Linguistic Text Steganography Using Transformer Models

Haya Gamal Abdel Mohsen Dr. Abeer Hamdy Dr. Khaled Nagaty

ABSTRACT

With NLP's great innovation in technological advancements, Text Steganography, which works by unsuspiciously hiding confidential data in innocuous digital text, turns out to have the most considerable attention due to its widespread usage in cybersecurity since that the textual information such as password authentication and banking credentials are considered to be the most sensitive and confidential information that need a solid protection from malicious attacking. Motivated by concerns for user anonymity and privacy, three linguistic methodologies based on Transformer mode architecture, GPT2 a pre-trained transformer language model created by OpenAI, BERT which is an edit-based transformer and finally, RoBERTa (Robustly Optimized BERT Pre-training Approach) which is a modified version of BERT for training optimization, is proposed to design three stegosystems which embeds a secret text message within a text medium (known as cover message) using their wide capabilities in generating conditional text that embeds the words of the secret message in such unprecedented way to extract the new steganographic text. Using this stegosystem, sender and receiver can exchange encrypted messages that keep their coherency while hiding the secret information undetectably by any third party. This approach is demonstrated on introducing a new steganographic approach using RoBERTa for the first time in comparison with the two other pre-trained models. The objective of this proposal is to come up with an approach that consumes reasonable capacity percentage of cover text without corrupting the context of steganographic text and at the same time hide as many words and sentences as possible, to output a context coherent steganographic text that attackers will not even manage notice the existence of such embedded message.

1 Introduction

Steganography is the science of hiding a message within another one without drawing any attention for any anomaly of the carrier message. The term was firstly in 1499 by Johannes Trithemius on his book "Steganographia" which was known as the book of magic. The process is generally hiding messages which is known as "secret message" to be part of another medium such as images, documents, or audio which is known as "cover medium" and sometimes a key is used called as "stego key" in the encoding and decoding processes between two parties. The output of this process is a steganographic text which is a text medium having the secret message hidden inside done using several different techniques. Steganography research done were more likely concerning the embedding of sensitive information within images or audio which in fact, are considered of high steganographic capacity mediums that allows large amount of information to be encoded within making small unrecognized changes to the cover medium. There are a lot of proposed techniques on the images/audio steganography field but the most common one is storing the sensitive data within the least significant bits of the file where it contains the least details of the image/audio file not changing anything in how the file appears to other parties. However, using those mediums is not the best type of steganography to hide critical information due to their widely known decoding techniques among attackers. Text steganography, on the other hand, is capable of securing such critical information.

Despite its lower capacity, text steganography is a secure, faster method for hiding textual critical information as text is a widely used way of communication. Text steganography involves lots of techniques from changing the formatting of an existing text by changing spelling, punctuation, font size, font color or hide pieces of secret-

text within white spaces which is called Format based approach, to converting the secret message into their equivalent ASCII values in order to generate a new formula for the steganographic text called Random & statistical generation, to replace words of cover medium with synonyms of secret message words which is known as the Linguistic method.

In order to deliver a steganographic text in such unnoticeable way is not perfectly accomplished through many text steganography techniques. It is believed that text steganography is the trickiest of all types as encrypting only a small fraction of messages can make the steganographic text suspicious due to its anomaly in appearance because of the deficiency of the redundant bits which is found only in images and audio files creating some capacity limitations. The problem with the format-based technique is the low hiding capacity and low distortion robustness against structural attacks as their hiding algorithms can sometimes lead to failure which can arouse suspicious to a human reader especially using font-sizing. Comparing the plaintext and the resulted steganographic text using Format-based methods makes manipulated parts very visible. Regarding the Random and Statistic generation, although the generated text has similar statistical properties (word length and letter frequencies) with the secret message, generated text does not always provide a coherent meaning which leads to the identification of the hidden communication in addition to its time consumption in encoding and decoding the secret text. This leaves some of the text steganography techniques with capacity limitations and no context coherence which leads to the ease of capturing and understanding by any attacker or third party.

