Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 23 (2009) 107–125

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Natural language watermarking via morphosyntactic alterations

Hasan Mesut Meral [a], Bülent Sankur [b], A. Sumru Özsoy [a,c], Tunga Güngör [c,d,*], Emre Sevinç [c]

[a] *Boğaziçi University, Linguistics Program, Bebek, İstanbul 34342, Turkey*
[b] *Boğaziçi University, Department of Electrical and Electronic Engineering, Bebek, İstanbul 34342, Turkey*
[c] *Boğaziçi University, Cognitive Science Program, Bebek, İstanbul 34342, Turkey*
[d] *Boğaziçi University, Department of Computer Engineering, Bebek, İstanbul 34342, Turkey*

Received 17 October 2007; received in revised form 17 April 2008; accepted 28 April 2008
Available online 3 May 2008

Abstract

We develop a morphosyntax-based natural language watermarking scheme. In this scheme, a text is first transformed into a syntactic tree diagram where the hierarchies and the functional dependencies are made explicit. The watermarking software then operates on the sentences in syntax tree format and executes binary changes under control of Wordnet and Dictionary to avoid semantic drops. A certain level of security is provided via key-controlled randomization of morphosyntactic tools and the insertion of void watermark. The security aspects and payload aspects are evaluated statistically while the imperceptibility is measured using edit-hit counts based on human judgments. It is observed that agglutinative languages are somewhat more amenable to morphosyntax-based natural language watermarking and the free word order property of a language, like Turkish, is an extra bonus.

Keywords: Natural language watermarking; Tree bank; Agglutinative; Morphosyntax; Text payload

1. Introduction

Natural language watermarking (NLW) is an emerging research area at the intersection of natural language processing and information security. It aims to hide information in texts with applications similar to those in multimedia watermarking (Cox et al., 2002). The goals could be to create a subliminal communication channel through which to transport hidden information, to enable content and authorship authentication, to enrich the text with metadata, to fingerprint it for distribution, etc. While natural language watermarking and multimedia watermarking share common goals, they employ very different techniques. A plethora of watermarking techniques have been explored for multimedia documents in the last decade (Cox et al., 2002) and some have even turned into industrial products. In contrast, studies on natural language watermarking are just starting as

[*] Corresponding author. Tel.: +90 212 3597094; fax: +90 212 2872461.
E-mail addresses: mesut.meral@boun.edu.tr (H.M. Meral), bulent.sankur@boun.edu.tr (B. Sankur), ozsoys@boun.edu.tr (A. Sumru Özsoy), gungort@boun.edu.tr (T. Güngör), emres@bilgi.edu.tr (E. Sevinç).

0885-2308/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2008.04.001

attested by the scarcity of related papers (Bergmair, 2004, 2007; Khankhalli and Hau, 2002; Bennett, 2004).

Initially, NLW researchers exploited watermarking techniques adapted from multimedia watermarking. These watermarking and/or steganographic techniques were non-linguistic in nature and made extensive use of character changes such as kerning, random assignment of character spaces, line shifting, word shifting and insertion of sound encoding (Bailer and Rathner, 2001). These "printed text" watermarking approaches had limited scope and were not robust against text reformatting and transcription attacks (Khankhalli and Hau, 2002). Since the exploitable redundancy and embedding opportunities in printed text are significantly less than in multimedia documents such as image, video, audio and graphics, attention was turned to linguistic tools. These tools can appear under the guise of semantic and syntactic transformations, morphological and punctuation manipulations, lexical substitutions, translations and word level typographical alterations (Topkara et al., 2005; Meral et al., 2006). We will refer to the term "natural language watermarking" as the information hiding techniques within a text exclusively based on linguistic tools, while the term "text-watermarking" loosely encompasses both document image formatting and linguistic manipulations.

The motivation for our work is to develop a novel NLW scheme that is imperceptible, secure and based on morphosyntactic manipulations of sentences. We develop our scheme based on the seminal work of Topkara et al. (2006b). The contributions of our work can be summarized as follows: (i) we analyze a fairly complete list of morphosyntactic tools. We observe that agglutinative languages with high suffixation, such as Turkish, constitute fertile ground for watermarking; (ii) our scheme allows trade offs between payload and security in a controlled manner, and these properties are quantified based on a language model and its statistics; (iii) imperceptibility or acceptability of manipulations is measured via edit statistics.

