STLC-KG: A Social Text Steganalysis Method Combining Large-Scale Language Models and Common-Sense Knowledge Graphs

Zhuang Wang[1], Linna Zhou[1]*, Xuekai Chen[1], Zhili Zhou[2], Zhongliang Yang[1]*

[1]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
[2] School of Artificial Intelligence, Guangzhou University, Guangzhou, China
yangzl@bupt.edu.cn

Abstract

Language steganography in social networks primarily focuses on embedding secret information into social media text efficiently to achieve covert communication. The misuse of such techniques could pose significant potential threats to public cyberspace, such as the spread of malicious code, commands, or viruses. Existing social text steganalysis techniques mainly focus on the analysis of individual social media texts. However, the information content in a single text is very limited, leading to poor detection performance in practical applications. To address this challenge, this paper proposes a social text steganalysis method that combines large-scale language models with common-sense knowledge graphs (STLC-KG). This method first uses knowledge graphs to expand the knowledge contained in the text under investigation, enriching its linguistic expression, and then utilizes large-scale language models to extract the linguistic features of the social text. The results of tests conducted on three mainstream social media platforms demonstrate that the proposed method significantly improves the performance of social text steganalysis.

Introduction

Steganography is one of the key technologies ensuring secure information transmission (Jiang et al. 2020). Its core purpose is to conceal the existence of secret information by embedding it within normal media, thereby protecting its security. This technique plays a critical role in various fields, such as user privacy protection, identity verification, commercial intelligence transmission, and military defense security. However, steganography can also be misused by malicious individuals for the transmission of illegal information, posing significant threats to social security.

Steganography is typically categorized by the type of carrier medium, including image-based, audio-based, video-based, and text-based steganography (Wang et al. 2024). Natural language is the most commonly used information carrier by humans. In public network environments, natural language-based linguistic steganography has become a research focus in recent years due to its wide range of applications and strong robustness. These studies predominantly utilize generative steganography strategies based on natural language. The core concept involves leveraging large-scale pretrained language models to learn the statistical language model of social texts. During the text generation process, by encoding the conditional probability distribution of each generated word (such as Huffman coding (Yang et al. 2018), arithmetic coding (Ziegler, Deng, and Rush 2019), or dynamic adaptive group coding (Zhang et al. 2021)), allowing secret information to be embedded within the generated natural language text. With the rapid development of large language models in recent years, these techniques have made significant strides. When combined with highly secure encoding strategies, these methods can even achieve the generation of nearly imperceptible steganographic text (Ding et al. 2023). This poses a significant challenge to the detection of steganographic texts in cyberspace.

Steganalysis technology is a countermeasure to steganography, aiming to detect the presence of secret information within carriers. Traditional steganalysis methods primarily rely on manually extracted statistical features from text

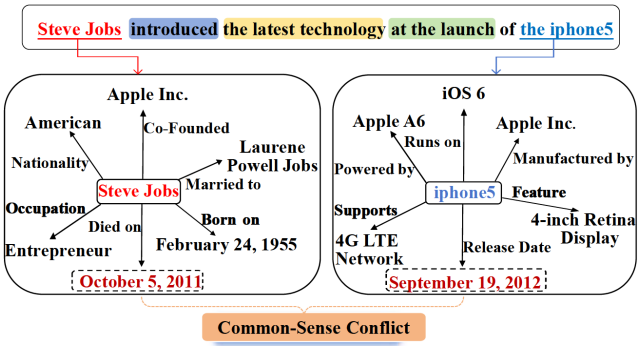

Figure 1: Although many sentences evade model detection through high text quality, the use of specialized domain content, or the complexity of the sentences, such flaws can be uncovered with the help of a commonsense knowledge base. For example, in the sentence: "Steve Jobs passed away on October 5, 2011; however, the iPhone 5 was released on September 19, 2012", the commonsense conflict is evident in the fact that : "Steve Jobs had already passed away by the time the iPhone 5 was released."

*Corresponding author.

(Samanta, Dutta, and Sanyal 2016; Xiang et al. 2007), but these methods are gradually becoming ineffective as the sophistication of covert carriers increases. In recent years, neural network models have been employed to learn the differences between normal and stego texts by analyzing large datasets, leading to more efficient detection. Notable steganalysis research includes: leveraging the sequence processing capabilities of RNNs and LSTMs (Yang et al. 2019; Zou et al. 2020), combining the advantages of LSTM and CNN (Niu et al. 2019), exploring the integration of CNNs with Gated Recurrent Units (GRUs) (Xu, Zhao, and Zhong 2021), and applying Graph Neural Networks (GNNs) to handle text graph structure data (Wu et al. 2021). Additionally, self-training and meta-learning strategies have been introduced (Wang et al. 2023; Wen et al. 2022b). The application of language models has further enhanced analysis performance, particularly with models like GPT-2, BART, and T5, which capture text features and patterns in-depth (Radford et al. 2019; Lewis et al. 2019; Raffel et al. 2020). Furthermore, optimizations in framework design and feature extraction methods, such as hierarchical mutual learning frameworks (Xue et al. 2022) and adaptive domain-invariant feature extraction (Xue et al. 2023), have been explored. However, with the rapid advancement of generative AI, steganography techniques based on automated text generation are producing increasingly difficult-to-detect steganographic texts. For example, steganography techniques combining large language models (LLMs) with existing encoding methods have been proposed (Wang et al. 2024) presenting widespread challenges to previous text steganalysis technologies.

