Natural Language Watermarking : Challenges in Building a Practical System

Mercan Topkara[a] Giuseppe Riccardi[b] Dilek Hakkani-Tür[c] Mikhail J. Atallah[a]

[a] Center for Education and Research in Information Assurance (CERIAS), Purdue University,
West Lafayette, IN, 47907, United States

[b] Department of Information and Communication Technology, University of Trento,
38050 Povo di Trento, Italy

[c] AT&T Labs-Research, 180 Florham Park Ave, P.O. BOX 971, Florham Park,
NJ, 07932-0971, United States

ABSTRACT

This paper gives an overview of the research and implementation challenges we encountered in building an end-to-end natural language processing based watermarking system. With natural language watermarking, we mean embedding the watermark into a text document, using the 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. Of particular interest is using the structure of the sentences in natural language text in order to insert the watermark. We evaluated the quality of the watermarked text using an objective evaluation metric, the BLEU score. BLEU scoring is commonly used in the statistical machine translation community. Our current system prototype achieves 0.45 BLEU score on a scale [0,1].

1. INTRODUCTION

Natural language watermarking is required when there is a need for meta-data binding on the text documents in such a way that the text and the meta-data are not separable. This meta-data can be anything from the owner of the document to the original physical features of the document. Even though being able to search and access immense amount of knowledge online has become a part of life, the owner or the authors of this digital text do not have control on how their data is distributed or re-used. Full control on the distribution of digital text can be provided through the use of natural language watermarking. Section 4 further discusses applications of natural language watermarking to other security and privacy problems.

In audio or image watermarking the input signal $s(t)$ is processed to insert the watermark $w(t)$ via a function $\hat{s}(t) = F(s(t), w(t), k)$, where k is the secret key. The watermarked signal $\hat{s}(t)$ is such that the $w(t)$ becomes either visible/audible or retrievable by applying the function $G(\hat{s}(t), k)$. The function $F()$ has to be designed such that the modified signal is *perceptually* equivalent to the original signal. Natural language watermarking poses two research challenges in contrast to audio and image watermarking. First, there is successful experimental work on developing models for auditory and visual perception, whereas automatic semantic text analysis and evaluation is not that well developed. Recent progress in machine translation research have led to a first step to address the *adequacy* of machine translated text,[1] while other text features such as *coherence* and *fluency* are being studied. Second, the number of bits that can be used to *carry* the watermark on the natural language text is less than for the audio or image case. For example, the entropy is less than 2 bits (character level [2]) for the English language and it is less than 5 bits for standard images such as *Lena*, *Clown* *.[3] Attacks such as text *cropping* can further decrease the available bits to store the watermark. It is worth noting that for steganography some of these constraints can be weakened (e.g., by increasing arbitrarily the size of the cover text). See Section 5 for further discussion on research challenges of natural language watermarking.

[*] Letter entropy for English is computed with n-gram stationary modeling of the source, $n = 1, 2, ..10$. Image entropy is computed for a 512x512 image and 256 colors, using first order stationary model

Of natural language the combinatorial nature creates another challenge for the embedding process. Natural language has a combinatorial syntax and semantics, and the operations on natural language constituents (e.g., phrases, sentences, paragraphs) are sensitive to the syntactic/formal structure of representations defined by this combinatorial syntax.

The stealthiness requirements for natural language watermarking depend on the genre of the text, on the writer and on the reader characteristics. The common requirements can be summarized as follows:

Meaning The meaning of the text is its value, and it should be preserved through watermarking in order not to disturb the communication. Unless a human warden is concerned, this is not the case for steganography.

Fluency Fluency is required to represent the meaning of the text in a clear and readable way.

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.

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 un-watermarked works of the same author would be successful unless the style is preserved.[4]

In order to build a system that can perform fully automatic natural language watermarking while preserving the common requirements for stealthiness, we base our work on well-established research in the area of statistical natural language processing.[5] This is the first time all the vital components of a natural language watermarking system based on embedding the watermark in the linguistic features of sentences are put together and evaluated with a well known benchmark test. We have built a system that can convert raw sentences into an internal representation - syntactic/semantic parse tree - and can re-generate them back to surface level sentences. Further details about the system can be found in Section 3, Section 6 and Section 7.

2. PREVIOUS WORK

In 2000, the idea of using the semantics and syntax of the text for inserting the watermark was proposed[6] . In that work, binary encodings of the words were used for embedding information into text by performing lexical substitution in synonym sets.

