STBS: A Statistical Algorithm for Steganalysis of Translation-Based Steganography

Peng Meng

mengpeng@mail.ustc.edu.cn

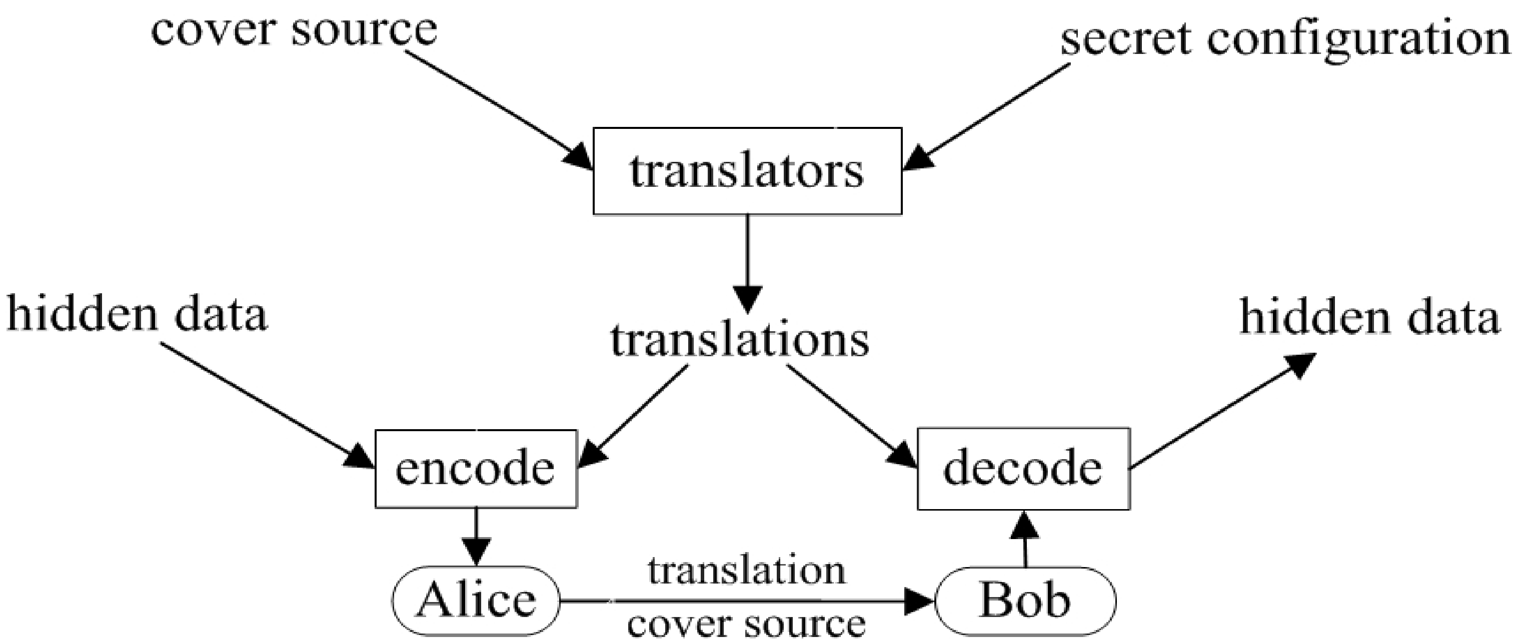
June 30, 2010

Outline

- Background
- Motivation
- Frequency Differences
- Features generation
- Experiment results
- Conclusion

Background

1. Lost in Translation (Lit)

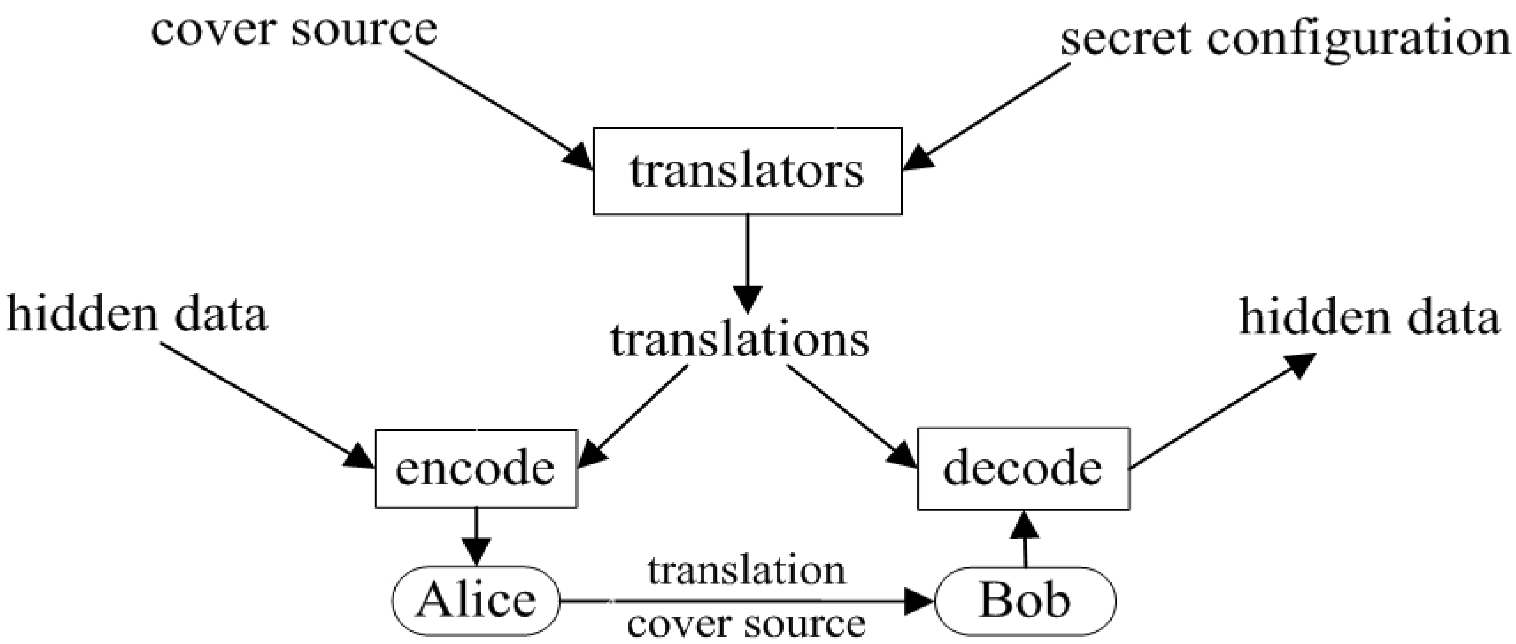


The sender:

- Get a cover text in the source language.
- Translate the cover text by many translators.
- Selects one of the translation results.
- Sends the cover text and stegotext to the receiver.

