Information Hiding for Text by Paraphrasing

Hiroshi Nakagawa
Information Technology Center,
The University of Tokyo
7-3-1, Bunkyo, Hongo
Tokyo, Japan, 113-0033
nakagawa@r.dl.itc.u-tokyo.ac.jp

Tsutomu Matsumoto
Yokohama National University
79-7, Tokiwadai,Hodogaya
Yokohama, Japan,240-8501
tsutomu@mlab.jks.ynu.ac.jp

Ichiro Murase
Mitsubishi Research
Institute,Inc.
Otemach 2-3-6, Chiyoda
Tokyo, Japan, 100-8141
murase@mri.co.jp

Abstract
Digital fingerprinting becomes paid growing attention as a technology resolving copyright problems. Previously, researchers have been only interested in image based digital fingerprinting where secret information is hidden in images, and text have not been the main target of hiding information. In this paper, we propose an information hiding method for text. Our information hiding method is based on paraphrasing that preserves the meaning. By paraphrasing or not, pieces of information are embedded. We experimentally evaluated the proposed method with Japanese manuals and user agreement forms of software, and found the paraphrased text, namely information hidden text, is preserving the meaning of original contents and very natural as language.

Introduction
Recently, as contents distribution via Internet is growing, copy-right protection about digital contents becomes paid attention and an important research issue. The basic technology for this purpose is so called “information hiding.” Nevertheless, text have not been paid attention as the media for information hiding. In this paper, we propose information hiding on text by paraphrasing.

1 Background
1.1 Basic Concepts and Back Ground
For readers who are not necessarily familiar with information security technology, we introduce several basic notions in this research. Information security technologies became used for coping with copy-right protection of digital contents including software. Digital watermarking and fingerprinting are of these technologies. Especially, digital fingerprinting is regarded as a promising method to prohibit or at least prevent illegal distribution of piracy contents because a certain authorized agent could or might identify who distributes the piracy contents with this method. Digital fingerprinting with information hiding technology is depicted as the following figure.

![Information hiding](image)

Figure 1. Information hiding
Software manufacturers sell the content in which the secret data:X that indicates an individual
consumer is embedded by the information hiding technology. Here the contents mean images, music, video, software, or documents which are accompanied with the software, such as manuals or user agreements. After many contents having been sold, once the authorized agent finds and confiscates a piracy content, he can identify who illegally sold the piracy content by extracting X from the confiscated content. Thus, the bad guys could/might get caught. In this figure, we call the original contents as “cover data”. This scheme is generally called “Steganography.” We call the cover data in which the secret data is embedded as “Stego data.” The previous works about information hiding including digital fingerprinting and watermarking mainly use an image as cover data, and even text is treated as an image (Bassil et al 1995, Bassil and O’Gorman 1996 The SNOW 2001). In order to use text as information hiding media, researchers developed the theory to generate natural language sentences with enough complex structure that embed a secret data (Wayner 1992, Wayner 1995, Chapman and Davida 1997). Their aim is to establish a secret communication channel that transmits an embedded data only. That means that the stego data is used as a cloak. Chapman and Davida(2001) recently developed an information hiding technology using text as cover data. They embed secret data by replacing several special words with another words having not exactly the same but similar meaning. For this, they develop by hand a dictionary in which number of pairs of words whose meanings are similar each other. The text their system produces is fairly natural, but still the meaning of sentences are not preserved. Therefore their system can be used only for a secret communication channel and could not be used for digital fingerprinting because in digital fingerprinting, the meaning of cover data should be preserved.

2 Information Hiding by Paraphrasing

2.1 Basic Idea

Our aim is to develop a digital fingerprinting system which could/might prevent illegal distribution of piracy software. Then the system we propose has the following scheme.

Contents (cover data): software and its accompanying documents such as manuals and user agreements. Thus they are called “cover text.”

Stego data: manuals and user agreement. Both of them are text. Thus in this system stego data is called “stego text.”

Secret data: digital data indicating who bought this software.

Information hiding method: paraphrasing with a special dictionary.