Although significant progress has been made in text steganalysis technology in recent years, several challenges still persist in practical applications. We believe the primary limitation lies in the fact that current social text steganalysis techniques primarily focus on analyzing individual social texts. However, the information contained within a single text is highly limited, and the features extracted from single-sentence content are insufficient to encompass the cognitive level of human understanding, as illustrated in Figure .1. This limitation hinders the model's comprehensive understanding of the text, resulting in poor detection performance in practical applications.

To address this challenge, we propose a social text steganalysis method that combines large-scale language models with commonsense knowledge graphs (STLC-KG). Specifically, the method first utilizes knowledge graphs to perform knowledge expansion on the text under analysis, enriching its linguistic expressions. Then, it employs large-scale language models to extract the linguistic features of social texts. Our contributions are as follows:

- We propose a novel framework for social text steganalysis, called STLC-KG. This framework first utilizes a large-scale commonsense knowledge base to enrich the knowledge of the social text under inspection, compensating for the limitations of individual text information. It then leverages large language models to extract semantic information from the enriched social text and finally analyzes whether any hidden information is present. To the best of our knowledge, this is the first attempt to combine large language models with knowledge graphs for text steganography detection.
- We applied the proposed steganalysis framework to numerous existing text steganography detection models. Across different datasets and steganography algorithms, the framework effectively improved detection performance, demonstrating its robustness and effectiveness.
- We tested the generalization ability of the framework and found that it not only enhances detection capabilities for individual text steganography algorithms but also shows significant advantages when dealing with mixed text steganography algorithms.

Related Work

Sequence Feature-Based Linguistic Steganalysis. (Wen et al. 2019) used Convolutional Neural Networks (CNNs) to extract local semantic features from continuous word vectors. In the same year, (Yang et al. 2019) employed unidirectional and bidirectional Recurrent Neural Networks (RNNs) for sequential feature extraction. (Yang, Huang, and Zhang 2019) introduced the TS-FCN algorithm, which maps word co-occurrence and correlation to high-dimensional space for efficient detection. (Zou et al. 2020) proposed a model combining pretrained language models, Long Short-Term Memory networks (LSTM), and interactive attention. Yang et al. (2020) developed a feature pyramid model that integrates LSTM features across layers. (Li and Jin 2021) utilized Capsule Networks for steganography detection by encoding feature details in capsule vectors. (Wen et al. 2022a) applied BERT with regularization constraints and a revisit mechanism to address generalization issues in domain-adapted text steganography detection.

Graph Network-Based Linguistic Steganalysis. (Wu et al. 2021) first applied GNN algorithms to text steganalysis by converting text into directed graphs, enhancing self-representation with global information, and addressing polysemy. (Xiang et al. 2022) introduced the LS-BGAT framework, using a Graph Attention Network (GAT) to integrate global information from sentences and the corpus. (Fu et al. 2022) combined Gated Graph Neural Networks, GAT, and GCA modules for detection.

Feature Fusion-Based Linguistic Steganalysis. (Niu et al. 2019) combined LSTM and CNN for steganalysis to capture both local and long-term semantic features. (Peng et al. 2021) used knowledge distillation with a pretrained BERT model to guide LSTM and CNN-based steganalysis models. (Yang et al. 2021) developed the SeSy framework, utilizing BERT's semantic representation and detecting syntactic changes after embedding secrets. (Xue et al. 2022) employed multi-model collaborative training and a mimicry loss function to integrate features across levels. (Guo et al. 2022) improved detection by combining TF-IDF with an Auto-Encoder for statistical and semantic features. (Xu, Zhang, and Liu 2023) enhanced both local and global features using a grouping mechanism to filter and weight important features, optimizing detection.