Background

1. Lost in Translation (Lit)

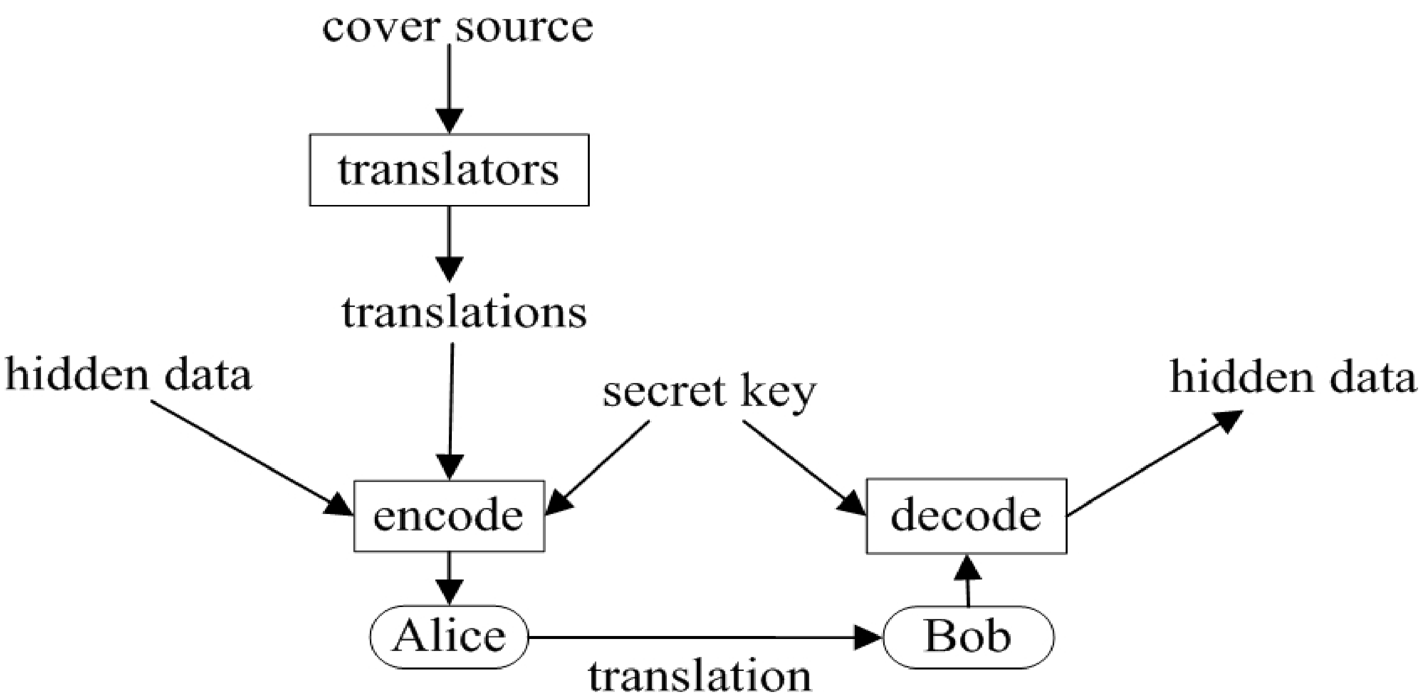


The receiver :

- Performs the same translation process.
- Compare the stegotext and the translation results to extract hidden message.

Background

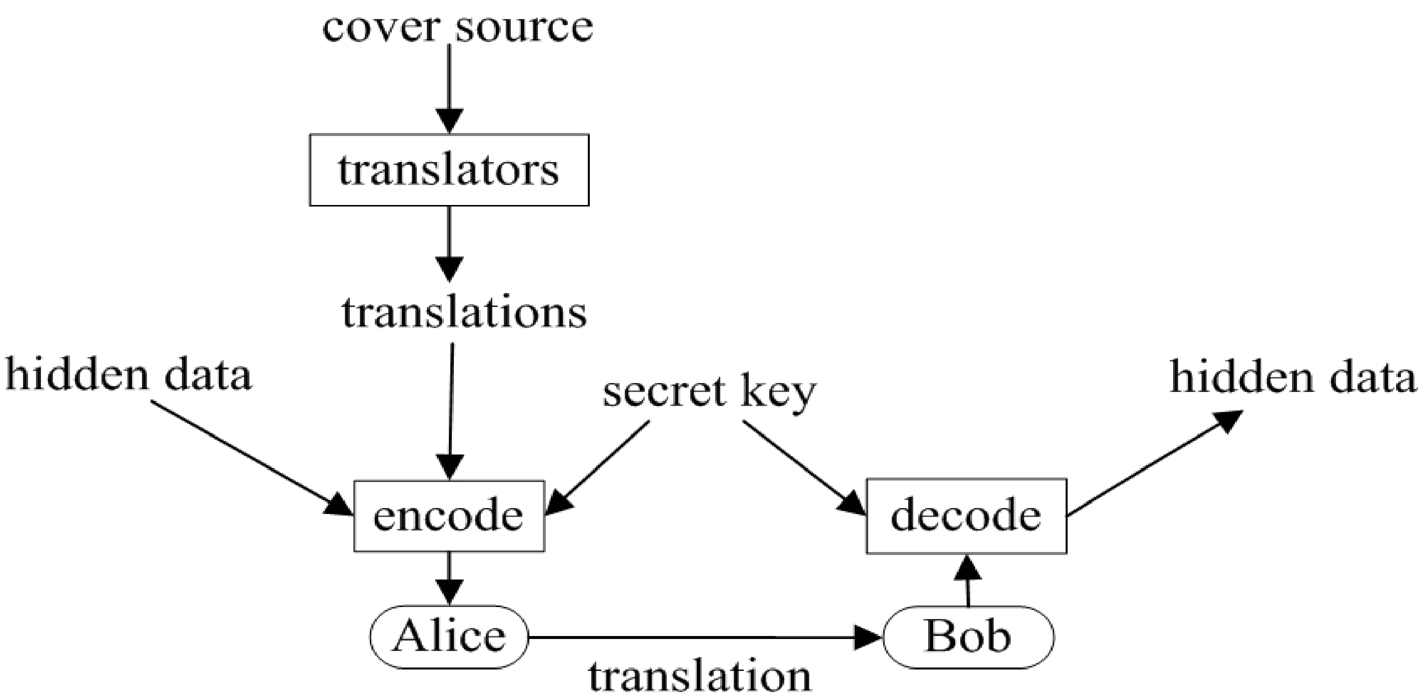
2. Lost in Just the Translation (LiJtT)



