SemPerGe: Unveiling Text-based Adversarial Attacks on Semantic Communication

Afia Anjum, Arkajyoti Mitra, Paul Agbaje, Md. Ahanaful Alam, Debashri Roy, Md Salik Parwez, Hebeeb Olufowobi
University of Texas at Arlington, Arlington, TX, USA
{afia.anjum, habeeb.olufowobi}@uta.edu

***Abstract*—Deep learning (DL)-based semantic communication (SC) redefines traditional communication by shifting the focus from reliable bit-by-bit transmission to conveying only task-relevant information, thereby reducing bandwidth usage. However, SC is vulnerable to adversarial attacks due to wireless channel exposure and the susceptibility of DL models to small input perturbations. While extensive research has explored adversarial attacks in image-based SC, there is limited research on text-based adversarial noise targeting text generation models. Therefore, we propose Semantic Perturbation Generator (SemPerGe), the first framework designed to craft targeted adversarial perturbations in transmitted text data within SC. SemPerGe operates without prior knowledge of the DL model architecture, parameters, or logits, instead leveraging off-the-shelf large language models to introduce semantic noise effectively. The framework is composed of two phases: (i) a Significant Token Marker that identifies crucial tokens influencing the semantics of transmitted content and (ii) a Perturbation Generator, which modifies these tokens to subtly alter the content's meaning while preserving linguistic coherence and grammatical structure. We evaluate SemPerGe against four baselines across datasets from two application domains, demonstrating its robustness and adaptability. Additionally, a user study confirms the stealthiness of the generated adversarial texts, with 96% of participants unable to detect adversarial modifications on average.**

I. Introduction

The advent of next-generation networks, such as 5G and beyond, alongside the proliferation of the Internet of Things (IoT) has significantly increased the demand for bandwidth due to the massive scale of simultaneous data transmissions [1]. Traditional communication systems transmit complete message data, typically encoded in bits or symbols, with an emphasis on reliability rather than efficiency [2]. To address the growing strain on bandwidth and spectrum, semantic communication has been proposed as a paradigm shift: by transmitting only task-relevant semantic information, this approach enables more efficient, goal-driven communication using deep learning (DL) models [3]. Unlike conventional systems that transmit exact message representations, semantic communication systems extract and transmit minimal yet sufficient information for downstream tasks (e.g., answering a question or performing a classification). This focus on intent rather than form introduces compression and robustness benefits, but it also opens new security risks. Specifically, the dependence on DL models renders semantic communication vulnerable to semantic-level adversarial attacks—perturbations that subtly shift the intended meaning without syntactic corruption [4].

In text-based semantic communication, such perturbations, referred to as *semantic noise*, can be especially damaging. Small word-level changes may dramatically shift meaning or mislead downstream tasks, particularly in text generation applications where coherent sequencing is crucial [5]. Prior studies have demonstrated the feasibility of over-the-air attacks in semantic communication [6], but most have focused on image data. Text-based adversarial attacks remain underexplored in this context despite posing unique challenges due to the fragility of meaning and coherence in natural language [7].

This paper addresses this gap by introducing the *Semantic Perturbation Generator* (SemPerGe), a novel framework for generating adversarial text perturbations designed to deceive text generation models within semantic communication frameworks. Unlike prior work focused on classifiers [8], SemPerGe targets generative tasks, which are inherently more resistant to shallow perturbations and require nuanced, semantically coherent alterations. Our approach assumes a *grey-box* adversary with partial access to transmitted semantic content—realistic in scenarios where intermediate representations or over-the-air text sequences are observable [6]. While the full-stack communication system may include source/channel encoders and encryption layers, we focus on vulnerabilities in semantic layers where DL-based generative models operate.

SemPerGe consists of two key components: (i) a Significant Token Marker that identifies semantically influential tokens using transformer attention and named entity recognition (NER), and (ii) a Perturbation Generator that leverages fine-tuned large language models (LLMs) to craft coherent adversarial replacements. Importantly, SemPerGe operates without access to model parameters or logits, highlighting the feasibility of semantic attacks using publicly available APIs and models.

Our contributions are summarized as follows:

- We introduce SemPerGe, a framework for generating adversarial semantic perturbations in text-generation-based semantic communication systems, using only publicly available LLMs without requiring internal access to the communication models.
- SemPerGe features a two-phase architecture: (i) **Significant Token Marker**, which heuristically identifies key semantic tokens using attention maps and NER, and (ii) **Perturbation**

Generator, which alters these tokens to induce semantic drift while maintaining syntactical coherence.
- We evaluate SemPerGe on two representative datasets across different application domains, benchmarking its attack success and stealth against four baseline methods. Our results demonstrate superior effectiveness in misleading generative models with minimal perceptibility.
- We conduct a human evaluation to assess fluency and detectability of adversarial examples, validating that SemPerGe-generated outputs remain semantically deceptive yet linguistically coherent to human readers.

II. Background and Related Work

A. Transformers

Transformers [9] have transformed NLP [10] by introducing an architecture capable of modeling long-range dependencies in text. Unlike traditional sequence models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, transformers employ self-attention mechanisms that allow them to process input tokens in parallel, thereby enabling efficient and global context capture [11].

The transformer architecture consists of an encoder-decoder structure. The encoder processes input sequences, while the decoder generates output sequences [12]. Input tokens are first transformed into high-dimensional embeddings that capture semantic meaning [13]. To retain token order, positional encodings are added to these embeddings.

The encoder comprises N identical layers, each containing a *multi-head self-attention* mechanism followed by a *feed-forward neural network* (FFNN). The attention mechanism computes attention scores between tokens to identify salient contextual relationships. Multiple attention heads capture diverse aspects of token interactions, enriching the overall contextual understanding. The outputs of the self-attention mechanism are refined by the FFNN and propagated through all layers. The decoder mirrors the encoder and also consists of N identical layers. Each layer includes a *masked multi-head self-attention* block (to ensure autoregressive generation), a *multi-head attention* block for cross-attending to the encoder's output, and an FFNN. The decoder shifts its input rightward, prepending a special <EOS> (end-of-sequence) token. Like the encoder, embeddings and positional encodings are applied [13]. The final output passes through a linear layer and a Softmax function to yield a probability distribution over possible next tokens [12].

