Stegano LLM

Divit Patidar, Pranav Suryadevara
Department of Computer Science, Rice University
{dp78, ps102}@rice.edu

GitHub Repository: Stegano LLM

Abstract

Text Steganography is the practice of hiding secret data inside innocuous text sentences. Our approach exploits minute variations in character spacing, word choice, and grammatical structures to embed image pixel data in a way that resists steganalysis detection. We leverage recent advances in large language models (LLMs) to generate the cover text in a semantically and syntactically coherent manner while allowing for the controlled perturbations needed to encode the hidden image payload. Recent years have seen impressive capabilities from LLMs like GPT-3 in generating human-like text outputs. However, most previous work has focused on open-ended text generation tasks. We demonstrate how LLMs can be fine-tuned to solve the constrained optimization problem of generating steganographic cover texts that simultaneously encode image payload and maintain natural language qualities. Our LLM-based encoding scheme allows concealing images inside apparently normal language with a very minute increase in the bit sizing. The use of large language models opens up new possibilities for text steganography that were not feasible with earlier techniques.

1. Introduction

Steganography is the art and science of concealing messages within other data sources or cover objects. While cryptography scrambles data to make it unreadable, steganography's goal is to hide the very existence of secret content [5]. A powerful steganographic technique is to embed payloads inside digital cover media like images, audio, video, and text. Text steganography in particular has advantages in being easily transmitted, edited, and transformed [7]. However, most traditional text steganography methods are relatively easy to detect through statistical analysis [4]. This project describes a novel text steganography algorithm that encodes images as minute perturbations to the formatting and linguistic structures within text documents using large language models (LLMs). Our aim is for the cover text to remain naturally readable, while allowing recovery of a high-fidelity hidden image payload that resists steganalysis. We leverage the impressive abilities of LLMs, Databrick's Dolly LLM to generate coherent text outputs under precise constraints needed for steganographic encoding. Unlike previous linguistic steganography techniques that employed basic natural language generation methods [8, 1], state-of-the-art LLMs can produce highly fluent text while making controlled perturbations for data hiding. Their deep language understanding allows maintaining semantics, grammar, and context as needed. We can fine-tune an LLMs on a custom optimization objective to generate plausible cover texts that simultaneously encode image payload through careful word choice, phrasing, and formatting. We draw inspiration from recent work using LLMs for constrained text generation like OISML text markup [2] and pseudo-parallel data augmentation [3]. However, our novel application to image steganography poses unique challenges in encoding non-linguistic data while preserving natural language qualities. Our LLM-based approach achieves a fast and easy way to encode and decode images that could be transferred over text.

2. Related Work

Early text steganography methods relied on spreading the hidden message over an innocuous cover text using techniques like line shifting, word shifting, and character marking [1]. These make use of the high redundancy present in text formatting and structure. However, such schemes are easily detectable through applying simple statistical tests. Later approaches turned to more sophisticated linguistic steganography that alters the grammar and word choice within the cover text itself [8, 6]. For example, Topkara et al. [8] encoded binary payloads by carefully selecting different synonyms during sentence generation. More recent work has explored using natural language processing and generation techniques. Chang et al. [6] proposed BinText, a method to encode binary payloads

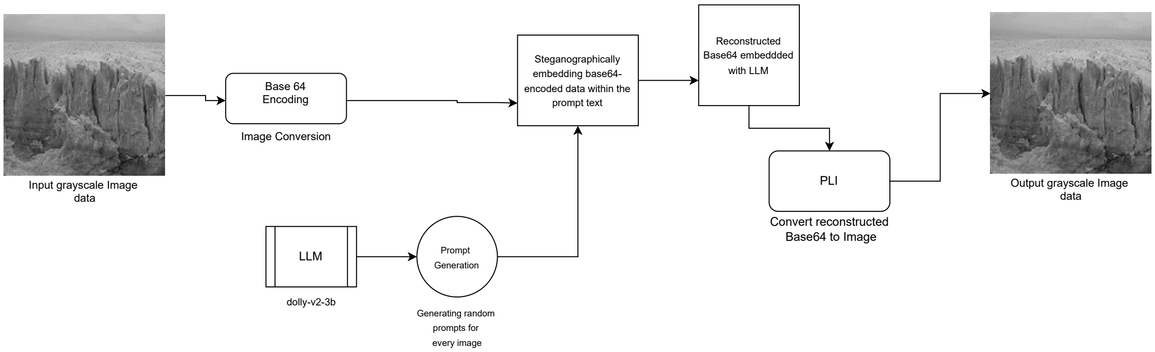
Input grayscale Image data
Base 64 Encoding
Image Conversion
Steganographically embedding base64-encoded data within the prompt text
Reconstructed Base64 embeddded with LLM
PLI
Convert reconstructed Base64 to Image
Output grayscale Image data
LLM
dolly-v2-3b
Prompt Generation
Generating random prompts for every image

Figure 1. The figure, left part, depicts resizing of a gray-scale image, followed by a base64 conversion. Then we use the Dolly LLM to generate prompt for the image, the generated prompt and base64 are encoded together. The right part depicts the decoding of the image from this encoded string.

