NLP models are for text classification, including document/sentence level sentiment classification tasks which lack the design to maintain semantic coherency with the seed sentence for ABSA task. The existing attack on ABSA uses language dependent rules and dictionaries, which can not be adapted to the Hindi language. In our work, we propose an attacking algorithm for aspect-based sentiment classification to address these limitations that can be generalized to low-resource languages. It uses SHAP, which is language independent component for word importance ranking, BERT-MLM, which can be applied to several languages, replace and insert operations which are language independent rules.

3 Threat Model

Our target model is a BERT based ABSA classifier. The adversary aims to generate adversarial samples against the target model due to their huge success in many NLP tasks, including ABSA (Liu et al., 2019; Xu et al., 2019; Sun et al., 2019).

Adversary's knowledge: The adversary has the black-box access of the target model. It queries the target model to get the prediction vector. The adversary does not have access to the data used to train the target model, rather it owns some test samples of similar distribution, which are used for the adversarial sample generation against the target model.

Adversary's goal: Given an input sentence S, consisting of n tokens $w_1, w_2, w_3, \ldots, w_n$ with m aspects asp = $asp_1, \ldots, asp_m$, where asp_i = $w_{s_i}, \ldots, w_{s_i+l_i}$ (contiguous subsequence of words from S), with ground truth sentiment label y_{asp_i} towards aspect asp_i, and a target model $M(S, asp_i) = y_{asp_i}$. Here l_i is the number of words in the asp_i (m and $l \geq 1$) and s_i is the starting index of asp_i. The goal of the adversary is to perform an un-targeted attack, i.e., find adversarial sample S_{adv} for aspect asp_i, causes M to perform misclassification, i.e., $M(S_{adv}, asp_i) != y_{asp_i}$. At the same time, S_{adv} should satisfy the following properties: i). S_{adv} should be semantically similar to S. This is achieved by $sim(S, S_{adv}) > \epsilon$, where sim is cosine similarity and ϵ is the threshold value. ii). S_{adv} should be grammatically correct.

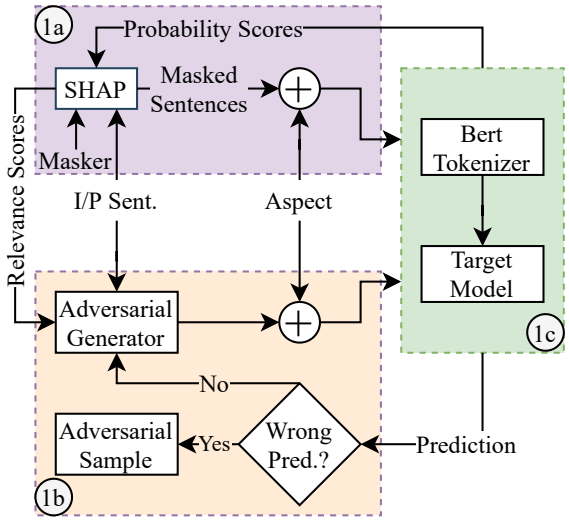

Figure 1: Proposed attack

iii). $HumanPred(S_{adv}, asp_i = y_{asp_i})$, where $HumanPred$ is classification by human. iv). S_{adv} should preserve the aspect asp_i for which the opinion is expressed. [1]

4 Methodology

We present model interpretable attack algorithm to generate high quality adversarial examples to assess the robustness of ABSA model by applying perturbations at the word-level. The detailed architecture of our proposed approach is depicted in Figure 1. There are 3 main components, *viz.* word saliency generator (1a), adversarial generator (1b) and the target model (1c). First, the saliency generator uses the model interpretation technique (SHAP) (Lundberg and Lee, 2017) to score the relevance of tokens in the input sentence for a given aspect. Relevance scores are used by the adversarial generator to generate the adversarial samples by applying perturbations at the word level. It runs iteratively until the generated adversarial sample is able to fool the target model.

