# Study of Chinese Text Steganography using Typos

Linna Zhou[1, 2] and Derui Liao[2]
[1] Beijing University of Posts and Telecommunications, Beijing, China
[2] University of International Relations, Beijing, China
E-mail: zhoulinna@mail.tsinghua.edu.cn
E-mail: *liaoderui@126.com

***Abstract*— Nowadays, with the Information Explosion and the rapid development of information technology, huge amounts of data are constantly being generated every day on the Internet. But most of the texts provided online is of a kind that usually contain many typos, which is very common among individual users, self-media, etc. However, disambiguation is human's talent, so these typos often do not frustrate human understanding the text, and sometimes it is even difficult to recognize some typos. This phenomenon appears both in English and Chinese, so it seems to be cross-lingual. Therefore, in such texts, it is not surprising that one can perform information-hiding by judiciously injecting typos. We studied Chinese typos in the text contents on Weibo or WeChat, and propose a text steganography method based on Chinese typos with the help of NLP, which can embed secret information by carefully injected typos and guarantee the security of the secret and the readability of the texts. Unlike format-based steganography algorithms, our algorithm can resist format adjustments, OCR re-inputs, etc. Furthermore, Weibo and WeChat platform contain many kinds of media, so by combining other algorithms, Cross-Media or even Cross-Social Network information hiding is practical.**

## I. INTRODUCTION

Steganography can hide secret information in the carrier and avoid the adversary discovering the existence of the secret information. Typically, the carrier covers images, soundtracks, text and some other media. Hiding secret information in text can take advantage of we use text as the most commonly information carrier on the Internet. However, comparing to the steganography using images as carriers, it seems hard to apply steganography in text. But compared with image, audio, video and other carriers, text have a wider range of applications and is easier to transfer. At present, text steganography research mainly focuses on two aspects: text format and text content. Steganography with text format can be achieved by adjusting invisible characters [1][2]. For steganography using text content, for example, synonym replacement [3][4] can be performed. Thanks to the development of nature language processing (NLP) and machine learning, there has been some researches hiding information in the text generated by neural network [5].

Noticing that typing errors (typos) exist widely in a variety of language texts, especially the text on the internet such as blog, online chat, email or Weibo/Twitter. Sometimes, they not only do not affect the understanding of the text, but even difficult to be discovered. Therefore, this phenomenon can be used to hide information.

Using the Chinese version of the BERT model, this paper proposes a method that can inject secret message into the Chinese plain text carrier by crating typos and at the mean time avoid affecting our understanding of the text. And we can recover the secret message without the original text carrier. This method can prevent the adversary from formatting modification, clearing format, modifying invisible characters, file format conversion and machine re-entry, because they do not affect the carrier that carries the secret.

## II. RELATED WORK

### *A Chinese typing errors*

The study of Chinese typos has a long history, mainly in the education and publishing, with the aim of reducing the occurrence of typos. Typing errors are a type of typo made by inputting with a keyboard. Geng [6] studied the relationship between Pinyin input method and network typos. He and Chen [7] studied the causes for typos.

### *B Masked Language Model, BERT and ERNIE*

In 2018, Google introduce a new language representation model, the BERT [8] (Bidirectional Encoder Representation from Transformers) pre-trained model, which is a state-of-the-art model until then. The Masked Language Model (Masked LM, MLM) is one of the tasks in pre-training. Masking some percentage of the input tokens at random, and predicting those masked tokens let the model to 'learn' the language in an unsupervised way and build the language model. However, BERT's MLM is modeled by words in English and character in Chinese, which might make it difficult for the model to learn the complete semantic representation especially in Chinese. ERNIE [9] (Enhanced Representation through Knowledge Integration) overcome that difficulty and perform better in Chinese.

### *C Text steganography and text steganography based on typing errors*

Hiding information in plain text can modify characters that do not contain semantics such as invisible characters or punctuation marks [10]. Gan [3] use the synonym and Shirali-Shahreza [11] use the abbreviation for embedding. Similar to those using neural networks to generate images for information hiding, Luo et al. [5] generate Tang poems carrying secret message by using neural networks.

