Promising Multi-Granularity Linguistic Steganography by Jointing Syntactic and Lexical Manipulations

Chengfu Ou, Lingyun Xiang*, Yangfan Liu

School of Computer and Communication Engineering, Changsha University of Science and Technology
hahally@stu.csust.edu.cn, xiangly@csust.edu.cn, flyvan@stu.csust.edu.cn

Abstract

Existing modification-based linguistic steganography methods primarily perform linguistic manipulations within a single embedding space to conceal secret information. However, these methods are stringently constrained by the original semantics of the cover text, making it struggle to achieve a satisfactory embedding capacity in a single embedding space. In this paper, we propose a novel Multi-granularity Modification-based Linguistic Steganography framework (MMLS) that hides secret information in both syntactic space and symbolic space, enhancing syntactic naturalness and semantic coherence while further increasing embedding capacity. Specifically, MMLS utilizes a paraphrase generation model to automatically modify the syntactic structure of the given original sentence, which enables the generation of paraphrases and the preservation of semantics simultaneously. Moreover, MMLS employs a distance-aware syntactic bins coding strategy to embed part of secret information into the syntactic space. This strategy utilizes a cluster-based way to partition the implicit syntactic space into a finite number of separate zones, thus increasing the number of candidate paraphrases and avoiding the selection of semantically distorted steganographic texts. Finally, the pre-trained BERT is used to replace some words in candidate paraphrases with their synonyms. Such a design embeds the remaining secret information into symbolic space while ensuring syntactic and semantic naturalness. Experimental results demonstrate that MMLS significantly outperforms existing methods in terms of semantic coherence, embedding capacity, and security.

Code — https://github.com/hahally/MMLS/

Introduction

Steganography is recognized as the art and science of embedding secret information within a public multimedia carrier without arousing suspicion from supervisors (Simmons 1984). Theoretically, any public multimedia carriers with redundant space, such as images (Zhou et al. 2023), audios (Wu et al. 2020), videos (Fan, Zhang, and Zhao 2022), texts (Krishnan, Thandra, and Baba 2017), etc., can be utilized to conceal secret information for communication. Among them, texts are the most common ones on today's social media networks, due to their simplicity and efficiency in conveying information. Therefore, linguistic steganography (Xiang et al. 2022), which employs texts as the carriers to conceal and transmit secret information, has garnered increasing attention in recent years. Modification-based linguistic steganography (MLS), as a significant branch of linguistic steganography, primarily conceals secret information into a given natural text by subtly modifying its content using specific transformations, which helps preserve the original semantic meaning, such as synonym substitutions(Chang and Clark 2010b, 2014; Huo and Xiao 2016; Xiang et al. 2018), paraphrasing (Chang and Clark 2010a; Wilson and Ker 2016), machine translation (Stutsman et al. 2006; Meng et al. 2011), word order adjustment (Chang and Clark 2012), and syntactic analysis (Murphy and Vogel 2007). Nevertheless, the inherent limitations posed by limited conversion rules and original semantic constraints result in a deficient embedding capacity of MLS. Moreover, inappropriate substitution operations are more likely to result in steganographic texts with syntactic unnaturalness and semantic inconsistencies (Grosvald and Orgun 2011), making them susceptible to be detected by linguistic steganalysis methods (Wen et al. 2019; Yang et al. 2019; Peng et al. 2021).

To tackle the aforementioned challenges, the latest works, such as substitution-based (Ueoka, Murawaki, and Kurohashi 2021; Xiang et al. 2023), paraphrasing-based(Yang et al. 2024) and syntactic transformation-based (Xiang, Ou, and Zeng 2024), strive to leverage the promising language models to enhance the diversity of linguistic transformations and move away from reliance on specific rules, thereby improving the performance of MLS. However, it is worth noting that these language models-based (LMs-based) methods are constrained by the original semantics of the cover text to perform minor linguistic transformation manipulations within a single redundant embedding space (symbolic space or syntactic space), thereby enhancing semantic consistency but limiting the embedding capacity. Therefore, LMs-based MLS gains less attention in improving embedding capacity.

A promising way is to combine syntactic and lexical linguistic manipulations to improve embedding capacity as shown in Figure 1. The syntactic transformation-based method first performs syntactic linguistic manipulations to change the syntactic structure of the cover text by the syntax-controlled paraphrase generation model. Subsequently, the

*Corresponding author: xiangly@csust.edu.cn

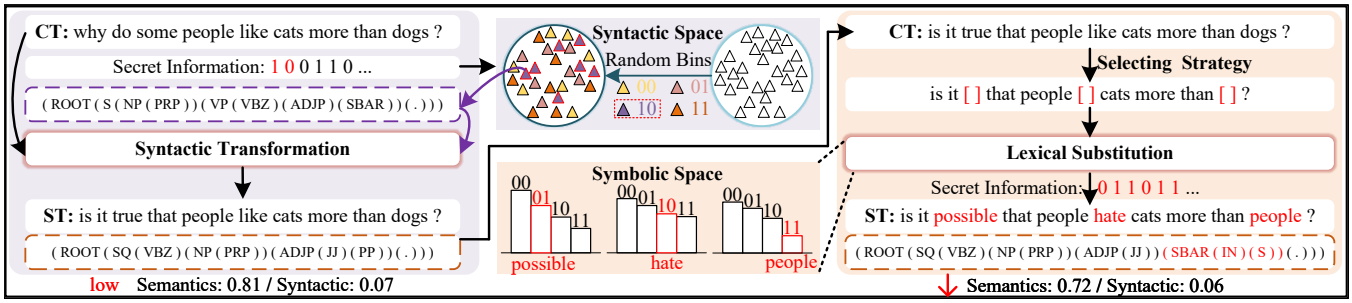

Figure 1: The overall sketch that directly joints syntactic transformation-based (left) and substitution-based (right) methods. "CT" and "ST" denote cover text and steganographic text, respectively. "Semantics" and "Syntactic" represent the semantic similarity with the cover text and the syntactic similarity with the target template, respectively.

