A Novel Approach for Semantic Pure Steganography

João Teixeira Figueira
Instituto Superior Técnico, Lisboa

***Abstract*—Steganography is the practice of concealing a message within some other carrier or cover message. It is used to allow the sending of hidden information through communication channels where third parties would only be aware of the explicit information in the carrier message. In this article propose a novel approach for text steganography that can be classified as pure steganography. The proposed algorithm uses the redundancy in language semantics as the space for the hidden message. It improves on existing algorithms by not requiring the message receiver to be aware of the specific redundancies of any cover message. We contextualize our system by thoroughly reviewing semantic steganography and the concepts surrounding it, and by surveying published systems in this area. Our results show that a semantic pure steganographic system is possible and can realistically be used, despite being limited by very low embedding rates.**

***Index Terms*—Steganography, Linguistic, Semantics, Monte Carlo**

I. Introduction and Background

Steganography systems describe methods for taking an ”innocuous” message, called *covertext* and embed it with some *plaintext* message that is desired to remain hidden, outputting a *stegotext*. This *stegotext* is a slightly altered version of the *covertext* that is still ”innocuous” and from which the *plaintext* is extractable. Effectively, steganography is the process of encrypting a message and having the output appear to be a non-encrypted message.

In certain contexts, a communication channel provider might refuse to relay messages that it can see are encrypted and does not know are trustworthy. Steganography finds applications in these situations and will continue to do so with the growing threat of mass surveillance.

In this article we propose a novel system for semantic steganography, which is the branch of text steganography that uses redundancies in the vocabulary of natural languages as the space for the *plaintext* message [1]. To do so, we first provide the necessary context by first conciliating and providing available information on semantic steganography, and then by surveying existing systems for semantic steganography.

A. The Steganographic Process

As described by Kingslin in [1], a steganography system can be divided into two components:

- An embedding or injection method, where a *covertext* is modified to receive the *plaintext*, outputting the *stegotext*. This is performed by the message sender.
- An extraction method, where the *plaintext* is extracted from the *stegotext*. This is done by the message receiver.

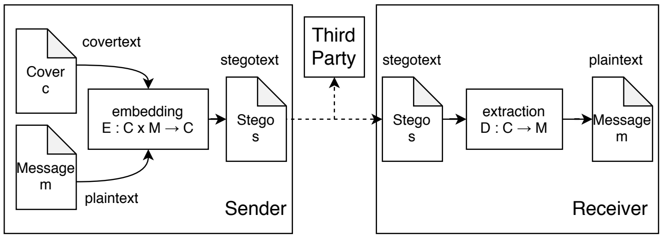

Fig. 1. Diagram showing a generic setting for steganography.

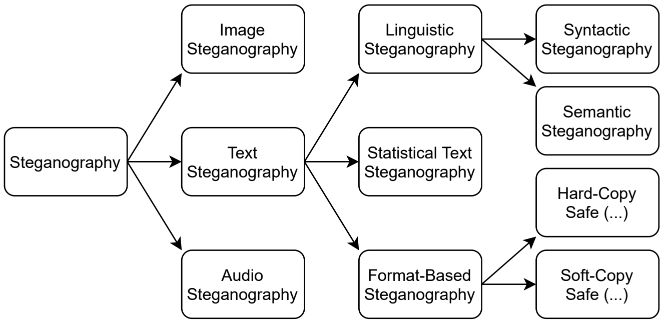

Fig. 2. Diagram showing the proposed hierarchy of the major families of steganographic systems.

A diagram explaining the usage of these two functions to hide and send messages can be seen in Figure 1.

Most Steganography methods can embed a hidden message of any nature into a cover message of a specific nature, *i.e.* the embedding and extraction functions will be constructed for the specific given source of covers [2]. As such, Steganography methods are usually classified according to the type of cover message they work with [1], [3]–[5].

Text steganography is the family of steganographic systems that use text as the cover message. It is often considered the most difficult type of steganography. As Sharma describes it in [6], this is because a text file lacks a large scale redundancy of information in comparison to the other digital media formats. In this section we provide and describe a hierarchy for the classification of text steganography systems. A diagram of this hierarchy can be seen in Figure 2.

B. Format-Based Text Steganography

Text steganography systems that alter the formatting of text are called format-based steganography systems. Altering the formatting of the text might involve operations such as slightly altering the size or color of letters, moving words or sentences a few millimetres, or even adding extra spaces between words [1], [3]–[5], [7]. These systems are the most commonly used for text steganography.

In [7], Bender states that these systems can be further divided into two categories: ”Soft-copy safe systems”, which are the systems in which the hidden message is not lost if the text is copied onto a different file, these include the insertion of spaces between words; And ”Hard-copy safe systems” which are systems in which the text formatting is closely related to the specific file format of the text, in these systems the hidden message is likely to be lost if the text is copied onto some other file, Bender [7] described that these systems can be treated as a ”highly structured image”.

C. Statistical Text Steganography

Statistical text steganography, often also called random text steganography [1], [3], [8], is the branch of text steganography that deals with hiding information in statistical properties of the *covertexts*. To achieve this, most statistical steganographic systems usually deal with generating the *stegotext* itself (a process mentioned in Section I-E). The *stegotext* is generated in such a way that the desired statistical properties of the text are verified.

D. Linguistic Text Steganography

Text steganography systems that deal with the linguistic properties of the *covertext* are called linguistic steganographic systems [1], [3], [4]. These systems perform modifications on the text itself and exploit the ambiguities or redundancies of natural languages.

As described by Kinglslin [1] and Singh [4], the family of linguistic steganography systems can be further divided according to which linguistic properties of the text are being used to embed the *plaintext*. As such, the following two subfamilies of linguistic steganography can be formalized:

- **Syntactic Text Steganography** Linguistic steganography systems that deal with the syntax of text are called syntactic text steganography systems. Such systems might change the grammatical structures of sentences to embed a hidden message. Simpler systems in this family might simply add or remove commas from text in places where their necessity is arguable (such as the Oxford comma).
- **Semantic Text Steganography** Semantic text stenography is the branch of text steganography that uses the redundancy of words as the space for the hidden message. Steganographic systems in this family rely on the natural redundancy and ambiguity of natural languages.