The morphosyntactic approach was chosen de facto since alternative approaches did not look very viable. Most languages in contrast to English are not very rich in synonyms; in fact, some languages like Turkish have a single word per concept, and borrowed synonyms from other languages look alien. Purely morphological watermarking was not considered since morphology and syntax work often hand in hand, hence should be better handled under the guise of morphosyntax. Watermarking via punctuation alterations is not a stable enough option and can easily result in unwanted stylistic and meaning differences. Semantic watermarking, as argued by Atallah et al. (2000), constitutes probably the most flexible and prolific approach to NLW. However, the present day technology does not yet offer adequate tools for semantic interventions and pragmatic extensions in the word domain leading to semantic watermarking. On the other hand, first, the morphosyntax of languages often offers a rich set of NLW tools (Topkara et al., 2006b). Second, the syntactic alterations are based on formal descriptions of linguistic expressions in a sentence domain for which parsers and transformers exist or can be built. Finally, we conjecture also that the manipulation of the morphosyntactic features would have less impact on the semantics (i.e. least semantic distortion) of the original text when compared with the alternatives of lexical and semantic feature transformations.

Although syntax-based NLW has been proposed before, the algorithm proposed here is novel in its pseudo-random recruitment of morphosyntactic tools and subjective assessment of watermarked texts. The proposed NLW algorithm is applicable to any language, given a repertoire of morphosyntactic alternative forms, and ancillary material such as Wordnet and Dictionary. It processes a text progressively at the sentence level and selectively implements feasible watermarking tools when it encounters their appropriate input representation. As a case study, imperceptibility and payload capacity are given for Turkish language, which is an agglutinative language rich in morphosyntax. In order to explore the watermarking potential of this agglutinative language, we introduce a two-level embedding algorithm, which takes as input a dependency tree and converts it into a structure representing hierarchical relations. To enhance robustness of the watermark we introduce randomized order of tool selection and insertion of a "pass" tool creating void watermarks. Finally, we measure imperceptibility via user feedback.

This paper is organized as follows: In Section 2, we discuss the current state of the art in NLW and we expound the methodological basis of our study. In Section 3, we introduce the NLW model and describe the functions of its modules. Section 4 provides the results of the watermarking experiments where we discuss the occurrence statistics, imperceptibility effects, security measures and payload. Conclusions are drawn in Section 5. Appendix A gives the NLW tool repertoire for Turkish and English.

2. Natural language watermarking

Here we describe the state of the art in NLW, discuss the three requirements of watermarking, namely, imperceptibility, payload, and robustness, and present the tools of NLW.

2.1. NLW requirements

As any data hiding algorithm, NLW must also satisfy the three basic desiderata: (i) imperceptibility, (ii) payload and (iii) robustness (Atallah et al., 2001). These are often contradictory requirements: for example, imperceptibility demands that the watermarking strength be subdued which in turn limits the payload. To the best of our knowledge, there does not yet exist a standard NLW method that satisfies all the above criteria. In the NLW context these requirements translate into the following:

(i) Imperceptibility is tantamount to semantic and stylistic equivalence, i.e., there should be negligible difference in meaning and style between the original text and the marked text. We monitor the impact of text manipulations via edit-hits, as detailed in Section 4.1 and statistically in Section 3.3.
(ii) The watermarking payload is simply the number of bits that can be reliably and imperceptibly embedded in a text. This can be measured in terms of bits per lexical unit or bits per sentence. Note that, if error-correcting codes are used, a larger number of sentences must be embedded to carry the payload.
(iii) Robustness implies that a watermark inserted into a text should resist against malicious or unintentional text alterations. One way to achieve this is obfuscating the watermark by randomizing the choice of watermark tools and occurrences. Another way is to use error-correcting codes on the watermark message.

2.2. NLW categories

A watermarking tool is defined as a modification rule in the text with two or more alternatives, each assigned to a bit pattern. If a tool presents only two alternatives or if its modification rule is constrained to two alternatives, then it can be used to embed one bit of information. The NLW tools in the literature can be grouped into six main categories:

(i) Lexical substitution, which amounts to changing the words with their synonyms.
(ii) Semantic transformation, which makes use of word sense disambiguation, semantic role parsing and anaphora resolution.
(iii) Syntactic transformation, which is used to paraphrase a sentence by means of semantically equivalent sentence structures such as active–passive transformations, clefting and extra-position.
(iv) Translation, which creates different versions of a sentence in the target language;
(v) Punctuation modifications.
(vi) Simulated typographical errors, e.g., as in chat correspondences.

We review below briefly the main modalities of NLW in the literature.