4.1 Word saliency generator

Our first step is to find the contribution of each word for the final prediction. In general, word importance is computed as the difference between a prediction for a given sentence S (with n words) and the expected prediction when the word w_j is not present in S and replaced by $[MASK]$. It is described in Equation 1.

$$\phi_j(S) = M(w_1, \ldots, w_n) - E[M(w_1, \ldots, w_{j-1}, [MASK], w_{j+1} \ldots, w_n)] \quad (1)$$

We use Shapely algorithm, inspired by coalitional game theory, to determine the relevance of each word in a given sentence, against the target model. Shapley calculates the relevance score for each word based on possible coalitions for a particular prediction. Equation 2 explains the computation using a value function, which calculates the feature importance over the difference in prediction with or without w_j, over all combinations. It is the shapley value of a feature calculated as the contribution to the payout, weighted and summed over all possible feature value combinations.

$$\phi_j(\mathrm{w}) = \sum_{Q \subseteq S \setminus j} \frac{|Q|!(|S|-|Q|-1)!}{|S|!} \left(\mathrm{v}_{(Q \cup \{j\})}(w) - \mathrm{v}_Q(\mathrm{w})\right) \quad (2)$$

$v_Q(w)$ (value function) is the payout function for coalitions of players (feature values), which denotes the influence of a subset of feature values. It generalizes (Equation 1), in the following form

$$\mathrm{v}_Q(w) = \mathbb{E}\left[M \mid W_i = w_i, \forall i \in Q\right] - \mathbb{E}[M] \quad (3)$$

Where M provides the prediction over the set of features provided, S is the complete set of features, $Q \in S$ is a subset of features, and $|\cdot|$ is the size of feature set (Štrumbelj and Kononenko, 2014).

To adapt SHAP for the BERT based ABSA task, we implement a custom function for pre-processing the input data to obtain the predictions from the target model. In addition, we create an explicit word masker to tokenize the sentence into sentence fragments consisting of words, which serves as a basis for word masking in SHAP (here mask refers to hiding a particular word from the sentence). The input sentence along with the designed masker is passed to SHAP, generating various masked combinations of the sentence. These masked sentence fragments are concatenated with the input aspect with the help of CLS and SEP tokens ($[CLS]$ sentence $[SEP]$ aspect $[SEP]$) and further passed to the BERT tokenizer. Concatenation of aspect to masked sentence helps for better prediction scores, and in turn helps Shapley to focus on words which are relevant for sentiment classification of the given aspect. BERT tokenizer converts the words to subwords and generates input, segment, and mask embeddings for

[1] Suppose we have three aspect terms ($asp1$, $asp2$, asp) in the sentence ($sent$), then we have defined tuples like ($sent$, $asp1$), ($sent$, $asp2$), ($sent$, $asp3$). Our framework attempts to generate 3 adversarial examples (one for each aspect).

each subword unit and generate final representation by performing summation of all the three embeddings (Devlin et al., 2018). Finally, this combined representation of these vectors for each masked version is passed to the target model to obtain the output probabilities, which are further returned to SHAP to obtain the relevance of each word for the final prediction. This whole process is illustrated in Figure 1 (1a).

Algorithm 1 Adversarial Sample Generation

Input: Text Sentence S = $w_1, w_2, ..., w_n$, label y, importance_scores I, aspect asp, threshold ϵ, Prb$_{actual}$
Output: Adversarial Text Sentence S_{adv}

1: Initialization: $S_{adv} \leftarrow S$
2: Create $W_{context}$ and W_{aspect}
3: Remove stop words from I
4: **for each** word $w_k \in$ descending order of I **do**
5: **if** $w_k \in W_{context}$ **then**
6: replace w_k in S by $[MASK]$
7: **else if** $w_k \in W_{aspect}$ **then**
8: **if** $len(W_{aspect}) == 1$ **then**
9: insert $[MASK]$ at start/end of w_k
10: **else if** $len(W_{aspect}) > 1$ **then**
11: insert $[MASK]$ at start/end of multi-word aspect asp
12: **end if**
13: **end if**
14: find CANDIDATES for $[MASK]$ using BERT-MLM
15: Success={}; ProbRed= {}
16: **for** $c_j \in$ CANDIDATES **do**
17: $S' \leftarrow$ Replace w_k with c_j in S_{adv}
18: $y_j \leftarrow M(S')$
19: $Prb_j \leftarrow M_{y_j}(S')$
20: **if** ((cos$(S', S) > \epsilon$) and $(y_j \neq y)$) **then**
21: Success[c_j] = cos(S', S)
22: **else if** ((cos$(S', S) > \epsilon$) and $(Prb_j < Prb_{actual})$) **then**
23: ProbRed[c_j] = $Prb_{actual} - Prb_j$
24: **end if**
25: **end for**
26: **if** $len(Success) > 0$ **then**
27: $S_{adv} \leftarrow$ replace $[MASK]$ in S with candidate word having the highest cos value
28: **else if** $len(ProbRed) > 0$ **then**
29: $S_{adv} \leftarrow$ replace $[MASK]$ in S with word which generates the lowest probability for y
30: **end if**
31: **end for**

4.2 Adversarial generator

After finding relevance scores, we iteratively perturb the words in descending order of their relevance scores until the attack is successful. We create two different sets of words for aspect W_{aspect} and contextual words $W_{context}$. Let's consider an example sentence, "boot time is super fast, around anywhere from 35 seconds to 1 minute", with aspect boot time. Here W_{aspect} = (boot, time) and $W_{context}$ = (is, super, fast, around, anywhere, from, 35, seconds, to, 1, minute). At a given position in the sentence, we apply two kinds of perturbations, depending on the word type. The detailed process of generation is illustrated in Algorithm 1.