E. Classifications for Embedding Functions

The embedding and extraction functions are the defining element of a steganographic system. As inverse functions, these two methods are co-dependant and need to be jointly defined. For their relevance, steganographic systems can be classified according to the working principles of these functions. The following classifications where proposed by Kaufmann in [2]:

- **Steganography by Cover Modification** Steganography systems in which the embedding function alters an existing *covertext* are called steganography by cover modification. This is the most common working principle of steganographic systems and is the one shown in Figure 1. In [9], Osman considers that this category is further dividable into substitution-based systems, in which parts of the cover message are replaced; and injection-based systems, in which new elements are inserted into the cover message.
- **Steganography by Cover Synthesis** The generation of a *stegotext* based on the *plaintext* is called steganography by cover synthesis (or generation). This type of steganography can be seen as difficult as it might be hard to generate a cover message that is natural and innocuous.
- **Steganography by Cover Lookup** Steganography by cover lookup describes steganographic systems in which the cover messages are preexisting and not modified in any way. In these systems, the message sender will use the extraction function on all available cover messages and choose the one that produces the desired *plaintext*.

F. Purity of Steganographic Systems

Steganographic systems can be classified according to the requirement of prior information exchange. Usually this information exchange relates to some security measure of the steganographic system, such as a secret key. More generically, what needs to be exchanged are some additional parameters that are needed in both the embedding and extraction methods. Classifying steganographic systems based on this is relevant, as the prior exchange of information might not be always feasible.

The following are the three classifications most commonly discussed when studying this property of steganographic systems [10]–[12]:

- **Pure Steganography** A pure steganographic system, as formalized by Katzenbeisser in [10], is a steganographic system that does not require the prior exchange of some secret information. These systems are solely secured by the iniquity of the *stegotext* and rely third parties not being aware that there exists some hidden message [11].
- **Secret Key Steganography** A secret key steganographic system is defined as a system that requires the prior sharing of a secret key. This secret key, often called *stego-key* [11], is required as an additional parameter in the embedding and extraction functions. Secret key steganography is closely related to symmetric cipher encryption.
- **Public Key Steganography** Public key steganographic systems take concepts from public key cryptography for added security. These systems require the usage of two keys, one public and one secret. The message receiver will generate both keys using some key generation function and will place the public key in some publicly available source. The public key is then used by the message sender in the embedding function to generate

TABLE I
EXAMPLE OF A SHORT SYNONYM TABLE, USED IN [7].

big	large
small	little
chilly	cool
smart	clever
spaced	stretched

the *stegotext*. To extract the *plaintext* from the message, the original secret key needs to be used in the extraction function.

G. On The Security of Steganography Systems

The primary objective of any steganography system is to provide a hidden channel for communication, such that third parties can intercept the cover messages and not be suspicious that these messages are carrying a hidden embedded message. Some third parties, might, however, be aware of the possibility of usage of steganography in a certain communication channel. In this situation, they might use the extraction functions of some steganography systems to "screen" messages for possible hidden embedded messages. Because of this, it is always ideal to first encrypt the hidden message by using, for example, some simple symmetric-key encryption algorithm. If the extraction function can be used on any cover message and have some output, the natural randomness of some *covertext* should be indistinguishable from the *ciphertext* produced by some cryptosystem [10].

II. RELATED WORK

In our research, the following systems for semantic steganography were surveyed and are here described.

A. Synonym Table Steganography Systems

Semantic steganography systems use the redundancy in the words of natural languages as the space for a hidden message. The most trivial implementation of such a system would be one that replaces words in the *covertext* with their synonyms. In our survey, the majority of such systems make usage of a synonym table (exemplified in Table I) that is shared between the message sender an receiver. These tables, of usually two columns, pair words with their synonyms.

In these systems, the hidden message is encoded into the choice of synonyms that was used in the *covertext*. This way, each word in the *covertext* (that can be replaced by a synonym) will encode a character of the *plaintext*, corresponding to which column of the synonym table it is in.

In the approaches described by Bender [7], Rafat [13], and Shirali-Shahreza [14], [15], the *plaintext* is first converted into a binary string. This way, a two-column synonym table can be used to encode the hidden message (there is one column for each character of the hidden message alphabet $\Sigma = \{0, 1\}$).

In all of these systems [7], [13]–[15], the embedding method functions as follows, for a given *covertext* and *plaintext*:

1) The *plaintext* is converted into an alphabet Σ such that $|\Sigma| = c$, where c is the number of columns in the synonym table.
2) The *covertext* is scanned and occurrences of words in the synonym table are identified.
3) The n^{th} identified word of the *covertext* is replaced with a synonym from the table's column corresponding to the n^{th} character of the *plaintext*.

This embedding method is further clarified in Figure 3.

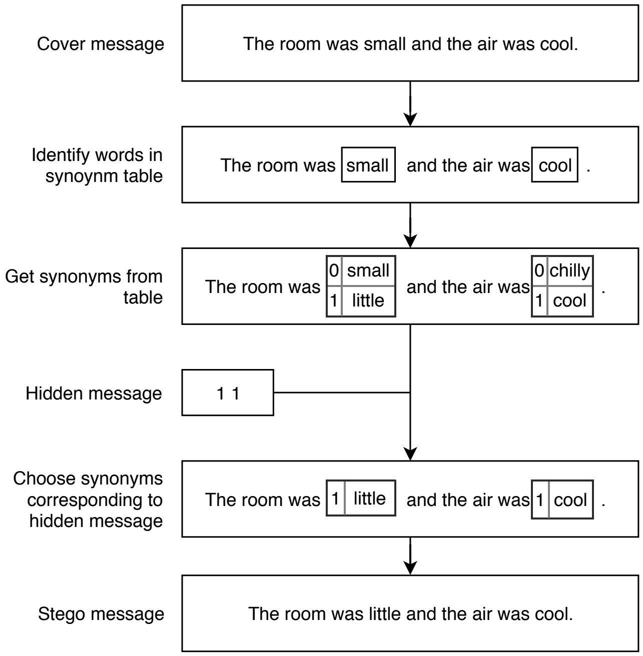
Cover message
The room was small and the air was cool.
Identify words in synoynm table
The room was small and the air was cool .
Get synonyms from table
The room was 0 small 1 little and the air was 0 chilly 1 cool .
Hidden message
1 1
Choose synonyms corresponding to hidden message
The room was 1 little and the air was 1 cool .
Stego message
The room was little and the air was cool.

Fig. 3. Diagram exemplifying the embedding process of the *plaintext* "11" into the *covertext* "The room was small and the air was cool." using a synonym table steganographic system.

The *stegotext* generated by the message sender using the aforementioned embedding method is sent to the message receiver. Here, the corresponding extraction method is applied. It can be described as follows:

1) The *stegotext* is scanned and occurrences of words in the synonym table are identified.
2) The n^{th} character of the *plaintext* will correspond to the column of the n^{th} identified word of the *stegotext*.

This extraction method is further clarified in Figure 4.

The authors in [6], [7] explain the usage of these systems in a generic context, without specifying how a synonym table is constructed. It is not entirely trivial how these tables should be constructed. Words that seem synonymous in certain contexts might not be interchangeable in other contexts [7], [16].

