CERIAS Tech Report 2006-61

The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym

by Mikhail J. Atallah
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym Substitutions

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

ABSTRACT

Information-hiding in natural language text has mainly consisted of carrying out approximately meaning-preserving modifications on the given cover text until it encodes the intended mark. A major technique for doing so has been synonym-substitution. In these previous schemes, synonym substitutions were done until the text "confessed", i.e., carried the intended mark message. We propose here a better way to use synonym substitution, one that is no longer entirely guided by the mark-insertion process: It is also guided by a resilience requirement, subject to a maximum allowed distortion constraint. Previous schemes for information hiding in natural language text did not use numeric quantification of the distortions introduced by transformations, they mainly used heuristic measures of quality based on conformity to a language model (and not in reference to the original cover text). When there are many alternatives to carry out a substitution on a word, we prioritize these alternatives according to a quantitative resilience criterion and use them in that order. In a nutshell, we favor the more ambiguous alternatives. In fact not only do we attempt to achieve the maximum ambiguity, but we want to simultaneously be as close as possible to the above-mentioned distortion limit, as that prevents the adversary from doing further transformations without exceeding the damage threshold; that is, we continue to modify the document even after the text has "confessed" to the mark, for the dual purpose of maximizing ambiguity while deliberately getting as close as possible to the distortion limit. The quantification we use makes possible an application of the existing information-theoretic framework, to the natural language domain, which has unique challenges not present in the image or audio domains. The resilience stems from both (i) the fact that the adversary does not know *where* the changes were made, and (ii) the fact that automated disambiguation is a major difficulty faced by any natural language processing system (what is bad news for the natural language processing area, is good news for our scheme's resilience). In addition to the above-mentioned design and analysis, another contribution of this paper is the description of the implementation of the scheme and of the experimental data obtained.

Categories and Subject Descriptors

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

General Terms

Security, Design

Keywords

Information Hiding, Natural Language Text, Homograph, Synonym Substitution

1. INTRODUCTION

In recent years, there has been an increased interest in using linguistic techniques for designing information hiding systems for natural language text. These techniques are based on using the knowledge of language to generate or rewrite a document in order to encode hidden information [25].

Even though there is a growing interest in information hiding into natural language, there has not been much movement in the direction of quantification that makes possible using the considerable theoretical work on the analysis of the communication channel established by information hiding. To avail oneself of the information hiding model proposed by Moulin et al in [14] requires quantification of the distortion effect of each linguistic transformation. In this paper we carry out such an analysis, using a natural language watermarking system based on a novel twist on the old idea of synonym substitution. Section 2.1 will discuss how we use the existing information hiding model for the natural language domain.

*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.

MM&Sec'06, September 26–27, 2006, Geneva, Switzerland.

Publicly available methods for information hiding into natural language text can be grouped under two branches. The first group of methods are based on generating a new text document for a given message. Spammimic [8] 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, where there is also a need for robustness 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 information hiding into natural language, refer to Section 5.

The watermarking system proposed in this paper is based on improving resilience of synonym substitution based embedding by ranking the alternatives for substitution according to their ambiguity and picking the one that has maximum ambiguity within the synonyms (subject to not exceeding the maximum cumulative distortion limit). The encoding is designed in a way that the decoding process does not require the original text or any word sense disambiguation in order to recover the hidden message. This system follows the Kerckhoff's rule, namely, that the decoding process depends only on the knowledge of the secret key and public domain information (no "security through obscurity").

It is possible to determine infringement of copyright using simple string matching if the infringement is in the form of verbatim copying of the text. However the adversary can foil the string matching based infringement detection, through automated meaning preserving changes to the text [25]. A desired natural language watermark should be resilient against these modifications. Refer to Section 2.2 for more discussion on the model of adversary.

The detection of copyright infringements on web publishing could be one of the major applications of natural language watermarking. In this application the copyright holder will be able to find out infringements by running a web crawler that detects the copyright holder's watermark; or by subscribing to a web crawler service that searches for watermarked text on the web. In order to realize this, it is crucial that the watermark detection can be performed automatically without human intervention. This requirement is satisfied by the system introduced in this paper.

In case a news agency *does not watermark* its news articles, but uses a web crawler to search for the illegal copies of the articles on the internet. An adversary, who wants to re-publish an article from this agency, can perform synonym substitution to deceive a string matching based web crawler. The infringement detection can be performed by checking whether the words in the suspicious copy and the original document are synonyms.

The details of the proposed watermarking system are explained in Section 3, followed by experimental results in Section 4.

Even though we have focused our attention directly on synonym substitution based watermarking, the analysis and discussions made in this paper shed light on the information theoretic analysis of other systems that achieve information hiding through approximately meaning-preserving modifications on a given cover text.

2. FRAMEWORK

This section discusses the general framework we use, including our model of the adversary. Where appropriate, we explain how the peculiarities of the natural language application domain pertain to the framework.

2.1 Review of Distortion Quantification

Here we briefly review the general model proposed by Moulin et al in [14] and use the same notation, as applicable. The later section on experimental results (Section 4) will explain how we computed the values for the below equations. In this notation, random variables are denoted by capital letters (e.g. S), and their individual values are donated by lower case letters (e.g. s). The domains over which random variables are defined are denoted by script letter (e.g. $\mathcal{S}$). Sequences of N random variables are denoted with a superscript N (e.g. $S^N = (S_1, S_2, ..., S_N)$).

Natural language watermarking systems aim to encode a watermark message, M, into a given source document, S^N, using a shared secret, K^N, where K^N is the only side information shared between the encoding and decoding processes. The goal of the encoding process is to maximize the robustness of watermark against possible attacks while keeping the distortion inflicted on the S^N during watermarking within allowable limits. There are two distortion constraints on a given natural language watermarking system.

The first distortion constraint is introduced to capture the fact that the watermark encoding process, $f_N : \mathcal{S}^N \times \mathcal{M} \times \mathcal{K}^N \rightarrow \mathcal{X}^N$, has to preserve the "value" of the source document, while creating the watermarked document X^N. Moulin et al formalizes this constraint as below:

$$\sum_{s^N \in \mathcal{S}^N} \sum_{k^N \in \mathcal{K}^N} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{M}|} p(s^N, k^N) d_1^N(s^N, f_N(s^N, m, k^N)) \leq D_1 \tag{1}$$

where p is the joint probability mass function and d_1 is a nonnegative distortion function defined as $d_1 : \mathcal{S} \times \mathcal{X} \rightarrow \mathbb{R}_+$. The distortion functions d_i [1] are extended to per-symbol distortion on N-tuples by $d_i^N(s^N, x^N) = \frac{1}{N} \sum_{k=1}^{N} d_i(s_k, x_k)$.