Chapman and Davida (1997) were the first to introduce the lexical substitution method where word pairs such as ‘large/big’ are used. This simple method changes selected lexical items with their synonyms without operating on the sentence structure. Bolskhakov (2004), Taskiran et al. (2006), Topkara et al. (2006a), Atallah et al. (2000) have automated this method further by using parts of speech taggers and Winstein T-Lex (Tyrannosaurus-Lex)[1], and have made it more robust via Wordnet:Similarity. While the method of lexical substitutions is straightforward, an obvious weakness is the ease with which the pirate can inverse the embeddings. Furthermore, not all languages have profuse synonyms, and even if so, there may still be semantic and pragmatic differences among the members of the synonym sets. This requires robust word sense disambiguation

[1] Tyrannosaurus-Lex system available at http://www.fb10.unibremen.de/anglistik/langpro/nlgtable/nlg-table-root.htm.

that is presently available only for a few languages. For this reason, Bergmair and Katzenbeisser (2004) used a Human Interactive Proof system, which is based on the fact that machines cannot disambiguate senses of words easily, but humans can do so with a higher accuracy. Murphy and Vogel (2007a) notes that word sense disambiguation systems such as the one offered in Mihalcea and Csomai (2005) struggle to achieve more than 60% on general text. Topkara et al. (2006a) overcame the vulnerability of synonym substitution by using a quantitative resilience criterion for choosing a particular lexical item in a given synonym set. Taskiran (2006) obtained security enhancement by choosing a particular synonym in a given set based on the statistics of *n*-gram word strings rather than choosing the mere option, that is, the first form which native speakers would select. The success rate of this model is measured in terms of semantic equality using a trained Support Vector Machine (SVM) classifier, and is found to be 84.9%. In another work, Murphy and Vogel (2007b) have remarked that lexical substitutions achieve low payload. Wu and Stinson (2007) compare two approaches to NLW – one that is based on Text Meaning Representation (TMR) and one that is not – and find the TMR system to be more robust. Of the two techniques proposed by the authors to improve robustness, the first one uses meaning representation to rank the text sentences to determine those targeted for watermarking and literal representation to embed the watermark bit. In the second, the watermark is embedded on a random basis. Edit distance is used to search for an error-tolerant watermark.

The work of Atallah et al. (2002) belongs to the second class of NLW methods that use semantic transformations focused on word sense disambiguation, semantic role parsing and anaphora resolution. The properties of the lexical items are based on ontological semantics and lexical items are denoted with their semantic features such as [+/− animate, +/− human, +/− female, +/− wood, etc.] (Raskin and Nirenburg, 2003). They chose in text only semantic features without any impact on the sentence or that cause only slight meaning changes. These semantic features are added, deleted or moved around. A degree of security is achieved for this NLW tool by using false positives and random modifications on non-watermarked sentences in order to confuse the adversary. Semantic transformations necessitate some sophisticated tools, such as Wordnet: Similarity and Text Meaning Representations in order to implement deep semantic processing. However, these tools are not readily available for most languages and their outputs on semantic parsing, anaphora resolution and word sense disambiguation may not be very robust. Moreover, semantic role identification required by semantic transformation-based watermarking is still problematic, as reported by Murphy and Vogel (2007a).

The third and the most extensively used category of NLW is based on syntactic manipulations of sentences. Our work on morphosyntactic watermarking takes place in this category. Typical of syntactic transformations is the paraphrasing of a sentence by means of semantically equivalent sentence structures such as active–passive transformations, clefting and extra-position. This method is based on the fact that sentences are combinations of syntax and semantics, and the semantics of a sentence can be expressed by more than one syntactic structure (Liu et al., 2005; Topkara et al., 2006b; Murphy and Vogel, 2007a; Murphy, 2001; Meral et al., 2007). Some of the transformations, such as active vs. passive, are universal for the typology of languages that English and Turkish belong to, while there are language-specific structures that vary from language to language. Syntax-based manipulations obviously require more sophistication than lexical substitutions, and have to make use of such tools as parsers, Wordnet, electronic dictionaries and annotators. For example, Topkara et al. (2006b) have analyzed the sentences in Reuters Corpus with X-TAG and Charniak parsers, and converted the deep structure forms of the sentences into the surface structure formats via language generation tools such as DsynsS and Real Pro. One should note that NLW based on syntactic manipulations is not necessarily immune from semantic distortion.

The fourth NLW modality is translation. In this method the different permissible transformations of sentences in the source language are used to create alternatives of the original text, and then one of the possible translations is chosen to mark the watermark into the text (Grothoff et al., 2005; Stutsman et al., 2006). By using machine translation software to create alternative translations of a given text, Stutsman et al. (2006) have attained 0.33 bit/sentence watermarking payload. For the security aspect of the watermarking method, they have made use of human translators alongside machine translations.

We have not encountered any NLW study based on punctuation alterations. This may be due to the low expected payload and to the instability of punctuation in languages. For example, in Turkish comma can be used after the subject if it is relatively distant from the verb. However, this rule is not consistently enforced in all texts.

