Enhancing Contextual Compatibility of Textual Steganography Systems Based on Large Language Models

Nasouh AlOlabi
University of Sharjah
Sharjah, UAE
nasouhalolabi@gmail.com

Riad Simbol
University of Sharjah
Sharjah, UAE
rsimbol@sharjah.ac.ae

ABSTRACT

This study presents a systematic literature review on textual steganography, with a particular focus on the transformative impact of Large Language Models (LLMs). We trace the evolution of linguistic steganography from early format-based and statistical methods to advanced neural network models, culminating in the current LLM era. Our findings highlight that LLM-based approaches significantly enhance imperceptibility, embedding capacity, and naturalness in cover text generation, addressing long-standing challenges like the Psic Effect. We analyze various LLM-based models, including VAE-Stega, Co-Stega, Discop, and others, detailing their architectures, performance metrics (e.g., BPW, PPL), and unique contributions to the field. The review underscores the shift towards more sophisticated, context-aware, and robust steganographic systems, paving the way for future research in secure and imperceptible covert communication.

KEYWORDS

Systematic Literature Review, Other keywords identifying your study separated by comma

1 INTRODUCTION

This is a test citation [1].

This review explores how large language models (LLMs) are transforming linguistic steganography, the practice of hiding messages in text. We focus on the unique challenges and advances in using LLMs for secure, imperceptible, and high-capacity covert communication.

1.1 Overview of Information Security and Concealment Systems

Information security systems include **encryption**, **privacy**, and **concealment** (steganography).

1.1.1 Encryption Systems and Privacy Systems. These protect content but reveal that secret communication is happening, which can attract attention.

1.1.2 Concealment Systems (Steganography). Steganography hides the existence of information by embedding it in ordinary carriers (e.g., text, images). Text is a challenging carrier due to its low redundancy and strict semantics.

1.2 Introduction to Steganography

Steganography is often explained by the "Prisoners' Problem," where Alice and Bob must communicate secretly under surveillance. The goal is to embed messages so they are undetectable to an observer.

Steganography methods include **carrier selection**, **carrier modification**, and **carrier generation**.

- **Carrier modification:** Hide information in existing text with minimal changes.
- **Carrier generation:** Generate new text that encodes information, allowing higher capacity but requiring naturalness.

1.3 The Significance of Linguistic Steganography

Linguistic steganography enables covert communication, especially where encryption is suspicious. Text is a robust, ubiquitous carrier but presents challenges in balancing imperceptibility and capacity. Advances in deep learning and LLMs improve text quality and security, while related fields like watermarking focus on tracing content origin.

1.4 Scope of the Review

This review covers LLM-based linguistic steganography, focusing on methods, evaluation, challenges, and future directions.

2 STEGANOGRAPHY AND LARGE LANGUAGE MODELS

Large Language Models (LLMs) have emerged as a significant development in the field of natural language processing, profoundly impacting text generation and related applications like steganography and watermarking. Here's a breakdown of their emergence and impact:

2.1 Capabilities and Approximating Natural Communication

LLMs are **generative models** that can **approximate complex distributions like text-based communication**. They represent the best-known technique for this task. These models operate by taking context and parameters to output an explicit probability distribution over the next token (e.g., a character or a word). The next token is typically sampled randomly from this distribution, and the process repeats to generate output of a desired length. Training LLMs involves processing vast amounts of data to set parameters and structure, enabling their output distributions to approximate true distributions in the training data. The **quality of content generated by generative models is impressive** and continues to improve. This has led to LLMs blurring the boundary of high-quality text generation between humans and machines. LLMs are increasingly used to generate data for specific tasks, such as tabular data, relational triples, sentence pairs, and instruction data, often achieving satisfactory generation quality in zero-shot learning for

specific subject categories. They have also shown capabilities in mimicking language styles and semantics, and their generalization ability allows them to comprehend the semantics of context.