The second constraint denotes the maximum distortion an adversary can introduce on the modified document, Y^N, without damaging the document's "value" for the adversary. The constraint on the attack channel for all $N \geq 1$ is formalized as below:

$$\sum_{x^N \in \mathcal{X}^N} \sum_{y^N \in \mathcal{Y}} d_2^N(x^N, y^N) A^N(y^N | x^N) p(x^N) \leq D_2 \tag{2}$$

where $A^N(y^N|x^N)$ is a conditional probability mass function that models an adversary who maps $\mathcal{X}^\mathcal{N}$ to $\mathcal{Y}^\mathcal{N}$, and d_2 is the adversary's distortion function (similar to d_1). The decoder process receives Y^N.

For image, video or numeric databases, the space can be modeled as a Euclidean space and the effect of changes on the objects can be quantified as a continuous function [20, 14]. However, it is rather hard to model the natural language text input. The value of a natural language document is based on several properties such as meaning, grammaticality and style. Thus, the distortion function should be designed to measure the distortion in these properties.

In fact we cannot even talk of a distance in natural language processing, as the triangle inequality need not be satisfied. For example, both "lead" and "blend" are synonyms

[1] $i \in 1, 2$

of different senses of the word "go", as the following entries (obtained from WordNet) indicate:

- blend, go, blend in – (blend or harmonize; "This flavor will blend with those in your dish"; "This sofa won' t go with the chairs")
- go, lead – (lead, extend, or afford access; "This door goes to the basement"; "The road runs South")

The difference between the word "lead" and the word "go", and the difference between the word "blend" and the word "go", are rather low, whereas the difference between "blend" and "lead" is high. Figure 1 uses *pathlen* measure to illustrate the difference between word senses.

We cannot use that part of [14] that assumes a Euclidean distance, since the triangle inequality does not hold in the natural language framework of our application. However, the other requirements that the difference function must obey, are satisfied, namely

Boundedness This is the requirement that the distortion is finite. This holds in our case, because no matter how different two sentences are, our difference function between them will produce a finite outcome.

Symmetry This is the requirement that $d(a, b) = d(b, a)$. That we satisfy this follows from the fact that the numbers we use for differences are weights of edges in an *undirected* graph (as will become apparent in section 3).

Equality This is the requirement that $d(a, b) = 0$ if and only if $a = b$. This holds in our case.

2.2 Model of the Adversary

The Achille's heel of traditional synonym-substitution based watermarking is an adversary who, randomly and in a wholesale fashion, carries out synonym substitutions. While such an adversary may be effective against these previous uses of the synonym substitution technique, our scheme thwarts such an adversary (as will be discussed later in the paper). However, thwarting such a random adversary is not a fair criterion, as it is a rather naive form of attack. Therefore our model of the adversary is one who fully knows our scheme (except the key) and has the same knowledge and computational capabilities (including automated natural language processing tools, and access to all the databases used by the encoding and decoding processes). The security of our scheme therefore does not depend on an assumption of naive ignorance on the part of the adversary, rather, it depends on the following facts.

First, the approximately meaning-preserving changes that we make are in the direction of more ambiguity, and automated disambiguation is harder for the adversary than it is for us because we start with a less ambiguous (original document) document than the one in the hands of the adversary (watermarked document). A human, however, is able to quickly disambiguate when reading the marked text: We are exploiting the well-established fact in the natural language processing community, that humans are much better than computers at disambiguation [18].

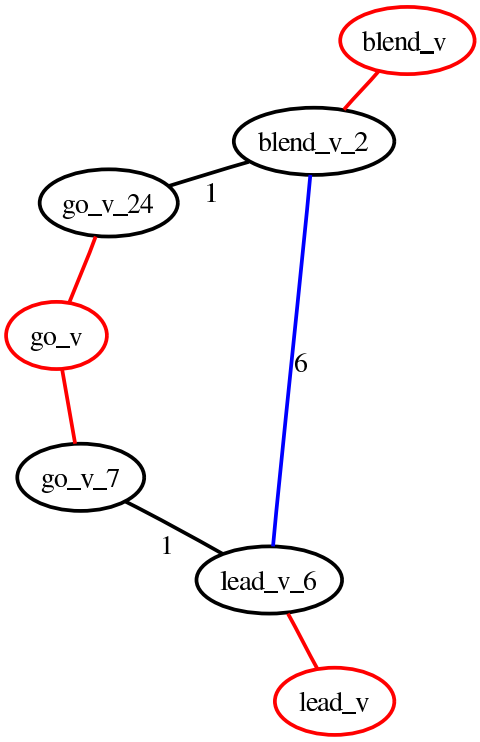

Figure 1: An example illustrating how the differences in natural language need not satisfy the triangle inequality. The presented differences are calculated by *pathlen* measure of WordNet::Similarity library

Second, we carry out substitutions not only for the purpose of encoding the mark in the text, but also for the purpose of getting as close as possible to the allowable cumulative distortion limit, an idea that was previously suggested in a broader framework (see [19]). That is, we keep doing transformations even after the mark is embedded, for the sole purpose of accumulating enough distortion to get close to the allowable limit. This is crucial: The adversary, not knowing the key, does not know *where* we carried out the modifications (as that choice is key-based), and trying to "un-do" them by wholesale application of transformations will cause the adversary to exceed the allowable distortion limit (because s/he started out close to it in the first place).

In practice the adversary is not limited to synonym substitutions; s/he can also make meaning-preserving syntactic changes which effect the ordering of words without altering them [25]. This sort of attacks are more problematic in languages with free word order (e.g. Finnish, Hindi, Turkish); since the adversary can put the words in any permutation without damaging the text too much. If the adversary performs a more sophisticated attack using syntactic modifications, even though the root words will be preserved, their order may change in the copied text. In this case, the watermarking mechanism should take into account the possible syntactic modifications. The watermarking mechanism can use an auxiliary fixed syntax with a fixed word order for watermark embedding and detection purposes (e.g. subject, object, verb). In addition to this, ambiguity may be used to prevent from syntactic modifications as a pre-emptive defense at the watermark embedding time (e.g. using pronouns as ambiguous references).

