Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model

Honai Ueoka **Yugo Murawaki** **Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
hueoka@icn.cce.i.kyoto-u.ac.jp
{murawaki, kuro}@i.kyoto-u.ac.jp

Abstract

With advances in neural language models, the focus of linguistic steganography has shifted from edit-based approaches to generation-based ones. While the latter's payload capacity is impressive, generating genuine-looking texts remains challenging. In this paper, we revisit edit-based linguistic steganography, with the idea that a masked language model offers an off-the-shelf solution. The proposed method eliminates painstaking rule construction and has a high payload capacity for an edit-based model. It is also shown to be more secure against automatic detection than a generation-based method while offering better control of the security/payload capacity trade-off.

1 Introduction

Steganography is the practice of concealing a message in some cover data such that an eavesdropper is not even aware of the existence of the secret message (Simmons, 1984; Anderson and Petitcolas, 1998). While images, videos, and audio have been dominant cover media (Fridrich, 2009), natural language is a promising choice, thanks to the omnipresence of text (Bennett, 2004).

Formally, the goal of linguistic steganography is to create a steganographic system (*stegosystem*) with which the sender *Alice* encodes a secret message, usually in the form of a bit sequence, into a text and the receiver *Bob* decodes the message, with the requirement that the text is so natural that even if transmitted in a public channel, it does not arouse the suspicion of the eavesdropper *Eve*. For a stegosystem that creates the text through transformation, we refer to the original text as the *cover text* and the modified text as the *stego text*. A stegosystem has two objectives, *security* and *payload capacity*. Security is the degree of how unsuspicious the stego text is while payload capacity is the size of the secret message relative to the size of the stego text. The two objectives generally exhibit a trade-off relationship (Chang and Clark, 2014).

Edit-based approaches used to dominate the research on linguistic steganography. Arguably, the most effective approach was *synonym substitution* (Chapman et al., 2001; Bolshakov, 2005; Taskiran et al., 2006; Chang and Clark, 2014; Wilson and Ker, 2016), where a bit chunk was assigned to each member of a synonym group, for example, '0' to *marry* and '1' to *wed*. The cover text *She will marry him* was then modified to the stego text *She will wed him* such that the latter carried the secret bit sequence '1'.

This conceptual simplicity was, however, overshadowed by the complexity of linguistic phenomena such as part-of-speech ambiguity, polysemy, and context sensitivity. For this reason, edit-based approaches were characterized by the painstaking construction of synonym substitution rules, which were tightly coupled with acceptability checking mechanisms (see Chang and Clark (2014) for a review and their own elaborate method). With all these efforts, edit-based stegosystems suffered from low payload capacity, for example, 2 bits per sentence (Chang and Clark, 2014).

With advances in neural language models (LMs), edit-based approaches have been replaced by generation-based ones (Fang et al., 2017; Yang et al., 2019; Dai and Cai, 2019; Ziegler et al., 2019; Shen et al., 2020). In these approaches, bit chunks are directly assigned to the conditional probability distribution over the next word estimated by the LM, yielding impressive payload capacities of 1–5 bits per word (Shen et al., 2020).

However, it remains challenging for an LM to generate so genuine-looking texts that they fool both humans and machines (Ippolito et al., 2020) even if they do not encode secret messages. It is also worth noting that generation-based stegosystems do not necessarily cut out the need for cover texts, as Ziegler et al. (2019) and Shen et al. (2020)

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5486–5492
June 6–11, 2021.