In [15], Shirali-Shahreza explored the usage of words that have different spellings in American English and European English (for example, "Candy" and "Sweets"). This work is futher extended by Rafat's research [13].

Acronyms and their unabbreviated counterparts can also be seen as synonyms. In [14], the authors explored the application of the abbreviations and acronyms commonly used in SMS messages for such a system.

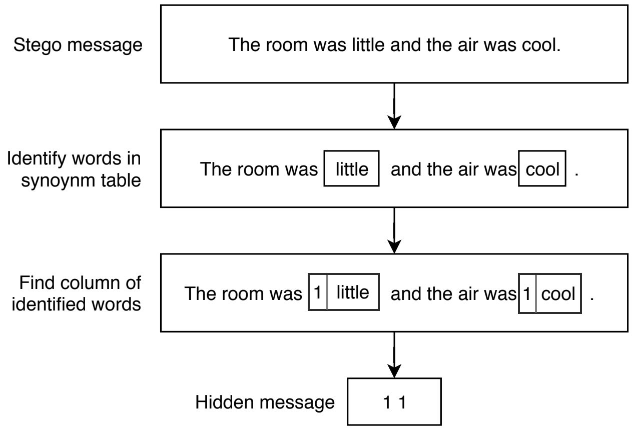
Stego message
The room was little and the air was cool.
Identify words in synoynm table
The room was little and the air was cool .
Find column of identified words
The room was 1 little and the air was 1 cool .
Hidden message
1 1

Fig. 4. Diagram showcasing the extraction process from the *covertext* generated in figure 3.

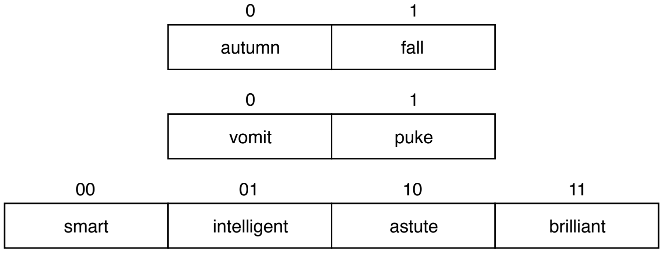

Fig. 5. Examples for sets of synonyms and the bits they can encode, as described by Winstein in [16].

B. Variable Synonym Cardinality Steganography Systems

In the examples described in previous section, the synonym table has a set number of columns and, as such, all words in such a synonym table are restricted to having that set number of possible replacements. Naturally, words can have differing numbers of synonyms. These words have a potential to encode more information that is not being exploited by the described system.

The most trivial solution for this problem is the one described by Winstein [16] in his description for a "naive algorithm". The described approach groups words into sets of mutually interchangeable synonyms. The system embeds a binary message into the *covertext*, each word can embed as many bits as the base two logarithm of the number of words in its synonym set. As such, the number of elements in these sets of synonyms is restricted to being some power of 2. This approach is exemplified in Figure 5. A similar approach is also used in [17] and [18].

C. Winstein's Ideal Coding

In [16], Winstein improves on the aforementioned "naive algorithm" by proposing a related system in which the synonym sets can have any number of words (as opposed to only powers of 2). His proposal consists on converting the hidden message into a *multi-base number* (each digit may have a different base), where each digit corresponds to a word in the synonym table, and the base of each digit is the number of replacements that word can have. This solution can be visualised with the diagram in Figure 6.

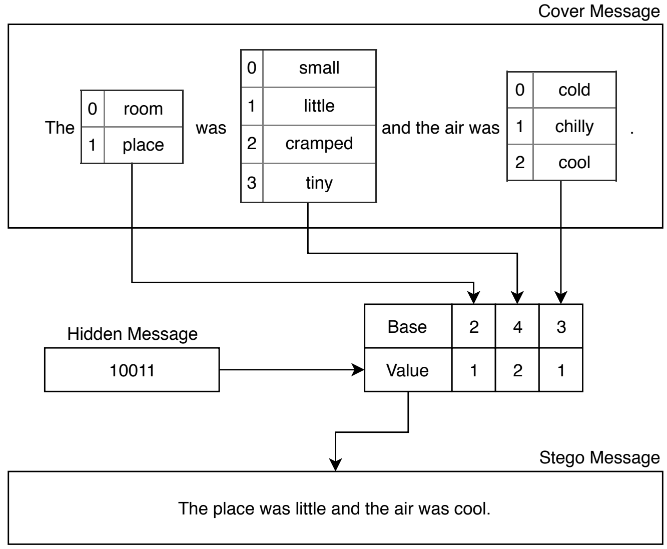

Fig. 6. Diagram showcasing the embedding of the hidden message "10010" into the *covertext* "The room was small and the air was cool.", using the *multi-base number* approach described by Winstein in [16].

D. Mimic Functions

A well known approach for semantic steganography is the one proposed by Wayner in his articles [19] and [20]. Here Wayner described the construction of mimic functions and their applications for text steganography.

A mimic function f is described as the function that alters the statistical properties of a text file A to be the same as some other file B. Formally, if $p(t, A)$ is the probability of a substring t occurring in A, then the mimic function f encodes A so that $p(t, f(A))$ approximates $p(t, B)$.

Wayner introduces mimic functions as the inverse of Huffman compression functions [21].

Wayner improved on his system by joining it with context-free grammars to ensure the sentences maintain grammatical consistency. This improved the iniquity of the *stegotext*, but it still remained mostly devoid of meaning.

E. Markov Chain Based Text Steganography

In [22], Dai introduced the usage of Markov chains for text steganography, this research was continued in [23]. Dai's proposal involves constructing a Markov model for the desired *covertext*.

In Dai's approach, the transitions of the Markov model are labelled with parts of the hidden message. To synthesise the *stegotext*, the *plaintext* is used to determine the sequence of state transitions that is done over the model. This process is exemplified Figure 7.

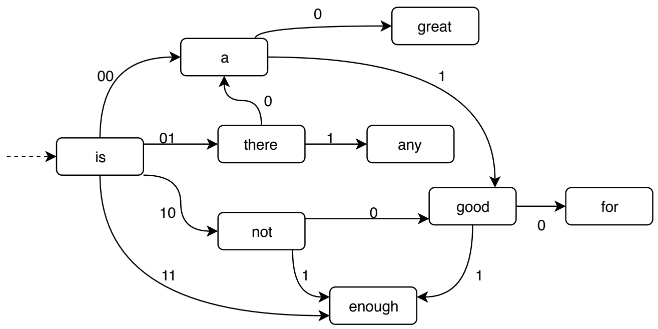

Fig. 7. Diagram of a steganography system constructed using a Markov model, as described in [22].Here, the hidden message ”0100” would synthesize the *stegotext* ”is there a great”.