Note that the adversary in our scheme uses an automated process to attack the watermark. Our aim is to raise the bar for the cost of removing the watermark message. In this sense, our scheme can be considered successful if it forces the adversary to manually process the document for removing the watermark.

3. SYNONYM SUBSTITUTION BASED WATERMARKING SYSTEM

Most of the previous work on information hiding in natural language text was designed to get better as the accuracy of natural language processing tools improves. In [4], Bergmair discusses the need for an accurate word sense disambiguator for fully automating a synonym substitution based steganography system that requires sense disambiguation both at encoding and decoding time. Topkara et al [24] give examples of how accuracy of the text processing tools affects the quality of the watermarked sentences.

Whereas previous work in this area typically benefits from progress in natural language processing, in the present paper we propose a watermarking system that benefits from the difficulty of automated word sense disambiguation, as it increases the adversary's complexity of removing the hidden message.

We propose a lexical watermarking system that is based on substituting certain words with more ambiguous words from their synonym set. Here by ambiguous word, we mean a word that is a member of several synonym sets and/or has many senses. For example, if the geographic context is North Carolina, then "the Raleigh-Durham area" can equivalently be re-stated as "the triangle" where "triangle" refers to the "research triangle area". The adversary now has to figure out that this particular "triangle" is neither a three-sided polygon, nor a musical instrument, nor a situation in which two people are competing for the love of the same third person. The difficulty of the adversary's task of automated disambiguation is widely accepted in the natural language processing community. Although our implemented system cannot yet carry out the above specific transformation (because its public knowledge base does not yet contain the specific equivalence it uses), we mentioned it because it perfectly exemplifies the kinds of substitutions we seek (and that will undoubtedly be present in a more refined version of our prototype).

Homograph is a more specific linguistic term used for the "ambiguous" words. Two or more words are homographs if they are spelled the same way but differ in meaning and origin, and sometimes in pronunciation. For example the word "bank" is a homograph, and means either a financial institution, the edge of a stream, or a slope in the turn of a road. We have implemented our system to consider the words with more than one sense as homographs, and only homographs within a synonym set are considered as the target words for synonym substitution.

An example of what our system does carry out today is when we encounter the word "impact" as a verb in our cover text: We will find that it is a member of {affect, impact, bear upon, bear on, touch on, touch} synonym set. The verbs "affect" and "touch" are possible alternatives for replacing the verb "impact". Our system favors replacing the word "impact" with the word "touch" over the word "affect", because the expected distortion that will be imposed by the verb "touch" on the adversary, $E(d_2(touch; impact, s2))$, is higher than the expected distortion, $E(d_2(affect; impact, s2))$, that will be imposed by the verb "affect". $E(d_2(w_c; w_o, s_o))$ is the average difference of every sense of watermark carrying word, w_c, to the original (word,sense) pair, (w_o, s_o). Refer to Section 3.1 for the details of how this expected distortion is calculated in our system. See Figure 2, for a simplified illustration of this embedding. For simplicity of the graph, word sense nodes are collapsed into one node for each word except the verb "impact", whose sense is learned from the cover text. Only relevant nodes are colored and labeled. Edge weights are again omitted for simplicity. "affect" has five senses, "touch" has fifteen senses, "impact" has two senses, "bear on" has four senses, "touch on" has four senses, and "bear upon" has only one sense.

In our scheme, more information is available about the sense (meaning) of the words at the watermark embedding time, since the original document is available. The watermarking process in this paper, replaces as many as possible words with one of the homographs in their synonym set. Hence the watermarked text has "blurred" meaning and it becomes harder for an adversary to perform word sense disambiguation on it (i.e., the ambiguity has increased in such a way that it is harder to find the correct synonym of the words without human intervention). In such a setting, the adversary will not be willing to replace every homograph word with a non-homograph automatically and the watermark will be successfully retained. Note that, it may also be possible to magnify this asymmetry of information further by consulting to the actual author of the text during watermark embedding, for the correct sense of a word at watermarking time.