The sender:

- Get multiple translation results.
- Hash the individual translated sentences into bit strings.
- Select one of the translation results.
- Transfer the stegotext to the receiver.

2.Lost in Just the Translation (LiJtT)



The receiver:

- Breaks the received text into sentences.
- Hash each sentence into bit strings.
- Extract hidden bits from hash strings.

Outline

- Background
- Motivation
- Frequency Differences
- Features generation
- Experiment results
- Conclusion

Motivation

- **Difficult to distinguish the stegotext from originally translated text.**
  - Machine translated texts are very “noisy”.
  - Each sentence of stegotext comes from different machine translators.
- **Our problem**
  - How to distinguish the stegotexts from natural language texts.
  - How to distinguish the stegotexts from machine translated texts.

Outline

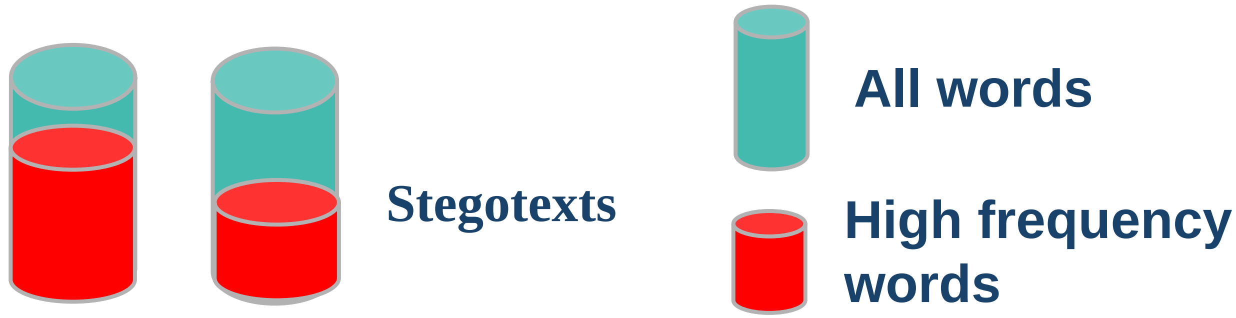
- Background
- Motivation
- Frequency Differences
- Features generation
- Experiment results
- Conclusion

Frequency differences

Word frequency differences

- **Findings**

Natural language texts
machine translated texts

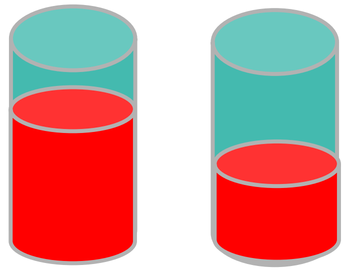


Frequency differences

Word frequency differences

❖ Findings

Natural language texts
machine translated texts



Stegotexts

All words

High frequency words

❖ Example

German | To | English

Google translator → Company

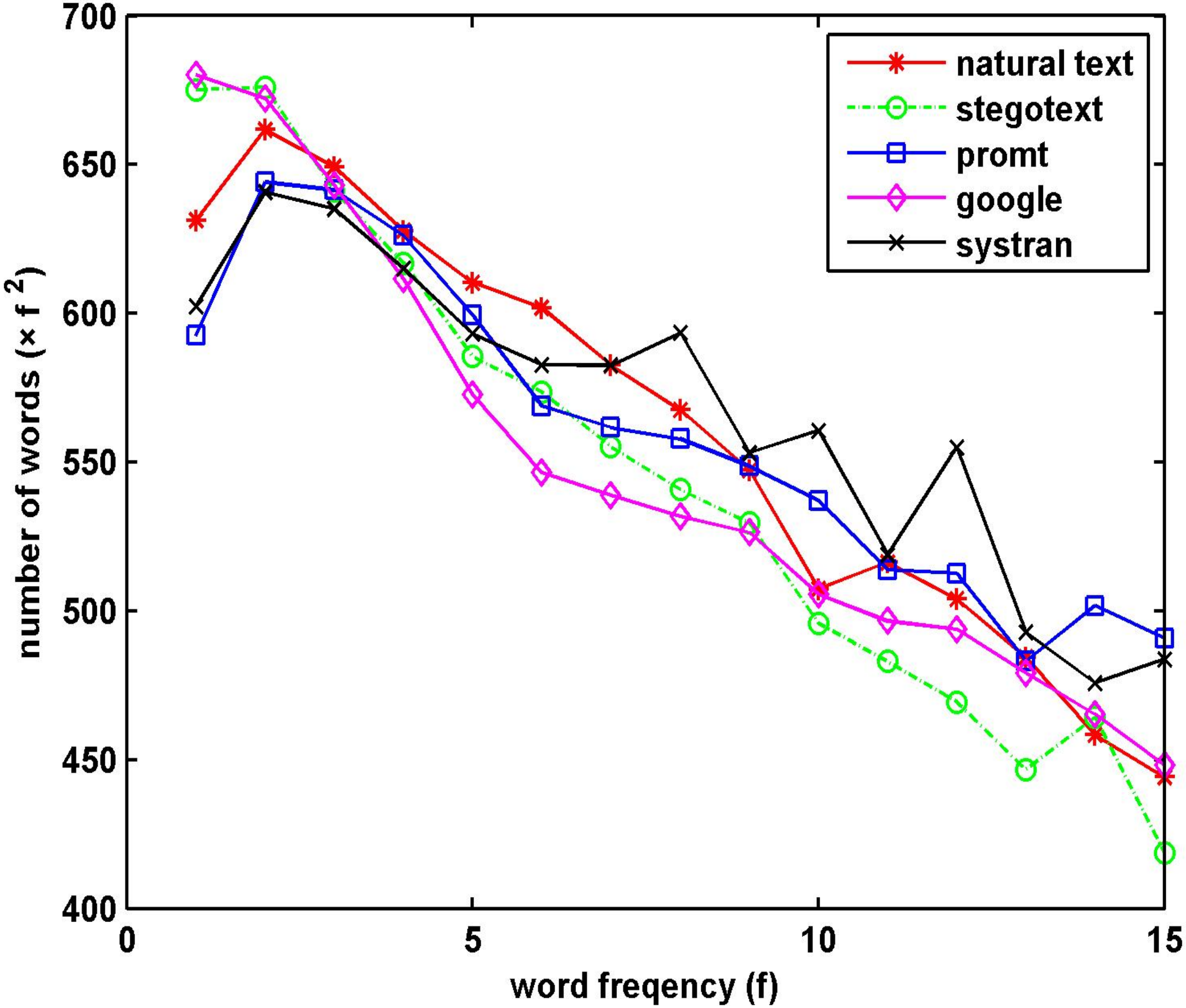
Unternehmen — Systran translator → Enterprise