In later work[7,8] Atallah et al. have proposed two algorithms that embed information in the tree structure of the text. The watermark is not directly embedded in the text, as is done in lexical substitution, but in the parsed representation of sentences. Utilizing the intermediate representation makes these algorithms more robust to attacks compared with lexical substitution systems. The difference between the two proposed algorithms in[7,8] is that the first one modifies syntactic parse trees of the cover text sentences for embedding while the second one uses semantic tree representations. Selection of sentences that will carry the watermark information depends only on the tree structure. Once the sentences to embed watermark bits are selected, the bits are stored by applying either *syntactic* or *semantic transformations*. Semantic transformations in that work was designed for preserving the meaning of the overall text, but not necessarily preserving the meaning of every individual sentence. More detailed discussion of these works can be found in.[9]

The above mentioned works were tested at the proof-of-concept level and were not evaluated on a large corpus. In this study, we are building an end-to-end natural language watermarking system -that integrates an English language parser and an English language generator- and evaluate its performance.

"Forced repatriation is against international conventions on refugees ."

Figure 1. A sample sentence taken from the Reuters Corpus.[10] Its publication day is 24th of August 1996.

```
( (S (NP (JJ forced) (NN repatriation))
     (VP (MD is)
      (PP (IN against)
       (NP (NP (JJ international) (NNS conventions))
        (PP (IN on) (NP (NNS refugees))))))
     (. .)))
```

Figure 2. Syntactically Annotated (parsed) Sentence - output of Charniak Parser

```
DSYNTS:
 against[ class:preposition ]
  ( I repatriation[ class:common_noun article:no-art number:sg ]
      (  ATTR forced[ class:adjective ])
    II convention[ class:common_noun article:no-art number:pl ]
      (  ATTR international[ class:common_noun article:no-art number:sg ]
         II on[ class:preposition ]
           (  II refugee[ class:common_noun article:no-art number:pl ]))
    III be[ class:verb number:sg person:3rd case:nom  tense:pres aspect:simple ])
END:
```

Figure 3. Deep Syntactic Structure (DSyntS) Format of RealPro

3. WATERMARKING AT SENTENCE LEVEL

The approach described in this document is based on syntactically modifying the sentence structure. In order to be able to automatically manipulate the sentences, we first use natural language parsers to get an internal representation of the sentence and later use natural language generators to revert the parse tree into raw sentence form. Surface level sentences are usually annotated with extra information in order to make them more useful for statistical Natural Language Processing (NLP). An example of such annotation is part-of-speech (POS) tagging where information about each word's part of speech (such as verb, noun, adjective) is added to the corpus in the form of tags. In NLP *parsing* is defined as processing input sentences and producing a data structure for them.[11] 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. Refer to Figure 2 and Figure 6 for two different versions of syntactic parsing of the sample sentence in Figure 1 with two different parsers, Charniak parser and XTAG parser[†] respectively.

The *natural language generation* (NLG) task is defined as the process of constructing natural language output from linguistic information, which is usually in the format of an internal information representation such as semantic parse tree or dependency tree, generated according to some communication specifications. Refer to Figure 3 for the Deep Syntactic Structure (DSyntS)[12] representation of the sentence in Figure 1. DSyntS format is used by RealPro[13] for sentence level surface realization.

In order to convert syntactic parse trees into DSynS, we need to convert phrase structures to dependency structures .[14,15] Refer to Figure 4 and Figure 7 for two different versions of dependency trees for the sample sentence in Figure 1. Both of the dependency trees encode the tree depicted in Figure 5.

In our implemented system, syntax based linguistic transformations are used in order to preserve the meaning and the grammaticality of the cover text. Fluency is preserved by applying the transformations at the sentence level. Since the meaning of the sentence is preserved through this process, the flow of information delivery in the text will stay the same. We are still exploring the ways for preserving the style. The readers are referred to Section 5 for further discussions.

[†] We excluded the feature tags of the words from the XTAG parse for readability issues. Refer to Appendix 11 for the list of features of XTAG parse output features.