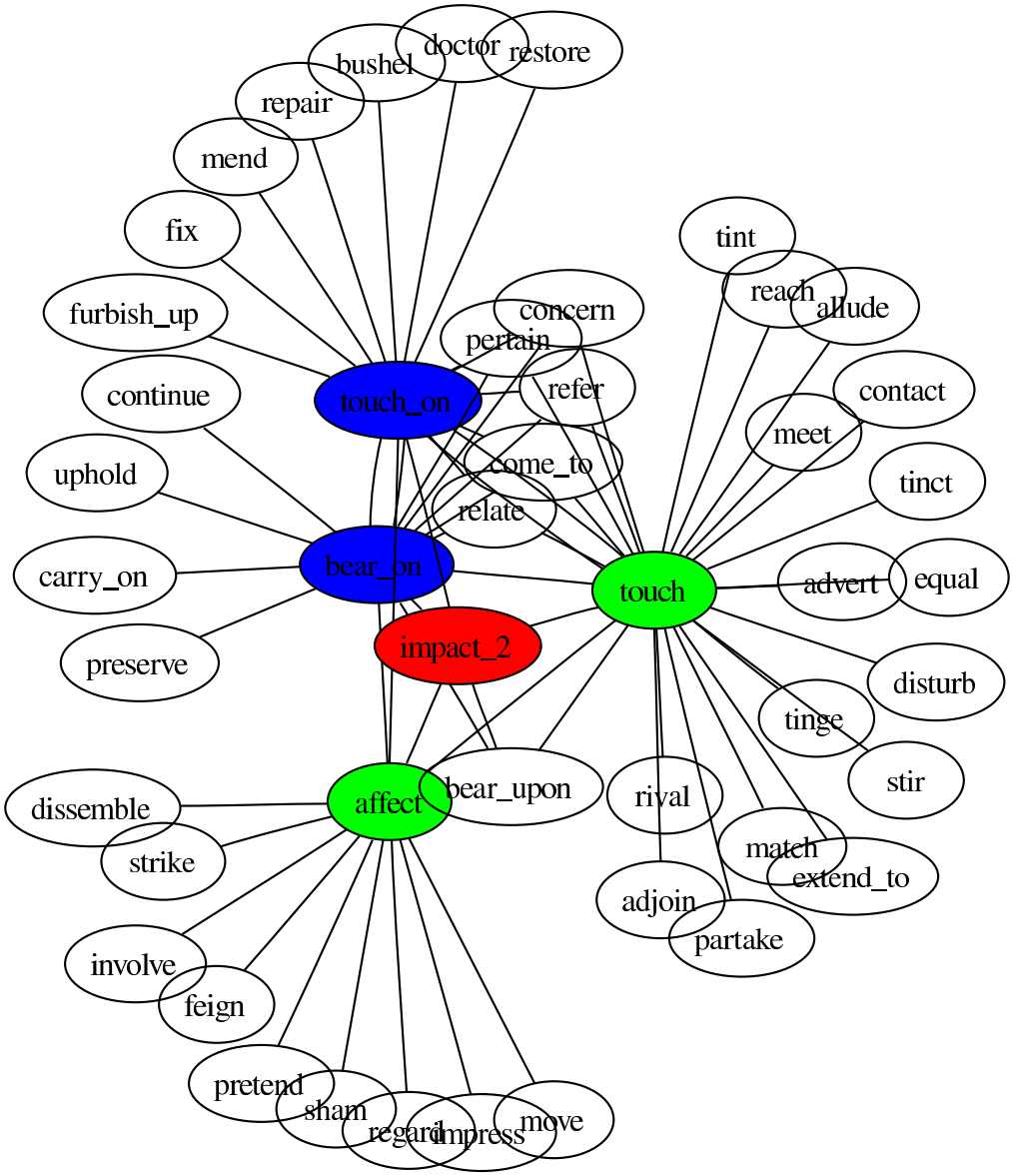

Figure 2: A sample colored graph that shows the connections of the verb "impact". For simplicity of the graph, word sense nodes are collapsed into one node for each word and edge weights are omitted. Only relevant nodes are colored and labeled.

As an example, consider the sentence "he went without water and food for 3 days" coming from a watermarked text. If the adversary had replaced the word "went" with the word "survived" then the change in the meaning is minimal. However, if he had replaced "went" with "died", the meaning of the sentence would be taken very far from its original meaning. Yet both "survive" and "die" are synonyms of different senses of the word "go".

Loosely speaking, we are using ambiguity in natural language to imitate one-way hash functions. For example, when given as original sentence, "the gas station is after the cant on highway 52." we can replace the word "cant" with its synonym, "bank". The transformed sentence will now say "the gas station is after the bank on highway 52", where it is not obvious whether the gas station is after the financial institution, after the inclined slope on the turn of the road, or after the stretch of road that briefly adjoins the river's bank. Our system uses this deliberate injection of ambiguity whenever possible, and replaces words with more ambiguous words from their respective synonym set.

The decoding process is not dependent on the original text and there is no need to know the sense of a word in order to decode the message. This simplicity of decoding process makes it computationally light, and it enables the copyright infringement detection to be performed by a web crawler on a large number of online documents.

The details of the encoding and decoding processes are explained in the next subsection.

3.1 The Encoding and Decoding Algorithms

Our system is based on building a weighted undirected graph, G, of (word,sense) pairs, where an edge between two nodes represents that they are synonyms. In our experimental implementation, the synonym sets of words are taken from WordNet [9]. Each weight on a graph's edge is a measure of the similarity between its two endpoints.

Several different techniques and similarity functions have been proposed in the natural language processing literature to quantify the similarity of two words. A large number of these techniques are based on WordNet, which is an electronic dictionary that organizes English nouns, verbs, adjectives and adverbs into synonym sets, each representing one underlying lexical concept [9]. See Table 1 for statistics about the content of WordNet [2]. There are several semantic relations that link the synonym sets in WordNet such as "is-a-kind-of", or "is-a-part-of" relations. Some of the word similarity functions are available as a Perl Library called WordNet::Similarity [15, 17]. WordNet::Similarity package implements six different similarity measures that are in some way based on the structure or the content of WordNet.

Three of the WordNet::Similarity measures are based on the information content of common subsumers of two concepts. The Resnik measure is based on using the information content of most specific common subsumer, Least Common Subsumer (LCS), of two concepts as a similarity value. The Lin measure scales the information content of the LCS by the sum of the information contents of the two concepts, while the Jiang and Conrath measure takes the difference of this sum and the information content of the LCS. The information content of a concept is learned from a sense-tagged corpus, Semcor [13].

[2]http://wordnet.princeton.edu/man/wnstats.7WN, last visited on May 9th 2006

The other three similarity measures are based on path lengths between pairs of concepts. The Leacock and Chodorow measure is based on scaling the shortest path length between two concepts by the maximum path length found in the subtree of the "is-a" hierarchy that includes these two concepts. The Wu and Palmer measure is calculated by finding the depth of the LCS of the two concepts and scaling it by the sum of the depths of two concepts. The *path* measure is a baseline metric and is equal to the inverse of the length of the shortest path between the two concepts.

Another method for measuring the similarity between the words is statistically analyzing a large and balanced text corpus in order to learn the mutual information of the two words, and resemblance of their context. Several mutual information measures have been proposed for calculating word similarity, see [23] for a good survey of these measures.

Language models can be used to learn more information about the similarity of the context of two concepts. A Language Model (LM) is a statistical model that estimates the prior probabilities of n-gram word strings [21]. An n-gram LM models the probability of the current word in a text based on the $n-1$ words preceding it; hence, an n-gram model is a $n-1^{th}$ order Markov model, where, given the probability of a set of n consecutive words, $W = \{w_1, \ldots, w_n\}$, the LM probability is calculated using

$$P(w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_0, \ldots, w_{i-1}), \qquad (3)$$

where the initial condition $P(w_1|w_0)$ is chosen suitably. A set of words $C = \{c_1, \ldots, c_k\}$ may be considered "similar" if $P(c_j|w_1, \ldots, w_n) \simeq P(C|w_1, \ldots, w_n)P(c_i|C)$ holds for all $c_j \in C$. Readers are referred to [7] for a heuristic algorithm that can be used to compute C using language models.

In our experiments we have used the Wordnet::Similarity measures for simplicity. In future work, we will compare the change in the quality and resilience of the watermarking when a corpus-based measure is used. More details about our experiments can be found in Section 4.