Contextual words perturbations: We perform a replace operation to perturb the contextual words in order to generate semantically coherent and grammatically correct sentences (lines 4-6). For each sentence, we opt not to perturb the stop words as they may affect the grammatical correctness of the sentence. Let w_k be the word to be perturbed in a sentence S. We apply a mask operation at k^{th} position so that later we can replace it with another word that satisfies the properties of the adversarial example. Mask operation at w_k is applied as follows: $S = w_1, ..., w_{k-1}, [MASK], w_{k+1}, .., w_n$.

Aspect terms: If the word w to be perturbed is an aspect term, then we do not perform a replace operation; instead, we change the context around the aspect term by inserting a token in front or end of it. We do so to preserve aspect terms of the sentence, which is essential to preserve semantics of the sentence (lines 7-13). Let, w_k is the word for perturbation. Then, there are two cases,
i). w_k is complete aspect term asp, then mask operation is applied as follows:
$S = w_1, ..., w_{k-1}, [MASK], w_k, w_{k+1}, .., w_n$.
ii). If the aspect term is the multi-word aspect, and w_k is one of its words, then we insert $[MASK]$ in front of the first word of the aspect term to preserve the complete aspect term. Let us say asp consists of w_{k-1} and w_k words, then mask operation is applied as follows:
$S = w_1, ..., [MASK], w_{k-1}, w_k, ..., w_n$.

After applying the mask operation, this masked sentence is fed into BERT-MLM following Garg and Ramakrishnan (2020); Li et al. (2020b) to generate top j CANDIDATES for the masked position. It ensures that the generated sentence preserves fluency and is grammatically correct. Furthermore, BERT-MLM considers the whole context when predicting the masked word; hence, the predicted word is context-aware. While replacing the contextual word, we omit the candidate words with different Part-of-speech (POS) tag than that of w_k to ensure grammatical correctness.

Semantic preservation: BERT-MLM generates contextual candidate words, but does not assure semantic similarity with the actual sentence. We use the cosine similarity metric to measure the similarity between adversarial and actual sentence (for

each candidate) (Morris et al., 2020). We use Sentence Transformer to generate sentence representations (Reimers and Gurevych, 2019a, 2020). All the candidate words having cosine similarity above the threshold ϵ and are able to mislead the target model are added to the **Success**, and candidates which reduces the probability of actual class are added to **ProbRed** (lines 15-25).

4.3 Final adversarial sample

The candidate word that can successfully mislead the ABSA classifier with the highest semantic similarity score with the seed sentence or generate the lowest probability of actual class is chosen for final adversary generation (lines 26-30). The steps are repeated until the adversarial example can fool the target model.

5 Experimental Setup

We use BERT-base and BERT-base-multilingual as target models for English and Hindi, respectively. For adversarial example generation, we set the value of top candidates j to 50 and the threshold value ϵ to 0.8. To identify the POS tags, we use stanfordnlp library [2] for English and Hindi.

Datasets: To evaluate our proposed attack, we use the following ABSA datasets:

- **SemEval-14 laptop dataset**: This dataset is released as part of a shared task on ABSA (Pontiki et al., 2014) and consist of reviews from the laptop domain.
- **ABSA Hindi dataset:** Hindi ABSA dataset was released by Akhtar et al. (2016a) containing reviews from 12 domains. We use the 70%, 20%, and 10% split for train, test, and validation as done in Akhtar et al. (2016b).

