

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Natural Language Watermarking Using Chinese Syntactic Transformations

Hong Wang, Xingming Sun, Yuling Liu and Yongping Liu
School of Computer and Communication, Hunan University, Changsha, China

Abstract: In this study, a novel natural language watermarking scheme of Chinese texts based on syntactic transformations is proposed. In the proposed scheme, the so-called sentence weight defined in the study is used to carry the watermark bit. During the embedding process, the transformable sentences are first grouped and then their sequence is permuted to enhance the imperceptibility of watermark bit hiding positions. To improve watermarking reliability and capacity, error correcting coding is also applied.

Key words: Text watermarking, natural language watermarking, syntactic transformation, text protection

INTRODUCTION

The past two decades have seen extensive research on watermarking techniques for multimedia documents (Cox *et al.*, 1997; Busch *et al.*, 1999; Xu *et al.*, 1999). However, digital watermarking methods for text documents are very limited and these approaches mainly focus on watermarking by modifying the appearance of text elements, such as lines, words, or characters (Brassil *et al.*, 1999). Watermarks inserted by the above methods cannot resist attacks such as scanning and OCR (optical character recognition) or reformatting. Thus, Natural Language Processing (NLP) tools have been introduced to study text watermarking. Previous studies on Natural Language (NL) watermarking mainly focused on the synonym substitution approach (Atallah *et al.*, 2000; Chiang *et al.*, 2003; Yang *et al.*, 2007).

The first syntax-based NL watermarking scheme was described in Atallah *et al.* (2001) which employed transformations of deep syntactic structures. Implementation details about this scheme could be found by Topkara *et al.* (2006a). To overcome the word precision problem caused by synonym substitution, Topkara *et al.* (2006b) proposed a new syntactic watermarking sentence selection criteria based on feature functions. Different from previous syntactic approaches, the watermark bits in Gupta *et al.* (2006) were embedded into the LSB of the word numbers in the watermarked sentence. To make a try of watermarking natural language text on the semantic level, watermarking based on a very special type of sentences with presuppositions was described in Vybornova and Macq (2007). A detailed and comprehensive bibliography of natural language watermarking can be found in Bergmair (2007).

Currently, the study on NL watermarking mainly focuses on English. As far as we know, there has been

little Chinese-specific syntactic watermarking research reported to date. In this study, a novel Chinese syntactic watermarking scheme based on dependency grammar and syntactic transformations is proposed.

In this study, we aim to improve the current situations in NL watermarking on the sentence level in the following ways:

- Provides a Chinese-specific syntactic watermarking approach. Present proposed scheme relies on the fact that during a syntactic transformation, the same words in the original sentence will usually change their positions in the transformed one. If different word positions are assigned with different weights, the transformable sentence (i.e., sentences that would retain their semantic meanings during syntactic transformations) weight will vary accordingly. This variation in sentence weights is used to carry the watermark bit.
- Improve the watermarking capacity reduction problem caused by the usage of marker sentences (Atallah *et al.*, 2001; Atallah *et al.*, 2002). In this scheme, sequence permutation will be first applied to all the transformable sentence indexes before watermark embedding. Thus all transformable sentences would be available during embedding.
- Increase the successful embedding probability. During the embedding process, transformation(s) of a single sentence might not lead to a desired bit change, for example, the number of words in the sentence will not change during many syntactic transformations. This fact was ignored in Gupta *et al.* (2006). In this study, we try to fix this problem with the help of embedding by a group of sentences and error correcting coding.

SENTENCE TRANSFORMATION

Syntactic transformations tools: As we all know, sentence syntactic transformations involve generations of new sentences that have the same meaning or near meaning as the source sentences. As in English and other languages, there are some linguistic constructions in Chinese that can be transformed into each other whilst the semantic meanings of the sentences are preserved. Typical transformation tools include topicalization, Verbal Phrase (VP)-preposing, cleft sentences, etc. of course, there also exist a number of Chinese-specific transformations, for instance, BA (把) construction ⇔ BEI (被) construction, reflexive doubling, etc.