B. Semantic Communication

Traditional communication systems emphasize accurate data transmission, commonly measured by bit error rate (BER) or symbol error rate (SER) [14]. However, as communication systems evolve toward interconnected, intelligent applications, the focus shifts toward conveying the *meaning* or *semantics* of information. Semantic communication addresses this shift by transmitting only task-relevant content, thereby eliminating redundancy and enabling more efficient communication [15].

Comparison with Traditional Communication. Figure 1 compares traditional and semantic communication systems. In Fig. 1(a), traditional architectures include a *source encoder*, *channel encoder*, and *modulator* at the transmitter, paired with corresponding decoders and a demodulator at the receiver [16]. These modules aim to reliably transmit the full message.

In contrast, Fig. 1(b) introduces a semantic layer that incorporates deep learning-based *semantic encoders* and *semantic decoders*, often built on transformer-based LLMs [17]. Here, the semantic encoder extracts task-relevant features, and the decoder reconstructs them using background knowledge and context, improving communication efficiency and robustness.

C. Related Work

DL-based semantic communication systems, while efficient and task-aware, inherit vulnerabilities from their underlying neural network models, particularly to adversarial perturbations [18]. Research in this area has focused primarily on disrupting semantic communication either directly (by modifying inputs) or over-the-air (by injecting perturbations into the transmission channel).

Hu *et al.* [5] proposed an attack that perturbs the semantic encoder's input directly to deceive the decoder at the receiver. However, this assumes access to the transmitter's data, which limits applicability in realistic over-the-air threat models. Sagduyu *et al.* [19] demonstrated that even minor perturbations added over-the-air can manipulate received semantics, misleading both the decoder and downstream task classifiers. Bahramali *et al.* [4] introduced universal adversarial perturbations that are input-agnostic, aiming to broadly degrade model performance without targeting specific outputs. Li *et al.* [7], [20] developed a method using a surrogate encoder and decoder queries to craft perturbations without requiring access to the actual model. However, most existing research in semantic communication security focuses on *image*-based data, where imperceptible pixel-level changes are effective in altering semantic outcomes.

In contrast, adversarial research in *text*-based domains, particularly NLP, has largely targeted text classifiers. For instance, TEXTBUGGER [8] introduces character-level and word-level perturbations (e.g., typos, misspellings) that successfully mislead sentiment or toxicity classifiers. SemAttack [21] iteratively modifies words using typos or synonyms to cause classification shifts. TextGuise [22] alters important tokens using synonyms or emojis, and LLM-Attack [23] replaces critical tokens based on model logits. While these methods are effective for classification models, they often fail to deceive *text generation models*, which are more resilient to superficial perturbations due to their focus on coherent sequences.

Despite advancements in adversarial NLP, there remains a critical gap in targeting text generation within semantic communication. Prior approaches either rely on classification-specific vulnerabilities or assume impractical access to system internals. We address this gap by proposing SemPerGe, a novel framework that introduces adversarial perturbations into text

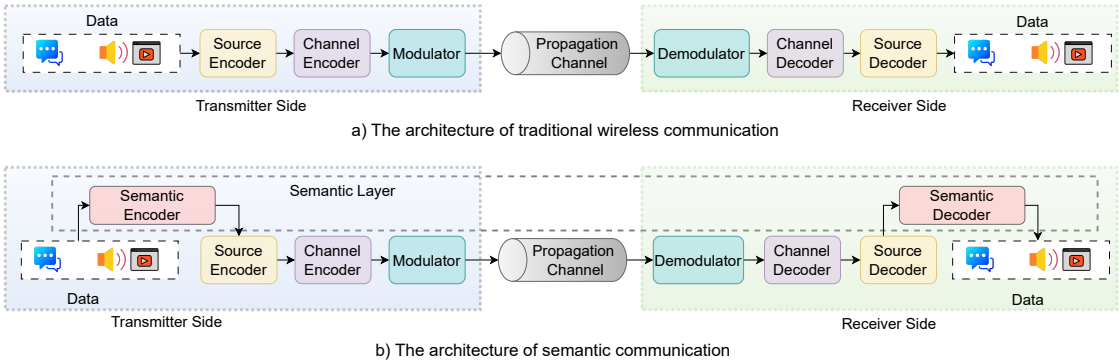


Figure 1: **Comparison of traditional wireless vs. semantic communication:** Semantic communication introduces a semantic encoder at the transmitter to extract task-relevant information and a semantic decoder at the receiver to reconstruct it.

data transmitted over semantic communication systems. Unlike prior text-based attacks that rely on misspellings or synonym substitution, SemPerGe identifies semantically significant tokens and perturbs them in a way that shifts meaning while preserving fluency and grammaticality. Leveraging off-the-shelf LLMs for adversarial text generation, SemPerGe enables effective deception of both transmitter and receiver in semantic communication channels, representing a robust approach to text-based semantic perturbation.

III. Threat Model

Semantic Communication. We consider a wireless semantic communication system consisting of a transmitter $\mathcal{T}$, receiver $\mathcal{R}$, and a communication channel, as shown in Fig. 1(b). $\mathcal{T} \in \mathcal{V}$, where $\mathcal{V}$ is the set of legitimate nodes, sends an input sequence $s = [w_1, w_2, \ldots, w_L]$ composed of L words.

The transmitter employs four key components: *semantic encoder*, *source encoder*, *channel encoder*, and *modulator*, sequentially transforming s into a modulated analog signal:

$$x = M_\delta(C_\alpha(Q_\gamma(S_\beta(s)))) \tag{1}$$

where, S_β, Q_γ, C_α, and M_δ denote the semantic, source, channel encoders, and modulator with respective parameters.

The receiver observes:

$$y = hx + n \tag{2}$$

where, h represents channel fading and n is additive noise. The receiver recovers the input via demodulation and decoding:

$$\hat{s} = S_\lambda^{-1}(Q_\zeta^{-1}(C_\mu^{-1}(M_\Gamma^{-1}(y)))) \tag{3}$$

where, $M_\Gamma^{-1}(y)$, demodulates the signal with parameters Γ, $C_\mu^{-1}(.)$ corrects channel errors, $Q^{-1}\zeta(.)$ decompresses the signal, and $S^{-1}\lambda(\cdot)$ restores the sequence's meaning.

