Improving the Robustness of Summarization Systems with Dual Augmentation

Xiuying Chen[1], Guodong Long[2], Chongyang Tao[3†], Mingzhe Li[4], Xin Gao[1†], Chengqi Zhang[2], Xiangliang Zhang[5,1†]

[1]Computational Bioscience Reseach Center, KAUST
[2]AAII, School of CS, FEIT, University of Technology Sydney
[3]Microsoft [4]Ant Group [5]University of Notre Dame
xiuying.chen@kaust.edu.sa

Abstract

A robust summarization system should be able to capture the gist of the document, regardless of the specific word choices or noise in the input. In this work, we first explore the summarization models' robustness against perturbations including word-level synonym substitution and noise. To create semantic-consistent substitutes, we propose a SummAttacker, which is an efficient approach to generating adversarial samples based on language models. Experimental results show that state-of-the-art summarization models have a significant decrease in performance on adversarial and noisy test sets. Next, we analyze the vulnerability of the summarization systems and explore improving the robustness by data augmentation. Specifically, the first brittleness factor we found is the poor understanding of infrequent words in the input. Correspondingly, we feed the encoder with more diverse cases created by SummAttacker in the input space. The other factor is in the latent space, where the attacked inputs bring more variations to the hidden states. Hence, we construct adversarial decoder input and devise manifold softmixing operation in hidden space to introduce more diversity. Experimental results on Gigaword and CNN/DM datasets demonstrate that our approach achieves significant improvements over strong baselines and exhibits higher robustness on noisy, attacked, and clean datasets[1].

Perturbation class: *Typo*	
Input	ministers from the european union and its mediterranean neighbors gathered here under heavy security on monday for an unprecedented conference (→confecence) on economic and political cooperation .
Original Summary	eu mediterranean nations meet for first-ever conference on cooperation. ✓
Perturbed Summary	eu mediterranean ministers meet in greece under heavy security. ×
Perturbation class: *Synonym substitution*	
Input	judge leonie brinkema ordered september ## conspirator zacarias moussaoui removed from the court here on monday after he repeatedly rejected his court- appointed defense attorney (→barrister) .
Original Summary	moussaoui removed from court after rejecting defense attorneys. ✓
Perturbed Summary	moussaoui removed from court after rejecting defense barris. ×
Input	president barack obama is imploring voters to support his government (→party) 's economic policies even though he acknowledged that those policies haven't brought about a recovery less than two months before the midterm elections .
Original Summary	obama says voters should back his economic policies.✓
Perturbed Summary	obama urges voters to back gop economic policies. ×

Table 1: Examples of vulnerability to BART-based summarization model. All examples show an initially correct summary turning into a wrong summary due to small changes in the input, *e.g.,* mis-spelling and synonym substitution.

1 Introduction

Humans have robust summarization processing systems that can easily understand diverse expressions and various wording, and overcome typos, misspellings, and the complete omission of letters when reading (Rawlinson, 2007). However, studies reveal that small changes in the input can lead to significant performance drops and fool state-of-the-art neural networks (Goodfellow et al., 2015; Belinkov and Bisk, 2018; Cheng et al., 2018). In text generation fields such as machine translation, Belinkov and Bisk (2018) showed that state-of-the-art models fail to translate even moderately noisy texts, Cheng et al. (2018) found that the generated translation is completely distorted by only replacing a source word with its synonym. However, the robustness on summarization models is less explored. Here, we show three summarization examples from the Gigaword dataset in Table 1. A fine-tuned BART model will generate a worse

† Corresponding author.

[1] https://github.com/iriscxy/robustness

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6846–6857
July 9-14, 2023 ©2023 Association for Computational Linguistics

summary for a minor change in the input including misspelling errors and synonym substitution, which often happen in practice due to the carelessness and habit of word usage in writing. Take the second case for example, an English user and an American user who use *barrister* or *attorney* will obtain summaries of different qualities. In the third case, a synonym word replacement even changes the subject of canvassing. Such weakness of summarization systems can lead to serious consequences in practice.

Despite its importance, robustness in summarization has been less explored. Jung et al. (2019) and Kryściński et al. (2019) examined positional bias and layout bias in summarization. Liu et al. (2021) introduced multiple noise signals in self-knowledge distillation to improve the performance of student models on benchmark datasets, but they did not explicitly evaluate the robustness of summarization models against noise.

Hence, in this work, we first evaluate the robustness of the existing state-of-the-art summarization systems against word-level perturbations including noise and adversarial attacks. The noise consists of natural human errors such as typos and misspellings. To create the adversarial attack test set, we come up with a model named SummAttacker. The core algorithm of SummAttacker is to find vulnerable words in a given document for the target model and then apply language models to find substituted words adjacent in the opposite direction of the gradient to maximize perturbations. We validate the effectiveness of SummAttacker on benchmark datasets with different attributes, *i.e.,* Gigaword and CNN/DailyMail. Experiment results show that by only attacking one word (1% token) in Gigaword and 5% tokens in CNN/DailyMail, the existing summarization models have drastically lower performance.