Since our system uses manuals or user agreement for information hiding, and these kinds of text carry the important information for general users, the meaning of them should be preserved even if we embed them the secret data. Thus in our information hiding system for text, what we have to develop is a paraphrasing method which preserves the meaning of original text. For this, we need a dictionary for paraphrasing which consists of many pairs of linguistic expressions having the same meaning. In addition, 0 or 1 is assigned to each linguistic expression of a pair respectively. Let me show you how the secret data is embedded into the following example of cover text:

Cover text: “A user can make a copy of an original content only once.”

Dictionary for paraphrasing:
Suppose the dictionary contains the following paraphrasing rules.

- can : 0 ←→ be able to : 1
- copy : 0 ←→ make a copy of : 1
- only : 0 ←→ at most : 1

If the secret data to be embedded is 101, then the stego text is:

“A user is able to copy an original content at most once.”

The underlined parts are hiding the secret data. Obviously, with the same dictionary, we can extract the embedded data from the stego text.

2.2 Suitable Paraphrasing Patterns

The first thing to consider in building a dictionary for paraphrasing is what kinds of linguistic patterns are able to be used to hide secret data without changing the meaning of the original text. Taking into account the condition of preserving meaning, any paraphrasing which requires a precise syntactic analysis such as
scrambling for juxtaposition and passive – active voice conversion is inadequate. We only use (a) synonymous phrase like “can” and “be able to”, “do not” and “don’t”, and (b) notational and/or transcriptional ambiguity like “Girisia” and “Girisha” for “Greece” in Japanese.

The next thing to consider is the direction of paraphrasing. For instance, “can” and “be able to” can be paraphrased each other, in other words, bi-directionally. However “murder” can be paraphrased into “kill” but can not in the other direction. For instance, “We have to kill time till the train comes” can not be paraphrased into “We have to murder time till the train comes” because “kill” has wider usages or meaning than “murder.” This kind of phenomena is inherent in case that a pair of words are not the exactly same, but one has wider meaning than the other, like “kill” and “murder.” As for Japanese, “X suru”, where X is a sahen-noun, means “do X” in English, is extremely widely used. Since this pattern almost always can be paraphrased by “X wo suru” and the other way around, namely bi-directional, it is extremely powerful paraphrasing pattern. However, we have exceptional cases. For instance, “shijou wo tyousa suru (‘investigate the market’)” can not be paraphrased into “shijou wo tyousa wo suru (‘do investigation about the market’)” because of the famous “double-wo constraint” in Japanese. One more thing we have to think about is whether the effect of paraphrasing is confined within the paraphrased part. The “double-wo constraint” is an example.

### 2.4 Description of Paraphrasing Patterns

Considering those aspects described in 2.3, we carefully worked out pairs of linguistic patterns to be paraphrased by hand, ending up with high quality 208 paraphrasing patterns in Japanese. Of course, at first glance, much more paraphrasing patterns might be automatically extracted from corpora with statistical NLP technologies. However we could not do it partly because we have not yet had any corpus for paraphrasing in Japanese. More importantly, to recognize two linguistic patterns having the same meaning in various contexts is still too hard even for today’s NLP technologies. Thus in our implementation at this moment, we use that 208 hand crafted patterns. Apparently many of them can only be applied under specific context despite of our intention to seek context free paraphrasing patterns. Thus, in our paraphrasing pattern dictionary, patterns are of the form of

\[
(expression1, expression2, condition)
\]

where \(expression1\) can be paraphrased into \(expression2\) but not vice versa and \(condition\) is described as the conditions about five words left and right of \(expression1\) (or 2). Namely these ten words are depicted as follows.

\[
\text{LF(5)} \ldots \text{LF(1)} \quad \text{SELF} \quad \text{RG(1)} \ldots \text{RG(5)}
\]

where \(SELF\) is the word(s) to be paraphrased. These conditions are described with POS of some of these ten words and \(SELF\) itself. We can use “NEQ” meaning “not equal” \(&&\) logical connectives like \(\&\) (conjunction), \(||\) (disjunction) as well as =. Thus we can describe more complicated condition like “(LF(3) NEQ verb) and (RG(2)=noun).” If \(condition\) is “nill”, the paraphrasing can always be possible. Now we describe the characteristics of these paraphrasing patterns.

- 56 patterns out of the whole 208 patterns have this type of conditions.
- 24 patterns have the condition about RG(1). Within them, only one pattern include the condition about RG(1) and RG(2).
- 35 patterns have the condition about LF(1).
- 4 patterns have the condition about both RG(1) ands LF(1).
- 21 patterns are paraphrasing a Japanese character string into a string including Chinese characters, or vice versa.