Wayner [12] first hide information by modifying the order of characters in English words. Liu [13] use matrix coding to hide information in English text. Topkara [14] use computationally asymmetric transformations (CAT) to resist synonym replacement attack and use typing errors which is another existing word to avoid spelling check.

For example, we use Liu's method[13] to hide 'cn' (bit stream = \0x63\0x6E\0x00, 24 in length) in our abstract:

> Now**da**ays, with the Information Explosion and the rapid develop**em**nt of information technology, huge amounts of data are constantly being generated every day on the Inter**en**t. But most of the texts provided online is of a kind that usually contain many t**py**os, which is very common among individ**au**l users, self-media, etc. However, disambiguation is human's talent, so these typos often do not frust**ar**te human understanding the text, and sometimes it is even difficult to recognize some typos. This phenomenon appears both in English and Chinese, so it seems to be cross-lingual. Therefore, in such texts, it is not surprising that one can perform information-hiding by judiciously injecting typos. We studied Chinese typos in the text contents on Weibo or WeChat, and propose a text steganography method based on Chinese typos with the help of NLP, which can embed secret information by carefully injected typos and guarantee the security of the secret and the readability of the texts. Unlike format-based steganography algorithms, our algorithm can resist format adjustments, OCR re-inputs, etc. Furthermore, Weibo and WeChat platform contain many kinds of media, so by combining other algorithms, Cross-Media or even Cross-Social Network information hiding is practical.

These methods based on typing errors/typos performs well in English text, but the research on Chinese text seems relatively rare.

## III. TYPING ERRORS (TYPOS) IN CHINESE SOCIAL MEDIA TEXT

Every day, huge amounts of information are produced on social media web sites, including images, audio, video, and lots of text. At the same time, with the rapid development of China's technology and the Internet, the number of Chinese Internet users has expanded rapidly. So Chinese messages on the Internet should be countless. Weibo (Sina) and WeChat(Tencent) can be considered as the two giants in Chinese social media, including more than 462 million monthly active users of Weibo, and WeChat has more than 1 billion. Huge amounts of information are bound to be generated every day by the huge amount of user, and we found typing mistakes usually appears in text posted on Weibo or WeChat. The existence of typos is so extensive that sometimes typos can appear in text posted by official accounts when it is real-time-posted. Typos also appear in some long articles published on Weibo or WeChat Official Accounts, but this always does not affect the reader's understanding. The contents posted by personal accounts are usually more casual with more typos, and some Weibo bloggers even make typos as his blogging style. Sometimes, typos can help making humors in social media.

Moreover, more and more people are surfing the Internet on smart phones. Using a smart phone to surf the Internet must face the difficulty of inputting via the virtual keyboard displayed on the screen. Although the screen on smartphones is getting larger and larger, the virtual keyboard is much smaller than the real one and there is no gap among the keys. Therefore, typos are more likely to occur when typing on smartphones. The typos cannot be completely avoided even though some input methods try hard in spell correction.

In English, words are composed of letters and divided by spaces. But the Chinese character set is too large to be arranged on the keyboard, so we have several Chinese input methods such as Pinyin, Wubi, Bihua, Writing and so on. For the keyboard, we have 9 key (T9) and 26 key (T26). Different Chinese input method have to solve different problems, such as one-to-many mapping and handwriting recognition. So it can be inferred that Chinese typos contain the following types:

- Selecting the wrong word(s)
- Pressing the wrong key (keys)
- Pressing more/less key (keys)
- Duplicated/Missing character(s)/word(s)
- Others

Selecting the homonym, which means selecting the wrong word with the same pinyin sequence or a similar pinyin sequence in the candidates list, is common when selecting the wrong word. For pressing the wrong key, homonym might occur if another final key is pressed. But a wrong initial key leads a completely wrong word, sometimes might be confusing. For example, typing 我好羡慕啊 in T9 keyboard might press 964269426682 in order or 96 for short (with the help of the suggestion supported by the input method). But if 99 is pressed, the result might be another word: 真相大白天, a completely wrong word. Pressing more/less key might lead to a similar result like pressing the wrong key.