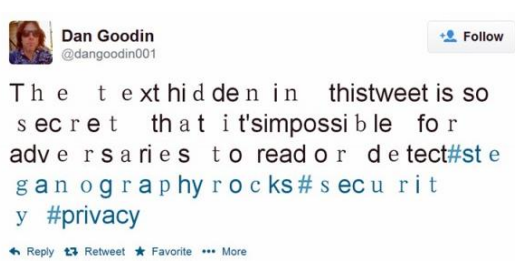

Figure 1: Example of the failure of Format-based generated steganographic text using font-sizing method

In recent years, significant advances have been featured in text generation fields using Deep Learning especially works in

language models whose main focus is high quality readable text generation that are proposed in many applications such as neural machine translation, image captioning or dialogue systems. From this point, a linkage between high-capacity language models and steganography is proposed in this project at which linguistic steganography can be generalised well hiding not only the content of the secret message, but the fact that a secret message has been sent at all due to the fluency of paragraphs produced using the three well known transformer models GPT2, BERT and RoBERTa which are pretrained language models of diverse datasets taking the secret message as a sequence of tokens and embedding them in a formulated nonsuspicious context coherent steganographic text which appears as a normal text of any existing topic.

2 Related Work

In this section, we summarize related work, then present out proposal.

There are two types of linguistic steganography, Generation-based and Edit-based approaches. Generation-based approaches aim to directly output the steganographic text by generating a series of words based on a language model. LSTM, a Generation-based approach, is well-known for its NLP innovations in text generation tasks, not only this but in steganographic purposes since it is one of the efficient generation-based steganographic methods that aims to directly output the steganographic text by generating a serious words based on a language model. An approach was proposed with title Generating a steganographic text using LSTM in 2017, the approach starts with taking the secret message then converting it into bit sequence and using the vocabulary bin in LSTM which is considered the shared key between the sender and receiver in addition to the LSTM language model, the secret message bit sequence is divided into chunks of equal size each corresponds to a bit block that consist of list of random tokens chosen randomly. While Edit-based approaches is based on editing a given cover text or secret message itself to be transformed into an innocent cover text. B. Gupta and S. Kumar in paper concerns were mainly on the lexical side of generating the steganographic text Using the Part-of-Speech Tagging technique in NLP. Their method was focused on keeping the original meaning of the cover text by lexical substitution which is replacing a word by a word of the same part of speech tag; “verb from cover text” is to be replaced with “verb from secret text” etc. By this method, it will not be recognizable that the text has a hidden secret text within, as it avoids any anomaly in the context coherence of the steganographic text by trying to craft cover text in the related context of secret text. The Location of the secret message words replacement in cover text is calculated by $\text{Count POS}_{\text{cover}}/\text{Count POS}_{\text{secret}}$ where POS means the count of this part of speech within the cover text (for example 5 nouns) divided by that of the secret text (1 noun). So that 1 noun in the secret text will be replaced in the position of the 5th noun out of the 5 nouns of the cover text (5/1=5).

3 Proposed Method

In this paper, we revisit edit-based linguistic steganography using the Transformer models. Our key idea is that a masked language model (masked LM), which was first introduced with BERT (Devlin et al., 2019) one of the edit-based transformer models, that offers an off-the-shelf solution for previously mentioned problems of output stego texts. Additionally, a comparison is to take place between three well-known transformer models, GPT2 (Generative Pre-trained Transformer 2) as a generation-based method, BERT (Bidirectional Encoder Representations from Transformers) and its modified version, RoBERTa (Robustly Optimized BERT Pre-training Approach) as Edit-based methods for language modeling. It is important to note the Generation-based method do not need cover texts in steganographic purposes but edit-based requires cover texts. From this point, a comparison is proposed to finalize which steganographic approach is the best fit for ensuring a safe, unsuspicious secret communication between sender and receiver throughout a public channel monitored by eavesdroppers.