These transformations typically involve three major words operations within a sentence, namely, movement, addition and deletion as exemplified in the Table 1.

Dependency analysis: A dependency relationship is an asymmetric binary relationship between a word called head (or governor, parent) and another word called modifier (or dependent, daughter) (Hays, 1964). In dependency theory, sentence structures are represented as a set of dependency relationships. The dependency relationships from a tree connect all the words in a sentence and this tree is also called a dependency (parse) tree. As this kind of structures is relatively theory independent, easier to understand and can be encoded of argument and valence structure directly, it is widely used in many applications of computational linguistics. In this study, the generation component is also based on the dependency theory.

Transformation rules: Currently, based on the study of Li (1988), are manually collected more than 100 transformation rules. These rules are based on dependency relations and are written in a way that can be recognized by computer programs.

A typical transformation rule is as follows:

101 #R<HEAD>+R<ADV>+R <POB>&P<nd>?R<SBV>
= C<SBV> |C[<ADV>:<POB>]

Symbol = in the above lines divides the rule into two parts: the left part denotes the partial structure to be matched in the source sentence dependency tree; the right part represents the words for the above partial structure in the newly transformed sentence.

On the left part, 101 is a rule id; R<> denotes some dependency relationship (for example, HEAD means the headword, ADV means adverbial, POB means preposition-object and SBV means subject-verb); symbol + means that the very next word after previous word(s) will be evaluated; symbol & implies conditions on both sides of & should hold simultaneously; P<> denotes the part of speech of the word in question (in the above rule, a directional noun); ? means that some conditions (in this rule, an SBV relationship) should be met after previous words.

On the right part, C<SBV> represents the SBV word and the word(s) on its subtree node(s) in the dependency tree and all these words are reordered as in the source sentence, C[<ADV>:<POB>] represents all the words between the ADV word id and POB word id and Symbol "|" means the concatenation of the contents of C<SBV> and C[<ADV>:<POB>].

This rule says that if a relation sequence like HEAD+ADV+POB is found in the dependency tree, if the word with the POB relation is a directional noun and if there exists any SBV word after these words, the transformation will swap the words between the ADV word and the POB word with the SBV word and its subtree node words, while the rest words in the source sentence will be left untouched.

All the rules are stored in database files and indexed by dependency relations or keywords. Besides these handcrafted rules, we are also developing a component for automatic extraction of transformation rules based on paraphrase pairs.

Table 1: Typical Chinese syntactic transformations

Types	Transformation	Before transformation	After transformation
Movement	BA-BEI construction	我把这本书看完了。I have read this book	这本书被我看完了。This book has been read by me
	ZAI (在) construction	在黑板上他写了几个字。On the blackboard he wrote a few words	他在黑板上写了几个字。He wrote a few words on the blackboard
	Topicalization VP-preposing	我很喜欢小周。I like Xiao Zhou very much 王刚绝对不会背叛朋友。Wang Gang would never turn against friends	小周, 我很喜欢。Xiao Zhou, I like her very much 背叛朋友, 王刚绝对不会。Turning against friends, Wang Gang would never do it
Addition	Flip-flop sentences	六个人睡两张床。Six people sleep in two beds	两张床睡六个人。Two beds sleep six people
	Modal auxiliaries	这件事他自己能解决。He himself can solve this matter	这件事他能自己解决。He can solve this matter himself
Deletion	Reflexive doubling	老婆老扯自己后腿。Lao Jiang always holds himself back	老婆自己老扯自己后腿。Lao Jiang always holds himself back
	Negation of a rhetorical question	小张怎么可能考试作弊呢? How could Xiao Zhang cheat on tests?	小张不可能考试作弊。Xiao Zhang couldn't cheat on tests

Meaning-preserving sentences generation: As it is known, Natural Language Generation (NLG) in NLP is no easy task. Generation of grammatically correct and semantically equivalent sentences pairs (also called paraphrasing) has become a very important part in syntax-based NL watermarking. In this study, we use a similar approach as described in Takahashi *et al.* (2001) to generate meaning persevering (or near-persevering) sentences, but mainly on the sentence level.