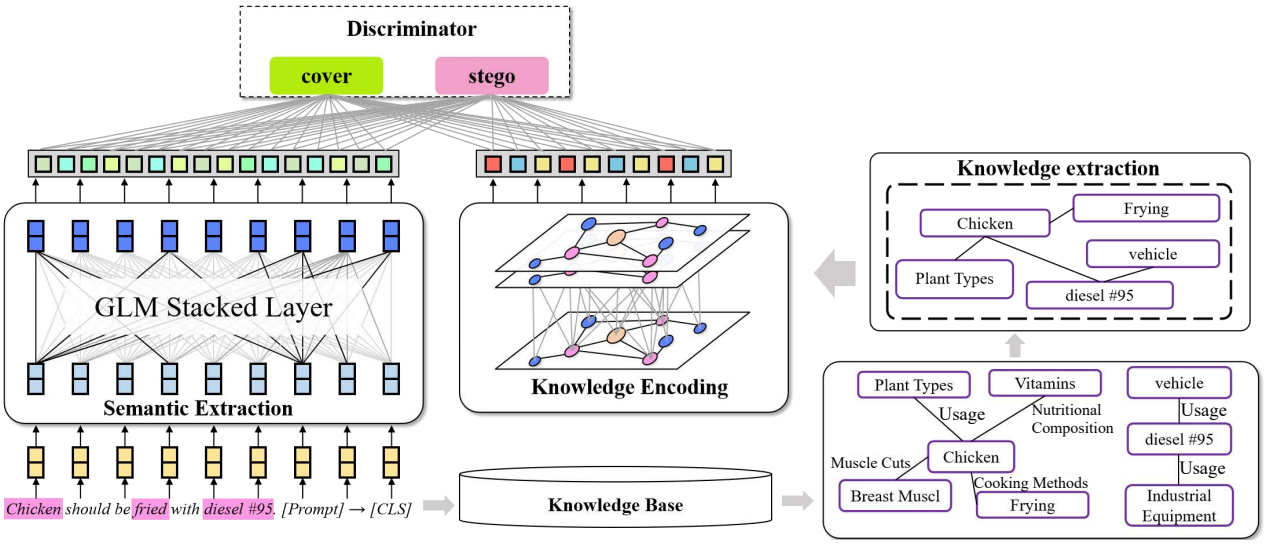

Figure 2: The proposed STLC-KG framework first enriches the analyzed text by expanding its knowledge using a knowledge graph. Then, it leverages a large language model to extract linguistic features from social texts. Finally, it combines and analyzes the semantic and knowledge-based features to determine whether hidden information is present.

Method

The overall architecture of the proposed STLC-KG model is shown in Figure.2. It consists of two parts: the left part is the semantic extraction based on fine-tuning large language models, and the right part is the knowledge encoding based on knowledge feature maps. The former uses a general-purpose generative large language model to extract hybrid semantic features and combines various fine-tuning methods to facilitate the understanding of textual content. The latter links the entities extracted from the input text to a general knowledge base to form a knowledge feature map, independently encodes the knowledge, and evaluates the cognitive logic.

Semantic Extraction Module

To detect the differences between normal and steganographic text, we utilize the open-source bilingual dialogue model ChatGLM2-6B [1] for semantic encoding. This model, with 6.2 billion parameters, has been pre-trained on 1.4TB of Chinese and English tokens. Its strong language understanding capabilities allow us to effectively analyze text for steganographic content.

The GLM model is pre-trained using autoregressive masked language modeling, which aids in capturing textual structure and contextual dependencies. During pre-training, multiple masked words are sampled, and the model attempts to restore the original sequence by maximizing the likelihood of correct predictions:

$$\max_{\theta} Ez \sim Z_m \Big[\sum i = 1^m log\ p_\theta(s_{zi} | x_{corrupt}, s_{z<i}) \Big], \quad (1)$$

where $x_{corrupt}$ is the input text with masked content, and the model learns to reconstruct the sequence based on previous predictions.

[1] https://github.com/THUDM/ChatGLM2-6B

To better capture relative positional information, we adopt Rotary Position Embedding (RoPE) (Su et al. 2024), which integrates relative positional dependencies into the self-attention mechanism, ensuring consistency between the expanded knowledge graph and the original text. The RoPE method modifies the query (q) and key (k) vectors as follows:

$$\widetilde{q_m} = f(q_m, m), \quad \widetilde{k_n} = f(k_n, n), \quad (2)$$

ensuring that the positional relationship between words is preserved.

Fine-tuning. After obtaining the pre-trained model, we applied three popular parameter-efficient fine-tuning algorithms to optimize performance: Partial Fine-tuning (Freeze) (Fu et al. 2023), Low-Rank Adaptation (LoRA) (Yu et al. 2023), and Prefix-Tuning (Li and Liang 2021). Each algorithm enhances large model performance for steganalysis by targeting specific aspects: (1) Partial Fine-tuning (Freeze): Adjusts only top or task-specific layers, preserving the model's generalization from pre-training. (2) LoRA: Updates only auxiliary network parameters, accelerating training and reducing storage. (3) Prefix-Tuning: Uses continuous prefix encodings to improve task adaptability, avoiding local optima from prompt selection. We compare the effectiveness of these methods in text steganography detection in the experimental section.

Knowledge Encoding Module

To make the knowledge encoding method more widely applicable and capable of computing independently and in parallel with the semantic extraction system, a more general and standardized common-sense knowledge base should be used, collecting richer connection information for knowledge enhancement. ConceptNet is a free semantic network that originated from the 1999 crowdsourcing project Open

Mind Common Sense. It focuses on words used in natural language rather than named entities, making it more suitable for pre-trained model encoding by treating words as the smallest units of representation.The current version, ConceptNet 5, contains over 28 million relationship descriptions and 34 types of relations. These include predefined multilingual general relations such as "related to", "form of ", and "has a", as well as informal relations extracted from natural text that are closer to natural language descriptions, such as "on the top of" and "caused by" .

In the model, we first select words from the text to be detected that can be found in the knowledge base and are not stop words. These words are referred to as knowledge mentions. Using the Lookup function provided by ConceptNet's Web API, we find knowledge entities that are similar to the target knowledge mentions or are on the same path (edge) to construct the graph structure. Since it is not convenient to manually assign edge weights during the initial graph construction, we exclude relations such as "Antonym" and "Distinct From" to ensure that the added knowledge information is as minimally detrimental and sufficiently effective as possible.

