Text Watermarking Algorithm Based on Semantic Role Labeling

Jianping Chen, Fangxing Yang, Haiying Ma, Qiuru Lu

School of Computer Science and Technology
Nantong University
Nantong, Jiangsu, China, 226019
chen.jp@ntu.edu.cn

***Abstract*—This paper proposes a new approach for text watermarking that uses the semantic roles to embed watermark information. The technology of natural language processing is applied to find and label the three types of semantic roles A0, A1 and ADV in a text. A watermark message is converted into the hexadecimal Unicode and then compressed with the Huffman encoding to form a digit string that consists of the digits 0, 1 and 2. Let the three types of the semantic roles correspond to the three kinds of the digits one to one. The watermark digits are embedded by mapping each digit into the location of a semantic role of the corresponding type. The algorithm does not make any change to the format and content of a text. It has good features of concealment and robustness and can resist various text format transformations and watermark attacks.**

Keywords—text watermarking; natural language processing; semantic role; location mapping

I. INTRODUCTION

With the rapid development of the Internet and information technology, more and more text works and documents are issued and distributed in digital forms (electronic versions). It provides the convenience for people's life and work, but also brings the problem that a digital text can be easily duplicated and pirated. The protection of the copyrights of digital text products is an important issue in the field of information security. Digital text watermarking is a kind of technology that can be used to protect the copyrights of digital text products [1-2]. It embeds a piece of information (watermark) that represents the author, producer or owner into a text with no or little effect to the text. The watermark can be extracted later if needed to verify the ownership of the text and identify the privacy. In addition, the text watermarking technology can be also used to hide secret information in a text, to identify whether the content of a text is tampered, or to track information in a text, and so on [3-4].

Currently, there are two major categories of the methods for text watermarking. One category is generally called the text format based methods and the other is called the natural language based methods. The text format based methods embed watermarks by slightly changing the formats of a text, such as adjusting the row spacing or word spacing, modifying the font or size, and so on [5-8]. This kind of methods usually have simple algorithms and are easy to implement. But their attack-resistance abilities and robustness are not strong. The embedded watermarks may be destroyed by transformations of text formats. The natural language based methods embed watermarks by making use of the redundancy of natural language semantics. The language attributes such as vocabulary, grammar, or structure are used to encode watermark information [9-11]. Most of the current implementations are through the synonym substitution or syntactic transformations [12-13]. In comparison with the text format based methods, the natural language based methods have better features of concealment and robustness. The transformations of text formats have no effects on the watermarks. Because of the complexity of the human languages especially some languages like the Chinese, however, the synonym substitution or syntactic transformation may cause the problem of the semantic ambiguity or even semantic change [14]. Moreover, it does not apply to the cases in which the content of a text should not be changed any.

In this paper, we propose a new natural language based watermarking technique, which is based on semantic role labeling. The technology of natural language processing is applied to find the semantic roles in a text. Three major types of the semantic roles are selected and used as the carriers for watermark embedding. A watermark is embedded by mapping the watermarking information into the locations of the selected semantic roles. The algorithm does not make any change to both the format and the content of a text. The embedded watermark has strong concealment and robustness and can resist common text format transformations and watermark attacks.

The remainder of the paper is organized as follows. Section 2 describes the basic idea of the proposed algorithm. Section 3 discusses the technique of semantic role labeling. Section 4 is the pre-processing of watermark information including the Huffman encoding. Section 5 are 6 describe the processes of watermark embedding and watermark extraction, respectively. Presented in Section 7 are performance analysis and experiment tests. Finally, conclusions and next work appear in Section 8.

II. BASIC IDEA OF THE ALGORITHM

The semantic role labeling is a commonly used form of semantic analysis. It labels certain words in a sentence as the semantic roles for a given predicate [15]. There are generally six kinds of core semantic roles including A0, A1, to A5. A0 usually represents the doer of the action in a sentence, i.e. the

agent. A1 represents the influence of the action. A2 to A5 have different meanings according to different predicates. The rest are additional semantic roles, such as LOC representing the location and TMP representing the time. The roles whose semantic relationships are relatively indistinct are marked as ADV [16].