Since Chinese typos are completely different from the ones in English, the English text steganography algorithms based on input errors available are sometimes difficult in applying to Chinese

## IV. STEGANOGRAPHY METHOD

### *A Embedding*

BERT can predict the masked token in the context and its probability, so we can let BERT to predict each word in the context and determine whether it is the same as the original word. Therefore, we can recover some missing or wrong word in the context by using BERT. However, BERT's bidirectional structure makes it difficult recover the original token when modifying more than one tokens each time. Different from English that have separators between words, Chinese separate the words by its semantics. To get Chinese word, Chinese Word Segmentation is required. But the result of Chinese Word Segmentation might change if the context is modified.

Selecting the wrong homonym is selected here to create typos. We replace a character by another character with the same or a similar pronunciation each time. There is one transform at most in each instance. A word with more than two characters is chosen to carry information. We use a very simple method to carry secret bit of the secret message: the index of the character being modified represents the secret bit.

This method can encode more than one bit in a single word when the word has more than 2 characters. But this situation is rare and it may increase the complexity of the steganography method. A zero bit needs only one character to embed, but it may confuse the program when extracting. Therefore, we only transform one of the first two characters of a word that has more than two characters even if the word consists of more than two characters.

However, similar to Topkara's work [14], some words should not be modified. Otherwise, it may cause confusion in understanding. This is discussed in the following section: C Untouchable words.

Embedding follows the following steps:

a) Tokenize the text and divide them into segments by an *instance separator*. Then merge the segments into instances of length less than a given length: $len_{instance}$ (including *[CLS]* and *[SEP]* markers). Then mask and predict one token each time to get each token's prediction and probability in the text.
b) Apply Chinese Word Segmentation to each instance.
c) Keep the tokens that:
   i. The token is a Chinese character, not Number, English word, symbol, emoji, etc.
   ii. The predicted token is the same as the actual token, which we called *hit*
   iii. The probability of the predicted token is greater than $P_{embed_{token}}$
   iv. The token is not in the stop word table
   v. The character of the token has more than one homonym
d) An instance contains homonym predicted token should be dismissed because it might cause fault when extracting secret message. Although we can inject more than two typos in one instance, or replace it with the predicted token to solve this problem, it seems unworthy here.
e) Keep the words that
   i. Only contain tokens selected from step c)
   ii. Contain more than 2 characters
   iii. All characters *hit* c)ii
   iv. The average probability of its tokens is greater than $P_{embed_{word}}$
   v. Its characters should not be tokenized, but we never found BERT to tokenize Chinese characters, so it is free from considering this.
f) Select one word from step e) in each instance to transform for embedding a single bit:
   i. If the secret bit is 0: Replace the first character with its homonym
   ii. If the secret bit is 1: Replace the second character with its homonym
g) Embed all the bits of the secret message into the carrier. Choose another instance separator or instance if it is not big enough for embedding all the secret bits.

### *B Extracting*

After embedding, the Chinese Word Segmentation result of the text has a great probability change. Therefore, it is necessary to find out the typos and recover it before applying Chinese Word Segmentation on the text and figure out which character is transformed in a word.

We extract the secret message by the following steps:

a) Tokenize the text and divide them into instance by an *instance separator*. Then mask and predict one token each time to get each token's prediction and probability in the text. The same as the first step in A Embedding.
b) Keep the tokens that:
   i. Never hit
   ii. The prediction probability is greater than $P_{extract_{token}}$ ( $P_{extract_{token}}$ is greater than $P_{embed_{token}}$ or $P_{embed_{word}}$)
   iii. The token is not in the stop word table
   iv. The predicted token is a homonym of the actual word
c) Each instance should only have one eligible token, otherwise error occurred.
d) Replace the tokens filtered from step b), and mark its position
e) Apply Chinese Word Segmentation to the instance and get the word with the modified token
f) The index of the modified character in the word represents the secret bit
g) Get all the secret bit and recover the secret message

It might seems a little bit confusing and an example might help, the example will be shown in V EXPERIMENT AND EXAMPLE.