TBS → Company Enterprise

A high frequency word in cover text

Word frequency differences

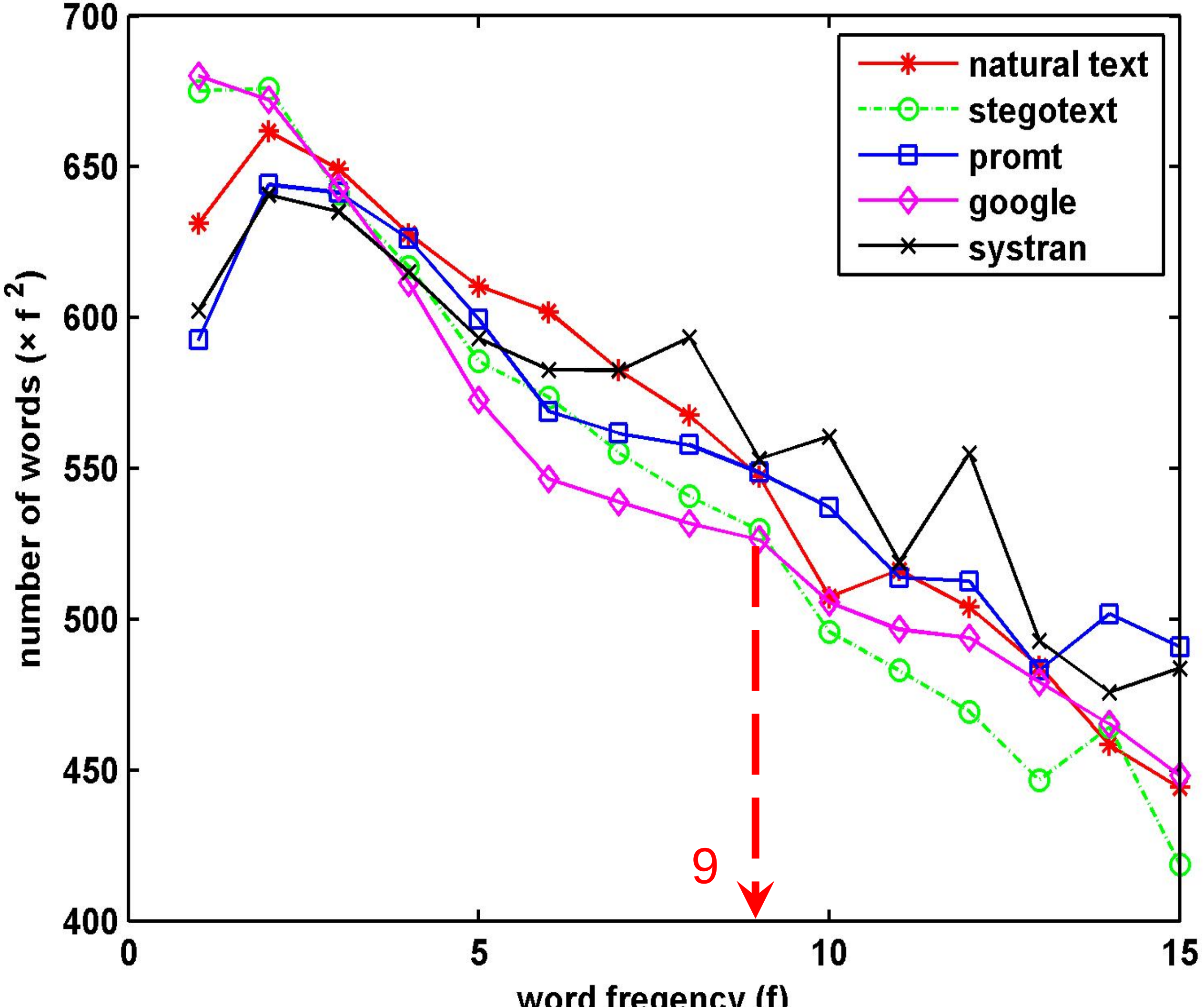
❖ Statistical Results



Stegotexts have fewer high-frequency words!