We next conduct a vulnerability analysis and propose two corresponding solutions to improve robustness. Our first conjecture is that worse summaries can be caused by replacing common words with uncommon and infrequently-used words, which the model might not understand well. Hence, we employ the outputs from SummAttacker as inputs for the encoder, so as to improve the diversity in the discrete input space. The second influencing factor is that the attacked inputs introduce more variations in the latent space. Correspondingly, we aim to expose the model to more diverse hidden states in the training process. Specifically, we build soft pseudo tokens by multiplying the decoder output probability with target token embeddings. These soft pseudo tokens and original tokens are then manifold softmixed on a randomly selected decoder layer to enlarge the training distribution. The interpolations leveraged in deeper hidden layers help capture higher-level information, improve semantic diversity, and provide additional training signal (Zeiler and Fergus, 2014). Experiments show that our dual augmentation for both encoder and decoder improves the robustness of summarization models on noisy and attacked test datasets.

Our main contributions are as follows:

- We empirically evaluate the robustness of recent summarization models against perturbations including noise and synonym substitutions.
- To improve the robustness of summarization models, we propose a dual data augmentation method that introduces diversity in the input and latent semantic spaces.
- Experimental results demonstrate that our augmentation method brings substantial improvements over state-of-the-art baselines on benchmark datasets and attacked test datasets.

2 Related Work

We discuss related work on robust abstractive summarization, adversarial examples generation, and data augmentation.

Robust Abstractive Summarization. Ideally, a robust text generation system should consistently have high performance even with small perturbations in the input, such as token and character swapping (Jin et al., 2020), paraphrasing (Gan and Ng, 2019), and semantically equivalent adversarial rules (Ribeiro et al., 2018). Considerable efforts have been made in the text generation field. For example, Cheng et al. (2019) defended a translation model with adversarial source examples and target inputs. However, the robustness in the summarization task has been less explored. Jung et al. (2019) and Kryściński et al. (2019) showed that summarization models often overfit to positional and layout bias, respectively. In contrast, in this work, we focus on the robustness of summarization models against word-level perturbations.

Adversarial Examples Generation. Classic attacks for text usually adopt heuristic rules to modify the characters of a word (Belinkov and Bisk,

2018) or substitute words with synonyms (Ren et al., 2019). These heuristic replacement strategies make it challenging to find optimal solutions in the massive space of possible replacements while preserving semantic consistency and language fluency. Recently, Li et al. (2020) proposed to generate adversarial samples for the text classification task using pre-trained masked language models exemplified by BERT. In this paper, we focus on attacking summarization models, which is a more challenging task, since the model compresses the input, and perturbations on unimportant parts of the source might be ignored.

Data Augmentation. Data augmentation aims to generate more training examples without incurring additional efforts of manual labeling, which can improve the robustness or performance of a target model. Conventional approaches introduce discrete noise by adding, deleting, and/or replacing characters or words in the input sentences (Belinkov and Bisk, 2018). More recently, continuous augmentation methods have been proposed. Cheng et al. (2020) generated adversarial sentences from a smooth interpolated embedding space centered around observed training sentence pairs, and shows its effectiveness on benchmark and noisy translation datasets. Xie et al. (2022) proposed a target-side augmentation method, which uses the decoder output probability distributions as soft indicators. Chen et al. (2023) selectively augmented training dataset considering representativeness and generation quality. In this work, we propose a dual augmentation method that utilizes discrete and virtual augmented cases.

3 The Proposed SummAttacker

Formally, given a trained summarization model with parameters $\boldsymbol{\theta}$, the purpose of an attacking model is to slightly perturb the input x such that the summarization output of the perturbed $\hat{x}$ deviates away from the target summary y:

$$\{\hat{x} | \mathcal{R}(\hat{x}, x) \leq \epsilon, \underset{\hat{x}}{\operatorname{argmax}} - \log P(y | \hat{x}; \boldsymbol{\theta})\}, \quad (1)$$

where $\mathcal{R}(\hat{x}, x)$ captures the degree of imperceptibility for a perturbation, *e.g.,* the number of perturbed words. To make a maximal impact on the summarization output with a perturbation budget ϵ, a classical way is to launch gradient-based attacks (Cheng et al., 2019). In this section, we propose a SummAttacker for crafting adversarial samples

