# Practical Linguistic Steganography using Contextual Synonym Substitution and Vertex Colour Coding

**Ching-Yun Chang**
University of Cambridge
Computer Laboratory
Ching-Yun.Chang@cl.cam.ac.uk

**Stephen Clark**
University of Cambridge
Computer Laboratory
Stephen.Clark@cl.cam.ac.uk

## Abstract

Linguistic Steganography is concerned with hiding information in natural language text. One of the major transformations used in Linguistic Steganography is synonym substitution. However, few existing studies have studied the practical application of this approach. In this paper we propose two improvements to the use of synonym substitution for encoding hidden bits of information. First, we use the Web 1T Google n-gram corpus for checking the applicability of a synonym in context, and we evaluate this method using data from the SemEval lexical substitution task. Second, we address the problem that arises from words with more than one sense, which creates a potential ambiguity in terms of which bits are encoded by a particular word. We develop a novel method in which words are the vertices in a graph, synonyms are linked by edges, and the bits assigned to a word are determined by a vertex colouring algorithm. This method ensures that each word encodes a unique sequence of bits, without cutting out large number of synonyms, and thus maintaining a reasonable embedding capacity.

## 1 Introduction

Steganography is concerned with hiding information in a cover medium, in order to facilitate covert communication, such that the presence of the information is imperceptible to a user (human or computer). Much of the existing research in steganography has used images as cover media; however, given the ubiquitous nature of electronic text, interest is growing in using natural language as the cover medium. Linguistic Steganography—lying at the intersection of Computational Linguistics and Computer Security—is concerned with making changes to a cover text in order to embed information, in such a way that the changes do not result in ungrammatical or unnatural text.

A related area is natural language watermarking, in which changes are made to a text in order to identify it, for example for copyright purposes. An interesting watermarking application is "traitor tracing", in which documents are changed in order to embed individual watermarks. These marks can then be used to later identify particular documents, for example if a set of documents—identical except for the changes used to embed the watermarks— have been sent to a group of individuals, and one of the documents has been leaked to a newspaper.

In terms of security, a linguistic stegosystem should impose minimum embedding distortion to the cover text so that the resulting stegotext in which a message is camouflaged is inconspicuous, resulting in high *imperceptibility*. In addition, since steganography aims at covert communication, a linguistic stegosystem should allow sufficient embedding capacity, known as the *payload*. There is a fundamental tradeoff between imperceptibility and payload, since any attempt to embed more information via changes to the cover text increases the chance of introducing anomalies into the text and therefore raising the suspicion of an observer.

A linguistic transformation is required to embed information. Transformations studied in previous work include lexical substitution (Chapman and Davida, 1997; Bolshakov, 2004; Taskiran et al., 2006; Topkara et al., 2006b), phrase paraphrasing (Chang and Clark, 2010), sentence structure manipulations (Atallah et al., 2001a; Atallah et al., 2001b;



*Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1194–1203,
MIT, Massachusetts, USA, 9-11 October 2010. 

Liu et al., 2005; Meral et al., 2007; Murphy, 2001; Murphy and Vogel, 2007; Topkara et al., 2006a) and semantic transformations (Atallah et al., 2002; Vybornova and Macq, 2007). Many of these transformations require some sophisticated NLP tools; for example, in order to perform semantic transformations on text, word sense disambiguation, semantic role parsing and anaphora resolution tools may be required. However, the current state-of-the-art in language technology is arguably not good enough for secure linguistic steganography based on sophisticated semantic transformations, and the level of robustness required to perform practical experiments has only just become available. Hence many existing linguistic stegosystems are proof-of-concept implementations with little practical evaluation of the imperceptibility or payload.

### 1.1 Synonym substitution

Synonym substitution is a relatively straightforward linguistic steganography method. It substitutes selected words with the same part of speech (PoS) synonyms, and does not involve operating on the sentence structure so the modification can be guaranteed to be grammatical. Another advantage of this method is that many languages are profuse in synonyms, and so there is a rich source of information carriers compared with other text transformations.

There are two practical difficulties associated with hiding bits using synonym subsitution. The first is that words can have more than one sense. In terms of WordNet (Fellbaum, 1998), which is the electronic dictionary we use, words can appear in more than one synset. This is a problem because a word may be assigned different bit strings in the different synsets, and the receiver does not know which of the senses to use, and hence does not know which hidden bit string to recover. Our solution to this problem is a novel vertex colouring method which ensures that words are always assigned the same bit string, even when they appear in different synsets.

The second problem is that many synonyms are only applicable in certain contexts. For example, the words in the WordNet synset {*bridge*, *span*} share the meaning of "a structure that allows people or vehicles to cross an obstacle such as a river or canal or railway etc.". However, *bridge* and *span* cannot be substutited for each other in the sentence "suspension bridges are typically ranked by the length of their main span", and doing so would likely raise the suspicion of an observer due to the resulting anomaly in the text.