substitution-based method replaces the selected words in the input text with candidate words from symbolic space. Nevertheless, due to the particularity of the linguistic steganography task, directly jointing syntactic transformation-based and substitution-based methods will encounter some potential conflicts. For example, as shown in Figure 1, the syntactic transformation-based method can induce semantic deviation or syntactic errors in the generated text. Moreover, lexical linguistic manipulations further cause global semantic and minor syntactic changes, resulting in the steganographic text with syntactic unnaturalness and semantic deviation.

To overcome the limitations mentioned above, we propose a novel Multi-Granularity Modification-based Linguistic Steganography (MMLS) framework, which incorporates the advantages of syntactic transformation-based methods and substitution-based methods. More specifically, we introduce a syntax-controlled paraphrase generator by employing abundant syntactic structure information as a supplement to modify the expression forms with slight semantic changes in sentences. Differing from the previous method (Xiang, Ou, and Zeng 2024), which randomly and evenly divides an extensive collection of syntactic templates into a finite number of subsets, i.e., bins, this paper proposes a distance-aware syntactic bins coding strategy to refine the partitioning for subsets. Meanwhile, a pre-trained BERT is utilized to replace some words with their context-related synonyms by an adaptive autoregressive coding strategy. Moreover, we perform consistency checking to ensure that the steganographic text has low semantic distortion and satisfies the target syntactic template. Our major contributions are as follows:

- To our knowledge, it is the first modification-based linguistic steganography work that performs linguistic manipulations in multi-granularity spaces to embed secret information, attaining a substantial embedding capacity.
- MMLS leverages a novel distance-aware syntactic bins coding strategy to efficiently mitigate the inherent interference introduced by randomness in previous work, thereby further enhancing semantic coherence.
- MMLS utilizes a consistency checking unit on generated texts in terms of syntactic and semantics to ensure that secret information is successfully embedded and extracted.
- Extensive experimental results show that MMLS not only significantly improves semantic coherence and security, but also achieves a satisfactory embedding capacity compared to baseline methods.

Related Work

MLS primarily conceals secret information into a given natural text by subtly modifying its content using specific linguistic transformations while preserving the original semantic meaning. Initially, most researchers focus on constructing the painstaking transformation rules to control the intensity of modifications to the cover texts.

Synonym substitutions Chang and Clark (2014) propose to develop a novel linguistic steganography method based on context synonym substitution and vertex coding algorithm. Moreover, It uses the Google n-gram corpus for checking the applicability of a synonym in context. Huo and Xiao (2016) propose a linguistic steganography method based on vector distance of two-gram dependency collocations to eliminate the obvious mistakes and logical misconceptions resulting from the inaccuracy of candidate synonyms. Xiang et al. (2018) combine arithmetic coding and synonym substitutions to quantize synonyms employed for carrying payload into an unbalanced and redundant binary sequence, which is compressed by adaptive binary arithmetic coding losslessly to provide a spare for accommodating additional data.

Paraphrasing Chang and Clark (2010a) propose to hide information in a cover text by using a large paraphrase dictionary, which consists of abundant painstaking paraphrase rules, and use the Google n-gram corpus and a CCG parser to certify the paraphrasing grammaticality and fluency. Wilson and Ker (2016) amplify the extent of substitution transformations from mere synonymous words to more semantically similar phrases by employing paraphrasing rules and use distortion measures to automatically produce the best embedding steganographic text.

Syntactic transformation Murphy and Vogel (2007) perform a set of automated and reversible syntactic transforms that can hide information without changing the meaning or style of a text, achieving a success rate of 96% and bandwidth of 0.3 bits per sentence. Chang and Clark (2012) propose to use word order adjustment as the linguistic trans-

formation to change the expression of the cover text. Meanwhile, they leverage a maximum entropy classifier to determine the naturalness of sentence permutations and select the promising expression to embed secret information.

Language models With advances in deep learning and natural language processing, much attention has been paid to improving the performance of MLS by language models. For example, Ueoka, Murawaki, and Kurohashi (2021) propose a novel substitution-based method that uses pre-trained model BERT to predict the candidate words at masked positions according to its context and improve the embedding capacity of MLS. Moreover, this method utilizes a sliding window masking strategy and skips the inappropriate subwords to avoid distorting the masking positions. Xiang et al. (2023) proposes a causal perception-guided linguistic steganography by elaborate and secure lexical substitutions using BERT. Yang et al. (2024) propose a novel MLS method based on an information encoding strategy, which utilizes a pivot translation-based paraphrasing and semantic-aware bins coding to change the expression of a given text. Xiang, Ou, and Zeng (2024) propose a novel sentence-level linguistic steganography framework, which employs abundant structural information as constraints to guide the syntax-controlled paraphrase generation model to modify expression forms of the cover text.

