Effective Linguistic Steganography Detection

Chen Zhi-li, Huang Liu-sheng, Yu Zhen-shan, Zhao Xin-xin, Zheng Xue-ling
Department of Computer Science and Technology, University of Science and Technology of China, National High Performance Computing Center(Hefei), Hefei, Anhui 230027, China
zlchen3@mail.ustc.edu.cn

Abstract:

Linguistic steganography is an art of concealing secret messages. More specifically, it takes advantage of the properties of natural language, such as the linguistic structure to hide messages. In this paper, an effective method for linguistic steganography detection is presented. In virtue of the concepts in area of information theory, the method uses an information-entropy-like statistical variable of words in detected text segment together with its variance as two classification features. The Support Vector Machine is used as classifier. The method was centered on detection for small size text segments estimated in the hundreds in words. Its achievement is simple and its execution is fast and relatively accurate. In our experiment of detecting the three different linguistic steganography methods: NICETEXT, TEXTO and Markov-Chain-Based, the accuracy exceeds 90%. As a result, our method can be used as a common pre-detection method followed by a more specific and accurate detection method.

Key words: linguistic steganography, detection, information, entropy

1. Introduction

As an ancient art, steganography has revivified with the advent of computer. Nowadays, we have vast accessible bodies of data for steganography usage and increasingly sophisticated techniques for its implementation. While much of the recent research in steganography, especially linguistic steganography has been centered on how to hide secret messages in cover media, it is significant to exploit effective approaches to steganalysis which usually attempts to detect statistical anomalies in cover data which predict the presence of hidden information. This paper focuses on investigation of detection approaches to linguistic steganalysis and attempts to detect linguistic steganography information-theoretically.

Methods for linguistic steganography are usually of linguistically-driven generation or modification [1]. For example, method NICETEXT and TEXTO are of linguistically-driven generation while synonym-substitution method is of linguistically-driven modification. Commonly speaking, linguistically-driven generation method has a high hiding capacity, a cover text looking like natural text but making little sense which may pass a computer sensor but hardly a careless reader. It mainly used to hide a great bulk of encrypted secret information during transmission. Linguistically-driven modification method has a low hiding capacity, a cover text looking like natural text both syntactically and semantically. It is mostly used to hide more secret information being transferred and used in text watermark.

This paper provides an approach to linguistic steganography of linguistically-driven generation. Since the kind of linguistic steganography has faults in statistical properties of the cover text, we design an algorithm for detection that using information entropy-like statistical variable. The algorithm turns out to be so simple that no extra data such as corpus, dictionaries and so on is needed except for the training text segments that used for training classifier and the detected text segment.

The remainder of the paper is structured as follows. Firstly, Section 2 reviews related work. In Section 3, our detection method is described. In Section 4, the experimental results are presented and a discussion is made. Finally, Section 5 is the conclusion.

2. Related work

2.1. Linguistic Steganography

The previous work in linguistic steganography was mainly focused on how to hide information in steganography. The simplest method of modifying text for embedding a message is to substitute selected words by their synonyms so that the meaning of the modified sentences is preserved as much as possible. One steganography approach that is based on synonym substitution is the system proposed by Winstein [2].

This work was supported by the NSF of China (Grant No. 60773032 and 60703071 respectively), the Ph.D. Program Foundation of Ministry of Education of China (No. 20060358014), the Natural Science Foundation of Jiangsu Province of China (No. BK2007060), and the Anhui Provincial Natural Science Foundation (No. 070411043).

There are some other approaches. Among them NICETEXT and TEXTO are most famous.

NICETEXT [3][4] system generates natural-like cover text using the mixture of word substitution and Probabilistic Context-free Grammars (PCFG). There are a dictionary table and a style template in the system. The style template can be generated using PCFG or a sample text. The dictionary is used to randomly generate sequences of words, while the style template selects natural sequences of parts-of-speech when controlling generation of word, capitalization, punctuation, and white space. NICETEXT system was intended to protect the privacy of cryptograms to avoid detection by censors.