Topkara et al. (2007) present an interesting sixth class of NLW, where typos are seeded into the original text as if they occurred naturally during transcription. These are generated by computationally ambiguous transformations. Words that have multiple typo potential are selected for watermark encoding. For instance, for the word "cake", which has more than one possible typo such as 'ake' 'cakw' 'axe' and so on, the typo 'ake' is preferred over 'cakw' since the latter has a smaller correction list as compared to 'ake'. This ambiguity on corrections of typos is also instrumental in confounding the adversary who wants to guess/destroy the watermark by corrections (Topkara et al., 2007). One advantage of this method is that semantic equality of the watermarked text is guaranteed since typos do not perturb the meaning of text and humans are good at disambiguation. Furthermore this method does not require software tools such as parsers, language generators or Wordnet since watermark embedding and detection are done via typo confusion matrices. One disadvantage is that typos are visible to the adversary and steganography, which contradicts the aim of information hiding while ensuring discreteness.

2.3. Morphosyntactic watermarking

It is well known that the structure of sentences is determined by syntactic rules while the meaning is determined by the semantic component of the grammar. In theory, a given linguistic structure has a particular meaning defined by the related syntax–semantics combination. In practice, however, all languages possess forms that carry very similar or identical semantic interpretations and yet they have different syntactic structures. For example, an active sentence leads to the same semantic interpretation as its passive counterpart disregarding some pragmatic differences. In another instance, a temporal modification in a sentence can be expressed via a lexical item or via a suffix form. For instance in English, the linguistic notion 'possession' can be expressed either with the preposition 'of' or with the suffix '-s', e.g. "The book of John" versus "John's book". Although this sort of alternatives is limited and has sometimes semantic and pragmatic consequences in English, they occur profusely in some other languages such as Turkish. It is these types of syntactic form alternatives in natural languages, in a sense morphosyntactic redundancy, that enable the application of watermarking. Thus, when a proposition can be expressed in more than one syntactic structure, each case can code a particular bit combination. In the above two instances, active and passive forms can be mapped to logical 0 and 1, or vice versa; similarly, lexical and suffix-based modification can encode logical 0 and 1 bits. If M is the number of syntactic alternatives, then they can be used to encode $\log_2 M$ bits. Most often, however, one has $M = 2$.

We consider a (morpho)syntactic tool to be any syntactic structure that allows for variants under minimal semantic change. Table A1 in Appendix A contains 20 instances of morphosyntactic tools for Turkish and five such tools for English. The illustrative sentences in this table show the structural alterations of the original text by morphosyntactic transformations.

In Turkish the nominal morphosyntactic features such as case, person and number agreement information, and verbal features such as tense, aspect, modality and voice information are coded on the lexical categories. For instance, the syntactic Tool #5 in Table A1 includes the alteration of the linguistic form [Verb-NOUN-POSS when] with another linguistic form [verb-NOUN-POSS-LOC].

All the tools listed in Table A1 are intended for bidirectional or duplex use. That is to say, they have to be applied in the forward and backward senses to yield binary alternatives. For example, the adverb can precede or follow the subject. For the simplicity of the discourse, we will declare one direction of the tool as "forward" (say, adverb precedes the subject in Tool #2), and the other direction as "backward" (adverb follows the subject), although obviously these assignments are totally arbitrary. In the sequel, we illustrate these syntactic manipulations via syntactic tree diagrams where one tree variety is mapped to one logical bit value, while the synonymous variety is mapped to the alternate bit value. We have included for comparison purposes in this list the tools used by Topkara et al. (2006b) and Murphy and Vogel (2007a) for English. Though these lists do not constitute exhaustive inventories for the respective languages, nevertheless they provide sufficient arsenal to prove the feasibility of (morpho)syntax-based NLW.

One can observe in Table A1 that tools have widely varying occurrence probability. Also nearly a quarter of sentences do not permit any watermarking action (last row in the table), that is, none of the 20 tools are applicable, except word order change. In contrast, with the use of the word order change tool, the percentage of unproductive sentences reduces to 1.8% from 24%. Transformations based on word order alternations are common to many languages (Göksel and Ozsöy, 2003). Word order changes are generally related to discourse structure. English, for example, allows topicalization, which moves constituents to sentence initial position. Thus, in the sentence "John said Jane is not happy", "Jane is not happy" can be topicalized yielding the output "Jane is not happy, John said". This transformed sentence can be mapped onto logical bit 1 while the original sentence is mapped to 0. Languages vary with respect to the degree of word order alternation they license. In this respect Finnish, Turkish and Latin allow relatively the greatest freedom, while Russian, Polish, Hungarian and Czech languages allow some word order manipulations.