Proposed Method

Overall Architecture

As illustrated in Figure 2, the proposed MMLS framework includes three processing units, i.e., *syntactic transformation unit*, *lexical substitution unit*, and *consistency checking unit*. Concretely, the syntactic transformation unit is utilized to automatically modify the syntactic structure of the given cover text and generate paraphrases (defined as intermediate steganographic texts) by the syntax-controlled paraphrase generator. As a result, some secret information is embedded into syntactic space while preserving the semantics unchanged. Subsequently, the lexical substitution unit is employed to replace certain carefully selected words from an intermediate steganographic text by a pre-trained BERT, thereby hiding the remaining secret information in symbolic space. To ensure semantic coherence and the correct extraction of secret information, the consistency checking unit is adapted to filter candidate items from the lexical substitution unit to avoid text with illegal syntactic and semantics.

Syntactic Transformation Unit

Figure 2 shows the overall sketch of the syntactic transformation unit. There are two main components: the syntax-controlled paraphrase generator and the distance-aware syntactic bins coding strategy, that play crucial roles in this workflow. As shown in Figure 2, Firstly, secret information is mapped into a discrete syntactic space by distance-aware syntactic bins coding strategy. Then the corresponding syntactic template is sampled from selected zones in syntactic space. Finally, the cover text and sampled syntactic template are fed to the syntax-controlled paraphrase generator to yield a steganographic text, which is expected to satisfy any syntactic template in selected zones while having the same meaning as the cover text.

Syntax-Controlled Paraphrase Generator To eliminate the reliance on a special set of manually designed rules in traditional MLS methods, we propose to leverage a syntax-controlled paraphrase generator to generate paraphrases with various syntactic structures, improving the quality and diversity of sentence transformations. As shown in Figure 2, the syntactic template, which guides the generation of different expressions, is represented as a partial constituency parse tree with H layers. The syntactic template with larger H can provide more detailed syntax structure information, which may lead to a better quality paraphrase.

In this paper, we use the architecture of the SI-SCP model (Yang et al. 2022) as the implementation of our syntax-controlled paraphrase generator. The syntax-controlled paraphrase generator mainly consists of three parts that are semantic encoder, syntactic encoder, and sentence decoder. Their structures are the same as the transformer architecture (Vaswani et al. 2017), where the core function is primarily a multi-head attention mechanism. Formally, we can represent the above process of modeling for the syntax-controlled paraphrase generator as follows:

$$\begin{cases} H_x = SemEncoder(x) \\ H_t = SynEncoder(t) \\ y = Decoder(H_t, H_x) \end{cases}, \tag{1}$$

where $SemEncoder$, $SynEncoder$ and $Decoder$ represent semantic encoder, syntactic encoder and sentence decoder, respectively. y is denoted as the generated paraphrase, which is semantically consistent with x while adhering to the syntactic structure prescribed by t.

Distance-Aware Syntactic Bins Coding Strategy To mitigate the inherent interference introduced by randomness in previous work (Xiang, Ou, and Zeng 2024), we propose a novel distance-aware syntactic bins coding (DBC) strategy to partition syntactic templates into disjoint subsets by spectral clustering [1] (Ng, Jordan, and Weiss 2001).

Without the loss of generalization, let $T = \{t_i\}_{i=1}^{m}$ represent the syntactic template set, i.e., syntactic space, which includes m non-repeating syntactic templates. In this paper, we leverage the Stanford CoreNLP toolkit (Manning et al. 2014) to extract the syntactic template from each sentence. Given a pre-determined integer k representing the bits of secret information embedded within each syntactic template, the binary coding states can be determined as $B = \{b_i\}_{i=1}^{2^k}$, where $b_i \in \{0,1\}^k$. We can construct a mapping function $\mathcal{F}$ to establish a close connection between the syntactic space and the binary code space, which maps each element in T to a unique binary code in B. To this end, we define the disjoint subsets as $bins = \{bin_i\}_{i=1}^{2^k}$ divided from T, where $T = \bigcup_{i=1}^{2^k} bin_i$ and $bin_i \bigcap bin_j = \emptyset, \forall\, 1 \le i \ne j \le 2^k$. For each $i \in [1, 2^k]$, $\mathcal{F}$ maps syntactic templates in bin_i to the same binary code b_i corresponding to indices i, i.e.,

$$\forall\, t_{i,*} \in bin_i, \mathcal{F}(t_{i,*}) = b_i, \tag{2}$$

[1] Implemented with Sklearn package. (https://scikit-learn.org)

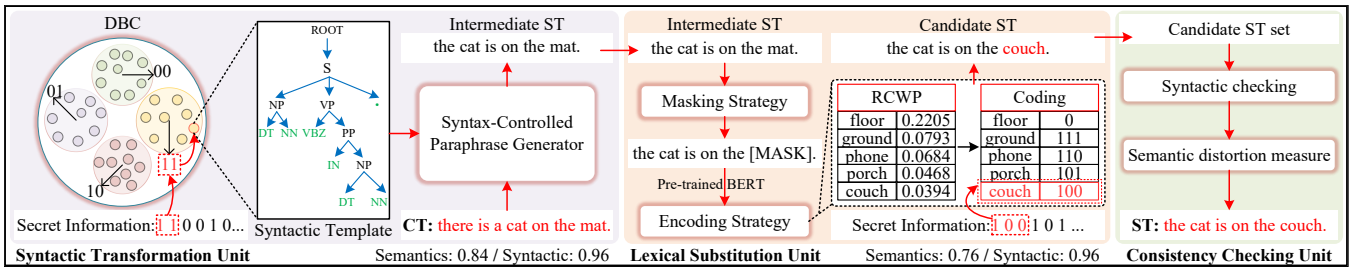

Figure 2: The overall sketch of the proposed multi-granularity modification-based linguistic steganography framework.

