# Words Are Not Enough: Sentence Level Natural Language Watermarking

Mercan Topkara Umut Topkara Mikhail J. Atallah *
Department of Computer Sciences
Purdue University
West Lafayette, IN, 47906, USA
mkarahan,utopkara,mja@cs.purdue.edu

## ABSTRACT

Compared to other media, natural language text presents unique challenges for information hiding. These challenges require the design of a robust algorithm that can work under following constraints: (i) low embedding bandwidth, i.e., number of sentences is comparable with message length, (ii) not all transformations can be applied to a given sentence (iii) the number of alternative forms for a sentence is relatively small, a limitation governed by the grammar and vocabulary of the natural language, as well as the requirement to preserve the style and fluency of the document. The adversary can carry out all the transformations used for embedding to remove the embedded message. In addition, the adversary can also permute the sentences, select and use a subset of sentences, and insert new sentences. We give a scheme that overcomes these challenges, together with a partial implementation and its evaluation for the English language. The present application of this scheme works at the sentence level while also using a word-level watermarking technique that was recently designed and built into a fully automatic system ("Equimark"). Unlike Equimark, whose resilience relied on the introduction of ambiguities, the present paper's sentence-level technique is more tuned to situations where very little change to the text is allowable (i.e., when style is important). Secondarily, this paper shows how to use lower-level (in this case word-level) marking to improve the resilience and embedding properties of higher level (in this case sentence level) schemes. We achieve this by using the word-based methods as a separate channel from the sentence-based methods, thereby improving the results of either one alone. The sentence level watermarking technique we introduce is novel and powerful, as it relies on multiple features of each sentence and exploits the notion of orthogonality between features.

*Portions of this work were supported by Grants IIS-0325345, IIS-0219560, IIS-0312357, and IIS-0242421 from the National Science Foundation, and by sponsors of the Center for Education and Research in Information Assurance and Security.


*MCPS'06,* October 28, 2006, Santa Barbara, California, USA.


## Categories and Subject Descriptors

H [**Information Systems**]: Models and Principles—*Security*

## General Terms

Algorithms, Design, Security

## Keywords

natural language watermarking

## 1. INTRODUCTION

Digital text forms one of the largest chunk of digital data people encounter daily. Internet has become one of the main destinations for knowledge acquisition, coupled with a steady increase in the amount of digital information resources it harbors. Many newspapers, magazines, scientific journals and conferences started providing all articles in digital format; and the number of digital libraries, personal blogs and open access encyclopedias are increasing daily. Besides, e-mail has become the main communication medium for many people. Even though being able to search and access immense amount of knowledge online has become a part of everyday life, it is still an open question as to how the authors or owners of digital text will have control on how their data is distributed or re-used. Rights management problems are more serious for text than they are for image, video and audio since it is much easier for users to download and manipulate copyrighted text and re-use it free from control.

Publicly available methods for information hiding into natural language text can be grouped under two categories. The first group of methods are based on *generating* a new text document for a given message. Spammimic [5] is an example of this first group. The second group of methods are based on linguistically *modifying* a given cover document in order to encode the message in it. Natural language watermarking systems (and this paper's framework) fall under the second type of systems. In watermarking there is also a need for resilience against an adversary who is attempting to destroy the mark without destroying the *value* of the watermarked document. For a review of closely related work in

sentence level watermarking and how our approach differs, refer to Section 5 and for a review of information hiding into natural language text, refer to [21] and [4].

In Section 2 we propose a rather generic information hiding algorithm into natural language text, where the carrier medium and the adversary model presents unique challenges. We provide a highly flexible system where (i) the watermark reading process is freed from using the exact same statistical Natural Language Processing (NLP) tools that were used while the watermark was being embedded, (ii) the watermark detection requirements are adjusted to be able to stand a given amount of attacks (i.e., embedding threshold is higher than detecting threshold) (iii) the complex and rich feature set of sentences are exploited to increase the bandwidth and robustness.

Natural language watermarking can be done at several levels of granularity, from word-level, to sentence-level, paragraph-level, all the way up to document or even collection-of-documents level. We have recently designed a word-based technique [22] and system ("Equimark") that achieves good embedding and resilience properties through synonym substitutions. When there are many alternatives to carry out a substitution on a word (that was selected as a message carrier), Equimark prioritizes these alternatives according to their ambiguity, and uses them in that order.

The present paper works at the sentence level while using the previous word-based technique. Unlike Equimark, whose resilience relied on the introduction of ambiguities, the present paper's sentence-level technique is more tuned to situations where very little change to the text is allowable (i.e., when style is important). Secondarily, this work shows how to use lower (word-level) marking to improve the resilience and embedding properties of higher (sentence-level) schemes. This is achieved by using the word-based methods as a separate channel from the sentence-based methods, thereby achieving better results than either one alone. Details of sentence level watermarking and the design decisions are discussed in Section 3.

Section 4 provides implementation and performance details of the presented algorithm for natural language watermarking at sentence level. This is the first system that brings together all the vital components of a sentence level watermarking system, and the first study that evaluates the quality of processes (i.e., sentence parsing, sentence generation, automatic linguistic transformations) of sentence level watermarking on a large data set.

## 2. INFORMATION HIDING ALGORITHM

In this section we describe an algorithm that can be used to hide information into any data as long as it has multiple features. Although the algorithm is presented for the specific case of natural language text, it can potentially be used in other domains (more on this later).