$ChildIndex$	$Child$	$POSTag_{child}$	$ParentIndex$	$Parent$	$POSTag_{parent}$
0	forced	JJ	2	repatriation	NN
1	repatriation	NN	2	against	IN
2	is	MD	2	against	IN
3	against	IN	1	**TOP**	-
4	international	JJ	2	conventions	NNS
5	conventions	NNS	1	against	IN
6	on	IN	2	conventions	NNS
7	refugees	NNS	1	on	IN
8	.	.	-1	-	-

Figure 4. Sentence Dependency Structure produced by LEXTRACT using output of Charniak Parser

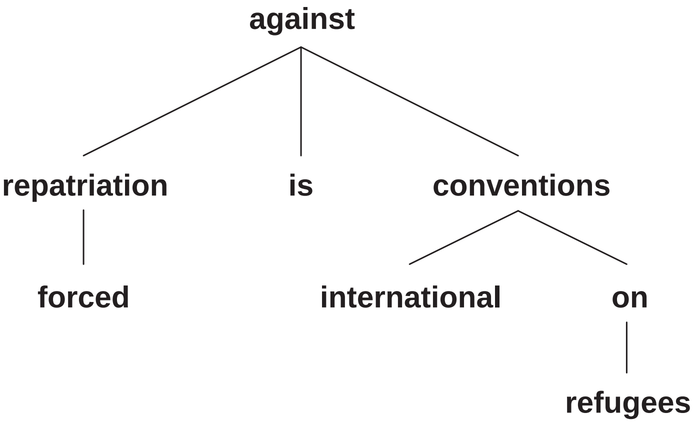

Figure 5. Dependency tree for the sentence in Figure 1.

3.1. Sentence Based Linguistic Transformations

Synonym substitution is the most widely used linguistic transformation for information hiding systems since it is the simplest transformation. Synonym substitution has to take the sense of the word into consideration. In order to preserve the meaning of the sentence the word should be substituted with a synonym in the same sense. For example the word “bank” has at least three different senses as a financial institution, a river edge, or something to sit on. An electronic dictionary like Wordnet that classifies all words and phrases into synonym sets can be used to search for words that are synonyms for a given word. However, determining the correct sense of a given word, referred to as the word sense disambiguation task in NLP, may present hard problems since it is hard to even derive a general definition for word sense.[16]

Our main focus is on applying *syntactic transformations*, such as passivization and clefting, which change the syntactic structure of a sentence with little effect on its meaning. Some of the common syntactic transformations in English are listed in Table 1. In addition to these, there is another group of syntactic transformations that are solely based on the categorization of the main verb of the sentence. Verbs can be classified according to shared meaning and behavior, and different classes of verbs allow different transformations to be performed in the sentence.[17] Examples of a transformation known as the locative alternation are given below.

```
Jack sprayed paint on the wall.          ⇒  Jack sprayed the wall with paint.
Henry cleared the dishes from the table.  ⇒  Henry cleared the table of the dishes.
```

4. APPLICATIONS

Natural language watermarking provides a way to hide meta-data information into the cover text in a way that it is not easy to separate the meta-data from the text without changing the semantic form or syntactic form of

Transformation	Original sentence		Transformed sentence
Passivization	The dog kissed the big boy.	⇒	The big boy was kissed by the dog.
Topicalization	I like bagels.	⇒	Bagels, I like.
Clefting	He bought a brand new car.	⇒	It was a brand new car that he bought.
Extraposition	To believe that is difficult.	⇒	It is difficult to believe that.
Preposing	I like big bowls of beans.	⇒	Big bowls of beans are what I like.
There-construction	A unicorn is in the garden.	⇒	There is a unicorn in the garden.
Pronominalization	I put the letter in the mailbox.	⇒	I put it there.
Fronting	"What!" Alice cried.	⇒	"What!" cried Alice.

Table 1. Some common syntactic transformations in English.

the sentences in the text.

In addition to meta-data binding for copyright protection, natural language watermarking provides an integrity checking method that "travels with the content" whenever the content is (mis)used, which makes it very valuable for applications that involve private communication. For example, phishing - a web based deception technique - exploits the fact that customers of online services trust e-mail communication. This attack is successful partially due to the fact that secure e-mail systems are not commonly employed by the non-tech-savvy users. There is a great need for binding the source information to the documents involved in private communication. See[18] for a system that uses watermarking against phishing.

Another relevant problem is enforcing security policies on private communications. An example of such a system would be e-mail communications that involve groups of people where each of the participants has a different level of access control rights. In such systems, unless the security level is bound to the text content, there is no possibility of enforcing security policies automatically when the encryption or digital signature is separated from the document (whereas a watermark inherently "travels with the content"). In addition, robust natural language watermarking algorithms will enable a wide range of applications such as text auditing, tamper-proofing, and traitor tracing.

As a future research topic, we will work on watermarking ensembles of text documents collectively. This would create new application areas that concern the access control policies for digital libraries.