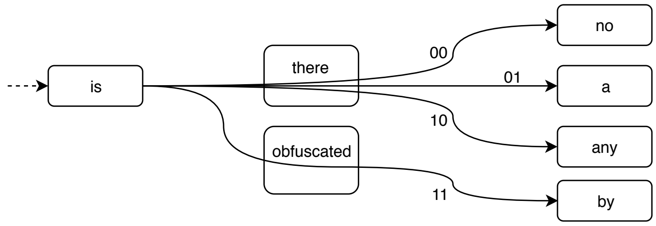

Fig. 8. Diagram of grouped state transitions as described by Moraldo [24]. By grouping two consecutive transitions, the transition ”is there” is made more probable than ”is obfuscated”, as expected in text.

F. Moraldo's Fixed Size Steganography

In [24], Moraldo described how Dai's Markov systems produce ”unnatural” looking text by not taking into account the probability of transitions. With the way that transitions are labelled, any outgoing transition from any given state has the same probability of occurring in a *stegotext*.

Moraldo's solution involves grouping multiple consecutive transitions together and labeling these groups with parts of the hidden message. More probable state transitions will occur in more of these labelled groups. This way, the resulting *stegotext* will have word sequences that occur with the frequency that is expected of a real text. This system is exemplified in Figure 8.

G. Markov Chain with Huffman Coding

In [25], the authors also explore the problem of ensuring a natural probability distribution of transitions on a Markov based steganographic system. For their approach, the authors make use of Huffman coding to construct a tree for the transitions at each step of the Markov model. More frequent transitions are labelled with shorter labels and are thus more likely to appear in the hidden message. This is exemplified in Figure 9. This system shares a lot of similarities with the mimic functions described by Wayner [19] and with Moraldo's approach [24].

III. Approach and Implementation

Semantic steganographic systems that use synonym tables can be thought of as requiring two main operations that are shared by the embedding and extraction functions.

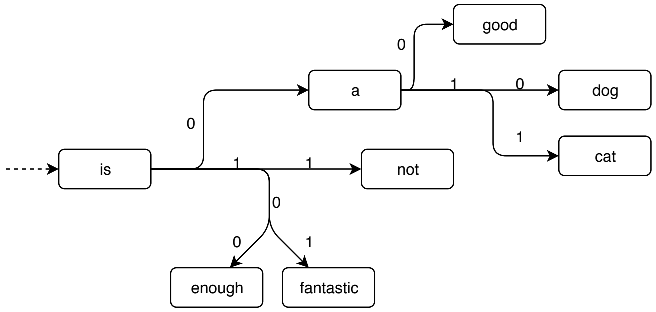

Fig. 9. Diagram exemplifying the usage of ”chained” Huffman trees to label transitions in a Markov model, as described by Zhongliang [25].

- The identification of replaceable substrings in the cover message.
- The labeling of the possible replacements for the identified substrings with characters from the hidden message alphabet.

In synonym table semantic steganography, the hidden message is constructed from the concatenation of the labels of the identified replaceable sections, using the shared synonym table.

In this section, we propose a novel algorithm for semantic steganography that does not require the sharing of a synonym table. To do so, we first define how the two aforementioned operations can be performed in the absence of a shared synonym table.

For the following sections describing our approach, it is assumed that the message sender has access to some unspecified synonym table. This table can have a variable number of synonyms for each word.

A. Replaceable Substring Identification

Our approach to have the message sender and receiver agree on which will be the replaceable sections is the following: The cover message is divided into substrings (or sections) of a fixed size. This fixed size should be large enough to ensure that at least one replaceable substring can be found by the message sender inside the fixed-size sections. If each fixed-size section contains a replaceable substring, then the whole section can be thought of as a replaceable substring.

To try to maintain the replacement sets of each fixed-size section independent from other fixed-size sections, we can define the size of each section as a number of words. This way no word is cut between two sections.

B. Substring Replacement Set Construction

To the possible replacements for each section, the message sender can use his synonym table to identify the replaceable words within each section and then the set of possible replacements for each word. If we are treating the entire fixed-size section as one replaceable substring of the message, we can compute its set of replacements from the Cartesian product of replacement sets of the identified replaceable words. A section will have as many replacements as the product of the sizes

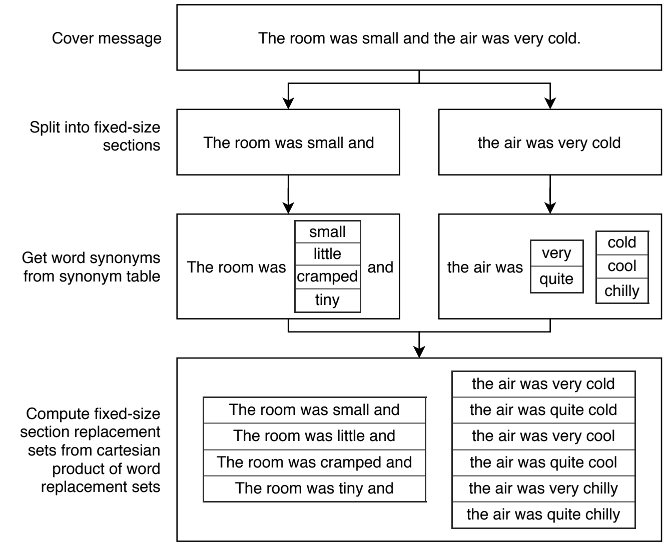

Fig. 10. Diagram showing the process of computing the replacement sets for fixed-size sections of a message. In this example, the size of each section is 5 words.

of replacement sets contained in it. This procedure can be visualised in Figure 10.

C. Section Replacement Labelling

In most synonym table steganographic system, word replacements are labelled according to their column in the synonym table. In our approach, since we are assuming that the message receiver does not have access to the table used by the message sender, an alternative labelling method needs to be implemented.

The proposed solution is as follows: A function that behaves similarly to a hashing function can be used to map fixed-size sections to characters of the hidden message alphabet. This function, which we will refer to as a stego-hashing function, should, for any given fixed-size section, deterministically output a character of the hidden alphabet, with uniform probability over the alphabet.

To construct a stego-hashing function, any existing hashing function that operates on strings can be used. The output of such function just needs to be limited to the size of the hidden message alphabet, this can be done with the modulo operator. As such, if $h : S \to \mathbb{N}$ is an existing hashing function that operates over strings, we can define the stego-hashing function $H(s) = h(s) \bmod |\Sigma|$. The resulting value is used to index a character of the hidden message alphabet.