All selected knowledge mention words in the sentence (the purple parts in Figure.2) are first connected to the external words they perceive (the blue dots on the right side of Figure.2). Next, neighboring nodes within 2 hops in the sentence are connected, and initialized using pre-trained language model word vectors, ensuring that the information in the knowledge feature graph has sufficient learning space. Finally, all nodes within the sentence are connected to a root node with a randomly initialized vector (the yellow dot in Figure.2) to represent the global semantic information of the knowledge feature graph.Each knowledge subgraph is represented as $G(\gamma, \mathcal{E})$, where $\gamma = \{v_1, v_2, \ldots, v_m\}$,$v_i$ representing the i-th node, m represents the number of nodes, The edge set is represented in the form of an adjacency matrix $E \in \{0,1\}^{n \times n}$. When extracting features, a two-layer Graph Attention Network (GAT) is used for information aggregation, the node representation update for each layer can be expressed as:

$$\vec{v}_i^{(l+1)} = f^{(l)} \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \cdot \vec{v}_j^{(l)} \right), \tag{3}$$

here, $\vec{v}_i^{(l)}$ The feature vector of the i -th node at the l-th layer $\mathcal{N}(i)$ is the set of neighboring nodes of node v_i, α_{ij} is the attention coefficient between nodes v_i and v_j, and $f^{(l)}$ is the activation functionThe attention coefficient α_{ij} is typically calculated in the following manner:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^T \left[\vec{v}_i^{(l)} \| \vec{v}_j^{(l)}\right]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\vec{a}^T \left[\vec{v}_i^{(l)} \| \vec{v}_k^{(l)}\right]\right)\right)}, \tag{4}$$

here, $\vec{a}$ is the trainable weight vector, $\|$ denotes the vector concatenation operation, and LeakyReLU is the rectified linear unit activation function with a small positive slope.

Loss Function

To strengthen the common features among similar instances and distinguish the differences between dissimilar instances, in the training process of the STLC-KG model, we adopted the idea of contrastive learning and designed an additional loss function $\mathcal{L}'$.

Specifically, within each mini-batch, the current sample S_t is compared with its positive sample S_p (from the same class) and negative sample S_n (from a different class). The contrastive loss function L' is defined as:

$$\mathcal{L}' = \max_{\theta} \left(0, d\left(S_t, S_p\right) - d\left(S_t, S_n\right) + \Delta\right), \tag{5}$$

here, $d(\cdot)$ represents the Euclidean distance between two samples, and Δ is the margin between positive and negative sample pairs. By maximizing this loss function, samples from the same class are encouraged to be closer together in the embedding space, while samples from different classes are pushed further apart.

The final total loss L_{total} includes the original classification loss L and the contrastive learning loss L', and the model is trained by minimizing L_{total}:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \mathcal{L}', \tag{6}$$

here, L is the original loss function, such as the binary cross-entropy loss function, which handles the basic classification task; L' is the contrastive learning loss function mentioned above, which is responsible for enhancing the discriminative power of the features.

Experiments and Analysis

Dataset

To simulate real-world social network scenarios, we selected three widely used corpora in current linguistic steganalysis methods: TWEET (Go, Bhayani, and Huang 2009), MOVIE (Maas et al. 2011), and NEWS (Yang et al. 2022). We fine-tuned the GPT-2 language model on these corpora to learn the linguistic expression habits of normal text.Next, we used two conditional probability encoding methods, Arithmetic Coding (AC) (Ziegler, Deng, and Rush 2019) and Adaptive Dynamic Grouping (ADG) (Zhang et al. 2021), to automatically select the optimal embedding capacity based on probability. We combined the generated steganographic texts with normal texts, shuffled and mixed them, and assigned the same labels. The number of normal texts in the datasets was randomly sampled to ensure that the total amount of steganographic texts remained equal, and the data were divided into training and testing sets with an 8:2 ratio, ensuring no overlap between the samples. Detailed information is provided in Table 1. Here, BPW represents the average number of bits embedded per word in a sentence; Word Count indicates the average number of words per sentence; and Knowledge References denotes the average number of knowledge references per sentence.

Comparison of Different Fine-Tuning Methods

To compare the adaptability of several fine-tuning methods on various mixed hidden datasets, we conducted experi-

		BPW	Word Count	knowledge References
TWEET	covers	9.65	1.93	-
	stegos	3.87	8.97	1.87
MOVIE	covers	22.43	4.23	-
	stegos	3.55	23.88	4.41
NEWS	covers	28.61	5.09	-
	stegos	3.49	29.94	5.25

Table 1: Details of the steganalysis dataset.

ments where these fine-tuning methods were used individually, without incorporating the knowledge feature graph enhancement. Unified training parameters were set to test their performance. The parameters were as follows: mini-batch size (Batch Size) was 16, learning rate (Learning Rate) was [learning rate value], number of training epochs (Epoch) was 10, weight decay (Weight Decay) was 0.1, and warm-up ratio (Warm up Ratio) was 0.1. For LoRA fine-tuning, the rank matrix (LoRA Rank) dimension was set to 8, with a scaling factor of 16. For P-Tuning fine-tuning, the prefix length was set to 16.