Let $D$ be a natural language text document consisting of $n$ sentences $d_1, \ldots, d_n$. Let $F$ be a set of Boolean returning functions on sentences, where each such function indicates the presence (or lack thereof) of a particular property in a sentence, e.g., an $f_i(d_j)$ could be an indicator of whether the sentence $d_j$ is passive or active, or whether it contains two nouns, or whether it contains a particular class of words, or whose hash has a least significant bit of 1, etc. We call each such function $f_i$ a feature function.

Let $T$ be the set of transformations that are available for modifying the sentences (e.g., synonym substitution, passivization). For each $t \in T$, $t(d_i)$ denotes the outcome of applying that particular transformation to $d_i$ (the "transformed" version of $d_i$). We use $\delta_1(t(d_i), d_i)$ to denote the amount of distortion that $d_i$ undergoes as a result of using transformation $t$ on it. Likewise $\delta_2(t(d_i), d_i)$ denotes the expected distortion that the adversary will cause after modifying $t(d_i)$ without the knowledge of $d_i$.

We henceforth use $M$ to denote the message to be embedded.

Our algorithm for information hiding into natural language text works under the following demanding constraints:

- The number of sentences $n$ can be small, i.e., comparable with message length $|M|$; contrast this with the earlier work in [2, 3] where $n$ needed to be much larger than $length(M)$.

- Relatively few transformations (if any) could be applicable to a given $d_i$, e.g., it is not possible to passivize a sentence with an intransitive verb ("I run every morning").

- A sentence may be transformable into a relatively small number of alternative forms, as there may only be a small number of transformations applicable to it. This limitation is governed by the grammar and vocabulary of the natural language (e.g. small number of synonyms, small number of paraphrases, rigid word ordering).

- The adversary can permute sentences, select a subset of the sentences, and insert new sentences. The resilience we achieve can handle arbitrary permuting, and extensive but not massive subset selection (e.g., selecting zero sentences) and insertion (e.g., many new sentences). More on how this resilience is quantified will be given in Section 4.

The feature functions in $F$ serve two distinct purposes: (i) some of them serve as indicators of the presence of a mark (we will generically denote functions used for this selection purpose with $f_s$); (ii) others will be used to actually help encode the bit(s) of $M$ that are embedded in a sentence (we will generically denote functions used for this embedding purpose with $f_e$). We said "help encode" because the $f_e(d_i)$ need *not* necessarily agree with the bit(s) of $M$ that $d_i$ is helping encode: The relevant bit of $M$ is encoded in the aggregate distribution properties of all such sentences that encode that bit (more on this later); a similar technique of using aggregate properties for encoding was done in [19], although in our case the sentence subsets that encode different bits can overlap which helps increase both capacity and resilience (in [19] these subsets were disjoint – no two items contributed to more than 1 bit of $M$).

In this framework, the process of embedding 1 bit, consists of transforming a number of sentences, $d_i$, so that their $f_s(d_i) = 1$ (i.e., they are selected for embedding), and making their $f_e(d_i)$ collectively encode the appropriate bits of $M$ by deviating significantly from the expected distribution of $f_e(d_i) = 1$. Embedding transformations either set $f_s(d_i) = 0$ to de-select a sentence, or set both $f_s(d_i) = 1, f_e(d_i) = 1$. The detection of this bit, consists of finding out whether the aforementioned statistical deviation holds.

It is important to have the flexibility to unmark a sentence, since in many occasions a transformation $t$ will not be able to yield $f_e(t(d_i)) = 1$.

The $f_s$ and $f_e$ need to be defined on an indivisible data unit, such as a word, a phrase or a sentence, depending on the adversary model of a particular information hiding application. For example one model could assume that the adversary cannot divide a sentence into two sentences.

The embedding process will be subject to a maximum allowed distortion threshold $\sum_{d_i \in D} \delta_1(d'_i, d_i) \leq \Delta_1$, where $d'_i$ is a message carrying sentence derived from $d_i$. $\Delta_1$ captures the tolerable loss in value of $D$ (in case of watermarking) or the loss of stealthiness of covert channel (in case of steganography). In case of watermarking the embedding process also aims to maximize $\sum_{d_i \in D} \mathbb{E}[\delta_2(d'_i, d_i)]$, which captures the expected distortion that the adversary will cause while attempting to remove the embedded message from $d'_i$.

