Text Semantic Steganalysis Based on Word Embedding

Xin Zuo, Huanhuan Hu, Weiming Zhang(✉), and Nenghai Yu

CAS Key Laboratory of Electromagnetic Space Information,
University of Science and Technology of China, Hefei, China
zhangwm@ustc.edu.cn

Abstract. Most state-of-the-art detection methods against synonym substitution based steganography extract features based on statistical distortion. However, synonym substitution will cause not only statistical distortion but also semantic distortion. In this paper, we propose word embedding feature (WEF) to detect the semantic distortion. Furthermore, a fused feature called word embedding and statistical feature set (WESF) which consists of WEF and statistical feature based on word frequency is designed to improve detection performance. Experiments show that WESF can achieve lower detection error rates compared with prmethods.

Keywords: Text steganalysis · Semantic distortion
Word embedding

1 Introduction

Linguistic steganography is the art of hiding messages in digital text without drawing suspicion from steganalysis [1,2]. Linguistic steganography can be broadly divided into two main categories. One is text generation based steganography [3,4]. The generated text can be easily distinguished from natural texts by steganalysis [5]. The other is cover modification based steganography [6–10]. In this category, synonym substitution (SS) based steganography is widely used as it is robust and effective. And the meaning of the text remains almost unchanged during the embedding process, such as T-lex [11] and CoverTweet [12].

In recent years, a few of linguistic steganalysis paradigms have been developed to detect SS steganography. The very first attack of SS steganography was described by Taskiran *et al.* [13]. In this work, 3-g language model was used to distinguish cover and stego text by Support Vector Machine (SVM). With the help of Google, Yu *et al.* [14] constructed a detector to evaluate suitability of synonyms for their context in text. Although it can achieve reliable results when the embedding rates were very high, it had to access Google frequently, which led to a very low running speed. Chen *et al.* [15] proposed the concept of context cluster score (CCS) to evaluate the fitness of the substitution of SS steganography. Xiang *et al.* [16] extracted a statistical feature from synonym appeared in text based on word frequency.

X. Sun et al. (Eds.): ICCCS 2018, LNCS 11066, pp. 485–495, 2018.
https://doi.org/10.1007/978-3-030-00015-8_42

All the methods mentioned above treat words as atomic units. It is difficult to find the relationship between each synonym and its context, just like one-hot representation. Recently, numerous effective word embedding methods have been developed, such as word2vec [17], fasttext [18], and wordRank [19], to describe the semantic relations among words in vector space. For example, *vector("King") – vector("Man") + vector("Woman")* results in a vector that is closest to the vector representation of the word *Queen* [17]. What's more, those methods are all based on statistical distortion such as: the offsets of word frequency or N-gram. However, synonym substitution will cause not only statistical distortion but also semantic distortion: the offsets of synonym in semantic space.

In this paper, we propose a new steganalysis scheme to analyze SS steganography. Since SS steganography would cause statistical and semantic distortion, we extracted statistical feature based on word frequency and semantic feature based on word2vec [17]. Experiments results verify the effectiveness of the proposed steganalysis method for different embedding rates compared with previous methods.

In the next section, we briefly introduce the previous work. The new scheme is explained in Sect. 3. All experimental results are listed and interpreted in Sect. 4. Future directions and a summary appear in Sect. 5.

2 Previous Work

2.1 Xiang et al.'s Features

To describe Xiang et al.'s feature [16], we first define some notations as follow.

Definition 0. A synset is a set of words with the similar meaning, and the dimension of a synset is the number of synonyms it contains. For example, [Cow, Cattle] is a synset that contains two synonyms, and the dimension of this synset is 2.

Definition 1. Attribute pair of a synonym is defined as its position in a synset and the dimension of the synset, denoted by an ordered pair $< pos, dim >$, where $pos \epsilon \{0, 1, ..., dim - 1\}$.

Definition 2. Relative frequency $p(j, k)$ of attribute pair $< j, k >$ in a text is given by

$$p(j, k) = \frac{f(j, k)}{\sum_{i=0}^{k-1} f(i, k)}, \tag{1}$$

where $f(j, k)$ is the number of total occurrences of $< j, k >$ in the text, and $\sum_{i=0}^{k-1} f(i, k)$ represents the total number of attribute pairs that appear in the text.