The message sender can use this stego-hashing function to compute the label for each section replacement of the cover message. Then, the sender can select any replacement of each section that hashes to (has been labelled with) the desired character of the hidden message. The n^{th} character of the hidden message will correspond to the label of the n^{th} selected fixed-size section replacement. This process is exemplified in Figure 11.

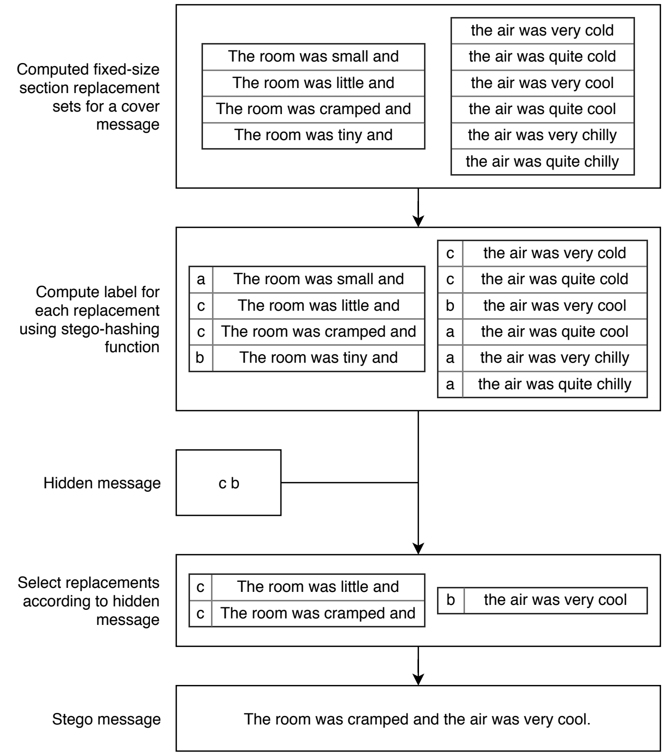

Fig. 11. Diagram showing the embedding process of the proposed approach. This example uses the replacement sets computed for the cover message in Figure 10.

D. Hidden Message Extraction

To extract the *plaintext* from the *stegotext*, the message receiver simply needs to split the *stegotext* into sections and compute the stego-hash for each. The concatenation of these stego-hashes is the *plaintext*. This process does not make use of any synonym table. This is exemplified in Figure 12.

The major advantage of our approach over related systems becomes apparent in the extraction method. The extraction process is very light and does not require the sharing of a synonym table. The message sender and receiver need only to agree on the size of sections and on the used stego-hashing function, which should be encompassable in a very short message.

E. System Implementation

An implementation of this system, as described, was constructed in Java. The system was made so as to allow for an external synonym table to be imported. It is made available, in working condition, on a Github repository [26].

F. Synonym Table Construction

For usage with our implementation of this project, we constructed a synonym table for the English language.

The base for our synonym table is WordNet [27], [28]. WordNet is a large lexical database of English constructed and made available by Princeton University. This database is composed of synsets. Each synset is a set of words labelled

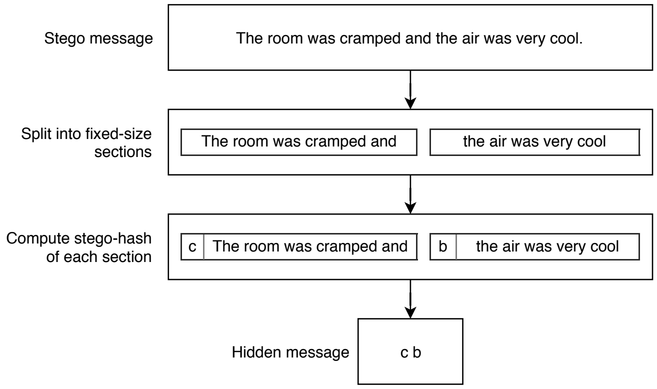

Fig. 12. Diagram showing the extraction process of the proposed approach. This example uses the *stegotext* computed in Figure 11.

by a meaning. Each word in the synset can, in some context, take the meaning of the synset. We firstly constructed a python parser for WordNet.

One limitation of WordNet is that all words are in their basic, non-inflected forms. To account for this, we pluralize the words in noun synsets using the Inflect [29] library, and conjugate the verb synsets with the MLConjug [30] library.

The desired synonym table should map each word to a set of words that can replace it in any context, we define these as the ”safe” replacement words.

To do this, we use the various synsets that a word appears in to identify which words can always replace it, regardless of the meaning that it takes on in a given context. As such, the set of possible replacements for a word can be defined as the intersection of all synsets in which that word appears.

The resulting replacement table is usable in many semantic steganography systems beyond the system proposed in this document and was made openly available on a Github repository [31], along with the code used to create it.

G. Embedding Failure Probability

When the message sender computes the possible replacements for a fixed-size section, there is a probability that none of the replacements will hash to the desired hidden alphabet character. If this happens, the embedding might be considered impossible for that specific *covertext* and *plaintext* pair. Because of this failure probability, the embedding algorithm can be considered a Monte Carlo randomized algorithm. In [2], the concept of embedding effectiveness is described as relating to this probability of an embedding failure.

The message sender and receiver will want to negotiate parameters for the system that minimize this failure probability. As such, it is relevant to estimate it.

In developing this project we have deduced the following formula for the embedding probability for a hidden message character $m \in \Sigma$ on a fixed-size section c with r possible replacements:

$$EP(m,c) = f(r) = 1 - (1 - \frac{1}{|\Sigma|})^r. \tag{1}$$

By applying this formula to all the sections of a message, we get the formula for embedding probability of a pair hidden message $M = \{m_1, m_2, ..., m_n\}$ and cover message $C = \{c_1, c_2, ..., c_n, ...\}$, where the i^{th} section of C has r_i possible replacements:

$$EP(M,C) = \prod_{i=1}^{n}(1 - (1 - \frac{1}{|\Sigma|})^{r_i}). \tag{2}$$

These formulas are useful for computing the embedding probability for a known cover message that has been “pre-processed”. To use these formulas, the number of section replacements needs to be known for each fixed-size section in the cover message.

By computing the expected value for this formula, we can obtain a more “generic” formula for the embedding probability on a cover message. If the individual sections and their replacements are not known, the probability distribution of the number of replacements can instead be studied. The following formula for a lower bound was obtained using Jensen’s inequality [32] and Equation 2:

$$EP(M,C) \geq (\prod_{r=1}^{\infty}(1 - (1 - \frac{1}{|\Sigma|})^r)^{RP(w,r)})^n \tag{3}$$

This formula is more useful in that it can be used to estimate the embedding probabilities for unknown cover messages, based only on their length n and the probability of a section of w words having r replacements, $RP(w,r)$. This probability is very dependant on the synonym table and it is not trivial to compute.