After graph G is formed, we select a subgraph, G^W of G using the secret key k. This subgraph selection is performed over the words that have homographs in their synonym sets. After this, we use k once more to color the graph in such a way that approximately half of the homograph neighbors of a non-homograph word are colored with blue to represent the encoding of "1", and the other half are colored with green to represent the encoding of "0", while non-homographs are colored with black to represent "no-encoding". See Figure 2 for a simplified example where the word "impact" is colored "red" only to show that it is the word that will be replaced during embedding.

At encoding time, we calculate the expected distortion value for the adversary, which in some sense measures how hard it would be for the adversary to find the original word, w_o, given the mark carrying word, w_c. Note that, if the adversary can replace w_c with w_o, then not only the mark bit encoded by that word will be removed, the distortion introduced by the watermarking process will also be undone. In our implementation, $E(d_2(w_c; w_o, s_o))$ is calculated by summing up the differences of every sense of w_c to the original (word,sense) pair, (w_o, s_o) normalized over the number of senses of w_c, which is denoted with $|S(w_c)|$. This is formalized as below:

Category	Unique Strings	Synsets	Word-Sense Pairs	Monosemous Words	Polysemous Words	Polysemous Senses
Noun	117097	81426	145104	101321	15776	43783
Verb	11488	13650	24890	6261	5227	18629
Adjective	22141	18877	31302	16889	5252	14413
Adverb	4601	3644	5720	3850	751	1870
Total	155327	117597	207016	128321	27006	78695

Table 1: Wordnet2.1 Database Statistics

$$E(d_2(w_c; w_o, s_o)) = \frac{\sum_{s_i \in S(w_c)} sim(w_c, s_i; w_o, s_o)}{|S(w_c)|} \quad (4)$$

where s_o is the sense of the original word, w_o, in the original document, and $sim(w_c, s_i; w_o, s_o)$ is the similarity based difference between (word,sense) pairs, it increases as the words get more dissimilar.

If there are more than one candidate homograph with the same color (the color that is required to encode the current bit of the message, m) then the one with the maximum $E(d_2())$ value is picked. Since we are using the WordNet-based similarity measures, the difference between pairs from the same synonym set is the same for all the words in that set, which makes the encoding distortion identical for all alternative pairs from the same synonym set. However, this need not be the case if alternative similarity measures are used. The following are summaries of the encoding and decoding algorithms, based on the above discussion.

Steps of the encoding algorithm:

- Build graph G of (word,sense) pairs. Use WordNet to find synonym sets of (word, sense) pairs. In addition, connect different senses of the same word with a special edge in order to follow the links to every neighbor of a word independent from its senses.
- Calculate differences between the (word,sense) pairs, $d(wi_{sense_k}, wj_{sense_l})$, using a similarity measure. Assign these values as edge weights in G.
- Select a subgraph G^W of G using the secret key k.
- Color the graph G^W. Detect the pairs of words (w_i, w_j), where w_i and w_j are in the same synonym set with one of their senses, and have more than one sense. In other words, these words act as homographs. Color w_i and w_j with opposite colors in graph G^W, using k to decide which one gets to be colored in blue (i.e, encodes a "1") and which one gets to be colored in green (i.e., encodes a "0"). Color non-homographs as black.
- $c = 1$
- For each word w_i in the cover document S
  - $bit_c = M[c]$
  - if $w_i \in G^W$ then replace w_i with the neighbor that carries the color that encodes bit_c
    if there are more than one neighbor that encodes bit_c
    for each, w_j, of these neighbors calculate $E(d_2(w_j; w_i, s_k)) = \frac{\sum_{s_l \in S(w_j)} sim(w_j, s_l; w_i, s_k)}{|S(w_j)|}$
    pick the neighbor with the maximum $E(d_2(w_j; w_i, s_k))$ value
    Increment c (if $c = |M| + 1$ then set $c = 1$)

If the cover document's size is long enough the message, M is embedded multiple times. We assume that the message M, that is input to the watermarking system, has already been encrypted and encoded in a way that it is possible to find the message termination point when reading it sequentially from an infinite tape. The encrypted M could have an agreed-upon fixed length (symmetric encryption preserves length, so we would know how to chop the decoded message for decryption). Or, alternatively, if the length of M is unpredictable and cannot be agreed upon ahead of time, the encrypted M could be padded at its end with a special symbol, i.e. #, that would act as a separator between two consecutive copies of the encryption.

Steps of the decoding algorithm:

- Build the same graph G of (word,sense) pairs using the same difference function as the one used for the encoding process
- Select a subgraph G^W of G using the secret key k
- Color the graph G^W using k (as for the encoding process)
- $c = 1$
- For each word w_i in the cover document S
  - if $w_i \in G^W$ then check the color of the node that represents w_i.
    if it is black, move to the next word
    if it is blue, assign 1 to $M[c]$ and increment c
    if it is green, assign 0 to $M[c]$ and increment c

The decoding algorithm is simply a series of dictionary lookups. We envision that this simplicity will enable our system to be used for watermaking online text documents. Then, web crawlers that are indexing web pages can also check for watermarks or metadata embedded using our system in the pages they visit.

3.2 Context-Dependent Synonyms

The notion of a synonym that WordNet subtends is, as in any fixed dictionary, rigid in the sense that it fails to capture the notion of a *context-dependent* synonym: A synonym relationship that holds only within a particular text document. Such a relationship holds between words that in general are not synonyms, but that in a particular text's context are de facto synonyms, pretty much the way "the sleuth" is synonym of "Sherlock Holmes" in most of Arthur Conan Doyle's books, but is a synonym of "Hercule Poirot" or "Miss Marple" in some of Agatha Christie's books.

Fully automating the encoding of such a scheme is highly nontrivial and error-prone, and we therefore envision a semi-automatic interactive encoding mechanism where the text's

author decides on the acceptability (or lack thereof) of substitutions proposed by the system, or even suggests substitutions from scratch. This is especially appropriate in texts where the quality of the prose is as important as the meaning it carries, or when a fully automatic process is likely to introduce unacceptable errors. For example, Daniel Defoe would not approve a proposed substitution of "Friday" by "sixth day of the week" in the context of his Robinson Crusoe book.

There are two benefits to this more general notion of a synonym: (i) it increases the repertoire of effective synonyms available, thereby increasing encoding capacity; and (ii) it is harder for the adversary to un-do, both because of the context-dependency and because an outside (possibly human) adversary incurs a larger time penalty to analyze and comprehend the text than the author.