The synonyms in the synset are sorted in the descending order of their frequencies. When $j < h$, $h \epsilon \{1, .., k - 1\}$, the cover text would contain more synonyms with attribute pair $< j, k >$ than the ones with attribute pair $< h, k >$.

$$f_c(j,k) > f_c(h,k), j < h, \tag{2}$$

$$p_c(j,k) - p_c(h,k) > 0, j < h, \tag{3}$$

where the subscript c represents the cover text.

Due to the randomness of the message, if a synonym w contains a secret message and its attribute pair is $< pos, k >$, pos may be a random value varying from 0 to $k-1$ in a stego text. Therefore, the proportion of synonyms with attribute pair $< j, k >$ is $1/k$. The relationship of cover and stego can be deduced as following equations:

$$f_s(j,k) = (1-r)f_c(j,k) + \frac{1}{k} r \sum_{i=0}^{k-1} f_c(i,k), \tag{4}$$

$$p_s(j,k) - p_s(h,k) = \frac{f_s(j,k) - f_s(h,k)}{\sum_{i=0}^{k-1} f_s(i,k)} = (1-r)(p_c(j,k) - p_c(h,k)), \tag{5}$$

where the subscript s represents the stego text, r is the embedding rate. As $0 < r < 1$, thus

$$p_s(j,k) - p_s(h,k) < p_c(j,k) - p_c(h,k), j < h. \tag{6}$$

The final feature vector proposed in [16] includes six elements such as $p(0,2) - p(1,2)$, $p(0,3) - p(1,3)$, $p(0,3) - p(2,3)$, $p(0,4) - p(1,4)$, $p(0,4) - p(2,4)$, and $p(0,4) - p(3,4)$.

2.2 Chen et al.'s Features

Considering the fitness of synonym and its context, Chen et al. [15] proposes the concept of context cluster score (CCS). For a synonym S_i and its context $C_i = \{c_{i,0}, c_{i,1}, ..., c_{i,2W-1}\}$, the number of element compositions for a S_i is not more than $2^{2W} - 1$. Each element composition is a context cluster, and the CCS of context cluster ς is denoted by V_ς:

$$V_\varsigma = \frac{f_\varsigma K^\alpha}{\sum_{i=0}^{K-1} lg(1 + f_i)}, \tag{7}$$

where K represents the number of the elements in ς, $f_\varsigma, f_0, ..., f_{K-1}$ represent the frequency of ς and the frequencies of the elements in ς respectively, and α is accelerating exponent.

The context fitness of the ith synonym denoted by γ_i is defined as follows:

$$\gamma_i = \frac{1}{2^{2W} - 1} \sum_{\varsigma \epsilon \Phi} V_\varsigma, \tag{8}$$

where Φ is the context cluster set.

On the basis of context fitness, two classification features: Context Maximum Rate (CMR) and Context Maximum Deviation (CMD) of a text are denoted by λ and θ, respectively:

$$\lambda = \frac{1}{n}\sum_{i=0}^{n-1}[\gamma_i = \gamma_{i,max}], \tag{9}$$

$$\theta = \frac{1}{n}\sum_{i=0}^{n-1}(\gamma_i - \gamma_{i,max})^2, \tag{10}$$

where $[\gamma_i = \gamma_{i,max}] = \begin{cases} 1 & \gamma_i = \gamma_{i,max} \\ 0 & \gamma_i \neq \gamma_{i,max} \end{cases}$ and $\gamma_{i,max}$ is the maximum context fitness of the synset.

2.3 Word2Vec Model

Word2vec [17] model was proposed by Mikolov *et al.* to learn distributed representations of words. It has two mirror frame named CBOW(continuous bag-of-words) and Skip-gram. The frameworks are displayed in Fig. 1. There are both three layers in these two frames: input layer, projection layer, output layer. In CBOW model, it uses several history words and future words to estimate current word. In order to achieve this goal, CBOW builds a log-linear classifier with future and history words at the input to classify the current word. The Skip-gram is similar to CBOW, it also builds a log-linear classifier but the input of this classifier is each current word. And the aim of this classifier is to predict words within a certain range before and after the current word.