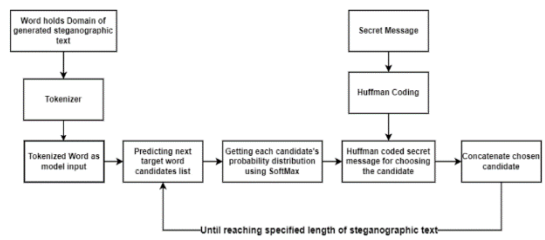

Figure 2: proposed Generation-based stegosystem

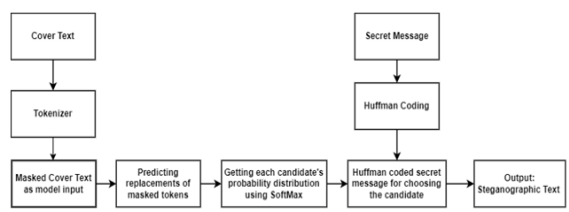

Figure 3: proposed Edit-based stegosystems

Motivated by LSTMs as steganographic method, three mentioned models were built with some modifications in choosing the next predicted word that fits the meaning of the sentence and at the same time, encode a bit chunk from the secret message bit sequence. And by applying the masking strategy, one can adjust the payload capacity required to encode secret message given. Finally, each model generates stego texts by using their main components and modifications described in paragraphs below.

Why Transformer Models?

3.1 Attention Mechanism - How Words are Related

One of the important reasons to propose Transformer models is their attention mechanism. It states that it is not sufficient to understand the individual words but to understand how the words relate to each other in the context of the sentence. For each input token, three main vectors are created:

Query each input token is represented with its query, which is used to score against all other words in order to select the token of the most probability to be related to the input token.

Key each word has a label known as a key, which matches to the search for relevant words.

Value once each word is labelled to how relevant it is to the query, values are detailed representation of each word.

to calculate the score of each word in model vocabulary to know how well they match, a dot product calculation is done between query vector of the token searching for its relevant ones by each key vector of each word from the candidates. A matrix is resulted of the scores in order to be slapped on masking process. Masking is a method to hide unwanted small scores that is

implemented as a matrix called attention mask. It sets the cells wanted to be masked to negative infinity or a very large negative number. After applying the attention mask, only the large scores are remained on the matrix. Then, a SoftMax activation function is applied on each row in order to produce the actual scores needed for self-attention layer. This makes transformer models a better fit for generating context coherent steganographic texts due to their deep understanding of relationships between words within same sentence.

Self-Attention Mechanism for Generation-based

Self-attention only allows model to peak on the left side of word input sequence in finding the most suitable word that is relevant to the context of the right sided tokens of the sequence only masking or eliminating learning context of words in the right-handed side. The masked self-attention layer is located in each of the decoder blocks of the GPT2 where each token passes through along the path. However, Edit-based approaches' innovation key was focusing on both sides of a token (bidirectional) than focusing only on the left-handed side word as implemented in GPT2. Therefore, it is trained bidirectionally to read the text input all at once, this characteristic allows the model to learn the context of one word based on all its surrounding both left and right words applying the attention mechanism.

3.2 Multi-Head Attention

Multi-head attention allows the model to jointly attend to the information from different representations at different positions. Multi-head refers to the multiple times where masked self-attention is conducted on different parts of the key, query, and value vectors those parts are called heads at which each head is a row of each of the key, query, value vectors in order to result a matrix conducting all relevant aspects for this input token. In the previous part, splitting the resulted vector of multiplying the input with the weight of the decoder block gives a long vector for each of the query, key, and value of this input token. Splitting attention heads means reshaping those long vectors into three 12x64 matrices (key, query, value) since 12 is the number of rows (attention heads) that also refers to the 12 attention heads a Transformer base size has.

3.3 Masked LM

The essential component of the two proposed edit-based stegosystems, BERT and RoBERTa, is a masked LM. The pretrained models is usually fine-tuned on downstream tasks, but for our purpose we keep it intact.

Masking is the process similar to fill in the gaps which are represented in [MASK] token. Given a cover text, some tokens were to be replaced with the [MASK] token, the masked LM is then trained to recover those tokens based on the context. As humans, according to general world knowledge, we can come to guessing the right words to fill in the mask tokens but how BERT/RoBERTa does that? they do not know the actual words, but it does know the linguistic patterns given as word embeddings and the context of the rest of input sequence that helps in predicting the **logits**, which are the suitable list of candidates whose probability distribution over the vocabulary is higher than assigned probability threshold $\boldsymbol{p}$. Logits are also known as the final hidden vectors corresponding to masked tokens are fed into the output SoftMax layer to get the real probabilities.

RoBERTa – Newly proposed Steganographic method

RoBERTa stands for Robustly Optimized BERT Pre-training Approach. Researchers at Facebook and Washington University presented this model at which it has the **similar architecture** as **BERT** with some simple design changes in its architecture and training procedure. In this study, the focus will be mainly on RoBERTa as it was not proposed before as text steganography approach that showed efficient results due to the modifications below.