5. CHALLENGES IN NATURAL LANGUAGE WATERMARKING

NLP addresses the problem of parsing, understanding and generating natural language automatically.[5] For the purposes of text watermarking, NLP techniques can be used to modify the input text and generate a new text with the same meaning, where the watermark information is hidden in the structure of the new text. This requires the natural language analysis tools to provide a robust representation of the syntactic/semantic structure of the sentences of the text, applicable to large data sets from different domains. One challenge for natural language analysis is the problem of ambiguity, which results in multiple interpretations. The following are commonly used examples of ambiguity at the lexical, syntactic and semantic levels, respectively:

- *I can can the can.* (The part of speech tag of each occurrence of the word *can* is different)
- *I saw a woman with a telescope.* (The noun phrase can be attached to both the noun phrase or the verb phrase in the syntactic structure)
- *Iraqi head seeks arms.* (The word *head* can be interpreted as 'chief' or 'body part' and *arms* can be interpreted as 'weapons' or 'body parts', respectively.

Large amount of data/knowledge is required to be able to build models that disambiguate generic natural language sentences.

The ultimate goal of NLP systems is to be able to process any natural language sentence. However, as a result of ambiguity and data coverage, the state-of-the-art tools' accuracies vary widely with the type of analysis.

Parser	Input Format	Output Format	Accuracy
Charniak, 2000	Raw sentence	Word level parse in PennTreebank Format	90.1%
XTAG, 2001	Raw sentence	Word level parse in Tree-Adjoining Grammar Format	87.7%

Table 2. Properties of syntactic parsers used in these experiments.

For example, for part-of-speech tagging of English, the best accuracy is around 98% for a given domain,[19] for syntactic parsers the accuracy based on labeled precision and recall is around 91%.[20] In previous work, we surveyed the state-of-the-art NLP tools and data resources, see[9] for further information.

Even if we get the most out of existing NLP tools, the evaluation of natural watermarking algorithms present unique and difficult challenges. Natural language watermarking requires evaluation systems for the stealthiness of the embedding. This requires systems that can evaluate the grammaticality and the fluency of the generated text. Most of the state of the art natural language evaluation tools that were developed for evaluating the grammar and fluency of machine translation systems may be adapted to evaluate watermarking systems up to a level. For previous research on this topic, refer to.[21]

Preserving the style of a document depends on being able to automatically evaluate the characteristics of a writer's style. Length of sentences and paragraphs, or usage of clauses or percentage of the passive sentences can be counted as style characteristics. See[4] for a discussion of using style and expression for identifying linguistic similarity. If we can quantifiably evaluate writer's characteristics, we can use an on-the-fly damage control system in order to minimize the deviation from them. In a previous work,[22] we presented a new protocol that works in conjunction with the information hiding algorithm to systematically improve the stealthiness.

6. EXPERIMENTAL SETUP

There are several different NLP tools involved in the process of a natural language based information hiding system. It is a non-trivial task to integrate several NLP systems to build a system that takes raw sentence as input and after processing the sentence reverts it back to the raw sentence (surface) level. To the best of authors' knowledge this is the first work towards building such a system[‡]. In this paper, we present the results of the baseline coverage and quality tests performed on a system that can parse a sentence syntactically, process the parser output and re-generate the sentence back to surface level using a natural language generation tool.

Natural language parsers are required to get the syntactic/semantic tree structure of the sentences with the lexical features extracted, and natural language generators are required in order to convert the transformed structures back to the natural language text. The conversion of parser outputs to natural language generation inputs without information loss is crucial for automating the embedding.

We decided to apply our watermarking algorithm to the English language due to the fact that it is the most studied language in natural language processing research. It is very easy to access highly accurate off-the-shelf language analysis tools and very rich data resources for English.

Data Resources We tested our system on 683 sentences from the Reuters corpus.[10] We picked three publication days at random[§]. Later, from the articles that were published on these days, we picked the sentences that can be parsed in less than 30 seconds with the XTAG parser. We are also using Wordnet[23] 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.

[‡] Natural language paraphrasing systems are under another category where paraphrasing capabilities of the system depends on the templates and rules learned from a training corpus. See[9] for more information on paraphrasing

[§] 24th of August 1996, 20th of October 1996 and 19th of August 1997

```
( S_r ( S_f ( NP ( N_r ( A forced )
                        ( N_f repatriation ) ) )
            ( S_r ( NP  )
                  ( VP_r ( V is )
                         ( VP ( V v )
                              ( PP_1 ( P against )
                                     ( NP_r ( NP_f ( N_r ( N international )
                                                         ( N_f conventions ) ) )
                                            ( PP ( P on )
                                                 ( NP ( N refugees ) ) ) ) ) ) ) ) )
      ( Punct . ) )
```

Figure 6. Syntactically Annotated (parsed) Sentence - output of XTAG Parser

(alphaW0nx0Pnx1[against] (alphaNXN[repatriation]<NP_0> betaAn[forced]<N>) (alphaNXN[conventions]<NP_1> betaNn[international]<N> (betanxPnx[on]<NP> alphaNXN[refugees]<NP>)) betaVvx[is]<VP>)

Figure 7. Sentence Dependency Structure, output of XTAG. See Figure 5 for a depiction of this tree.