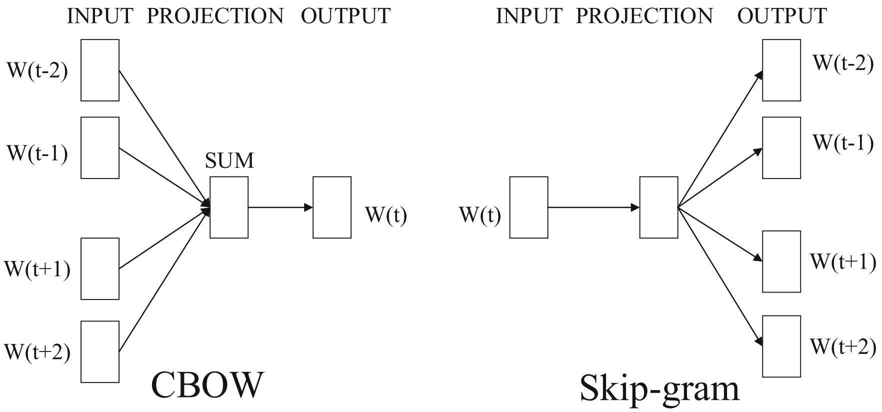

Fig. 1. Word2Vec model framwork.

3 The Proposed Scheme

Although the synonyms in a synset have the similar meaning, the substitution of synonyms would still cause a semantic mismatch in the context. After we are aware of this, the process of steganalysis seems like playing a *banked close game.* For a text, all synonyms are replaced by blanks. For each blank, there is a synset. And we need to choose the one that best fits the context from this synset. We confidently think our choice is the right answer. So if the synonym appears in the text and our choice are not the same, this synonym is mismatched with its context. The number of mismatch can be used to distinguish cover from stego.

In a text, we can extract a sequence of synonyms $S = \{S_1, ..., S_i, ..., S_n\}$ with the help of thesaurus. S_i represents the *ith* synonym that appears in the text, and $\hat{S}_i$ denotes the synset of S_i. For any synonym S_i, we can also extract the N words before and after it as its context, denoted by $C_i = \{c_{i,0}, c_{i,1}, ..., c_{i,2N-2}, c_{i,2N-1}\}$. And the size of N is called context window size. $\overrightarrow{\hat{S}}_i$ and $\overrightarrow{C}_i$ represent the vector space representation of $\hat{S}_i$ and C_i respectively.

Since context words are often not of the same importance, we give different weights to the context words. $W = \{w_0, w_1, ..., w_{2N-2}, w_{2N-1}\}$ is the weights of $\overrightarrow{C}_i$, called context weights. And the value of context weights W will be discussed in Sect. 4. We calculate the vector representation of the weighted context, denoted as WC_i.

$$WC_i = \sum_{j=0}^{2N-1} w_j \overrightarrow{c}_{i,j}. \tag{11}$$

The energy function is defined as the inner product of two vectors, denoted by $E(\overrightarrow{A}, \overrightarrow{B})$.

$$E(\overrightarrow{A}, \overrightarrow{B}) = \overrightarrow{A} \cdot \overrightarrow{B}, \tag{12}$$

where $\overrightarrow{A}$ and $\overrightarrow{B}$ represent two word vectors. For example, the energy of synonym $\overrightarrow{S}_i$ and its weighted context WC_i can be calculated by $E(\overrightarrow{S}_i, WC_i) = \overrightarrow{S}_i \cdot WC_i$.

Using the energy function, we can calculate the conditional probability of $\overrightarrow{S}_i$ with weighted context WC_i, denoted by $P(\overrightarrow{S}_i|WC_i)$.

$$P(\overrightarrow{S}_i|WC_i) = \frac{e^{E(\overrightarrow{S}_i, WC_i)}}{\sum_{v \epsilon \overrightarrow{\hat{S}}_i} e^{E(v, WC_i)}}. \tag{13}$$