where b_i represents the binary code assigned to subset bin_i and $t_{i,*}$ is any syntactic templates in bin_i. For example, all syntactic templates in bin_4 are mapped to $b_4 =$ "11" if $k = 2$, i.e, $\mathcal{F}(e) =$ "11" if $e \in bin_4$. Obviously, we can encode and decode a given sentence as a binary code by identifying its syntactic template.

It is noted that our focus is on how to construct disjoint subsets $bins$ (i.e., $\mathcal{F}$). There is an underlying assumption: when a syntactic template guides a syntax-controlled paraphrase generator to perform syntactic transformations on the given sentence, the syntactic template of the generated sentence should be consistent with or similar to the cover text. Under this assumption, the core idea of DBC is to group similar syntactic templates into the same subset by spectral clustering. To this end, we establish an undirected weighted graph $G = (T, S)$, where nodes correspond to syntactic templates set T and edges S are typically represented by a weighted adjacency matrix that reveals the similarities between syntactic templates. In this work, we mainly explore the construction of a similarity matrix based on the distance between syntactic templates, resulting in disjoint subsets $bins$. We propose three strategies (DBC-LD, DBC-TED, and DBC-CD) based on three different distance measures, i.e., Levenshtein Distance, Tree-Edit Distance (Zhang and Shasha 1989) and Cosine Distance. Our proposed MMLS uses the DBC-CD strategy in syntactic transformation unit.

(1) DBC-LD first linearizes the syntactic template for tree structure as a string sequence. For example, the syntactic template in Figure 2 can be linearized as "(ROOT(S(NP(DT NN)VP(VDZ PP(IN NP(DT NN))).)))". Afterward, calculating the Levenshtein distance between any two syntactic templates becomes straightforward. Levenshtein distance is the minimum number of edit operations (i.e., substitution, deletion, and insertion) required to convert one string into another, typically solved using dynamic programming. To accommodate edge weights in spectral clustering, we convert the distance matrix into a similarity matrix as follows:

$$S_{i,j} = 1 - \frac{LD_{i,j}}{max(L_{t_i}, L_{t_j})}, \tag{3}$$

where $S_{i,j}$ is the edge weight between syntactic templates t_i and t_j and $LD_{i,j}$ is the Levenshtein distance between t_i and t_j. L_{t_i} and L_{t_j} represent the sequence lengths of t_i and t_j.

(2) DBC-TED can directly handle syntactic templates without linearization, preserving more information about the skeleton structure. TED is a measure of similarity between two tree-structured data objects, which is defined as the minimum number of operations required to transform one tree into another. These operations typically include node insertion, node deletion, and node substitution. Therefore, DBC-TED accounts for not only the labels of the nodes but also the entire subtree structures, ensuring a comprehensive comparison. Similarly, after obtaining the distance matrix calculated according to TED, it is transformed into a similarity matrix as follows:

$$S_{i,j} = 1 - \frac{TED_{i,j}}{max(L_{t_i}, L_{t_j})}, \tag{4}$$

where $TED_{i,j}$ is the tree-edit distance between t_i and t_j. L_{t_i} and L_{t_j} represent the number of nodes for t_i and t_j.

(3) DBC-CD, compared to the above two strategies, is no longer limited to the shallow structural similarity of syntactic templates but rather focuses on the distance relations of syntactic templates on the latent feature space. Syntactic templates can be mapped as continuous feature representations with a fixed dimension in latent space by the syntactic encoder of the syntax-controlled paraphrase generator according to Eq. (1). As a result, averaging the feature vectors of all nodes can accommodate syntactic templates with any number of nodes. The cosine similarity score between syntactic templates t_i and t_j is calculated as follows:

$$S_{i,j} = \frac{Z_{t_i} \cdot Z_{t_j}}{\|Z_{t_i}\| \|Z_{t_j}\|}, \tag{5}$$

where Z_{t_i} and Z_{t_j} are the syntactic feature representations of t_i and t_j, respectively.

Suppose the current embedded k bits secret information is $b =$ "11" as shown in Figure 2, all syntactic templates in subset bin_4 are selected to guide the transformations of cover text, resulting in sufficient intermediate steganographic texts $\mathbb{O} = \{O_i\}_{i=1}^{|bin_4|}\}$ provided to find a promising one to replace cover text x for embedding information b. Note that if $\mathcal{F}(T_x) = b$, where T_x is the syntactic template of x, it indicates that cover text x can successfully conceal secret information b. In this case, cover text x will be added to the intermediate steganographic text set $\mathbb{O}$.

Lexical Substitution Unit

Figure 2 presents the overall sketch of the lexical substitution unit. In general, given a text where some tokens are replaced with the special token [MASK], BERT is trained to

recover the original tokens based only on their context. In this work, we first determine the replaceable positions in a given text by a *masking strategy* and then leverage the pre-trained BERT to establish an appropriate replaceable candidate word pool (RCWP) for each target position according to its context. Finally, we utilize the *encoding strategy* to embed secret information by selecting words from RCWP.