A study of Turkish (Slobin and Bever, 1982) has revealed the following statistics in the constituent ordering of a sentence: SOV: 48%, OSV: 8%, SVO: 25%, OVS: 13%, VSO: 6% and VOS $\ll$ 1%. Thus one scheme to execute watermarking via word order changes is to consider a two state machine, where one state is SOV (occurring with 48% chance) and the other state amasses the SVO, VSO, VOS, OVS and OSV cases (occurring with 52% chance). An SOV sentence is permuted to one of the non-SOV cases with transition probability of the corresponding cases. For example, the SOV → OVS transition probability should be made 13% to preserve the natural trends of the language. The watermark decoder considers all SOV sentences to be one mark value and any of the non-SOV states as the other mark value. In the hypothetical case of a language where all orderings occurred with equal probability, the 6-state transition diagram would have enabled to embed $\log_2 6$ bits/sentence. In this work we did not make full use of the word order alternation tool (Tool #20), but restricted it to Adverbial Displacement (Tool #2 in Table A1). Turkish adverbs can precede or follow the subject. Given that adverbs have their lexical semantics as well as semantic connotations due to their positioning, Meral et al. (2007) have limited adverb displacements to a set of 20 temporal adverbs (today, in the evening, in the morning, at night, tomorrow, next/last year, the following day, this century, etc.).

The active-to-passive transformation is common to languages that mark subjects nominative and objects accusative. The transformation has an effect on the argument structure and the morphology of the verb. In the active-to-passive direction, the transformation reduces the number of arguments of the verb by one. In the passive-to-active direction, the transformation has the reverse effect: it increases the number of arguments by one. Languages also exhibit crosslinguistic variation in the productivity of the transformation in that while some languages restrict the transformation to transitive verbs only, others also allow intransitive verbs to passivize. English is an example of the former type, Turkish and German of the latter.

NLW analysis of the database sentences showed that 31.5% of them could be subjected to active-to-passive transformation. Since passive sentences are not always liked we had to use active–passive transformation with parsimony. We lowered the rate of active sentence conversion into passive sentence to 3.2% in order not to perturb the distribution of the two structural variants within the text. To ensure imperceptibility of the tool, we employed the following heuristic to restrict the active-to-passive transformation. Only those sentences in which the head noun and the verb are found in both active and passive relations in the data were targeted. The application of the tool converting a passive sentence into its active counterpart is naturally restricted only to sentences which have an agent phrase. Therefore additional means of constraining the active-to-passive transformation to equalize the distribution of the active and passive constructions in the text was achieved by limiting the input sentences to those which have verbs whose passive counterparts occur with animate subjects within the text.

3. Watermark embedding and detection

The rationale of our watermarking method is to weave through the text by applying feasible syntactic watermarking tools to sentences according to a scheme. An input text is first parsed and transformed into its hierarchical representation, and then syntactic tools are selected and operated on the individual sentences. In the simplest model, we assume that the watermark consists of a sequence of m bits to be embedded in a text of n sentences, $n > m$. The m bits are embedded sequentially in consecutive sentences that are found to be watermarkable. Alternatively, for security and stylistic control, watermarking occasions can be skipped, that is, one can leave a watermarkable sentence un-watermarked. This is called the pass option, and this option also takes place in the round-robin (see Sections 3.2 and 3.3) as if it were a watermarking tool. Finally, the sequential presentation of watermark bits can be abandoned in favor of an interleaved presentation of these bits over paragraphs as a protection against burst errors (see Section 3.3).

The details of the watermarking algorithm at the sentence level and text level are described below.

3.1. Sentence-level preprocessor for watermarking

The preprocessing of the input text for natural language watermarking is described in Fig. 1, where the curly boxes denote the text in its successive stages of processing and the rectangular boxes correspond to operators.

The annotator operator serves to parse the sentences into smaller categories with mutual dependencies. The annotator (Oflazer, 2003) processes the raw sentences with morphosyntactic features and functional dependencies, and outputs the treebank representations of its sentences as shown in Fig. 1. For treebank representation, one first extracts morphosyntactic features (i.e., case features of nouns, tense and person features of verbs) of the lexical items, and then marks functional dependencies of the words, such as subject, modifier and verb (Eryiğit et al., 2006; Eryiğit and Oflazer, 2006). The treebank representation of sentences is not always suitable for the application of the watermarking tools. For instance, Tool #3 (see Appendix A) necessitates a deeper syntactical representation, which makes explicit the functional dependencies on larger constituents such as phrases and clauses. We developed a tree transformer that converts the treebank format to a syntactic tree structure. The transformer represents the words in a treebank sentence as nodes in a hierarchy and builds the parent–child relationships between these nodes according to the functional dependencies among the words. In this way, the phrases are grouped and the hierarchical relationships between the phrases, rather than the dependencies between individual words, are obtained. We note that transformation of treebank sentences into hierarchical representation is crucial for correct operation of watermarking tools.