For a given sentence, a dependency analysis (for Chinese, word segmentation is also required) is applied and all word dependency relations of this single sentence will be obtained. A search in the transformation rules database for the dependency relations or the headword is carried out. If matched, the sentence will be marked as transformable and transformation(s) will be applied according to the transformation rule(s) when needed; otherwise, the sentence will be marked as untransformable.

WATERMARKING ALGORITHM

Selection of watermarking sentences: In natural language watermarking, the sentences to be watermarked must be selected in a pseudorandom fashion as a function of a secret key. Otherwise, an adversary knowing the algorithm will be able to extract the watermarking bits easily. There are a number of ways to select a sentence from the cover text for watermarking, for example, relying on the dependency trees (Atallah *et al.*, 2001), word features (Topkara *et al.*, 2006b) or sentence sequence permutation (Gupta *et al.*, 2006).

Here, it is use the algorithm introduced in Aura (1996) to generate the desired sentence sequence is used:

- Let $S[n]$ be an array of size n where the indexes of transformable sentences are stored in the order as they appear in the original text. Let k be a secret prime key. $P_k^n(y)$ means the y -th element in the pseudorandom permutation of the set $\{0, 1, 2, \dots, n-1\}$ controlled by k .
- Let H be a hash function such as SHA-1. First construct a pseudorandom function of argument x that depends on the parameter k : $f_k(x) = \text{SHA-1}(kox)$, where kox is the concatenation of the bit strings k and x .
- Let i be a bit string of length $2l = \lceil \log(n) \rceil$ and these $2l$ bits are required to represent the index of any given sentence in the set $\{0, 1, 2, \dots, n-1\}$. Divide i into two parts, L and R , of length l and the key k into four parts k_1, k_2, k_3 and k_4 .
- Compute:

```

L = L ⊕ fk1(R)
R = R ⊕ fk2(L)
L = L ⊕ fk3(R)
R = R ⊕ fk4(L)
return L o R
    
```

The returned value is $P_k^{2^{2l}}(\text{val}(i))$, where $\text{val}(i)$ is a function that converts the bit string i into its corresponding decimal value. Let $\text{val}(i)$ take value from the set $\{0, 1, 2^{2l}-1\}$ in the ascending order respectively, then we will obtain a pseudorandom permutation of $\{0, 1, 2^{2l}-1\}$.

- Compute the values $P_k^{2^{2l}}(0), P_k^{2^{2l}}(1), P_k^{2^{2l}}(2), \dots, \Lambda$, discarding any numbers above $n-1$ from the sequence and take the remaining values for $P_k^n(0), P_k^n(1), P_k^n(2), \dots, \Lambda$. Thus, the new sentence indexes are given by $\{P_k^n(0), P_k^n(1), P_k^n(2), \dots, P_k^n(n-1)\}$.
- The resulting permuted sentence sequence is as follows:

$$S[P_k^n(0)], S[P_k^n(1)], L, S[P_k^n(n-1)]$$

Watermarking embedding: Before proceeding to watermarking embedding, we describe how the watermark bits will be carried in the cover text.

First, we introduce some notations that will be used here:

- $d(s)$: The number of words in a given sentence s
- $m(p)$: The number of morphemes in the word at a given position p
- s_j : A sentence chosen for transformation
- w^p : The weight at position p
- wm_i : The i -th watermark bit and hence the whole watermark to be embedded is $wm_0, wm_1, \dots, wm_{i-1}, \dots$

Sentence weight W_j for s_j is defined by the following formula:

$$W_j = \sum_{p=0}^{d(s_j)} w^p m(p)$$

For simplicity, the following weights in Table 2 will be used in this study.