3.3 Generalization Substitutions

Further resilience can be provided by the use of meaning-preserving generalizing substitutions (replacing the specific by the general, e.g., "lion" by the less specific "big cat" or the even more general "carnivore"). As stated earlier, WordNet includes a "`is_a`" types of hierarchies that we could use to achieve this – we could advantageously "move up" one of these `is_a` hierarchies of WordNet in a manner that does not destroy meaning: For example, it may be perfectly acceptable to replace "lion" with "big cat" or even with "carnivore" even though the latter two are not synonyms of the former (or of each other), as lion-big_cat-carnivore form a chain of ancestors in the `is_a` hierarchy. But the substitution of "lion" with "carnivore" would not be acceptable if the text also contains much about a wolf (which is also a carnivore), although the substitution of "lion" with "big cat" is still acceptable (as would be the substitution of "wolf" with "canine"). More formally, when making a substitution we are allowed to move up a particular link in the `is_a` hierarchy as long as doing so does not gobble up an extra descendent node that also appears in the text (as that would be meaning-damaging). Because in a WordNet hierarchy there can be more than one parent, it is possible that we are unable to move up one link, but able to move up another link: For example, in a text with a camel and a kangaroo, we cannot generalize "kangaroo" to "herbivore" but we can generalize it to "marsupial".

The above process makes it harder for the adversary to using such transformations to attack the watermark, for two reasons: (i) Many of these substitutions have already been applied (by the encoding process) to the maximum extent possible ("as far up the hierarchy as possible"); and (ii) most hierarchies have a higher branching factor going down than going up (many children, fewer parents), and therefore replacing the general by the specific in an attack is problematic (the adversary has more choices).

4. EXPERIMENTAL RESULTS :EQUIMARK

We have implemented a natural language watermarking tool, EQUIMARK [3], according to the system design proposed in Section 3. Equimark is implemented in Perl and uses WordNet::Similarity and WordNet::QueryData libraries [15, 16] [4]. WordNet 2.1 is used during the experiments presented in this section.

We used *pathlen*() function of WordNet::Similarity library in order to learn the difference between (word,sense) pairs, in other words $sim(w_j, s_l; w_i, s_k) = pathlen(w_j, s_l; w_i, s_k)$. *pathlen*() outputs the length of shortest path between the (word,sense) pairs in the "is-a" hierarchy of WordNet. As we have mentioned in Section 3, *pathlen* measure is a baseline metric and is equal to the inverse of the length of shortest path between the two concepts.

In future versions, *pathlen*() can be replaced with other similarity measures.

Equimark builds the graph, G_W, as follows. $listAllWords()$, a function from the WordNet::QueryData library, is used to generate a list of words, L. Later the content of L is increased by exploring the synonyms of the words in the initially generated list. After this step, Equimark assigns random colors to all words in L, using the secret key k. Later, words from L are processed by taking one word, w_i, at a time from the beginning of L and starting a breadth first exploration of WordNet, where w_i is the root. If a node is assigned blue or green initially, during the breadth first traversal, we try to color the neighbors (synonyms) of each word with the opposite of the word's color. Whenever there is a conflict, Equimark colors the newly explored node with black, marking it as a "no-encoding" word.

A node gets an encoding color, i.e. blue or green, if it has more than one sense and there is at least one synonym for each one of its senses. All single sense words (monosemous words) are colored with black, since they are not "ambiguous", they do not increase the resilience if they are used for marking.

We color a word with black if it does not have any synonyms for one of its senses, even though it has more than one sense. For example, consider the word "jury":

- jury – (a body of citizens sworn to give a true verdict according to the evidence presented in a court of law)
- jury, panel – (a committee appointed to judge a competition)

We can not include the word "jury" in our set of encoding words in the current system, because if we color it with blue or green, and if it appears with its first sense in the cover document we will not be able to undo the effect of its color to the encoding. In the future versions, Wet Paper Codes [10] can be used in order to be able to increase the capacity of watermarking with Equimark. When the wet paper codes are used, we can mark the word "jury" as a stuck cell if it is used in its first sense in the cover document. But, when it is used in its second sense, we can mark it as a changeable cell and use it for encoding. Note that, if word sense disambiguation was possible at watermark decoding time, this limitation would not be an issue, since we could discard those word senses without synonyms (our decoding algorithm is a series of dictionary lookups).

An adversary who is aware of the fact that some of the words in G_W have to be colored "black" due to the phenomenon explained in the previous paragraph, can find instances of those words in the watermarked text (by checking if a word does not have synonyms for at least one of

[3]The source code of this watermarking system will be available at *http://homes.cerias.purdue.edu/∼mercan/equimark/*.

[4]These libraries are implemented in Perl language and they are freely available from CPAN website at http://cpan.org/. Last visited on May 9th 2006.

Category	Black	Green	Blue
Noun	141408	7873	7998
Verb	7716	1919	1885
Adverb	4107	261	257
Adjective	19058	1940	1976
Total	172289	11993	12116

Table 2: A sample coloring performed by Equimark

its senses). Later, the adversary can randomly alter those words, since there is a non-zero probability of substituting them with an encoding word. This attack will add noise into the decoded message. In the scope of this paper, the effects of such attacks are not quantified. But as mentioned above, future versions of Equimark will be enhanced to be able to prevent the effect of noise addition to the watermark message either by the use of wet paper codes or error correction codes; besides increasing the bandwidth, this enhancement will also increase the resiliency of the system.

We still included a "relaxation" on the above coloring restriction in our implementation by checking the WordNet frequency value for the words that have more than one sense and some of them have at least one synonym, we check the senses that do not have any synonyms: if the frequency of this (word,sense) pair is below a threshold, we interpret it as "this pair is very rarely used in the language", and it is unlikely that we will encounter this particular sense of w_i in the cover document. Thus, we color w_i with an encoding color. See Table 2 for a sample distribution of colors for different word categories.

Equimark takes in four inputs: a cover document, a message M, a colored graph G_W and an embedding distortion threshold D_1.

In our experiments, we took our cover documents from a sense-tagged corpus, Semantic Concordance (SemCor) [13]. We used SemCor2.1 [5] since its sense-tags are mapped to WordNet 2.1 instead of the original SemCor that is tagged with WordNet 1.6 senses. SemCor has three parts: first part, *brown1*, consists of 103 semantically tagged Brown Corpus [12] files, in which all content words are tagged; second part, *brown2*, consists of 83 semantically tagged Brown Corpus files, in which all content words are tagged; third part, *brownv*, consists of 166 semantically tagged Brown Corpus files, in which only verbs are tagged. We have used nouns, and verbs as watermark carrying words.

