

Natural Language Watermarking

Foundations for Individual Marking of Text Data

Oren Avni (Halvani)

Institut für Sprach- und Literaturwissenschaft,
Technische Universität Darmstadt,
64289, Darmstadt
Oren@Halvani.de

Abstract

This paper discusses natural language watermarking, which analyzes patterns inside sentences of a given natural language text document, in order to embed individual watermark messages. The term "natural language watermarking" stands for the process of the embedding of watermark messages into a text document, using natural language components as the carrier, in such a way that the modifications are imperceptible to the readers and the embedded information is robust against possible attacks. The paper presents several methods that process the watermark embedding automatically, without any interaction of the user. This, however, can often cause semantic distortion after applying the embedding transformation. For this reason, the usability as well as the assets and drawbacks of the described methods are additionally analyzed with respect to important watermarking properties such as fragility, robustness, attacks, complexity or capacity.

Keywords: text watermarking, natural language watermarking, linguistic steganography.

1 Introduction

Today and certainly in the foreseeable future, the largest part of the information in our society is available as texts of the most different kinds. The total amount of textual information is undergoing continuous growth. Even more important, the internet has led to a highly accelerated growth in the amount of machine-readable data. Besides this

process' positive impact on the emergence of an "information society", it has however also complicated the separation of contents authored by different intellectual property owners.

Digital watermarks are increasingly used to cope with this challenge and prove content authorship for digital information. For multimedia files like images, audio, or video, such measures have been in use since the last two decades. For text documents, to my best knowledge, there are no practicable and especially automated methods for covert information embedding at the moment.

At the end of the 1990s several text-formatting related watermark techniques (e.g., *Line/Word-Shift Coding* or *Case Alternation*) have been developed and after that improved by various researchers (Adnan M. Alattar et al., 1997) to make text watermarking become reality. The watermark embeddedness however, has been found out to be very easy to destroy (for example through simple retype/lowercase attacks). A couple of years later, another approach to embed watermarks through judicious injection of typos was proposed by (Atallah et al., 2007), which based on the idea that error-prone sort of text, such as emails, forums or blogs can carry watermark bits in a quite resilient manner. At the same time however, this kind of technique may not be used for sensitive texts like scientific or literature texts.

Consequently, the vision to embed watermarks at the language level became more realistic and since this time, a lot of research work has been done in this field. During this time, the term "Natural Language Watermarking" has originated and represents an own discipline, which has been derived from

the well-known discipline of steganography. The following figure (Cox et al., 2008) illustrates the number of publications at the IEEE archive during 1991 - 2006:

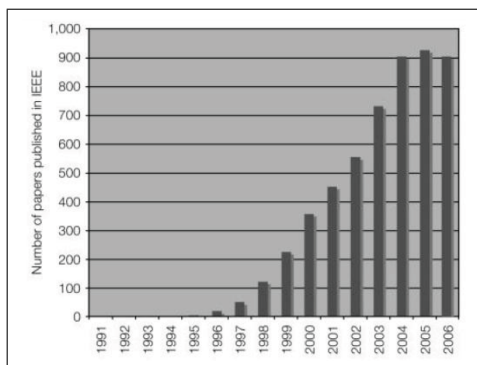


Figure 1: Publishings at the IEEE referring to the subject of "digital watermarks and steganography"

It can be seen from this image, that the interest in this field has rapidly grown over the last two decades.

The paper is structured as follows, in section 2 a symbolic notation for the most important terms in this paper is introduced, followed by the relation between both disciplines Natural Language Watermarking (NLW) and Natural Language Processing (NLP) in section 3. Afterwards some characteristics and requirements of NLW are stated in section 4. Next to that, an overview about several embedding methods, that embed watermark messages into text documents are given in section 5. The target language for these methods is originally german, as this language was a requirement for my bachelor thesis (Oren Halvani, 2010), where this paper is based on. As a consequence, the adaptivity between german and other languages is outlined here but also in section 6 ("known challenges"). Finally the conclusions are given in Section 8.

It should be mentioned at this point, that this paper does not provide a survey of the current state of the art in NLW, since there are a lot of papers, which have been already published so far. (R. Bergmair, 2007) for example, provides a comprehensive bibliography according to this subject.