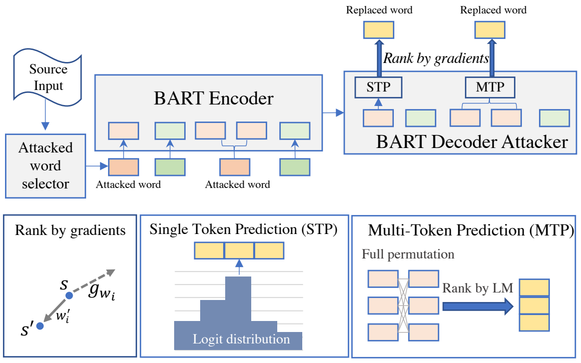

Figure 1: Overview of SummAttacker. It first selects vulnerable words to attack, and then replaces them with words based on language model (LM) prediction and gradient-based ranking. The replacement word w_i' changes the model state s to s' in the opposite direction of optimization, $-\mathbf{g}_{w_i}$.

that may differ only a few words from genuine inputs but have low-quality summarization results. Due to its capacity and popularity, we take BART (Lewis et al., 2020) as the backbone summarization model, as shown in Fig.1.

Attacked Word Selector. Since it is intractable to obtain an exact solution for Equation 1, we, therefore, resort to a greedy approach to circumvent it. In BART kind of summarization model based on Transformer architecture, the sequence representation vector $\boldsymbol{s}$ of input tokens in x is first projected to keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ using different linear mapping functions. At the t-th decoding step, the hidden state of the previous decoder layer is projected to the query vector $\boldsymbol{q}_t$. Then $\boldsymbol{q}_t$ is multiplied by keys $\boldsymbol{K}$ to obtain an attention score $\boldsymbol{a}_t$ and the t-th decoding output:

$$\operatorname{Attn}(\boldsymbol{q}_t, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{a}_t * \boldsymbol{V}, \ \boldsymbol{a}_t = \operatorname{softmax}\left(\frac{\boldsymbol{q}_t \boldsymbol{K}^T}{\sqrt{d}}\right),$$

where d is the hidden dimension. A token that obtains the highest attention score over all decoding steps is the most important and influential one to the summarization model. We select the word w_i to attack if it contains or equals the most important token. To avoid changing factual information, we restrict w_i not to be people names and locations.

Attacking with LM and Gradients. Next, we aim to find a replacement word that is semantically similar to w_i but is adversarial to the summarization model. Language models are empowered to generate sentences that are semantically accurate, fluent, and grammatically correct. We take advantage of this characteristic to find a replacement word w_i' for the target word w_i. The general idea is to first iden-

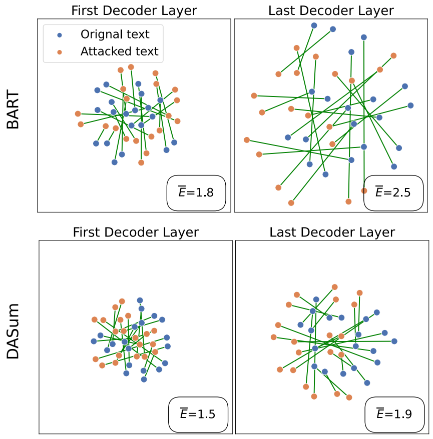

Figure 2: t-SNE visualization of the hidden states in BART and our DASum, when taking original and attacked inputs. $\overline{E}$ is the average Euclidean distance of paired original and attacked states before using t-SNE.

tify the top likely candidates that are predicted by the language model for w_i, and then select the best candidate with the guidance of prediction gradient.

Concretely, we first feed the tokenized sequence into the BART model to get a prediction for the attacked word w_i. As shown in Fig.1, for w_i with a single token, we use STP (Single Token Prediction) operation to simply obtain the top K predictions that are semantically similar to w_i. For w_i with multiple tokens, we have MTP (Multi-Token Prediction), which lists $c \times K$ possible combinations from the prediction, where c is the token number in the word. Then we rank the perplexity of all combinations to get the top-K candidate combinations, denoted as $\mathcal{V}_K$. We filter out stop words and antonyms using NLTK and synonym dictionaries.

Following the idea of a gradient-based attack model, we then find the most adversarial word w_i' that deviates from w_i towards a change aligned with the prediction gradient:

$$\begin{aligned} \mathbf{g}_{w_i} &= \nabla_{\mathbf{e}(w_i)} \log P(y|x;\boldsymbol{\theta}), \\ w_i' &= \underset{w \in \mathcal{V}_K}{\operatorname{argmax}} \ \operatorname{sim}\left(\mathbf{e}(w) - \mathbf{e}\left(w_i\right), -\mathbf{g}_{w_i}\right), \end{aligned} \tag{2}$$

where sim$(\cdot,\cdot)$ is cosine distance, and $\mathbf{e}$ is word embedding function. As shown in Fig. 1, the replacement word w_i' changes the model state s to s' in the opposite direction of optimization, $-\mathbf{g}_{w_i}$.

4 Dual Augmentation

With the proposed attacking model, we first analyze the influences of attacking, and then propose our DASum to counter the negative effects.