Thus

$$W_j = \sum_{p=0}^{d(s_j)} 2^p m(p)$$

Table 2: The word weight

Position (p)	p	...	2	1	0
Weight (w ^p)	2 ^p	...	2 ²	2 ¹	2 ⁰

Table 3: Word length distribution in skeleton dependency corpus of IR-LAB at HIT

Morpheme(s) in a word	1	2	3	4	>4
No. of words	89519	94149	11500	3899	1266
Percentage (%)	44.685099	46.996251	5.740442	1.946259	0.631948

According to the statistics, roughly 13% of Chinese words have one morpheme, more than 70% of them have two morphemes and less than 10% of them have three or more morphemes (<http://www.hongen.com/pc/newer/ime/znabc/imejsab1.htm>).

In order to obtain first-hand statistics, here a detailed analysis of the Skeleton Dependency Corpus by HIT IR-Lab (<http://ir.hit.edu.cn/demo/ltp/>) is given which consists of 50,000 sentences and 858,506 words.

From Table 3, it can be seen that words in a single sentence are not of equal length. Most words have one or two morphemes, a few words have three morphemes and less than 3% of the words have 4 or more morphemes. During a typical syntactic transformation, the word positions will always change and thus the number of morphemes in the same position will also change. If we assign different word positions with different weights, the weight value for the whole sentence will change during the transformation. We can utilize the weight value to carry the watermarking bit. Usually, a binary valued function may be first applied for the weight value of the original sentence and thus a new value (either 1 or 0) for the sentence will be obtained. In order to enhance watermarking robustness, the probability that the new value is 1 or 0 should be evenly distributed.

In this study, a special quadratic residue function is adopted to meet this goal. An integer q is called a quadratic residue modulo z if it is congruent to a perfect square (mod z); i.e., if there exists an integer y such that: $y^2 = q \pmod{z}$; Otherwise, q is called a quadratic nonresidue (mod z). It is noticed that modulo a prime number there are an equal number of residues and nonresidues. And thus based on Busch (2007), the quadratic residue function $qr(\text{key}, W)$ is given in the following:

$$qr(\text{key}, W) = \begin{cases} 1, & \text{if } w \text{ is a quadratic residue (mod key)} \\ 0, & \text{otherwise} \end{cases}$$

where, key is a prime number and W is the weight value.

For example, for the following sentence:

S₁: 这件事他自己能解决. (He himself can solve this matter).

$m(p)$ at position p are listed as follows: 1 1 1 1 2 1 2 and the sentence weight for S_1 :

$$W_1 = \sum_{p=0}^6 2^p m(p) = 132$$

The quadratic residue function $qr(\text{key}, W_1)$ is applied to W_1 to obtain a binary value, where key is assigned with a large prime number, for example 256203221.

And it can be easily computed that $qr(\text{key}, W_1) = 1$.

If the watermarking bit is 1, no change is needed and otherwise, a transformation as reflexive movement is taken and consequently, we get the new transformed sentence:

S₂: 这件事他能自己解决. (He can solve this matter himself).

For S_2 , the corresponding value after transformation is: 1 1 1 1 1 2 2 and the weight value for S_2 :

$$W_2 = \sum_{p=0}^6 2^p m(p) = 130$$

And now the new value $qr(\text{key}, W_2) = 0$.

In this way, a watermarking bit is embedded into the sentence weight. However, as it often occurs, $qr(\text{key}, W_2)$ might not always produce an expected value even if all the syntactic transformation tools for these sentences are exhausted. In the above example, if $qr(\text{key}, W_2)$ is still 1 after all possible transformations, what shall we do?

This problem has been ignored by Gupta *et al.* (2006). Atallah *et al.* (2001) proposed that some new sentences could be inserted into the cover text to obtain the required tree structures, but what if that fails, insert another new sentence? Moreover, inserting new sentences would definitely change the meaning of the whole text.