Masking Strategy Concretely, let the intermediate steganographic text $O = \{w_i\}_{i=1}^{n}$ with n words. To ensure that secret information can be extracted correctly, the data sender and receiver must mask the same position. In this paper, we try to perform a simple and effective masking strategy. Following (Ueoka, Murawaki, and Kurohashi 2021) , we set a step s and skip "anchor" words (such as punctuation, number, non-initial subwords, and stopwords) to control the positions of [MASK]. Mathematically, the masking strategy can be defined as:

$$M(w_i) = \begin{cases} [\text{MASK}], & \text{if } i\%s = 0 \text{ and } w_i \notin Q \\ w_i, & otherwise \end{cases}, \quad (6)$$

where $M(\cdot)$ represents the masking function and Q denotes as the "anchor" words set. w_i is the i-th word in O. Significantly, modifying "anchor" words is a very dangerous move that may lead to awkward expressions, arousing the supervisor's suspicion. According to Eq. (6), a new text O_{mask} with [MASK] can be inputted to BERT, which provides a probability distribution for each masked position based on its suitability within the given context.

Encoding Strategy Subsequently, an information encoding strategy is employed to choose a word as the output for the present masked position, whose probability is proportional to its prediction probability obtained by BERT. To avoid contextual semantic conflicts, we utilize an "autoregressive" strategy to predict words for each masked position from left to right under the control of secret information, enhancing the semantic coherence between contexts.

Formally, for the first masked position in O_{mask}, let $P = \{p_i\}_{i=1}^{|V|}$ denote the predicted probability distribution in descending order for words in the vocabulary $V = \{v_i\}_{i=1}^{|V|}$. To avoid selecting inappropriate words with low probability, we set a threshold τ to truncate the probability distribution, thereby obtaining the replaceable candidate word pool RCWP, which satisfies the following criteria:

$$\forall v_i \in \text{RCWP}, \frac{p_i}{p_1} \geq \tau, \quad (7)$$

where v_i is the i-th word in RCWP and p_i is the probability for v_i. The initial RCWP contains the word v_1 with the highest probability p_1. To perform Huffman coding to map each candidate word in RCWP to binary code, we normalize the corresponding prediction probabilities for RCWP to construct a Huffman tree, where each leaf node represents a candidate word assigned the unique binary code.

The corresponding leaf node (i.e., candidate word) is selected as the output of the current masked position according to the secret information that needs to be hidden. Then, O_{mask} is updated by replacing the first [MASK] with the determined candidate word that carries secret information. Perform the above operation for each of the next masked positions in turn to obtain a candidate steganographic text O'. Therefore, the candidate steganographic text set $\mathbb{O}' = \{O'_i\}_{i=1}^{|bin_4|}$ can be produced from intermediate steganographic text set $\mathbb{O}$ by lexical substitution unit.

Consistency Checking Unit

After undergoing syntactic transformation and lexical substitution, we need to perform the consistency checking unit to ensure that the secret information is successfully embedded. Especially, improper syntactic transformations can result in the generated paraphrases not conforming to the expected syntactic structures and even deviating from the original semantics. This further leads to poor quality in the subsequently generated candidate steganographic texts, making them easily detectable by steganalysis methods. To mitigate this problem, we perform a syntax checking and semantic distortion measure to output high-quality texts.

Specifically, we first perform the Stanford CoreNLP toolkit to extract syntactic templates from $\mathbb{O}'$ and then further check and filter out paraphrases whose syntactic templates do not belong to bin_4, obtaining a temporary steganographic text set $\mathbb{C} = \{C_i\}_{i=1}^{|\mathbb{C}|}$. Subsequently, a semantic distortion measure is utilized to the semantic distance between $C_i \in \mathbb{C}$ and cover text x.

$$s_i = \frac{E_x \cdot E_{C_i}}{\|E_x\| \, \|E_{C_i}\|}, \quad (8)$$

where E_x and E_{C_i} are the semantic feature vectors with a fixed dimension corresponding to x and C_i, respectively. E_x and E_{C_i} can be calculated by averaging H_x and H_{C_i} according to Eq. (1), respectively. The semantic distortion scores for $\mathbb{C}$ can be obtained by Eq. (8). Ultimately, the temporary steganographic text with the lowest score will be selected as the final steganographic text x'.

Secret Information Extraction

To extract secret information from multi-granularity spaces (syntactic space and symbolic space) without errors, the data sender must share the same DBC strategy and lexical substitution unit with the data receiver. It's worth noting that the proposed MMLS can independently extract the secret information in parallel from syntactic space and symbolic space.

For the secret information in syntactic space, it is easy to extract b from steganographic text x' through Stanford CoreNLP toolkit and $bins$. The syntactic template $T_{x'}$ is first identified from x'. Then, the data receiver can determine the subset contained $T_{x'}$ to extract the corresponding secret information b. It is pointed out that there is no need for the data receiver to keep the trained syntax-controlled paraphrase generator and the original cover text, which reduces the shared side information and improves security to some extent. For the secret information in symbolic space, the data receiver needs to find the same masked position according to the masking strategy, and then execute the shared encoding strategy to recover the Huffman tree through prediction probability from BERT. Secret information can be extracted from the encoded leaf nodes according to the current word.