The frequently appearing conditions are the followings.

1. \(SELF = \) the beginning of the sentence
2. \((\text{sitagatte} (‘therefore’), \text{yotte} (‘thus’)), \quad SELF = \) the beginning of the sentence.
3. \(SELF = \) the end of the sentence
4. \((\text{dekimasu} (‘can do’), \quad \text{kanou-desu} (‘be possible’), \quad SELF = \) the end of the sentence,)
The example is:
(6) *tukau* (‘use’), *siyou-suru* (‘do use(sahen-noun)’, LF(1) NEQ noun)
(7) LF(1) = Sahen-noun.
The example is:
(8) *(sikane-masu) (could not do)*, *dekimasan* (‘be not possible’), LF(1) = Sahen-noun)

We also have the paraphrasing patterns whose conditions are combination of LEFT and RIGHT like
(9) *(tukai-kata) (the way to use)*, *siyouhou* (‘usage’), (LF(1) NEQ noun) && (RG(1) NEQ noun)).

The very general paraphrasing patterns are about saken nouns like
(10) (sahen-noun *site* (‘do and’), sahen-noun *si* (‘doing’), nil)
(11) (sahen-noun *surukoto-ga-deki* (‘can do’), sahen-noun *deki* (‘can’), nil)
(12) *(wo) (accusative post postional particle)* sahen-noun *sura* (‘do’), sahen-noun *you* (‘as being done’), RG(1) = noun)
(13) *(wo) (accusative post postional particle)* sahen-noun *suru-koto* (‘do’ – complementizer ), *no* (‘of’) sahen-noun, nil)
(14) (sahen-noun *suru-tame-no* (‘to do’), sahen-noun *no-tame no* (‘for doing’), nil)

The following paraphrasing patterns are based on equivalency of two expressions.
(15) *(koremade) (till now’), *juurai* (‘Previously’), nil)
Since this paraphrasing is applicable on the other way around, we have the following reverse direction rule, too.
(16) *(juurai) (Previously’), koremade (‘till now’), nil)
(17) *(kanren-sita* (‘related to’), *kanren-suru* ( ‘relating to’ ), (RG(1)=noun))

In (17), syntactically, the difference of these two expressions is past (*sita*) and present progressive (*suru*). However, this past form is semantically a perfective and semantically used as an adjective. Thus, when the following word is a noun stated as this rule’s condition, these two can be interchangeable each other.

The following two patterns are examples of Japanese adverbs having almost the same meaning and interchangeable. Of course, they have a subtle difference in meaning or nuance. They differ in specific contexts that can be prohibited by the condition.
(18) *(hotondo* (‘almost’), *taitei* (‘usual’), RG(1)=no(‘of’))||RG(1)=verb)
(19) *(taitei* (‘usual’), *hotondo* (‘almost’), RG(1)=no(‘of’))||RG(1)=verb)

It took almost four months to work out these paraphrasing patterns and verify them linguistically. However, it is desirable to acquire paraphrasing patterns from corpora with machine learning techniques no matter how hard it is, if we want to apply this system into various languages.

3 Experimental Evaluation

3.1 Purposes of Experiments

The text based information hiding system we propose here has to pass at least four tests. The first one is whether stego texts the system generates can not be detected as a stego text by machines with NLP technologies. The technologies we employ at this moment are statistical tests about position and frequency of spaces and special codes like escape-sequence, word N-gram, especially uni-gram, bi-gram, and syntactic structure of sentences. These statistical values are highly dependent on the domain. Thus, the detection system has to be customized to each domain. It ends up with too much cost. Even if it is possible, the paraphrasing rules are handcrafted so carefully that linguistic statistics is very little effected. In sum, the system easily passes the first test. The second test is whether a stego text retains the same meaning of the original text. The third test is whether our digital fingerprinting system is collusion secure, or not. The weakest point of digital fingerprinting is the situation where several people compare their
stego texts to try to find where secret data are hidden. Since, actually, the fourth test is the proper topic of information security, we do not deal with this test in this paper. If you are interested in it, see (Ikeda, et al. 2000). In the rest of this paper, we report the experimental results of the second and third test that are more strongly related to NLP.