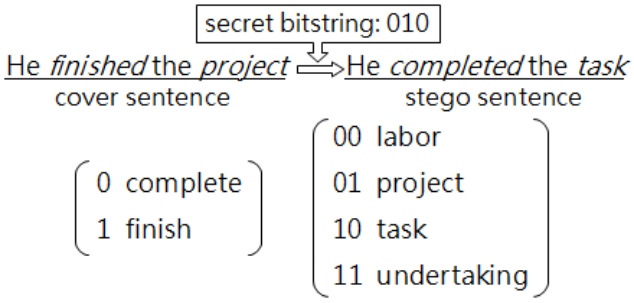


Figure 1: An example of the basic algorithm

Our solution to this problem is to perform a contextual check which utilises the Web 1T n-gram Google n-gram corpus.[1] We evaluate the method using the data from the English Lexical Substitution task for SemEval-2007.[2] The resulting precision of our lexical substitution system can be seen as an indirect measure of the imperceptibility of the stegosystem, whereas the recall can be seen as an indirect measure of the payload.

The paper is organised so that the contextual check is described first, and this is evaluated independently of the steganographic application. Then the vertex colouring method is presented, and finally we show how the contextual check can be integrated with the vertex colouring coding method to give a complete stegsosystem. For readers unfamiliar with linguistic steganogaphy, Section 2 has some examples of how bits can be hidden using textual transformations. Also, Chang and Clark (2010) is a recent NLP paper which describes the general linguistic steganography framework.

## 2 Related Work

In the original work on linguistic steganography in the late 1990s, Winstein proposed an information hiding algorithm using a block coding method to encode synonyms, so that the selection of a word from a synset directly associates with part of the secret bitstring (Bergmair, 2007). Figure 1 illustrates the embedding procedure of this approach. In this example, the bitstring to be embedded is 010, which

[1] www.ldc.upenn.edu/Catalog/docs/LDC2006T13/readme.txt
[2] http://www.dianamccarthy.co.uk/task10index.html

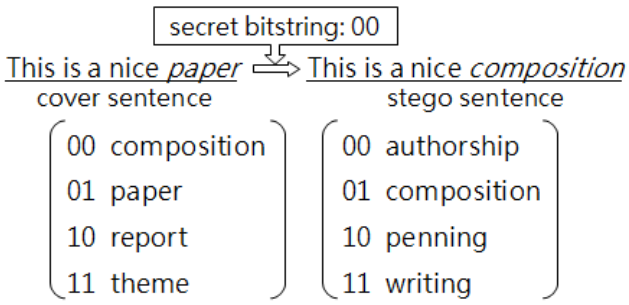


Figure 2: An example of applying the basic algorithm to overlapping synsets

can be divided into two codewords, 0 and 10, and the information carriers in the cover text are the words *finished* and *project*. According to the encoding dictionary, *complete* represents 0, and *task* represents 10; hence these words are chosen and replace the original words in the cover text (with suitable suffixation). The stego sentence "He completed the task" is then sent to the receiver. In order to recover the message, the receiver only needs a copy of the encoding dictionary, and the decoding algorithm simply reverses the process used to encode the hidden bits. Note that the receiver does not need the original cover text to recover the information.

This algorithm requires synonym sets to be disjoint; i.e. no word may appear in more than one synonym set, since overlapping synsets may cause ambiguities during the decoding stage. Figure 2 shows what happens when the basic algorithm is applied to two overlapping synonym sets. As can be seen from the example, *composition* is represented by two different codewords and thus the secret bitstring cannot be reliably recovered, since the receiver does not know the original cover word or the sense of the word. In order to solve this problem, we propose a novel coding method based on vertex colouring, described in Section 4.

In addition to the basic algorithm, Winstein proposed the T-Lex system using synonym substitution as the text transformation. In order to solve the problem of words appearing in more than one synonym set, Winstein defines *interchangeable words* as words that belong to the same synsets, and only uses these words for substitution. Any words that are not interchangeable are discarded and not available for carrying information. The advantage in this approach is that interchangeable words always receive the same codeword. The disadvantage is that many synonyms need to be discarded in order to achieve this property. Winstein calculates that only 30% of WordNet can be used in such a system.

Another stegosystem based on synonym substitution was proposed by Bolshakov (2004). In order to ensure both sender and receiver use the same synsets, Bolshakov applied *transitive closure* to overlapping synsets to avoid the decoding ambiguity. Applying transitive closure leads to a merger of all the overlapping synsets into one set which is then seen as the synset of a target word. Consider the overlapping synsets in Figure 2 as an example. After applying transitive closure, the resulting set is {'authorship', 'composition', 'paper', 'penning', 'report', 'theme', 'writing'}.

Bolshakov (2004) also uses a method to determine whether a substitution is applicable in context, using a collocation-based test. Finally, the collocationally verified synonyms are encoded by using the block coding method. This is similar to our use of the Google n-gram data to check for contextual applicability.