3.2. Text-level watermark embedding

The morphosyntactic watermarking tools are applied according to a randomization regime to input text that has been annotated and transformed sentence by sentence. The watermark embedding is achieved in three steps: (i) probing of the sentence for permissible syntactic tools, (ii) choosing pseudo-randomly one of the tools, if any available, under stylistic and security constraints, and (iii) modifying the text according to this morphosyntactic tool. These steps are illustrated in Fig. 2 and explicated below:

> *Watermark Tester*: This operator takes the parsed text and checks for the applicability of NLW tools sentence by sentence. When the morphosyntactic features in a sentence enable a watermarking tool, it is added to the list of applicable tools for that sentence. For example, the sentence "I ate a lot" does not allow any of the tools listed in Table A1, while the sentence "Yesterday, when Ayşe and Ali were carrying the furniture I came home and changed the lock" allows the application of as many as four tools in Table A1 (namely, adverb displacement, conjunct order change, Verb1 and verb2/verb+(y)Ip verb2, Verb-NOUN-POSS when/verb-NOUN-POSS-LOC) and pass tool.

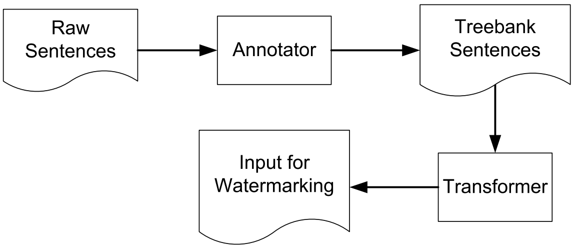

Fig. 1. Preprocessing of the text for sentence level watermarking: annotator parses sentences into mutually dependent lexical groups; transformer converts dependencies at the clause and phrase level.

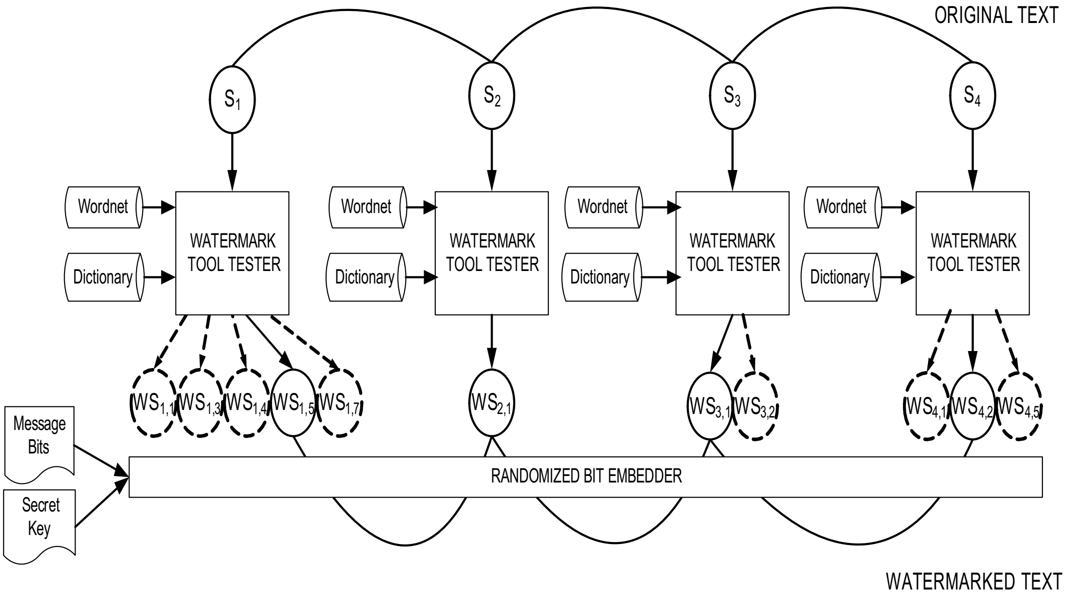

Fig. 2. The watermarking algorithm at the text level: S_1, S_2, S_3, S_4 stand for the sequel of input sentences. The watermark tester checks and lists the tools applicable for each sentence. The sentence sequel $WS_{1,5}$, $WS_{2,1}$, $WS_{3,1}$, $WS_{4,2}$ constitutes the watermarked text in this instance. The chosen tool $WS_{i,j}$ is shown with a solid circle.

Watermark Selector: This operator selects one tool from the pool of watermarking tools. This choice is dictated on the one hand by stylistic concerns (e.g., passivization of sentences in a sequel may be disturbing) and on the other hand by security concerns. A measure of security is obtained by switching between tools in some random order, under the control of a secret key shared between the encoder and decoder sites. Alternative schemes would be to select the tools in a round-robin fashion or apply some sort of weighted fair queuing, with weights possibly inversely proportional to their occurrence probability to balance their appearances.