TEXTO [5] is a text steganography program designed for transforming uuencoded or pgp ascii-armoured ascii data into English sentences. It was written to facilitate the exchange of binary data, especially encrypted data. TEXTO works just like a simple substitution cipher, with each of the 64 ascii symbols used by pgp ascii armour or uuencode from secret data replaced by an English word. Not all of the words in the resulting text are significant, only those nouns, verbs, adjectives, and adverbs used to fill in the preset sentence structures. Punctuation and "connecting" words (or any other words not in the dictionary) are ignored.

Markov-Chain-Based is another approach proposed by [6]. It builds a state transfer chart of Markov signal source from a sample text. A part of state transfer chart tagged with equal probabilities that are represented with one or more bit(s) is illustrated by Fig. 1. Then the algorithm uses the chart to generate cover text according to secret messages.

There are some drawbacks in the above approaches illustrated. For example, the first approach sometimes replaces words synonyms that do not agree with correct English usage or the genre and the author style of the given text. And the later three approaches are detectable by a human warden. They are probably only used in communication channels where only computers act as attackers.

2.2. Linguistic Steganalysis

A few detecting algorithms has been proposed. The paper [7] brought forward an attack against systems based on synonym substitution, especially the system presented by Winstain. The 3-gram language model was used in the attack. The experimental accuracy of this method on classification of steganographically modified sentences was 84.9% and that for unmodified sentences was 61.4%. Another detecting algorithm was proposed by the paper [8], using the measurement of correlation between sentences enlightened by the design ideas of conception chart. The accuracy of the detecting simulation using this algorithm was 76%. The two methods fell short of accuracy that the practical employment of detecting requires. In addition, the first method required a great lot of computation to calculate a large number of parameters of the 3-gram language model while the second one requires a database of rules consuming a lot of work.

This research examines drawbacks of the last three steganography approaches, aiming to accurately detect the employment of the three approaches in small text segment. We have used information entropy-like statistical variable to distinguish between stego-text segments and normal text segments.

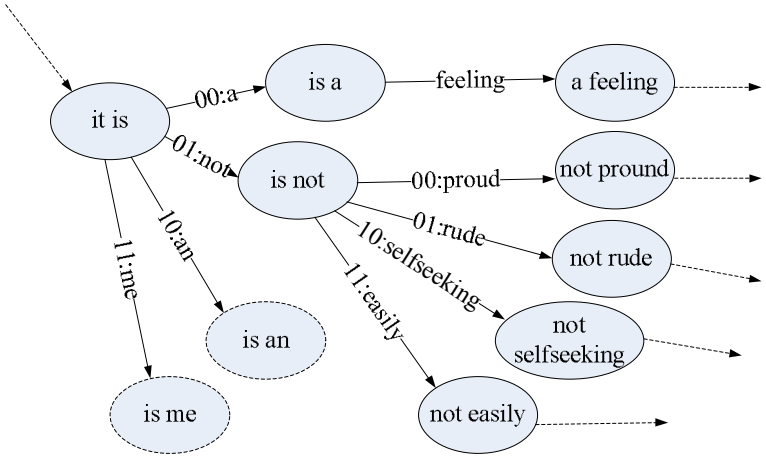


Figure 1. A part of tagged state transfer chart

3. Our Method

In information theory area, the occurrence of an event implies the information value coming with it. The greater the occurrence is, the smaller its information value is. On the other hand, linguistic steganography usually alters the occurrences of words in the cover text. As a result, we have tried to detect the linguistic steganography using information entropy directly, but its results come out poor. The detection accuracy is about sixty or seventy percents, which is not practical. Then we have something unusual changed in the definitions of information and information entropy to get new statistical variables, the results become fairly promising.