For any synonym v in the synset $\overrightarrow{\hat{S}}_i$, we can calculate $P(v|WC_i)$. We think v with maximum conditional probability is the one that best fits the context, so if $v \neq \overrightarrow{S}_i$, the synonym $\overrightarrow{S}_i$ that appears in the text is not fit with the context. In other words, S_i and $\overrightarrow{C}_i$ are mismatched, denoted by $M(\overrightarrow{S}_i, WC_i)$, as follows:

$$M(\overrightarrow{S}_i, WC_i) = \begin{cases} 0 & P(\overrightarrow{S}_i|WC_i) > P(v|WC_i), v \epsilon \overrightarrow{\hat{S}}_i, v \neq \overrightarrow{S}_i \\ 1 & else \end{cases}. \tag{14}$$

We calculate the cosine distance between $\overrightarrow{S}_i$ and WC_i as another measure of mismatch. Denote by $MC(\overrightarrow{S}_i, WC_i)$ as follows:

$$MC(\overrightarrow{S}_i, WC_i) = \begin{cases} 0 & \cos(\overrightarrow{S}_i, WC_i) > \cos(v, WC_i), v\epsilon \overrightarrow{\hat{S}}_i, v \neq \overrightarrow{S}_i \\ 1 & else \end{cases}. \tag{15}$$

The reason we choose the cosine distance is that it can better characterize the similarity between words. Word2vec [17], fasttext [18] and wordRank [19] are all use cosine distance to calculate similarity of words.

We calculate the two mismatches for each of the synonyms appearing in the text, and get two binary sequences with the same length. The sum of the two binary sequences yields a new sequence L. The value of L ranges from 0 to 2. The elements of L are divided into $\lceil n/M \rceil$ groups. And the parameter M is called cluster size. After summarizing each group, we get sequence L'. The features are composed of mean, variance, third order central moments, kurtosis, and skewness of L'.

For example, in Fig. 2, there are two binary sequences: $seq1$ and $seq2$ that calculated by Eqs. (14) and (15), where $n = 15$ and $M = 5$. The WEF is [5 1 0 1.5 0] calculated from L'.

seq1	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**
seq2	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**
L	**0**	**2**	**1**	**1**	**1**	**0**	**0**	**2**	**1**	**2**	**1**	**1**	**0**	**2**	**0**
L'			**6**					**5**					**4**		

Fig. 2. Example of extracting WEF from text.

The details of the extraction procedure of the WEF are described in Algorithm 1.

The proposed text semantic steganalysis process is shown in Fig. 3. The dashed box represents steganalysis with the fused feature WESF.

4 Experiments

4.1 Experimental Setup

The thesaurus is extracted from Wordnet [20], the mean of synset size is 2.26 words. The synonyms in the synset are sorted in the descending order of their frequencies which are derived from a huge corpus.

Algorithm 1. Extraction procedure of the WEF:

```
Get the synonym sequence and its contexts of text.
Calculate two binary sequences $seq1$ and $seq2$ by using Eq. (14) and Eq. (15).
Calculate $L$ by summing $seq1$ and $seq2$.
Divide $L$ into $\lceil n/M \rceil$ groups.
Calculate $L'$ by summing each group.
Calculate mean, variance, third order central moments, kurtosis, and skewness of $L'$.
```

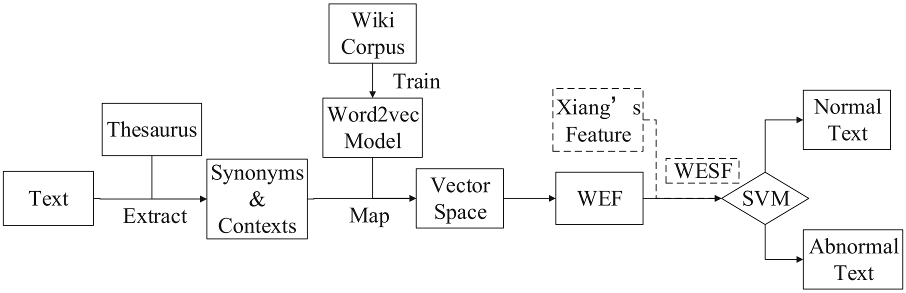

Fig. 3. Proposed text semantic steganalysis process.

All our experiments are carried out on Wiki corpus. The texts are all segmented from this corpus, and the size of text ranges from 5k Bytes to 200k Bytes. The detectors are trained as binary classifiers implemented using SVM [21] with linear kernel.