### *C Untouchable words*

Like the text steganography using typos in English, modifying some characters in Chinese might leads to confusion. What's more, a Chinese character usually carry more information than an English character, which makes is more difficult in embedding. Bad modification can cause ambiguity, distorted facts, or even political errors. These problems are currently solved by the following methods:

a) The token that *hit* with a high probability is usually occur in many texts because the model has learnt it from a huge amount of data. For humans, the modification of these words generally does not affect understanding. And the low embedding rate can also reduce the occurrence of this problem.
b) A stop word list is maintained to filter out the characters that may constitute untouchable words and skip them when embedding. Although this method can be effectively, maintaining a all-inclusive stop word list costs more than gain. So, we only put the dangerous characters in the stop word list.
c) Attention Mechanism in machine translation can figure out words that are important for understanding the text. And we can only transform the unimportant tokens in the text. Because the BERT uses Transformer that contains Attention Mechanism, it seems convenient to apply it here, although it is not implemented.
d) Limiting the use scope also works. We studied typos in Weibo and WeChat, which is not a formal situation. The users interact frequently, and they can complete disambiguation through communication.

## V. EXPERIMENT AND EXAMPLE

### A *Our method*

We run the Pytorch version BERT model in Python whose parameters are 12-layer, 768-hidden, 12-heads, and 110M parameters Chinese models. Using a period (。) as the instance separator, the Chinese version of our abstract can be divided in to 5 instances, which can carry 5 bits at most. Our Chinese abstract carries secret message 0b1101 is as follows:

如今在一个**网落（网络）**技术快速发展的信息爆炸的时代，网络上每天都有巨量的数据在不断地产生。而网络上的文本往往包含许多拼写错误，这一点在个人用户、自媒体等发布的内容中尤其地明显。并且这种错误往往不影响人类对文本意义的理解，有时甚至难以被发现。这个**限象（现象）**在英文中有出现，在中文中更是如此，所以这一现象似乎是跨语言的。因此，在这样的文本中，人们可以通过明智地输入拼写错误来执行信息隐藏，这并不奇怪。本文**计话（计划）**针对新浪微博或者微信上的文本内容进行中文打字错误的研究，以期望能利用现今的自然语言处理技术研究出基于中文打字错误的信息隐藏算法，该算法能利用精心伪造的打字错误将秘密信息嵌入到中文文本中，并保证文本的可读性与秘密信息的隐蔽性。不同于基于格式的隐写**算发（算法）**，该算法可以抵抗格式调整、OCR 重新输入等。并且由于微博、微信包含了多种媒体，该算法可以联合其他算法进行跨媒体甚至跨社交网络的信息隐藏。

And we can extract the secret message without the original text:

```
high prob unhit: ------------------
6 落 络 0.9979057312011719 0
118 限 现 0.9992528557777405 0
194 话 划 0.9971093535423279 0
318 发 法 0.9993120431900024 0
groups:------------------------------
如今在一个网落技术快速发展的信息爆炸的时代，网络上每天都有巨量的数据在不断地
如今在一个网络技术快速发展的信息爆炸的时代，网络上每天都有巨量的数据在不断地
typo at [6] : 网络 5 6 1
这个限象在英文中有出现，在中文中更是如此，所以这一现象似乎是跨语言的。因此，
这个现象在英文中有出现，在中文中更是如此，所以这一现象似乎是跨语言的。因此，
typo at [2] : 现象 2 2 0
本文计话针对新浪微博或者微信上的文本内容进行中文打字错误的研究，以期望能利用
本文计划针对新浪微博或者微信上的文本内容进行中文打字错误的研究，以期望能利用
typo at [3] : 计划 2 3 1
不同于基于格式的隐写算发，该算法可以抵抗格式调整、OCR重新输入等。
不同于基于格式的隐写算法，该算法可以抵抗格式调整、OCR重新输入等。
typo at [11] : 算法 10 11 1
并且由于微博、微信包含了多种媒体，该算法可以联合其他算法进行跨媒体甚至跨社
并且由于微博、微信包含了多种媒体，该算法可以联合其他算法进行跨媒体甚至跨社
No typo
Msg: 0b1101

Process finished with exit code 0
```