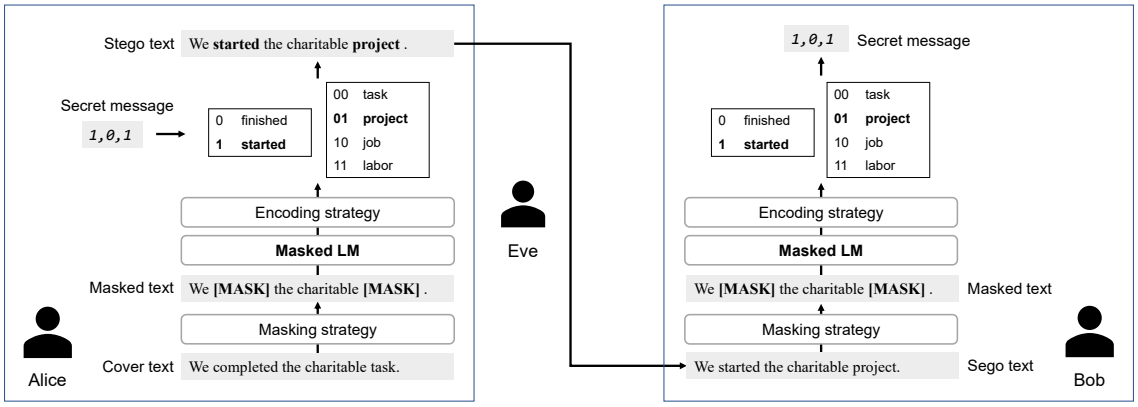

Figure 1: Overview of the proposed method. Alice (sender) and Bob (receiver) share the masked language model (masked LM) and the masking and encoding strategies in advance. Alice masks some tokens in the cover text and makes the masked LM generate a vocabulary distribution for each masked token. Bit chunks are assigned to some high-probability subwords in the distribution from which one is chosen according to the secret message. The stego text is then transmitted in a public channel Eve (eavesdropper) monitors. Receiving the stego text, Bob performs mostly the same procedure to decode the secret message.

conditioned generation on human-written introductory sentences to ensure the stego text quality.

In this paper, we revisit edit-based linguistic steganography. Our key idea is that a masked language model (masked LM), which was first introduced with BERT (Devlin et al., 2019), offers an off-the-shelf solution. Usually treated as an intermediate model with no direct application, the masked LM drastically simplifies an edit-based stegosystem. It eliminates painstaking rule construction because it readily offers a list of words applicable in the context. As illustrated in Figure 1, all Alice and Bob have to share is the masking and encoding strategies in addition to the masked LM.

In our experiments, we showed that the proposed method had a high payload capacity for an edit-based model. As expected, the amount was far smaller than those of generation-based models, but the proposed method offers better control of the security/payload capacity trade-off. We also demonstrated that it was more secure against automatic detection than a generation-based method although it was rated slightly lower by human adversaries.

Our code is available at https://github.com/ku-nlp/steganography-with-masked-lm.

2 Proposed Method

2.1 Masked LM

The essential ingredient of the proposed edit-based stegosystem is a masked LM. It was first introduced along with BERT (Devlin et al., 2019) as an effective pretraining strategy for the Transformer-based (Vaswani et al., 2017) neural net. The pretrained model is usually fine-tuned on downstream tasks, but for our purpose we keep it intact.

Given a text in which some tokens were replaced with the special token `[MASK]`, the masked LM is trained to recover the original tokens based only on their context. As a result of the training, it provides a probability distribution over the vocabulary for each masked token according to the applicability in the given context. Note that high probability items are not necessarily synonymous with the original tokens but nevertheless fit into the context.

Our key insight is that we can use these probability distributions to encode a secret message in the form of a bit sequence. As shown in Figure 1, Alice and Bob share some *encoding strategy* with which bit chunks are assigned to some high probability items. Alice creates a stego text by choosing items that correspond to the secret message. Bob in turn decodes the secret message by selecting bit chunks that correspond to each token in the stego text. The only remaining requirement for Alice is to share some *masking strategy* with Bob in advance so that Bob can correctly identify the tokens to be masked.