Data	Freeze	LoRA	P-Tuning V2
TWEET	**70.07**	68.63	68.99
MOVIE	72.30	70.44	**73.50**
NEWS	76.61	75.82	**78.75**

Table 2: Detection F1 Scores of the GLM Model under Different Fine-Tuning Methods (%).

Based on existing prompt learning experiences (Liu et al. 2023), when converting a generative language model to an understanding model, it is essential to ensure that the template content closely aligns with the steganalysis task, remains as concise as possible, and suits the reading comprehension style of the generative language model. Through some trial and error, we set the prompt template as follows: "Please determine whether the sentence is steganographic text, output 1 or 0, the text is as follows →[Text to be detected]". The appearance of "→" helps the model recognize the prediction indicator. This template will be applied to both the Freeze method and the LoRA method.

The experimental results are shown in Table 2. It can be observed that the Freeze method achieved the best performance on the TWEET dataset, which has shorter sample lengths, while the P-Tuning V2 method performed best on the longer MOVIE and NEWS datasets. We speculate that this is because the TWEET dataset contains shorter texts with simpler language structures, eliminating the need for re-parameterization of the large model. On the other hand, P-Tuning V2 achieves the original logic of generation through prediction classification task labels, making it more effective in datasets with lengthy content and stronger contextual relevance. Therefore, in subsequent experiments, we continued to use the best fine-tuning method for the respective datasets.

Comparison with the Baseline Model

For comparison with the baseline model, various steganographic detection models from different frameworks were evaluated, including FCN (Samanta, Dutta, and Sanyal 2016), RBC (Niu et al. 2019), BERT (Zou et al. 2020), SeSy (Yang et al. 2021), and SSM (Xu, Zhao, and Zhong 2021). FCN utilizes word co-occurrence consistency for feature mapping, RBC captures both local and long-term semantic features, BERT leverages contextual word relationships, SeSy combines semantic features with dependency syntax features, and SSM introduces feature interaction methods for steganographic text detection in mixed scenarios. "GLM" represents a steganalysis method based on the large language model GLM, while "GLM+KE" represents our proposed social text steganalysis model (STLC-KG), which combines large language models with commonsense knowledge graphs.

Single Steganographic Algorithm Detection. The evaluation results of the experiments are presented in Table 3. Based on the data in Table 3, we can draw the following conclusions: Firstly, it can be observed that our proposed STLC-KG (GLM+KE) model, achieved the highest detection accuracy across all datasets, with an average improvement of 4.18% over the best steganalysis model, SSM. The highest improvement was 6.35% on the MOVIE dataset, which is based on AC encoding, while the lowest improvement was 2.51% on the NEWS dataset, which is based on ADG encoding. Secondly, the steganalysis model showed an improvement in detection accuracy when knowledge enhancement (KE) was added, regardless of whether the datasets were based on AC or ADG encoding. The highest improvement reached 2.82%, and in almost all cases, precision and recall were also improved. This indicates that integrating external knowledge into the model can effectively enhance its performance across different datasets, further demonstrating that leveraging large-scale commonsense knowledge graphs can enrich the knowledge content of the social texts being analyzed, thereby overcoming the limitations of individual text information. Thirdly, compared to the baseline models and the baseline models with added KE, the steganalysis model based on the Generative Language Model (GLM) exhibited higher accuracy, precision, and recall. This confirms that applying prompt learning to generative large language models can fully exploit the domain adaptation and semantic information extraction capabilities of large models for steganalysis.Fourthly, the proposed steganalysis framework effectively improved detection performance across different datasets and steganography algorithms, demonstrating its robustness and effectiveness.

Mixed Steganographic Algorithm Detection. To further assess our model's detection capability in real-world network environments and test the generalization ability of our framework, we mixed steganographic texts generated by different algorithms. The evaluation results of the experiment are presented in Table 4. Based on the data in Table 4, we can draw the following conclusions:Firstly, it can be