2 Notation

Before introducing the discipline of NLW, the reader should familiarize himself with the com-

mon used terms and notation in NLW. The following table describes several symbols, that will be used in the entire paper, in order to keep the readability as compact as possible.

Symbol	Description
\mathcal{T}	The original text document
\mathcal{T}'	The watermarked text document
\mathcal{W}	The watermark message, formally defined as $\mathcal{W} \in \bigcup_{i \in \mathbb{N}} \{0, 1\}^i$
\mathcal{W}_{bit}	A single watermark message bit
$\mathcal{T} =_{sem} \mathcal{T}'$	Semantic equivalence between \mathcal{T} and \mathcal{T}' .
$\mathcal{T} =_{syn} \mathcal{T}'$	Syntactic equivalence between \mathcal{T} and \mathcal{T}' .

Table 1: Symbolic notations for NLW

Both equivalence relations: $\mathcal{T} =_{syn} \mathcal{T}'$ and $\mathcal{T} =_{sem} \mathcal{T}'$ are discussed in detail in section 5.

The watermark message \mathcal{W} is the most important part of NLW. To put it simply, \mathcal{W} is nothing but a simple bitstring, consisting of "0" and "1". Usually \mathcal{W} is chosen by the user (the author) but it can also be generated by the NLW system together with a secret key provided by the user. There are several possibilities to interpret the embedding of \mathcal{W} . For instance, let $\mathcal{W} = 1001$ and \mathcal{S} be the NLW system. \mathcal{S} can then embed \mathcal{W} as follows. The first "1" tells \mathcal{S} to lookup the first available sentence in \mathcal{T} which can be taken to embed a single \mathcal{W}_{bit} . In contrast, the next two "0" informs \mathcal{S} to skip sentences that can carry exactly two \mathcal{W}_{bit} . The last "1" tells \mathcal{S} again to use the next available sentence to embed the last \mathcal{W}_{bit} . Note that this is only one option to embed \mathcal{W} into \mathcal{T} . Of course the "0" in \mathcal{W} can also be embedded, but this would be very expensive, since \mathcal{T} usually offers only a limited capacity to embed \mathcal{W} .

In (Oren Halvani, 2010) it has been statistically shown, that a text document (category: press releases) with an average size of 4.23 KByte can carry 1-4 \mathcal{W}_{bit} , depending on one specific method. Therefore only "1" are considered to be embedded within \mathcal{T} .

3 The link between NLW and NLP

Given the fact that this paper links two disciplines together, it should be mentioned, how exactly these both relate to each other. First of all, NLW combines many tools and techniques, which

have their roots in the NLP domain, for instance:

- Tokenization/pattern matching
- POS-Tagging
- (Chunk)-Parsing
- Text simplification/paraphrasing
- Word sense disambiguation
- and many more

Most of these methods are usually treated as standalone components in a NLP environment, where NLW acts as an instance of it, in order to embed watermark messages into a document. To accomplish this task, NLW has to combine various components together, due to the fact, that there is rarely only one component in a typical NLW application. The term "NLP pipeline" is commonly used here, where the text runs through a linear arrangement of the involved NLP components. For example:

Tokenization → *POS-Tagging* → *Parsing* → ...

After the NLP preprocessing work is done, the NLW methods can be applied to complete the task. As a consequence, NLW would not be realizable without the tools and techniques provided by NLP.

4 Characteristics of NLW

This section depicts the most important characteristics and requirements, which have to be taken into account, when building a NLW System (Topkara et al., 1997).

1. **Meaning:** The meaning of the text is its value, and it should be preserved through watermarking in order not to disturb the communication.
2. **Grammaticality:** The embedding process should comply to the grammar rules of the language, in order to preserve the readability of the text. Preserving grammaticality is also required to be robust against statistical attacks that can automatically check for grammatical abnormalities in the text.

3. **Fluency:** Fluency is required to represent the meaning of the text in a clear and readable way.
4. **Style:** Preserving the style of the author is very important in some domains such as literature writing or news channels. Moreover, attacks based on profiling the author using unwatermarked works of the same author would be successful unless the style is preserved.