MESSAGE INSERTION
**Let** $M$ be a message $(m_1 \ldots m_w)$
**Let** $D[]$ be a document of $n$ sentences, $d_i = D[i]$
**Let** $K$ be a secret key
**Let** $T[i]$ be the set of transformations applicable to $d_i$
**Let** $F$ be a set of boolean "feature" functions on $D$ ($f(d_i)$ returns a 1 if $d_i$ contains feature $f \in F$)
**Let** $F_s \subset F$ be the subset of message-presence indicator functions
**Let** $F_e \subset F$ be the subset of message-embedding indicator functions
**Let** $\hat{\Delta}_1$ be the maximum allowable distortion
**Let** $C[i]$ be the subset of message bits that $d_i$ contributes to encoding
**Let** $\text{GAIN}(d_j, d_k) \leftarrow \frac{\mathbb{E}[\delta_2(d_j, d_k)]}{\delta_1(d_j, d_k)}$ (intuitively, this is the "resilience gained", the distortion caused by the adversary per unit of distortion caused by the embedding)
**Let** BITSUCCESS$(f_s^i, f_e^i, D)$ return 0 if the $i^{th}$ message bit $m_i$ was not successfully encoded in the (modified) $D$ using $f_s^i$ and $f_e^i$; otherwise it returns a positive number that measures the statistical significance of the existence of $m_i$ in $D$ (the "strength" of the signal using, e.g., $\chi^2$)
**for each** $l = 1, 2, \ldots, |M|$
    Use $K$, $l$, $m_l$ as seeds to randomly select from $F$ an $f_s^l$ and an $f_e^l$
$D^0[] \leftarrow D[]$
$\Delta_1 \leftarrow 0$
**while** $T \neq \emptyset$
    For the bit $l$ that is in most dire need of help because of weak signal (i.e., having low $\chi^2$ score), try to help it as follows:
    For each sentence $d_j$, choose the transformation $t_{l,j} \in T[j]$ that helps the encoding (the $\chi^2$) of that bit $l$ while maximizing $\text{GAIN}(t(d_j), d_j^0)$ (i.e., maximizing resilience).
    Among all such pairs $(d_j, t_{l,j})$ choose the one that has highest $\text{GAIN}(t(d_j), d_j^0)$, call that pair $(d_i, t)$.
    $d'_i \leftarrow t(d_i)$
    $\Delta'_1 \leftarrow \Delta_1 - \delta_1(d_i^0, d_i) + \delta_1(d_i^0, d'_i)$
    **if**$(\Delta'_1 > \hat{\Delta}_1)$
        $T[i] \leftarrow T[i] - t$
        **if**$(T[i] = \emptyset)$
            $T \leftarrow T - T[i]$
        **continue**
    $\Delta_1 \leftarrow \Delta'_1$
    $d_i \leftarrow d'_i$
    $C[i] \leftarrow C[i] \cup \{l\}$
    Update $T[i]$ such that it includes only the transformations that would improve the current strength of all the message bits in $C[i]$.
**if** any of the $|M|$ bits was not successfully encoded, i.e., if some BITSUCCESS$(f_s^l, f_e^l, D) =$ FALSE
    **return** FALSE
**return D**

Note that the above algorithm continues to perform transformations until the maximum distortion $\Delta_1$ is reached, even after the message is successfully embedded. This is necessary to limit the flexibility of adversary.

Some of the modifications that the adversary performs on a sentence will not change the contribution of the sentence to the embedded message. In this paper, we do not leverage on such difference among modifications of the adversary. The current scheme simply tries to maximize the number of alternative sentences that the message is embedded in order to maximize its resilience against adversaries' modifications. An improved scheme should prefer to embed in those sentences which have a higher likelihood to carry the same embedded message even after the adversary's modifications.

We now describe the MESSAGE DETECTION algorithm, which reads an embedded message from a document that has undergone a message embedding. This algorithm does not, require the message $M$ to be available. If $M$ is available, the algorithm can be modified to use the BITSUCCESS for quantifying the confidence that the cover document $D$ carries $M$. We require that $M$ carries an error correction code, and it is possible to detect the termination of $M$ when $M$ is received as a growing string.

MESSAGE DETECTION
**Let** $M, D, K, F$ be defined as above
**Let** MAXMESSAGE be the largest message size
**Let** TERMINATED$(M)$ be a boolean function that decodes the partial message $M$ and returns TRUE if the message has terminated
**for each** $i = 1 \ldots$MAXMESSAGE
    Use $K$, $i$, 0 as seeds to randomly select from $F$ an $f_s^0$ and an $f_e^0$
    Use $K$, $i$, 1 as seeds to randomly select from $F$ an $f_s^1$ and an $f_e^1$
    **if**(BITSUCCESS$(f_s^0, f_e^0, D) >$ BITSUCCESS$(f_s^1, f_e^1, D)$)
        $M[i] \leftarrow 0$
    **else**
        $M[i] \leftarrow 1$
    **if** (TERMINATED$(M)$)
        **break**
**return** $M$

The algorithms that we have given in this section can be used to embed messages into any kind of collections of data units, where we are under similar constraints as natural language text but at the same time have a flexibly large number of features and a limited number of transformations in the arsenal of information hiding.

# 3. SENTENCE LEVEL WATERMARKING

We distinguish two types of modifications that can be used for watermarking text: The robust synonym substitution introduced in [22], and syntactic sentence-paraphrasing. Compared to naive synonym-substitution, robust synonym substitution introduces ambiguities in order to make it harder for the modification to be undone. Such modifications can somewhat damage the precision of the individual words used in the text, e.g., replacing " a slope in the turn of the road" with "bank". Sentence-level paraphrasing, on the other hand, typically does little damage to the precision of words, but may damage the stylistic value of the sentence. An