The disadvantage of Bolshakov's system is that all words in a synonym transitive closure chain need to be considered, which can lead to very large sets of synonyms, and many which are not synonymous with the original target word. In contrast, our proposed method operates on the original synonym sets without extending them unnecessarily.

## 3 Proposed Method and Experiments

### 3.1 Resources

We use WordNet (Fellbaum, 1998) to provide sets of synonyms (synsets) for nouns, verbs, adjectives and adverbs. Since the purpose of using WordNet is to find possible substitutes for a target word, those synsets containing only one entry are not useful and are ignored by our stegosystem. In addition, our stegosystem only takes single word substitution into consideration in order to avoid the confusion of finding information-carrying words during the decoding phase. For example, if the cover word 'complete' is replaced by 'all over', the receiver would not know whether the secret message is embedded in the word 'over' or the phrase 'all over'. Table 1 shows the statistics of synsets used in our stegosystem.

| | noun | verb | adj | adv |
|---|---|---|---|---|
| # of synsets | 16,079 | 4,529 | 6,655 | 964 |
| # of entries | 30,933 | 6,495 | 14,151 | 2,025 |
| average set size | 2.56 | 2.79 | 2.72 | 2.51 |
| max set size | 25 | 16 | 21 | 8 |

Table 1: Statistics of synsets used in our stegosystem

For the contextual check we use the Google Web 1T 5-gram Corpus (Brants and Franz, 2006) which contains counts for n-grams from unigrams through to five-grams obtained from over 1 trillion word tokens of English Web text. The corpus has been used for many tasks such as spelling correction (Islam and Inkpen, 2009; Carlson et al., 2008) and multi-word expression classification (Kummerfeld and Curran, 2008). Moreover, for the SemEval-2007 English Lexical Substitution Task, which is similar to our substitution task, six out of ten participating teams utilised the Web 1T corpus.

### 3.2 Synonym Checking Method

In order to measure the degree of acceptability in a substitution, the proposed filter calculates a substitution score for a synonym by using the observed frequency counts in the Web n-gram corpus. The method first extracts contextual n-grams around the synonym and queries the n-gram frequency counts from the corpus. For each $n$, the total count $f_n$ is calculated by summing up individual n-gram frequencies, for every contextual n-gram containing the target word. We define a *count function* $Count(w) = \sum_{n=2}^{5} log(f_n)$ where $log(0)$ is defined as zero. If $Count(w) = 0$, we assume the word $w$ is unrelated to the context and therefore is eliminated from consideration. We then find the maximum $Count(w)$ called $max$ from the remaining words. The main purpose of having $max$ is to score each word relative to the most likely synonym in the group, so even in less frequent contexts which lead to smaller frequency counts, the score of each synonym can still indicate the degree of feasibility. The substitution score is defined as $Score(w) = Count(w) \div max$. The hypothesis is that a word with a high score is more suitable for the context, and we apply a threshold so that synonyms having a score lower than the threshold are discarded.

Figure 3 demonstrates an example of calculating the substitution score for the synonym 'pole' given the cover sentence "This is not a very high *bar*." First of all, various contextual n-grams are extracted from the sentence and the Web n-gram corpus is consulted to obtain their frequency counts. $Count(\text{'}pole\text{'})$ is then calculated using the n-gram frequencies. Suppose the threshold is 0.37, and the max score is 52. Since $Count(\text{'}pole\text{'})$ is greater than zero and the substitution score $Score(\text{'}pole\text{'})$ is 0.44, the word 'pole' is determined as acceptable for this context (even though it may not be, depending on the meaning of 'bar' in this case).

| | | |
|---|---|---|
| $f_2$=525,856 | high *pole* | 3,544 |
| | *pole* . | 522,312 |
| $f_3$=554 | very high *pole* | 84 |
| | high *pole* . | 470 |
| $f_4$=72 | a very high *pole* | 72 |
| | very high *pole* . | 0 |
| $f_5$=0 | not a very high *pole* | 0 |
| | a very high *pole* . | 0 |
| Count('*pole*')=log($f_2$)+log($f_3$)+log($f_4$)+log($f_5$)=23 | | |
| Score('*pole*')=Count('*pole*')/$max$=0.44>0.37 | | |

Figure 3: An example of using the proposed synonym checking method

### 3.3 Evaluation Data

In order to evaluate the proposed synonym checking method, we need some data to test whether our method can pick out acceptable substitutions. The English Lexical Substitution task for SemEval-2007 has created human-annotated data for developing systems that can automatically find feasible substitutes given a target word in context. This data comprises 2010 sentences selected from the English Internet Corpus[3], and consists of 201 words: nouns, verbs, adjectives and adverbs each with ten sentences containing that word. The five annotators were asked to provide up to three substitutes for a target word in the context of a sentence, and were permitted to consult a dictionary or thesaurus of their choosing.