Fig. 1 Extraction program.

Using special kinds of typos can imitate a low-quality text typed by a careless human, which might cause less suspicion.

### B *Example*

We use the Chinese version abstract as the text to illustrate the embedding and extracting steps of hiding a secret message of 0b1101.

In short, embedding contains dividing, predicting, filtering and replacing.

a) Dividing

We first divide the text into segments by an *instance separator*: "。". And then we merge those segments into instance no longer than a given length $len_{instance} = \#\#\#$. Here are some examples:

Table 1 Examples of instances after dividing and masking

| **Dividing examples** |
|---|
| [CLS][MASK]今在一个网络技术快速发展的信息爆炸的时代，……有时甚至难以被发现。[SEP] |
| [CLS]如[MASK]在一个网络技术快速发展的信息爆炸的时代，……有时甚至难以被发现。[SEP] |
| [CLS]如今[MASK]一个网络技术快速发展的信息爆炸的时代，……有时甚至难以被发现。[SEP] |
| … |
| [CLS][MASK]个现象在英文中有出现，……这并不奇怪。[SEP] |
| [CLS]这[MASK]现象在英文中有出现，……这并不奇怪。[SEP] |
| … |

b) Predicting

Then we feed the instances to BERT to get the prediction of each *MASK*.

Table 2 Examples of prediction during embedding

| **Real** | **Prediction** | **Probability** | **Hit** |
|---|---|---|---|
| 如 | 如 | 0.9307 | ✓ |
| 今 | 今 | 0.8824 | ✓ |
| 在 | 是 | 0.9621 | × |
| …… | …… | …… | …… |
| 网 | 网 | 1.0000 | ✓ |
| 络 | 络 | 0.9979 | ✓ |
| 技 | 技 | 0.9999 | ✓ |
| …… | …… | …… | …… |

c) Filtering

We apply Chinese Word Segmentation to the instances, keep the valid tokens, and get the embeddable words. We set the $P_{embed_{word}} = 0.94$ and $P_{embed_{word}} = 0.92$ here.

Table 3 Example of filtering during embedding

| **Instance** | **Index** | **Word** | **Prediction** | **Hit** | **Chinese** | **Probability** | **Embeddable** | **Why** | **Homonyms** |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 如今 | 如今 | ✓ | ✓ | 0.906548 | × | low prob | - |
| 0 | 1 | 在 | 是 | × | ✓ | 0.962071 | × | not word or unhit | - |
| 0 | 2 | 一个 | 这个 | × | ✓ | 0.896691 | × | not word or unhit | - |
| 0 | 3 | 网络 | 网络 | ✓ | ✓ | 0.998935 | ✓ | - | 望、往、王……；落、罗、洛…… |
| 0 | 4 | 技术 | 技术 | ✓ | ✓ | 0.999660 | ✓ | - | 机、计、级……；数、书、属…… |
| …… | …… | …… | …… | …… | …… | …… | …… | …… | …… |

d) Replacing

Finally, we choose a random word in one instance to inject secret bit until we hide all bits into the text. Here we choose the first word for simplification.