To deal with this problem, we adapt the embedding approach proposed by Grothoff *et al.* (2005) to meet our goal. We can first partition all the transformable sentences $S[n]$ into groups of size g each. Thus here are $[n/g]$ groups of unused transformable sentences, namely, $G_0, G_1, \dots, G_{[n/g]-1}$. Taken group G_0 as an example, it may be used the group weight $GW_0 = \sum_{j=0}^g W_j$ to carry the

watermarking bit. As $qr()$ is a binary valued function, $qr(\text{key}, GW_0)$ changes with probability 0.5 when w_j is modified. For any single transformable sentence, at least 2 changes (including the original sentence) are available due to syntactic transformations. For the smallest value, $g = 2$ (if $g = 1$, the scheme will degrade to the case of a single sentence transformation), here will be at least 4 possible changes in $qr(\text{key}, GW_0)$. Thus, the probability of a successful watermark bit embedding for G_0 is at least $1 - \frac{1}{2^4} = 0.9375$. If $g = 3$, the successful probability is at least 0.99609375. For any g , the probability that watermarking fails is less than $\frac{1}{2^{2g}}$.

The second approach is to deal with the above problem is to apply some error correcting codes during watermarking to correct the bits that cannot be embedded accurately. For the ease of discussion, Hamming coding is applied in this study to make a demonstration. The original watermark can be divided and encoded into small packets of 15 bits each (11 watermark bits plus 4 check bits). When $g = 2$, an embedding is successful with probability 0.9375, i.e., 14 out of 15 embeddings are successful. For Hamming codes, when the code word has 15 bits, 4 check bits can successfully correct one wrong bit. Hereby, for any 15-bit-long watermark packet (11 watermarking bits), the information can be correctly embedded and extracted with probability 1.

Final scheme uses a combined approach of these two approaches to make a trade-off, where $g = 2$ and Hamming coding scheme are applied.

The watermarking algorithm is given in the below:

- (1) A dependency analysis is carried for all the sentences in the whole text, during which transformation applicability is determined. Sentences that can be transformed are classified as transformable and unused. Denote these sentence indexes by $S[n]$, where n is the number of all transformable sentences.
- (2) Divide $S[n]$ into groups of size g each. Randomly select a group of sentences G_j from $G[n/g]$ according to our permutation-based selection algorithm.
- (3) Calculate the weight value GW_j for G_j and compare its quadratic residue function value with the watermarking bit w_m . If equals, no change is taken; otherwise, transform one of the sentences in G_j using available syntactic tools until $(qr(key, GW_j))$ equals w_m . If all transformations fail, no change is made to G_j .
- (4) Delete the above watermarking bit w_m from the watermark and mark G_j as used.
- (5) Repeat steps (2-4) until the whole watermark information is embedded.

Watermarking extraction: The watermark bits are extracted in the same permuted sequence used while embedding. The extraction process is similar to embedding process except that in this case we just

calculate quadratic residue function value with the prime key and the sentence weights as the input arguments. Finally, Hamming coding is applied to all watermark packets lest there might be some incorrect bits due to syntactic transformations or attacks from the adversaries.

EXPERIMENTS AND RESULTS

Syntactic transformation for Chinese involves several NLP tools, for example, a word segmenter, a syntactic parser and a generator or a paraphrasing tool. In our experiment system, the LTP NLP platform for processing Chinese documents is used (<http://ir.hit.edu.cn/demo/ltp/>).

Currently, to the best of the authors' knowledge, there is no Chinese generation or paraphrasing system available and to meet our goal, we develop a transformation component based on the Chinese dependency grammar and the paraphrasing algorithm described by Takahashi *et al.* (2001). However, as the project aims at syntactic level watermarking, this component deals with sentence level transformations only and does not support paraphrasing on the word level. In addition, for the ease of manipulating xml files produced by the LTP NLP platform, an open source XML package called TinyXML (<http://sourceforge.net/projects/tinyxml>) is employed.