2.2 Masking Strategy

We have various design choices for masking and encoding strategies, which affect both security and payload capacity. For masked LM training, BERT randomly masked about 15% of tokens in the input, but we need to ensure that both Alice and Bob mask

the same tokens. In this paper, we present a simple strategy. As a general rule, we mask every one in f tokens in the input, but we skip tokens if they match any of the following criteria:

1. A punctuation or number.
2. A stopword.
3. A non-initial subword, which BERT's standard tokenizer marks with the initial "##".

Editing subwords is dangerous because there is no 100 percent guarantee that Bob's subword tokenization reproduces Alice's original segmentation. For example, if "##break" in the word "un ##break ##able" is replaced with "#us", the subword tokenizer would segment the new word into "un ##usable", distorting the masking positions. We will revisit this problem in Section 3.4.

The hyperparameter f is expected to control the security/payload capacity trade-off. A large f lowers the payload capacity but is likely to increase the difficulty of detection. We also anticipate that since the tokens we decide to skip do not have many good alternatives, not masking them is good for the stego text quality.

2.3 Encoding Strategy

We use block encoding for simplicity. For each masked token, we select and sort items whose probabilities are greater than p. To avoid distorting masking positions, we drop items that are to be skipped in the masking phase. Let n be the largest integer that satisfies $2^n \leq c$, where c is the number of the remaining items. Each item is given a unique bit chunk of size n. Coding is an active research topic (Dai and Cai, 2019; Ziegler et al., 2019; Shen et al., 2020) and is orthogonal to our core proposal.

3 Experiments

We tested the proposed method with several configurations and compared it with a generation-based method. To assess security, we employed automatic discriminators and human adversaries.

3.1 Models and Data

BERT For the proposed edit-based method, we used BERT (Devlin et al., 2019) as the masked LM. Specifically, we used Google's $\texttt{BERT}_{\texttt{Base, Cased}}$ model and Hugging Face's `transformers` package (Wolf et al., 2020) with default settings. Given a random bit sequence as the secret message and a paragraph as the cover text, the model encoded bit chunks on a sentence-by-sentence basis. When the bit chunks reached the end of the secret message, the process was terminated, discarding the remaining sentences in the given paragraph. The last bit chunk usually exceeded the limit, and the remainder was filled with zeros.

GPT-2 Ziegler et al. (2019) built a state-of-the-art generation-based model on top of the GPT-2 neural LM (Radford et al., 2019). We used their original implementation[1] to encode random bit sequences. We set the option `finish_sent` to true to avoid terminating generation at the middle of a sentence. We tested the temperature parameter $\tau = \{0.4, 0.7, 1.0\}$. Since the generation was conditioned on context sentences, we supplied the first three sentences of a paragraph.

Data We extracted paragraphs from the English part of the CC-100 dataset (Wenzek et al., 2020) and used them as the cover texts for BERT and as the contexts for GPT-2.[2] For each stegosystem, we also extracted texts that were comparable to the corresponding stego texts in terms of length. We refer to them as *real texts*.

3.2 Automatic Detection

We trained discriminators to distinguish stego texts from real texts. This corresponds to a situation unusually favorable to Eve as she has access to labeled data, though not to secret messages. A practical reason for this is that after all, we cannot build discriminators without training data. Besides, a stegosystem's performance is deemed satisfactory if it manages to fool the discriminator even under such disadvantageous conditions.

For each stegosystem, we fine-tuned the same $\texttt{BERT}_{\texttt{Base, Cased}}$ model on the binary classification task. The details are explained in Appendix A.

3.3 Human Evaluation

We asked Amazon Mechanical Turk[3] workers to give 5-point scale ratings on the stego and real

[1] https://github.com/harvardnlp/NeuralSteganography

[2] Ziegler et al. (2019) used the CNN/Dailymail (Hermann et al., 2015; Nallapati et al., 2016) as the contexts. We found, however, that the resulting stego texts were excessively easy for automatic discriminators to distinguish from real news articles, presumably due to domain mismatch with a web corpus on which GPT-2 had been trained. That is why we chose CC-100, a web corpus, in our experiments. Note that this setting may have worked slightly against the proposed method because BERT was mainly trained on Wikipedia.