The Wiki corpus is divided into three parts, training set, validation set and test set, where proportions are 0.7, 0.1 and 0.2. The training set is used to train the word2vec using skip-gram model [17]. The dimension of every word vector is 400-D. We set the window size of word2vec to 5 and abandon the words appeared less than 5 times in training set. We use gensim which is a package for Python to bulit our word2vec model. The validation set is used for model selection. Here, it is used to select cluster size M (see Sect. 3) and determine context weights W. The test set is used to generate the cover text and stego text for evaluating the performance of steganalysis.

We selected two SS steganography techniques: T-lex [11] and its variant Ctsyn [16]. Since CoverTweet needs to be manually selected when multiple candidate results appeared, we do not discuss here. T-lex uses WordNet to select synonyms with correct senses. Only the words appeared in the identical synset in WordNet [20] database are grouped in a synonym set. Messages can be embedded into cover text as follow. First, encode the message letters with Huffman coding. Then represent the encoded binary string in multi-base form. Finally, choose which synonym appears in the text according to the multi-base form. The only difference between T-lex and Ctsyn is the coding strategy. Ctsyn [16] constructs a binary tree for each synset with the synonyms as the leaves while T-lex [11] sets the synsets $sn_0, sn_1, ..., sn_n$ with sizes $k_0, k_1, ..., k_n$.

4.2 Experimental Results

The first part of experiments is to find the effect of context weights W on the detection performance. We consider two cases: one is that all words in context have same weights, the other is that the word which is closer to current synonyms has larger weight. In this case, the context weights are sampled from the Gaussian distribution $N(\mu, \sigma^2)$. We suppose that the context words before and after the synonyms have the same weights, so we set μ to 0. Under this assumption, the weights are only related to the variance of the Gaussian distribution. We observe the average detection error for T-lex when steganalyzing with WEF with different σ^2. In Fig. 4, as σ^2 increases, the average detection error decreases first and then increases, and we get lowest detection error when σ^2 is equal to 4. According to the above experiment, we set $\mu = 0$ and $\sigma^2 = 4$ for Guassian weights. The abscissa of samples are $[-5\ -4\ -3\ -2\ -1\ 1\ 2\ 3\ 4\ 5]$. And we set size of context window N to 5 and the cluster size M to 40. We observe the average detection error for T-lex and Ctsyn when steganalyzing by WEF with different context weights.

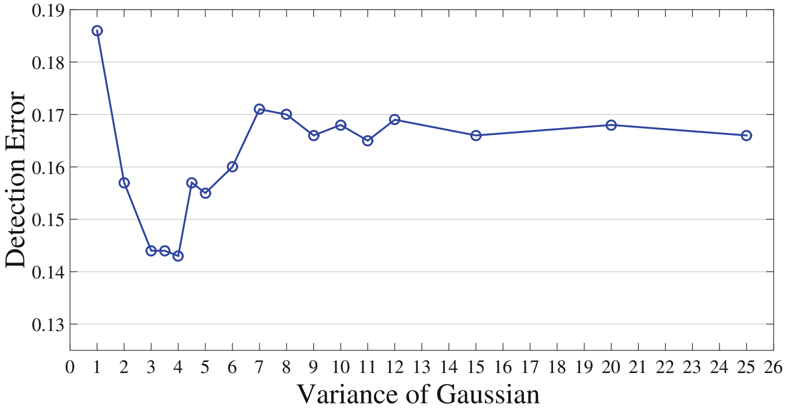

Fig. 4. Average detection error for T-lex [11] with different Gaussian weights.

Our next experiment is aimed at finding the effect of cluster size M on the detection effect. The WEF is extracted with Gaussian weights as described in previous experiment. We observe the average detection error for T-lex when steganalyzing with WEF with different cluster size M. In Fig. 5, as M increases, the average detection error decreases first and then increases in each embedding rate. Also we found out that the average detection error is minimum when M is 40 or 50 in each embedding rate, and 40 is more stable in each embedding rate.

The next part of experiments is to compare the detection performances of Xiang et al.'s features [16], Chen et al.'s features [15] and WEF. As described in the above two experiments, we use Gaussian weights and set cluster size to 40. The context window size $N = 5$. The results are listed in Table 1 and Fig. 6.