Method	Parameters	bpw↑	BLEU↑	MAUVE↑	SIM↑	Acc↓	F1↓
PhraseLS	-	0.3112	0.3577	0.7786	0.6456	0.7819	0.7953
SPLS	s=3,l=1	0.3333	0.2579	0.7958	0.6314	0.7380	0.7396
HISS	H=4, k=4	0.3332	0.4781	0.9202	0.6760	0.7310	0.7095
LSCD	H=4, k=4	0.3327	**0.4765**	**0.9323**	**0.7904**	**0.7195**	0.6846
MMLS	H=4, k=4/s=2,τ=0.1	**0.6305**	0.3299	0.8585	0.7377	0.7205	**0.6702**
PhraseLS	-	0.2416	0.3577	0.7786	0.6456	0.7542	0.7604
SPLS	s=4,l=1	0.2500	0.2579	0.7958	0.6314	0.7187	0.7173
HISS	H=4, k=3	0.2487	0.5731	0.9557	0.7356	0.7005	0.6424
LSCD	H=4,k=3	0.2486	**0.5748**	**0.9643**	**0.8307**	**0.6550**	**0.5988**
MMLS	H=4, k=3/s=2,τ=0.1	**0.5561**	0.3900	0.8719	0.7716	0.6875	0.6381

Table 1: Comparative experimental results of different linguistic steganography methods.

Experiments and Analysis

Datasets and Implementation Details

In the experiments, we select the QQP-Pos dataset (Yang et al. 2022) consisting of 140000 training samples, 3000 validation samples, and 3000 test samples to train the syntax-controlled paraphrase generator. We set the hidden state size to 256, the filter size to 1024, and the head number to 4. The number of layers of the semantic encoder, sentence decoder, and syntactic encoder are set to 4, 4, and 3, respectively. We use Adam optimizer (Kingma and Ba 2015) with a learning rate of 1e-4, and the number of training epochs is 50, with a batch size of 32. Moreover, BERT is initialized with pre-trained *bert-base-uncased* from Hugging Face[2]. The above test samples are also used as cover texts. And we generate 3000 steganographic texts for each experiment.

Evaluation Metrics

We test our method from three aspects: embedding capacity, text quality, and security. The **embedding capacity** is usually evaluated by embedding rate, which means the average bits of secret information embedded per word (bpw). For **text quality**, we measure text fluency and semantic consistency of the generated steganographic texts by employing BLEU(Papineni et al. 2002), MAUVE (Pillutla et al. 2021) and SIM (Zhang et al. 2020) as metrics. MAUVE tends to measure the difference in statistical distribution between generated steganographic text and naturally innocent text in terms of KL divergence. SIM is utilized to measure semantic similarity between steganographic texts and cover texts. To assess the **security**, we select a promising steganalysis method (Peng et al. 2021) to distinguish steganographic texts from cover ones. The detection Accuracy (Acc) and F1-score (F1) are employed as metrics for evaluating the anti-steganalysis ability of linguistic steganography methods. The lower the Acc and F1, the stronger the security.

Baselines

For a fair comparison, we rebuilt some typical baselines as follows. (1) *HISS* (Xiang, Ou, and Zeng 2024) utilizes a syntax-controlled paraphrase generator to modify the syntactic template of the cover text automatically and then embeds secret information into syntactic space by syntactic bins coding strategy. (2) *SPLS* (Yang et al. 2024) employs advanced paraphrasing techniques based on pivot translation to modify the given cover text and embeds secret information into symbolic space. (3) *PhraseLS* (Wilson and Ker 2016) is a traditional MLS method that uses paraphrase rules for word or phrase substitution. Moreover, to evaluate the performance of the proposed DBC strategy, we implement three variant steganography methods with different distance measures (LD, TED, and CD), namely *LSLD*, *LSTED*, and *LSCD*, respectively. They embed secret information in syntactic space by skipping the lexical substitution unit.

[2]https://huggingface.co

Results and Analysis

Comparative Experiment To evaluate the performance of our proposed MMLS, we show the comparative results between MMLS and other baselines in Table 1. For similar embedding capacity, LSCD shows higher values on BLEU, and MAUVE, and lower values on detection Acc, and F1, which implies it significantly outperforms other baselines regarding text quality and anti-steganalysis capability. Although the text quality of MMLS is lower than that of LSCD, the embedding capacity is nearly doubled and the steganography resistance is only slightly reduced. Since the lexical substitution unit utilizes the pre-trained BERT model to predict candidate words for elaborately masked positions and sets a threshold τ to avoid selecting inappropriate words, it only makes slight changes in the intermediate steganographic text generated by the syntactic transformation unit. Compared to baselines, MMLS still has the highest SIM values and lowest Acc values. Particularly, MMLS achieves SIM values that are 5% higher than other baselines, which shows MMLS's superior capability in preserving semantic consistency. MMLS achieves a large embedding capacity while improving semantic consistency and security.