Parsers Our development uses *XTAG* parser [¶][24] and Charniak parsers [‖][25] for parsing, dependency tree generation and feature extraction. We used *LEXTRACT* [**][14] for converting phrase structures to dependency structures.

Generator We used *RealPro*[††][13] for natural language generation.

Experimental System Schema Refer to Figure 8 and Figure 9 for the depictions of the currently tested baseline systems. Natural Language WaterMarking (NLWM) System I is simpler than NLWM System II in the sense that it uses only the output of XTAG parser. NLWM System I replaces the dependency tree required by DSyntS generation with the derivation tree output of the XTAG parser. We have built NLWM System II in order to increase the information going into the conversion step, and to benefit from the different strengths of the two parsers. XTAG parser's output is very rich in the sense that it includes a large feature set for each word. A large set of features is required both for robust watermark embedding and for the accurate conversion of the parser's output to the generator's input. On the other hand, Charniak parser is more accurate than XTAG parser. Having more accurate dependency trees leads to better generation quality.

7. RESULTS AND THEIR QUANTITATIVE EVALUATION

The evaluation of natural language watermarking systems present unique and difficult challenges compared to the evaluation of audio, image or video watermarking systems. Automatic semantic text analysis is not as developed as the automatic analysis of audio and visual documents. Even though recent progress in Machine Translation(MT) research have led to a first step to address the *adequacy* of machine translated text, evaluating other text features such as *coherence* and *fluency* are being studied.

Due to these limitations, we decided to focus our evaluation tests to check the success of our system in *re-generating* a sentence that is as close to the original as possible. This can be achieved by using MT evaluation systems, since they are already based on checking the quality of the output of a MT system by comparing it to a reference -high quality- translation. We used the MT Evaluation Tool Kit [‡‡] of NIST (available at[26]) to

[¶] Freely available at http://www.cis.upenn.edu/ xtag/swrelease.html. In our experiments, we used *lem*0.14.0.*i*686.*tgz*

[‖] Freely available at ftp://ftp.cs.brown.edu/pub/nlparser/. We used *charniak_parser*03.*tar.gz*.

[**] We acquired a copy of LEXTRACT system directly from Fei Xia, http://faculty.washington.edu/fxia/.

[††] See http://www.cogentex.com/technology/realpro/index.shtml for access to the software.

[‡‡] 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.

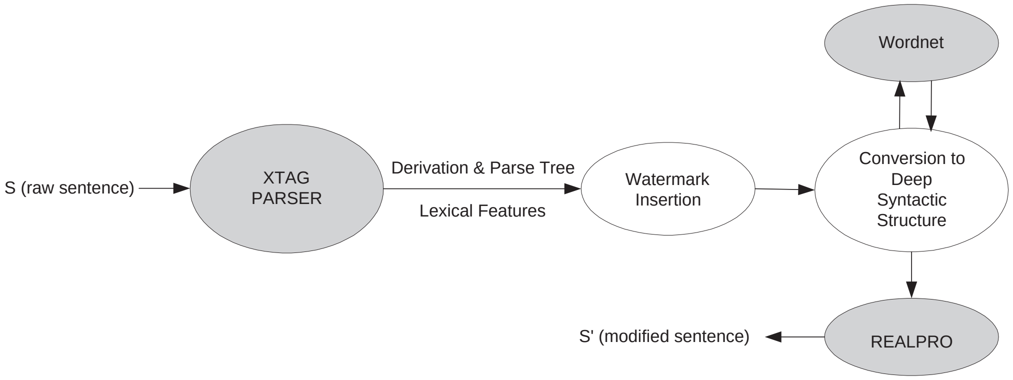

Figure 8. NLWM System I: A schema of the system that is being developed and designed for the baseline evaluations of the NLWM system. This implementation generates parse and derivation trees with $XTAG$[24] parser and uses *RealPro*[13] for surface realization.