The number of replacements for a section is the number of possible combinations of replaceable words within the section. If we assume that the numbers of alternatives for words in a section are independent or almost independent, each word can be interpreted as one of w independent trials, and that each will have an outcome that is a number $a \in \{1, 2, ..., k\}$ of alternatives with a known probability $\{p_1, p_2, ..., p_k\}$. As such, the numbers of words in that have a replacements follow a multinomial distribution.

To compute the probability that a section of w words may have r replacements, $RP(w,r)$, it is first necessary to determine how r can be described as a multiplication of the possible numbers of alternatives that words might have.

If for some value of $r \in \mathbb{N}$ there exist l sets $E_1, E_2, ..., E_l$, such that each set $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,k}\} \in \mathbb{N}_0^*$ verifies

$$\sum_{j=1}^{k} e_{i,j} = w \quad \text{and} \quad \prod_{j=1}^{k} j^{e_{i,j}} = r. \tag{4}$$

Then the probability that a section of w words may have r replacements can be computed with the multinomial probability mass function, as

$$RP(w,r) = \sum_{i=1}^{l} \frac{w!}{e_{i,1}!e_{i,2}!...e_{i,k}!} p_1^{e_{i,1}} p_2^{e_{i,2}} ... p_k^{e_{i,k}}. \tag{5}$$

TABLE II
Probabilities of a word having n replacements, as sampled from the corpus described in Section IV-A, using the synonym table described in III-F.

Replacements	Probability
not replaceable	96.260%
2	2.040%
3	0.722%
4	0.488%
5	0.186%
6	0.125%
7	0.101%
8	0.044%
9 or more	0.034%

IV. Evaluation

A. Replacement Table

One of our main contributions was the synonym table constructed as described in Section III-F.

The utility of a synonym table is dictated by how frequently it can find a replaceable word, how many replacements can it find, and how natural are the replacements.

Given the way that the synonym table was constructed, it is ensured that it will never replace a word with another that would not be a fit for that context. This has the shortfall that the synonym table is somewhat restricted and will find fewer replaceable words than if this was not verified.

Given that different words appear in text with different frequencies, an appropriate way to extract some first order statistics of the synonym table is to randomly sample words from candidate covertexts and to count the number of replacements found for each. To do this we used a 3 million word subset of the COHA corpus [33], along with a 2 million word subset of the GloWbE corpus [34]. Together, these two corpora provide a very wide and unfocused sample of the English language. Table II shows the probabilities of words being replaceable with a certain number of replacements, as sampled from this corpus.

As is shown, the replacement table can find replacements for about **3.74%** of words randomly sampled from English language texts.

In Section III-G we stated that a multinomial distribution can be used to explain the distribution in the number of replacements of a section, if we assume that the number of alternatives for words in a section are independent from other words. To defend this statement, we compare the probability of a word being replaceable to the probability of consecutive replaceable words. As sampled from the same dataset as described in Section IV-A, each word has a **3.740%** probability of being replaceable, and a word that comes after a replaceable word has a **3.836%** probability of being replaceable itself. These values are very similar and show that the replaceability of a word is very independent from the replaceability of words in its immediate neighbourhood.

B. Steganography System

Given the Monte Carlo nature of our system, the embedding probability is one of the main factors to be evaluated. In [2], embedding effectiveness is named as a measure for this embedding reliability .

Equation 5 provides a formula to compute the probability mass function for the number of replacements that a section is expected to have. To validate this formula, the computed expected probability is compared to the relative frequencies of numbers of section replacements, as sampled from the dataset described in Section IV-A. These results are plotted out in Figure 13.

The predicted probabilities are a very close fit to the measured values, which validates the provided formula. The biggest difference between the projected and measured values is seen for sections that are not replaceable. The formula predicted a **2.210%** frequency for these sections, but a real occurrence rate over the dataset was measured to be **4.358%**. We estimate that this discrepancy is caused by the not perfect independence of word replacements. The dataset might contain areas with ”noisier” text with no replaceable words, these might span multiple sections, each of with will have no replacements, increasing the probability of non-replaceable sections, as measured.

The formula described in Equation 3 provides a lower bound for the expected value of the embedding probability (or effectiveness) of our system. This formula (and the tightness of its bound) can be validated by sampling candidate cover messages and comparing the measured embedding probability to the lower bound given by the formula. These values were computed and are plotted out in Figure 14.

The effect of section size on the embedding probability of messages can also be studied using the described setup. The measured values for embedding probability are compared to the values predicted by Equation 3 and Equation 5 as plotted in Figure 15.

The results shown in Figure 14 show that the provided formula offers a very close lower bound to the real probability values. This allows for this formula to be used as a fully analytical tool for the negotiation of parameters in a system like this. As is clarified in this plot, longer messages have lower embedding probability due to requiring embedding successes over more sections.

In Figure 15 it is shown that, while the predicted values do closely follow the measured probabilities, the expected lower bound can take values that are above the real probabilities. This can be explained by the fact that the formula in Equation 5 predicted a lower frequency for sections with no replaceable words. Because of this, it is shown that, if instead of the values predicted by this formula, the real measurements for the distribution of section replacements are used, the formula will provide a true lower bound for the embedding probability. In this figure the correlation between section size and embedding probability is clarified. Longer section sizes will have higher probability of having more replacements, and provide greater embedding effectiveness.

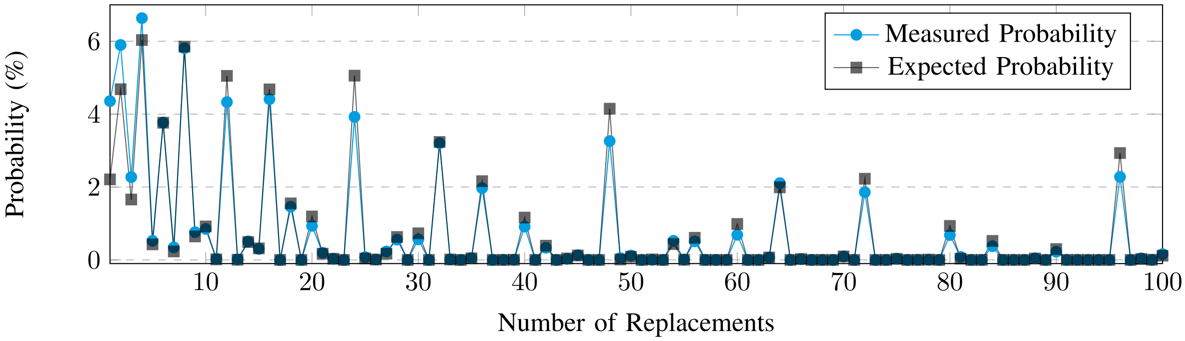

Fig. 13. $RP(w, r)$, Probability distribution for number of replacements, on sections of $w = 100$ words. The probability mass function in Equation 5 is compared to the results obtained from sampling the dataset described in Section IV-A.