The application at $\mathcal{R}$ is an LLM-enabled chatbot that processes $\hat{s}$ and generates responses for transmission back to the user.

Adversary's Goal. The adversary $\mathcal{A}$ aims to launch semantic perturbation attacks that subtly alter the meaning of a message without affecting its syntactic or grammatical correctness. The goal is to deceive the chatbot and the end-user in the considered application scenario.

Example Scenario. Suppose a user sends the query, "I am planning a trip to Chicago at night. What are the ten best things to do in Chicago at night?" The semantic encoder may condense this to "ten best things to do in Chicago at night." The adversary intercepts this message and alters "night" to "day," preserving grammar and structure but significantly shifting intent. The chatbot's response, aligned with "daytime" activities, is then reconstructed at the user's end as if answering the original "nighttime" query—creating a semantic inconsistency undetectable by bit-level integrity checks.

Such an attack is especially critical in safety-sensitive domains. For example, altering a medical chatbot query from "safe nighttime medications" to "daytime medications" could lead to hazardous advice.

Adversary's Knowledge and Capabilities. The adversary $\mathcal{A}$ is assumed to be external to the legitimate set $\mathcal{V}$ but within wireless range of both $\mathcal{T}$ and $\mathcal{R}$. $\mathcal{A}$ can: (i) Intercept and decode over-the-air signals using known physical-layer attack techniques, including eavesdropping, jamming, or side-channel leakage [6], [24]. (ii) Approximate channel parameters using DL-based modulation classification and channel estimation [25]. (iii) Inject perturbed semantic signals by exploiting timing information and retransmission intervals.

While $\mathcal{A}$ lacks access to internal parameters of the semantic encoder/decoder or LLM chatbot, they can infer message domains through intercepted traffic, enabling domain-specific semantic perturbations. This reflects a *grey-box* threat model, realistic for over-the-air attacks in wireless environments, where the adversary has knowledge of the wireless stack but limited access to application-layer DL models.

IV. SemPerGe: Semantic Perturbation Generator

The goal of the proposed semantic noise injection attack is to introduce subtle but targeted modifications to over-the-air (OTA) textual transmissions, such that the adversarial message preserves grammatical and contextual coherence while misleading downstream receiver-side applications (e.g., LLM-based chatbots). As illustrated in Fig.2, the adversary intercepts a transmission by launching a jamming attack, reconstructs the message, generates a semantically perturbed version using the proposed SemPerGe module, and transmits this adversar-

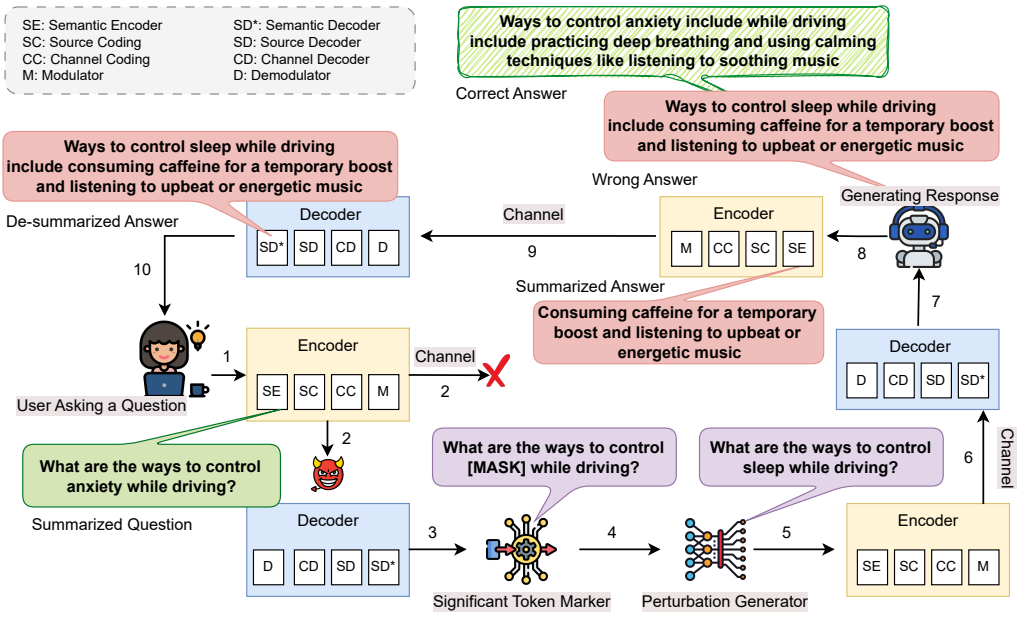


Figure 2: **Overview of the proposed attack:** User sends question to a chatbot, which is processed and transmitted. An adversary intercepts and halts this transmission with a jamming attack and extracts the data. Using SemPerGe, the adversary generates adversarial data which is transmitted to the chatbot before the user's original message is retransmitted. Chatbot responds to the adversarial question, and this response reaches the user after passing through various decoders, including a semantic decoder.

ial message before the original retransmission. The chatbot, unaware of the perturbation, responds based on the injected content, thereby subverting the original user-intended interaction. This section presents the design of SemPerGe, which lies at the core of this attack. SemPerGe operates in two phases: (1) identifying tokens that carry high semantic weight using attention-based analysis and NER, and (2) replacing these tokens using a fine-tuned generative model to create natural-sounding yet adversarial variants.

Given a reconstructed sequence s' obtained by the adversary from intercepted transmissions, SemPerGe aims to generate a modified version of s' that shifts its semantic intent while maintaining surface-level fluency. To do this, it first identifies semantically significant tokens in s' and then replaces these tokens in a way that maximally alters meaning without triggering syntactic or fluency violations. The two phases are:

Phase 1: Identifying Semantically Significant Tokens. This phase, termed the *Significant Token Marker*, seeks to isolate a subset of tokens in s' whose modification is likely to change the overall meaning of the sentence. Our design assumption is that the adversary does not have access to the exact models used at the receiver or transmitter. Instead, the adversary uses a transformer-based encoder model (pre-trained and publicly available) to analyze the text.

