Semantic Watermarking of Plain Text

Sasank Jampana
Department of Computer Science
University of Auckland
sjam048@ec.auckland.ac.nz

Abstract

Watermarking text has usually been considered as a subset of watermarking images with special characteristics, where imperceptible variations of inter-word and inter-line spacing are used to embed watermarks. These techniques are vulnerable to OCR attacks and cannot be applied to plain text. Syntactic and semantic watermarks were developed to overcome these difficulties. In a syntactic watermark the structure of the sentence is changed to embed a watermark, while in a semantic watermark the text is modified without changing it's meaning to embed a watermark. This report examines synonym substitution and natural language watermarking, two promising semantic watermarking techniques.

1. Introduction

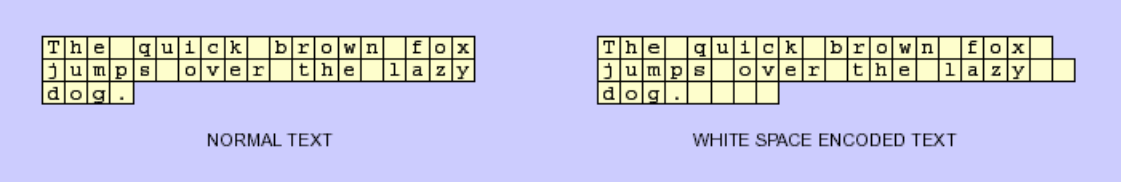
There has been a lot of research and development in the field of watermarking to help address the ownership problems of various media content [5]. The characteristics of a watermark are, they should be resilient, detectable only by the owner or the person possessing the secret key and must be easily produced by the watermarking software. This applies to watermarks in text documents too.

The definition of watermarking given by Nagra et al. [6] is as follows, “We define watermarking as the process of embedding a small amount of identifying information in media and we define such embedded information a watermark.”

Watermarking text should be considered in special light, because of the peculiarities and differences between text and other forms of media such as images, audio and video.
Text documents normally lack the kind of redundant information that is available in other media [3]. This has led to research in areas of watermarking where text is considered as an image with certain special characteristics. In these kind of watermarking techniques there are imperceptible changes made between the inter-word and inter-line spacing to embed a watermark.

There are three basic forms of watermarking plain text.
Open Space Watermarking,
Syntactic Watermarking and
Semantic Watermarking.

Open Space Watermarking. This category of watermarking techniques takes advantage of a reader's inability to 'see' the white spaces at the end of a line or in between words. These white spaces are used to embed the watermark in these documents [3,5].



Embedding a watermark using Open Space Watermarking. Source: bender et al. [3]

Though it is easy to implement open space watermarking it is susceptible to OCR attacks. It may not be feasible to employ this method of watermarking when the text documents are distributed in a format where, the document is rendered according to the settings of the user's personal text editor [5].

Syntactic Watermarking. Watermarking text by changing the structure of the sentence without changing any words. It utilizes the ambiguity in punctuation marks [3,5].

Syntactic watermarking resist OCR attacks, but its application is limited because an adversary can easily attack the ambiguous use of punctuation and the watermark can be removed.

Semantic Watermarking. Here the actual words of the text are changed in order to embed a watermark in the document. [3,5] Semantic watermarks are resistant to OCR attacks and are much more robust compared to syntactic watermarks.

In this paper I look at the two of the working models employed in text watermarking that use semantic watermarking and analyze their resistance to certain attacks. The first watermarking technique is synonym substitution presented by Jensen [5] and the second is the natural language watermarking (NLW) by Atallah et al. [1,2]. In the final section these watermarking schemes are compared.

2. Synonym Substitution

The document model is presented, based on which the synonym substitution is performed, which the author of [5] intends to use for fingerprinting text in logical markup languages. Fingerprinting is a special case of watermarking, used to identify the user of the document. I referred to watermarking in the rest of this section when [5] was trying to fingerprint a document.