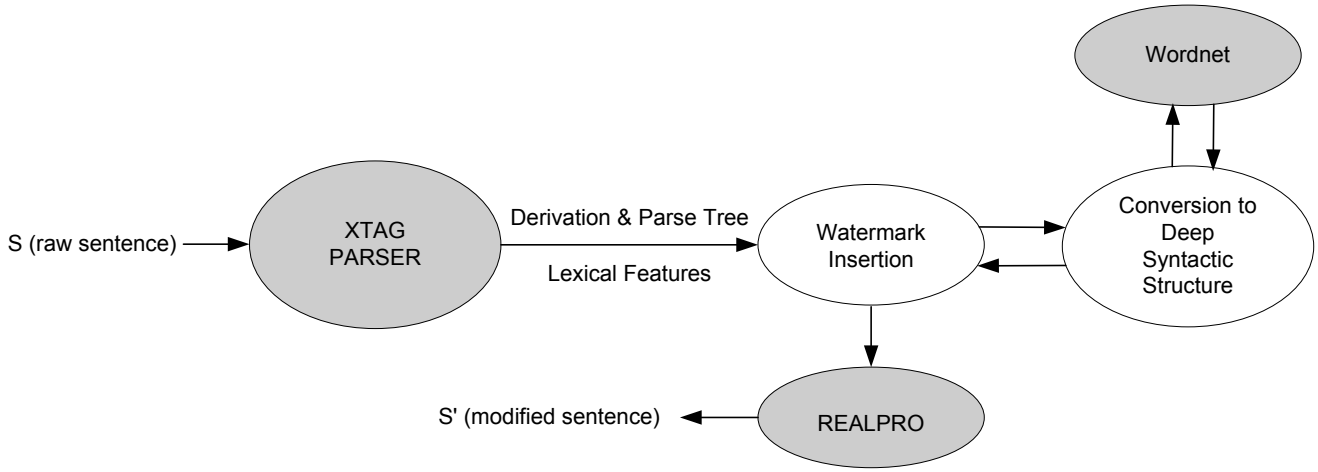


**Figure 1: A schema of the system that is being developed and tested for the baseline evaluations of the proposed sentence-level natural language watermarking system. This implementation extracts features, generates parse and derivation trees with *XTAG* parser and uses *RealPro* for surface realization.**

example that points out a possible one-wayness of sentence-level watermarking happens when the original sentence is "April had to hold a party", and gets transformed into "A party must be held by April". It will be hard for an adversary to undo this embedding without performing a co-reference resolution and context analysis on the full text. See Section 4.1 for several examples of sentence-level paraphrasing.

Depending on the type of the text (e.g., multimedia content, editorials, news reports, user manuals, etc), the requirements for the preservation of precision and the preservation of style can both vary. In user manuals, style requirements are less stringent, whereas precision cannot be compromised. Style is a more important value of editorials, whereas precision is more important in newswire. A text that accompanies video, audio or pictures as a secondary information resource, may have less stringent requirements on both style and precision. In addition to the above-mentioned differences between this work and [22], another difference is that whereas [22] focuses on precision, in this paper we investigate a method that can be used to trade precision for style.

## 3.1 Selection of sentences

As stated earlier, the sentence features used for selection are different and orthogonal to those used for embedding the message bit(s). We next discuss two alternatives for sentence-selection (the embedding of message bits is covered in the next section).

A subset of the vocabulary is pre-selected as mark-carrying (that subset is not known to the adversary). The message bits are inserted only in those sentences that contain a word from that subset. Of course this means that some inputs will not contain enough words from that special subset, and hence will be deficient in terms of their "markability". To avoid such situations, the selected vocabulary subset is chosen using a language model for the specific domain, to ensure that long enough sentences from this domain will usually not be so deficient; a language model for financial analysis texts will be different from a language model for Jack London's works, and the vocabulary-subset for the former will be very different from the latter's.

Alternatively to the above language-model based approach to select mark-carrying sentences, synonym substitution can be used to flexibly and adaptively mark sentences. In such a case, mark words are added to the text by replacing their synonyms in the original text. The slight shift in meaning due to synonym-replacement can be viewed as a robustness advantage: The adversary trying to do it wholesale will degrade the value of the work beyond desired limits.

## 3.2 Embedding

We now assume that the sentence at hand is selected for message-bits insertion (possibly using one of the two methods described earlier). As the features for embedding are orthogonal to those for selection, we can carry out the embedding without changing the "selected" status of a sentence. The way embedding is done by modifying the embedding features until they "speak the desired message bits". In the framework that is described here we distinguished between selection and embedding features. The selection features are determined by Equimark, where a sentence that has a word from the selected subset of the vocabulary is an information-carrier, and the embedding features are based on sentence-level linguistic features which can be "number of prepositions in a sentence", "a sentence being passive or active", "distance of certain functional words", or "the verb classes [12] of the verbs in a sentence".

The task of creating a statistically significant deviation in the distribution of embedding features in a selected set of sentences is not independent from the features that are used for selection and embedding. This distribution is based on the correlation between selection features and embedding features. For example, Sigmund Freud has a tendency of using double-negation(e.g. "this is not insignificant"), and if we were to watermark a text that heavily quotes from him, and if the words that are related to psychological research are used as selection features ("mark-carrying" words), the embedding feature of "sentence carrying double-negation" will be correlated with these selection features.

**"the democratic party has denied the allegations"**

**Figure 2: A sample sentence taken from the Reuters Corpus. Its publication day is 8th of January 1997.**

```
( S_r ( NP_r ( D the )
             ( NP_f ( N_r ( A democratic )
                          ( N_f party ) ) ) )
      ( VP_r ( V has )
             ( VP ( V denied )
                  ( NP_r ( D the )
                         ( NP_f ( N allegations ) ) ) ) ) )
```

**Figure 3: Syntactically parsed sentence, output of XTAG Parser on the sentence given in Figure 2**

# 4. SYSTEM IMPLEMENTATION AND EXPERIMENTS

The purpose of our experiments is not to stress-test the embedding capacity of our scheme, rather, it is to demonstrate the possibility of applying it on a real-life test case (Reuters [17] is a common benchmark used in NLP research). Therefore the reported embedding rates are not indicative of the potential of our proposed scheme, because what we implemented is only a partial system that uses (i) a small fraction of the available repertoire of transformations (only two of them), and (ii) the specific implementation of these transformations has a very restrictive domain of input sentences to which they apply (for ease of implementation).

The approach described in this paper is based on syntactically modifying the sentence structure. In order to be able to automatically manipulate the sentences, we first derive a structural representation of the sentences (through parsing [10]) and later revert this representation into surface sentence form (through generation [10]).