Table 4 Examples of replacing during embedding

| Instance | Index | Original | Typo |
|---|---|---|---|
| 0 | 5 | 网络 | 网**落** |
| 1 | 118 | 现象 | **限**象 |
| 2 | 193 | 计划 | 计**话** |
| 3 | 317 | 算法 | 算**发** |

Extracting contains dividing, predicting, filtering, recovering and extracting.

a) Dividing and Predicting

Like the steps in embedding, we can get each tokens' probability, some of them are shown below.

Table 5 Examples of prediction during extracting

| Real | Prediction | Probability | Hit |
|---|---|---|---|
| 如 | 如 | 0.9395 | ✓ |
| 今 | 今 | 0.8917 | ✓ |
| 在 | 是 | 0.9576 | × |
| …… | …… | …… | …… |
| 网 | 网 | 0.3542 | ✓ |
| **落** | **络** | **0.9979** | **×** |
| …… | …… | …… | …… |

b) Filtering

Some tokens are of high probability to hit but not, that might be typos we injected. We set the $P_{extract_{token}} = 0.95$ here.

The examples are shown in Table 6 below.

c) Recovering and Extracting

We recover those typos and record their locations, then we apply Chinese Word Segmentation to the instances, and figure out the index of the homonym character in those typo words. So that we can extract the secret message bits.

Table 7 Examples of correction during extracting

| Index | Typo | Correction |
|---|---|---|
| 6 | 落 | 络 |
| 118 | 限 | 现 |
| 194 | 话 | 划 |
| 318 | 发 | 法 |

The example of extraction is shown in Table 8 below.

Table 6 Examples of filtering during extracting

| Index | Real | Prediction | Probability | Hit | stop word | Homonym | Recover |
|---|---|---|---|---|---|---|---|
| 0 | 如 | 如 | 0.939501 | ✓ | - | - | × |
| 1 | 今 | 今 | 0.891668 | ✓ | - | - | × |
| 2 | 在 | 是 | 0.957569 | × | × | × | × |
| …… | …… | …… | …… | …… | …… | …… | …… |
| 5 | 网 | 网 | 0.354176 | ✓ | - | - | × |
| 6 | 落 | 络 | 0.997906 | × | × | ✓ | ✓ |
| 7 | 技 | 技 | 0.999677 | ✓ | - | - | × |
| …… | …… | …… | …… | …… | …… | …… | …… |

Table 8 Example of extracting

| Injected | Recovered | Chinese Word Segmentation | Typo at word | Typo index |
|---|---|---|---|---|
| 如今在一个网**落**技术快速发展的信息爆炸的时代，……有时甚至难以被发现。 | 如今在一个网**络**技术快速发展的信息爆炸的时代，……有时甚至难以被发现。 | 如今\|在\|一个\|网**络**\|技术\|快速\|发展\|的\|信息\|爆炸\|的\|时代\|，……有时\|甚至\|难以\|被\|发现\|。 | 6 | 1 |
| 这个**限**象在英文中有出现，……这并不奇怪。 | 这个**现**象在英文中有出现，……这并不奇怪。 | 这个\|**现**象\|在\|英文\|中有\|出现\|，……这\|并\|不\|奇怪\|。 | 2 | 0 |
| 本文计**话**针对新浪微博或者微信……并保证文本的可读性与秘密信息的隐蔽性。 | 本文计**划**针对新浪微博或者微信……并保证文本的可读性与秘密信息的隐蔽性。 | 本文\|计**划**\|针对\|新浪\|微博\|或者\|微信……并\|保证\|文本\|的\|可读性\|与\|秘密\|信息\|的\|隐蔽性\|。 | 3 | 1 |
| 不同于基于格式的隐写算**发**，该算法可以抵抗格式调整、OCR 重新输入等。 | 不同于基于格式的隐写算**法**，该算法可以抵抗格式调整、OCR 重新输入等。 | 不同于\|基于\|格式\|的\|隐写\|算法\|，\|该\|算**法**\|可以\|抵抗\|格式\|调整\|、\|OCR\|重新\|输入\|等\|。 | 10 | 1 |
| 并且由于微博、微信…… | No typo | - | - | - |

### C Other approaches

We studied Liu's method[13] and modified it to apply Chinese text steganography. We hide 0b11010000 00000000 there.