The semantic roles essentially reveal the semantic relations between the core verb and the other parts of speech in a sentence. They are the character types abstracted from the sentence according to the semantic relations. These types of roles are fixed and unchangeable. They do not appear explicitly in a text and will not be noticed by the readers. Using these semantic roles as the carriers for watermark embedding has strong concealment and robustness. Meanwhile, there are a considerable amount of semantic roles in a text, which can provide a large space for watermark embedding. Therefore, we propose an approach for text watermarking that makes use of the semantic roles to embed watermark information. We choose certain types of the semantic roles to do the watermark embedding. Find all these types of the semantic roles in a text and label their locations. Watermark information is converted into a digit string and each digit is embedded by mapping it into the location of a semantic role.

Among the above mentioned different types of the semantic roles, A0, A1 and ADV exist most often in a text. We choose these three major types of the semantic roles to design our algorithm.

III. SEMANTIC ROLE LABELING

Semantic role labeling is based on the analysis of word segmentation, part-of-speech tagging and parsing, which belong to the technology of natural language processing. The related techniques are fairly mature now and specialized software systems that can perform the processing are available, such as the Language Technology Platform (LTP) developed by the Harbin Institute of Technology in China [17]. Fig. 1 shows an example of the analysis result (XML format) after word segmentation, part-of-speech tagging and parsing for a sentence using the LTP system.

```
<?xml version="1.0" encoding="utf-8" ?>
  <xml4nlp>
    <note sent="y" word="y" pos="y" ne="y" wsd="y" srl="y"/>
     <doc>
       <para id="0">
         <sent id="0" cont="We are all Chinese people">
          <word id="0" cont="We" pos="r" ne="O" parent="2" relate="SBV" >
          <word id="1" cont="all" pos="d" ne="O" parent="2" relate="ADV" >
          <word id="2" cont="are" pos="v" ne="O" parent="-1" relate="HED">
                  <arg id="0" type="A0" beg="0" end="0"/>
                  <arg id="1" type="ADV" beg="1" end="1"/>
                  <arg id="2" type="A1" beg="3" end="4"/>
     </word>
     <word id="3" cont="Chinese" pos="ns" ne="S-Ns" parent="4" relate="ATT" >
     <word id="4" cont="people" pos="n" ne="O" parent="2" relate="VOB" >
       </sent>
     </para>
    </doc>
</xml4nlp>
```

Fig. 1. Word segmentation, tagging and parsing

Syntactic relations, semantic roles and other information are indicated with a number of node labels, including *doc*, *para*, *sent*, *word*, and *arg* etc. *doc* is a document node containing the content of a text. *para* is a paragraph node, including *id* attribute, where *id* is the sequence number of the paragraph. *sent* is a sentence node including *id* and *cont* attributes, where *id* is the sequence number of the sentence and *cont* is the content of the sentence. *word* is a word node having the attributes of *id* and *cont*, where *id* is the sequence number of the word in the sentence and *cont* is the content of the word. *pos*, *ne*, *parent*, and *relate* are optional nodes, where *parent* is the *id* number of the word's parent node and *relate* is the corresponding relationship. *arg* is an information node of the semantic role. The predicate in a sentence has a number of such nodes. Each of them has the attributes of *id*, *type* and so on, where *id* is the sequence number of the node and *type* represents the type of the semantic role.

The process to label the semantic roles in a text is as follows. Traverse through the XML file such as the one shown in Fig.1 to find the *word* node whose *relate* is 'HED' (HED represents the core relationship). Then traverse through the *type* attributes of the *arg* node under this *word* node and check the type to see if it is the type we need (i.e. the type A0, A1 or ADV). If it is, take out the *id* of this *arg* node and the *id* of the *sent* node of this *word* node's *parent* node and the *id* of the *para* node of the *sent* node's *parent* node. These three *id* values are used to express the location of the semantic role. We use the first letter of the node name plus the corresponding *id* value to formulate the expression as follows:

$$L = p + para.id + s + sent.id + a + \arg.id$$

For instance, if the *id* of the *arg* node for a semantic role is 2, the *id* of the related *sent* node is 3 and the *id* of the related *para* node is 9, its location is expressed as:

$$L = p9s3a2$$

Take the text shown in Fig. 1 as an example. Since it has only one paragraph and one sentence, the location of the role A0 is expressed as *p*0*s*0*a*0; the location of A1 is expressed as *p*0*s*0*a*2; the location of ADV is expressed as *p*0*s*0*a*1.