The methods, presented in section 5 are primary focussing on (1),(2) and (3). Besides the four requirements there are many properties of NLW, where only the most important of them are mentioned here:

- **Fragility/robustness:** Fragility and robustness measures for the resilience of \mathcal{W} against certain manipulations to the carrier \mathcal{T} . Such manipulations can be for example file-format conversions, that convert a file like a PDF into another format like TXT. The goal of both properties in this case, is to make \mathcal{W} resistant against conversions, so that \mathcal{W} remains the same, even if \mathcal{T} has been converted into other formats for multiple times.
- **Attacks:** In the context of NLW the term "attack" stands for various manipulations with the goal to damage watermarks. Usually there is a distinction between "blind attacks" and "non-blind attacks". From the technical view however, there is no clear distinction, since it correlates to the attacker's intention. The following attacking methods are known for all kinds of digital watermarking and should be named at this point:
 - Attack against the robustness
 - Attack against the uniqueness of the author
 - Attack against the watermark itself
 - Attack against the communicability of the watermark onto other media
 - Attack against a set of different copies (of the work)
- **Complexity:** This property is a measure for the expense of embedding \mathcal{W} into \mathcal{T} .
- **Capacity:** The capacity property indicates how many \mathcal{W}_{bit} can be embedded (theoretically) into the carrier \mathcal{T} .

5 Embedding methods

The following section describes my own methods (besides the synonym substitution method) and give an explanation for each one, where and how it can be used to embed watermark messages. These methods are mainly grouped into the following categories:

- **Syntactic transformation:** This category consists of syntactic methods, where their goal is to embed \mathcal{W} through alternations of single words, phrases or even whole sentences. For these methods it is remarkable that the length of \mathcal{T} will always remain the same after applying a syntactic transformation. With other words, new text such as words, phrases, sentences or even punctuation are not allowed to be inserted into \mathcal{T} .
- **Semantic transformation:** This category in contrast, consists of semantic methods, where the goal here is to embed \mathcal{W} through substitutions of words, phrases and in some cases also whole sentences. Single words for example, are typically replaced by these methods through other words, that share the same sense as the originals.

5.1 Enumeration modulation

The enumeration modulation method is based on the idea, that the order of elements (conjuncts) inside a coordination structure, can be altered, as far as they are constituents, that are not subordinate to each other. A coordination structure could look for example like this:

- $\mathcal{T} = \textit{Paris is Europe's most } \textit{fashionable, exclusive and elegant city}...$

The bolded words are all of the same type (adjective) and describe the noun "city", thus they form independent constituents. For these constituents, an alternation of the order can be applied, without losing the semantic meaning of the whole sentence, for instance:

- $\mathcal{T}'_1 = \textit{Paris is Europe's most } \textit{exclusive, fashionable and elegant city}...$
- $\mathcal{T}'_2 = \textit{Paris is Europe's most } \textit{elegant, fashionable and exclusive city}...$
- $\mathcal{T}'_3 = \textit{Paris is Europe's most } \textit{fashionable, elegant and exclusive city}...$

Coordinations appear often in news texts, thus this method would best fit to these sort of text, especially because news texts are less sensitive to modification of style rather than literature texts. The enumeration modulation method can be applied to any language, that supports coordination structures. To my best knowledge, this includes all Indo-European languages (e.g., English, German, Norwegian, Dutch, etc.).

5.2 Conjunction modulation

The aim of this method is to embed individual \mathcal{W}_{bit} into \mathcal{T} in the way that two open class words (e.g., nouns, verbs, adverbs or adjectives), which are connected through coordinating conjunctions like "and", "or", "but", etc. are being swapped each other. The following example demonstrates the idea:

- $\mathcal{T} = \textit{EU member states including } \textit{Spain and Germany } \textit{voiced concerns about}...$
- $\mathcal{T}' = \textit{EU member states including } \textit{Germany and Spain } \textit{voiced concerns about}...$