Removing the Next Sentence Prediction (NSP) Objective

RoBERTa shows that better results were given without using related sentences as model is not able to learn on long dependencies, Conditioning on unrelated context from another document adds noise to masked language model and lastly, A model benefits from longer full-length contexts (full sentences). And this was shown in the results of Roberta as it gives best results for full long length sentences to achieve context coherency in produced steganographic texts.

More Data, Larger Batch Size and Train Longer

BERT was originally trained for 1 million steps with a batch size od 256 sequences. For better results, RoBERTa is trained with 125 steps for 2,000 sequences. This has two advantages as the large batches improve perplexity on masked language modelling objective. RoBERTa is trained with 160 GB including Book Corpus + English Wikipedia (16 GB), CC-News (76 GB), OpenWebText (38 GB) and Stories (30 GB) comparing to the only 13 gigabytes of data that BERT is trained on.

Masking Strategy

In BERT architecture, masking the cover text during data preprocessing in BERT Tokenizer results in a single static mask which means same input masks are fed to the model on every single epoch. This happens as BERT duplicates the same input data 10 times masking each one differently and passes the 10 into the training process once among all epochs so that if the model have 40 epochs each masked sequence was seen 4 times with the same mask. This means that there is not enough learning of relationships of words. Dynamic masking, introduced in RoBERTa has shown impressive training process at which it generates a new masking pattern on every epoch. By this strategy, the model learns much more about replacements for each mask token on each different masked input sequence so learning rate increases and RoBERTa can creatively predict replacements words giving a context coherent text. All these modifications showed how RoBERTa can be a great fit to be introduced for the first time for steganographic purposes for achieving one of the hideous problems faced, the context coherency, due to poor training of models.

3.4 Encoding Strategy

Instead of simple block encoding used in LSTMs, Huffman encoding was used for applying the steganography. The secret message is sent to the Huffman coding converter, converting it into a compressed bit sequence such that payload capacity (number of bits being embedded inside text) is reduced so that it can embed as many bits as possible. This text compression technique is used mainly to reduce the bit sequence size of secret message to be encoded so it consumes a reasonable capacity payload to avoid corruption of context.

Our key insight for proposed edit-based approaches is how one candidate will be chosen out of the output list, is by using Huffman Encoding by which a candidate is chosen per [MASK] token by dividing the bit sequence into chunks by which each is converted into its decimal equivalent which will hold the index of the chosen word out of the list of candidates as replacement for [MASK] and the process continues until all mask tokens are replaced. Since GPT2 is a generation-based steganographic approach which is not

restricted by size of a cover text. A random word is given of a specific context as an input to GPT2 to generate words sequence of same context. For each position, a list of candidates is resulted that are multiplied by SoftMax activation function to give each a probability distribution of how suitable each word is according to a score. The encoded secret message which is resulted from Huffman coding is divided into bit chunks, each chunk is then converted into its decimal equivalent which will hold the index of the chosen word out of the list of candidates as the next predicted word.

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, if not, the remainder was filled with zeros.

4 Experiments

We tested the three proposed methods with several configurations. To assess context coherency, grammatical correctness, and delivery of correct facts, we employed human adversaries. Additionally, Masking capacity experiments were conducted on different hyperparameters of cover texts and secret message for testing payload capacity of each method.

4.1 Datasets

Generation-based Method Dataset 3 Datasets each consist of articles of different domains (political, medical, and academic) so that the GPT2 word input is randomly selected from them.

Edit-based Methods Dataset On the other side, Edit-based approaches proposed can specify the context or domain where we want to hide the secret message by providing a cover text of the required domain. A Dataset was formulated of 200 cover texts of different domains (political, medical, and academic) and of different lengths for payload capacity evaluation.

Each model has to output a stego text for each of the domains mentioned for reasonable comparison.

Secret Messages Dataset A Dataset consists of 150 secret messages of different lengths for payload capacity evaluation.