3.2 Details of Experimentation

The experimental information hiding system is depicted shown in Figure 2.

![Figure 2. The outline of experimental system](image)

After morphological analysis, the system scan each sentence backward to find out and uniquely identify all words (word sequences) that are possible to be paraphrased by looking up the paraphrasing patterns dictionary described in the previous section. The identified parts that can be paraphrased are tagged. Then, the system embeds a whole document a secret data sequence, that is a random binary sequence in this experiment, using these tags, and generates a stego text. Finally, the stego text is reviewed and evaluated by human experts as described in 3.3.

3.3 Text and Human Reviewers

The texts we used in this experiment are two kinds of Japanese documents. One is software manuals of various lengths. The other is user agreement of software. In each category, we used two 5KB length documents and one 25KB length document. In addition, we used two 50KB length software manuals. All of these documents are real ones. That means that the documents and software products are actually sold in the market.

In reviewing and evaluating, two Japanese major software companies cooperated with us. The human reviewers who review and evaluate the resultant stego texts are a pair of technical writers for manual based stego texts and a pair of legal experts for user agreement based stego texts. Each of them is working at a software development company. In each pair, one has the experience of more than ten years and the other has the experience less than ten years.

3.4 Meaning Preservation

Since our digital fingerprinting system is aimed at information hiding that preserves the meaning of original contents, we, firstly, evaluate how well the stego text generated by our method preserves its original meaning. For this purpose, the human reviewers read both of the original text and its stego text, and try to find the place where the meaning is not preserved. As for paraphrasing patterns dictionary, we use two types. One, which we call “C”, consists of linguistically generic paraphrasing patterns that do not depend on each domain. The other, which we call “T”, consists of domain specific terminologies like “data base” and “DB.” Table 1 shows the dictionary’s size of paraphrasing patterns and kinds of patterns. The dictionary C is worked out by ourselves. The domain specific paraphrasing patterns of dictionary T are worked out by two software companies participating this experimentation. Actually T is combined with C, henceforth C+T, and used in transforming into stego texts.

Table 1. Contents of Paraphrasing Patterns Dictionary

<table>
<thead>
<tr>
<th>Dictionary (a)</th>
<th>(b)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover text in Japanese</td>
<td>morphological analysis by Chasen (Matsumoto, et al. 1997)</td>
<td>extracting paraphrasable parts</td>
</tr>
</tbody>
</table>
In the first column, M and L mean paraphrasing patterns for manuals and user agreement respectively. A and B mean each company cooperating this experiments. For instance, TMA means the dictionary of domain specific terms for manuals made by the company A. (a) and (b) correspond to the categories introduced in 2.2, namely synonymous phrase and Notational and/or transcriptional ambiguity respectively.

We apply these two types of dictionary, say C and C+T to the text described in 3.3. The human reviewers read a stego text and its original text and compare them in terms of their meaning. The evaluated results are shown in the following tables.

**Table 2. Evaluation of stego texts**

<table>
<thead>
<tr>
<th>Cat, Dic</th>
<th>Par-able</th>
<th>Para-ed</th>
<th>Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 manuals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dic = C</td>
<td>4.89</td>
<td>2.53</td>
<td>9%</td>
</tr>
<tr>
<td>2 user agreements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dic = C</td>
<td>8.0</td>
<td>4.1</td>
<td>10%</td>
</tr>
<tr>
<td>3 manuals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dic = C+T</td>
<td>11.7</td>
<td>4.87</td>
<td>42%</td>
</tr>
<tr>
<td>2 user agreements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dic = C+T</td>
<td>20.2</td>
<td>11.6</td>
<td>12%</td>
</tr>
</tbody>
</table>

The first column indicates what kind of cover text is used by applying what type of dictionary. The second column, Para-able, denotes the average number of parts that can be paraphrased per 1KB text. The third column denotes how many parts are actually paraphrased averagely per 1KB by embedding random binary sequence. The fourth column denotes the ratio of paraphrased parts where reviewers detect that the meaning is not preserved, against total paraphrased parts.