Watermark Embedder: This operator transforms the sentences in order to embed a watermark. The chosen morphosyntactic change takes place if the watermark bit to be embedded and the state of the sentence do not agree; else, no change is made in the sentence. For instance, assume that the tool chosen is the Tool #2, "Adverb displacement tool", that is, the adverb position vis-à-vis the subject. Let this adverb rule be as follows: adverb preceding the subject is "1" and adverb succeeding the subject is "0". Now suppose that the message bit is "0": if the actual sentence has already its adverb after the subject, then no action takes place; otherwise, the subject and the adverb swap their positions.

We illustrate the procedure, in Fig. 2, where the notation $WS_{\text{sentence,tool}}$ indicates that the test sentence possesses that specific tool numbered according to the list in Table A1. For example, sentence 1, S_1, possesses the set of five tools (1, 3, 4, 5 and 7) or $WS_{1,1}$, $WS_{1,3}$, $WS_{1,4}$, $WS_{1,5}$, $WS_{1,7}$. The watermark randomizer algorithm chooses one of these tools (say, $WS_{1,5}$). Finally, the embedder watermarks that sentence (embeds 0 or 1 bit value to the sentence) based on the status of the sentence vis-à-vis Tool #5. This procedure is repeated for each sentence. A sample four-sentence long text is shown in Fig. 2, which contains the sequel of $WS_{1,5}$, $WS_{2,1}$, $WS_{3,1}$, and $WS_{4,2}$ tools. Each such possible sequence is called a "watermark path".

The watermarking algorithm described in Fig. 2 includes finer but crucial arrangements. Morphosyntactic tools cannot be applied automatically since one can incur into semantic and pragmatic problems resulting in unconventional and ungrammatical cases. We try to overcome this problem by incorporating Wordnet (Bilgin et al., 2004) and a dictionary (Turkish Dictionary, 2005) into the watermarking process. We remark that there has been some criticism on online lexical resources such as Wordnet. However, we took the large frequency with which Wordnet is being referenced as a token of its utility and reliability. These language resources provide semantic and pragmatic extensions of lexical items. For example, Wordnet can indicate whether a noun is semantically sensitive to a watermarking manipulation. A case in point is animacy and passivization, since intransitive verbs with inanimate subjects resist passivization, while intransitive verbs with animate subjects can undergo passivization freely in Turkish. Likewise, if the application of a particular syntactic rule is known to be sensitive to the voice features of a verb (e.g., transitive), then the dictionary is consulted for the relevant information. In summary, semantic properties of lexical items are considered before allowing the application of a syntactic tool for watermarking.

The whole watermarking procedure is illustrated in Fig. 3. The preprocessed sentences in tree form are fed into the watermarking algorithm.

The text-watermarking encoder functions as follows:

Step 1: Input the test sentence to the syntactic parser which outputs its dependency chain
Step 2: Transform the dependency chain of the sentence to yield a parsed syntactic tree
Step 3: Check out the availability of tools and select "randomly" one tool from the pool, taking into consideration the past record of tool usage
Step 4: Implement the tool(s): if the watermark message bit agrees with the condition of the sentence with respect to that tool, then leave the sentence unchanged; else, apply the syntactic alteration.

Similarly, the watermark extractor functions as follows:

Step 1: Input the possibly marked sentence and run the syntactic parser
Step 2: Obtain the parsed syntactic tree
Step 3: Estimate the pool of potentially applicable tools and determine the one(s) that must have been applied according to the randomizing or round-robin scheme
Step 4: Check the direction of the tool: if the tool was applied in a forward manner, then decide for "1"; if it was applied in the backward sense, then decide for "0".

3.3. Robustness and security of NLW

The robustness of NLW is defined as its resistance to innocent text alterations and to malicious attacks intended to invalidate the hidden message. The adversary can tamper with the text in order to make the hidden message inaccessible. These attacks could consist of detection of morphosyntactic occurrences and their random alterations, creation of arbitrary embeddings for obfuscation, and insertion or deletion of sentences to desynchronize the algorithm.

Taskiran et al. (2006) have observed that lexical substitutions are not particularly strong against attacks since they can be sensed and destroyed by further substitutions. As a countermeasure, Atallah (2000) makes false positives and random modifications on non-watermarked sentences while synonym pairs are chosen using a quantitative resilience criterion by Topkara et al. (2006a). In another work, lexical substitutions are used alongside syntactic transformations to mislead the adversary (Topkara et al., 2006b). A statistical model that estimates the *n*-gram word strings for their synonym substitutions is proposed by Taskiran et al. (2006).

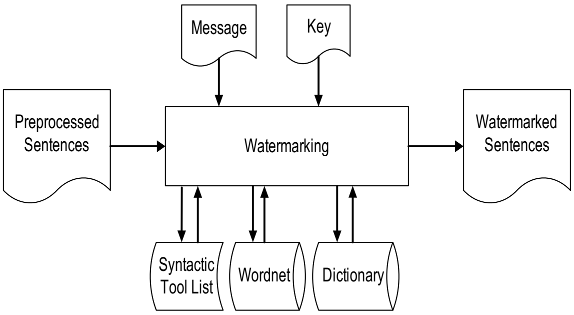

Fig. 3. Syntactic sentence-level watermarking: preprocessed sentences are in the parsed syntactic tree form.