4.2 Human Evaluation

A survey was conducted on a group of students to give 5-point scale ratings on resulted stego-texts from the three proposed models to give an evaluation about which model give the most non-suspicious sentences according to grammatical correctness, context coherency and delivering a correct fact for political stego-texts only. Each model's hyperparameters were tuned to generate stego texts with comparable length. Hyperparameters were as follows: probability distribution threshold = 0.01 and both stop words and sub words were skipped. Participants were asked to rate texts (each model propose 4 stego texts for each domain) with a Likert scale from 1 which is the lowest to 5, as shown in the table below, the ratings were described with the instructions shown for each aspect. The survey consists of fifteen questions, five stego texts for each model. Thirty-four students have participated in doing the survey. Survey screenshot is shown in Appendix A.

Code is available online on: GitHub - hayagamal/Linguistic-Text-Steganography-Using-Transformer-Models

Rating	***Description***
5	Very Good
4	Good
3	Barely Acceptable
2	Poor
1	Very Poor

Table 3: Ratings explanations given in the human evaluation

4.3 Payload Capacity

Besides using Huffman coding in compressing the bit sequence of secret message so to consume the least capacity as possible, an experiment was done with different lengths of secret messages being encoded within different lengths of cover texts on edit-based approaches to get to know how much of payload capacity which is number of bits of secret message is being fully encoded within the steganographic text resulted. Moreover, to evaluate how efficient is each model in encoding the longest secret message within the least payload capacity. Here is the parameters specified for the experiment.

Secret Message	**Content**
Long Secret Message	Meet me at Downtown at 9
Short Secret Message	Help

Table 4: Content Hyperparameters for Secret Messages

Cover Text	**Size**
Long Cover Text	~35-40 tokens
Short Cover text	~10-15 tokens

Table 5: Size Hyperparameters for Cover texts

Experiments were done considering the same hyperparameters of masking interval = 3 and probability distribution threshold = 0.01. Since GPT2 is not restricted by a cover text and masking specific tokens, so it is not restricted by a specific length to encode a secret message, whatever the length of it is, therefore, it can encode as long a secret message as required. However, in edit-based approaches, the capacity load per experiment will be calculated as:

$$\frac{Total\ number\ of\ tokens\ in\ steganographic\ text}{total\ number\ of\ masked\ tokens}$$

Lengths Experiments
Short secret message within short cover text
Short secret message within long cover text
Long secret message within short cover text
Long secret message within long cover text

Table 6: Size Hyperparameters for Cover texts

Masking Capacity Experiments
30% Masking capacity at masking interval = 2
20% Masking capacity at masking interval = 3
10% Masking capacity at masking interval = 4

Table 7: Size Hyperparameters for Cover texts

5 Results

5.1 Human Evaluation Results

Histograms were implemented showing the Human evaluation results on

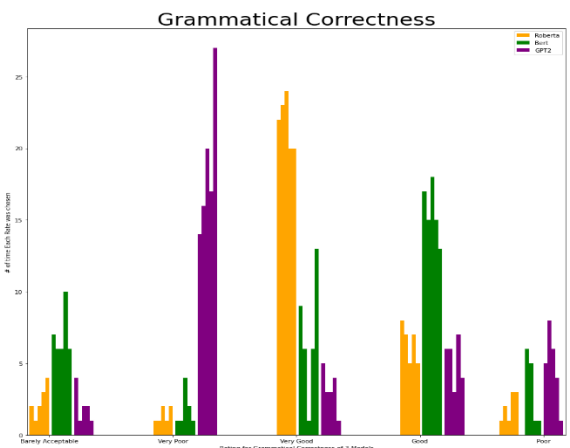

Figure 4: Human Evaluation Grammatical Correctness results

The first histogram is showing each question's evaluation for the three models, considering the labels above and the bars, it is shown that GPT2 shows the poorest grammatical correctness out of the three models as approximately 20 participants out of 34 rated the GPT2 results of 'very poor' grammatical correctness, while RoBERTa shows the most grammatically correct steganographic texts as between 20 to 25 rates out of 34 agreed that RoBERTa's stego results are of very good grammar structure. while BERT resulted fairly good grammatically correct steganographic text according to our participants.