Repeat the process to find all the three types of the semantic roles in a text and express their locations using the above formula. Put the location data into three sets respectively according to their types. That is, set A0 contains the locations of the semantic roles of the type A0; set A1 contains the locations of the semantic roles of the type A1; and set ADV contains the locations of the semantic roles of the type ADV.

IV. PRE-PROCESSING OF WATERMARK INFORMATION

Usually a watermark message is converted into binary bits to embed using the ASIC code. This just applies to the case that the watermark message is composed of the English characters. In order to embed watermark messages in different languages including the English, Chinese and the others, we consider converting a watermark message into the Unicode, in which each character is expressed as four hexadecimal numbers. For example, the words "Nantong University" in the Chinese consist of four Chinese characters and its Unicode is "\u5357\u901a\u5927\u5b66".

If we directly embed the hexadecimal Unicode numbers, we would need 16 different types of the semantic roles. This is not convenient for implementation. So we use the method of the Huffman encoding to compress the number of the code elements of the hexadecimal Unicode. As discussed previously, three types of the semantic roles A0, A1 and ADV are chosen for watermark embedding. To match with the number of the types of the selected semantic roles, we use the ternary Huffman encoding to reduce the number of the code elements from 16 to 3. As a result, the hexadecimal Unicode is converted into the Huffman code that consists of the digits of 0, 1 and 2. For example, the above Unicode representing the Chinese words "Nantong University" is converted into the Huffman code as follows:

"20212211122022021011121201212021220112202202122001010".

V. WATERMARK EMBEDDING

A watermark is embedded by mapping the watermark information into the locations of the selected semantic roles. The locations of the three types of the semantic roles in the text that we want to watermark are found and labeled using the method described in Section 3. Their location data are stored in three sets A0, A1 and ADV. A watermark message is converted into the form of the Huffman code that is a digit string of 0, 1 and 2, which is discussed in Section 4. Let the three types of the semantic roles correspond to the three kinds of the digits one to one. Namely, A0 corresponds to 0; A1 corresponds to 1; ADV corresponds to 2. Map every digit in the digit string representing the watermark into a location of one semantic role in one of the sets. If the digit is 0, map it with a location in set A0. If the digit is 1, map it with a location in set A1. If the digit is 2, map it with a location in set ADV. Take out the location data mapped by every digit from the corresponding set and put it into a file. Repeat the process to map all the digits in the watermark digit string into the locations in set A0, A1, or ADV. This completes the embedding of the watermark. A file containing the location data of the selected semantic roles is produced. Fig. 2 shows the process of the watermark embedding.

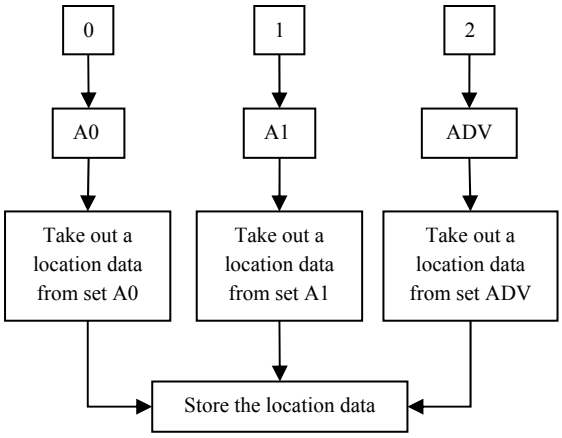

Fig. 2. Process of watermark embedding

VI. WATERMARK EXTRACTION

The extraction of a watermark is based on the location data of the semantic roles in the file produced in the watermark embedding. We use the location data to find the semantic roles at these locations. According to the relationship between the three types of the semantic roles (A0, A1, ADV) and the three kinds of the digits (0, 1, 2), make a reverse mapping to restore the digit string that represents the watermark message. The detailed process is as follows.