[3] https://www.mturk.com/

Model	Parameters	Bits/word ↑	Acc ↓
BERT	$f = 3$ $p = 0.02$	0.204	**0.586**
GPT-2	$\tau = 1.0$	**1.67**	0.819

Table 1: Results of automatic detection.

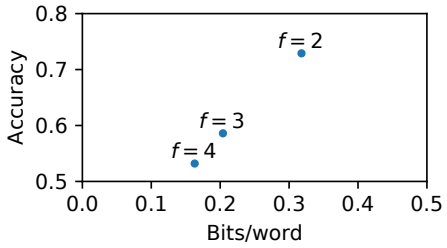

Figure 2: The effect of the masking interval f.

texts according to naturality and correctness. Since we found a consistent bias toward shorter texts, we tuned each stegosystem's hyperparameters to generate stego texts with comparable length. The details are explained in Appendix B.

3.4 Results

Table 1 shows the result of automatic detection. As expected, the proposed method, BERT, had a much lower payload capacity than the generation-based GPT-2 although it was high for an edit-based method. In practical situations, however, security is given priority over payload capacity. In this respect, BERT's performance was remarkable as its stego texts were nearly indistinguishable from real texts. By contrast, GTP-2's stego texts were easily detectable for the discriminator even though they were much shorter than BERT's.

Figure 2 shows the effect of the masking interval parameter f, with $p = 0.02$. We can observe a clear trade-off between the two objectives.

Figure 3 indicates the effectiveness of the mask skipping heuristics explained in Section 2.2. With $f = 4$ and $p = 0.02$, masking stopwords and subwords not only raised detection accuracy but also lowered payload capacities. Because these tokens did not have many good alternatives, they consumed only small bit chunks and simply damaged the stego text quality.

As we briefly discussed in Section 2.2, editing subwords may cause distortion in mask positions, leading to decoding failures. We quantified the risk, with the hyperparameter settings of $p = 0.02$ and $f = 3$. We found that 1.41% of the masked tokens

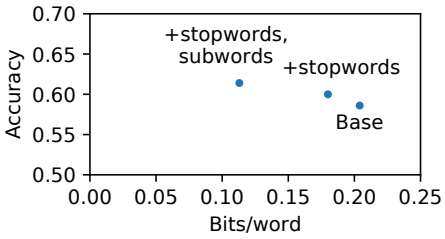

Figure 3: The effect of mask skipping heuristics. The plus sign indicates that the model stops skipping the specified class of tokens.

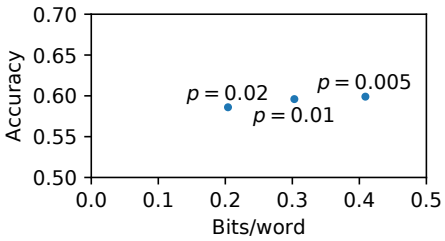

Figure 4: The effect of the probability threshold p.

had substitution candidates that did not reproduce the original segmentations. Although this danger applies equally to generation-based steganography built on top of subword LMs (Dai and Cai, 2019; Ziegler et al., 2019; Shen et al., 2020), to our knowledge, we are the first to point it out.

Figure 4 shows the effect of the probability threshold p. Lowering the threshold increases the payload capacity because the number of alternative tokens increases. It did sacrifice detection accuracy, but not as much as we expected.

As for human evaluation, Table 3 summarizes the results with average ratings. Overall, both methods achieved high average ratings, almost equal to that of the real texts. However, BERT slightly underperformed GPT-2. We conjecture that the quality of the cover texts affected the edit-based method more directly than the generation-based method. Following Ziegler et al. (2019), we initially used news articles for cover/real texts but switched to web texts because we noticed that the discriminator appeared to exploit the domain mismatch with a web corpus on which GPT-2 had been trained. Considering the massive quality improvement efforts given to GPT-2's training data, however, there seems to be much room to improve the quality of CC-100 (Wenzek et al., 2020).