We use the sentences in this gold standard as the cover text in our experiments so that the substitutes provided by the annotators can be the positive data for evaluating the proposed synonym check-

[3]http://corpus.leeds.ac.uk/internet.html

| | noun | verb | adj | adv |
|---|---|---|---|---|
| # of target words | 59 | 54 | 57 | 35 |
| # of sentences | 570 | 527 | 558 | 349 |
| # of positives | 2,343 | 2,371 | 2,708 | 1,269 |
| # of negatives | 1,914 | 1,715 | 1,868 | 884 |

Table 2: Statistics of experimental data

ing methods. Since we only take into consideration the single word substitutions for the reason described earlier, multi-word substitutes are removed from the positive data. Moreover, we use WordNet as the source of providing candidate substitutes in our stegosystem, so if a human-provided substitute does not appear in any synsets of its target word in WordNet, there is no chance for our system to replace the target word with the substitute and therefore, the substitute can be eliminated. Table 2 presents the statistics of the positive data for our experiments.

Apart from positive data, we also need some negative data to test whether our method has the ability to filter out bad substitutions. Since the annotators were allowed to refer to a dictionary or thesaurus, we assume that annotators used WordNet as one of the reference resources while generating candidates. Hence we assume that, if a word in the correct synset for a target word is not in the set produced by the human annotators, then it is inappropriate for that context and a suitable negative example. This method is appropriate because our steganography system has to distinguish between good and bad synonyms from WordNet, given a particular context.

For the above reasons, we extract the negative data for our experiments by first matching positive substitutes of a target word to all the synsets that contain the target word in WordNet. The synset that includes the most positive substitutes is used to represent the meaning of the target word. If there is more than one synset containing the highest number of positives, all the synsets are taken into consideration. We then randomly select up to six single-word synonyms other than positive substitutes from the chosen synset(s) as negative instances of the target word. Figure 4 shows an example of automatically collected negative data from WordNet given a target word and its positive substitutes. The synset {'remainder', 'balance', 'residual', 'residue',

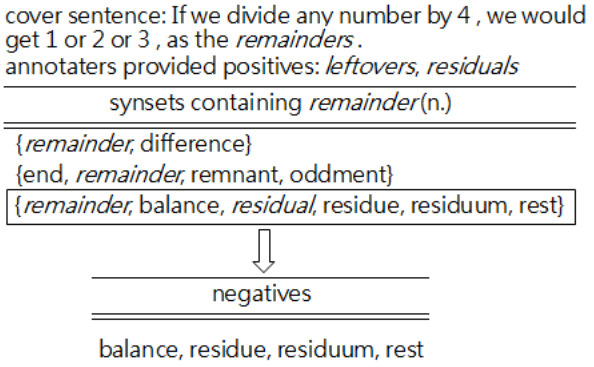


Figure 4: An example of automatic negative data

| | noun | verb | adj | adv |
|---|---|---|---|---|
| # of true negatives | 234 | 201 | 228 | 98 |
| # of false negatives | 9 | 20 | 28 | 16 |

Table 3: Annotation results for negative data

'residuum', 'rest'} is selected for negative data collection since it contains one of the positives while the other synsets do not. We assume the selected synset represents the meaning of the original word, and those synonyms in the synset which are not annotated as positives must have a certain degree of mismatch to the context. Therefore, from this example, 'balance', 'residue', 'residuum' and 'rest' are extracted as negatives to test whether our synonym checking method can pick out bad substitutions from a set of words sharing similar or the same meaning.

In order to examine whether the automatically collected instances are true negatives and hence form a useful test set, a sample of automatically generated negatives was selected for human evaluation. For each PoS one sentence of each different target word is selected, which results in roughly 13% of the collected negative data, and every negative substitute of the selected sentences was judged by the second author. As can be seen from the annotation results shown in Table 3, most of the instances are true negatives, and only a few cases are incorrectly chosen as false negatives. Since the main purpose of the data set is to test whether the proposed synonym checking method can guard against inappropriate synonym substitutions and be integrated in the stegosystem, it is reasonable to have a few false negatives in our experimental data. Also, it is more harmless to rule out a permissible substitu-

| PoS | Acc% | P% | R% | F% | Threshold |
|---|---|---|---|---|---|
| noun | 70.2 | 70.0 | 80.2 | 74.7 | 0.58 |
| verb | 68.1 | 69.7 | 79.5 | 74.3 | 0.56 |
| adj | 72.5 | 72.7 | 85.7 | 78.7 | 0.48 |
| adv | 73.7 | 76.4 | 80.1 | 78.2 | 0.54 |

Table 4: Performance of the synonym checking method

tion than including an inappropriate replacement for a stegosystem in terms of the security. Table 2 gives the statistics of the automatically collected negative data for our experiments.

Note that, although we use the data from the lexical substitution task, our task is different: the possible substitutions for a target word need to be fixed in advance for linguistic steganography (in order for the receiver to be able to recover the hidden bits), whereas for the lexical substitution task participants were asked to discover possible replacements.