Our proto-type system is called NLWM, which is implemented in Visual C++ 6.0 on the Windows XP platform. The present study gives implementation of two typical linguistic constructions for Chinese, namely BA BEI construction and ZAI construction with generation component.

The system is tested on 16628 sentences in the PH Corpus (<ftp://ftp.cogsci.ed.ac.uk/pub/Chinese>, only part of the corpus, 1.341 Mb raw texts, 388,668 words). And to avoid segmenting or parse errors caused by natural language preprocessing tools, the NLWM system has used the Dependency Relation Corpus (<http://ir.hit.edu.cn/demo/ltp/>, 10,000 sentences with Part-of-Speech (POS) and dependency relations tagged, 200,333 words) provided by the IR-LAB, HIT for a second test.

From Table 4, when $g = 1$, some watermark length bits (in the header of the whole watermark) are not correctly embedded even if all the available transformations are

Table 4: Transformable sentences and watermarking capacity

	Hamming coding	PH Corpus (part)			Dependency Corpus, HIT		
		Transformable sentences	Bits embedded	Bits extracted	Transformable sentences	Bits embedded	Bits extracted
$g = 1$	NO	1812	800	ERROR	1235	585	ERROR
$g = 2$	YES	1812	660	660	1235	451	451
$g = 3$	NO	1812	604	603	1235	411	411

taken and thus the watermark length information is lost. Consequently, the whole watermark information could not be embedded properly. In this case, an ERROR occurs. When $g = 2$, there are 906 groups in the PH Corpus and 617 groups in the Dependency Corpus. Using Hamming coding, we can embed 60 watermark packets (15 bits each) in the former corpus and 41 packets in the latter one. When $g = 3$, there are 604 groups in the PH Corpus and 411 groups in the Dependency Corpus. Without error correcting code coding, the successful embedding rate is more than 0.99609375. Thus, no error correcting code is needed to improve the embedding reliability.

From Table 4, it can be shown that when $g = 2$ and with hamming codes, we have more watermarking capacity than the cases when $g = 1$ or $g = 3$. This also supports previous theoretical analysis.

CONCLUSIONS

This study introduces a new watermarking scheme based on Chinese syntactic transformations. The watermark message is hidden in the sentence weights of the transformable sentences in the given Chinese text. To cope with the unexpected values during syntactic transformations, a watermarking bit is embedded into the sentence weights of a group of transformable sentences. Also, error correcting coding such as Hamming coding is applied to enhance watermarking reliability.

However, NLWM system has implemented only two typical Chinese constructions and this restricts its wide usage in real applications. Thus, working on implementing other typical syntactic constructions is in process. In later research, better error correcting codes that can correct more bits while having less coding redundancy will be tried.

The proposed watermarking scheme is only applicable to languages that need segmenting, e.g., Chinese, Japanese and Korean. But, it can also be extrapolated to English and many other occidental languages. As it can be seen, in such languages, the number of letters in the word at the same position in a given sentence might change during syntactic transformations. Thereby we can employ this value change to carry the watermark bit. This will be included in our future research.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (No. 60573045), National Basic Research Program of China (No. 2006CB303000) and Key Program of National Natural Science Foundation of China (No. 60736016).