如今在一个**络网**技术快速发展的信息爆炸的时代，网络上每天都有巨量的数据在不断地产生。而网络上的文本往往包含许多拼写错误，这一点在个人用户、自媒体**发等**布的内容中尤其地明显。并且这种错误往往不影响人类对文本意义的理解，有时甚至难以被发现。这**现个**象在**文英**中有出现，在中文中更是如此，所以这一现象似乎是跨语言的。因此，在这样的文本中，人们可以通过明智地输入拼写错误来执行信息隐藏，这并不奇怪。本文计划针对新浪微博或者微信上的文本内容进行中文打字错误的研究，以期望能利用现今的自然语言处理技术研究出基于中文打字错误的信息隐藏算法，该算法能利用精心伪造的打字错误将秘密信息嵌入到中文文本中，并保证文本的可读性与秘密信息的隐蔽性。不同于基于格式的隐写算法，该算法可以抵抗格式调整、OCR重新输入等。并且由于微博、微信包含了多种媒体，该算法可以联合其他算法进行跨媒体甚至跨社交网络的信息隐藏。

It can handle Chinese text as skillful as English text. The extraction also works. And it has a higher embedding rate than our method. However, the injection using swapping the neighboring characters works well in English but seems weird in Chinese. Because it is almost impossible to type a word with reversed characters or flip a pair of characters among neighboring words in Chinese. And that might cause suspicion.

### D Embedding rate

Since the number of embeddable words is not only related to the length of the text and the length of each instance but also related to the semantics of the context, the embedding rate is not stable and is depended on the carrier.

In our experiment, we choose the full stop punctuation mark in Chinese ("。") to divide the text into instances, so the text can be divided into 5 instances and it can embed 5 bits message at most. The carrier consists of 387 characters and is of 9240 bits length in UTF-8, so the embedding rate is 0.54‰. However, another *instance separator* is acceptable. If using non-Chinese character (NCC) as an *instance separator*, the text can be divided in to 24 instances which 24 bits of secret message can be embedded at most and the embedding rate can reach 2.60‰. We also test for some other text gathered from the internet randomly using the full stop punctuation mark in Chinese ("。") and non-Chinese character (NCC) as the *instance separator*, the results is shown below:

Table 9 Results of the embedding rate experiments

| Carrier size (bit) | Embeddable word count (at most) | | Embedding rate (at most) (‰) | |
|---|---|---|---|---|
| | 。 | NCC | 。 | NCC |
| 9240 | 5 | 24 | 0.54 | 2.60 |
| 18800 | 26 | 81 | 1.38 | 4.31 |
| 21392 | 19 | 67 | 0.89 | 3.13 |
| 5008 | 4 | 22 | 0.80 | 4.39 |
| 1366 | 4 | 10 | 2.99 | 7.49 |
| 6464 | 6 | 31 | 0.93 | 4.80 |
| 384 | 1 | 3 | 2.6 | 7.81 |
| 1472 | 1 | 6 | 0.68 | 4.08 |
| 1968 | 3 | 11 | 1.52 | 5.59 |
| 2736 | 3 | 9 | 1.10 | 3.29 |

We can find that the embedding rate is not stable. And sometimes using NCC as *instance separator* might generate too short instance for the model to get enough sematic feature. Therefore, the embedding rate might actually less than those shown in the table.

### E Undetectability

The undetectability is highly related to the length of the instance. Modifying just one word in an instance is usually not easy to be noticed. A homonym can be easily disambiguation when people reading Chinese text, so it might not frustrate understanding. Although typos appear in each sentence seems very strange, we have seen such a blogger whose Weibos contain so many typos that sometimes there are more than one typo in a single sentence. So, in such a situation like chatting or social media like Weibo and Twitter, typos can be regarded as miss-typing or just a careless personal habit. All in all, it might seem strange when embedding too densely, but it is less likely to cause suspicion. However, more Steganalysis should be done to make it more security.