Apply the processing of the word segmentation, part-of-speech tagging and parsing to the text that we want to extract the watermark, and get an analysis result (XML file) that is similar to the one shown in Fig. 1. Open the file produced in the watermark embedding that contains the location data of the semantic roles. Successively take out a location datum from the file, which consists of the *id* values i.e. the sequence numbers of the paragraph, the sentence and the word. According to these sequence numbers, find the location in the XML file. Check the type of the semantic role at this location and make a reverse mapping according to the type. That is, if the type is A0, generate a digit "0"; if the type is A1, generate a digit "1"; if the type is ADV, generate a digit "2". After all the locations in the file are reversely mapped, a digit string consisting of 0, 1 and 2 which represents the watermark message is obtained. The process is shown in Fig 3.

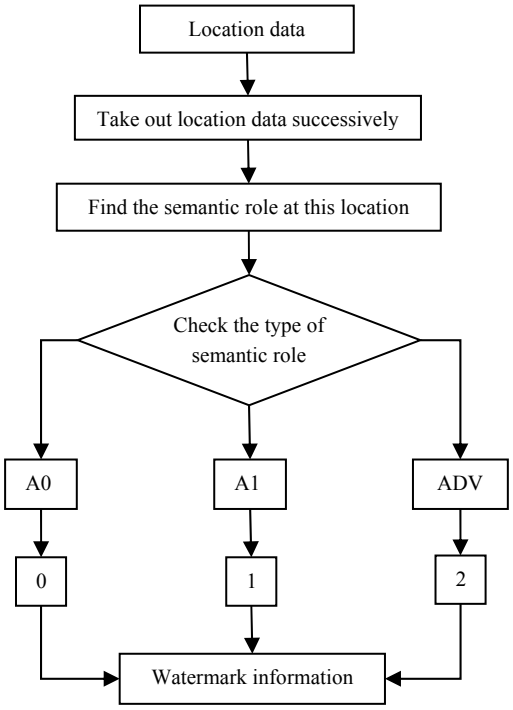

Fig. 3. Process of watermark extraction

Finally, the Huffman decoding is performed on the digit string to convert it into the Unicode. Then the Unicode is converted back into the characters and the original watermark message is hence obtained.

VII. Performance Analysis and Expriment tests

The above presented algorithm does not make any change to both the format and the content of the text watermarked. It has not any effect to the use of the original text. There are no traces with the embedding of a watermark. The watermark will not be perceived and discovered, and has strong concealment. In addition, the algorithm has very good features of attack resistance and robustness.

The algorithm is built on the basis of the syntactic analysis and semantic roles. It has nothing to do with any formats of a text. The transformations and attacks at format level (such as fonts, sizes, colors, spacing, punctuation etc.) will not affect the correct extraction of the watermark. For the transformations and attacks at the level of words, the synonym substitution does not change the semantic role and hence has no effect to the watermark. The operation of inserting or deleting some words may affect the extraction of the watermark depending on the positions of the operation. If an insertion or deletion operation breaks the watermark which means that the semantic role is changed, it will also certainly change the meaning of the sentence. At the level of sentences, since the selected semantic roles are the core roles in a sentence, the attack is not easy to implement. If an attack succeeds, it will destroy the core structure of the sentences and seriously change the meaning of the text.

Based on the presented algorithm, an experiment system is designed. We use the system to embed and extract a watermark message in a text and carry out experimental tests to the algorithm and watermark. We made various changes to the formats of the text after a watermark is embedded, including changing row spacing, word spacing, font size, font color, etc. It is shown that the watermark can be extracted correctly without being influenced. We selected some words and phrases including the ones that relate to the three types of the semantic roles to do synonym substitutions. The results of the experiments show that the watermark can still be extracted correctly. We also tested adding or deleting some words and phrases. If the removed part is related to the three types of the semantic roles or the added part changes any of the three types of semantic roles, the watermark extraction could be affected and some part of the watermark message might not be extracted correctly. In these cases, however, the meaning of the related sentences is changed seriously. This implies that the normal use of the text will be affected and the attack lost its significance.