### 3.4 Results

The performance of the proposed checking method is evaluated in terms of accuracy, precision, recall and balanced F-measure. Accuracy represents the percentage of correct judgements over all acceptable and unacceptable substitutions. Precision is the percentage of substitutions judged acceptable by the method which are determined to be suitable synonyms by the human judges. Recall is the percentage of substitutions determined to be feasible by the human annotators which are also judged acceptable by the method. The interpretation of the measures for a stegosystem is that a higher precision value implies a better security level since good substitutions are less likely to be seen as suspicious by the observer; whereas a larger recall value means a greater payload capacity since words are being substituted where possible and therefore embedding as much information as possible.

In order to derive sensible threshold values for each PoS, 5-fold cross-validation was implemented to conduct the experiments. For each fold, 80% of the data is used to find the threshold value which maximises the accuracy, and that threshold is then applied to the remaining 20% to get the final result. Table 4 gives the results for the synonym checking method and the average threshold values over the 5 folds. In addition, we are interested in the effect of

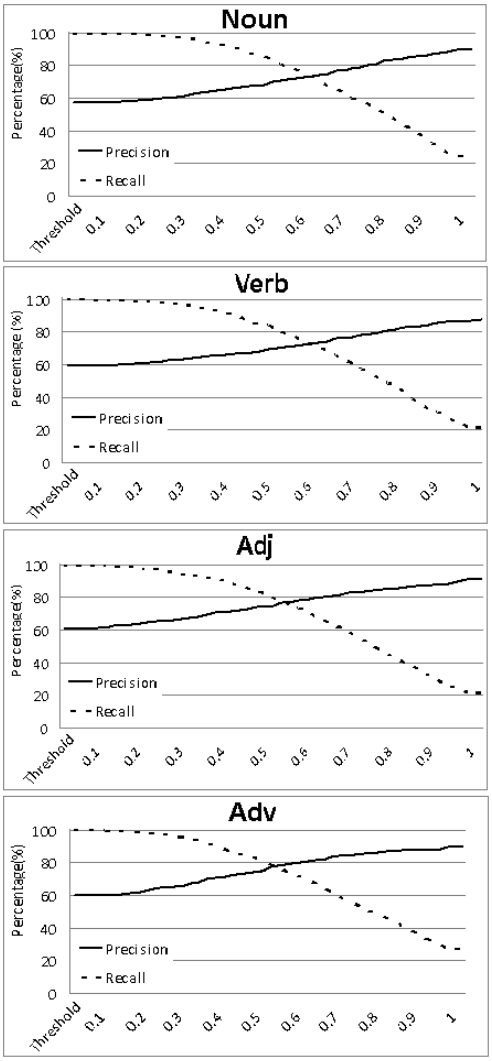


Figure 5: System performance under various thresholds

various thresholds on the system performance. Figure 5 shows the precision and recall values with respect to different thresholds for each PoS. From the graphs we can clearly see the trade-off between precision and recall. Although a higher precision can be achieved by using a higher threshold value, for example noun's substitutions almost reach 90% precision with threshold equal to 0.9, the large drop in recall means many applicable synonyms are being eliminated. In other words, the trade-off between precision and recall implies the trade-off between imperceptibility and payload capacity for linguistic steganography. Therefore, the practical threshold setting would depend on how steganography users want to trade off imperceptibility for payload.

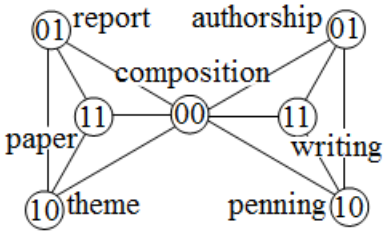


Figure 6: An example of coloured synonym graph

## 4 Proposed Stegosystem

### 4.1 The Vertex Coloring Coding Method

In this section, we propose a novel coding method based on vertex colouring by which each synonym is assigned a unique codeword so the usage of overlapping synsets is not problematic for data embedding and extracting. A vertex colouring is a labelling of the graph's vertices with colours subject to the condition that no two adjacent vertices share the same colour. The smallest number of colours required to colour a graph G is called its chromatic number $\chi(G)$, and a graph $G$ having chromatic number $\chi(G) = k$ is called a $k$-chromatic graph. The main idea of the proposed coding method is to represent overlapping synsets as an undirected $k$-chromatic graph called a synonym graph which has a vertex for each word and an edge for every pair of words that share the same meaning. A synonym is then encoded by a codeword that represents the colour assigned by the vertex colouring of the synonym graph. Figure 6 shows the use of four different colours, represented by '00', '01', '10' and '11', to colour the 4-chromatic synonym graph of the two overlapping synsets in Figure 2. Now, the overlapped word 'composition' receives a unique codeword no matter which synset is considered, which means the replacement of 'paper' to 'composition' in Figure 2 will not cause an ambiguity since the receiver can apply the same coding method to derive identical codewords used by the sender.