2.1. Document Model

According to [5] a document is considered to be an ordered hierarchy of content objects (OHCO), where the document can be divided into sub-sections and they are further divided until at the lowest level we have the basic text units. [5] When synonym substitution is performed, the basic text unit chosen decides the effectiveness of the watermark. The bigger the basic text unit, the greater is the chance of a meaningful substitution. So, in the watermarking of a document where the paragraph is chosen as the basic text unit, watermarking is more robust than where words are chosen as the basic

text unit. This hierarchical structure of a document reduces the impact of the synonym substitution with distance from the modified text [5].

Synonym Substitution is a technique in semantic watermarking where one word is substituted for another word. The substitution must satisfy these two conditions,

1. The meaning of the transformed text must remain the same after the substitution.
2. The transformed sentence should satisfy the hash value that needs to be embedded as a watermark.

In order that the first condition is satisfied, the substitution must identify the possible safe transformations if they exist. Safe transformations are those transformations that can be effected without any consideration for the context of the word [5]. If there are no safe transformations possible, then the nearest possible meaning must be substituted in the place of that word.

The resultant transformation is then checked if they satisfy the hash value. This transformation is then performed for all the basic text units until the watermark has been embedded.

The hash function used in the synonym substitution technique has the property that a change in text causes a change in hash value.

$$H(T+\Delta) = H(T) + \delta$$

In Synonym Substitution, Jensen [5] uses the sum modulo of the ASCII value for the letters in the hashed text as the hash function.

2.2. Inserting Watermarks

The mechanism used for inserting watermark in the text using synonym substitution is[5],

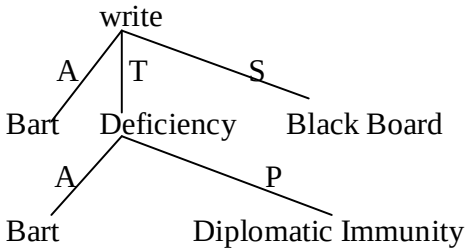
A key is used for inserting fingerprints and it is kept secret from the user. There is also a random number associated with the key to spread the fingerprint throughout the text.
The random number seeds a pseudo-random number generator to select the first sentence to watermark in the text. The hash function is applied to the selected text. If the hash value doesn't lie within the key interval then one of the words of the text is substituted with a synonym so that the resulting text's hash value would lie within the key interval.
The user chooses from the available synonyms, so that the resulting text conveys the same meaning. This process is repeated until all the watermarking bits have been inserted in the text.

3. Natural Language Watermarking

A semantic watermarking scheme for natural languages based on ontological semantics is presented by [1] with the following properties. Let T be the natural text to be watermarked and W the watermark shorter than T. The watermarked text T^1 is produced such that, it has the same meaning as T. It should be legally valid. W should be readable only with the secret key and even without the availability of T.

This approach uses three static and three dynamic resources. The three static resources are Ontology (O), Lexicon (L) and Text Meaning Representation (TMR). "The TMR is a list of propositions describing the events and concepts that represent the meaning of the text" [2]. The dynamic resources are Syntactic Parser (P), Analyzer (A) and a Generator (G) [1]. The parser generates a tree for each of each of the sentence as shown in the below example.
e.g.: Bart wrote, "I don't have diplomatic immunity", on the black board.

```
(write
        (agent Bart)
        (theme Deficiency
                (agent Bart)
                (property Diplomatic Immunity))
        (surface Black Board))
```



If the TMR doesn't yield the required bits for watermarking, another correct bit sequence is generated, by transforming the TMR so that it yields the required bits. This transformation is done without any significant change in the meaning of the text [1].

The most common transformations are [1],
Adjunct Movement. An adjunct phrase is added to a sentence.
Clefting. A noun phrase is added to the subject of a sentence.
Passivization. If a sentence contains a transitive verb, then it can be changed to passive voice without any change in meaning of the sentence.