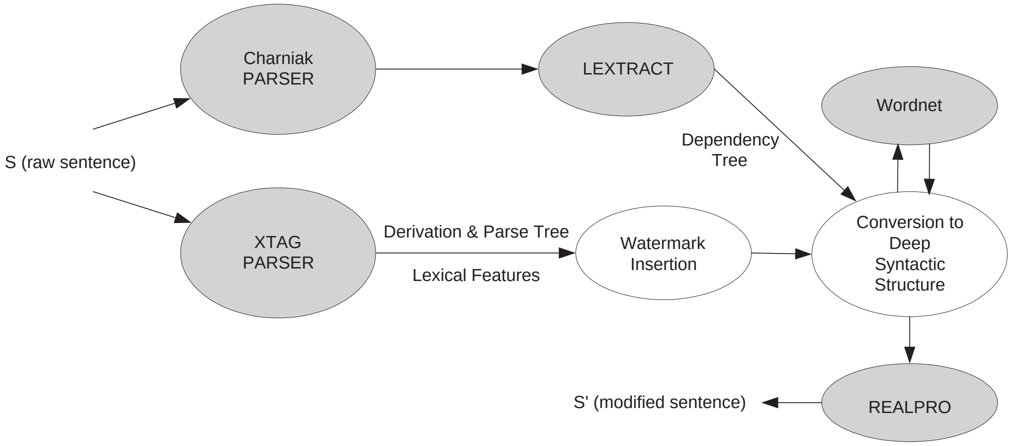

Figure 9. NLWM System II: A schema of the system that is being developed and designed for the evaluations of the NLWM system. It generates the parse tree with $XTAG$[24] parser. It generates the dependency tree with the Charniak parser[25] and *lextract*.[14] Later *RealPro*[13] is used for surface realization.

			Cumulative N-gram scoring		
	1-gram	2-gram	3-gram	4-gram	5-gram
NIST:	7.3452	9.0238	9.2225	9.2505	**9.2536**
BLEU:	0.8511	0.6694	0.5448	**0.4532**	0.3821

Table 3. The Evaluation of System I, based on the latest release of MT Evaluation Kit by NIST (May 20th 2004) [26] According to the results of NIST 2005 Machine Translation Evaluation (MT-05), best score for BLEU 4 gram was achieved on "Arabic-to-English Task Unlimited Data Track" and it was 0.5137.[28]

evaluate the quality of the re-generated sentences in our system. This toolkit outputs scores for BLEU (BiLingual Evaluation Understudy) metric[1] and NIST metric.[27]

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 with BLEU, in 2002, the NIST metric was introduced.[27] 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 refer to.[21]

Both BLEU and NIST metrics are sensitive to the number of reference translations. The more reference translations per sentence there are, the higher the BLEU and NIST score are. Papineni et al. states that,[1] 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 translation. We used the original sentences as the reference translation, since our quality test was based on *re-generating* a sentence that is as close to the original as possible. We tagged each sentence as a separate document due to the fact that our system is performing conversion at the sentence level.

Table 3 and Table 4 show the evaluation results for NLWM System I and NLWM System II respectively. These are the lower bounds to these systems' accuracy. NLWM System I scores 0.4532. This score also contains the cases where the generated sentence is grammatically correct and carries the same meaning but the order of the words are not same with the original sentence. An example of such a case happens when *"doctors said he was alive but in critical condition."* goes through NLWM System I, it turns to *"he was alive but in critical condition doctors said."*. This sentence translation scores 0.7260 with the BLEU 4-gram metric.

Even though the NLWM System II uses more information, its score of 0.2439 is lower than that of the NLWM System I. NLWM System II suffers from the fact that the combination algorithm for the outputs of two different parsers is very simple: Currently, feature and POS tag information is taken from the *XTAG parser* output and it is loaded to the dependency tree generated by *Charniak parser* and *LEXTRACT*. As a future work, we are planning to improve the combination algorithm for this system. We are also limited by the capabilities of the parser and the surface realizer. Due to the fact that RealPro is not designed for English to English translation, it has a limited expression power. A few examples of such limitations are handling of punctuation or adjuncts. Refer to RealPro General English Grammar User Manual[12] for further details on the capabilities of RealPro. An NLWM system with a full coverage will be more flexible while selecting sentences and performing embedding transformations.

8. FUTURE WORK

As the next step, we will integrate the remaining components of the watermarking algorithm to NLWM System I and NLWM System II, while improving the quality and the coverage as needed.