Looking at these results, you see that around 5 bits can be embedded into 1KB text only with C dictionary, but if we use C+T dictionary, more than 10 bits can be embedded into 1KB text. It is apparent, but considering the dictionary sizes shown in Table 1, C dictionary that consists of paraphrasing patterns for ordinary sentences is more efficient than T dictionary. As for real application, a 25KB length manual can embed more than 70bits. It is enough to embed user identification data even being encrypted. Unexpectedly, if using C dictionary, user agreements embed almost two times more bits than manuals. This is because user agreements contain less domain specific terminologies, in other words, contain more ordinary words or expressions that can be paraphrased by rules of C dictionary. Another important feature is that human experts detect more paraphrased parts that do not preserve their original meaning if we use C+T dictionary. Considering the fact that T dictionaries are worked out by software manufactures themselves, the meanings of domain specific words are very sensitive in their actual use.

Another thing we are interested in is the difference of quality of the paraphrased parts paraphrased by type of paraphrasing patterns, say (a) and (b). The results are shown in Table 3.

**Table 3. The ratio of detected paraphrased parts**

<table>
<thead>
<tr>
<th>Paraphrasing type</th>
<th># of paraphrase</th>
<th># of detected</th>
<th>Detected ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) synonym</td>
<td>1350</td>
<td>172</td>
<td>12.7%</td>
</tr>
<tr>
<td>(b) ambiguity</td>
<td>240</td>
<td>8</td>
<td>3.3%</td>
</tr>
</tbody>
</table>

As for the paraphrasing patterns type, synonyms with no abbreviation is the most important resource of paraphrasing. But notational and/or transcriptional ambiguity has the best quality in preservation of meaning. Looking more closely, the frequent non meaning preserving paraphrasing are “Kanou( ‘be possible’)” – “Dekiru( ‘can’)” pair and “Baai( ‘case’)” – “Toki(’when’)” pair. In fact, they are very frequent. Thus we need more sophisticated conditions to apply them.
3.5 Naturalness

Since information hiding is aimed at deceiving human attackers, a stego text has to be so natural that any one take it as ordinary text. Then the second experiment is to evaluate how natural stego texts are as Japanese texts. This evaluation is done under the same experimental condition of evaluation described in 3.4 except for embedded secret data is different. In this experiment, reviewers only read a stego text and evaluate how natural a stego text as a Japanese manual or a user agreement. Their evaluation is done on the discrete measure of 1 (not tolerable), 2 (little awkward), 3 (ordinary), 4 (natural) and 5(excellent). All the results are averaged and shown in Table 4.

Table 4. Evaluation as language

<table>
<thead>
<tr>
<th>Cat, Dic</th>
<th>As manuals</th>
<th>As manuals or user agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 manuals Dic = C</td>
<td>3.69</td>
<td>3.56</td>
</tr>
<tr>
<td>2 user agreements Dic = C</td>
<td>4.75</td>
<td>4.25</td>
</tr>
<tr>
<td>2 manuals Dic = C+T</td>
<td>3.25</td>
<td>3.0</td>
</tr>
<tr>
<td>2 user agreements Dic = C+T</td>
<td>3.75</td>
<td>2.75</td>
</tr>
</tbody>
</table>

This time, reviewers indicate the part where paraphrasing deteriorates language quality. That means semantically awkward. The result is shown in Table 5.

Table 5. Number of detected deteriorated paraphrased parts

<table>
<thead>
<tr>
<th>Paraphrasing type</th>
<th># of detected paraphrase</th>
<th># of detected</th>
<th>Detected ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) synonym</td>
<td>1424</td>
<td>322</td>
<td>22.6%</td>
</tr>
<tr>
<td>(b) ambiguity</td>
<td>264</td>
<td>105</td>
<td>11.2%</td>
</tr>
</tbody>
</table>

As seen in Table 4, the quality of stego texts as Japanese are around 3 and 4. That means that the stego texts have ordinary Japanese language quality. So this result is very encouraging. On the other hand, as seen on Table 5, (b) ambiguity shows the best performance in retaining language quality.

Conclusion

We propose the text based information hiding method and experimentally evaluate it. The results are very promising. For actual use of digital fingerprinting, we need the combination of our system and other type of information hiding like image based ones or software source and object code based ones.

References


Maxemchuk, N.F. Electronic Documents Distribution, AT&T Technical Journal, 73-5, 7-.