Table 2 shows good and bad stego texts produced by the BERT-based method. In the first example, BERT successfully suggested context-aware words,

Cover text	Stego text	Rating
Switzerland also has an **amazing** scientific community that includes **Geneva** University and CERN, which is one of the top **research** institutes in the world and is home to the **world**'s largest particle physics **laboratory**.	Switzerland also has an **international** scientific community that includes **Basel** University and CERN, which is one of the top **physics** institutes in the world and is home to the **world**'s largest particle physics **laboratory**.	5.0
Allowing local authorities to **increase** that charge puts the negative **political** feedback, particularly in areas where **compliance** is less, like Donegal, on to the local **councils** and protects the central government.	Allowing local authorities to **file** that charge puts the negative **negative** feedback, particularly in areas where **opposition** is less, like Donegal, on to the local **government** and protects the central government.	2.8

Table 2: Two examples of stego texts produced by the proposed edit-based method. The last column indicates average ratings by crowdworkers.

BERT	GPT-2	Real texts
4.32 ± 0.97	4.43 ± 0.89	4.54 ± 0.78

Table 3: The results of human evaluation. The ratings range from 1 to 5, and higher is better.

e.g. *Basel* for a university in Switzerland. In the second example, a single mistake, the unnatural repetition of *negative*, had a critical impact on human raters. Finally, we confirmed that the current sentence-wise encoding created a risk of discrepancies between the first and second sentences.

Editing proper nouns like Geneva is prone to factual errors. One may feel tempted to apply a part-of-speech tagger or a named entity tagger to skip proper nouns. Just like subword substitution, however, a naïve application of automatic analysis does not guarantee the sameness of the masking positions. A good compromise with a guarantee of success in decoding is to skip words with capitalized letters. Solving this problem at its source is an interesting direction for future research.

4 Conclusions

In this paper, we demonstrated that the masked language model could revolutionize edit-based linguistic steganography. The proposed method is drastically simpler than existing edit-based methods, has a high payload capacity, and allows easy control of the security/payload capacity trade-off.

The masked language model is a general framework adopted by many BERT-like models, of which attempts to handle longer texts (Beltagy et al., 2020; Wang et al., 2020) are particularly relevant to steganography. Tailoring the training procedure to steganography is also an interesting research direction.

Ethical Considerations

This paper works on steganography. Unlike cryptography, steganography conceals the fact that a secret message is being transmitted as well as its contents. Steganography can be just fun, but it usually involves a conflict of interest between two parties: those who want to censor media and those who want to evade detection. Depending on value judgments, either one or both can be evil. Steganography is an effective tool to counter censorship in countries where encryption is illegal and visibly encrypted messages may be incriminating. However, it can also be used to transfer malicious data. As such, steganography can be seen as a dual-use technology.

References

Sahar Abdelnabi and Mario Fritz. 2020. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. arXiv:2009.03015.

Ross J Anderson and Fabien AP Petitcolas. 1998. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv:2004.05150.

Krista Bennett. 2004. Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. Technical report, Center for Education and Research in Information Assurance and Security, Purdue University.

Igor A. Bolshakov. 2005. A method of linguistic steganography based on collocationally-verified synonymy. In *Information Hiding*, pages 180–191, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ching-Yun Chang and Stephen Clark. 2014. Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. *Computational Linguistics*, 40(2):403–448.

Mark Chapman, George I. Davida, and Marc Rennhard. 2001. A practical and effective approach to large-scale automated linguistic steganography. In *Information Security*, pages 156–165, Berlin, Heidelberg. Springer Berlin Heidelberg.