Model	TWEET						MOVIE						NEWS					
	AC			ADG			AC			ADG			AC			ADG		
	ACC	P	R	ACC	P	R	ACC	P	R	ACC	P	R	ACC	P	R	ACC	P	R
FCN	65.56	67.14	63.75	64.89	67.87	62.78	64.67	64.62	63.19	64.54	64.43	63.23	63.74	63.64	62.69	63.96	63.75	62.71
FCN+KE	**66.94**	**67.23**	**64.57**	**66.83**	66.50	**63.57**	**65.32**	**65.74**	**63.74**	**65.35**	**65.67**	**64.25**	**64.78**	**64.59**	**63.60**	**64.79**	**64.58**	**63.19**
RBC	64.47	68.46	62.17	64.31	67.25	62.09	67.63	67.89	65.34	66.95	67.74	65.24	67.85	66.90	64.42	66.96	66.11	64.15
RBC+KE	**66.37**	68.26	**63.35**	**65.59**	**67.88**	**62.96**	**68.79**	**68.47**	**66.04**	**68.81**	**68.59**	**67.01**	**69.54**	**68.31**	**67.19**	**68.34**	**67.14**	**65.69**
BERT	67.98	69.89	64.79	67.08	69.59	64.56	70.97	71.39	67.74	69.95	70.98	66.38	72.31	73.97	69.54	71.58	73.37	69.73
BERT+KE	**69.32**	**70.15**	**64.97**	**67.95**	**69.98**	64.35	**73.39**	**71.67**	**68.25**	**71.37**	**71.69**	**67.48**	**74.15**	**74.19**	**69.79**	**72.81**	**73.62**	**71.53**
SeSy	70.19	70.79	67.17	69.14	70.13	67.98	76.49	75.84	71.42	75.81	75.59	70.38	78.16	78.70	76.49	78.91	77.59	75.19
SeSy+KE	**72.02**	**71.41**	**67.23**	**71.96**	70.09	67.38	**77.56**	**77.07**	**72.08**	**76.89**	**76.52**	**72.32**	**80.05**	**79.56**	**77.18**	**80.27**	**79.91**	**75.44**
SSM	72.94	71.93	67.59	71.77	71.32	66.96	78.41	77.78	74.65	78.31	77.48	74.69	82.71	81.36	80.29	81.98	81.34	79.47
SSM+KE	**73.98**	**73.09**	**69.47**	**72.63**	**72.85**	**69.33**	**80.10**	**79.98**	**76.16**	**79.29**	**79.08**	**75.67**	**83.19**	**81.72**	**81.46**	**82.49**	**82.17**	79.36
GLM	75.02	75.87	70.49	74.98	74.67	71.71	82.70	82.35	78.49	81.36	81.93	77.49	83.87	82.18	82.63	82.91	82.49	80.51
GLM+KE	**76.99**	**76.47**	**71.02**	**76.57**	**74.91**	**72.09**	**84.76**	**84.29**	**80.25**	**83.73**	**82.64**	**79.61**	**85.68**	**83.59**	**83.64**	**84.49**	**83.52**	**81.59**

Table 3: Discrimination effect of each detection model on the AC and ADG encoded datasets (%).

Model	TWEET			MOVIE			NEWS		
	ACC	P	R	ACC	P	R	ACC	P	R
FCN	63.58	66.56	60.86	62.78	63.40	62.17	61.96	62.35	61.57
RBC	63.47	67.25	60.09	64.98	66.71	63.34	64.46	65.14	63.79
BERT	66.46	69.52	63.66	66.90	69.98	64.08	70.49	72.96	68.18
SeSy	68.11	69.71	66.38	71.47	73.64	69.42	75.80	76.50	75.11
SSM	68.74	70.81	66.79	71.44	74.47	68.65	76.76	78.14	75.43
GLM	70.02	72.49	67.71	73.38	75.37	71.49	78.65	80.81	76.60
GLM+KE	**70.69**	**73.44**	**69.02**	**74.76**	**76.28**	**74.25**	**80.09**	**81.55**	**78.64**

Table 4: Discrimination effect of each detection model on different datasets(%).

observed that our proposed model also achieved the highest detection accuracy across all datasets, with an average improvement of 2.87% over the best existing steganalysis model in the mixed steganography scenarios, reaching an accuracy of 80.09% on the NEWS dataset. Secondly, the advantage of fine-tuning the language model is more evident in long-text datasets like MOVIE and NEWS compared to the short-text TWEET dataset, which is consistent with our observations in Table2. Thirdly, incorporating knowledge information enables existing models to better filter specific features of steganographic texts, significantly improving detection accuracy and recall rates. The improvement is particularly notable in the MOVIE and NEWS datasets, as these datasets contain a higher average number of knowledge items and richer knowledge information, which are difficult for linguistic steganography techniques to accurately replicate. The knowledge feature graph effectively alleviates this limitation. Additionally, after introducing knowledge encoding, high-parameter models like GLM exhibit greater performance gains compared to models with poorer detection performance. This indicates that the approach helps models avoid over-optimization for specific linguistic patterns, enhancing diversity. It confirms that our proposed framework not only improves the detection of individual steganography algorithms but also demonstrates significant advantages in handling mixed steganography algorithms.

Discussion on the Number of Knowledge Additions

In the knowledge enhancement module, the text is linked to the common knowledge base, and knowledge features are extracted from these connected feature maps. The number of knowledge items used can affect detection performance differently. To explore this impact, preliminary experiments were conducted using the proposed method across various datasets to determine the optimal number of knowledge branches, as shown in Figure 3. To avoid interference from descriptive information in the common knowledge base, the introduced knowledge items consisted solely of words, and the edge initialization weights did not carry semantic information. It was observed that adding more nodes was necessary to have an effect on each detected word. Too few knowledge items resulted in insufficient information in the graph network, while too many knowledge items introduced more noise. Based on the data shown in Figure.3, using 45 knowledge items produced the best overall effect. Therefore, in subsequent experiments, the number of knowledge items added for each entity mention was set to 45.