99.6% of synsets in WordNet have size less than 8, which means most of the synsets cannot exhaust more than a 2-bit coding space (i.e. we can only encode at most 2 bits using a typical synset). Therefore, we restrict the chromatic number of a synonym graph $G$ to $1 < \chi(G) \leq 4$, which implies the maximum size of a synset is 4. When $\chi(G) = 2$, each

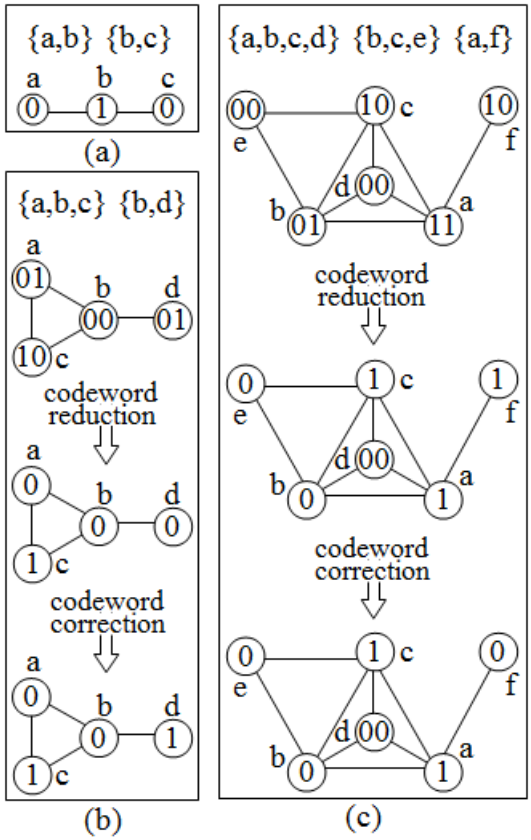


Figure 7: Examples of 2,3,4-chromatic synonym graphs

vertex is assigned a single-bit codeword either '0' or '1' as shown in Figure 7(a). When $\chi(G) = 3$, the overlapping set's size is either 2 or 3, which cannot exhaust the 2-bit coding space although codewords '00', '01' and '10' are initially assigned to each vertex. Therefore, only the most significant bits are used to represent the synonyms, which we call *codeword reduction*. After the codeword reduction, if a vertex has the same codeword, say '0', as all of its neighbors, the vertex's codeword must be changed to '1' so that the vertex would be able to accommodate either secret bit '0' or '1', which we call *codeword correction*. Figure 7(b) shows an example of the process of codeword reduction and codeword correction for $\chi(G) = 3$. For the case of $\chi(G) = 4$, codeword reduction is applied to those vertices that themselves or their neighboring vertices have no access to all the codewords '00', '01', '10' and '11'. For example, vertices $a$, $b$, $c$, $e$ and $f$ in Figure 7(c) meet the requirement of needing codeword reduction. The codeword correction process is then further applied to vertex $f$ to rectify its accessibility.

Figure 8 describes a greedy algorithm for constructing a coded synonym graph using at most 4 colours, given $n$ synonyms $w_1$, $w_2$,..., $w_n$ in the overlapping synsets. Let us define a function $E(w_i, w_j)$ which returns an edge between $w_i$ and $w_j$ if $w_i$ and $w_j$ are in the same synset; otherwise returns false. Another function $C(w_i)$ returns the colour of the synonym $w_i$. The procedure loops through all the input synonyms. For each iteration, the procedure first finds available colours for the target synonym $w_i$. If there is no colour available, namely all the four colours have already been given to $w_i$'s neighbors, $w_i$ is randomly assigned one of the four colours; otherwise, $w_i$ is assigned one of the available colours. After adding $w_i$ to the graph $G$, the procedure checks whether adding an edge of $w_i$ to graph $G$ would violate the vertex colouring. After constructing the coloured graph, codeword reduction and codeword correction as previously described are applied to revise improper codewords.

**INPUT:** a synonym list $w_1$, $w_2$,..., $w_n$ and an empty graph $G$
**OUTPUT:** a coded synonym graph $G$ using at most four colours

**FOR** every synonym $w_i$ in the input list
    initialize four colours as available for $w_i$
    **FOR** every $w_j$ in graph $G$
        **IF** $E(w_i, w_j)$ **THEN**
            set $C(w_j)$ as unavailable
        **END IF**
    **END FOR**
    **IF** there is a colour available **THEN**
        assign one of the available colours to $w_i$
    **ELSE**
        assign one of the four colours to $w_i$
    **END IF**
    **ADD** $w_i$ to graph $G$
    **FOR** every $w_j$ in graph $G$
        **IF** $E(w_i, w_j)$ and $C(w_i)$ is not equal to $C(w_j)$ **THEN**
            **ADD** edge $E(w_i, w_j)$ to $G$
        **END IF**
    **END FOR**
**END FOR**
*codeword reduction*
*codeword correction*
**OUTPUT** graph $G$