The output of the parsing may represent either the morphological, syntactical, or semantical structure of the sentence or it may represent a combination of these. Figures 2, 3 and 4 show a sentence in surface form, its syntactic parse tree and derivation tree (dependency tree) obtained using XTAG parser. We can use output of XTAG parser to find out features of sentences [23, 20] such as voice, question, superlative etc.

Our transformations use both the parse tree and the derivation tree in order to perform embedding transformations.

We transform a sentence that has been selected for watermark embedding as follows:

1. Parse the sentence by XTAG parser.
2. Verify if the sentence already carries the embedding feature. If so return, else go to next step.
3. For each available transformation;
   (a) Verify if the transformation is applicable to the sentence (e.g. for passivization, the root of the syntactic tree has to be a transitive verb). Refer to Figures 2, 3 and 4 for examples of information used at this step.
   (b) Embedding operation is performed in two steps:
      i. Re-write the dependency tree based on the design of the transformation. Refer to Figure 6 to see a transformed dependency tree generated during passivization. Note that "by" is added and made the parent of the subtree that has the subject of original sentence.
      ii. Convert the modified XTAG output into a deep syntactic structure (in DSyntS format) that reflects the "transformed" features of the sentence. Refer to Figure 7 for the deep syntactic structure representation of the sentence in Figure 2 after going through a passivization transformation.
   (c) Use RealPro to convert the resulting deep syntactic structure into surface sentence form. Figure 8 shows the result of realization for our example case.
   (d) Verify if the transformed sentence carries the embedding feature. If so, record the distortion value.
4. Commit the embedding transformation that imposes minimum distortion.

Comparing Figure 7 and Figure 9 will show the main idea behind the design of passivizing transformation implemented for this framework.

**Data Resources** We tested our system on 1804 sentences from the Reuters corpus [17]. We picked eleven publication days at random[1]. Later, from the articles that were published on these days, we picked the first 1804 sentences that are parsed with the XTAG parser. We are also using Wordnet [8] as a data resource for converting plural nouns to singular forms, and verbs into their base forms. This conversion is required for complying with the requirements of DSyntS.

**Parsers** Our implementation uses *XTAG* parser [2] [23] for parsing, dependency tree generation (which is called a derivation tree in the XTAG jargon) and linguistic feature extraction.

**Generator** We used *RealPro*[3] [11] for natural language generation.

[1]24th of August 1996, 20th of October 1996, 19th of August 1997 and 8 consecutive days from 1st of January 1997 to 8th of January 1997

[2]Available at http://www.cis.upenn.edu/ xtag/swrelease.html. In our experiments, we used $lem0.14.0.i686.tgz$

[3]See http://www.cogentex.com/technology/realpro/ for access to RealPro.

```
( alphanx0Vnx1[denied] ( alphaNXN[party]<NP_0> betaAn[democratic]<N> betaDnx[the]<NP> )
                       ( alphaNXN[allegations]<NP_1> betaDnx[the]<NP> ) betaVvx[has]<VP> )
```

**Figure 4: Sentence Dependency Structure, output of XTAG. See Figure 5 for a depiction of this tree.**

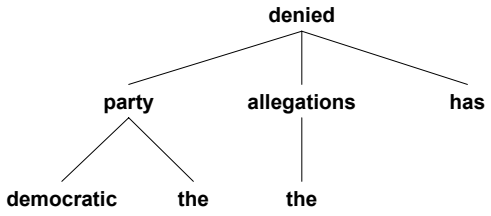


**Figure 5: Depiction of the dependency tree in Figure 5 for the sentence in Figure 2.**

```
( alphanx0Vnx1[denied] ( betanxPnx[by]<NP_r> ( alphaNXN[party]<NP_1> betaAn[democratic]<N> betaDnx[the]<NP> ) )
                       ( alphaNXN[allegations]<NP_0> betaDnx[the]<NP> ) betaVvx[has]<VP> )
```

**Figure 6: Sentence dependency structure for the watermark carrying sentence in Figure 8 generated by passivization process.**

```
DSYNTS:
deny[ class:verb voice:pass mood:past-part case:obj  taxis:perf tense:pres  ]
(   II by[  ]
    (   II party[ class:common_noun article:no-art case:nom person:3rd number:sg  ]
        (   ATTR democratic[ class:adjective  ]
            ATTR the[ class:article  ]             ))
    I allegation[ class:common_noun article:no-art case:nom person:3rd number:pl  ]
    (   ATTR the[ class:article  ]                  ))
END:
```

**Figure 7: Final DSyntS representation for the watermark carrying sentence shown in Figure 8, generated by the passivization process when the original sentence's XTAG parse output in Figure 3 and the dependency tree in Figure 4 are given.**

**"the allegations have been denied by the democratic party"**

**Figure 8: Watermarked version of the sample sentence in Figure 2**

```
DSYNTS:
deny[ class:verb voice:act mood:past-part case:obj  taxis:perf tense:pres ]
(  I party[ class:common_noun article:no-art case:nom person:3rd number:sg  ]
    (  ATTR democratic[ class:adjective  ]
       ATTR the[ class:article  ]   )
   II allegation[ class:common_noun article:no-art case:nom person:3rd number:pl  ]
   (  ATTR the[ class:article  ]    )
 )

END:
```

**Figure 9: The DSyntS format generated for the sentence in Figure 2, if it was directly processed by conversion process without any transformation process' interference.**

| | Applicable sentences | Successfully transformed sentences |
|---|---|---|
| Passivization: | 54 | 20 |
| Activization: | 26 | 11 |

**Table 1: Review of linguistics transformation success on the dataset of 1804 sentences from Reuters corpus.**

Refer to Figure 1 for the depiction of the currently tested baseline system. Table 2 shows an evaluation of this system without the watermarking step. As explained in Section 4.3, these scores are generated by systems that were specifically designed for evaluating machine translation systems, and they do not perfectly capture semantic resemblance of two sentences.