Using already sense-tagged input text was a natural choice for our experiments since it let us focus on analyzing the amount of distortion the embedding incurs on the cover document and the resilience we can achieve. In another setting, at the encoding time, the author of the document might be prompted for help on disambiguating the sense of ambiguous words. As we have mentioned before, there is no need for sense disambiguation at decoding time.

Equimark restricts the cumulative embedding threshold to be below D_1. We used $pathlen()$ as the difference function. Refer to Section 2.1 for a discussion of the model proposed by Moulin et al.

In our experiments, $d_1(w_c, s_c; w_o, s_o)$ was always equal to 1 as we consider substitution only between the words from the same synonym set. $d_1(w_i, s_l; w_j, s_k) = 0$ only when $w_j = w_i$, since $pathlen(w_i, s_l; w_j, s_k)$ is length of the shortest path between the two concepts. However, usage of other similarity measures as the difference function might change this.

[5]Downloadable from Rada Mihalcea's homepage at http://www.cs.unt.edu/ rada/downloads.html. Last visited on May 9th 2006.

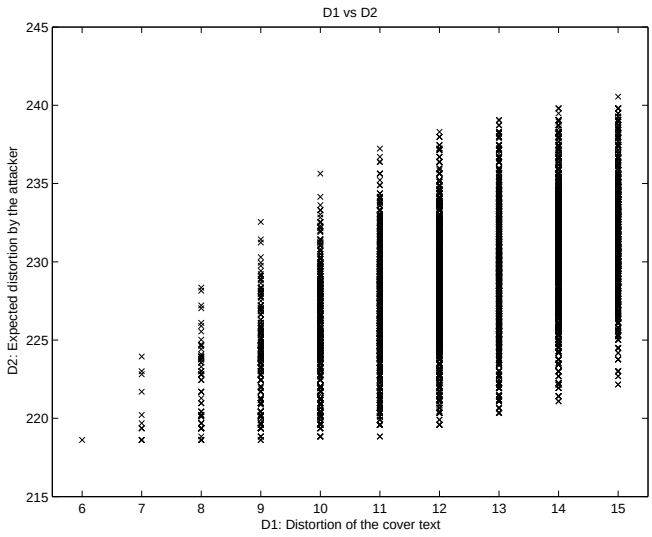

Figure 3: Each point indicates a successful watermark insertion. The X-axis is the incurred distortion to the cover document and Y-axis is the expected distortion that will be incurred by adversary to undo the watermark.

Our embedding algorithm picks synonyms that have maximum expected d_2 as described in Section 3 and Equation 4.

Refer to Figure 3 for an analysis of the relationship between the embedding distortion (x-axis), D_1, and the resilience (y-axis), D_2.

The graph in the Figure 3 corresponds to an experiment run over one of the *brown1* files, which has 1815 tagged tokens. Here, the watermark was a random string of 10 bits, for the sake of readability of the graph. Larger number of watermark bits create more branches in the search tree and results in very dense graphs. The maximum distortion, D_1 in this experiment was limited to 15 substitutions.

5. RELATED WORK

Natural language information hiding systems, that are based on modifying a cover document, mainly re-write the document using linguistic transformations such as synonym substitution [26, 2], paraphrasing [3, 24] or translation to another language [11]. Most of the proposed information hiding systems are designed for steganography. Even though the steganography systems do not need to take into consideration an active warden, they still need to obey to stealthiness constraints. This brings both watermarking and steganography to the same point: imposing minimum embedding distortion to the cover document. This requirement is also used as a justification for performing isolated changes on the cover document by doing in place replacement of mark carrying units i.e. words or sentences, with a synonym instead of text-wide-transformations. To the best of authors' knowledge this is the first work that quantifies distortion done on

the cover document and that is guided by both the message insertion and the resilience requirements.

T-Lex is one of the first implemented systems that embed hidden information by synonym substitution on a cover document [26, 4]. T-Lex first generates a database of synonyms by picking the words that appear only in the same set of synonym sets from WordNet. For example, if the synonym sets are given as $S_1 : \{w_1, w_2\}$, $S_2 : \{w_1, w_2, w_3\}$ the words w_1 and w_2 are inserted into this database as synonyms and w_3 is filtered out. The intersections between distinct synonym sets are eliminated to avoid usage of ambiguous words for encoding. This filtering causes the use of uncommon words while performing the substitutions (e.g. replacing "nothing" with "nada") due to the fact that common words tend to span through several unrelated synonym sets [22]. A given message is embedded into the cover text using the synonym set database as follows. First, the letters of the message text are Huffman coded according to English letter frequencies. The Huffman coded message is embedded into message carrying words in the cover text by replacing them with their synonyms in the synonym database of T-Lex . The synonym sets in this database are interpreted as a mixed-radix digit encoding according to the set elements' alphabetical order.

In [4], Bergmair provides a survey of linguistic steganography. He also discusses the need for an accurate word sense disambiguator for a fully automated synonym substitution based steganography, where sense disambiguation is required both at decoding and encoding time. The lack of accurate disambiguation forces the synonym substitution based information hiding systems to restrict their dictionaries to a subset of words with certain features. Besides decreasing the communication bandwidth, such restrictions cause the systems to favor use of rare words for encoding information [22]. In another work, Bergmair et al. proposes a Human Interactive Proof system which exploits the fact that even though machines can not disambiguate senses of words, humans can do disambiguation highly accurately [5].

Grothoff et al presented so far the only steganography system that advantageously exploits the weaknesses of current natural language processing tools [11]. They use the low quality of automatically translated text to conceal the existence of an embedded stego message. In their system, several machine translation systems are used to have several alternative translations for a given sentence, if the machine translation systems had been perfect they would have produced very similar or the same translation sentence. A similar approach has been introduced for using different MP3 encoders for steganography in [6]. In the system proposed by Grothoff et al., the quality of the output document is less important than being able to deliver the message and the stealthiness of the communication. Thus, they do not need to limit the distortion done on a cover document as long as it carries the message and shows the statistical and stylistic characteristics of machine translation systems' output.