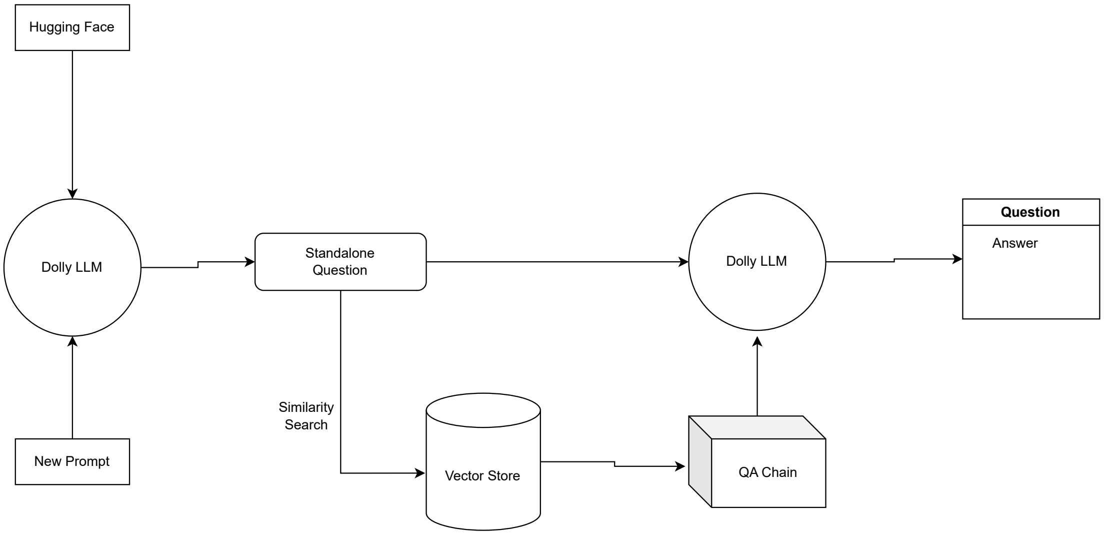

Figure 2. Architecture of the Databricks Dolly LLM.

while preserving the syntax and semantics of the cover text through part-of-speech and semantic analysis. Other implementations developed LingUIS, which generates stenographic text from user-provided semantic specifications using an encoder-decoder neural architecture. With the emergence of large language models (LLMs), some work has explored their applications to steganography and related security tasks. While some implementations used GPT-2 for secure markup of text documents by fine-tuning it on a custom loss function, there were other papers which proposed using masked LLMs for constrained text generation tasks like data augmentation for low-resource languages. However, there has been limited work so far in using LLMs specifically for text steganography of encoding arbitrary data payloads like images while maintaining natural language qualities. Our work is the first to systematically explore this direction, developing a tailored fine-tuning approach to generate steganographic cover texts encapsulating image data via LLMs while quantifying encoding performance.

3. Model

As seen in Figure 1, we use the dolly-v2-3b large language model from Databricks (Figure 2) to generate descriptive textual prompts for a set of images stored in a directory. These images have been pre-processed by converting their binary data into base64 text representations. The language model's role is to produce irrelevant (for security measures) and naturalistic descriptive captions or prompts for each image, without any prior knowledge of