Word frequency differences

❖ Statistical Results

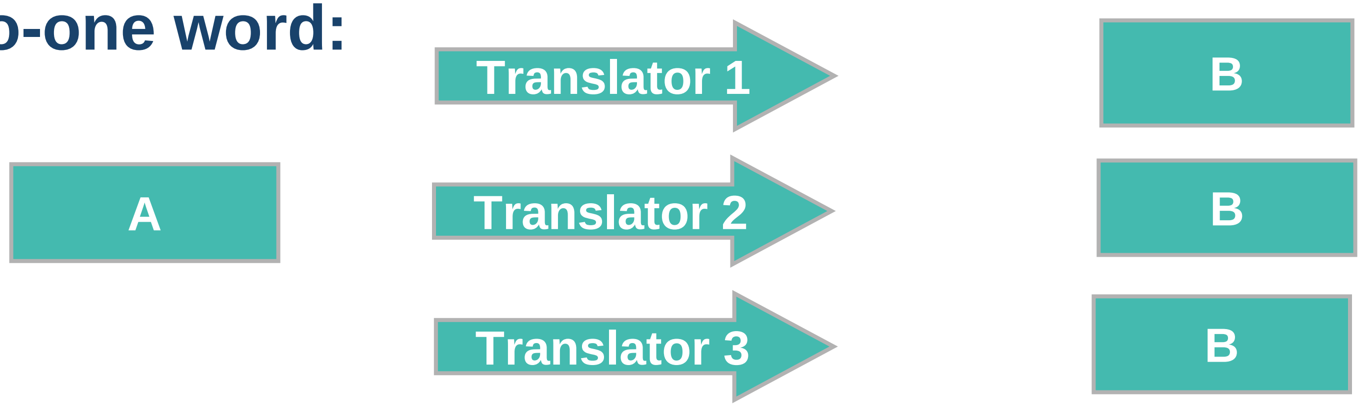


Stegotexts have fewer high-frequency words!

Frequency differences

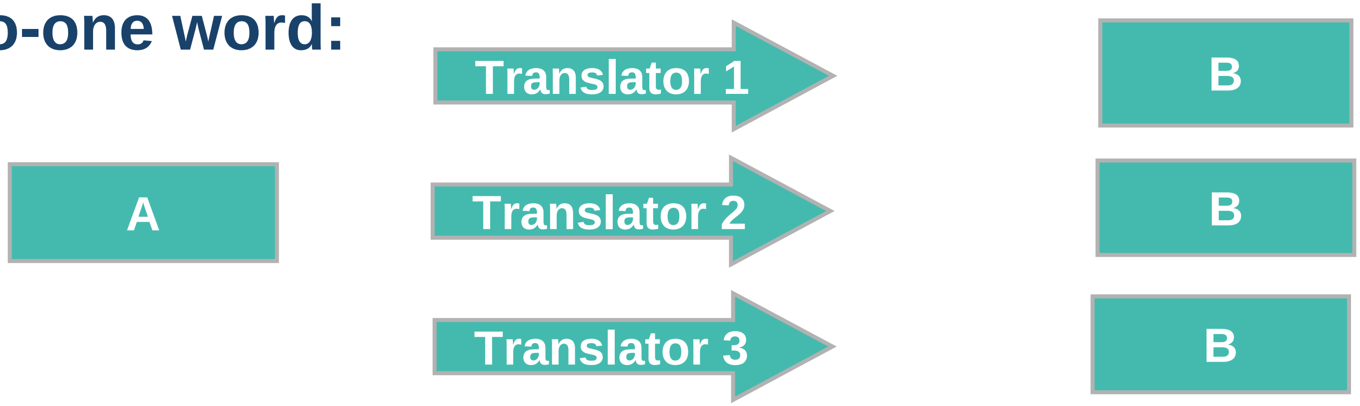
Expand word frequency differences

- One-to-one word:

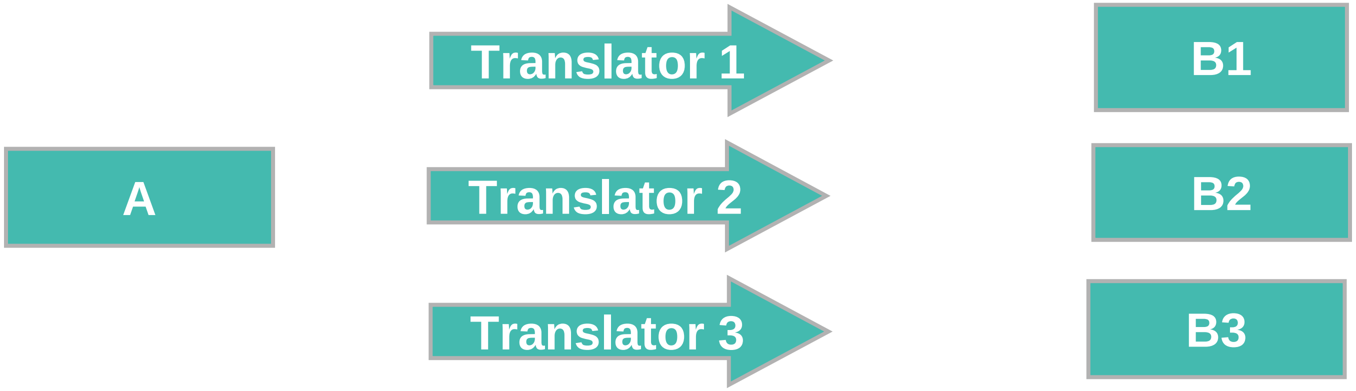


Expand word frequency differences

- One-to-one word:



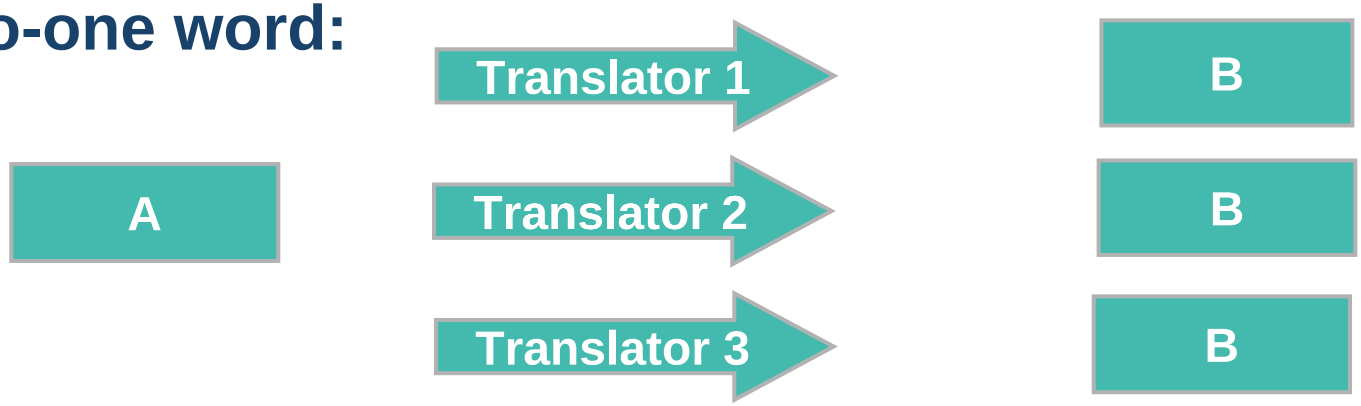
- One-to-many word:



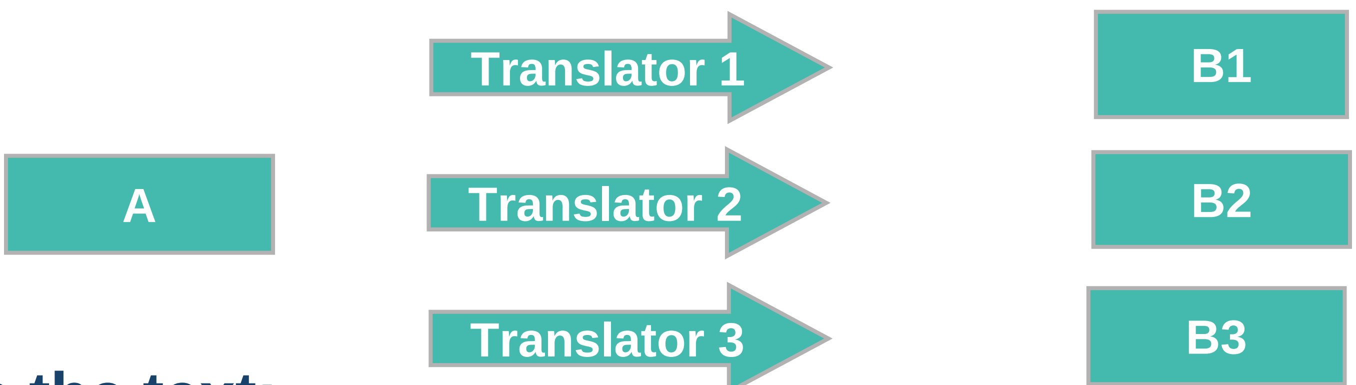
Frequency differences

Expand word frequency differences

- One-to-one word:



- One-to-many word:

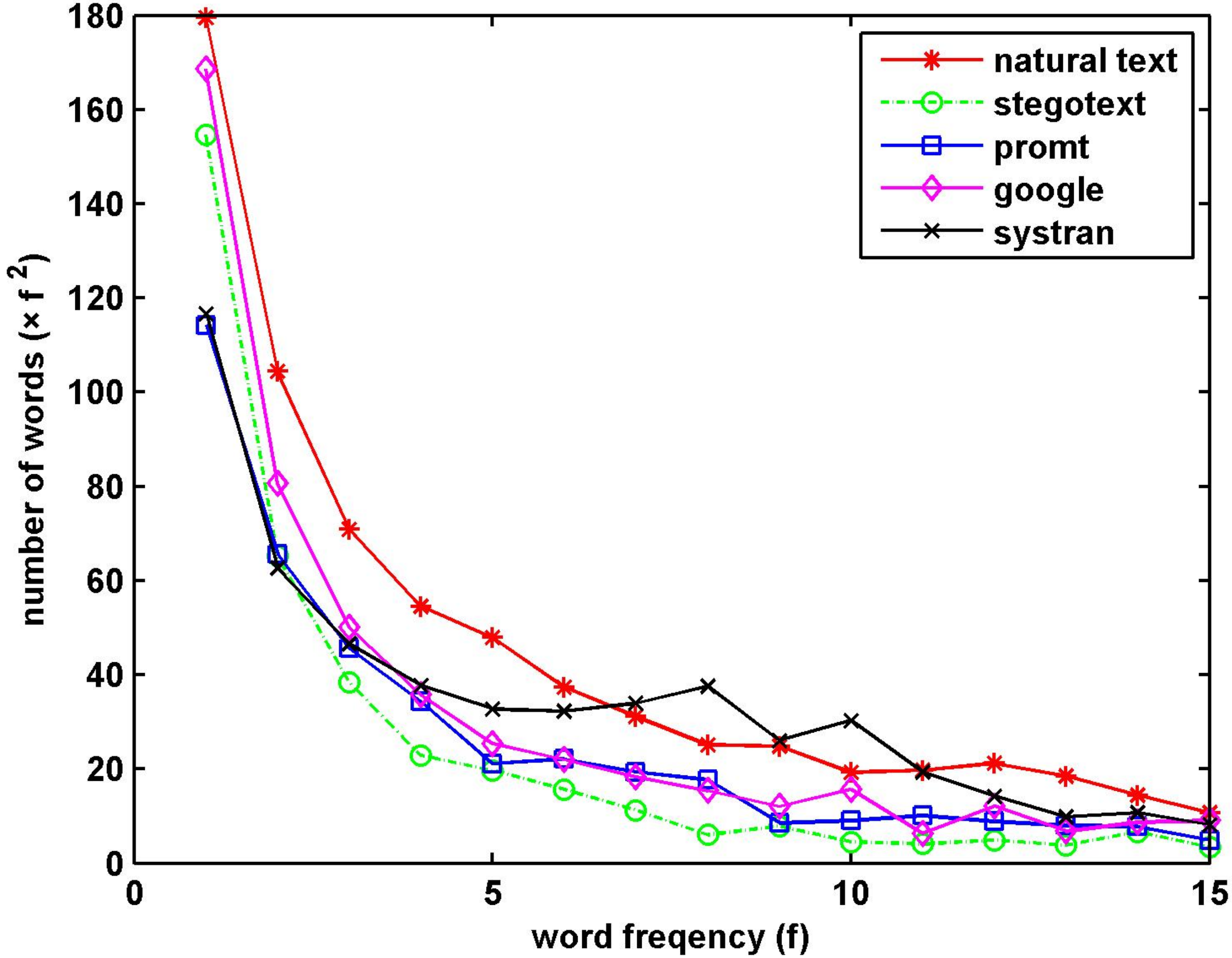


- **Refine the text:**

If all the one-to-one words are deleted ,
then the word frequency differences will be expanded