Vulnerability Analysis. We first look into the word perturbation in attacked inputs that result in worse summaries. Our conjecture is that worse summaries can be caused by replacing common words with uncommon and infrequently-used words, which the model might not understand well. Through the analysis of 50 worse summary cases, our conjecture is verified by the observation that the frequency of the replacement words is 4 times lower than the original words on average. Especially for those worse summaries including unexpected words not existing in the input, we found that the co-occurrence of the unexpected word in the generated summary and the replacement word in the input is usually high. Take the third case with unexpected work *gop* in Table 1 for example, the co-occurrence for the word pair $\{$*party*, *gop*$\}$ is 6 times higher than that of $\{$*government*, *gop*$\}$. These analysis results imply that the model's vulnerability is highly related to the word frequency distribution and the diversity of the training documents.

Next, we investigate the influence of attack in the latent space. It is well known that in the text generation process, a change of a predicted preceding word will influence the prediction of words after it, since the following prediction will attend to the previously generated words (Lamb et al., 2016). This error accumulation problem can be more severe in attacked scenarios since the perturbations can bring more variety in the decoder space. To verify our assumption, we evaluate the change in hidden states of the BART model for 20 cases in the original and the corresponding attacked test sets. The top part of Fig.2 visualizes the hidden states in the first and last BART decoder layer. It can be seen that as the information flows from the low to high layers in the decoder, the hidden states in the latent space show larger diversity, as the distances between paired hidden states get larger. We also calculate the Euclidean distance $\overline{E}$ of paired states, which increases from 1.8 to 2.5. To improve the summarization robustness against attacks, the decoder could be trained with augmentation in latent space to comfort with diversity.

Augmentation Design. Based on the above analysis, we first propose to incorporate the corpus obtained by SummAttacker as *augmentation input for encoder*, so as to improve the diversity of words in training documents (illustrated as yellow squares with solid lines in Fig.3(a)). To alleviate the impact of perturbation on the decoding process, we

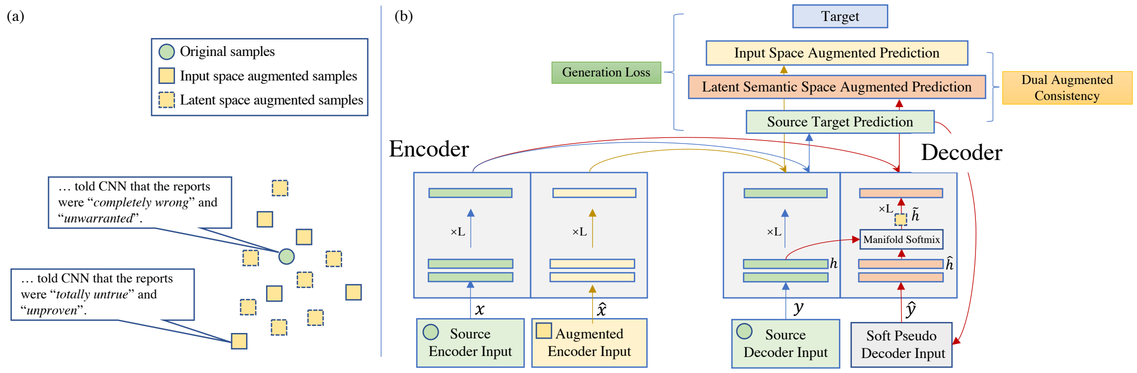
(a)
Original samples
Input space augmented samples
Latent space augmented samples
… told CNN that the reports were "*completely wrong*" and "*unwarranted*".
… told CNN that the reports were "*totally untrue*" and "*unproven*".
(b)
Target
Input Space Augmented Prediction
Generation Loss
Latent Semantic Space Augmented Prediction
Dual Augmented Consistency
Source Target Prediction
Encoder
Decoder
×L
×L
×L
×L
$\tilde{h}$
Manifold Softmix
h
$\hat{h}$
x
$\hat{x}$
y
$\hat{y}$
Source Encoder Input
Augmented Encoder Input
Source Decoder Input
Soft Pseudo Decoder Input

Figure 3: (a) Illustration of training examples sampled from vicinity distributions that could cover variants of literal expression under the same meaning. (b) The architecture of our dual data augmentation approach.

propose a continuous data augmentation method in the *latent space of decoder*, where multiple virtual representations are constructed for each training instance to make the decoder be exposed to diverse variants of the latent representation of the same input document (illustrated as yellow squares with dash lines in Fig.3(a)).

Input Space Augmentation. The input space augmentation in the encoder side is straightforward, as the output from SummAttacker can be directly employed as encoder inputs. Concretely, we use SummAttacker to automatically generate an augmented input document for the original document, denoted as $\hat{x}$. We then train the summarization model with the original and augmented dataset, where the training objective is denoted as $\mathcal{L}_o = \log P\left(y|x\right)$ and $\mathcal{L}_e = \log P\left(y|\hat{x}\right)$, respectively. We also randomly add noisy words in both inputs. We show this process in Fig.3(b), where we draw the same encoder twice to denote the training on original and augmented inputs.