REFERENCES

- Atallah, M.J., C.J. McDonough, V. Raskin and S. Nirenburg, 2000. Natural language processing for information assurance and security: An overview and implementations. Proceedings of the 2000 Workshop on new Security Paradigms, 10-13 September, ACM Press, New York, USA., pp: 51-65.
- Atallah, M.J., V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed and S. Naik, 2001. Natural language watermarking: Design, analysis and a proof-of-concept implementation. Information Hiding: 4th International Workshop, Lecture Notes in Computer Science, April, Springer, pp: 185-199.
- Atallah, M.J., V. Raskin, C. Hempelmann, M. Karahan, R. Sion, U. Topkara1 and K.E. Triezenberg, 2002. Natural language watermarking and tamper proofing. Lecture Notes in Computer Science, 7-9 October, Springer, pp: 196-212.
- Aura, T., 1996. Practical invisibility in digital communication. Lecture Notes in Computer Science, 30th May-1st June, Springer, pp: 265-278.
- Bergmair, R., 2007. A comprehensive bibliography of linguistic steganography. Proceedings of the 9th Conference on Security, Steganography and Watermarking of Multimedia Contents, Feb., 2007, 65050W-65050W.
- Brassil, J.T., S. Low and N.F. Maxemchuk, 1999. Copyright protection for the electronic distribution of text documents. Proceedings of the IEEE, July 1999, pp: 1181-1196.
- Busch, C., W. Funk and S. Wolthusen, 1999. Digital watermarking: From concepts to real-time video applications. IEEE Comput. Graph. Appl., 19: 25-35.
- Busch, J., 2007. On the optimality of the binary algorithm for the Jacobi symbol. Fundam. Inform., 76: 1-11.
- Chiang, Y.L., L.P. Chang, W.T. Hsieh and W.C. Chen, 2003. Natural language watermarking using semantic substitution for Chinese text. Digital Watermarking: Second International Workshop, Lecture Notes in Computer Science, 20-22 October, Springer, pp: 129-140.
- Cox, I.J., J. Kilian, T. Leighton and T. Shamoan, 1997. Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Proc., 6: 1673-1687.
- Grothoff, C., K. Grothoff, L. Alkhutova, R. Stutsman and M. Atallah, 2005. Translation-based steganography. Lecture Notes Comput. Sci., Springer, 3727: 219-233.
- Gupta, G., J. Pieprzyk and H.X. Wang, 2006. An attack-localizing watermarking scheme for natural language documents. Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, March 21-24, ACM Press, pp: 157-165.

- Hays, D., 1964. Dependency theory: A formalism and some observations. *Language*, 40: 511-525.
- Li, L.D., 1988. *Comparative Transformation Grammar of Chinese*. China Social Sciences Press, (In Chinese).
- Takahashi, T., T. Iwakura, R. Lida, A. Fujita and K. Inui, 2001. KURA: A transfer-based lexico-structural paraphrasing engine. *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium Workshop on Automatic Paraphrasing: Theories and Applications*, 27-30 November, pp: 37-46.
- Topkara, M., C.M. Taskiran and E.J. Delp, 2005. Natural language watermarking. *Security, Steganography Watermarking Multimedia Contents*, 5681: 441-452.
- Topkara, M., G. Riccardi, D. Hakkami-Tur and M.J. Atallah, 2006a. Natural language watermarking: Challenges in building a practical system. *Proceedings of the SPIE International Conference on Security, Steganography and Watermarking of Multimedia Contents*, 15-19 January, San Jose, CA., pp: A720-A720.
- Topkara, M., U. Topkara and M.J. Atallah, 2006b. Words are not enough: Sentence level natural language watermarking. *Proceedings of the ACM Workshop on Content Protection and Security*, 27 October, ACM Press, pp: 37-46.
- Vybornova, O. and B. Macq, 2007. Natural language watermarking and robust hashing based on presuppositional analysis. *Proceedings of IEEE International Conference on Information Reuse and Integration*, 13-15 August, pp: 177-182.
- Xu, C.S., J.K. Wu, Q.B. Sun and K. Xin, 1999. Applications of digital watermarking technology in audio signals. *J. Audio Eng. Soc.*, 47: 805-812.
- Yang, J.L., J.M. Wang, C.K. Wang and D. Y. Li, 2007. A novel scheme for watermarking natural language text. *Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 26-28 November, pp: 481-484.