The datasets are annotated with four classes, *viz.*, positive, negative, neutral, and conflict. Both the datasets contain fewer instances of conflict class. So, we focus on 3 classes by excluding the conflict class. (more details are present in A.1)

Baselines: We define the following baselines:

- **Baseline 1 (Li et al., 2020b)**: Baseline 1, BERT-attack, uses BERT-MLM to perturb the words using replace operation and used LOO for word importance ranking.
- **Baseline 2 (Garg and Ramakrishnan, 2020)**: Baseline 2, BAE, uses BERT-MLM to perturb the words using replace and insert operations, where word importance is calculated using LOO method. It performs insert operation after the replace operation.
- **Baseline 3 (Li et al., 2018):** Baseline 3, Textbugger, uses character-level and word-level perturbations. It searches for nearest neighbors in the embedding space using Glove model. Like baseline 1 and 2, it does not make any difference in the type of words (context words or aspect terms).
- **Baseline 4:** We extend the baseline 1 to Hindi ABSA setting. We use mBERT-MLM for word perturbation against the target model.
- **Baseline 5:** A state-of-the-art model proposed by Hofer et al. (2021) to attack an English ABSA model. It uses leetspeak (LEET), common mis-spellings (TYPO), or misplaced commas (PUNCT) to generate the adversarial examples. We implement all the three methods of attack for English. Their proposed LEET and TYPO are only applicable to English languages. However, the PUNCT attack can be applied to the Hindi language.
- **Baseline 6:** Baseline 6 extends baseline 2 to Hindi ABSA setting.

Baseline 1, 2, and 3 are originally proposed for text classification tasks. We extend them to an ABSA task by passing a pair of input containing sentence and aspect separated by $[SEP]$ token. We implement baseline 2 and 3 using TextAttack framework.

Evaluation metrics: To measure the effectiveness of the attack, we calculate (i). *Before-attack-accuracy and After-attack-accuracy* (*BA* and *AA*): The Before-attack-accuracy is estimated on the test set, and After-attack-accuracy is calculated on the adversarial test set; (ii). *Attack success rate (SR):* the percentage of adversarial examples that can successfully attack the target model; (iii). *Perturbation ratio (PR):* the ratio of words perturbed in the sentence to the total number of words in the sentence ; and (iv). *Semantic similarity (SS):* this is computed between the adversarial and actual sentence using the cosine similarity metric, which makes use of Sentence transformers (Reimers and Gurevych, 2019b) to generate sentence representations. In

[2] https://stanfordnlp.github.io/stanfordnlp/

Method	BA	AA	SR	PR	SS	ATCR
English						
b1	76.82	26.67	65.57	15.4	0.89	20.42
b2		22.06	70.68	14.51	0.83	25.53
b3		22.22	70.46	16.8	0.85	29.32
LEET		57	25	-	0.70	9.41
TYPO		59	22.5	-	0.61	10.30
PUNCT		69	10	-	**0.97**	-
Ours		**11.02**	**87.09**	**13**	0.92	**0**
Hindi						
b4	74.71	32.53	57.74	17	0.80	27.38
PUNCT		70.4	8.5		**0.97**	-
b5		28.00	62.05	16.2	0.83	-
Ours		**20.4**	**73.51**	**15**	0.89	**0**

Table 1: Experimental results. Here, b1: baseline 1, b2: baseline 2, b3: baseline 3, b4: baseline 4, and baseline 5 (LEET, TYPO, PUNCT), b5: baseline 5

Language	Type	GC	SP	HP
English	Baseline 1	4.1	3.7	71%
	Our	4.3	4.2	80%

Table 2: Human evaluation

Language	Ranking	AA	PR
English	random	33.90	21
	LOO	26.67	14.9
	SHAP	11.02	13
Hindi	random	48.00	24
	LOO	29.81	16.5
	SHAP	20.40	15

Table 3: Ablation experiment results

Setup	BA	AA	PR
English (10% adv)	76.81	20.63	14
(50% adv)	76.76	21.11	15
(100% data)	75.71	37.14	16
Hindi (10% data)	75.05	32.65	16.2
(50% data)	74.63	34.35	18.5
(100% data)	74.94	39.22	20

Table 4: Adversarial training results

addition to this, we also compute (v). Aspect terms change ratio (*ATCR*), which is defined as the ratio of the number of sentences where the aspect terms have been changed to the total number of sentences. We define this metric to illustrate the need for special design for ABSA adversarial generation.

6 Experimental Results and Analysis

Experimental results for both the languages for all the metrics are summarized in Table 1. We observe that our proposed attack outperforms all the baselines in terms of attack success rate, perturbation ratio, and semantic similarity. For English language, success rate of our model is higher than the other baselines by 21.52-77.09%. Our model achieves an average semantic similarity of 0.92 with actual sentences, higher than all the baselines except baseline 3-PUNCT. The semantic similarity of baseline 3-PUNCT is higher because it adds only comma after an important word. However, its success rate is only 10%, least among other attacks. ATCR ratio is highest for baseline 3. LEET and TYPO has less ATCR ratio, but their attack success rate (SR) is very less. Our proposed method is able to preserve the aspect terms, so ATCR ratio is 0%. In addition to this, our proposed method requires a few perturbations to execute a successful attack.