In addition, we will work on improving the evaluation mechanism by exploring specific evaluation methodologies for natural language watermarking systems. Such an evaluation system would be capable of checking the effect of the watermarking process on the meaning, the fluency, the grammaticality, and the style of the cover

			Cumulative N-gram scoring		
	1-gram	2-gram	3-gram	4-gram	5-gram
NIST:	6.2987	7.3787	7.4909	7.4962	**7.4965**
BLEU:	0.7693	0.5096	0.3484	**0.2439**	0.1724

Table 4. The Evaluation of System II, based on the latest release of MT Evaluation Kit by NIST (May 20th 2004) [26] According to the results of NIST 2005 Machine Translation Evaluation (MT-05), best score for BLEU 4 gram was achieved on "Arabic-to-English Task Unlimited Data Track" and it was 0.5137.[28]

text. After building a system that can qualify the change in these characteristics of the cover text, we will use an on-the-fly (damage) control protocol[22] to improve the quality and stealthiness of the watermarking algorithm. Current MT evaluation systems are helpful only in checking how close the phrases in the newly generated sentence is to the phrases in the original sentence due to the fact that MT evaluation is based on comparing the MT system's output to the reference -high quality- translations.

Another future research topic is the analysis of using larger natural language components such as paragraphs or full text structure as information carriers. All of the information hiding systems developed so far perform embedding either at the word (synonym substitution) or the sentence level. Digital libraries contain collections of several documents, and such ensembles of text should be collectively watermarked. The carrier of collective watermarking systems would be any component that lowers the value (either form or meaning) of the document ensemble.

9. CONCLUSION

In this paper, we gave an overview of the research and implementation challenges we encountered in building an end-to-end natural language processing based watermarking system. To the best of authors' knowledge this is the first time, an English language parser and an English language generator is integrated into a working system that takes raw sentence as input and after processing the sentence reverts it back to the surface level. We have performed tests on two different systems. The first one uses only the *XTAG Parser* for parsing and *RealPro* for generation. This system achieves 0.4532 BLEU 4 score in NIST benchmark evaluation tests. The second system uses *XTAG Parser* and *Charniak Parser* for parsing, *LEXTRACT* for building the dependency tree from *Charniak Parser's* output and *RealPro* for generation. This second system achieves 0.2439 BLEU 4 score in NIST benchmark evaluation tests.

We have also discussed the general requirements of a natural language watermarking system such as preserving meaning, fluency, grammaticality and style. There are many research challenges involved in building a system that can fulfill these requirements.

In addition, we have listed applications of natural language watermarking to mitigating security and privacy requirements of information exchange based on text. Natural language watermarking provides an integrity checking method that "travels with the content" whenever the content is (mis)used, which makes it valuable for applications that involve private communications.

10. ACKNOWLEDGMENTS

The authors would like to thank Umut Topkara, Prof. Edward J.Delp, Prof. Victor Raskin from Purdue University; Cuneyt Taskiran from Motorola; Srinivas Bangalore from AT&T Labs − Research; and Owen Rambow from Columbia University for helpful discussions and feedback.

REFERENCES

1. K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," *Proceedings of 40th Annual Meeting of the ACL*, July 2002, Philedelphia.
2. G. Potamianos and F. Jelinek, "A study of n-gram and decision tree letter language modeling methods," *Speech Commun.*, vol. 24, no. 3, pp. 171–192, 1998.