We extract token-level attention scores from all layers and heads of the encoder model. Each attention layer l contains multiple heads and yields an output tensor $O_l \in \mathbb{R}^{|J| \times d_{\text{model}}}$, where $|J|$ is the number of tokens in s' and d_{model} is the embedding dimension. We aggregate the attention weights per token as:

$$S'_{l,j} = \left(\sum_{\phi=1}^{d_{model}} O_{l,j,\phi} \right) \tag{4}$$

Collecting these scores across all L layers, we form the attention score matrix $\mathbf{S}' \in \mathbb{R}^{L \times |J|}$:

$$\mathbf{S}' = \begin{bmatrix} S'_{1,1} & \cdots & S'_{1,|J|} \\ \vdots & \ddots & \vdots \\ S'_{L,1} & \cdots & S'_{L,|J|} \end{bmatrix} \tag{5}$$

To quantify the importance of each token, we compute the frequency with which a token has the highest attention score across layers:

$$j_{max}(l) = \arg\max_j S'_l, \qquad c_j = \left(\sum_{l=1}^{L} \mathbf{1}_{[j_{max}(l)=j]} \right) \tag{6}$$

Normalizing these frequencies using a softmax operation yields the token significance vector $\lambda \in \mathbb{R}^{1 \times |J|}$:

$$\lambda = \text{Softmax}([c_1, c_2, \ldots, c_{|J|}]) \tag{7}$$

To further refine token importance based on task-specific salience, we incorporate an application-specific NER model. Tokens identified as named entities (e.g., names, locations, organizations) are boosted in their significance score within λ, while others are down-weighted. Finally, the top-ranked tokens in the adjusted λ are selected for perturbation and masked in s' using the [MASK] token.

Phase 2: Generating Adversarial Perturbations. The second phase, the *Perturbation Generator*, replaces masked tokens in s' with contextually coherent but semantically divergent

alternatives. A naive use of masked language models (MLMs) often results in either exact token recovery or substitution with close synonyms—behaviors that do not achieve our adversarial goal of semantic deviation.

Instead, we use a fine-tuned sequence-to-sequence (seq2seq) text generation model [26], trained on a custom attack-specific dataset, to predict adversarial substitutions. For each masked token m, we consider a candidate set $\mathcal{Y} = [m, y_1, \ldots, y_N]$ of possible replacements. We identify a synonym subset $\mathcal{Y}' \subseteq \mathcal{Y}$ using lexical similarity tools (e.g., WordNet and embedding-based cosine similarity), and ensure that the replacement token is drawn from $\mathcal{Y} \setminus (\mathcal{Y}' \cup m)$.

The generator is trained to optimize a balance between semantic shift and grammaticality, thereby producing adversarial messages that are both deceptive and natural-sounding. This distinguishes our method from standard paraphrasing techniques or simple synonym substitutions.

This multi-step pipeline ensures that injected messages (i) bypass syntactic anomaly detectors, (ii) exploit semantic vulnerabilities in downstream models (e.g., LLMs), and (iii) maintain realism, making them difficult to detect using standard methods. The next section will evaluate the effectiveness of SemPerGe across application-level tasks and threat models.

V. Evaluation

Experimental Setup. We conduct all experiments on an Ubuntu 18.04.6 system equipped with a single NVIDIA RTX *A*6000 GPU (48 GB VRAM). The proposed SemPerGe framework is implemented in PyTorch and uses additional libraries from Hugging Face for accessing pre-trained models and APIs. Unless stated otherwise, all experiments use a learning rate of 1×10^{-4}, batch size of 8, and 20 training epochs for fine-tuning, with early stopping triggered after 5 epochs without validation loss improvement.

Datasets. To evaluate the effectiveness of adversarial attacks in semantic communication, we consider both general-domain and specialized-domain question-answer datasets: WebQuestions [27], benchmark QA dataset with 6,642 question-answer pairs, focusing on factual questions involving named entities commonly queried online; and AI-Medical-Chatbot dataset [28], large-scale dataset (approx. 257,000 dialogues) featuring realistic doctor-patient conversations across diverse medical domains. It enables the evaluation of semantic vulnerabilities in safety-critical applications like healthcare.

These datasets are selected to assess the generalizability of SemPerGe across domains with different linguistic characteristics and sensitivity levels.

A. Semantic Perturbation Generation

Semantic Communication Setup. We implement a lightweight semantic communication model using GPT-4 [29] as both encoder and decoder. The encoder summarizes the question-answer pair to minimize transmission size, while the decoder reconstructs the content post-transmission. This aligns with recent efforts to reduce bandwidth usage in AI-driven communication systems.

Perturbation Target. We focus on perturbing questions (not answers), as these initiate the QA exchange. Attacking questions ensures stealthier manipulation of downstream answers while preserving dialogue structure.

1) Significant Token Masking: To identify influential tokens, we use dataset-specific pre-trained BERT models for attention extraction. Specifically, BERT-base-uncased [30] for WebQuestions dataset, while BioClinicalBERT [31] model is used for AI-Medical-Chatbot datasets. These BERT-based models, known for their robust, contextualized embeddings and effectiveness across diverse NLP tasks, are well-suited for reliably extracting attention scores.

Token-level significance scores are computed using Equations 5–7, incorporating both model attention and context relevance. We further enhance this by incorporating NER: (i) BERT-base-NER [32] for general entities (e.g., PERSON, LOCATION), and (ii) Medical-NER [33] for medical-specific entities (e.g., SIGN_SYMPTOM, DISEASE). This dual-layer scoring ensures perturbations target semantically impactful words, not just syntactically prominent ones. Tokens with the highest combined scores are masked for transformation.

2) Perturbation Generation: To generate plausible replacements for masked tokens, we fine-tune FLAN-T5 [34] on custom synthetic datasets created via ChatGPT. Each masked question is associated with three diverse rewordings that avoid simple synonym substitutions, to ensure a significant and notable shift in chatbot responses. We manually validate all outputs to ensure they introduce subtle semantic drift while maintaining grammatical and contextual fluency.

Fine-tuning is conducted separately for WebQuestions and AI-Medical-Chatbot, using a 70:20:10 (training, validation, and testing) split. The best-performing model for each dataset was selected based on the lowest observed validation loss. FLAN-T5 then generates adversarially perturbed questions from test inputs. We use GPT-4 and Medichat-Llama3-8B to generate corresponding answers for attack evaluation.