We would like to emphasize that the current system is limited by the capabilities of the parser and the surface realizer. XTAG may not be able to analyze a given sentence into its structural representation. Even though the XTAG parser is very powerful, it is not 100% accurate. Moreover, it has a limited coverage of vocabulary, and adding new words to its dictionary is not trivial, because every word in its dictionary is represented with several tree structures that conform to its usage in the language grammar. RealPro may not be able to generate an expected realization of a given deep syntactic structure in DSyntS format. RealPro is not designed for English to English translation, hence it has limitations when used for this purpose. For instance it can only handle a subset of uses of punctuation. Refer to RealPro General English Grammar User Manual [6] for further details on the capabilities and shortcomings of RealPro. A natural language watermarking system that has overcome these limitations will have more coverage while selecting sentences and performing embedding transformations on them. Therefore as the NLP systems improve, the watermarking system in this paper, will get more resilient and will provide higher bandwidth.

## 4.1 Sentence Level Linguistic Transformations

We have implemented transformation algorithms for two linguistic transformations: "activization" and "passivization". Their success rate is listed in Table 1. We marked the grammatical sentences in the output of the system as successfully transformed. We haven't excluded from this set, the grammatical output sentences, whose meaning have changed from their originals after the embedding transformation. One example is as follows:
**Original :** presidential elections must be held by october
**Transformed :** october had to hold presidential elections

In addition to these two transformations, if a sentence is analyzed using XTAG and then RealPro is used to generate a surface sentence from this analysis, this process may generate an output sentence that differs from the original sentence. In cases where such output sentences are grammatical, we observe that these sentences have gone through some syntactic transformations. Two of the transformations that occur consistently are special versions of "adjunct movement" and "topicalization".

Examples for the aforementioned transformations are given below, these sentences are taken from the data set introduced above, which is a subset of Reuters Corpus [17]:

**Passivization**
Original: this frank discussion will close this chapter
Transformed:
(i)this chapter, by this frank discussion, will be closed
(ii)this chapter will be closed by this frank discussion
**Activization**
Original: the devices were disrupted safely by the washington bomb squad
Transformed: the washington bomb squad safely disrupted the devices
**Adjunct Movement**
Original: now they are just attractive
Transformed: they are now just attractive
**Topicalization**
Original: doctors said he was alive but in critical condition
Transformed: he was alive but in critical condition doctors said

An example of two transformations that can be performed on the same sentence is as below:
**Original:** he said canada and britain recently rejected the idea
**After passivization:** he said the idea was recently rejected by canada and britain
**After adjunct movement:** he said the idea was rejected recently by canada and britain

## 4.2 Resilience Discussion

The reader may have observed that the transformations we use are typically reversible, i.e., the adversary can apply them wholesale everywhere. There two answers to this.

The first is that wholesale application of transformations (so as to "flatten" everything) has serious drawbacks: It is computationally expensive, it significantly changes the style of a document, and ambiguity can make it hard to automatically carry out (e.g., "A party must be held by April"). When embedding, we do not suffer the ambiguity drawback (because the initial "April had to hold a party" was not ambiguous), nor do we apply the process wholesale (we use the secret key to choose where to selectively apply it).

The second point is that the resilience of our scheme does not hinge on the non-reversibility of the transformation (e.g., passivization is easily reversible), rather, it relies on the fact that the adversary does not know the key-selected embedding features: The transformation is usually reversible in a multiplicity of ways (even if by trivial adjunct-movement), and the adversary does not know the impact of each of these ways on the secret embedding features (one of them neatly un-does the embedding action, but the adversary does not know which one). When a transformation is reversible in a unique way, we can either introduce multiplicity (e.g., by doing non-embedding transformations combined with the uniquely reversible embedding one), or we can combine the uniquely reversible embedding transformation with the robust synonym-substitution mechanism of [22] or with judicious (and ambiguity-increasing) removal of repeated information (a special and tractable case of co-references). For example, in the following dialog about the quality of a restaurant's food we can replace the *Bob*'s statement in to "Me too.":
**Original:**
John : I liked the food but I prefer my spouse's cooking. How about you?

Bob : I too prefer my spouse's cooking.
**Transformed:**
John : I liked the food but I prefer my spouse's cooking. How about you?
Bob : Me too.

## 4.3 Evaluation of Watermarking

The evaluation of natural language watermarking systems presents different challenges compared to the evaluation of audio, image or video watermarking systems. Even though recent progress in Machine Translation(MT) research addresses the quantification of *intelligibility* of machine translated text, MT evaluation systems fall short in the appropriate quantification of natural language watermarking quality, because changes we make that are meaning-preserving (and therefore acceptable) are scored poorly by existing MT evaluation systems: They measure such things as "distance of word positions from original position" and "matches / mismatches between words", rather than "difference in meaning". However, existing MT evaluation systems are quite valuable in measuring coverage of a watermarking system, defined as how applicable it is to various sentences (does it apply to a tiny fraction of sentences, or to most ?).

However, in order to be able to quantify the performance of the sentence level watermarking systems with a universally recognized evaluation metric, we decided to focus our baseline evaluation tests to check the success of our system in *re-generating* a sentence that is as close to the original as possible. This task can be achieved by using MT evaluation systems, since they are already based on checking the quality of the output of an MT system by comparing it to a high-quality reference translation. The results of this test measure the coverage of a natural language watermarking system.