Latent Semantic Space Augmentation. Based on the vulnerability analysis in the decoding process, we are motivated to mitigate the impact of adversarial attacks by exposing the decoder to diverse variants of the latent representations. The variants are established by an adversarial input and a manifold softmix technique applied on randomly selected layers in the decoder.

We first define a virtual adversarial decoder input $\hat{y}_t$ apart from the original input y_t by integrating the embedding of words that are all likely to be generated. Let $\mathbf{l}_t$ be the decoder's predicted logits before softmax, where $t \in \{1, 2, ..., m\}$, $l_t[v]$ be the logit of v token, and m is the token length of y. We compute the pseudo decoder inputs as:

$$\hat{y}_t = \frac{\exp\left(\mathbf{l}_t/T\right)}{\sum_{v=1}^{|\mathcal{V}|} \exp\left(l_t[v]/T\right)} \mathbf{W}, \tag{3}$$

where $\mathcal{V}$ is the vocabulary size, $\mathbf{W}$ is the word embedding matrix with size $|\mathcal{V}| \times d$, T is the softmax temperature.

Next, we construct the virtual adversarial hidden states in the decoder by interpolating $\boldsymbol{h}^k$ and $\hat{\boldsymbol{h}}^k$, which are the hidden states of inputs $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ at a randomly selected k-th layer:

$$\tilde{\boldsymbol{h}}^k = \lambda \boldsymbol{h}^k + (1-\lambda)\hat{\boldsymbol{h}}^k, \tag{4}$$

where λ is the mixup ratio between 0 and 1. The mixup layer $k \in [0, L]$, where L is the decoder layer number.

In the decoding process, $\hat{y}_t$ servers as variants of y_t and integrates the embedding of words that are likely to be generated in each step. The variants of hidden states $\tilde{\boldsymbol{h}}^k$ behave like the hidden states of attacked input text. The latent space augmentation objective is $\mathcal{L}_d = \log P\left(y|x, \hat{y}\right)$. As shown in Fig.3, the latent semantic space augmented prediction is a kind of additional training task for decoder with variant samples indicated by yellow squares with dash lines. Note that our proposed manifold softmix differs from the target-side augmentation in Xie et al. (2022), which mixed the pseudo decoder input with the ground truth input in the word embedding layer, and only introduces low-level token variations.

Lastly, according to recent studies (Chen et al., 2020), maximizing the consistency across various augmented data that are produced from a single piece of data might enhance model performance. Herein, we minimize the bidirectional Kullback-Leibler (KL) divergence between the augmented

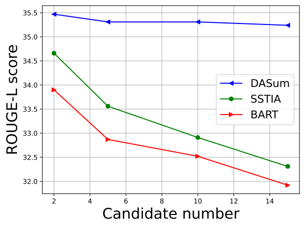

Figure 4: Performance of different models on the Gigaword test set when attacked by SummAttacker with different candidate number K.

Dataset		Semantic	Grammar	Similarity
Gigaword	Original	4.4	4.7	-
	Adversarial	4.1	4.5	0.96
CNN/DM	Original	4.4	4.6	-
	Adversarial	4.0	4.2	0.94

Table 2: Human and automatic evaluation of the adversarial samples from SummAttacker, as well as the original samples for taking a reference.

data and real data, to stabilize the training:

$$\mathcal{L}_c = \mathcal{D}_{KL}\left(P\left(y|x\right) \| P\left(y|x,\hat{y}\right)\right) + \mathcal{D}_{KL}\left(P\left(y|x\right) \| P\left(y|\hat{x}\right)\right). \tag{5}$$

Our final loss function is defined as $\mathcal{L}_o + \mathcal{L}_e + \mathcal{L}_d + \mathcal{L}_c$.

5 Experimental Setup

5.1 Dataset

We experiment on two public datasets, Gigaword (Napoles et al., 2012) and CNN/DM (Hermann et al., 2015), which have been widely used in previous summarization works. The input document in Gigaword contains 70 words, while CNN/DM consists of 700 words on average. Hence, we can examine the effectiveness of our methods on datasets of different distributions.

5.2 Comparison Methods

Our baselines include the following models:

BART (Lewis et al., 2020) is a state-of-the-art abstractive summarization model pretrained with a denoising autoencoding objective.

ProphetNet (Qi et al., 2020) is a pre-training model that introduces a self-supervised n-gram prediction task and n-stream self-attention mechanism.

R3F (Aghajanyan et al., 2021) is a robust text generation method, which replaces adversarial objectives with parametric noise, thereby discouraging representation change during fine-tuning when possible without hurting performance.

SSTIA (Xie et al., 2022) augments the dataset from the target side by mixing the augmented decoder inputs in the embedding layer.