B. Evaluation Metrics

To assess the effectiveness of our generated adversarial perturbations, we employ four standard metrics: BLEU, ROUGE, grammatical correctness, and attack success rate, representing both linguistic similarity metrics and semantic robustness. In addition, we examine the text expansion rate to ensure compactness in adversarial questions, which is vital in semantic communication.

BLUE Score. The Bilingual Evaluation Understudy (BLEU) metric [35] measures how closely the generated adversarial question matches the original by evaluating the shared n-grams (consecutive words). Scores range from 0 to 1, with higher values indicating greater similarity. It helps verify that only selected tokens are altered, preserving the question's structure.

ROUGE Score. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [35] measures overlap with the original question, emphasizing recall rather than precision. Using ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum, we

assess matches at different n-gram scales and sequence levels, ensuring key elements are preserved while only selected tokens are altered.

Text Expansion Rate (TER). Captures character-length inflation in adversarial queries, calculated as the difference in the character count between the original and adversarial text. Lower TER is preferred to preserve transmission efficiency.

Grammatical Correctness Score (GCC). Scores the linguistic fluency of perturbed text using a pre-trained grammar checker [36], assigning a score between 0 and 1 based on grammatical integrity. High scores reduce the chance of human or automated detection.

Attack Success Rate (ASR). Measures the proportion of queries where perturbed questions cause a semantic change in chatbot answers. Semantic change is defined using cosine similarity of sentence embeddings (from Sentence-BERT [37]). Responses with similarity < 0.99 are marked as successful attacks. To avoid false positives from rephrased but equivalent responses, we set the temperature value of generative models to 0 to ensure determinism and also manually validate such edge cases. Figure 3 shows examples from the WebQuestions dataset, showcasing varying semantic scores obtained by threshold analysis.

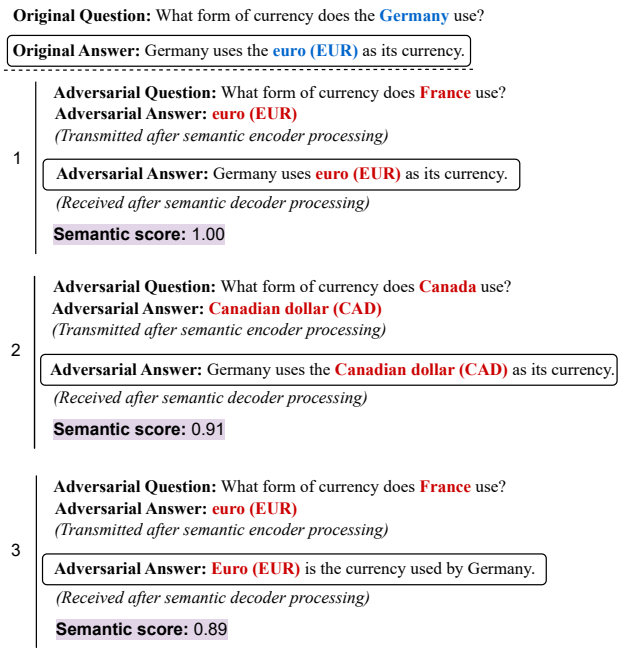


Figure 3: **Illustrative examples of semantic similarity thresholds.** (1) Failed attack (score = 1.0), (2) Successful attack (score < 0.99), and (3) False positive due to paraphrasing.

C. Baselines

We benchmark SemPerGe against four state-of-the-art black-box adversarial NLP attacks:

TextBugger [8]. This method perturbs characters or replaces words using GloVe neighbors.

SemAttack [21]. This method applies typos, GloVe-based replacements, and BERT-vocab neighbors.

Dataset: WebQuestions
BLEU: 0.807
ROUGE-1: 0.833, ROUGE-2: 0.800, ROUGE-L: 0.833, ROUGE-LSum: 0.833

Original: What document did **Thomas Jefferson** wrote?
Adversarial: What document did **Albert Einstein** write?

Dataset: AI-Medical-Chat
BLEU: 0.882
ROUGE-1: 0.900, ROUGE-2: 0.888, ROUGE-L: 0.900, ROUGE-LSum: 0.900

Original: I have **small** jaw bones, **scoliosis**, and swollen face. Please help.
Adversarial: I have **big** jaw bones, **fever**, and swollen face.

Figure 4: **SemPerGe output with lower ROUGE and BLEU scores:** Reductions in scores often result from correcting grammatical errors and omitting filler sentences.

TextGuise [22]. This approach adds emojis or dictionary meanings for semantic drift.

LLM-Attack [23]. LLM-Attack substitutes word with synonyms suggested by LLMs, selecting the alternative text that best aligns semantically with the original text.

These baselines iteratively remove words from the target text, evaluating their impact on target model to determine significance and select optimal perturbations. Assuming the adversary lacks access to target models (GPT-4 and Medichat-Llama3-8B), we employ GPT-2 for this task.

D. Attack Performance

Table I compares performance across all metrics for both datasets. While baseline methods score slightly higher on BLEU and ROUGE, largely due to their reliance on direct token substitutions, they often introduce noticeable artifacts such as unnatural phrasing or character-level noise. In contrast, SemPerGe uses generative rewriting to maintain naturalness and stealth. Despite a marginal drop in BLEU (e.g., 0.99 vs. 1.00), SemPerGe achieves higher GCC scores due to fluent rephrasings, lower TER, and a more compact input reformulation. This approach often introduces grammatically enhanced or contextually refined alternatives and omits redundant filler phrases, all while preserving the original semantic intent. Figure 4 illustrates examples where SemPerGe's outputs yield slightly lower ROUGE and BLEU scores—not due to semantic degradation, but because of deliberate grammatical improvements or removal of non-essential phrases (e.g., "wrote" improved, or filler phrases like "Please help" removed). These align with SemPerGe's goals of stealth and semantic quality.

SemPerGe generates shorter text: -1.52% (Web) and -2.09% (Medical) character reduction. In contrast, baselines generally increase length: TextBugger shows marginal changes, while SemAttack and LLM-Attack increase text size by 2–3%; TextGuise increases it drastically (107% Web, 59.18% Medical) by replacing words with verbose definitions.

Grammatical accuracy is high for SemPerGe (0.970 Web, 0.945 Medical), close to the original inputs and higher than most baselines. TextBugger and TextGuise degrade syntax quality significantly, while LLM-Attack maintains grammar but lacks stealth.