This method is slightly different from the previous enumeration modulation method in such a way that, instead of swapping certain conjuncts inside a coordination structure, the conjunction modulation method swaps only two words, which are standing outside an enumeration. The idea however, is still the same. The degree of freedom, to change the order of constituents inside a sentence, preserves the syntax and in most cases the semantic of the sentence. In some cases however, the style of the sentence might be distorted, since there are well-known phrases, where the order of constituents plays a prominent role, e.g.,:

- $\mathcal{T} = \textit{...in American pop folklore by Arthur Penn's 1967 film } \textit{Bonnie and Clyde}.$
- $\mathcal{T}' = \textit{...in American pop folklore by Arthur Penn's 1967 film } \textit{Clyde and Bonnie}.$

In the case of \mathcal{T}' it must not be possible to swap the order of the names, since "Bonnie and Clyde" forms a immutable phrase, which has been used as titles/headline in movies, books or also in the news. Hence the equivalence $\mathcal{T} =_{sem} \mathcal{T}'$ is not given in this specific case. One possibility to handle this problem, is to look up well-known phrases in dictionaries, wordlists/gazetteer lists and to tell the embedding method to skip them, if they appear

in \mathcal{T} . This workaround however, is aligned with some problems for instance such as coverage, but it might work in many cases if large dictionaries are used and updated constantly.

Besides this issue, the conjunction modulation is a quite useful method, due to the fact that conjunctions are used very widely in almost any kind of text. Thus, the capacity to embed \mathcal{W} is often bigger in relation to other methods. Likewise the enumeration modulation method, this method has the same advantage to be adapted into other languages, rather than german.

5.3 Center permutation

This method arose from the flexibility to alternate constituents in the midfield of a sentence, within the German language. The background of the idea explains itself as follows, in German the most sentences have the same structure, consisting of a prefield followed by an infinite verb, the midfield and (optional) the backfield. Unlike in English, the midfield in German includes often independent constituents, which can be altered the same way, as shown in the previous methods. The center permutation uses this degree of freedom to embed a \mathcal{W}_{bit} through a rearranging of these constituents.

The below illustration of a syntax tree shows the splitting of the following german sentence into its chunks, where the red marked one are the constituents of the midfield:

- $\mathcal{T} = \text{''Opa Rainer hat seinem Enkel das Buch gestern geschenkt''}$

The sentence can be translated to English as:

- $\mathcal{T} = \text{''Yesterday grandpa Rainer gives his grandchild the book as a present''}$

As one can see, the midfield consists of the following three constituents:

- $\sigma_1 = \text{''seinem Enkel''}$ ("his grandchild")
- $\sigma_2 = \text{''das Buch''}$ ("the book")
- $\sigma_3 = \text{''gestern''}$ ("yesterday")

Each one of the σ_i can be re-arranged within the midfield in this specific case. Therefore there are exactly $3! = 6$ possibilities to embed a \mathcal{W}_{bit} without a loss of syntax or semantic:

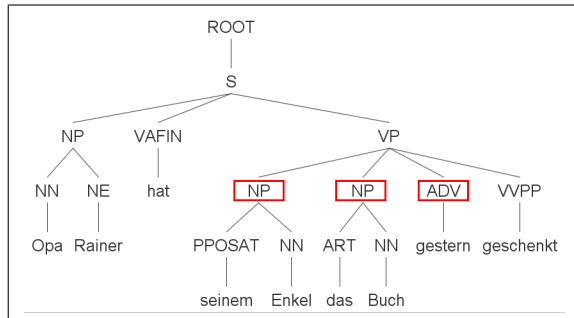


Figure 2: The syntax tree of the prefield, infinite verb, midfield and the backfield”

- $\mathcal{T}' = \dots \boxed{\sigma_1} \boxed{\sigma_3} \boxed{\sigma_2} \dots$
- $\mathcal{T}' = \dots \boxed{\sigma_2} \boxed{\sigma_1} \boxed{\sigma_3} \dots$
- $\mathcal{T}' = \dots \boxed{\sigma_2} \boxed{\sigma_3} \boxed{\sigma_1} \dots$
- $\mathcal{T}' = \dots \boxed{\sigma_3} \boxed{\sigma_1} \boxed{\sigma_2} \dots$
- $\mathcal{T}' = \dots \boxed{\sigma_3} \boxed{\sigma_2} \boxed{\sigma_1} \dots$

In order to detect such constituents in \mathcal{T} , chunk parsers are typically used. Once they are found, it must be ensured that their alternation is justifiable. This can be inspected through a statistical lookup of the modified midfield against a large corpus. If enough phrases which include the same modification can be found in the corpus (threshold depends on certain corpus parameters), then the transformation can be applied with a clear conscience.