2.2 Role in Generative Linguistic Steganography

LLMs are considered **favorable for generative text steganography** due to their ability to generate high-quality text. Researchers propose using generative models as steganographic samplers to embed messages into realistic communication distributions, such as text. This is a departure from prior steganographic work and is motivated by the public availability of high-quality models and significant efficiency gains. LLMs like **GPT-2, LLaMA, and Baichuan2** are commonly used as basic generative models for steganography. Existing methods often use a language model and steganographic mapping, where secret messages are embedded by establishing a mapping between binary bits and the sampling probability of words within the training vocabulary. However, traditional "white-box" methods require sharing the exact language model and training vocabulary, which limits fluency, logic, and diversity compared to natural texts generated by LLMs. They also inevitably change the sampling probability distribution, posing security risks. New approaches, like **LLM-Stega**, explore **black-box generative text steganography using the user interfaces (UIs) of LLMs**, overcoming the need to access internal sampling distributions. This method constructs a keyword set and uses an encrypted steganographic mapping for embedding, proposing an optimization mechanism based on reject sampling for accurate extraction and rich semantics. Another framework, **Co-Stega**, leverages LLMs to address the low capacity challenge in social media by increasing the text space for hiding messages (through context retrieval) and **raising the generated text's entropy via specific prompts** to increase embedding capacity. This approach also aims to maintain text quality, fluency, and relevance. The concept of **zero-shot linguistic steganography** with LLMs utilizes in-context learning, where samples of covertext are used as context to generate more intelligible stegotext using a question-answer (QA) paradigm. LLMs are also used in approaches like **ALiSa**, which directly conceals token-level secret messages in seemingly natural steganographic text generated by off-the-shelf BERT models equipped with Gibbs sampling. The increasing popularity of deep generative models has made it feasible for provably secure steganography to be applied in real-world scenarios, as they fulfill requirements for perfect samplers and explicit data distributions.

2.3 LLM-Based Steganography Models

2.3.1 Evaluation Metrics.

Imperceptibility Metrics. Perceptual: PPL, Distinct-n, MAUVE, human evaluation. Statistical: KLD, JSD, anti-steganalysis accuracy, semantic similarity.

Embedding Capacity Metrics. Bits per token/word, embedding rate.

2.4 Challenges and Limitations in Steganography with LLMs

2.4.1 Perceptual vs. Statistical Imperceptibility (Psic Effect). Improving perceptual quality can reduce statistical security, and vice versa.

2.4.2 Low Embedding Capacity. Short texts and strict semantics limit how much information can be hidden.

2.4.3 Lack of Semantic Control and Contextual Consistency. Ensuring generated text matches intended meaning/context is difficult.

2.4.4 Challenges with LLMs in Steganography. LLMs may introduce unpredictability, bias, or leak information.

2.4.5 Segmentation Ambiguity. Tokenization can cause ambiguity in how information is embedded or extracted.

A primary challenge in steganography, particularly when utilizing Large Language Models (LLMs), revolves around the **distinction between white-box and black-box access**. Most current advanced generative text steganographic methods operate under a "white-box" paradigm, meaning they require direct access to the LLM's internal components, such as its training vocabulary and the sampling probabilities of words. This presents a significant limitation because many state-of-the-art LLMs are proprietary and are accessed by users primarily through black-box APIs or user interfaces. Consequently, these white-box methods are often impractical for real-world deployment with popular commercial LLMs. Furthermore, methods that rely on modifying the sampling probability distribution to embed secret messages inherently introduce security risks because they alter the original distribution, making the steganographic text statistically distinguishable from normal text.

Another significant hurdle is **ensuring both the quality and imperceptibility of the generated text**, encompassing perceptual, statistical, and cognitive imperceptibility. While advancements in deep neural networks have improved text fluency and embedding capacity, older models or certain embedding strategies can still produce texts that lack naturalness, logical coherence, or diversity compared to human-written content. Linguistic steganography methods often struggle to control the semantics and contextual characteristics of the generated text, leading to a decline in its "cognitive-imperceptibility". This can make concealed messages easier for human or machine supervisors to detect. Although models like NMT-Stega and Hi-Stega aim to maintain semantic and contextual consistency by leveraging source texts or social media contexts, this remains a complex challenge.