Expand word frequency differences

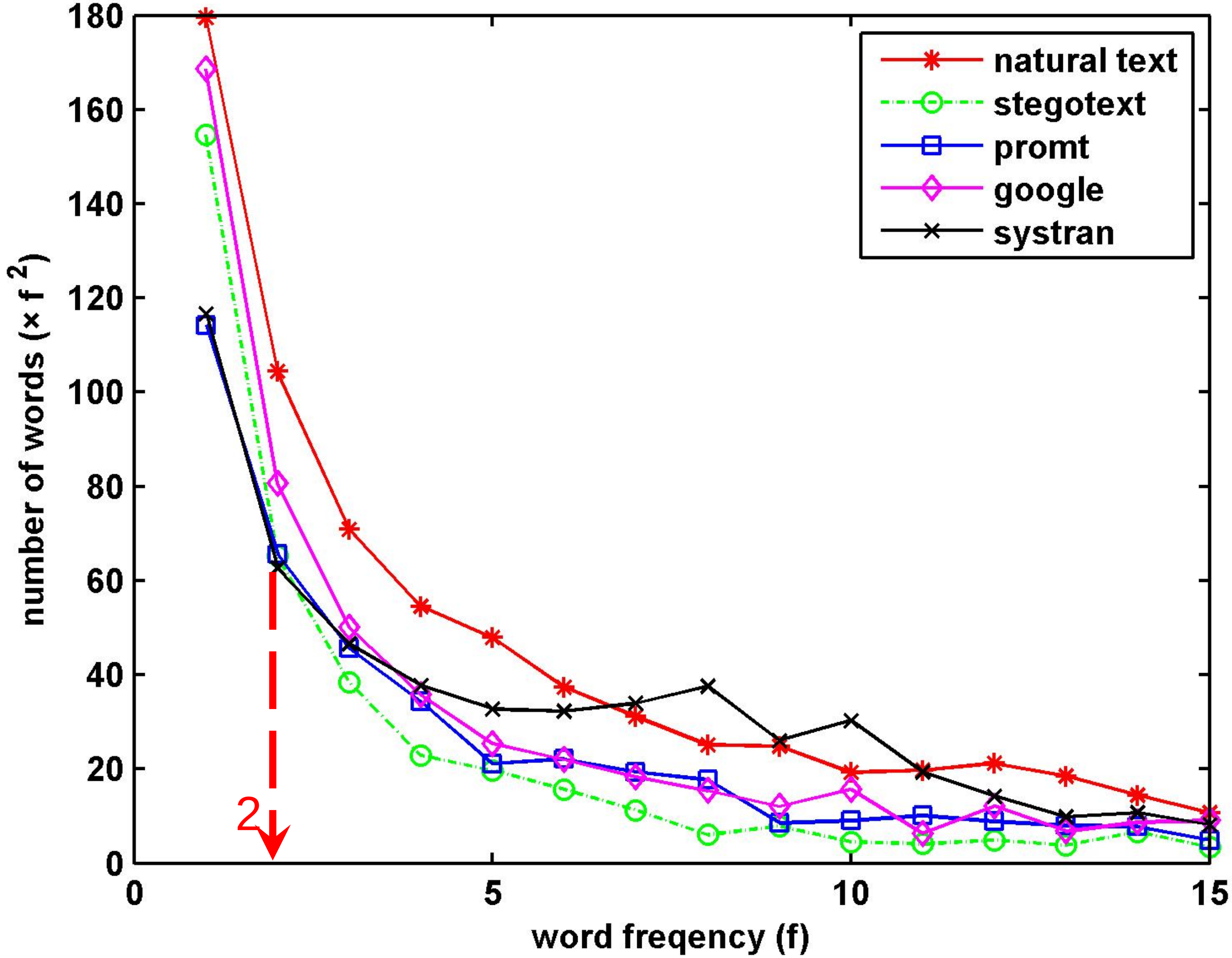
❖ **Statistical Results**



The word frequency differences are expanded!

Expand word frequency differences

❖ Statistical Results



The word frequency differences are expanded!