Attack Model	SemPerGe		TextBugger		TextGuise		SemAttack		LLM-Attack	
Dataset	Web	Medical	Web	Medical	Web	Medical	Web	Medical	Web	Medical
BLEU	**0.997**	**0.987**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ROUGE-1	**0.998**	**0.993**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ROUGE-2	**0.997**	**0.990**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ROUGE-L	**0.998**	**0.993**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ROUGE-Lsum	**0.998**	**0.993**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
*Text Expansion Rate	**-1.52%**	**-2.09%**	-0.36%	1.24%	107%	59.18%	2.70%	3.35%	1.37%	2.04%
*Grammatical Correctness	**0.970**	**0.945**	0.912	0.895	0.889	0.856	0.896	0.894	0.960	0.941
*Success Rate	**91.71%**	**86.19%**	5.42%	2.84%	13.25%	6.18%	1.60%	6.67%	2.41%	7.03%

Table I: **Performance comparison of SemPerGe and baselines on text-generation models:** 'Web' and 'Medical' denote WebQuestions and AI-Medical-Chatbot datasets. Metrics marked with * indicate those most related to the attack efficiency. Negative 'Text Expansion Rate' values signify text length reduction. Baselines show higher BLEU and ROUGE scores as they do not use LLMs.

Attack Model	**Text** (WebQuestions dataset)	**Text** (AI-Medical- Chatbot dataset)
Original	How deep is Lake Merritt Oakland?	How to increase my height? I am 23 years old.
SemPerGe	How deep is Lake Tahoe Nevada?	How to increase my weight? I am 23 years old.
TextBugger	How de ep is Lake Merritt Oakland?	How to increase my hei ght? I am 23 years old.
TextGuise	How :l deep :l is Lake Merritt Oakland?	How to increase my The vertical distance from the ground to the highest part of a standing person? I am 23 years old.
SemAttack	How sediment is Lake Merritt Oakland?	How to increase my tallness? I am 23 years old.
LLM-Attack	How abyssal is Lake Merritt Oakland?	How to increase my length? I am 23 years old.

Table II: **Adversarial Examples from SemPerGe and Baselines on WebQuestions nad AI-Medical-Chatbot Datasets:** SemPerGe successfully modifies the semantics of the question, leading to adversarial answers, whereas baselines fail to alter semantics using simple character-level, synonym and dictionary-based perturbations.

SemPerGe also achieves the highest ASR: 91.71% (Web) and 86.19% (Medical), far outperforming baselines (max 13.25% Web, 7.03% Medical). The primary reason for baseline models' lower ASR is their focus on deceiving text classifiers, rather than text generation models, typically achieved through minor grammatical errors or synonym replacements. To the best of our knowledge, there is no prior research focusing on generating text-based adversarial perturbations to deceive text generation models in the semantic communication scenario.

Examples of adversarial texts generated by SemPerGe and baseline methods on one sample of the WebQuestions and AI-Medical-Chatbot datasets are shown in Table II. In the Web example, it replaces "Merritt Oakland", the significant token, with "Tahoe Nevada," misleading the model. Baselines make minor changes (e.g., inserting emojis) that do not alter outputs. In the Medical case, SemPerGe replaces "height" with "weight," changing the clinical advice. However, the minimal changes generated by the baselines exhibit minimal to no changes in meaning.

Timing Analysis. In over-the-air attack scenarios, speed is critical. Adversaries must transmit modified content before the original is re-sent. Thus, the time to generate adversarial text directly impacts attack feasibility. Our timing measurements isolate generation time, excluding any communication delays.

Fig. 5 compares the average time required to generate adversarial text across different models on the datasets. On the WebQuestions dataset, SemPerGe requires approximately 0.15s per sample on average. In comparison, LLM-Attack, TextGuise, TextBugger, and SemAttack require 4.74s, 5.86s, 6.23s, and 129.50s, respectively. On the Medical dataset, SemPerGe also leads at 0.74s, while the baselines again show higher delays: 6.64s (LLM-Attack), 7.20s (TextGuise), 8.00s (TextBugger), and 115.31s (SemAttack). The baselines' longer times result from repeated target model queries to find perturbation candidates. SemAttack is especially slow due to exhaustive synonym searches across the entire vocabulary.

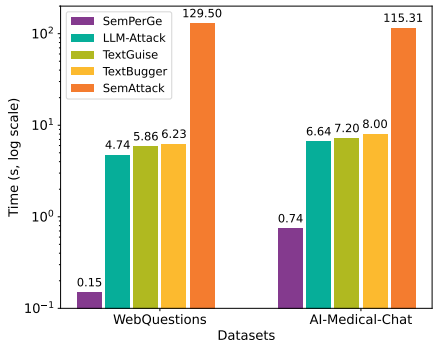


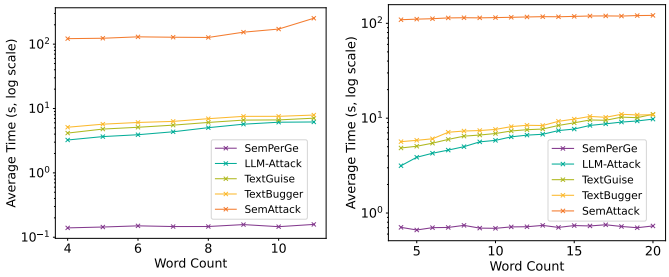
Figure 5: **Average adversarial text generation time comparison:** SemPerGe consistently requires less time for adversarial text generation compared to baseline methods.

Figures 6(a) and 6(b) show how generation time scales with input length. SemPerGe remains nearly constant regardless of text length, while baseline methods exhibit increasing latency, further highlighting SemPerGe's scalability.

VI. Further Analysis

A. Transferability

We evaluate SemPerGe's transferability beyond text generation by testing its effectiveness against text classifiers—a key area in adversarial NLP. This analysis targets sentiment classification using the Stanford Sentiment Treebank dataset (via Hugging Face [38]) and three widely-used classifiers: $model_1$ [39] (RoBERTa, fine-tuned on TweetEval [40]), $model_2$ [41] (an enhanced, robust variant of $model_1$), and $model_3$ [42] (BERT, trained on multilingual product reviews).