Semantically empty phrases can also be added along with the above to perform the required transformations. Some examples of these phrases are generally speaking, basically, it seems that, etc. [1]

3.1. Watermark Insertion

A secret key is used to watermark the text. The secret key used here is a large prime p. Each of the sentences in the text has a corresponding tree and each of the trees has a corresponding binary string. The string determined by the quadratic residue of the hash value of the prime.

Now a marker is chosen, and part of the watermark is inserted to the sentence next to the watermark. "Marker is a sentence whose successor in the text is used to store bits of watermark…" [1] Here we use the above-mentioned transformations to insert the watermarks. If the transformations fail to insert the watermark bits in the sentence, a new sentence is chosen for the process to be repeated. The whole process is complete when all the watermarking bits have been inserted in the text.

3.2. Improved Watermarking Scheme

Atallah et al. [2] present an improved watermarking scheme where the trees are built based on the TMRs that are obtained from the analysis of the sentences of the text. This

building of TMR trees is known as ‘aborization’. Here a TMR tree usually represents a single sentence. TMR trees are built based on the principles, as given by [2]

1. The event proposition acts as the root of the TMR tree.
2. The filled slots of a concept are then suspended from the TMR tree. When more than one proposition share the same concept, the second one can be suspended from the first.

When there are many nodes at the same level from the meaning of the sentences the least significant one is placed at the bottom, so that it will be the first one examined.

3.3 Inserting the Watermark

Co-References. Generally, all text within a document when well written is cohesive. The theme of the text is established early in the document and all text that follows is related to the theme in a manner that adds more information to it. In the TMR proposition lists, the information that has been co-referenced between the sentences is presented as another parameter at the end of the list. [2]

Fact Database. Another tool used, where the information is made available from the database resource of the ontological semantics. This information is structured in a tree like hierarchy. [2]

To insert the watermark, the information from both the co-References and the fact database is taken into account. The information that has the most number of co-references is easy to manipulate because it can safely be removed or repeated. The information that is available from the fact database is used to add or substitute for parts of the TMR tree. The Least Significant Bits of a text are identified to make the following transformations, so that the meaning of the sentences is modified only in the least possible way. Accommodating the watermarking bits into the text is done by the following three transformations,

Grafting. Copying information from one sentence to another. It uses the additional information available in the co-references to copy them [2].

Pruning. Removing repeated information. When there is a co-reference for which there is more information repeated it is used for pruning. The likelihood of complete loss of information is avoided because the information is repeated at another place. It is always best to prune a co-reference that has a concept repeated more than twice [2].

Substitution. Replacing with the equivalent information. The fact database is used to make conclusions based on the information available. This information can then be appropriately added or substituted for parts of the text, without any significant change in the overall meaning [2].

These three methods can also be combined together to perform the required transformation. When any of the co-referenced part of the text is used for manipulation once, it is flagged. This helps in preventing the same co-referenced text from being pruned or grafted multiple times.

4. Analysis

We discuss the ability of the above watermarking techniques to withstand attacks and the feasibility of the watermarking schemes.

4.1. Security Analysis

There are three kinds of attacks on a watermark they are additive attack, distortive Attack and collusive attack.

Collusive Attack. When the attacker uses different copies of the document to identify the fingerprints and remove them [4,5]. The collusive attack is studied in the context of fingerprinting so I will be discussing only about the additive and distortive attacks.

Distortive Attack. When the attacker changes the document so that the existing watermark cannot be recognized [1,2,4,5].

In Synonym Substitution document, the attacker must perform 37 transformations on a text of 100 units with the key length 5 to get a 90% chance of destroying a watermark carrying text [5].

In Natural Language Watermarking, insertion of a new sentence has a probability of $2\alpha/n$ chance of damaging the watermark. When a meaning modifying transformation is performed on a sentence or when a contiguous block of sentences are moved to another place, they have a probability of less than $3\alpha/n$ chance of damaging the watermark.

Where α is the no.of sentences carrying watermarking bits and n is the total no. of sentences in the text document [1].