The same phenomenon is observed for Hindi. Our proposed attack method achieves 15.77-65.01% higher success rate than the other baselines. Notably, our method also outperforms baseline 1 and baseline 3-PUNCT in attack success rate. It requires fewer average modifications to input text compared to baseline 2. It needs to perturb only an average of 15% of the words to perform a successful attack. However, baseline 2 perturbs 17% of the words in input space.

Human evaluation: We also perform human evaluation to see the effectiveness of our proposed attack. We randomly select 100 samples from English language for baseline 1 (the strongest baseline) and our proposed attack. A total of 3 linguists (annotators) having post graduate level experience, with good knowledge of English and Hindi from India were employed for annotations. They were advised to mark the (i). grammatical correctness score (GC) on the scale of 1-5, (ii). sentiment class towards given aspect to evaluate the human prediction consistency (HP), and (iii). semantic preservation score (SP) on the scale of 1-5 to see whether they retain the meaning of actual sentences or not. For HP metric, annotators were asked to write the overall polarity of the adversarial sample in 3 categories *viz.* neutral, negative, and positive. They were provided with gold labeled samples to gain deep understanding of sentiment labels before actual annotations. Further, they were also advised to refrain from being biased towards either a specific demographic area, religion, or ethnicity while annotating the samples. Results are shown in Table

	Sentence	Aspect	Model Output
Actual	the apple engineers have not yet discovered the delete key	delete key	negative
Baseline 1	the **security authorities** have not yet **announced** the delete key		neutral
Ours	the apple engineers have not **until** discovered the **standard** delete key		neutral
Actual	air has higher resolution but the fonts are small	fonts	negative
Baseline 1	air has **more** resolution but the **applications** are **tiny**		negative
Ours	air has higher resolution but the **available** fonts are **tiny**		positive
Actual	i was given a demonstration of windows 8	windows 8	neutral
Baseline 1	i was given a demonstration of windows 8		neutral
Ours	i was given a demonstration of **the** windows 8		positive

Table 5: Adversarial samples generated by different methods

2. We observe that our proposed method obtains higher GC, HP, and SP scores than the baseline 1, illustrating the fact that our generated adversarial samples are more semantically coherent and grammatically correct.

Ablation study: We perform an ablation experiment to observe the effectiveness of the saliency generator in our proposed method and observe the change in after-attack accuracy and perturbation rate when it is removed. First, we rank the words in random order and then, use LOO method for word ranking. Table 3 shows the results for both the languages. We observe that there is an increment in after-attack accuracy and perturbation ratio for English as well as Hindi language (more ablation experiments are present in A.2).

Adversarial training: We observe the model robustness with adversarial training as a defence mechanism, using our proposed algorithm to generate adversarial samples for training data, followed by fine-tuning the target model on the combined original training data and adversarial training data. We design three strategies here: (i). actual training data + randomly sampled 10% adversarial data, (ii). actual training data + randomly sampled 50% adversarial data, (iii). actual training data + complete adversarial data. After fine-tuning the target model, we attack the target model with the proposed algorithm. Results for both the languages are shown in Table 4. We observe that the after attack accuracy and perturbation ratios are increased after adversarial training. This illustrates that the model becomes more robust against adversarial attacks as more adversarial examples are added to the training set. It can be also observed that the adversarial training reduces the actual test accuracy of the target model by a small percent, i.e., 1% in case of the English language, which is in line with Jia et al. (2019). However, in the case of Hindi, almost no drop in accuracy is observed; instead, the accuracy is increased by 0.34%. This demonstrates that our proposed algorithm can be used to improve the robustness of ABSA models.

Detailed analysis: For detailed qualitative analysis, we manually analyze the adversarial samples generated by baseline 1 and our proposed attack. As our saliency generator finds relevance of every word by considering various combinations, so it can decide better order of word perturbations. Examples 1 and 2 illustrate the importance of our saliency generator. In example 1, both the attack strategies successfully fools the classifier. But our proposed attack can generate a semantically similar example with fewer perturbations. Here, baseline 1 perturbs 3 words. However, our proposed attack perturbs one contextual word and alters the context around multi-word aspect term to perform a successful attack. This illustrates the improvement using SHAP scores compared to the method used in Garg and Ramakrishnan (2020); Li et al. (2020b). In example 2, baseline 1 performs an unsuccessful attack even by perturbing 3 words, including aspect word. On the top, changing the aspect *fonts* to *application* alters the semantics of the sentence. Our proposed method preserves the aspect information and requires only 2 modifications to execute a successful attack. In example 3, baseline 1 cannot find appropriate replacements to fool the target model. However, our method is able to fool the target model by inserting word *the* in front of the multi-word aspect term (more detailed qualitative analysis for all the baselines is present in section A.3).