## VI. CONCLUSIONS

In this paper we proposed a text steganography method using typos with the help of BERT. We inject secret message in the Chinese text by creating typos, and then find out these typos to restores the secret message using a natural language processing model, BERT without using the original text. This method can apply in informal situations such as social media like Weibo and WeChat that originally contains many typos. And it can also be applied to digital watermarking to protect information security and protect copyright in Chinese plain text media. Combing text steganography method with other steganography method can design a cross-language or cross-media steganography system.

Although the experiment is successful, there are still some shortcomings like costing huge computing power, limited embedding rate, and slightly obvious typos generated. In the future, we would consider improving the embedding rate by designing and using more powerful NLP models, applying coding scheme and using more type of typos. And the invisibility of typo can be improved by using carefully selected typo candidates. We would study its robustness and other evaluation in the future.

## VII. ACKNOWLEDGMENT

The work is supported by the National Natural Science Foundation of China (NSFC) under Grant no.U1536207, the National Key Research and Development Program of China under no.2016QY08D1600 and the National Key Research and Development Project of China under no.2016YFB0801405.

## REFERENCES

[1] 张楠(Zhang Nan), et al. "基于文本格式的文本信息隐藏方法研究综述."( A Survey of Text Information Hiding Methods Based on Text Format) *信息化研究(Informatization Research)* 03(2017):5-10.

[2] 曹卫兵(Cao Weibing, et al.). "基于文本的信息隐藏技术."( Text-based information hiding technology) *计算机应用研究(Application Research of Computers)* 20.10(2003):39-41.

[3] 甘灿(Gan can), et al. "一种改进的基于同义词替换的中文文本信息隐藏方法."( An improved Chinese text information hiding method based on synonym replacement) *东南大学学报(自然科学版)( Journal of Southeast University(Natural Science Edition))* 37.s1(2007):137-140.

[4] Xiang, Lingyun , et al. "A Word-embedding-based Steganalysis Method for Linguistic Steganography via Synonym-substitution." *IEEE Access* (2018):1-1.

[5] Luo, Yubo , and Y. Huang . "Text Steganography with High Embedding Rate: Using Recurrent Neural Networks to Generate Chinese Classic Poetry." *the 5th ACM Workshop* ACM, 2017.

[6] 耿亮(Geng Liang). *拼音输入法视角的网络错别字研究——以网络新闻与网络聊天错别字为例(Research on Network Typos from the Perspective of Pinyin Input Method——Taking Network News and Network Chat Typo Words as Examples)*. Diss. 上海师范大学(Shanghai Normal University), 2010.

[7] 贺洪花(He Honghua) and 陈铭.(Chen Ming) "网络传播中的错别字探究."(Investigation of typos in online communication) *艺术科技(Art Science and Technology)* 27.2(2014):103-103.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805, 2018.*

[9] Sun, Yu , et al. "ERNIE: Enhanced Representation through Knowledge Integration." (2019).

[10] Weibing, Cao, et al. "Technology of information hiding based on text document." *Application Research of Computers* 20.10 (2003): 39-41.

[11] Shiralishahreza, Mohammad, and M. H. Shiralishahreza. "Text Steganography in SMS." *International Conference on Convergence Information Technology* 2007.

[12] Peter Wayner. Hiding Information in the Order of Letters. http://www.wayner.org/books/discrypt2/wordsteg.html

[13] Liu, Minhao , Y. Guo , and L. Zhou . "Text Steganography Based on Online Chat." *International Conference on Intelligent Information Hiding & Multimedia Signal Processing* IEEE Computer Society, 2009.

[14] Topkara, Mercan , U. Topkara , and M. J. Atallah . "Information hiding through errors: a confusing approach." *Proc Spie* 6505(2007).