5.3 Implementation Details

We implement our experiments in Huggingface on NVIDIA A100 GPUs, and start finetuning based on pretrained models facebook/bart-large. Concretely, there are 12 encoding layers in the encoder and the decoder. The activation functions are set to GeLUs and parameters are initialized from $\mathcal{N}(0, 0.02)$. We use Adam optimizer with ϵ as 1e-8 and β as (0.9, 0.98). We used label smoothing of value 0.1, which is the same value as Vaswani et al. (2017). Then attacking candidate number K is set to 10 based on the parameter study. The learning rate is set to 3e-5. The warm-up is set to 500 steps for CNN/DM and 5000 for Gigaword. The batch size is set to 128 with gradient accumulation steps of 2. Following Xie et al. (2022), the temperature in Equation 3 is set to 0.1 for CNN/DM and 1 for Gigaword, and the mixup ratio λ in Equation 4 is set to 0.7. We set the attack budget to 1% tokens for Gigaword and 5% tokens for CNN/DM, based on the consideration of attacking performance and semantic consistency. We use the original dataset plus the augmented cases generated by SummAttacker as our training dataset, where we also randomly add 30% natural human errors to improve the understanding of noises. The training process takes about 8 hours and 4 hours for CNN/DM and Gigaword.

5.4 Evaluation Metrics

We first evaluate models using standard ROUGE F1 (Lin, 2004). ROUGE-1, ROUGE-2, and ROUGE-L refer to the matches of unigrams, bigrams, and the longest common subsequence, respectively. We use BERTScore (Zhang et al., 2020) to calculate similarities between the summaries. We further evaluate our approach with the factual consistency metric, QuestEval (Scialom et al., 2021) following Chen et al. (2022). It measures to which extent a summary provides sufficient information to answer questions posed on its document. QuestEval considers not only factual information in the generated summary, but also the information from its source text, and then gives a weighted F1 score.

Dataset	Model	Traditional Metric			Advanced Metric			
		ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	QE(R)	QE(P)	QE(F1)
Gigaword	BART	35.23	15.64	32.52	87.33	22.42	22.32	22.37
	ProphetNet	35.56	15.87	32.79	88.45	23.48	23.76	23.62
	R3F	35.69	16.29	32.91	88.60	23.05	23.79	23.42
	SSTIA	36.55	16.90	33.25	88.72	23.52	24.01	23.76
	DASum	**38.15**	**18.53**	**35.31**	**88.90**	**27.39**	**28.95**	**28.17**
	DASum w/o $\mathcal{L}_e$	36.71	18.17	34.01	88.61	24.89	26.63	25.76
	DASum w/o $\mathcal{L}_d$	37.36	18.31	34.64	88.71	24.64	26.93	25.79
	DASum w/o $\mathcal{L}_c$	37.21	18.30	34.32	88.64	25.56	26.19	25.87
CNN/DM	BART	36.45	12.29	33.36	87.23	22.05	17.47	19.76
	ProphetNet	36.98	12.68	33.8	87.33	22.28	17.43	19.85
	R3F	37.28	12.98	34.83	87.59	22.14	17.88	20.01
	SSTIA	37.49	13.05	35.15	87.69	22.46	17.96	20.21
	DASum	**42.17**	**18.06**	**39.08**	**88.90**	**28.66**	**25.62**	**27.14**

Table 3: Performance of baselines and our model DASum on perturbed inputs by SummAttacker (the attack budget is 1% and 5% tokens in Gigaword and CNN/DM datasets respectively. Numbers in **bold** mean that the improvement to the best baseline is statistically significant (a two-tailed paired t-test with p-value <0.05).

6 Experimental Results

6.1 SummAttacker Evaluation

Before reporting the summarization performance boosted by our proposed dual augmentation strategy, we first set up human and automatic metrics to evaluate the quality of the generated adversarial augmentation cases. For human evaluation, we ask annotators to score the semantic and grammar correctness of the generated adversarial and original sequences, scoring from 1-5 following Jin et al. (2020) and Li et al. (2020). We randomly select 100 samples of both original and adversarial samples for human judges. Each task is completed by three Ph.D. students. For automatic metric, following Li et al. (2020), we use Universal Sentence Encoder (Cer et al., 2018) to measure the semantic similarity between the adversarial and the original documents.

As shown in Table 2, the adversarial samples' semantic and grammatical scores are reasonably close to those of the original samples. The scores are generally higher on Gigaword dataset than CNN/DM. This corresponds to the setting that the number of attacked words is larger on CNN/DM dataset. The kappa statistics are 0.54 and 0.48 for semantic and grammar respectively, indicating moderate agreements between annotators. For the automatic evaluation, the high semantic similarity demonstrates the consistency between the original and attacked documents.