VIII. Conclusions and Next Work

A novel text watermarking algorithm is proposed in this paper, which uses the semantic roles in a text to embed watermark information. The embedding and extraction of a watermark is achieved by mapping the watermark information into the locations of selected semantic roles. The algorithm does not make any change to both the format and the content of a text, and hence has no side effects on the original text. The watermark embedded cannot be perceived and discovered. It has strong concealment and robustness and can resist common text format transformations and watermark attacks. Based on the algorithm, we are developing a software system that can be used to embed and extract watermark information in practical digital texts. Using the system we are going to do further tests and experiments.

Acknowledgement

This research work was financially supported by the National Natural Science Foundation (No. 61402244), the Jiangsu Provincial Natural Science Foundation (BK2015272) and the Nantong Municipal Application Research Foundation (No. GY2015012) of China.

References

[1] Z. Jalil and A. M. Mirza, "A review of digital watermarking techniques for text documents," Proceedings of the 2009 International Conference on Information and Multimedia Technology, IEEE, 2009, pp. 230-234.

[2] M. Liu, B. Sun, and Y. Guo, "Survey of text watermarking," Journal of Southeast University (Natural Science Edition), vol. 37(Z1), 2007, pp. 225-230.

[3] W. Qi, X. Li, and B. Yang, "Text watermarking algorithm for tracking," Journal of Communication, vol. 29(10), 2008, pp. 183-189.

[4] Q. Chen and X. Xing, "A digital watermarking technology used for covert communication in electric power system," Electric Power Science and Engineering, vol. 30(2), 2014, pp. 12-15.

[5] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Electronic marking and identification techniques to discourage document copying," IEEE Journal on Selected Areas in Communications, vol. 13(8), 1995, pp. 1495-1504.

[6] J. T. Brassil, S. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," Proceedings of the IEEE, 1999, pp. 1181-1196.

[7] F. Cai, Y. Liu, and X. Yin, "Text watermarking for word documents," Computer Science, vol. 39(11A), 2012, pp. 39-40.

[8] X. Liang, Z. Yuan, and M. Huang, "Text digital watermarking algorithm based on line spacing code," Information Technology, vol. 32(3), 2008, pp. 38-41.

[9] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," IBM Systems Journal, vol. 2420(3), 1995, pp. 313-336.

[10] M. Atallah, C. McDonough, and S. Nirenburg, "Natural language processing for information assurance and security: an overview and implementations," Proceedings of the 9th ACM/SIGSAC New Security Paradigms Workshop, Ireland, 2000, pp. 51-65.

[11] M. L. Mali, N. N. Patil, and J. B. Patil, "Implementation of text watermarking technique using natural language watermarks," Proceedings of the 2013 International Conference on Communication Systems and Network Technologies, IEEE, 2013, pp. 482-486.

[12] M. J. Atallah, V. Raskin, and M. Crogan, "Natural language watermarking: design, analysis, and a proof of concept implementation," Lecture Notes in Computer Science, 2001, pp. 185-200.

[13] C. Gan, X. Sun, and Y. Liu, "An improved steganographic algorithm based on synonymy substitution for Chinese text," Journal of Southeast University, vol. 37(Z1), 2007, pp. 137-140.

[14] Y. Zhang, T. Liu, and Y. Chen, "Natural language watermarking," Journal of Chinese Information Processing, vol. 19(1), 2005, pp. 56-62.

[15] X. Wang, W. Sun, and Z. Sui, "Research of Chinese semantic role based on shallow syntactic analysis," Journal of Chinese Information Processing, vol. 25(1), 2011, pp. 116-122.

[16] J. Li, G. Zhou, and Q. Zhu, "Semantic role labeling for Chinese part-of-speech predicate," Journal of Software, vol. 22(8), 2011, pp. 1725-1737.

[17] W. Che, Z. Li, and T. Liu, "LTP: a Chinese language technology platform," Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 2010, pp. 13-16.