Example Illustration

In Table 5, we present several examples of steganographic text. These examples are challenging for previous steganog-

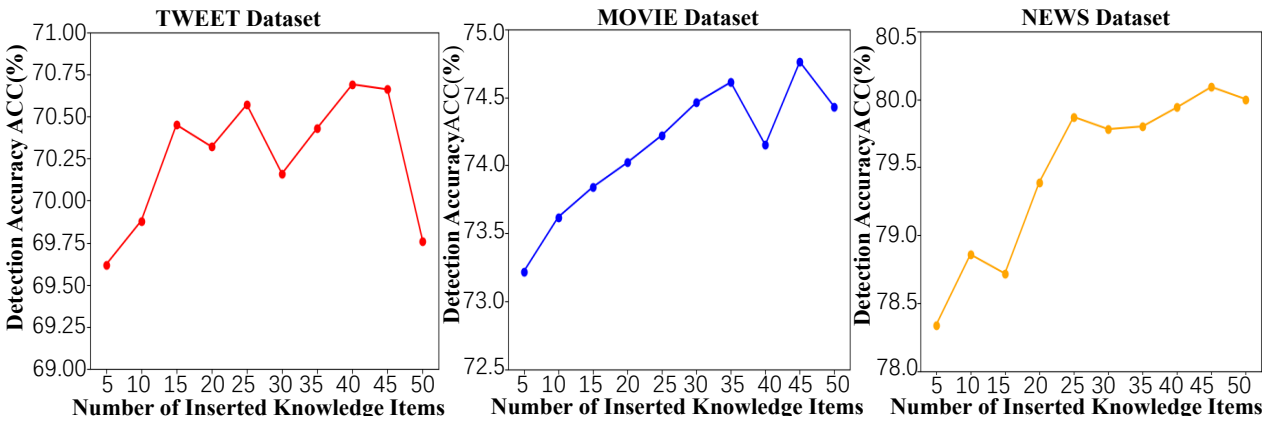

Figure 3: Impact of Using Different Numbers of Knowledge Branch Items on Detection Results.

Error Type	Example	Explanation
Inconsistency in Tense Singular-Plural	Nancy **was walking** down the street when suddenly she **will see** her friend.	The verbs "was walking" and "will see" show a tense inconsistency.
	We propose **a** novel wavelet-based bark coherence function, which **are** based on a wavelet series expansion.	The singular "a" and plural "are" are inconsistent in the sentence.
Domain-Specific Cognitive Error	**A high level of taxation** is the best way to **attract foreign investment**, as it shows fiscal soundness.	"High taxation" does not attract investment, and the reasoning is incorrect.
	We use dependency tree **pruning strategy** to optimize **semantic space**.	"Pruning strategy" is not used for optimizing "semantic space".
Spatiotemporal Inconsistency	**Steve Jobs** introduced the latest technology at the launch of **the iPhone 5**.	Steve Jobs had passed away before the launch of the "iPhone 5".
	Alice went to the store to buy some groceries. On her way **back home**, she remembered she left the stove on and **rushed back** to turn it off.	The action of leaving the stove on should occur at "home", not at the "grocery store".

Table 5: Some steganographic text examples detected by the proposed steganalysis model.

raphy detection models due to their high-quality text, domain-specific content requiring specialized understanding, and complex sentence structures. In this paper, by introducing a commonsense knowledge base, we enhance semantic understanding with logical reasoning, allowing us to identify subtle errors beyond mere text fluency, such as inconsistencies, domain-specific cognitive biases, and spatiotemporal discrepancies, which arise from embedding random secret information during natural language generation. These examples further demonstrate the distinction and advantages of our proposed steganography detection algorithm over previous works, contributing to improved detection performance for steganography in social media texts.

Conclusion

Existing AI content generation technologies, combined with text steganography encoding algorithms, generate realistic content on real social platforms. The diverse forms of expression pose a challenge for single semantic extraction models to fully analyze the text, making steganalysis tasks difficult. To address this issue, this paper proposes a steganography detection model based on fine-tuning large models and encoding knowledge feature graphs. On one hand, the model leverages the understanding and reasoning capabilities of high-parameter models to improve existing semantic extraction methods. On the other hand, it uses independently computable knowledge graphs combined with graph neural networks to capture semantic associations. Experimental results show that by utilizing general pre-trained models and common knowledge bases, the proposed method can comprehensively enhance the perceptual security, statistical security, and cognitive security of steganography detection, demonstrating practical value in real mixed hiding scenarios.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3303301, Grant 2023YFC3305401, Grant 2023YFC3305402 and in part by the National Natural Science Foundation of China (Nos.62302059 and 62172053).

References

Ding, J.; Chen, K.; Wang, Y.; Zhao, N.; Zhang, W.; and Yu, N. 2023. Discop: Provably secure steganography in practice based on" distribution copies". In *2023 IEEE Symposium on Security and Privacy (SP)*, 2238–2255. IEEE.

Fu, Z.; Yang, H.; So, A. M.-C.; Lam, W.; Bing, L.; and Collier, N. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 12799–12807.

Fu, Z.; Yu, Q.; Wang, F.; and Ding, C. 2022. HGA: hierarchical feature extraction with graph and attention mechanism for linguistic steganalysis. *IEEE Signal Processing Letters*, 29: 1734–1738.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12): 2009.