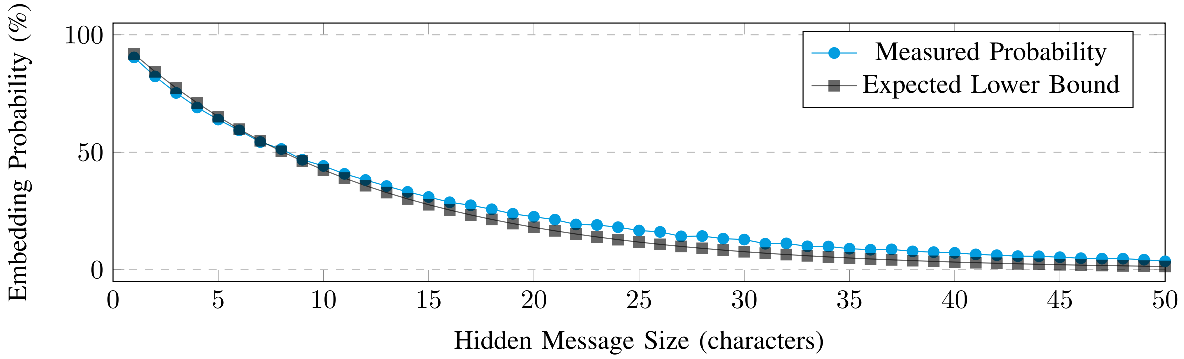

Fig. 14. Probability that a hidden message of size n is embeddable into some cover message. Using sections of 200 words, and assuming a hidden message alphabet of size 29. Values were measured as sampled from the the dataset described in Section IV-A, and are compared to the expected lower bound provided by Equation 3 with the replacement distribution computed with the probability mass function described in Equation 5.

The embedding capacity, or rate, is a very relevant parameter on the selection of steganographic systems. It relates to the amount of information that can be hidden on a certain cover message. For text steganography, this is calculated as the size of the hidden message divided by the size of the cover message. For the provided system, the embedding capacity is $\frac{1}{s}$, where s is the average size, in characters, of sections. We compare the effect of embedding rate on embedding probability with the results exposed in Figure 16.

As is shown, greater values for embedding probability can be obtained by sacrificing the embedding rate, this is done by increasing the section size.

V. Final Remarks

A. Significant Contributions

- Provided conciliation of published knowledge on semantic steganography, including a complete hierarchy of the areas of text steganography that was not found in entirety on a previously published article.
- Surveyed approaches to semantic steganography and provided original diagrams that help expose and simplify their function.
- Introduced and evaluated the first approach for a pure semantic steganography system.
- Constructed synonym table for the English language that can be applied to many steganographic systems beyond our own, and was made publicly available.

B. Comments on the System

In our evaluation of this system, we found that there are some properties to it that might limit its applicability in real life situations.

To ensure a reliable probability of embedding success, we found that cover messages would have to be divided into sections of at least 200 to 300 words. Each of these sections can encode a single character of the hidden message. This implies that cover messages for usage as input to our implementation of the system have to be extremely long documents. The most viable option for these, if the cover messages are not written from scratch, is that documents like books are used. A security risk with the usage of these cover messages is that, if the original document is publicly available, a third party might find compare the original document with the resulting covertext and identify that it has been tampered with, nullifying the point of using steganography in the first place.

The main cause for the low embedding rate are the low degrees of freedom for modifying the covertext. These degrees of freedom are dictated from the possible replacements of words as provided by the replacement table. Our synonym

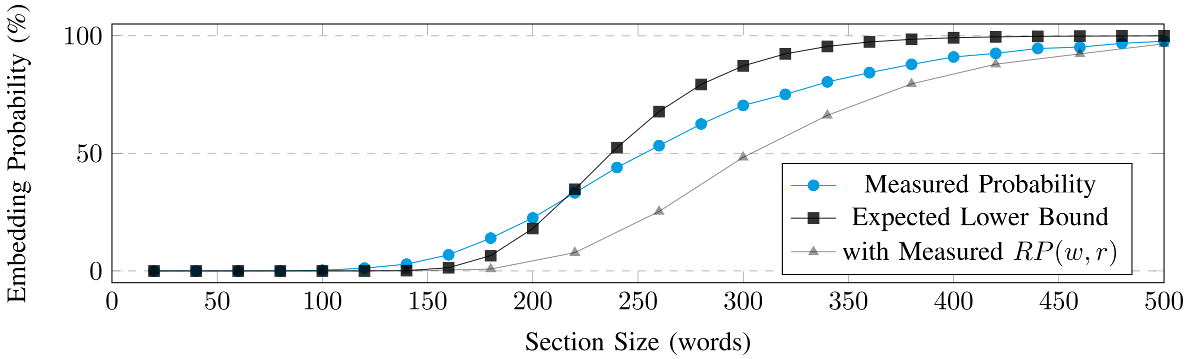

Fig. 15. Probability that a hidden message of 20 characters is embeddable into some cover message. Using a hidden message alphabet of size 29. Values were measured as sampled from the the dataset described in Section IV-A. The lower bounds were computed using the formula described in Equation 3, one used the replacement distribution $RP(w, r)$ from Equation 5, the other used the values for $RP(w, r)$ as measured and shown in Figure 13.

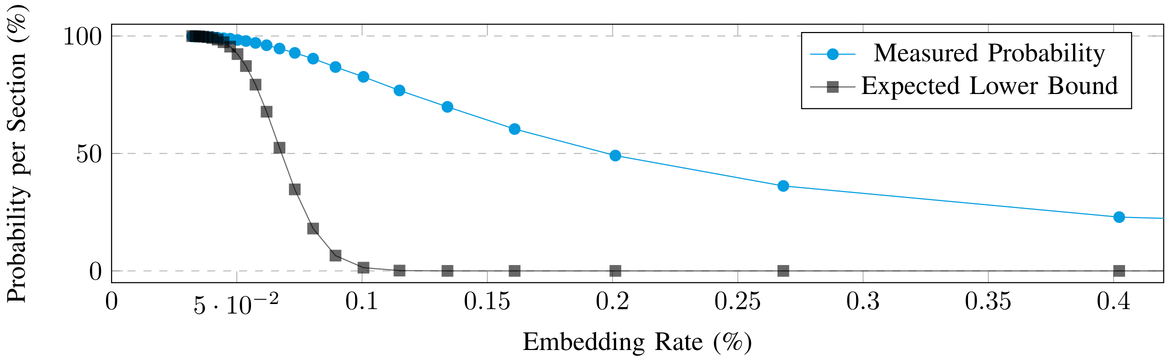

Fig. 16. Embedding probability for one section, assuming a hidden message alphabet of size 29. Values were measured as sampled from the the dataset described in Section IV-A. The expected lower bound was computed using the formula in Equation 3.

table has the low rate of only finding replacements for **3.74%** of words randomly sampled from English texts. Greater rates of replacements would result in greater embedding rates for the system, but a different approach to construct a synonym table might be necessary.