Privacy-preserving data mining techniques aim to ensure privacy of the raw local data while supporting accurate reconstruction of the global data mining models. *Data perturbation* is one of the approaches used in this area. This approach is based on distortion of the user data by user-set parameters in a probabilistic manner such that accurate models for joint data of several users can be generated by a central data mining process. The data miner is given the perturbed database, V, and the perturbation matrix, A, where $A_{vu} = p(u \rightarrow v)$ and $p(u \rightarrow v)$ is the probability of an original user record, $u \in U$, being perturbed into a record $v \in V$. U is the user's local copy of the database. After receiving V and A, the data miner attempts to reconstruct the original distribution of database U and generate data mining models for U.

Agrawal et al. in [1] proposes further randomization of perturbation parameters in, A, for each user separately in order to provide extra privacy for the users. Randomization of the perturbation parameters make it harder for the data miner to guess the original values of the records in U. Previous data perturbation systems were using deterministic perturbation matrices. Agrawal et al. quantify this extra privacy by enforcing an upper limit on the ratio of the probability of a record, $v \in V$ being perturbed from either of the two records, $u_1 \in U$ or $u_2 \in U$. This quantification is formalized below, where S_X is the domain set for the values of the records in database X:

- A randomization $R(U)$ is at most γ-amplifying for $v \in S_V$ if

$$\forall u_1, u_2 \in S_U : \frac{p(u_1 \rightarrow v)}{p(u_2 \rightarrow v)} \leq \gamma \qquad (5)$$

  where $\gamma \geq 1$ and $\exists u : p(u \rightarrow v) > 0$.

Above assertion also means that the ratio of any two matrix entries in A should not be more than γ.

This idea has a similar intuitive motivation as what we are proposing in this paper. While Agrawal et al. propose randomization of perturbation parameters for improving privacy of data mining, we propose increasing ambiguity of a watermark carrying document for improving the resiliency of watermarking.

6. CONCLUSION AND FUTURE WORK

We presented and discussed a synonym-based natural language watermarking system that we designed and built. This is the first instance of the use of quantified notions of differences between sentences in natural language information hiding. The use we make of such differences is twofold. First, we use them to maximize capacity without exceeding the maximum allowable cumulative distortion, and achieve resilience by giving preference to ambiguity-increasing transformations that are harder for the adversary to un-do. Second, we achieve additional resilience by getting close to the maximum allowable cumulative distortion ourselves, as a way of preventing the adversary from carrying out attacking transformations (as these are likely to push the text beyond the allowable distortion limit).

Future work will enhance the system in the following ways.

- Move from a solely WordNet-based difference function, to a more domain-specific difference function that is also corpus-based. For example, using the Reuters corpus in addition to WordNet will result in a better watermarking scheme for Reuters articles.

- Increasing the information-carrying capacity through the use of wet-paper codes [10]. The specific way this increases capacity was discussed in Section 4.

- Increasing the resiliency through the use of wet-paper codes or error correction codes. The specific way this increases resiliency was again discussed in Section 4.

- Making experiments to evaluate the performance of copyright infringement detection systems for two cases; where the original document is either watermarked or not-watermarked.

- Using a more powerful ontology to increase both capacity and resilience. This kind of knowledge base would allow such substitutions as replacing "Washington D.C." by "the capital". Such a powerful ontology can provide us the ability to re-write a sentence like "Bush returned to Washington D.C" as "The President came back to the capital".

7. REFERENCES

[1] S. Agrawal and J. R. Haritsa. A framework for high-accuracy privacy-preserving mining. In *Proceedings of the 21st International Conference on Data Engineering*, Tokyo, Japan, April 5-8 , 2005.

[2] 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.

[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] R. Bergmair and S. Katzenbeisser. Towards human interactive proofs in the text-domain. In *Proceedings of the 7th Information Security Conference*, volume 3225, pages 257–267. Springer Verlag, September, 2004.

[6] R. Böhme and A. Westfeld. Statistical characterisation of mp3 encoders for steganalysis. In *Proceedings of the ACM Multimedia and Security Workshop*, pages 25–34, Magdeburg, Germany, September 2004.

[7] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Compututational Linguistics*, 18(4):467–479, 1992.

[8] 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.

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

[10] J. Fridrich, M. Goljan, and D. Soukal. Efficient wet paper codes. In *Proceedings of 7th Information Hiding Workshop*, volume LNCS 3727, pages 204–218. Springer-Verlag, 2005.

[11] C. Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman, and M. Atallah. Translation-based steganography. In *Proceedings of Information Hiding Workshop (IH 2005)*, page 15. Springer-Verlag, 2005.

[12] H. Kucera and W. N. Francis. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, Rhode Island, 1967.

[13] S. Landes, C. Leacock, and R. I. Tengi. Building semantic concordances. In *In Fellbaum, C. (ed.) (1998) WordNet: An Electronic Lexical Database.*, Cambridge, Mass., 1998.

[14] P. Moulin and J. A. O'Sullivan. Information-theoretic analysis of information hiding. *IEEE Transactions on Information Theory*, 49:563–593, 2003.

[15] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::Similarity - Measuring the Relatedness of Concepts. In *Proceedings of Fifth Annual Meeting of the NAACL*, Boston, MA, May 2004.

[16] J. Rennie. Wordnet::Querydata: a Perl module for accessing the WordNet database. In *http://people.csail.mit.edu/ jrennie/WordNet*, 2000.

[17] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[18] P. Resnik. Selectional preference and sense disambiguation. In *the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington D.C., USA, April 1997.

[19] R. Sion, M. Atallah, and S. Prabhakar. Power: A metric for evaluating watermarking algorithms. In *IEEE ITCC*, pages 95–99, Las Vegas, Nevada, 2002.

[20] R. Sion, M. Atallah, and S. Prabhakar. Rights protection for discrete numeric streams. *IEEE Transactions on Knowledge and Data Engineering*, 18(5), May, 2006.

[21] A. Stolcke. Srilm - an extensible language modeling toolkit. In *Proceedings of International Conferrence on Spoken Language Processing*, 2002.

[22] C. M. Taskiran, U. Topkara, M. Topkara, and E. Delp. Attacks on lexical natural language steganography systems. In *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, 2006.

[23] E. Terra and C. Clarke. Frequency estimates for statistical word similarity measures. In *Proceedings of Human Language Technology Conference*, pages 244–251, Edmonton, Alberta, 2003.

[24] 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.

[25] 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.

[26] K. Winstein. Lexical steganography through adaptive modulation of the word choice hash. In *http://www.imsa.edu/ keithw/tlex/*, 1998.