7 Conclusion

In this paper, we have presented an effective algorithm to generate adversarial examples for assessing the resilience of the BERT based aspect based sentiment classification model. To generate adversarial examples, we exploit the model's explainability to identify the word saliency. We propose replace operation for contextual words and insert operation for aspect term to generate more semantically similar sentences. We have evaluated our proposed algorithm on two benchmark datasets, English and Hindi. Extensive experiments and human evaluation show that our proposed algorithm outperforms the state-of-art attack methods in success rate, perturbation ratio, and semantic preservation.

In our current work, we have evaluated the robustness of the sentiment classification task only. In future, we would extend this work to evaluate the robustness of both aspect term extraction and sentiment classification.

Acknowledgements

Asif Ekbal acknowledges the Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya Ph.D. scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

References

Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016a. Aspect based sentiment analysis in hindi: resource creation and evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2703–2709.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016b. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493.

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*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

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 (SPW)*, pages 50–56. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Pengxin Guo, Yuancheng Xu, Baijiong Lin, and Yu Zhang. 2020. Multi-task adversarial attack. *arXiv preprint arXiv:2011.09824*.

Nora Hofer, Pascal Schöttle, Alexander Rietzler, and Sebastian Stabinger. 2021. Adversarial examples against a bert absa model–fooling bert with l33t, misspellign, and punctuation. In *The 16th International Conference on Availability, Reliability and Security*, pages 1–6.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Sudhanshu Kumar, Mahendra Yadava, and Partha Pratim Roy. 2019. Fusion of eeg response and sentiment analysis of products review to predict customer satisfaction. *Information Fusion*, 52:41–52.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. 2016. Adversarial examples in the physical world.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2020a. Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Mamta Mamta, Asif Ekbal, and Pushpak Bhattacharyya. 2022a. Exploring multi-lingual, multi-task, and adversarial learning for low-resource sentiment analysis. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(5).

Mamta Mamta, Asif Ekbal, Pushpak Bhattacharyya, Tista Saha, Alka Kumar, and Shikha Srivastava. 2022b. HindiMD: A multi-domain corpora for low-resource sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7061–7070, Marseille, France. European Language Resources Association.

Mamta Mamta, Asif Ekbal, Pushpak Bhattacharyya, Shikha Srivastava, Alka Kumar, and Tista Saha. 2020. Multi-domain tweet corpora for sentiment analysis: Resource creation and evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5046–5054, Marseille, France. European Language Resources Association.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Ishani Mondal. 2021. Bbaeg: Towards bert-based biomedical adversarial example generation for text classification. *arXiv preprint arXiv:2104.01782*.

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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 annual meeting of the association for computational linguistics*, pages 1085–1097.

Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2020a. Generating fluent adversarial examples for natural languages. *arXiv preprint arXiv:2007.06174*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020b. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.

A Appendix

A.1 Experimental setup

To implement our model, we use the Python-based library Pytorch [3] and Hugging face implementation of BERT (Wolf et al., 2019). Target model for English (BERT-base) uses 12 layers of transformers block with a hidden size of 768 and number of self-attention heads as 12. It has 110M trainable parameters. Multilingual BERT is pre-trained on 104 languages including Hindi. Input consists of two segments, first contains the sentence and second part consists of aspect term, both are separated by $[SEP]$ tokens. We use the BertAdam optimizer to optimize the network weights based on the categorical cross entropy. The hyper-parameters of BERT are also fine-tuned for both the languages on the respective task datasets. We split 15% of the training data into validation set, used for fine-tuning the hyper-parameters. We show the dataset statistics in Table 7. We use the grid search to find the best set of hyper-parameters. All the hyper-parameters, along with the best set, are shown in Table 6. All the computations are performed on the Nvidia929GeForce GTX 1080 GPU with 12 GB memory.

Hyper-parameter	Values	Best
Learning rate	2e-5,3e-5,5e-5	3e-5
Batch size	8,16,32	16
Epochs	2,3,5	3

Table 6: Hyper-parameter values

A.2 More ablation studies

To investigate the performance of two perturbations, when applied individually, we carried out two ablation experiments where we (i). Perform only replace operation on contextual words and aspect words are left unchanged; (ii). Perform only insert operation for aspect words, and contextual words are not modified. Results are shown in Table 8. We observe that the after attack accuracies are changed to 26.23% and 51.47% only for replace operation and insert operation, respectively. The attack success rate for insert operation is very low as it only changes the context of aspect terms. By combining these two types of operations, our proposed method achieves a higher success rate.