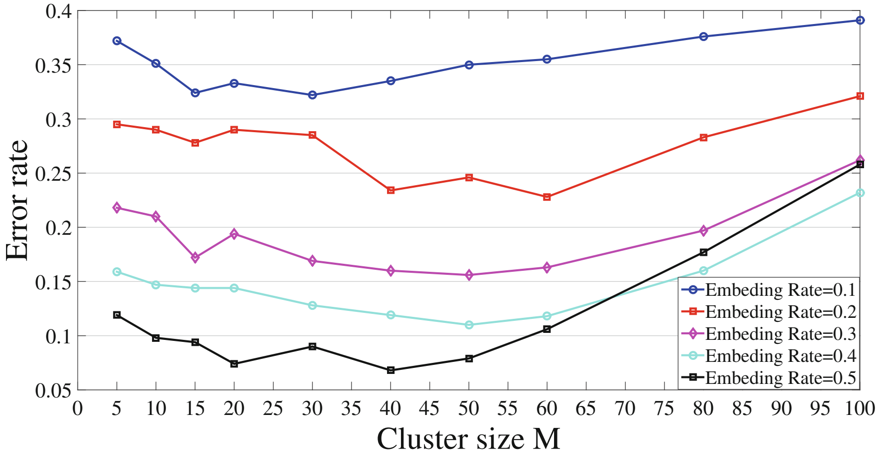

Fig. 5. Average detection error for T-lex [11] when steganalyzing with WEF with different cluster size M.

Firstly, the results indicate that WEF performs better at each embedding rate although WEF and Chen et al.'s features are both considering the fitness of synonyms. This is mostly because the performance of word2vec in semantics is better than statistic natural language processing. Secondly, although the Xiang et al.'s feature set can achieve reliable results when the embedding rate is high, the detection performance of WEF is still better than it. This proves that both the statistical distortion features and the semantic distortion features can distinguish cover and stego text, and it seems that semantic distortion features are more effective. We also notice that Ctsyn appears more secure than T-lex [11].

As synonym substitution will cause not only statistical distortion but also semantic distortion, we propose a fused feature called word embedding and statistical feature set (WESF) which consists of Xiang et al.'s features and WEF.

Table 1. Average detection error $\overline{P}_{\mathrm{E}}$ for two embedding algorithms and four steganalysis feature sets at various kinds of embedding rates.

Stego algorithm	Features	0.1	0.15	0.2	0.25	0.3	0.4	0.5
T-lex	Xiang6	0.395	0.337	0.287	0.248	0.198	0.141	0.09
	Chen	0.396	0.352	0.32	0.277	0.257	0.209	0.171
	WEF	0.334	0.253	0.210	0.186	0.143	0.085	0.064
	WESF	**0.313**	**0.226**	**0.172**	**0.146**	**0.110**	**0.056**	**0.042**
Ctsyn	Xiang6	0.396	0.343	0.293	0.252	0.204	0.144	0.092
	Chen	0.401	0.355	0.321	0.281	0.26	0.211	0.175
	WEF	0.341	0.263	0.218	0.191	0.139	0.116	0.076
	WESF	**0.322**	**0.236**	**0.177**	**0.152**	**0.097**	**0.071**	**0.046**

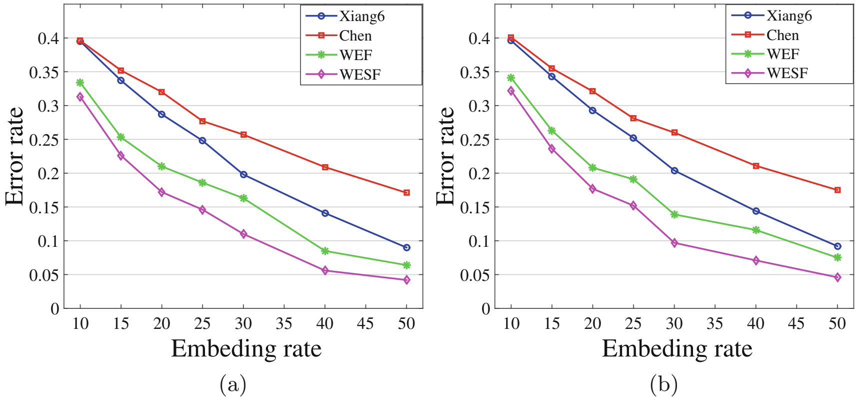

Fig. 6. Detection error for T-lex (a) and Ctsyn (b) with four steganalysis feature sets.