(a) WebQuestions dataset (b) AI-Medical-Chatbot dataset

Figure 6: **Perturbation generation time vs. word count:** Adversarial text generation times of SemPerGe show minimal variation across different text lengths.

To support this task, we adapt SemPerGe by replacing the NER module with a general-purpose part-of-speech (POS) tagger [43], focusing on adjectives, adverbs, verbs, and nouns—key carriers of sentiment. We retain the BERT-base-uncased model for token salience scoring and fine-tune a FLAN-T5 model on a synthetic dataset tailored for misleading sentiment classifiers. Unlike baselines, which require white-box access to the target model, our attack assumes a black-box setup. Thus, baselines use an alternative sentiment model [44] for guidance.

Table III shows that SemPerGe achieves the highest ASR, while maintaining BLEU and ROUGE scores comparable to baselines. It also introduces fewer character-level changes and maintains strong grammatical quality (0.845), close to the original dataset's average (0.851). In contrast, baselines exhibit lower ASRs and degraded fluency, reflecting their dependence on target model access for iterative optimization—an unrealistic assumption in black-box settings. A sample in Table IV illustrates that SemPerGe can flip sentiment labels while preserving syntactic fluency, whereas baselines often produce awkward or grammatically compromised outputs.

These results affirm SemPerGe's robustness and adaptability across NLP tasks, demonstrating strong performance even without access to the target model.

Metrics		SemPerGe	TextBugger	TextGuise	SemAttack	LLM-Attack
BLEU		**0.945**	1.00	1.00	1.00	1.00
ROUGE-1		**0.971**	1.00	1.00	1.00	1.00
ROUGE-2		**0.950**	1.00	1.00	1.00	1.00
ROUGE-L		**0.962**	1.00	1.00	1.00	1.00
ROUGE-LSum		**0.971**	1.00	1.00	1.00	1.00
*TER		**1.14%**	1.01%	9.11%	1.66%	1.81%
*GCC		**0.845**	0.814	0.822	0.815	0.821
*ASR	$model_1$ [39]	**73.09%**	40.65%	20.87%	46.15%	20.87%
	$model_2$ [41]	**64.32%**	29.67%	24.17%	36.26%	15.38%
	$model_3$ [42]	**61.81%**	35.14%	15.38%	35.27%	23.07%

Table III: **Comparison of attack performance on sentiment-classifiers using SemPerGe and baselines:** Metrics marked with * denote the metrics closely related to the attack efficiency. Baselines achieve higher BLEU and ROUGE scores due to their non-use of LLMs. $Model_1$, $model_2$, and $model_3$ are three considered sentiment classification models.

Attack Model	Text
Original	Time Warner's HD line up is crap.
SemPerGe	Time Warner's HD line up is good.
TextBugger	Time Warner's HD line up is cr ap.
TextGuise	Time Warner's HD line up is :l crap :l.
SemAttack	Time Warner's HD line up is bad.
LLM-Attack	Time Warner's HD line up is nonsense.

Table IV: **Adversarial Examples from SemPerGe and Baselines on Sentiment Dataset:** SemPerGe modifies the sentiment of the text leading to misclassification, while baselines fail to change the sentiment through simple character-level, synonym and emoji-based perturbations.

Metrics	Part 1	Part 2
Within Context	95%	-
Grammatically Correct	-	92.5%
Semantically Correct	-	100%

Table V: **User study results:** 95% of participants rated adversarial answers are within the context of the original questions, verifying attack stealthiness. 92.5% and 100% of participants rated the adversarial questions as grammatically and semantically accurate, respectively.

B. *User Study*

We conducted a human evaluation with 40 participants to assess the stealth and linguistic quality of adversarial questions generated by SemPerGe. The study, administered via Google Forms, was divided into two parts. In the first part, participants were presented 10 benign questions paired with adversarial answers (from WebQuestions and AI-Medical-Chatbot datasets) and asked whether the answers appeared contextually appropriate. This measured attack stealth—i.e., whether perturbations altered the original question's meaning in a detectable way. In the second section, evaluated 20 adversarially generated questions from the same datasets for grammatical and semantic correctness. This assessed whether the adversarial inputs preserved natural language fluency—important for remaining undetected by human or automated scrutiny.

As shown in Table V, 95% of participants found adversarial answers contextually aligned with their respective questions, suggesting strong stealth characteristics. In Part 2, 100% of participants rated adversarial questions as semantically accurate, and 92.5% judged them grammatically correct. The slight dip in grammatical ratings is attributed to preexisting grammatical issues in the original questions (see Section V-D).

VII. Conclusion

In this paper, we introduced SemPerGe, a novel framework for text-based adversarial attacks targeting black-box text-generation models, particularly question-answering systems. SemPerGe operates in two phases: identifying semantically significant tokens and perturbing them to induce adversarial semantic shifts while preserving grammatical and contextual coherence. Extensive experiments show that SemPerGe achieves high attack success rates and strong linguistic fidelity, outperforming state-of-the-art baselines. A user study further validated the stealth and fluency of generated adversarial in-

puts. Beyond QA, we demonstrated SemPerGe's transferability by successfully applying it to sentiment classification tasks, highlighting its adaptability across NLP domains. Future work will explore extending SemPerGe to other generative tasks such as text summarization, broadening its applicability and further investigating its impact in diverse adversarial settings.

REFERENCES

[1] P. Agbaje, A. Anjum, A. Mitra, E. Oseghale, G. Bloom, and H. Olufowobi, "Survey of interoperability challenges in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 838–22 861, 2022.

[2] Z. Yang, M. Chen, G. Li, Y. Yang, and Z. Zhang, "Secure semantic communications: Fundamentals and challenges," *IEEE Network*, 2024.

[3] Y. Liu, X. Wang, Z. Ning, M. Zhou, L. Guo, and B. Jedari, "A survey on semantic communications: technologies, solutions, applications and challenges," *Digital Communications and Networks*, 2023.

[4] A. Bahramali, M. Nasr, A. Houmansadr, D. Goeckel, and D. Towsley, "Robust adversarial attacks against dnn-based wireless communication systems," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 126–140.

[5] Q. Hu, G. Zhang, Z. Qin, Y. Cai, G. Yu, and G. Y. Li, "Robust semantic communications against semantic noise," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022.