3.1. Statistical Variable Definitions

We first define a score measurement to indicate the occurrences of words in the cover text. Let C be the total of occurrences of all words in the cover text, word x be one of the words, word x have n occurrences. Then the sore for word x is computed as follows.

$$S_x = \frac{1}{C}(\sum_{i=1}^{n} i) \qquad (1)$$

The equation is similar to equation that computes the frequency of the word x, except that we magnify the score caused by the *i*th occurrence of word x *i* times. It results in that even a little change in the distribution of word frequencies can cause a great difference of the sore. That is what the detection requires.

We can regard S_x as the probability of the word x, without consideration of what value range it lies. We define an information-like statistical variable of word x which we call it Detection Information (DI) as follows.

$$DI_x = \log\frac{1}{S_x} \qquad (2)$$

With the above definitions of S_x and DI_x, we can define the features for classification. Let the cover text contain N distinct words. The information-entropy-like statistical variable that we call it Detection Entropy (DE) of the cover text is defined:

$$DE = \sum_{i=0}^{N-1} S_i DI_i = -\sum_{i=0}^{N-1} S_i \log S_i \quad (3)$$

The DE variance-like statistical variable is

$$Var(DE) = \sum_{i=0}^{N-1} S_i (DI_i - DE)^2 \quad (4)$$

The statistical variables DE and Var(DE) then can be used as two classification features.

3.2 Suppose Vector Machine

SVM is a popular technique for classification [9]. A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training data set contains one class label and several features and that in the testing data set contains only the features. The goal of SVM is to produce a model which predicts the class label of data instances in the testing data set.

Given a training data set of instance-label pairs $(x_i, y_i), i = 1, 2, ..., l$, where $x_i \in R^n$ and $y_i \in \{-1, 1\}$, the support vector machines (SVM) require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i$$

$$\text{subject to } y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Where φ is the predefined function of x, w and b are the adjustable parameters of the decision function.

In this paper, we use an existent SVM classifier [8] to classify our experiment data without getting to the bottom of how SVM classifier works.

3.3 Detection Method Description

In our method, we apply the following procedure on each detected text segment to compute the two classification features described in section 3.1:

First, the text segment is parsed to words ignoring all the punctuations and white blanks encountered. Then all the distinct words, which may be different words originally or different forms of the same words, are found and the total of occurrences of all words, the number of occurrences of each distinct word are computed. At last, the score of each distinct word is computed by equation (1).

Second, we regard the score of each distinct word as its probability, in spite of whose value is not in the range of [0, 1], computing *DI* of each distinct word, which is like information value.

Finally, the *DE* and *Var(DE)* are computed by equation (3) and (4), using the *S*s and *DI*s computed in the first and second step.

After the above procedure is applied to the processed text segment, we get the classification features DE and Var(DE), the only values we will use later for SVM classification. The SVM classification includes two processes: training and testing. Both training and testing text segments experience the above procedure, and their classification features are extracted for classification. As a result, we can describe the detection procedure as Figure 2.

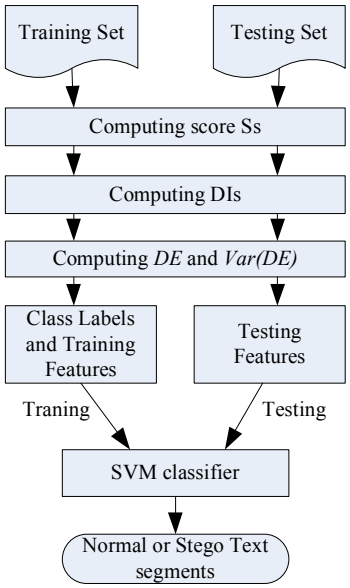


Figure 2. The detection procedure

4. Experimental Results and Discussion

4.1. Experimental Data Structure

In our experiment, experimental data set contains training data set and testing data set, both of which consists of good data set and bad data set. The training data set is used to train the detection algorithm to well distinguish between the "good" data and "bad" data. The testing data set is the data we want to detect. A corpus was built out of the novels written by Charles Dickens, a great English novelist in the Victorian period and named Charles-Dickens-Corpus. Another corpus was built out of novels written by a few novelists whose last name began with letter "s" and named S-Corpus. Finally, a third corpus was built from the cover text generated by the linguistic steganography algorithms we investigated: NICETEXT, TEXTO and Markov-Chain-Based, being called Bad-Corpus. Then the training data set was set up by building the good data set out of Charles-Dickens-Corpus and building the bad data set out of Bad-Corpus. The testing data set was set up in the same way with its good data set built out of S-Corpus. The composing of the experimental data set is shown in Table 1.