The detection errors of WESF is smaller than both WEF and Xiang et al.'s features, as shown in Fig. 6.

5 Conclusion

In this paper, we propose a novel steganalysis method named WESF to detect synonym substitution based steganography by making use of semantic and statistical distortion. For semantic distortion, we apply word2vec to quantify the distortion magnitude caused by synonym substitution with its context in vector space. We extracted 5-D features, whose detection effect is better than statistical distortion based steganalysis. For statistical distortion, we adopt the 6-D feature set proposed by Xiang et al. By combining the above semantic distortion features and statistical distortion features, we get an 11-D feature set whose detection performance is better than any other feature sets. Our future work includes applying improved semantic distortion to other linguistic steganography such as CoverTweet.

Acknowledgements. This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 61572452.

References

1. Pevný, T., Fridrich, J.: Benchmarking for steganography. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 251–267. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88961-8_18
2. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, New York (2009)
3. Chapman, M., Davida, G.: Hiding the hidden: a software system for concealing ciphertext as innocuous text. In: Han, Y., Okamoto, T., Qing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 335–345. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0028489

4. Liu, Y., Sun, X., Liu, Y., Li, C.T.: MIMIC-PPT: Mimicking-based steganography for microsoft power point document. Inf. Technol. J. **7**(4), 654–660 (2008)
5. Chen, Z., et al.: Linguistic steganography detection using statistical characteristics of correlations between words. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 224–235. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88961-8_16
6. Bolshakov, I.A.: A method of linguistic steganography based on collocationally-verified synonymy. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 180–191. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30114-1_13
7. Yuling, L., Xingming, S., Can, G., Hong, W.: An efficient linguistic steganography for Chinese text. In: 2007 IEEE International Conference on Multimedia and Expo, pp. 2094–2097. IEEE (2007)
8. Muhammad, H.Z., Rahman, S.M.S.A.A., Shakil, A.: Synonym based Malay linguistic text steganography. In: Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009, pp. 423–427. IEEE (2009)
9. Shirali-Shahreza, M.H., Shirali-Shahreza, M.: A new synonym text steganography. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2008, pp. 1524–1526. IEEE (2008)
10. Wilson, A., Ker, A.D.: Avoiding detection on twitter: embedding strategies for linguistic steganography. Electron. Imaging **2016**(8), 1–9 (2016)
11. Winstein, K.: Lexical steganography Through Adaptive Modulation of the Word Choice Hash (1998, unpublished). http://www.imsa.edu/~keithw/tlex
12. Wilson, A., Blunsom, P., Ker, A.D.: Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In: IS&T/SPIE Electronic Imaging, p. 902803. International Society for Optics and Photonics (2014)
13. Taskiran, C.M., Topkara, U., Topkara, M., Delp, E.J.: Attacks on lexical natural language steganography systems. In: Electronic Imaging 2006, p. 607209. International Society for Optics and Photonics (2006)
14. Yu, Z., Huang, L., Chen, Z., Li, L., Zhao, X., Zhu, Y.: Detection of synonym-substitution modified articles using context information. In: Second International Conference on Future Generation Communication and Networking, FGCN 2008, vol. 1, pp. 134–139. IEEE (2008)
15. Chen, Z., Huang, L., Miao, H., Yang, W., Meng, P.: Steganalysis against substitution-based linguistic steganography based on context clusters. Comput. Electr. Eng. **37**(6), 1071–1081 (2011)
16. Xiang, L., Sun, X., Luo, G., Xia, B.: Linguistic steganalysis using the features derived from synonym frequency. Multimed. Tools Appl. **71**(3), 1893–1911 (2014)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
19. Ji, S., Yun, H., Yanardag, P., Matsushima, S., Vishwanathan, S.: WordRank: Learning word embeddings via robust ranking. arXiv preprint arXiv:1506.02761 (2015)
20. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
21. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)