5.4 Hyponym & Troponym truncation

This method has been proposed in (Oren Halvani, 2010) as a draft, since it could not be evaluated, due to the lack of resources (primarily hyponym databases). The idea of this method is quite different in comparison to the previous presented methods. Instead of applying an alternation on the order of words, this method replaces a word ω with a suitable hyponym ω' , if and only if the following conditions hold:

- ω is either a noun or a verb
- ω is a direct hyponym of ω'
- ω' is a substring of ω

Besides these conditions it must be additionally ensured that the substitution is acceptable. This can be made possible through the same way, as mentioned in the previous methods (checking the phrase that includes the modified word against one

or better several corpora). The following example demonstrates the idea behind the method:

- \mathcal{T} = Our neighbour has been bitten by a **Staffordshire bull terrier** this morning.
- \mathcal{T}' = Our neighbour has been bitten by a **bull terrier** this morning.

The watermarked document \mathcal{T}' takes the loss of the name "Staffordshire" into account, since this information may be unnecessary, depending on the wider context of the sentence. This however, is a dangerous assumption of the method, since it cannot be 100% guaranteed if the loss is acceptable, even not, if a statistical lookup might suggest it. Therefore this method is declared as a "draft" until further investigation can be done.

5.5 Synonym substitution

The synonym substitution is a very well known method in the history of NLW. A couple of researchers have developed systems that use this method to embed watermark messages into texts. In (Winstein et al., 1999) Keith J. Winstein has implemented the Tyrannosaurus Lex framework, the first system that used substitution of synonyms. The idea behind this method is quite simple. First a knowledge base (e.g., WordNet) is required, in order to automatically generate synonym tables (also called synsets) for a given word ω , where ω can be either an adjective or an adverb. The reason for the limitation of these open word classes is, that they provide more synonyms as nouns or verbs (e.g., for the nouns "dog", "cat" or "mouse" there are no equivalent synonyms).

Once the synsets are generated, the senses within these synsets have to be considered as "interchangeable". In order to enable this, Winstein divided the further substitution process into the following four cases. Let α , β and γ be three words, than:

1. α and β are both single-sensed, and they are in the same synonym set $\{\alpha, \beta\}$. In this case, the substitution of β for α is clearly allowed under theoretically any circumstances (that is, for any sense used of either), and thus α and β are considered interchangeable.
2. α and β are doubly-sensed, as such: $\mathcal{S}_1 := \{\alpha, \beta\}$, $\mathcal{S}_2 := \{\alpha, \beta, \gamma\}$. Although α and β

are doubly-sensed, the unconditional substitution between the two can be still allowed, since for every possible sense used of α , β is also in the corresponding synonym set. Therefore the set of interchangeable words is $\mathcal{S}_1 \cap \mathcal{S}_2$.

3. $\{\alpha, \beta\}$, $\{\alpha, \gamma\}$ No substitution can be made here for any word, since there is no word which is present in all of the synonym sets of any other word.
4. α and β are both single-sensed, and they are in the same synonym set $\{\alpha, \beta\}$, but β has a space in it. Than a substitution is not allowed, since multi-word concepts that include spaces are not supported by the system.

Having all of these requirements together, one can than embed \mathcal{W} in the following simplified example. Let $\mathcal{W} = 101$ and $\mathcal{T} =$ "Darmstadt is a wonderful little city", where the braces indicate the synsets:

$$\text{Darmstadt is a } \left\{ \begin{array}{l} 00 \text{ wonderful} \\ 01 \text{ decent} \\ 10 \text{ fine} \\ 11 \text{ great} \end{array} \right\} \text{ little } \left\{ \begin{array}{l} 0 \text{ city} \\ 1 \text{ town} \end{array} \right\}$$

Than, \mathcal{W} is splitted up into $10 = \text{fine}$ and $1 = \text{town}$, so that the result becomes: $\mathcal{T}' =$ "Darmstadt is a fine little town".