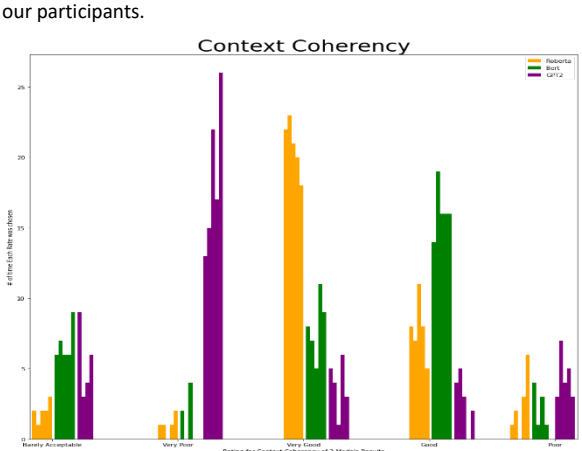

Figure 5: Human Evaluation Context Coherency results

The second histogram is showing each question's evaluation for the three models, considering the labels above and the bars, it is shown that GPT2 shows the poorest context coherency out of the three models as approximately between 15 to 25 participants out of 34 rated the GPT2 results of 'very poor' context coherency, while RoBERTa shows the most context coherent steganographic texts as between 20 to 25 rates out of 34 agreed that RoBERTa's stego results are of very good context coherency leaving BERT on average results.

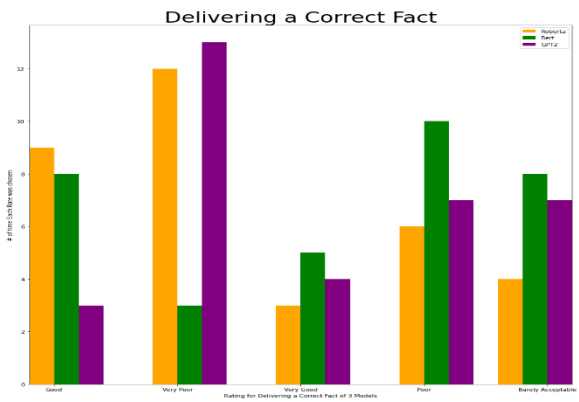

Figure 6: Human Evaluation Correct Fact Delivery results

One of the important factors to consider while testing if the model is well implemented is testing it on critical information to see if it delivers correct factors and can understand the criticalness of this data do not lose or deliver it wrongly leading to suspiciousness and corrupting the meaning of important data. The three models shows false political facts while being used in steganographic purposes which shows that transformers regarding of how powerful and efficient language models they are, they still cannot fully understand how critical a data can be to change its context and cause false spreading of news that can not only cause suspicion to the resulted steganographic text but disastrous consequences upon spreading such texts among a large group of people.

5.2 Payload Capacity Results

It was found that the masking interval, which is the masking strategy shared between the sender and receiver, is what decide on how many tokens will be masked within the cover text taking into consideration that BERT roughly masks 15% of cover text tokens as it's the baseline masking language model. Moreover, it was found that when the masking interval increases, the capacity payload of masking decreases but on the other hand the accuracy increases of the resulted steganographic text, as Figure 6 shows.

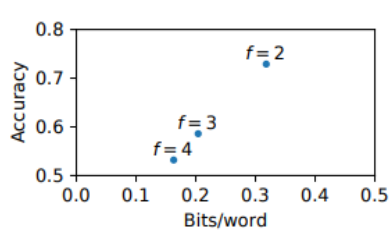

Figure 6: Tradeoff relationship between masking interval ***f*** and masking capacity (Bits/word)

As shown in Appendices ***B***,***C*** and ***D***, RoBERTa has showed success in more experiments in comparison with BERT conducted which are hiding the short secret message completely within both long and short cover text but regarding the long secret message, it was

BERT	RoBERTa
WordPiece Tokenizer	Byte Pairing Encoding
LZ4 is a lossless data compression algorithm that is focused on compression and decompression speed. It belongs to the LZ77 family of byte - oriented compression **algorithm**.	**LINK**4 is a **stateful** data compression algorithm that is focused on compression and decompression **techniques**. It belongs to the LZ77 family of byte-oriented compression **schemes**.

Table 8: Differences of understanding context due to different tokenizers

shown that it needs a very long cover text to be fully encoded inside it. Regarding BERT results for capacity, it only fully encoded the short also compressed secret message within long cover text which shows the limitations of masking process in addition to very low-capacity payload. Moreover, RoBERTa it is obvious that it can take up to masking approximately 30% to encode the whole secret message although it has the same masking interval as BERT which only mask approximately 15% of cover text within the same masking interval. A study was published days ago stating that masking up to 40% of input tokens can outperforms the 15% baseline masking process. But sometimes when highly increasing the masking rate can cause corruption to a large portion of cover text resulting in reducing the context size. However, increasing masking rating means more model predictions also more benefit to training as RoBERTa actually do from the beginning.