Frequency differences

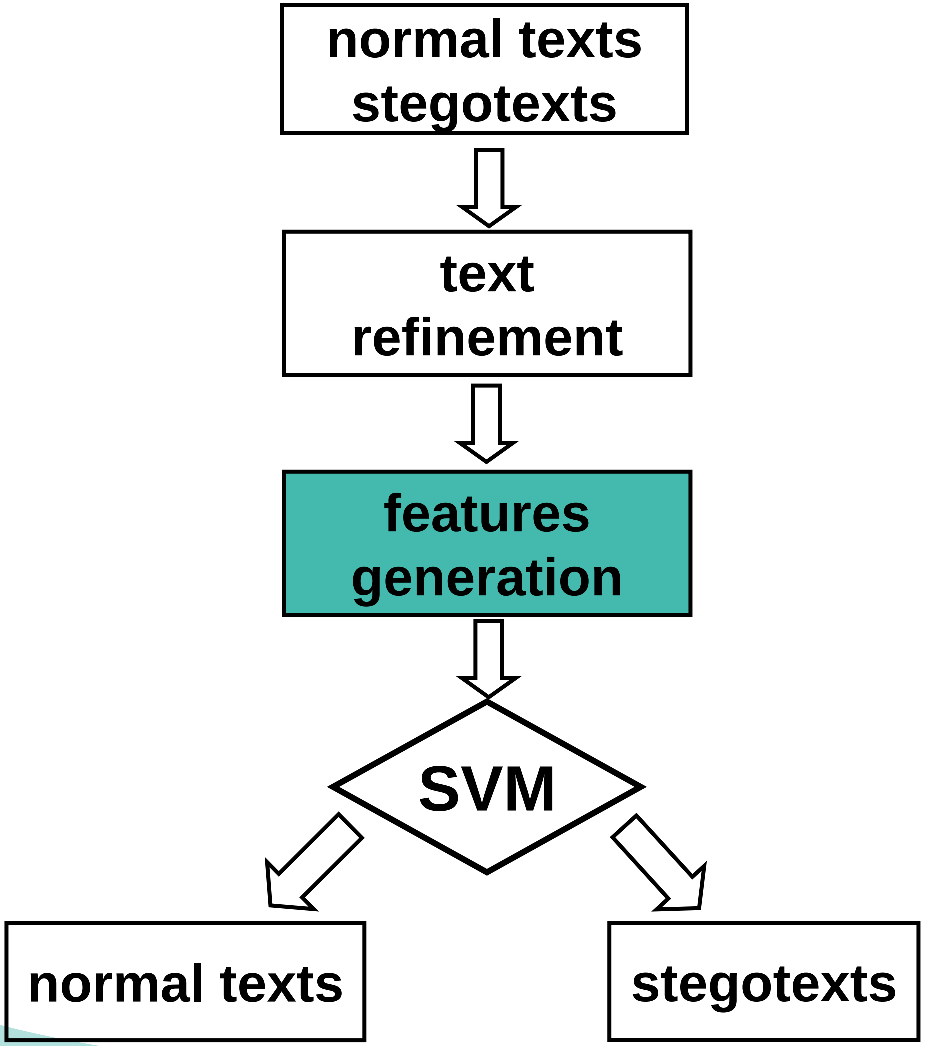
2-gram frequency difference

- 2-gram means 2 adjacent words.
- 2-gram frequency differences between normal texts and stegotexts are similar as word frequency differences.
- Deleting one-to-one 2-grams from both normal texts and stegotext also expands the 2-gram frequency difference between them.

Outline

- Background
- Motivation
- Frequency differences
- Features generation
- Experiment results
- Conclusion

Detection scheme



Our detection schema is an instance of two-class pattern recognition. A given text needs to be classified as either a stegotext (with hidden data) or as normal text (without hidden data).

Select features relevant to frequency differences is very important!

❖ Text formalization by word

The refined text by 1-gram (word) $\xrightarrow{\text{formalize}}$ $T1 = \{n_1^1, n_2^1, \ldots, n_{m_1}^1\}$

$n_i^1, 1 \leq i \leq m_1$

m1: highest word frequency

n^1_i: the number of different words whose frequency is i

Text formalization by word

The refined text by 1-gram (word) $\xrightarrow{\text{formalize}}$ $T1 = \{n_1^1, n_2^1, ..., n_{m_1}^1\}$

$n_i^1, 1 \le i \le m_1$

m1: highest word frequency

n^1_i: the number of different words whose frequency is i

Example:

"If a word is one to one word in the source language, its translation in the destination language is also called one to one word"

8 words appear one time: If, a, source, its, translation, destination, also, called.

5 words appear two times: to, language, in, the, is.

1 word appears three times: word.

1 word appears four times: one.

T1={ 8,5,1,1 }

Text formalization by 2-gram

The refined text by 2-gram $\xrightarrow{\text{formalize}}$ $T2 = \{n_1^2, n_2^2, \ldots, n_{m_2}^2\}$

$n_i^2, 1 \le i \le m_2$

m2: highest 2-gram frequency

n^2_i: the number of different 2-grams whose frequency is i

Features generation

❖ Text formalization by 2-gram

The refined text by 2-gram $\xrightarrow{\text{formalize}}$ $T2 = \{n_1^2, n_2^2, ..., n_{m_2}^2\}$

$n_i^2, 1 \leq i \leq m_2$

m2: highest 2-gram frequency

n^2_i: the number of different 2-grams whose frequency is i

Example:

"If a word is one to one word in the source language, its translation in the destination language is also called one to one word"

15 2-grams appear one time: If a, a word, word is,....

4 2-grams appear two times: one to, to one, one word, in the.

T2={ 15,4 }

Features generation

- Features generation formula

$$F_{b,c}^{a} = \sum_{i=c}^{m_b} (n_i^b * i^a)$$

a={0,1,2}
b={1,2};
c={1,5}

Features generation

- **Features generation formula**

$$F_{b,c}^{a} = \sum_{i=c}^{m_b} (n_i^b * i^a)$$

a={0,1,2}
b={1,2};
c={1,5}

- **Features generation**

We totally choose 12 features for SVM classifier.

Word features $F_{1,1}^{0}, F_{1,1}^{1}, F_{1,1}^{2}, F_{1,5}^{0}, F_{1,5}^{1}, F_{1,5}^{2},$

2-gram features $F_{2,1}^{0}, F_{2,1}^{1}, F_{2,1}^{2}, F_{2,5}^{0}, F_{2,5}^{1}, F_{2,5}^{2}$

Features generation

- **Features generation formula**

$$F_{b,c}^{a} = \sum_{i=c}^{m_b} (n_i^b * i^a)$$

a={0,1,2}
b={1,2};
c={1,5}

- **Features generation**

We totally choose 12 features for SVM classifier.

Word features $F_{1,1}^0, F_{1,1}^1, F_{1,1}^2, F_{1,5}^0, F_{1,5}^1, F_{1,5}^2,$

2-gram features $F_{2,1}^0, F_{2,1}^1, F_{2,1}^2, F_{2,5}^0, F_{2,5}^1, F_{2,5}^2$

Features	Means of the features
$F_{1,1}^0$	The number of word types, or in other words, the number of different words in the refined text.
$F_{1,1}^1$	The number of word tokens in the text, or the total number of words in the refined text.

Outline

- Background
- Motivation
- Frequency differences
- Features generation
- Experiment results
- Conclusion

Experiment Results

Text size	Text type	Accuracy (%) No refine	Accuracy (%) Refine
20K	Natural	98.2	100
	Translated	81.8	77.2
	Stego	86.1	89.3
40K	Natural	100	100
	Translated	81.8	92.9
	Stego	92.2	94.6

Our detection scheme can successfully distinguish stegotexts from natural language texts and originally translated texts.

Outline

- Background
- Motivation
- Frequency differences
- Features generation
- Experiments and results
- Conclusion

Conclusion

- TBS is a new kind of text steganography.
- We found a weakness of TBS: there are fewer high-frequency words in TBS generated texts.
- Based on the weakness we found, we gave a method to expand the difference between normal texts and stegotexts, and finally designed a method to steganalysis of TBS.
- The results show our scheme is promising on detecting TBS.