One can see from this example that the result may cause flaw, due to the fact that the biggest problem of synonym substitution is to find to most suitable sense for a given word ω in its surrounding context. Hence this method refers to the word sense disambiguation (WSD) discipline, which is a famous open problem in NLP. Many reasearch work all over the world has been done in this field for allmost 60 years. However, till today WSD is an unsolved task, and so also the synonym substitution method itself. To my best knowledge, there is no bullet proof system with the ability to substitute ω through a synonym such, that in any context the sense remains the same, as the sense of ω .

Another approach rather than substitute only single words, is to replace phrases through other suitable phrases that describes the same semantical meaning. This workaround does not need to take the surrounding context of ω into account, compared to the classical approach. Besides this, sub-

stituting phrases is less sensitive to ambiguity issue, since phrases often act as closed units in a sentence (constituents). The following example elucidates this approach:

- $\mathcal{T} = \text{In view of the fact that}$ students have to face difficult exams. . .
- $\mathcal{T}' = \text{Given that}$ students have to face difficult exams. . .

Due to my investigation in the NLW field (Oren Halvani, 2010), this approach seems to be unimplemented until today, possibly due to a lack of resources (e.g., phrase databases).

Note: The approach may not be confused with the technique of machine translation, since the targeted language is the same language itself. Instead the approach refers to the discipline of natural language generation (NLG), which is also been used in machine translation for years.

6 Known challenges

In this section several known problems of NLW are discussed. For each one of them a suggestion is explained, that describe how to counteract the problem or even how to avoid it.

6.1 Capacity

One of the biggest drawbacks of NLW methods is to gain a large number of \mathcal{W}_{bit} embeddedness (capacity). As a consequence, several methods have to work together in order to embed the entire \mathcal{W} cooperatively. As a result of this cooperation it can happen, that a couple of embedding methods may block each other during the watermark process, so that \mathcal{W} cannot be embedded through the methods, even if they are able to gain a bigger capacity as \mathcal{W} itself. One potential solution for this problem, is to use methods of only one category (e.g., syntactic transformations) rather than mixing both categories together, since a blockage of methods often occurs, when first applying semantic and afterwards syntactic transformations.

6.2 Different types of texts

Very often NLW methods, which work properly on one sort of text (e.g. newstexts, press releases, etc.) cannot be applied on other types (e.g., scientific papers). Hence, before applying NLW methods on unknown text, it should be ensured, that these are applicable to the given text. One option

for instance is to use text classification techniques, in order to organize texts into their appropriate categories. Text classification (also known as text categorization) is the task of automatically sorting a set of documents into categories from a predefined set. Once the category of a text is known, specific methods can be prioritised to be applied on it or even to be excluded from the watermark process, in order to perform better transformations.

6.3 Adaptivity between other languages

Most of the NLW methods are possible, because of the degrees of freedom within the language specific grammar. If these methods should also work in other languages, the same degrees of freedom must exist on the target language. This however is not realistic, since each language has its own rules of grammar and so its own degrees of freedom, that can be used to embed the watermark messages. Therefore it is very difficult to build methods, that work on several languages at once, even then, if the languages share alot of words or grammar rules (e.g., Belarusian and Ukrainian). For this reason it is recommended to use for each language a set of embedding methods, that don't have to deal with several languages at once.

6.4 Beyond syntax and semantics

Syntax and semantics are the most important properties, that must be preserved by NLW methods when applying text transformations. But very often these methods ignore the authors style of writing, so that as a result, modified text might became useless. In comparison with scientific texts, this problem occurs in literature texts more often. This problem is not easy to overcome, since almost every author has its own style of writing, besides for example scientific texts, where style is more or less the same. One possible solution could be to use methods, that cause only small loss of style (e.g., conjunction modulation) instead of methods like the Hyponym & Troponym truncation, which can lead often to a huge distortion. However, preserving style changings cannot be 100% guaranteed, unlike preserve syntax and semantics, because there is no method that can cover all possible author writing styles.