Additive attack. The attacker adds his own watermark to the existing watermarked document, so as to make it impossible to detect which watermark existed first [4,5].

In synonym substitution, the knowledge of a key is required to construct a valid fingerprint, however the issue when an additional watermark is added to the document and ownership is contested is not addressed by [5]. The ownership of the document could still be ascertained, if the author maintained a copy of the original document that has not been watermarked.

In Natural Language Watermarking, the authors suggest that the original document is not required because the watermark can be recovered from the text using the secret key [1,2]

However, the original needs to be retained in order to prevent additive attacks. There is a serious consideration for this attack because the process of embedding the watermark itself is not a secret, which aids the attacker.

4.2. Fidelity

Fidelity is defined as "the extent to which embedding the watermark deteriorates the original content" [6]. Natural Language Watermarking gives better fidelity compared to synonym substitution because NLW uses fact based fact database and co-references. This

helps NLW preserve the semantics to a greater extent compared to just synonym substitution.

The synonym substitution scheme presents the OHCO model of document and an assumption is made that the semantic impact decreases with hierarchical distance from the text. This need not necessarily be true in all cases. An example of this would be a document that discusses a theory and concludes by drawing on the various proofs presented throughout the document. Another plausible example is a mystery novel, where the climax draws on clues that have been sparingly interspersed.

5. Conclusion

The ability of the NLW to retain the semantics after watermarking and its ability to insert large watermarks into short text makes it a very useful tool for documents such as news relays [2]. NLW can also be used for non-critical classified information, especially in the military. One crucial requirement for using NLW is the ability to invest in the state of the art natural language processing techniques.

Synonym substitution has an advantage over NLW because it is easy to implement. It is not as expensive as NLW. There is lesser computational complexity. [5]. It can be used in manuals, business letters and other documents where giving some assistance during the synonym substitution process is not much hindrance.

But the ability to watermark plain text effectively is directly dependent on the linguistics and the capabilities of Natural Language Processing and Machine Translation.

6. References

[1] Atallah, M.J., V. Raskin, M. Crogan, C. F. Hempelmann, F. Kerschbaum, D. Mohamed, and S.Naik 2001. Natural Language Watermarking: Design, Analysis, and a Proof-of-Concept Implementation. In: I.S. Moskowitz (ed.), Information Hiding: 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 2001 Proceedings. Lecture Notes in Computer Science, Vol. 2137, pp. 185-199.

[2] Atallah, M.J., V. Raskin, C.F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K.E. Triezenberg 2002. Natural Language Watermarking and Tamperproofing. In: F.A.P. Petitcolas (ed.), Information Hiding: 5th International Workshop, IH 2002, Noordwijkerhout, The Netherlands, October 7-9, 2002 Proceedings.
Lecture Notes in Computer Science, Vol. 2578, pp. 196-211.

[3] Bender, W., D. Gruhl, N. Morimoto, A. Lu 1996. Techniques for Data Hiding.
IBM Systems Journal 35, Nos. 3-4, pp. 313-336.

[4] Collberg, C., and C. Thomborson 2000. Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection. In: IEEE Transactions on Software Engineering, August 2002. Vol. 28, Issue 8, pp. 735-746.

[5] Jensen, C.D. 2001. Fingerprinting Text in Logical Markup Languages.
In: G.I. Davida, Y. Frankel (eds.), Information Security: 4th International
Conference, ISC 2001 Malaga, Spain, October 1-3, 2001 Proceedings.
Lecture Notes in Computer Science, Vol. 2200, pp. 433-445.

[6] Nagra, J., C. Thomborson, and C. Collberg 2002. A functional taxonomy for software watermarking. In: M.J. Oudshoorn (ed.), 25th Australasian Computer Science Conference, ACSC 2002, Melbourne, Australia, January 2002 Proceedings. Conferences in Research and Practice in Information Technology, Vol. 4, pp. 177-186.