5.3 Tokenization

One of the important findings throughout this study is the effect of tokenization process in understanding the tokens, tokenize them then mask the cover text. Byte Pairing Encoding is known as a subword-based tokenization algorithm at which it is known for merging the characters till having several subwords that occurs frequently, so it has learnt a lot of subwords to used it for tokenization the input. As shown in the table above, the word 'compression' and 'losseless' is not sub-word-ly tokenized in BERT, but it is in RoBERTa. This makes the model, in the prediction process, see two tokens holding the meaning of one word so it can be chosen as a [MASK] token in masking process, therefore it would be replaced by a word giving the right context. This shows how the difference in tokenizer can lead to different understanding of the same word thus, different replacements that can give totally different context as shown in Table 8.

6 Conclusion and Future Work

Since Transformer Models achieved a great success in predicting context coherent words to their previous sequence, it would be a great approach to be used to generate steganographic text in a linguistic mean. With advances in neural language models (LMs), edit-based approaches have been replaced by generation-based ones. In Generation-based approach, bit chunks of secret message that has been converted using Huffman encoding, a data compression algorithm very useful for reducing capacity payload so that a cover medium can encode as many bits as possible of secret message. Those bit chunks are converted into their decimal equivalence and chooses the word/candidate whose index equals to this decimal equivalent over the candidates list of highest probability distributions for next predicted word estimated by the LM, yielding impressive payload capacities of 1–5 bits per word (Shen et al., 2020). However, it remains challenging for this LM to generate so genuine-looking texts that they fool both humans and machines. Two Edit-based approaches BERT and RoBERTa with comparison with one Generation-based approach GPT2 is proposed precisely in terms of context coherency, capacity payload and unsuspiciousness to decide which one is best fit for steganographic purposes assessing the previously mentioned terms by using human evaluation and payload capacity experiments. Generation-based stego texts were easily detectable due to its poor context coherency, and grammatically incorrectness. RoBERTa has shown best results in all but of lower payload capacity than that of the generation-based method, but it's high for an edit-based method due to its Dynamic masking strategy. One of the unexpected findings on this study was that it is not advisable to use proposed models in political domain due to sensitivity and criticalness of information used as cover. Additionally, the tokenizer is found to be a huge affecting factor in understanding the context correctly to predict fit words. Lastly, masking up to 40% of input tokens can outperforms the 15% baseline masking process. However, more than 40% can cause context corruption of stego text. Regarding future work, Deep testing on each of the three transformer models will be implemented using an enhanced larger collected dataset of different sizes of cover texts and secret message including different domains to test its reliability on outputting a correct steganographic text in other cases. Exploration on new steganographic methods can be done on RoBERTa such as Arithmetic coding that enhance the results. More experiments on tokenizers could be done for better understanding for sub-words. A common tokenizer is to be implemented from scratch to be trained so it can be used to tokenize cover texts input for BERT and RoBERTa to see how each model would react to the new tokenizer and which enhance their results, and which does not.

7 References

M. Agarwal, “Text steganographic approaches: A comparison,” *arXiv.org*, 14-Feb-2013. [Online]. Available: https://arxiv.org/abs/1302.2718. [Accessed: 09-Dec-2021].

“Ah4s: An algorithm of text in text ... - wiley online library.” [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1752. [Accessed: 09-Dec-2021].

"The Transformer Model", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/attention-is-all-you-need-e498378552f9. [Accessed: 10- Jun- 2022]

J. Alammar, “The illustrated GPT-2 (Visualizing Transformer language models),” *The Illustrated GPT-2 (Visualizing Transformer Language Models) – Jay Alammar – Visualizing machine learning one concept at a time.* [Online]. Available: https://jalammar.github.io/illustrated-gpt2/. [Accessed: 09-Dec-2021].

Additional informationNotes on contributorsBarnali Gupta Banik Barnali Gupta Banik, “Novel text steganography using natural language processing and part-of-speech tagging,” *Taylor & Francis*. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/03772063.2018.1491807. [Accessed: 09-Dec-2021].