[3] https://pytorch.org/

Dataset	Type	Samples	aspects	Pos	Neg	Neu	Con
SemEval	Train	3045	2458	987	866	460	45
	Test	800	654	341	128	169	16
Hindi	Total	5417	4509	1986	569	1914	40

Table 7: Data Statistics for English and Hindi datasets. pos: positive, neg: negative, neu: neutral, con: conflict

Language	Operation	AA	SR
English	Insert	51.47	40.42
	Replace	26.23	70.333
	Both	**11.02**	**87.09**

Table 8: Ablation experiments: Insert and replace operation

We also investigate the importance of POS constraints by removing them (for replace operation). We observe that removing the POS constraint increases the success rate to 89.93% and lowers the after attack accuracy to 10.63%. We manually analyze a few adversarial samples, which reveals that the removal of POS constraints affects the grammatical correctness of the sentence. So, the POS constraint step is to assure grammatical correctness.

Further, we replace the Stanford POS-tagger with NLTK POS-tagger [4] to observe the effect on the after attack accuracy and success rate of the model. NLTK POS-tagger yields the after attack accuracy of 17.6% with attack success rate of 77.25%.

A.2.1 Effect of semantic similarity constraint

To maintain semantic consistency with the original sentence, we preserve aspect terms and apply textual similarity constraint $(sim(S, S_{adv}) > \epsilon)$. We ablate the textual similarity constraint $(sim(S, S_{adv}) > \epsilon)$ to measure its effectiveness. Instead, we randomly choose a word from the set of candidates that can either decrease the classification probability or fool the classifier. We observe that removing semantic constraints decreases textual similarity to 0.82 (from 0.92) and increases the attack success rate to 89.22% (from 87.09%). It can also be observed that attacking a model without semantic similarity constraint (threshold constraint) becomes easier. However, the decline in the average semantic similarity between actual sentences and corresponding adversarial sentences indicates that there is a deterioration in the quality of generated examples. Examples shown in Table 9 demonstrate this fact. Although the generated adversarial sample can fool the classifier (model output changed to negative), it does not preserve the actual semantics and the original label (changed to negative) of the actual sentence.

A.2.2 Comparison of different similarity functions

We experimented with different similarity functions to observe the affect on attack accuracy and success rate. We measured semantic similarity with Jaccard similarity measure and Euclidean distance. For the Jaccard similarity measure and Euclidean distance measure, we set the threshold to 0.8 and 0.8 (1 - Euclidean distance), respectively. Results for both measures are shown in Table 10. Jaccard metric reduces the attack success rate to 58%. Similarly, Euclidean distance also reduces the attack success rate 31.35%.

A.2.3 Effect of threshold values

To study the effect of threshold values on attack success rate and semantic similarity, we perform various experiments with different values of ϵ. Results are shown in Table 11. We observe a trade-off between semantic similarity and attack success rate. With the increase in ϵ, semantic similarity increases, but the attack success rate decreases. The threshold value of 0.80 yields the attack success rate of 87.09% and semantic similarity of 0.92. However, the threshold value of 0.95 reduces the attack success rate to 56.35% and increases the semantic similarity to 0.973.

A.3 More qualitative Analysis

We further analyze the outputs of all the baselines for detailed analysis. Examples are shown in Tables 12 and 13. As indicated in example 1, baseline 1 violates semantic consistency (property 1), grammatical correctness due to incorrect article usage (property 2), and human predictions (property 4). Baseline 2 violates properties 1 and 2. Baseline 3 and LEET introduce misspellings, which also lacks semantic consistency. LEET replaces the word **excellent** with **3xc311nt**, which has no semantics. However, our proposed approach satisfies all the properties to execute a successful attack.

For example 2, the aspect term is **heat output**. BERT-attack (baseline 1) and BAE (baseline 2) require two perturbations and performed replace operation to execute a successful attack. However, the semantics of the original sentence are altered (property 1). Similarly, baselines 3 and LEET also lack

[4] https://www.nltk.org/

	Sentence	Aspect	Model Output	Human Pred.
Actual	the nicest part is the low heat output and ultra quiet operation	heat output	pos	pos
Adversarial (without constraint)	the **lowest** part is the low heat output and ultra quiet operation		neg	neg
Adversarial (with constraint)	the **best** part is the low heat output and ultra quiet operation		neg	neg

Table 9: Qualitative analysis of adversarial attacks with and without the semantic similarity constraint (threshold on cosine similarity). Here, pos: positive and neg: negative.