We used the MT Evaluation Toolkit [4] of NIST [14] to evaluate the quality of the re-generated sentences in our system. This toolkit outputs scores for BLEU (BiLingual Evaluation Understudy) metric [16] and NIST metric [7].

BLEU computes the geometric mean of the variable length phrase matches (precision) against reference translations. The BLEU metric ranges from 0 to 1. Only the translations that are identical to a reference translation will attain 1. BLEU measures translation accuracy according to the phrase matches with one or more high quality reference translations. BLEU has been found to generally rank systems in the same order as human assessments.

In the same year as BLEU (2002), the NIST metric was introduced [7]. The NIST metric is a modified version of BLEU where the arithmetic mean of information weight of the variable length phrase matches are used, instead of arithmetic mean of N-gram precisions. For previous research on MT evaluation we refer the reader to [9].

Both the BLEU and the NIST metrics are sensitive to the number of reference translations. The more reference translations per sentence there are, the higher the BLEU and NIST scores are. Papineni et al. states that [16], on a test corpus of about 500 sentences (40 general news stories), a human translator scored 0.3468 against four references and scored 0.2571 against two references. However, in our tests we were not able to provide more than one reference text, namely the original. We tagged each sentence as a separate document due to the fact that our system is performing conversion at the sentence level.

Table 2 shows the evaluation results of the system shown in Figure 1. This system scores 0.47. According to the results of the NIST 2005 Machine Translation Evaluation (MT-05), the best score for BLEU 4-gram was achieved on "Arabic-to-English Task Unlimited Data Track" and it was 0.5137 [15].[5]

This scoring also contains the cases where the generated sentence is grammatically correct and carries the same meaning but the order of the words is not the same as in the original sentence. An example of such a case happens when *"doctors said he was alive but in critical condition."* goes through the system depicted in Figure 1, it is transformed to *"he was alive but in critical condition doctors said."*. This sentence translation scores 0.7260 with the BLEU 4-gram metric.

Using BLEU for sentence by sentence distance evaluation is neither sufficient nor accurate for the task of evaluating natural language watermarking. BLEU is very sensitive to precision in words and their position in the generated sentence. Some of the transformations (e.g., passivization) change the word order heavily while keeping the meaning very close to the original. A better way of evaluating the distortion made by a natural language watermarking system is measuring the distortion at the full text level. Such an evaluation can be done in several ways: (i) by counting the number of sentences changed, (ii) by assigning weights to different types of changes (i.e., transformations) to indicate the amount of the distortion they impose on the sentences (For example verb particle movement can get higher weight than removal of double-negation), (iii) by generating a language model of the author and measuring the change in the probability of a watermark carrying sentence (iv) using summarization to detect the change in similarity between the original document and watermarked document.

# 5. RELATED WORK

After early work on word-based hiding in natural language text [1], attempts at sentence-level natural language watermarking include [2, 3]. In these algorithms, selection of sentences that will carry the watermark information depends only on bit string that are derived from their corresponding tree structures. The nodes of the tree $T_i$ for sentence $s_i$ of text are labeled in pre-order traversal of $T_i$. Then, a node label $j$ is converted to 1 if $j + H(p)$ is a quadratic residue modulo $p$, and to 0 otherwise, where $p$ is a secret key and $H()$ is a one-way hash function. A node label sequence, $B_i$, is then generated by traversing $T_i$ according to post-order. A rank, $d_i$, is then derived for each sentence $s_i$ using $d_i = H(B_i) \,\mathrm{XOR}\, H(p)$ and the sentences are sorted by rank. Watermark insertion starts from the least-ranked sentence $s_j$, embedding is done to $s_j$'s successor in the cover text. The sentence $s_j$ is referred as a *marker* sentence, since it points to a watermark carrying sentence. Watermark insertion continues with the next sentence in the rank ordered list. Once the sentences to embed watermark bits are selected, the bits are embedded by applying *syntactic trans-*

[4]mteval-v11b.pl, release date: May 20th, 2004. Usually length of phrases range between unigram to 4*gram* for BLEU metric and unigram to 5*gram* for NIST metric. In the tables presented here the range is between 1 to 9.

[5]Mentioned evaluations are performed by May 20th 2004 release of MT Evaluation Kit by NIST [14]

| | | | Cumulative N-gram scoring | | |
|---|---|---|---|---|---|
| | 1-gram | 2-gram | 3-gram | 4-gram | 5-gram |
| NIST: | 7.7169 | 9.7635 | 10.0716 | 10.1172 | **10.1269** |
| BLEU: | 0.8548 | 0.6768 | 0.5580 | **0.4705** | 0.4030 |

**Table 2: The cumulative evaluation of performance of the presented system on direct conversion from English back into English sentences. 1804 sentences from Reuters corpus are used.**

*formations* in [2] and by applying *semantic transformations* in [3].

These studies were important first steps but (unlike the present paper) had the following drawbacks:

- They used only one feature of the sentence to both select and embed, thereby implying that a sentence could not do both (it was the sentence that comes immediately after a selected sentence that carried embedded information).
- The above-mentioned requirement for immediate proximity between a select-marked sentence and its corresponding message-carrying sentence, implies not only lower embedding capacity, but also an increased vulnerability to re-ordering of sentences, selection of a subset of sentences, as well as insertion of new sentences.
- The proximity was actually not the only (or even the main) source of such vulnerability in these previous schemes: A more serious one was the fact that a random change in any sentence had a probability of around $|M|/n$ of damaging an embedded bit. This is negligible only for very large texts ($n \gg |M|$).
- The previous work required fully automated semantic parsing and co-reference resolution, which current natural language processing technology does not satisfactorily provide (it is currently very domain-specific and hence not widely applicable).