3. G. A. Mian, "Personal communication," 2005.
4. O. Uzuner and B. Katz, "Style vs. expression in literary narratives," *Proceedings of the Twenty-eighth Annual International ACM SIGIR Conference Workshop on Using Stylistic Analysis of Text for Information Access*, 2005.
5. C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* The MIT Press, 1999.
6. M. Atallah, C. McDonough, S. Nirenburg, and V. Raskin, "Natural Language Processing for Information Assurance and Security: An Overview and Implementations," *Proceedings 9th ACM/SIGSAC New Security Paradigms Workshop*, September, 2000, Cork, Ireland, pp. 51–65.
7. 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," *Fourth Information Hiding Workshop*, vol. LNCS, 2137, April, 2001, Pittsburgh, Pennsylvania, Springer-Verlag.
8. M. Atallah, V. Raskin, C. F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, "Natural Language Watermarking and Tamperproofing," *Fifth Information Hiding Workshop*, vol. LNCS, 2578, October, 2002, Noordwijkerhout, The Netherlands, Springer-Verlag.
9. M. Topkara, C. M. Taskiran, and E. Delp, "Natural language watermarking," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VII*, 2005.
10. "Reuters corpus," *http://about.reuters.com/researchandstandards/corpus/index.asp.*
11. D. Jurafsky and J. Martin, *Speech and Language Processing.* Upper Saddle River, New Jersey: Prentice-Hall, Inc, 2000.
12. "Realpro general english grammar user manual," *http://www.cogentex.com/papers/realpro-manual.pdf.*
13. B. Lavoie and O. Rambow, "A fast and portable realizer for text generation systems," *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997, Washington, DC.
14. F. Xia and M. Palmer, "Converting dependency structures to phrase structures," *Proceedings of the Human Language Technology Conference*, 2001.
15. C. Han, B. Lavoie, M. Palmer, O. Rambow, R. I. Kittredge, T. Korelsky, N. Kim, and M. Kim, "Handling stuctural divergences and recovering dropped arguments in a korean/english machine translation system," *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas*, 2000.
16. N. Ide and J. Vronis, "Word sense disambiguation: The current state of the art," *Computational Linguistics*, vol. 24, no. 1, 1998.
17. B. Levin, *English Verb Classes and Alternations: A preliminary investigation.* Chicago, IL: University of Chicago Press, 1993.
18. M. Topkara, A. Kamra, M. Atallah, and C. Nita-Rotaru, "Viwid: Visible watermarking based defense against phishing," *International Workshop on Digital Watermarking*, 15–17 September 2005, Siena, Italy.
19. H. Halteren, W. Daelemans, and J. Zavrel, "Improving accuracy in word class tagging through the combination of machine learning systems," *Coling*, vol. 27, no. 2, pp. 199–229, 2001.
20. M. Collins, "Head-driven statistical models for natural language parsing," *Coling*, vol. 29, Issue 4, 2003.
21. E. Hovy, M. King, and A. Popescu-Belis, "Principles of context-based machine translation evaluation," *Machine Translation*, vol. 16, pp. 1–33, 2002.
22. M. Topkara, U. Topkara, C. Taskiran, E. Lin, M. Atallah, and E. Delp, "A hierarchical protocol for increasing the stealthiness of steganographic methods," *Proceedings of the ACM Multimedia and Security Workshop*, 2004.
23. C. Fellbaum, *WordNet an electronic lexical database.* MIT Press, 1998.
24. XTAG, Research, and Group, "A lexicalized tree adjoining grammar for english," Tech. Rep. IRCS-01-03, IRCS, University of Pennsylvania, 2001.
25. E. Charniak, "A maximum-entropy-inspired parser," *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2000.
26. "Machine translations benchmark tests provided by national institute of standards and technology," *http://www.nist.gov/speech/tests/mt/resources/scoring.htm.*
27. G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," *Proceedings of ARPA Workshop on Human Language Technology*, 2002.
28. "Nist 2005 machine translation evaluation official results, date of release :mon, aug 1, 2005, version 3," *http://www.nist.gov/speech/tests/mt/mt05eval_official_results_release_20050801_v3.html.*

11. APPENDIX A : LIST OF XTAG PARSE OUTPUT FEATURES AND THEIR POSSIBLE VALUES

Feature	Value
<agr 3rdsing>	+,-
<agr num>	plur,sing
<agr pers>	1,2,3
<agr gen>	fem,masc,neuter
<assign-case>	nom,acc,none
<assign-comp>	that,whether,if,for,ecm,rel,inf_nil,ind_nil,ppart_nil,none
<card>	+,-
<case>	nom,acc,gen,none
<comp>	that,whether,if,for,rel,inf_nil,ind_nil,nil
<compar>	+,-
<compl>	+,-
<conditional>	+,-
<conj>	and,or,but,comma,scolon,to,disc,nil
<const>	+,-
<contr>	+,-
<control>	no value, indexing only
<decreas>	+,-
<definite>	+,-
<displ-const>	+,-
<equiv>	+,-
<extracted>	+,-
<gen>	+,-
<gerund>	+,-
<inv>	+,-
<invlink>	no value, indexing only
<irrealis>	+,-
<mainv>	+,-
<mode>	base,ger,ind,inf,imp,nom,ppart,prep,sbjunt
<neg>	+,-
<passive>	+,-
<perfect>	+,-
<pred>	+,-
<progressive>	+,-
<pron>	+,-
<punct bal>	dquote,squote,paren,nil
<punct contains colon>	+,-
<punct contains dash>	+,-
<punct contains dquote>	+,-
<punct contains scolon>	+,-
<punct contains squote>	+,-
<punct struct>	comma,dash,colon,scolon,nil
<punct term>	per,qmark,excl,nil
<quan>	+,-
<refl>	+,-
<rel-clause>	+,-
<rel-pron>	ppart,ger,adj-clause
<select-mode>	ind,inf,ppart,ger
<super>	+,-
<tense>	pres,past
<trace>	no value, indexing only
<weak>	+,-
<wh>	+,-