We also study the influence of the candidate number K in SummAttacker. In Fig. 4, all models perform worse when the input document is perturbed by SummAttacker with a larger K, since a better replacement word w_i' can be found in a larger search space. From the viewpoint of generating adversarial samples, it is not worth using a large K, because the time and memory complexity increase with K as well. Thus, we use K=10 in our setting.

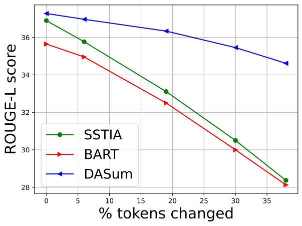

Figure 5: The impact of noise on the performance of summarization models on Gigaword. While SSTIA and BART show significant drops in all metrics, our DASum has a robust performance. The noise here consists of multiple human errors (typos, misspellings, etc.)

6.2 Robustness Evaluation

We next report the evaluation results of summarization models when the input documents are perturbed by natural human errors (noise) and synonym substitutions (based on SummAttacker).

Robustness on Noisy Datasets. Humans make mistakes when typing or spelling words, but they have the capability of comprehensive reading to understand the document without being interrupted by such noises. Thus, we first examine the robustness of the recent summarization models against natural human errors. Since we do not have access to a summarization test set with natural noise, we use the look-up table of possible lexical replacements (Belinkov and Bisk, 2018), which collects naturally occurring errors (typos, misspellings, etc.). We replace different percentages of words in the Gigaword test set with an error if one exists in the

Attacked Document & Reference	SSTIA on clean input	SSTIA on attacked input	DASum on clean input	DASum on attacked input
Doc: overcrowding and lick of illumination at exit popints at konkola stadiom in UNK province of zambia were among tu major lapses that lead to a stampede resulting in the dieth of ## sokker fun afrer ana africa coop... **Ref:** overcrowding lack of illumination leads to stampede in zambia: investigation	overcrowding blamed for stampede in zambia	## zambian soccer fans injured in stampede.	overcrowding blamed for soccer stampede in zambia	overcrowding blamed for soccer stampede in zambia
Doc: philippine president fidel ramos, who was hospitalized for the second time in ## days over the weekend, may need heart surgery, his spokesman (→spokesperson) said. **Ref:** philippine president hospitalized may need heart surgery	philippine president may need heart surgery	ramos may need heart surgery	philippine president may need heart surgery spokesman say	philippine president ramos may need heart surgery
Doc: gusty winds pushed a wildfire (→bonfire) closer to sun valley resort 's ski area, while hundreds more homes were ordered evacuated in the valley below." **Ref:** gusty winds whip idaho wildfire near sun valley ski area ; hundreds more homes evacuated	hundreds more homes evacuated as wildfire threatens ski resort	hundreds more homes evacuated as winds push bonfire closer to california ski resort	winds push wildfire closer to sun valley ski area	winds push wildfire closer to sun valley ski area

Table 4: Comparisons of summaries generated by baseline models and our method on the noisy document (the first row) and attacked document (last two rows). The missing information or inconsistent information caused by perturbations on the baseline model and the consistent information given by our model is highlighted.

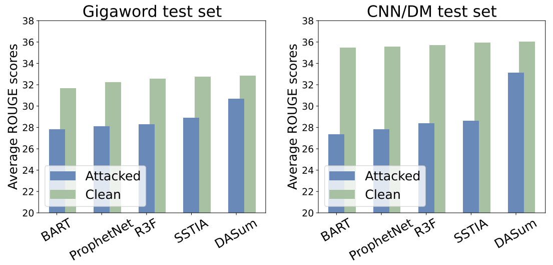

Figure 6: Performance of baselines and our model on attacked and clean Gigaword and CNN/DM test set.

look-up table. We show the performance of classic baseline BART, augmentation-based model SSTIA, and our model in Fig. 5. Both baseline models suffer a significant drop in all metrics when evaluated on texts consisting of different percentages of noise. Our DASum model is more robust and drops the least in all four metrics compared with baselines. We also give an example in the first row in Table 4. Humans are quite good at understanding such scrambled texts, whereas existing summarization models are still vulnerable to slight perturbations and then fail to capture the gist of the input document, due to the lack of robustness enhancement training.

Robustness on Datasets Perturbed by Adversarial Attacks. We next examine the robustness of summarization models on the test datasets perturbed by adversarial attacks. For the Gigaword dataset, we set attack budget ϵ to be only 1 word (1% tokens), and for CNN/DM we set ϵ to be 5% tokens of the input document.