C. Future Improvements

As the first system for semantic pure steganography, our proposed approach has potential for further developments. Some possible improvements are listed here.

A significant improvement in the construction of the synonym table would come from building a system that could identify the meaning being taken on by a word and identify the correct synset from which to get the possible synonyms. This would largely increase the number of possible replacements per word and the number of replaceable words, which would, in turn, result in a system with greater embedding rate.

Given these recent developments in machine learning for natural language, there are now systems that can rewrite sentences with different wording. One such system is the one developed by Xu [35] which uses the BERT deep learning model [36]. A system like this one could be used to replace the synonym table altogether by listing the possible rewritings of a sentence.

If the hashing stage of our system would also take in information from previous sections, then the past choices of replacements would have an effect on the result of other hashings. This way, if there is an embedding failure on a section with fewer replacements, then the system could backtrack to a section with multiple choices for viable replacements and change the selection. This would effectively perform a tree search algorithm over the replacements of sections. This modification could greatly improve the embedding rate.

VI. Conclusion

With this article we have provided the groundwork for constructing a system for semantic pure steganography, and have implemented it and thoroughly analysed its properties. In doing this, we have made multiple contributions to the field of steganography and more specifically semantic and text steganography. These contributions go beyond our implementation of the system itself.

Despite the downfalls of our system, we see that, as a whole, it is an important step in the development of future systems for semantic pure steganography, and that, with further developments such as the ones described in Section V-C, it can become a strong tool for truly innocuous communication.

References

[1] S. Kingslin and N. Kavitha, “Evaluative approach towards text steganographic techniques,” *Indian Journal of Science and Technology*, vol. 8, 11 2015.

[2] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[3] M. Agarwal, “Text steganographic approaches: A comparison,” *International Journal of Network Security and Its Applications*, vol. 5, 02 2013.

[4] H. Singh, “Analysis of different types of steganography,” *International journal of scientific research in science, engineering and technology*, vol. 2, pp. 578–582, 2016.

[5] M. Nosrati, R. Karimi, and M. Hariri, “An introduction to steganography methods,” *World Applied Programming*, vol. 1, pp. 191–195, 08 2011.

[6] S. Sharma, A. Gupta, M. C. Trivedi, and V. K. Yadav, “Analysis of different text steganography techniques: A survey,” in *2016 Second International Conference on Computational Intelligence Communication Technology (CICT)*, 2016, pp. 130–133.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313–336, 1996.

[8] S. Roy and M. Manasmita, “A novel approach to format based text steganography,” 01 2011, pp. 511–516.

[9] B. Osman, A. Yasin, and M. Omar, “An analysis of alphabet-based techniques in text steganography,” vol. 8, pp. 109–115, 01 2016.

[10] S. Katzenbeisser and F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, 1st ed. USA: Artech House, Inc., 2000.

[11] S. H. Abdullah, “Steganography methods and some application(the hidden secret data in image),” 2009.

[12] B. A. Sheelu, “An overview of steganography,” *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 11, 2013.

[13] K. F. Rafat, “Enhanced text steganography in sms,” in *2009 2nd International Conference on Computer, Control and Communication*, 2009, pp. 1–6.

[14] M. Shirali-Shahreza and M. H. Shirali-Shahreza, “Text steganography in sms,” in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, 2007, pp. 2260–2265.

[15] M. H. Shirali-Shahreza and M. Shirali-Shahreza, “A new synonym text steganography,” in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2008, pp. 1524–1526.

[16] K. Winstein, “Lexical steganography through adaptive modulation of the word choice hash,” 1998.

[17] M. Chapman, G. I. Davida, and M. Rennhard, “A practical and effective approach to large-scale automated linguistic steganography,” in *Information Security*, G. I. Davida and Y. Frankel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 156–165.

[18] H. Huanhuan, Z. Xin, Z. Weiming, and Y. Nenghai, “Adaptive text steganography by exploring statistical and linguistical distortion,” in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, 2017, pp. 145–150.

[19] P. Wayner, “Mimic functions,” *Cryptologia*, vol. 16, no. 3, pp. 193–214, 1992.

[20] ——, “Strong theoretical stegnography,” *Cryptologia*, vol. 19, no. 3, pp. 285–299, 1995.

[21] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[22] W. Dai, Y. Yu, and B. Deng, “Bintext steganography based on markov state transferring probability,” 11 2009, pp. 1306–1311.

[23] W. Dai, Y. Yu, Y. Dai, and B. Deng, “Text steganography system using markov chain source model and des algorithm,” *JSW*, vol. 5, pp. 785–792, 07 2010.

[24] H. Moraldo, “An approach for text steganography based on markov chains,” 09 2014.

[25] Z. Yang, S. Jin, Y. Huang, Y. Zhang, and H. Li, “Automatically generate steganographic text based on markov model and huffman coding,” 2018.

[26] J. Figueira, “A novel system for semantic steganography,” https://github.com/joaoperfig/semsteg.

[27] G. A. Miller, “Wordnet: A lexical database for english.” *Communications of the ACM*, vol. 38, 1995.

[28] C. Fellbaum, “Wordnet: An electronic lexical database.” 1998.

[29] J. R. Coombs, “Inflect,” https://github.com/jaraco/inflect.

[30] S. Diao, “Mlconjug,” https://github.com/SekouD/mlconjug.

[31] J. Figueira, “A wordnet replacement table,” https://github.com/joaoperfig/semsteg/tree/main/wordnet_parser.

[32] R. McEliecen, *The Theory of Information and Coding*. Cambridge University Press, 1977.

[33] M. Davies, “Corpus of historical american english (coha),” https://www.english-corpora.org/coha/, 2010.

[34] ——, “Corpus of global web-based english (glowbe),” https://www.english-corpora.org/glowbe/, 2013.

[35] L. Xu, I. Ramirez, and K. Veeramachaneni, “Rewriting Meaningful Sentences via Conditional BERT Sampling and an application on fooling text classifiers,” *arXiv e-prints*, p. arXiv:2010.11869, Oct. 2020.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv e-prints*, p. arXiv:1810.04805, Oct. 2018.