Channel entropy requirements and variability also pose a considerable challenge. Traditional universal steganographic schemes often demand that the communication channel maintains a minimum level of entropy, which is rarely consistent in real-world communication, especially in natural language. Moments of low or zero entropy can cause existing steganographic protocols to fail or necessitate the generation of extraordinarily long steganographic texts, making covert communication impractical. While schemes like Meteor attempt to adapt by fluidly changing the encoding rate proportional to instantaneous entropy, overcoming this variability without increasing detectability is difficult. The "Psic Effect" (Perceptual-Statistical Imperceptibility Conflict Effect) highlights

this dilemma, where optimizing for perceived quality might compromise statistical imperceptibility and vice-versa.

Furthermore, **segmentation ambiguity** introduced by subword-based language models, commonly used in high-performing Transformer architectures, presents a critical issue for provably secure linguistic steganography. When a sender detokenizes generated subword sequences into a continuous text (e.g., "any" + "thing" becoming "anything") before transmission, the receiver might retokenize it differently (e.g., as a single "anything" token), leading to decoding errors and affecting subsequent probability distributions. Existing disambiguation solutions typically involve modifying the token candidate pool or probability distributions, which renders them incompatible with the strict requirements of provably secure steganography that demand unchanged distributions. While SyncPool attempts to address this without altering the distribution, it may still lead to a reduction in the embedding rate due to information loss.

Additional limitations include: * **Computational Overhead**: LLMs, while powerful, incur a higher computational cost (3-5 times more than prior methods), which could impact real-time communication scenarios. * **Data Integrity and Reversibility**: Some linguistic steganography methods are not reversible, meaning the original cover text cannot be perfectly recovered after message extraction, which is undesirable for sensitive applications. Text data is generally less prone to lossy compression issues than other media, but incompleteness of the steganographic text can still damage the embedded bitstream. * **Ethical Concerns**: The use of pre-trained LLMs may inadvertently introduce ethical issues such as political biases, gender discrimination, or the generation of insulting content. * **Provable Security and Rigor**: Despite decades of research into provably secure steganography, practical systems have been hampered by strict requirements like perfect samplers and explicit data distributions. Many works from the NLP community, while generating convincing text, often lack rigorous security analyses and fail to meet formal cryptographic definitions, making them vulnerable to detection.

Despite their capabilities, generative models are still **far from perfect** in imitating real communication. A significant challenge for practical steganography is the difficulty of finding samplers for non-trivial distributions like the English language, which continues to evolve. When using approximate samplers, there's a risk that an adversary can detect a steganographic message by distinguishing between the real channel and the approximation. LLMs are known to make mistakes, including "hallucinations," which can lead to errors and erratic embedding during text generation, especially for long stego sequences. One critical issue is **segmentation ambiguity** in neural linguistic steganography. LLMs often use **subword tokenization**, meaning a single text can correspond to multiple token representations. If the sender and receiver have different understandings of segmentation, it can lead to incorrect message extraction and affect subsequent generation steps. Current provably secure methods have largely overlooked this. SyncPool is a proposed method to address this by grouping tokens with prefix relationships in the candidate pool without altering the original probability distribution. The **computational overhead of LLMs is higher** compared to prior methods (approximately 3x to 5x), potentially limiting real-time communication. The effectiveness of LLM-based steganography can be limited by the **entropy of the cover text** in social media contexts, as short, context-dependent replies have lower entropy, thus limiting hiding capacity.

3 LITERATURE REVIEW METHODOLOGY

3.1 Research questions

Here are the research questions addressed in this SLR:

- What is the state of published literature on steganographic techniques that leverage large language models (LLMs)?
- In which applications are steganographic techniques with LLMs being explored?
- What metrics and evaluation methods are used to assess the performance of steganographic techniques in LLMs, focusing on factors like capacity, security, and contextual compatibility?
- How are external knowledge sources (semantic resources) integrated into steganographic techniques with LLMs to enhance capacity or contextual relevance?
- What are the limitations and trade-offs associated with current steganographic techniques using LLMs, particularly concerning security, capacity, and contextual compatibility?
- What are the potential future research directions in steganography with LLMs, considering emerging trends and identified gaps in the literature?