The comparison of performance on attacked and clean datasets is shown in Fig.6. It can be seen that despite the perturbation being only on a few words, all four baselines suffer a significant drop in performance compared with their performance on the clean test set. Specifically, the ROUGE-1 score of the latest SSTIA model drops by 4.01 on Gigaword, and the average ROUGE score drops by 7.33 for R3F model on CNN/DM dataset. This highlights the vulnerability of the existing summarization models and also demonstrates the effectiveness of our attacking model. Nevertheless, the drop percentage of our model is the least compared with other baselines in all metrics. Specifically, our model drops the least with only 2.22 and 0.28 decreases in ROUGE-2 and BERTScore metrics, respectively, on the Gigaword dataset. We show the detailed performance on attacked set in Table 3. Our model outperforms baselines on two datasets in most metrics. Besides, we also observe that the summarization models of short documents are more vulnerable than those of long documents. One potential reason is that the summarization model is more dependent on each input word when the input is shorter. When the input is longer, the importance of each word decreases, since the model can resort to other sources to generate summaries.

Ablation Study. We first investigate the influence of ***input space augmentation***. As shown in Table 3, without the $\mathcal{L}_e$ loss, the performance drops the most. We also conduct diversity analysis on the inputs after augmentation, corresponding to the vulnerability discussion in §4. The ratio of uncommon words compared with the original common words increases by 30%, which directly verifies our assumption that introducing variations in the training dataset improves the robustness of the summarization model. Next, we study the effect of ***latent***

space augmentation. Specifically, the ROUGE-1 score of extractive summarization drops by 0.79 after the $\mathcal{L}_d$ is removed. This indicates that the model benefits from hidden states with more diversity in the training process. In addition, we compare the decoder hidden states of DASum with that of BART in Fig.2. The deviation of paired original and attacked hidden states in DASum is effectively reduced ($\overline{E}$ drops from 2.5 to 1.9 in the last layer). Thirdly, the performance of DASum w/o $\mathcal{L}_c$ shows that *dual consistency* can also help improve robustness. We also note that DASum is always more robust than the other two baselines, in regard to different attacking settings in Fig.5.

7 Conclusion

In this paper, we investigate the robustness problem in the summarization task, which has not been well-studied before. We first come up with a SummAttacker, which slightly perturb the input documents in benchmark test datasets, and causes a significant performance drop for the recent summarization models. Correspondingly, we propose a dual data augmentation method for improving the robustness, which generates discrete and virtual training cases in the same meaning but with various expression formats. Experimental results show that our model outperforms strong baselines.

Limitations

We discuss the limitations of our framework as follows:

(1) In this paper, we take an initial step on the robustness of the summarization system by focusing on word-level perturbations in the input document. However, in practice, the robustness of the summarization models is reflected in many other aspects. For example, the summarization performance towards sentence-level or document-level perturbations is also a kind of robustness.

(2) Although DASum greatly improves the generation quality compared with other augmentation-based models, it requires more computational resources with respect to the augmented dataset construction process. For large-scale datasets with long text (e.g., BigPatent (Sharma et al., 2019)), it is worth considering the time complexity of Transformer architecture.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. The work was supported by King Abdullah University of Science and Technology (KAUST) through grant awards FCC/1/1976-44-01, FCC/1/1976-45-01, REI/1/5234-01-01, RGC/3/4816-01-01, and RGC/3/4816-01-01;.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. *Proc. of ICLR*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. *Proc. of ICLR*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Xiuying Chen, Mingzhe Li, Xin Gao, and Xiangliang Zhang. 2022. Towards improving faithfulness in abstractive summarization. In *Proc. of NeurIPS*.

Xiuying Chen, Mingzhe Li, Jiayi Zhang, Xiaoqiang Xia, Chen Wei, Jianwei Cui, Xin Gao, Xiangliang Zhang, and Rui Yan. 2023. Learning towards selective data augmentation for dialogue generation. *Proc. of AAAI*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proc. of ACL*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. In *Proc. of ACL*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proc. of ACL*.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proc. of ACL*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *Proc. of ICLR*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proc. of NIPS*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? natural language attack on text classification and entailment. In *Proc. of AAAI*.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. 2019. Earlier isn't always better: Subaspect analysis on corpus and system biases in summarization. In *Proc. of EMNLP*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proc. of EMNLP*.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. *Proc. of NIPS*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. In *Proc. of EMNLP*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. In *Proc. of AACL*.

Courtney Napoles, Matthew R Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequencepre-training. In *Proc. of EMNLP*.

Graham Rawlinson. 2007. The significance of letter position in word recognition. *IEEE Aerospace and Electronic Systems Magazine*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proc. of ACL*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proc. of ACL*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. Questeval: Summarization asks for fact-based evaluation. In *Proc. of EMNLP*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proc. of ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NIPS*.

Shufang Xie, Ang Lv, Yingce Xia, Lijun Wu, Tao Qin, Tie-Yan Liu, and Rui Yan. 2022. Target-side input augmentation for sequence to sequence generation. In *Proc. of ICLR*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proc. of ECCV*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proc. of ICLR*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
in limitation section

☑ A2. Did you discuss any potential risks of your work?
in limitation section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
in introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

in experiment section

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
in experiment section

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
in experiment section

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
in experiment section

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
in experiment section

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
in appendix

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
in appendix

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.