Figure 8: Constructing a coloured synonym graph

### 4.2 Proposed Lexical Stegosystem

Figure 9 illustrates the framework of our lexical stegosystem. Note that we have preprocessed WordNet by excluding multi-word synonyms and single-entry synsets. A possible information carrier is first found in the cover sentence. We define a possible information carrier as a word in the cover sentence that belongs to at least one synset in WordNet. The synsets containing the target word, and all other synsets which can be reached via the synonym relation, are then extracted from WordNet (i.e. we build the connected component of WordNet which contains the target word according to the synonym relation). Words in these sets are then examined by the Google n-gram contextual checking method to eliminate inappropriate substitutions. If there is more than one word left and if words which pass the filter all belong to the same synset, the block coding method is used to encode the words; otherwise the vertex colouring coding is applied. Finally, according to the secret bitstring, the system selects the synonym that shares an edge with the target word and has as its codeword the longest potential match with the secret bitstring.

We use the connected component of WordNet containing the target word as a simple method to ensure that both sender and receiver colour-code the same graph. It is important to note, however, that the sender only considers the synonyms of the target word as potential substitutes; the connected component is only used to consistently assign the codes.

For the decoding process, the receiver does not need the original text for extracting secret data. An information carrier can be found in the stegotext by referring to WordNet in which related synonyms are extracted. Those words in the related sets undergo the synonym checking method and then are encoded by either block coding or vertex colouring coding scheme depending on whether the remaining words are in the same synset. Finally, the secret bitstring is implicit in the codeword of the information carrier and therefore can be extracted.

We demonstrate how to embed secret bit 1 in the

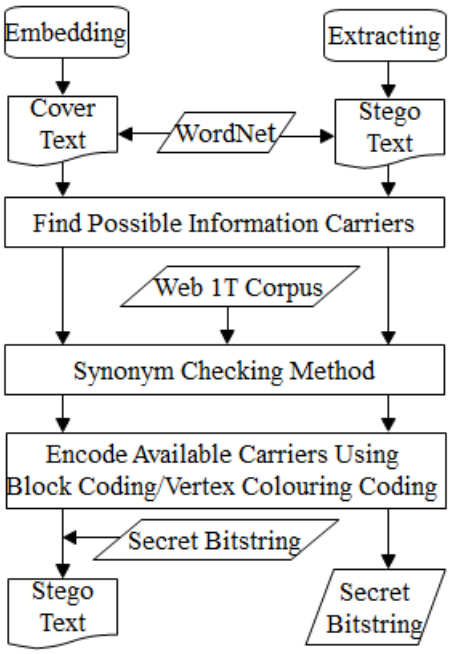


Figure 9: Framework of the proposed lexical stegosystem

sentence "it is a *shame* that we could not reach the next stage." A possible information carrier 'shame' is first found in the sentence. Table 5 lists the related synsets extracted from WordNet. The score of each word calculated by the synonym checking method using the Web 1T Corpus is given as a subscript. Assume the threshold score is 0.27. The output of the synonym checking method is shown at the right side of Table 5. Since the remaining words do not belong to the same synset, the vertex colouring coding method is then used to encode the words. Figure 10(a) is the original synset graph in which each vertex is assigned one of the four colours; Figure 10(b) is the graph after applying codeword reduction. Although both 'disgrace' and 'pity' are encoded by '1', 'pity' is chosen to replace the cover word since it has a higher score. Finally, the stegotext is generated, "it is a *pity* that we could not reach the next stage."

As a rough guide to the potential payload with this approach, we estimate that, with a threshold of 0.5 for the contextual check, the payload would be slightly higher than 1 bit per newspaper sentence.

## 5 Conclusions

One of the contributions of this paper is to develop a novel lexical stegosystem based on vertex colouring coding which improves the data embedding capacity compared to existing systems. The vertex colouring coding method represents synonym substitution as a synonym graph so the relations between words can be clearly observed. In addition, an automatic system for checking synonym acceptability in context is integrated in our stegosystem to ensure information security. For future work, we would like to explore more linguistic transformations that can meet the requirements of linguistic steganography — retaining the meaning, grammaticality and style of the original text. In addition, it is crucial to have a full evaluation of the linguistic stegosystem in terms of imperceptibility and payload capacity so we can know how much data can be embedded before the cover text reaches its maximum distortion which is tolerated by a human judge.