Comparison of Different Syntactic Coding Strategies As shown in Table 2, we present the experimental results of different syntactic bins coding strategies with different parameters (k and H). A larger k indicates a larger embedding capacity per sentence but a lower text quality. Since

Method	k	$H=4$					$H=3$				
		BLEU↑	MAUVE↑	SIM↑	Acc ↓	F1 ↓	BLEU↑	MAUVE↑	SIM↑	Acc ↓	F1↓
LSLD	1	0.7164	0.9427	0.8830	0.5735	0.5167	0.7883	0.9718	0.9097	0.5760	0.5504
	2	0.6469	0.9510	0.8531	0.6350	0.5913	0.6361	0.8858	0.8490	0.6920	0.6504
	3	0.5199	0.9068	0.8022	0.7185	0.6839	0.5147	0.8236	0.7904	0.7680	0.7542
	4	0.4131	0.8202	0.7576	0.7700	0.7519	0.4226	0.8055	0.7559	0.7985	0.7832
LSTED	1	0.7756	0.9868	0.9119	0.5110	0.5107	0.7983	0.9799	0.9134	0.5720	0.5687
	2	0.6463	0.9699	0.8590	0.6235	0.5680	0.6498	0.9041	0.8498	0.6885	0.6442
	3	0.5376	0.9497	0.8127	0.6740	0.6312	0.5017	**0.8991**	0.7920	0.7730	0.7628
	4	0.4504	0.9026	0.7698	0.7190	0.6929	0.4311	**0.8065**	0.7592	0.7805	0.7649
LSCD	1	**0.7887**	**0.9889**	**0.9187**	**0.5155**	**0.5073**	**0.8049**	**0.9900**	**0.9153**	**0.5490**	**0.5340**
	2	**0.6749**	**0.9762**	**0.8707**	**0.6005**	**0.5400**	**0.6775**	**0.9488**	**0.8633**	**0.6570**	**0.6163**
	3	**0.5748**	**0.9643**	**0.8307**	**0.6550**	**0.5988**	**0.5523**	0.8732	**0.8108**	**0.7300**	**0.6993**
	4	**0.4765**	**0.9323**	**0.7904**	**0.7195**	**0.6846**	**0.4324**	0.7925	**0.7623**	**0.7780**	**0.7508**

Table 2: Experimental results of different syntactic bins coding strategies in syntactic space steganography.

s	τ	$k=3$						$k=4$					
		bpw↑	BLEU↑	MAUVE↑	SIM↑	Acc↓	F1↓	bpw↑	BLEU↑	MAUVE↑	SIM↑	Acc↓	F1↓
2	0.1	0.5561	0.3900	0.8719	0.7716	0.6875	0.6381	0.6305	0.3299	0.8585	0.7377	0.7205	0.6702
	0.2	0.4543	0.4274	0.9024	0.7836	0.6895	0.6362	0.5338	0.3542	0.8807	0.7476	0.7175	0.6907
	0.3	0.3975	0.4527	0.9143	0.7916	0.6865	0.6423	0.4808	0.3742	0.9003	0.7549	0.7180	0.6901
3	0.1	0.4277	0.4586	0.9250	0.7961	0.6735	0.6329	0.5135	0.3785	0.9069	0.7575	0.7184	0.6891
	0.2	0.3702	0.4823	0.9101	0.8028	0.6685	0.6152	0.4555	0.3955	0.9024	0.7634	0.6910	0.6419
	0.3	0.3394	0.4975	0.9278	0.8076	0.6625	0.5877	0.4219	0.4082	0.8938	0.7677	0.6905	0.6134

Table 3: Experimental results of the different parameters in lexical substitution unit.

larger k implies less candidate steganographic text for each cover text, it is more likely to choose one with higher semantic distortion and makes steganographic text more prone to identification. When $H = 4$, it is evident from Table 2 that the text quality of LSCD is overall higher compared to other strategies. When $H = 3$, LSCD shows better SIM results than others, which implies LSCD's superior capability of preserving semantic consistency. From the results of BLEU and MAUVE, LSLD, LSTED, and LSCD can generate more fluent steganographic text conforming to the statistical distribution of natural text compared to HISS from Table 1. Additionally, at the same parameters set, LSCD has the best semantic coherence and anti-steganalysis ability in almost all methods. This is because LSCD utilizes more substantial structure information from hidden syntactic features captured by the syntactic encoder to compute the cosine similarity, which can measure the similarity between syntactic templates from a more comprehensive perspective.

Influence of Lexical Substitution Unit Parameters As shown in Table 3, when $H = 4$, we present the experimental results of the proposed MMLS under different parameters (k, s and τ). k controls the embedding capacity corresponding to syntactic space, while s and τ determine the embedding capacity corresponding to symbolic space. A larger s means that the interval between masked positions is larger, in other words, fewer positions are masked in each text. A larger τ may lead to a smaller size of RCWP. For a fixed s, text quality and anti-steganalysis ability increase gradually with an increasing τ, since more low-probability words are dropped in RCWP. When τ is fixed, a large s results in higher text quality and stronger anti-steganalysis ability, but the embedding capacity will also decrease. It should be noted that the text quality and anti-steganalysis ability change smoothly with variations in the parameters s and τ. It reveals that superimposing the substitution-based method on the syntactic transformation-baed method can further improve its embedding capacity at a relatively low cost of text quality and anti-steganalysis ability.