Table 1. the composing of experimental data set

Text Data Set	Data Type	Number of Files	Sum
Training Set	Good Set	117	217
	Bad Set	100	
Testing Set	Good Set	146	468
	Bad Set	322	

As shown in Table 1, we can see that the training data set contains the good set with its 117 text segments coming from Charles-Dickens-Corpus and the bad set with its 100 text segments coming from Bad-Corpus. The testing data set contains 146 text segments coming from S-Corpus in good set and 322 text segments coming from Bad-Corpus without overlapping with text segments in training data set in the bad set.

4.2. Results and Discussion

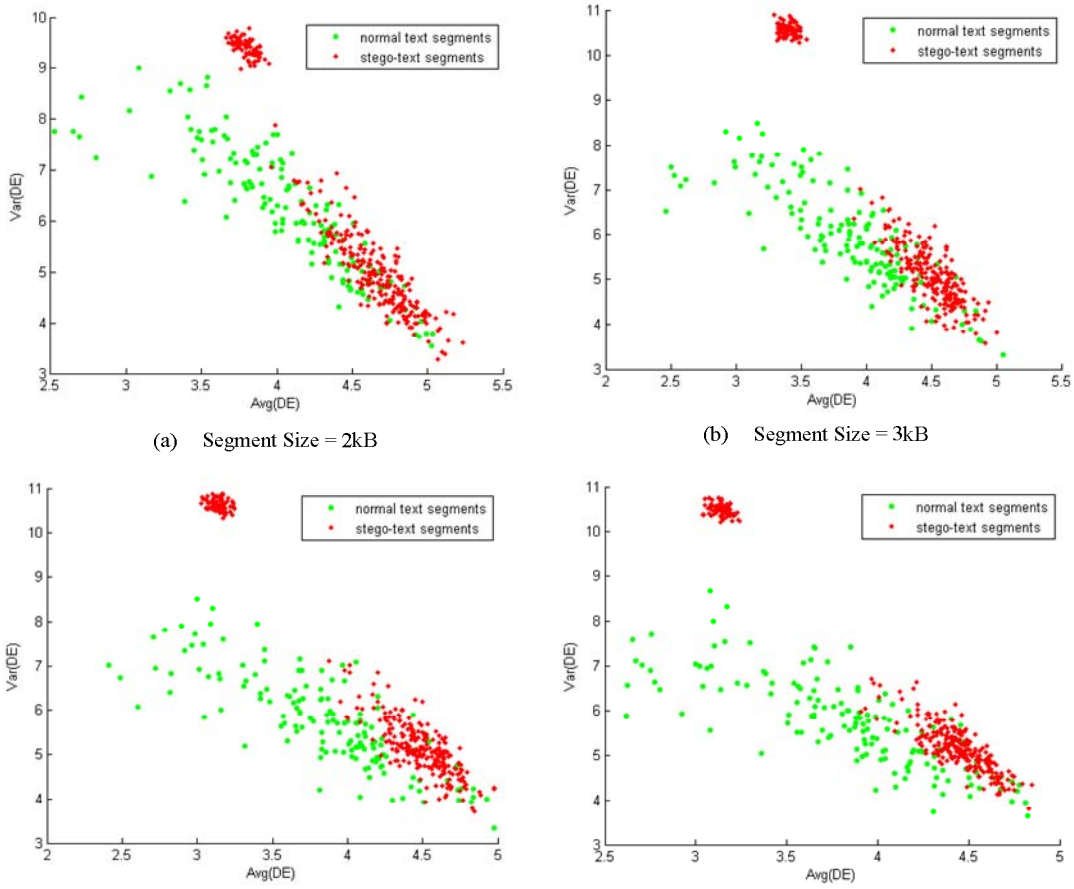
As described in section 4.1, there are hundreds of text files to be tested. In the experiment, we aim to detect text segments with a size smaller than 5kB. Different size of text segments should be detected to get a proper size in which case the detection accuracy is practically high. For each tested text file we read a certain size of segment to detect. The detection results of text segments of size 1kB, 2kB, 3kB, 4kB and 5kB are shown as Table 2.