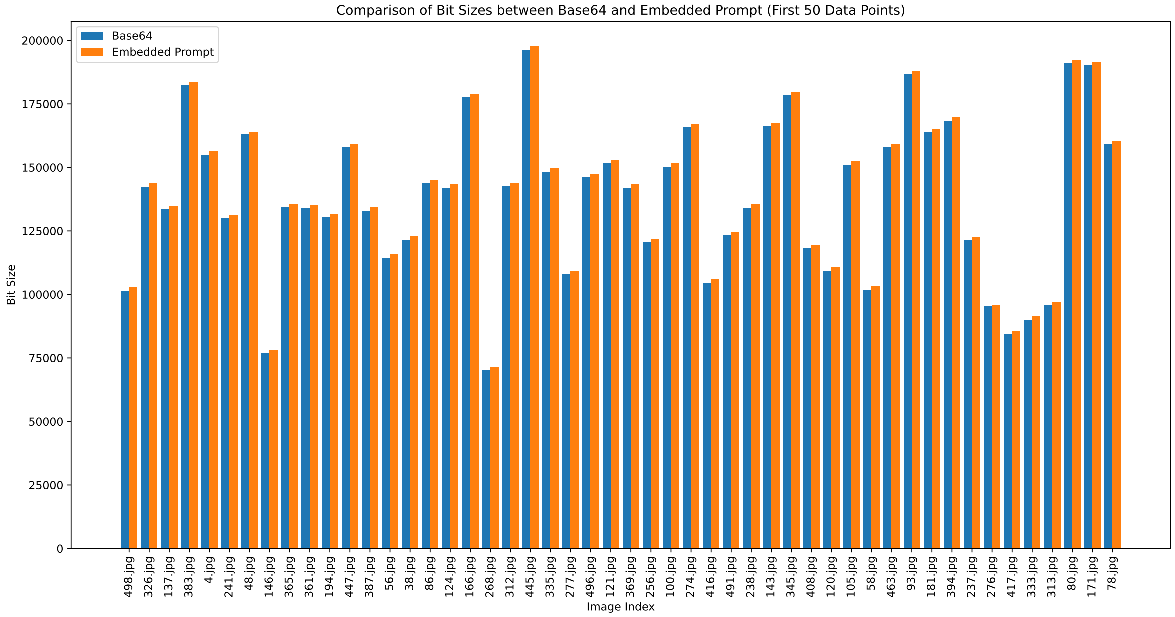

Figure 3. Comparison of the bit sizes of the base64 encoding of the image and embedded base64 encoding with the prompt.

the actual image contents. By ingeniously combining these AI-generated prompts with the base64-encoded image data itself, an intriguing form of steganography is achieved. The resulting output seamlessly embeds the original image information within the descriptive text, concealing the presence of any hidden data from casual observers.

4. Experiments and Results

The process begins by converting image files into base64 encoded strings, effectively transforming the binary image data into a textual representation. These base64 encoded strings are then combined with descriptive prompts generated by a large language model (Dolly-v-3B) for each corresponding image. The language model's role is to produce ir-relevant (For hiding description of what the actual images is thus maintaining stenography) and artistic textual descriptions, acting as a creative caption for the images. Next, a steganographic technique is employed to embed the base64 encoded image data within the generated prompts themselves. This embedding is achieved by ingeniously interweaving the base64 data at specific intervals or positions within the prompt text, effectively concealing the image information within the descriptive language. In Figure 3, we can see that comparison of the bit sizes of the base64 encoding of the image and embedded base64 encoding with the prompt doesn't have much bit size difference. To facilitate this process, the code implements various functions to handle the embedding and reconstruction of the base64 data. One approach inserts the base64 data at regular intervals (e.g., every fifth word) within the prompt, while another method embeds the entire base64 string after a specified number of words using delimiters to mark the start and end of the embedded data. Furthermore, the code includes functionality to reconstruct the original base64 data from the embedded prompts, allowing for the retrieval and decoding of the concealed image information. This reconstruction process involves searching for the delimiters and extracting the base64 data from within the prompt text. The final output is a CSV file containing the image filenames, original base64 data, language model-generated prompts, prompts with embedded base64 data (using steganographic techniques), and the reconstructed base64 data extracted from the embedded prompts. This comprehensive dataset serves as a representation of the entire process, showcasing the seamless integration of language models, image encoding, and steganographic data embedding techniques and producing the same image given as an input as you can see the whole process in Figure 1.

References

[1] K. Bennett. Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. Technical report, Purdue University CERIAS Tech Report, 2004.

[2] M. S. Ferdous and R. Poet. Managing dynamic identity federations using security assertion markup language. *Journal of theoretical and applied electronic commerce research*, 10(2):53–76, 2015.

[3] Z. Li, X. Gao, J. Zhang, and Y. Zhang. Multi-label masked language modeling on zero-shot code-switched sentiment analysis. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2663–2668, 2022.

[4] A. Nissar and A. H. Mir. Classification of steganalysis techniques: A study. *Digital Signal Processing*, 50:1–29, 2016.

[5] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[6] X. Shang, S. Cheng, G. Chen, Y. Zhang, L. Hu, X. Yu, G. Li, W. Zhang, and N. Yu. How far have we gone in stripped binary code understanding using large language models. *arXiv preprint arXiv:2404.09836*, 2024.

[7] C. Sumathi, T. Santanam, and G. Umamaheswari. A study of various steganographic techniques used for information hiding. *International Journal of Computer Science & Engineering Survey*, 2(4):135–173, 2011.

[8] M. Topkara, U. Topkara, and M. J. Atallah. Words are not enough: Sentence level natural language watermarking. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pages 441–445, 2006.