Conclusion

In this paper, we proposed a novel multi-granularity MLS framework MMLS to hide secret information into syntactic and symbolic spaces by jointing syntactic and lexical manipulations. Moreover, our proposed distance-aware syntactic bins coding strategy can mitigate the interference inherent introduced by randomness and further improve the semantic coherence with the cover text. Experimental results demonstrate that MMLS significantly outperforms existing methods regarding semantic coherence, embedding capacity, and security. In the future, we plan to explore more syntax-based steganographic coding strategies to improve embedding capacity and security.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No. 61972057).

References

Chang, C.-Y.; and Clark, S. 2010a. Linguistic Steganography Using Automatically Generated Paraphrases. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 591–599.

Chang, C. Y.; and Clark, S. 2010b. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1194–1203.

Chang, C. Y.; and Clark, S. 2012. The secret's in the word order: Text-to-text generation for linguistic steganography. In *Proceedings of COLING 2012, 24th International Conference on Computational Linguistics: Technical Papers*, 511–528.

Chang, C.-Y.; and Clark, S. 2014. Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. *Computational linguistics*, 40(2): 403–448.

Fan, P.; Zhang, H.; and Zhao, X. 2022. Adaptive QIM with minimum embedding cost for robust video steganography on social networks. *IEEE Transactions on Information Forensics and Security*, 17: 3801–3815.

Grosvald, M.; and Orgun, C. O. 2011. Free from the Cover Text: A Human-generated Natural Language Approach to Text-based Steganography. *Journal of Information Hiding and Multimedia Signal Processing.*, 2(2): 133–141.

Huo, L.; and Xiao, Y.-c. 2016. Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2776–2780.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

Krishnan, R. B.; Thandra, P. K.; and Baba, M. S. 2017. An overview of text steganography. In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, 1–6.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

Meng, P.; Shi, Y.-Q.; Huang, L.; Chen, Z.; Yang, W.; and Desoky, A. 2011. LinL: Lost in n-best list. In *Information Hiding: 13th International Conference*, 329–341.

Murphy, B.; and Vogel, C. 2007. The syntax of concealment: reliable methods for plain text information hiding. In *Security, steganography, and watermarking of multimedia contents IX*, volume 6505, 351–362.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, 849–856.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Peng, W.; Zhang, J.; Xue, Y.; and Yang, Z. 2021. Real-time text steganalysis based on multi-stage transfer learning. *IEEE Signal Processing Letters*, 28: 1510–1514.

Pillutla, K.; Swayamdipta, S.; Zellers, R.; Thickstun, J.; Welleck, S.; Choi, Y.; and Harchaoui, Z. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, 4816–4828.

Simmons, G. J. 1984. The prisoners' problem and the subliminal channel. In *Advances in Cryptology: Proceedings of Crypto*, 51–67.

Stutsman, R.; Grothoff, C.; Atallah, M.; and Grothoff, K. 2006. Lost in just the translation. In *Proceedings of the 2006 ACM symposium on Applied computing*, 338–345.

Ueoka, H.; Murawaki, Y.; and Kurohashi, S. 2021. Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5486–5492.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wen, J.; Zhou, X.; Zhong, P.; and Xue, Y. 2019. Convolutional neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(3): 460–464.

Wilson, A.; and Ker, A. D. 2016. Avoiding detection on twitter: embedding strategies for linguistic steganography. *Electronic Imaging*, 28: 1–9.

Wu, J.; Chen, B.; Luo, W.; and Fang, Y. 2020. Audio steganography based on iterative adversarial attacks against convolutional neural networks. *IEEE transactions on information forensics and security*, 15: 2282–2294.

Xiang, L.; Li, Y.; Hao, W.; Yang, P.; and Shen, X. 2018. Reversible Natural Language Watermarking Using Synonym Substitution and Arithmetic Coding. *Computers, Materials & Continua*, 55(3): 541–559.

Xiang, L.; Ou, C.; and Zeng, D. 2024. Linguistic Steganography: Hiding Information in Syntax Space. *IEEE Signal Processing Letters*, 31: 261–265.

Xiang, L.; Wang, R.; Yang, Z.; and Liu, Y. 2022. Generative Linguistic Steganography: A Comprehensive Review. *KSII Transactions on Internet and Information Systems*, 16(3): 986–1005.

Xiang, L.; Xia, J.; Liu, Y.; and Gui, Y. 2023. CPG-LS: Causal Perception Guided Linguistic Steganography. *IEEE Signal Processing Letters*, 30: 1762–1766.

Yang, E.; Bai, C.; Xiong, D.; Zhang, Y.; Meng, Y.; Xu, J.; and Chen, Y. 2022. Learning structural information for syntax-controlled paraphrase generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 2079–2090.

Yang, T.; Wu, H.; Yi, B.; Feng, G.; and Zhang, X. 2024. Semantic-Preserving Linguistic Steganography by Pivot Translation and Semantic-Aware Bins Coding. *IEEE Transactions on Dependable and Secure Computing*, 21(1): 139–152.

Yang, Z.; Wang, K.; Li, J.; Huang, Y.; and Zhang, Y.-J. 2019. TS-RNN: text steganalysis based on recurrent neural networks. *IEEE Signal Processing Letters*, 26(12): 1743–1747.

Zhang, K.; and Shasha, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6): 1245–1262.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations*.

Zhou, Z.; Dong, X.; Meng, R.; Wang, M.; Yan, H.; Yu, K.; and Choo, K.-K. R. 2023. Generative Steganography via Auto-Generation of Semantic Object Contours. *IEEE Transactions on Information Forensics and Security*, 18: 2751–2765.