3.2 Search query string

We used the following search query string for our initial literature search:

```
(steganography or watermark or "Information Hiding")
 and ("Large Language Model" or LLM or BERT or LAMA or GPT)
```

3.3 Study selection and quality assessment

We established the following inclusion and exclusion criteria for study selection:

3.3.1 Inclusion Criteria.

- **Full Text Access**: Studies for which the full text is available.
- **Language**: Publications written in English.
- **Peer-reviewed**: Articles published in peer-reviewed journals, conferences, or workshops.
- **Publication Date**: Studies published from 2018 onwards, to focus on recent advancements in LLMs.
- **Relevance**: Studies directly addressing steganography, watermarking, or information hiding techniques that utilize or are significantly impacted by Large Language Models (LLMs), BERT, LAMA, or GPT architectures.
- **Research Type**: Empirical studies, surveys, reviews, and theoretical contributions.

3.3.2 Exclusion Criteria.

- **Duplicated Studies**: Multiple publications reporting the same study will be excluded, with the most complete or recent version retained.
- **Incomplete or Abstract-only**: Studies for which only an abstract is available or the full text is incomplete.

- **Irrelevant Studies**: Publications not directly related to steganography with LLMs.
- **Non-English Publications**: Studies not published in English.
- **Non-peer-reviewed Sources**: Preprints, dissertations, theses, books, and book chapters (unless they are extended versions of peer-reviewed conference papers).

3.4 Bibliometric analysis

Briefly note if snowballing was used for additional sources.

4 RESULTS

5 LLM-BASED STEGANOGRAPHY APPROACHES

This section summarizes the main findings from the systematic literature review, focusing on the characteristics and performance of various LLM-based linguistic steganography and watermarking models.

5.1 LLM-Based Steganography Models

Our review identified several key LLM-based steganography models, each with unique approaches, strengths, and performance metrics:

- **VAE-Stega:** Utilizes Variational Auto-Encoders with BERT-BASE (BERT-LSTM) or LSTM-LSTM models. Achieves a bit per word (BPW) of 5.245, focusing on statistical fidelity. Evaluated using PPL, KLD, JSD, FE, CNN, and TS-CSW (Acc, R). Key results include PPL: 28.879, KLD: 3.302, JSD: 10.411, Acc: 0.600, R: 0.616.
- **General Framework for Reversible Data Hiding:** Employs BERTBase for Masked Language Modeling (MLM). Achieves a BPW of 4.152. Key results include BPW=0.5335, F1=0.9402, PPL=134.2199.
- **Co-Stega:** Leverages Llama-2-7B-chat, GPT-2, and Llama-2-13B. Demonstrates high capacity (10.42 BPW), fluency, semantic relevance, and strong resistance to steganalysis. Key results include SR1: 60.87
- **Joint Linguistic Steganography:** Combines BERT MLM with Graph Attention Networks (GAT). Yields a BPW of 2.251 and a PPL of 13.917. Other metrics include KLD=2.904, SIM=0.812, ER=0.365.
- **Discop (Provably Secure Steganography):** Uses GPT-2 for sampling. Achieves a BPW of 5.76. Focuses on provable security. Key results include Capacity: 5.76, Entropy: 6.08, Utilization: 0.95.
- **Generative Text Steganography with LLM:** A black-box approach with a BPW of 5.93 and a PPL of 165.76. Focuses on semantic similarity. Key results include SS: 0.5881, LS-CNN Acc: 51.55
- **Meteor (Cryptographically Secure Steganography):** Uses GPT-2 for sampling. Achieves 4.11 BPW. Focuses on cryptographic security. Key result: 3.09 bits/token.
- **Zero-shot Generative Linguistic Steganography:** Uses LLaMA2-Chat-7B and GPT-2. Achieves a BPW of 2.511 and a PPL of 8.81. Other metrics include JSDfull: 17.90, JSDhalf: 16.86, JSDzero: 13.40.
- **Provably Secure Disambiguating Neural Linguistic Steganography:** Uses LLaMA2-7b and Baichuan2-7b. Achieves 0.85 BPW with zero decoding error and provable security. Key results: Total Error: 0
- **A Principled Approach to Natural Language Watermarking:** Transformer-based encoder/decoder; BERT for distillation. Achieves 0.2 BPW. Focuses on meaning preservation and robustness. Key results: Bit acc: 0.994, Meteor Drop: 0.057, SBERT ↑: 1.227.
- **Context-Aware Linguistic Steganography Model Based on Neural Machine Translation:** Uses BERT (encoder) and LSTM (decoder). Achieves 3.275 BPW. Key results: BLEU: 30.5, PPL: 22.5, ER: 0.29, KL: 0.02, SIM: 0.86.
- **DeepTextMark:** Model-independent watermarking approach tested with OPT-2.7B. Achieves 1 BPW with high detection accuracy and robustness. Key results: 100
- **Hi-Stega:** Hierarchical Linguistic Steganography Framework combining retrieval and generation using GPT-2. Achieves 10.42 BPW with high payload and semantic coherence. Key results: ppl: 109.60, MAUVE: 0.2051, ER2: 10.42.
- **Linguistic Steganography: From Symbolic Space to Semantic Space:** Uses CTRL (generation) and BERT (semantic classifier). Achieves 0.08 BPW. Key results: Classifier Accuracy: 0.9880, Loop Count: 1.0160, PPL: 13.9565.
- **Natural Language Steganography by ChatGPT:** Uses ChatGPT 4.0. Achieves 0.144 BPW with natural concealment and scalability.
- **Rewriting-Stego:** Uses BART (bart-base2). Achieves 4 BPW with high capacity and naturalness. Key results: BPTS: 4.0, BPTC+S: 4.0, PPL: 62.1.
- **ALiSa (Acrostic Linguistic Steganography):** Based on BERT and Gibbs Sampling. Achieves 0.92 BPW. Key results: PPL: Natural = 13.91, ALiSa = 14.85, LS-RNN/LS-BERT Acc & F1 = 0.50.

6 CONTEXT AWARENESS

Linguistic steganography has evolved from early methods to advanced deep learning models and Large Language Models (LLMs). This progression focuses on improving imperceptibility, embedding capacity, and maintaining naturalness and semantic coherence in cover text.

6.1 Context life cycle

6.2 Context-awareness in Steganography

Here's an overview of this evolution:

- **Early and Format-Based Approaches** Early steganography modified existing carriers, like using whitespace or linguistic idiosyncrasies (e.g., synonym substitution). These methods, often rule-based and context-neglecting, resulted in unnatural text and limited capacity (typically <1 BPT), making them easily detectable.
- **Transition to Carrier Generation and Early Text Generation Models** A significant shift involved "carrier generation based steganography," where the carrier text is generated

to hide information, allowing greater freedom and higher embedding rates without altering original carrier statistics.
  - **Syntax Rules and Statistical Methods:** Early text generation for steganography, using syntax rules or statistical models like Markov models, produced easily recognizable texts, failing imperceptibility and security.
  - **Challenges of Early Generation:** The Psic Effect (perceptual-imperceptibility vs. statistical-imperceptibility conflict) was a key challenge. Generated text, though natural-looking, often had detectable statistical properties. Models also lacked semantic control, crucial for covert communication.
- **The Era of Custom Artificial Neural Network (ANN) Models** Advancements in ANNs and NLP led to sophisticated models for automatic steganographic text generation. These neural networks learned language models, encoding secret information by manipulating word probability distributions during generation.

7 DISCUSSION

Discuss implications and interpretation of the results.

8 CONCLUSION

Summarize the main findings and takeaways of the study.

REFERENCES

[1] Zhong-Liang Yang, Si-Yu Zhang, Yu-Ting Hu, Zhi-Wen Hu, and Yong-Feng Huang. 2020. VAE-Stega: linguistic steganography based on variational auto-encoder. *IEEE Transactions on Information Forensics and Security* 16 (2020), 880–895.