Another work that deals with sentence level syntactic watermarking is by Brian Murphy[13]. This thesis presents the results of linguistic analysis of several sentence level syntactic transformations (including adjunct movement, adjective reordering, verb particle movement) on a hand parsed corpus of 6000 sentences [18]. This work provides the first detailed insight into applicability and coverage of several sentence level transformations for information hiding purposes. It provides a detailed analysis of the challenges that are involved in writing a generic transformation rule for a natural language. The number of transformations that were analyzed was limited due to the fact that transformations were performed without the use of a surface level generator, thus they mainly cover the transformations that re-orders the words in a sentence (e.g. adjunct movement) or adds a fixed structure to a sentence (e.g. clefting) or removes a fixed structure from the sentence(e.g. *that/who be* removal).

# 6. CONCLUSION AND FUTURE WORK

We have presented a generic information hiding algorithm that works on any cover document that consists of multiply featured data units. This algorithm is designed to overcome the challenges of low embedding bandwidth, small number of transformations that can not be applied to any given data unit, and there are only a limited number of alternatives that a data unit can be transformed into in order to embed information in it.

We have also presented and analyzed the application of this generic algorithm to sentence level watermarking, which is a novel and powerful technique, as it relies on multiple features of each sentence and exploits the notion of orthogonality between features. We verified the practicality of this technique on a prototype natural language watermarking system and presented the performance results on this baseline system tested on a data set of 1804 sentences.

As a future work, we will work on designing an evaluation system that handles the idiosyncrasies of natural language watermarking, as well as improving the implemented system to adjust to the limitations of the NLP tools used in the process. We will also increase the accuracy of transformations by adding a more informed dictionary to increase the coverage and to overcome the conversion mistakes such as "october had to hold presidential elections".

# 7. ACKNOWLEDGMENT

The authors would like to thank Giuseppe Riccardi from University of Trento, Dilek Hakkani-Tur from ICSI at University of Berkeley, Srinivas Bangalore from AT&T Labs; and Owen Rambow from Columbia University for helpful discussions. We are also grateful to the three anonymous referees for their helpful comments and suggestions.

# 8. REFERENCES

[1] M. Atallah, C. McDonough, S. Nirenburg, and V. Raskin. Natural Language Processing for Information Assurance and Security: An Overview and Implementations. In *Proceedings 9th ACM/SIGSAC New Security Paradigms Workshop*, pages 51–65, Cork, Ireland, September, 2000.

[2] M. Atallah, V. Raskin, M. C. Crogan, C. F. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Proceedings of the Fourth Information Hiding Workshop*, volume LNCS 2137, Pittsburgh, PA, 25-27 April 2001.

[3] M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg. Natural language watermarking and tamperproofing. In *Proceedings of the Fifth Information Hiding Workshop*, volume LNCS 2578, Noordwijkerhout, The Netherlands, 7-9 October 2002.

[4] R. Bergmair. Towards linguistic steganography: A systematic investigation of approaches, systems, and issues. Technical report, University of Derby, November, 2004.

[5] M. Chapman and G. Davida. Plausible deniability using automated linguistic stegonagraphy. In *roceedings of the International Conference on Infrastructure Security*, pages 276–287, Bristol, UK, October 1-3 2002.

[6] CogenTex. Realpro general english grammar user manual.

[7] G. Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of ARPA Workshop on Human Language Technology*, 2002.

[8] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[9] E. Hovy, M. King, and A. Popescu-Belis. Principles of context-based machine translation evaluation. *Machine Translation*, 16:1–33, 2002.

[10] D. Jurafsky and J. Martin. *Speech and Language Processing*. Prentice-Hall, Inc, Upper Saddle River, New Jersey, 2000.

[11] B. Lavoie and O. Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, DC, 1997.

[12] B. Levin. *English Verb Classes and Alternations: A preliminary investigation*. University of Chicago Press, Chicago, IL, 1993.

[13] B. Murphy. Syntactic information hiding in plain text. Master's thesis, CLCS, Trinity College Dublin, 2001.

[14] N. I. of Standards and Technology. Machine translations benchmark tests provided by national institute of standards and technology. In *http://www.nist.gov/speech/tests/mt/resources/*.

[15] N. I. of Standards and Technology. Nist 2005 machine translation evaluation official results, date of release :mon, aug 1, 2005, version 3. In *http://www.nist.gov/speech/tests/mt/*.

[16] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the ACL*, Philadelphia, July 2002.

[17] Reuters. Reuters corpus. In *http://about.reuters.com/researchandstandards/corpus/*.

[18] G. Sampson. *English for the computer: the SUSANNE corpus and analytic scheme*. Oxford: Clarendon, 1995.

[19] R. Sion, M. Atallah, and S. Prabhakar. On watermarking numeric sets. In *Proceedings of the Workshop on Digital Watermarking*, Seoul, Korea, 2002.

[20] M. Topkara, G. Riccardi, D. Hakkani-Tur, and M. J. Atallah. Natural language watermarking: Challenges in building a practical system. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, 2006.

[21] M. Topkara, C. M. Taskiran, and E. Delp. Natural language watermarking. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, 2005.

[22] U. Topkara, M. Topkara, and M. J. Atallah. The hiding virtues of ambiguity: Quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of ACM Multimedia and Security Workshop*, Geneva, Switzerland, September 26-27, 2006.

[23] XTAG, Research, and Group. A lexicalized tree adjoining grammar for english. Technical Report IRCS-01-03, IRCS, University of Pennsylvania, 2001.