Language	Measure	AA	SR
English	Jaccard	32.53	58
	Euclidean distance	53.17	31.35
	Cosine	11.02	87.09

Table 10: Results on different similarity measures

Language	ϵ	AA	SR	SS
English	0.80	11.02	87.09	0.92
	0.82	16.98	82.60	0.925
	0.85	20.47	77.32	0.936
	0.87	23.65	74.84	0.944
	0.90	26.98	69.02	0.954
	0.92	31.75	63.40	0.964
	0.95	38.25	56.35	0.973

Table 11: Effect of threshold values

semantic consistency. Our proposed method requires only one perturbation and generates a more semantically coherent sentence than other baselines.

Similarly, baselines 1, 2, and 3 have altered the aspect term **compact computing** in example 3, affecting the semantic consistency (property 1). LEET and baseline 3 are also not able to maintain semantic consistency. However, our proposed approach preserves the aspect term and requires only 1 perturbation (insert operation) to execute a successful attack.

Example 4 also indicates that baseline 1 and baseline 2 cannot preserve property 1 (semantic consistency) and property 3 (human label prediction). However, the adversarial sentences generated by our proposed method satisfy all the properties of adversarial examples.

This detailed qualitative analysis illustrates that our proposed approach generates more grammatical and semantically coherent sentences with fewer perturbations than other baselines.

	Sentence	Aspect	Model Output	Human Pred.
Actual	they don't just look good; they deliver excellent performance	performance	pos	pos
baseline 1	they don't just look good; they deliver **an** performance		neg	neu
baseline 2	they don't just look good; they deliver **bad** performance		neg	neg
baseline 3	they don't just look good; they deliver **e×celt** performance		neg	neu
LEET	they don't just look good; they deliver **3xc3113nt** performance		neg	neu
PUNCT	they don't just look good; **,**they deliver excellent performance		neg	pos
Ours	they don't just look good; they deliver **good** performance		neu	pos
Actual	the nicest part is the low heat output and ultra quiet operation	heat output	pos	pos
baseline 1	the nicest part is the low heat output and **over** quiet **division**		neg	pos
baseline 2	the nicest part is the low heat output and ultra **weak reduced**		neg	pos
baseline 3	the nicest part is the low heat output and ultra **quit operaton**		neg	pos
LEET	the **nic35t** part is the low heat output and ultra quiet operation		neg	pos
PUNCT	the nicest **,** part is the low heat output and ultra quiet operation		neg	pos
ours	the **best** part is the low heat output and ultra quiet operation		neg	pos
Actual	the mac mini is probably the simplest example of compact computing out there	compact computing	pos	pos
baseline 1	the mac mini is probably the simplest **member** of **convex** computing out there		neg	aspect changed
baseline 2	the mac mini is probably the simplest example of **hard** computing out there		neg	aspect changed
baseline 3	the mac mini is **probabl** the **simpest** example of **pact** computing out there		neu	aspect changed
LEET	the mac mini is probably the simplest **3xampl3** of compact computing out there		neu	pos
PUNCT	the mac mini is probably the simplest**,** example of compact computing out there		pos	pos
Ours	the mac mini is probably the simplest example of **any** compact computing out there		neg	pos

Table 12: Detailed qualitative analysis of different methods. Here, pos: positive, neg: negative, neu:neutral

	Sentence	Aspect	Model Output	Human Pred
Actual	it is very easy to integrate bluetooth devices, and usb devices are recognized almost instantly	integrate bluetooth devices	pos	pos
baseline 1	it is very **hard** to integrate bluetooth devices, and usb devices are recognized almost instantly		neg	neg
baseline 2	it is very **hard** to integrate bluetooth devices, and usb devices are recognized almost instantly		neg	neg
baseline 3	it is very **uncomplicated** to integrate bluetooth devices, and usb devices are recognized almost instantly		neg	pos
LEET	it is very **345y** to integrate bluetooth devices, and usb devices are recognized almost instantly		pos	neu
PUNCT	it is very **easy,** to integrate bluetooth devices, and usb devices are recognized almost instantly		pos	pos
ours	it is very **basic** to integrate bluetooth devices, and usb devices are recognized almost instantly		neg	pos

Table 13: Detailed qualitative analysis of different methods. Here, pos: positive, neg: negative, neu:neutral