| cover sentence: It is a *shame* that we could not reach the next stage | |
|---|---|
| original synsets | retained synsets |
| $\{commiseration_{.28}, pity_{.97}, ruth_{.13}, pathos_{.31}\}$ | $\{commiseration, pity, pathos\}$ |
| $\{pity_{.97}, shame_1\}$ | $\{pity, shame\}$ |
| $\{compassion_{.49}, pity_{.97}\}$ | $\{compassion, pity\}$ |
| $\{condolence_{.27}, commiseration_{.28}\}$ | $\{commiseration\}$ |
| $\{pathos_{.31}, poignancy_{.31}\}$ | $\{pathos, poignancy\}$ |
| $\{shame_1, disgrace_{.84}, ignominy_{.24}\}$ | $\{shame, disgrace\}$ |
| $\{compassion_{.49}, compassionateness_0\}$ | $\{compassion\}$ |
| $\{poignance_{.12}, poignancy_{.31}\}$ | $\{poignancy\}$ |

Table 5: Synsets of 'shame' before and after applying the synonym checking method

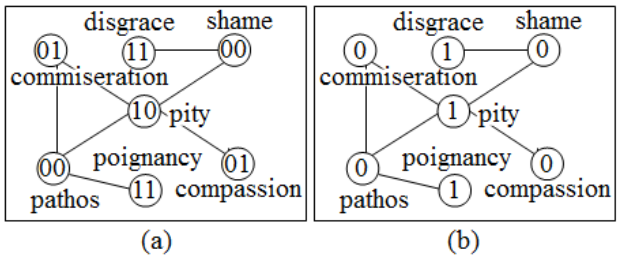


Figure 10: Synonym graph of 'shame'

## References


Mikhail J. Atallah, Craig J. McDonough, Victor Raskin, and Sergei Nirenburg. 2001a. Natural language processing for information assurance and security: an overview and implementations. In *Proceedings of the 2000 workshop on New security paradigms*, pages 51–65, Ballycotton, County Cork, Ireland.

Mikhail J. Atallah, Victor Raskin, Michael C. Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001b. Natural language watermarking: design, analysis, and a proof-of-concept implementation. In *Proceedings of the 4th International Information Hiding Workshop*, volume 2137, pages 185–199, Pittsburgh, Pennsylvania.

Mikhail J. Atallah, Victor Raskin, Christian F. Hempelmann, Mercan Karahan, Umut Topkara, Katrina E. Triezenberg, and Radu Sion. 2002. Natural language watermarking and tamperproofing. In *Proceedings of the 5th International Information Hiding Workshop*, pages 196–212, Noordwijkerhout, The Netherlands.

Richard Bergmair. 2007. A comprehensive bibliography of linguistic steganography. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505.

Igor A. Bolshakov. 2004. A method of linguistic steganography based on collocationally-verified synonym. In *Information Hiding: 6th International Workshop*, volume 3200, pages 180–191, Toronto, Canada.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. Technical report, Google Research.

Andrew Carlson, Tom M. Mitchell, and Ian Fette. 2008. Data analysis project: Leveraging massive textual corpora using n-gram statistics. Technical report, School of Computer Science, Carnegie Mellon University.

Ching-Yun Chang and Stephen Clark. 2010. Linguistic steganography using automatically generated paraphrases. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 591–599, Los Angeles, California, June. Association for Computational Linguistics.

Mark Chapman and George I. Davida. 1997. Hiding the hidden: A software system for concealing ciphertext as innocuous text. In *Proceedings of the First International Conference on Information and Communication Security*, volume 1334, pages 335–345, Beijing.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press, first edition.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using Google Web IT 3-grams. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249, Morristown, USA. Association for Computational Linguistics.

Jonathan K Kummerfeld and James R Curran. 2008. Classification of verb particle constructions with the Google Web 1T Corpus. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 55–63, Hobart, Australia, December.

Yuling Liu, Xingming Sun, and Yong Wu. 2005. A natural language watermarking based on Chinese syntax. In *Advances in Natural Computation*, volume 3612, pages 958–961, Changsha, China.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53, Prague, Czech Republic.

Hasan M. Meral, Emre Sevinc, Ersin Unkar, Bulent Sankur, A. Sumru Ozsoy, and Tunga Gungor. 2007. Syntactic tools for text watermarking. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, San Jose, CA.

Brian Murphy and Carl Vogel. 2007. The syntax of concealment: reliable methods for plain text information hiding. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, San Jose, CA.

Brian Murphy. 2001. Syntactic information hiding in plain text. Master's thesis, Trinity College Dublin.

Cuneyt M. Taskiran, Mercan Topkara, and Edward J. Delp. 2006. Attacks on linguistic steganography systems using text analysis. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6072, pages 97–105, San Jose, CA.

Mercan Topkara, Umut Topkara, and Mikhail J. Atallah. 2006a. Words are not enough: sentence level natural language watermarking. In *Proceedings of the ACM Workshop on Content Protection and Security*, pages 37–46, Santa Barbara, CA.

Umut Topkara, Mercan Topkara, and Mikhail J. Atallah. 2006b. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 164–174, Geneva, Switzerland.

M. Olga Vybornova and Benoit Macq. 2007. A method of text watermarking using presuppositions. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, San Jose, CA.