Falcon Dai and Zheng Cai. 2019. Towards near-imperceptible steganographic text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4303–4308, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Jessica Fridrich. 2009. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, pages 1693–1701. Curran Associates, Inc.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. arXiv:1412.6980.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar GuÌ‡lçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jiaming Shen, Heng Ji, and Jiawei Han. 2020. Near-imperceptible neural linguistic steganography via self-adjusting arithmetic coding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 303–313, Online. Association for Computational Linguistics.

Gustavus J Simmons. 1984. The prisoners' problem and the subliminal channel. In *Advances in Cryptology*, pages 51–67. Springer.

Cuneyt M. Taskiran, Umut Topkara, Mercan Topkara, and Edward J. Delp. 2006. Attacks on lexical natural language steganography systems. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 97–105. International Society for Optics and Photonics, SPIE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. arXiv:2006.04768.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Alex Wilson and Andrew D. Ker. 2016. Avoiding detection on twitter: embedding strategies for linguistic steganography. *Electronic Imaging*, 2016(8):1–9.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhong-Liang Yang, Xiao-Qing Guo, Zi-Ming Chen, Yong-Feng Huang, and Yu-Jin Zhang. 2019. RNN-Stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5):1280–1295.

Zachary Ziegler, Yuntian Deng, and Alexander Rush. 2019. Neural linguistic steganography. In *Proceedings of the 2019 Conference on Empirical Methods*

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1210–1215, Hong Kong, China. Association for Computational Linguistics.

A Details of Automatic Detection

The edit-based and generation-based methods differ considerably in the amount of text needed to encode a secret message because the latter has a higher payload capacity by design. For a fair comparison, we chose to encode bit sequences of equal size, meaning that the proposed method yielded longer stego texts. Specifically, we used random 32-bit sequences. The sequence of sentences was given to the discriminator at once. For GPT-2, we discarded the contexts and only fed stego texts into the discriminator.

For the binary classification task, we prepared the training, development and test sets with 8,000, 1,000, and 1,000 paragraphs, respectively. Each set had the same amount of stego and real texts, and they were comparable with respect to length. The discriminators were trained for 20 epochs, with the batch size of 32. We applied early stopping if the validation loss did not drop five times in a row. The model snapshots were saved for every 250 steps, and the one with the lowest validation loss was chosen. We used Adam (Kingma and Ba (2017)) as the optimizer with the learning late of 10^{-6}.

B Details of Human Evaluation

For the proposed edit-based method, we used the same masking and encoding strategies used for automatic detection. The hyperparameters were as follows: $p = 0.02$, and $f = 4$, stopwords and subwords were skipped. We used random 6-bit sequences as secret messages.

For the generation-based method, the temperature parameter τ was set to 0.7. We used random 45-bit sequences as secret messages.

We designed a MTurk HIT (human intelligence task) following Abdelnabi and Fritz (2020). Workers were asked to rate texts (each question had 5 texts) with a Likert scale from 1 (lowest) to 5 (highest). As shown in Table 4, the ratings were described with the instructions ranging from *"This sentence is completely understandable, natural, and grammatically correct"* to *"This sentence is completely not understandable, unnatural, and you cannot get its main idea"*. Each HIT had 5 texts, with stego texts from both methods and real texts of comparable length appearing in a random order. The questions also had a simple attention check and if the answer to the attention check was wrong, the corresponding HIT was discarded. We have set the reward per assignment at $0.3.

Rating	Description
5	The text is understandable, natural, and grammatically and structurally correct.
4	The text is understandable, but it contains minor mistakes.
3	The text is generally understandable, but some parts are ambiguous.
2	The text is mainly not understandable, but you can get the main ideas.
1	The text is completely not understandable, unnatural, and you cannot get the main ideas.

Table 4: Ratings explanations given in the human evaluation.

We observed that human raters strongly favored shorter texts. To verify the observation, we performed linear regression analysis with the number of words as a parameter. We found that it indeed had a statistically significant negative impact on the ratings with $p < 10^{-3}$ for the t-statistic.