[6] L. Lu, M. Chen, J. Yu, Z. Ba, F. Lin, J. Han, Y. Zhu, and K. Ren, "An imperceptible eavesdropping attack on wifi sensing systems," *IEEE/ACM Transactions on Networking*, 2024.

[7] Z. Li, J. Zhou, G. Nan, Z. Li, Q. Cui, and X. Tao, "Sembat: Physical layer black-box adversarial attacks for deep learning-based semantic communication systems," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[8] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] DeepLearning.AI, "A complete guide to natural language processing," 2023. [Online]. Available: https://www.deeplearning.ai/resources/natural-language-processing/

[11] G. Giacaglia, "How transformers work," *Toward Data Science*, 2019. [Online]. Available: https://towardsdatascience.com/transformers-141e32e69591

[12] S. More, "Transformer architecture , transformer model types and its use-cases," *Medium*, 2023. [Online]. Available: https://medium.com/@sandyonmars/transformer-architecture-transformer-model-types-and-its-use-cases-fb2afb89683c

[13] L. Bansal, "Transformer — attention is all you need easily explained with illustrations," *Medium*, 2021. [Online]. Available: https://luv-bansal.medium.com/transformer-attention-is-all-you-need-easily-explained-with-illustrations-d38fdb06d7db

[14] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[15] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, "Semantic communication with adaptive universal transformer," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 453–457, 2021.

[16] S. More, "Digital communication - quick guide," *Tutorialspoint*. [Online]. Available: https://www.tutorialspoint.com/digital_communication/digital_communication_quick_guide.html

[17] A. Anjum, M. E. Eren, I. Boureima, B. Alexandrov, and M. Bhattarai, "Tensor train low-rank approximation (tt-lora): Democratizing ai with accelerated llms," *arXiv preprint arXiv:2408.01008*, 2024.

[18] M. Shen, J. Wang, H. Du, D. Niyato, X. Tang, J. Kang, Y. Ding, and L. Zhu, "Secure semantic communications: Challenges, approaches, and opportunities," *IEEE Network*, 2023.

[19] Y. E. Sagduyu, T. Erpek, S. Ulukus, and A. Yener, "Is semantic communication secure? a tale of multi-domain adversarial attacks," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 50–55, 2023.

[20] Z. Li, X. Liu, G. Nan, J. Zhou, X. Lyu, Q. Cui, and X. Tao, "Boosting physical layer black-box attacks with semantic adversaries in semantic communications," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 5614–5619.

[21] B. Wang, C. Xu, X. Liu, Y. Cheng, and B. Li, "Semattack: Natural textual attacks via different semantic spaces," *arXiv preprint arXiv:2205.01287*, 2022.

[22] G. Chang, H. Gao, Z. Yao, and H. Xiong, "Textguise: Adaptive adversarial example attacks on text classification model," *Neurocomputing*, vol. 529, pp. 190–203, 2023.

[23] Z. Wang, W. Wang, Q. Chen, Q. Wang, and A. Nguyen, "Generating valid and natural adversarial examples with large language models," in *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2024, pp. 1716–1721.

[24] M. Alyami, I. Alharbi, C. Zou, Y. Solihin, and K. Ackerman, "Wifi-based iot devices profiling attack based on eavesdropping of encrypted wifi traffic," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 385–392.

[25] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[26] S. S. Hebbar, "Text generation v/s text2text generation," *Medium*, 2023. [Online]. Available: https://medium.com/@sharathhebbar24/text-generation-v-s-text2text-generation-3a2b235ac19b

[27] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013.

[28] R. M. Vsevolodovna, "ai-medical-chatbot," *HuggingFace*, 2024. [Online]. Available: https://huggingface.co/datasets/ruslanmv/ai-medical-chatbot

[29] OpenAI, "Gpt-4 is openai's most advanced system, producing safer and more useful responses," *OpenAI*, 2023. [Online]. Available: https://openai.com/index/gpt-4/

[30] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1. Minneapolis, Minnesota, 2019.

[31] E. Alsentzer, "Bio_clinicalbert," *HuggingFace*, 2020. [Online]. Available: https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT

[32] D. S. Lim, "bert-base-ner," *HuggingFace*, 2020. [Online]. Available: https://huggingface.co/dslim/bert-base-NER/tree/main

[33] S. M, "Medical-ner," *HuggingFace*, 2024. [Online]. Available: https://huggingface.co/blaze999/Medical-NER/tree/main

[34] Google, "google/flan-t5-base," *Hugging Face*, 2022. [Online]. Available: https://huggingface.co/google/flan-t5-base/tree/main

[35] Q. Herreros, T. Veasey, and T. Papaoikonomou, "Rag evaluation metrics: A journey through metrics," *Elastic Search labs*, 2023. [Online]. Available: https://www.elastic.co/search-labs/blog/evaluating-rag-metrics#n-gram-metrics

[36] X. Yang, "yang-grammer-check," *HuggingFace*, 2024. [Online]. Available: https://huggingface.co/xy4286/yang-grammer-check/tree/main

[37] S. Seelam, "Machine learning fundamentals: Cosine similarity and cosine distance," *Medium*, 2021. [Online]. Available: https://medium.com/geekculture/cosine-similarity-and-cosine-distance-48eed889a5c4

[38] S. NLP, "stanfordnlp/sentiment140," *Hugging Face*, 2022. [Online]. Available: https://huggingface.co/datasets/stanfordnlp/sentiment140/tree/main

[39] C. NLP, "cardiffnlp/twitter-roberta-base-sentiment," *Hugging Face*, 2020. [Online]. Available: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

[40] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "Tweeteval: Unified benchmark and comparative evaluation for tweet classification," *arXiv preprint arXiv:2010.12421*, 2020.

[41] C. NLP, "cardiffnlp/twitter-roberta-base-sentiment-latest," *Hugging Face*, 2022. [Online]. Available: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

[42] N. Town, "nlptown/bert-base-multilingual-uncased-sentiment," *Hugging Face*, 2020. [Online]. Available: https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

[43] Flair, "flair/upos-multi," *Hugging Face*, 2022. [Online]. Available: https://huggingface.co/flair/upos-multi

[44] C. NLP, "cardiffnlp/twitter-xlm-roberta-base-sentiment-multilingual," *Hugging Face*, 2022. [Online]. Available: https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment-multilingual