In the Table 2, we can see that the text segment size is rather small for a statistical algorithm to run. Each text segment only contains hundreds of words. However, the accuracies are relatively high, especially when the segment size is not smaller than 3kB.

Table 2. Detection results with different segment size

Text segment size	Estimation of word count	Success	Fail	accuracy
1kB	150~250	333	135	71.15%
2kB	350~450	371	97	79.27%
3kB	500~600	403	65	86.11%
4kB	650~800	426	42	91.03%
5kB	800~1000	435	33	92.95%

Figure 3 shows the classification features extracted from text segments with a size of 2kB, 3kB, 4kB, 5kB. Most of the normal text segments can be easily distinguished from stego-text segments. Comparing to methods that do not use any linguistic tools, such as part-of-speech tagging tool, word sense disambiguation tool and so on, our method with the accuracies for detecting such small size text segments is promising.



(a) Segment Size = 2kB (b) Segment Size = 3kB

(c) Segment Size = 4kB (d) Segment Size = 5kB

Figure 3. Classification features illustration of text segments of different size

5. Conclusion

In this paper, a statistical algorithm for linguistic steganography detection has been presented. The algorithm makes use of information entropy-like statistical variables of the text segment detected. In the experiment of detecting the three different linguistic steganography methods: NICETEXT, TEXTO and Markov-Chain-Based, the total accuracies on discovering stego-text segments and normal text segments are found to exceed 90% when the segment size is not greater than 5kB.

Many interesting and new challenges are involved in the analysis of linguistic steganography algorithms, which is known as linguistic steganalysis which have little or no counterpart in other media domains, such as images or video. Linguistic steganalysis performance strongly depends on many factors such as the length of the hidden message and the way to generate cover text and so on. Our algorithm is centered on detecting small text segments generated by linguistically-driven generation steganography and our experiment results show that a method of blind detection of small text segments generated by this kind of linguistic steganography is feasible.

6. References

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

[2] Winstein, Keith. Lexical steganography through adaptive modulation of the word choice hash. http://alumni.imsa.edu/~keithw/tlex/lsteg.ps. Ms.

[3] Chapman, Mark. Hiding the Hidden: A Software System for Concealing Ciphertext as Innocuous Text. http://www.NICETEXT.com/NICETEXT/doc/thesis.pdf. 1997.
[4] Chapman, Mark, George Davida and Marc Rennhard. A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography. Lecture Notes in Computer Science, Volume 2200, Springer-Verlag: Berlin Heidelberg. Jan 2001. 156-167.
[5] K. Maher. TEXTO. URL:ftp://ftp.funet.fi/pub/crypt/steganography/texto.tar.gz
[6] WU Shu-feng, HUANG Liu-sheng. Research on Information Hiding. Degree of master, University of Science and Technology of China, 2003.
[7] CM Taskiran, U Topkara, M Topkara et al. Attacks on lexical natural language steganography systems. Proceedings of SPIE, 2006.
[8] ZHOU Ji-jun, YANG Zhu, NIU Xin-xin et al. Research on the detecting algorithm of text document information hiding. Journal on Communications. Dec. 2004 Vol.25, No. 12, 97-101.
[9] CW Hsu, CC Chang, CJ Lin. A Practical Guide to Support Vector Classification. 2003. http://www.csie.ntu.edu.tw/～cjlin.