Guo, S.; Liu, J.; Yang, Z.; You, W.; and Zhang, R. 2022. Linguistic steganalysis merging semantic and statistical features. *IEEE Signal Processing Letters*, 29: 2128–2132.

Jiang, S.; Ye, D.; Huang, J.; Shang, Y.; and Zheng, Z. 2020. SmartSteganogaphy: Light-weight generative audio steganography model for smart embedding application. *Journal of Network and Computer Applications*, 165: 102689.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, H.; and Jin, S. 2021. Text steganalysis based on capsule network with dynamic routing. *IETE Technical Review*, 38(1): 72–81.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Maas, A.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 142–150.

Niu, Y.; Wen, J.; Zhong, P.; and Xue, Y. 2019. A hybrid R-BILSTM-C neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(12): 1907–1911.

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.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Samanta, S.; Dutta, S.; and Sanyal, G. 2016. A real time text steganalysis by using statistical method. In *2016 IEEE international conference on engineering and technology (ICETECH)*, 264–268. IEEE.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Wang, H.; Yang, Z.; Yang, J.; Chen, C.; and Huang, Y. 2023. Linguistic steganalysis in few-shot scenario. *IEEE Transactions on Information Forensics and Security*.

Wang, Y.; Song, R.; Zhang, R.; Liu, J.; and Li, L. 2024. LLsM: Generative Linguistic Steganography with Large Language Model. *arXiv preprint arXiv:2401.15656*.

Wen, J.; Deng, Y.; Wu, J.; Liu, X.; and Xue, Y. 2022a. Lifelong learning for text steganalysis based on chronological task sequence. *IEEE Signal Processing Letters*, 29: 2412–2416.

Wen, J.; Zhang, Z.; Yang, Y.; and Xue, Y. 2022b. Few-shot text steganalysis based on attentional meta-learner. In *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*, 97–106.

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

Wu, H.; Yi, B.; Ding, F.; Feng, G.; and Zhang, X. 2021. Linguistic steganalysis with graph neural networks. *IEEE Signal Processing Letters*, 28: 558–562.

Xiang, L.; Liu, Y.; You, H.; and Ou, C. 2022. Aggregating local and global text features for linguistic steganalysis. *IEEE Signal Processing Letters*, 29: 1502–1506.

Xiang, L.; Sun, X.; Luo, G.; and Gan, C. 2007. Research on steganalysis for text steganography based on font format. In *Third International Symposium on Information Assurance and Security*, 490–495. IEEE.

Xu, Q.; Zhang, R.; and Liu, J. 2023. Linguistic steganalysis by enhancing and integrating local and global features. *IEEE Signal Processing Letters*, 30: 16–20.

Xu, Y.; Zhao, T.; and Zhong, P. 2021. Small-scale linguistic steganalysis for multi-concealed scenarios. *IEEE Signal Processing Letters*, 29: 130–134.

Xue, Y.; Kong, L.; Peng, W.; Zhong, P.; and Wen, J. 2022. An effective linguistic steganalysis framework based on hierarchical mutual learning. *Information Sciences*, 586: 140–154.

Xue, Y.; Wu, J.; Ji, R.; Zhong, P.; Wen, J.; and Peng, W. 2023. Adaptive domain-invariant feature extraction for cross-domain linguistic steganalysis. *IEEE Transactions on Information Forensics and Security*.

Yang, H.; Bao, Y.; Yang, Z.; Liu, S.; Huang, Y.; and Jiao, S. 2020. Linguistic steganalysis via densely connected LSTM with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, 5–10.

Yang, J.; Yang, Z.; Zhang, S.; Tu, H.; and Huang, Y. 2021. SeSy: Linguistic steganalysis framework integrating semantic and syntactic features. *IEEE Signal Processing Letters*, 29: 31–35.

Yang, J.; Yang, Z.; Zou, J.; Tu, H.; and Huang, Y. 2022. Linguistic steganalysis toward social network. *IEEE Transactions on Information Forensics and Security*, 18: 859–871.

Yang, Z.; Huang, Y.; and Zhang, Y.-J. 2019. A fast and efficient text steganalysis method. *IEEE Signal Processing Letters*, 26(4): 627–631.

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.

Yang, Z.-L.; Guo, X.-Q.; Chen, Z.-M.; Huang, Y.-F.; and Zhang, Y.-J. 2018. RNN-stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5): 1280–1295.

Yu, Y.; Yang, C.-H. H.; Kolehmainen, J.; Shivakumar, P. G.; Gu, Y.; Ren, S. R. R.; Luo, Q.; Gourav, A.; Chen, I.-F.; Liu, Y.-C.; et al. 2023. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 1–8. IEEE.

Zhang, S.; Yang, Z.; Yang, J.; and Huang, Y. 2021. Provably secure generative linguistic steganography. *arXiv preprint arXiv:2106.02011*.

Ziegler, Z. M.; Deng, Y.; and Rush, A. M. 2019. Neural linguistic steganography. *arXiv preprint arXiv:1909.01496*.

Zou, J.; Yang, Z.; Zhang, S.; Rehman, S. u.; and Huang, Y. 2020. High-performance linguistic steganalysis, capacity estimation and steganographic positioning. In *International Workshop on Digital Watermarking*, 80–93. Springer.