6.5 Invertible NLW methods

In some critical application areas, it is not allowed to modify the contents of the text, like military,

legal or literature fields. Therefore, NLW methods must be able to restore the original contents of the text in such cases, because even a very slight change of the text could lead to fatal effects in a specific scenario. These kind of inversions however, are only possible for certain types of NLW methods. Syntactic transformation related methods, such as enumeration/conjunction modulation can be easily aligned to become invertible, since applying the same transformation on \mathcal{T}' will result in \mathcal{T} . For example let $\pi_{\mathcal{W}}(\mathcal{T})$ be a syntactical transformation, that results in \mathcal{T}' using enumeration modulation as the embedding method and \mathcal{W} as the watermark message. Then, the embeddedness of \mathcal{W} within \mathcal{T} becomes neutralized through the following transformation:

$$\pi_{\mathcal{W}}(\mathcal{T}) \rightsquigarrow \mathcal{T}', \pi_{\mathcal{W}}(\mathcal{T}') \rightsquigarrow \mathcal{T}$$

This however, works only under the assumption that the same \mathcal{W} is used and that the alternation of the word order is not done randomly by the embedding method.

In contrast to this, semantic transformation related methods are very difficult to become reversible, due the fact that in some cases its unknown, how often a transformation must be applied to obtain \mathcal{T} from \mathcal{T}' . For example, assume \mathcal{T}' is a result of applying the synonym substitution method on \mathcal{T} . A knowledge base, which is required to build the synonym sets (synsets) is often updated and as a result of it, the set of possible senses for a given word is not fixed, thus its unknown how many times a transformation must be applied to restore the original contents of the text.

7 Conclusion

Natural language watermarking using linguistics techniques is a relatively new field of research with large potential for many applications. Currently, there are no fully functional systems beyond the proof-of-concept level, although the interest in this field has grown rapidly over the last two decades (as shown in the introduction of this paper). There would be rapid improvements in NLW, if important resources could be worldwide freely and/or easily available for researchers in this field. Unfortunately many important resources such as large synonym/hyponym databases, parallel corpora or even word-webs for certain lan-

guages (e.g., "GermaNet" for the german language) cannot be accessed freely without licences, permissions or fees. As a result of if investigation on several methods and techniques cannot be carry out and thus research work may end at some point.

Besides a lack of resources there are other obstacles that have to be faced, when building NLW systems. One of these is for instance the acceptance by the community. E.g., for literary authors NLW may be unthinkable until researchers can fully guarantee, that their methods will not decrease the quality of the text contents. This goal however, is far away at the moment and thus researchers concentrate on certain types of text rather than to look at whatever arises.

References

- I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker. 2008. *Digital Watermarking and Steganography*. Second Edition Morgan Kaufmann Publishers, Amsterdam/Boston.
- Richard Bergmair. 2007. *A Comprehensive Bibliography of Linguistic Steganography*. University of Cambridge Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK.
- Oren Halvani. 2010. *Foundations for individual marking of text data*. Technische Universität Darmstadt, Fraunhofer-Institut SIT.
- Mercan Topkara, Giuseppe Riccardi, Dilek Hakkani-Tür, Mikhail J. Atallah. 2006. *Natural Language Watermarking: Challenges in Building a Practical System*. Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents.
- Adnan M. Alattar and Osama M. Alattar. 2004. *Watermarking electronic text documents containing justified paragraphs and irregular line spacing*. In Edward J. Delp III. and Ping Wah Wong, editors, Security, Steganography, and Watermarking of Multimedia Contents VI. Proceedings of the SPIE, volume 5306 of Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, pages 685–695.
- Mercan Topkara, Umut Topkara, Mikhail J. Atallah. 2007. *Information Hiding through Errors: A Confusing Approach*. Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, San Jose, CA.
- Keith J. Winstein 1999. *Lexical steganography through adaptive modulation of the word choice hash*. Was disseminated during secondary education at the Illinois Mathematics and Science Academy. The paper won the third prize in the 2000 Intel Science Talent Search.