M. Agarwal, “Text steganographic approaches: A comparison,” *arXiv.org*, 14-Feb-2013. [Online]. Available: https://arxiv.org/abs/1302.2718. [Accessed: 09-Dec-2021].

Fang, T., Jaggi, M. and Argyraki, K., 2022. *Generating Steganographic Text with LSTMs*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1705.10742> [Accessed 10 June 2022].

E. Girling, “Everything GPT-2: 1. architecture overview,” *Medium*, 18-Dec-2020. [Online]. Available: https://rowlando13.medium.com/everything-gpt-2-1-architecture-overview-132d16fe985a. [Accessed: 09-Dec-2021].

“The annotated GPT-2,” *Committed towards better future*, 18-Feb-2020. [Online]. Available: https://amaarora.github.io/2020/02/18/annotatedGPT2.html. [Accessed: 09-Dec-2021].

Medium. 2022. *Masked-Language Modelling With BERT*. [online] Available at: <https://towardsdatascience.com/masked-language-modelling-with-bert-7d49793e5d2c> [Accessed 10 June 2022].

Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2022. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1810.04805> [Accessed 10 June 2022].

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.,

2022. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1907.11692> [Accessed 10 June 2022].

Wettig, A., Gao, T., Zhong, Z. and Chen, D., 2022. *Should You Mask 15% in Masked Language Modeling?*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2202.08005> [Accessed 10 June 2022].

Ueoka, H., Murawaki, Y. and Kurohashi, S., 2022. *Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2104.09833> [Accessed 10 June 2022].

"Byte-Pair Encoding: Subword-based tokenization algorithm", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/byte-pair-encoding-subword-based-tokenization-algorithm-77828a70bee0. [Accessed: 12- Jun- 2022].

B. Zain, *Cs.uwaterloo.ca*, 2022. [Online]. Available: https://cs.uwaterloo.ca/~mli/Bin.pptx. [Accessed: 12- Jun- 2022].

"Deconstructing BERT, Part 2: Visualizing the Inner Workings of Attention", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1. [Accessed: 12- Jun- 2022].

8 Appendices

Appendix A

1r. The Fund praised the protectionists to support economic security in order to confront possible economic repercussions from the Russian-Ukrainian sanctions, asserting its unconditional support for Armenia to complete the economic reform project. *

	Very Poor	Poor	Barely Acceptable	Good	Very Good
Context Coherency (logical or consistent structure and meaning)	○	○	○	○	○
Grammatical Correctness	○	○	○	○	○
Delivering a correct fact	○	○	○	○	○

Appendix B

30% Masking Capacity (**masking interval = 2**) – Most Accurate		
Experiment 1	BERT	RoBERTa
Long Cover, Long Secret Message	~31% of secret message is encoded	~38% of secret message is encoded
Long Cover, **Short** Secret Message	All secret message is encoded	All secret message is encoded
Short Cover, Long Secret Message	~14% of secret message is encoded	~18% of secret message encoded
Short Cover, **Short** Secret Message	All secret message is encoded	All secret message is encoded

Table 8: Payload Capacity Experiment 1 Results

Appendix C

20% Masking Capacity (**masking interval = 3**) – Accurate		
Experiment 2	BERT	RoBERTa
Long Cover, Long Secret Message	~24% of secret message is encoded	~28% of secret message is encoded
Long Cover, **Short** Secret Message	All secret message is encoded	All secret message is encoded
Short Cover, Long Secret Message	~10% of secret message is encoded	~14% of secret message encoded
Short Cover, **Short** Secret Message	~65% of secret message is encoded	~69% of secret message is encoded

Table 9: Payload Capacity Experiment 2 Results

Appendix D

10% Masking Capacity (**masking interval = 4**) – Least Accurate		
Experiment 3	BERT	RoBERTa
Long Cover, Long Secret Message	~19% of secret message is encoded	~24% of secret message is encoded
Long Cover, **Short** Secret Message	~88% of secret message is encoded	All secret message is encoded
Short Cover, Long Secret Message	~8-9% of secret message is encoded	~13% of secret message encoded
Short Cover, **Short** Secret Message	~53% of secret message is encoded	~61% of secret message is encoded

Table 10: Payload Capacity Experiment 3 Results