These authors, having measured the frequency of each synonym in corpora, select the less frequent ones to alter the lexical items in the original text. For translation-based methods, Stutsman et al. (2006) provide both machine and human translations as a means to confuse the adversary. Finally, Atallah et al. (2002) further expand on false positives and random modifications introduced to a number of sentences that are not used for watermark embedding in order to prevent attacks by the adversary.

In our scheme, the security of the watermarked text is enhanced in two ways: (i) the pseudo-random order of tool selection, and (ii) the insertion of a "pass" tool creating void watermarks. As the text is woven with watermarking modifications, no apparent fixed watermarking pattern should be perceived. This is not only necessary from a stylistic point of view to avoid repetitive or patterned changes in the text, but also to preclude any hint of watermark presence to the adversary. The ordered application of watermarking tools can damage the naturalness of a text by creating a predictable sequence and it limits the number of feasible paths. Consider a text that contains n sentences such that the ith sentence permits λ_i morphosyntactic options. We take $\lambda_i = 1$ when the number of options is zero. Then the number of possible watermarking paths is given by $\prod_{i=1}^{n} \lambda_i$. For example, in the four-sentence long text in Fig. 2, one can conceive $5 \times 1 \times 2 \times 3 = 30$ paths (arrows in the lower part of the figure). Accordingly, the number of watermarked text alternatives increases geometrically with the text length, which in turn helps to obfuscate the watermarking or watermarked message.

Second, a "pass" or "escape code" is allowed as part of the tool list. This gives the option of not watermarking despite the availability of one or more morphosyntactic tool for that sentence. The escape code is selected randomly as all other tools according to the secret key. Topkara et al. (2006b) made a similar option possible by the insertion of fixed synonym substitutions. We emphasize again that the void case is essential for the obfuscation of the watermark. Furthermore, we can control the sparseness of the watermarking at the sentence level by adjusting the frequency of the void case. The sparser the embedding, the more massively the adversary has to do re-watermarking in order to corrupt the hidden message.

The security aspects of the proposed NLW scheme can be discussed under two scenarios, which are best explained in terms of the prisoner's problem (Simmons, 1983). The two inmates, Alice and Bob, wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of covert communication. Wendy can act as a passive warden or an active warden, as detailed below.

Passive warden case: Wendy controls all communication in the public channel, that is, published text, in order to detect any trace of secret embedding. If she detects the presence of hidden information she will destroy the message, hence impede the subliminal channel. This is the steganography problem where data must be hidden in such a subtle way that it should not arouse any suspicion.

The NLW scheme can function as a steganographic channel if its perceptibility is very low. The passive warden case is simplified if we assume as in (Bergmair and Katzenbeisser, 2005) that Wendy is a computer, while Alice and Bob are humans who have a language model superior to Wendy's. The passive warden case can be handled in two ways:

- For Wendy, a person, the security issue can be approached from a linguistic point of view. In this respect, the imperceptibility or naturalness measure of the stego-text can be based on empirical evidence of editing endeavors of test subjects on sample texts. We had noticed that subjects have editing urge even for non-altered sentences at a rate of 8.4% (see Table 2). We consider this percentage of editing on arbitrary non-watermarked texts as the noise floor and we can argue that the overall text will be steganographically safe if the sentences are watermarked with edit-hits below this floor. This lower level can be achieved by decreasing the embedding rate and by sparsifying the occurrence of disliked patterns like passivization. For example, in the randomized weighted fair queuing, the weight of the pass code can be increased so that syntactic modifications occur more sparsely.
- For Wendy, a person or a computer, the security issue can be approached statistically. According to the Kerchkoff's principle we assume that Wendy is capable of knowing the statistics of the cover-texts and all the embedding tools (the repertoire of morphosyntactic tools). She cannot decode the watermark message since she is not in possession of the secret key, since the secret key between Alice and Bob has been exchanged via a secure channel. However Wendy can attempt to statistically estimate the presence of NLW. Wendy possesses enough training material to measure the statistics of morphosyntactic tool occur-

rences both in the cover-text and in the stego-text. If these two probability vectors differ by more than a chosen δ-distance, then Wendy will destroy the message. Let $\{p_t\}$ be the set of occurrence probabilities of the morphosyntactic tools (as in Table A1) for cover-texts, and $\{q_t\}$ the same set for a stego-text. Notice that tools typically do not have symmetric probabilities. For example in Tool #7 in Table A1 "Verb-TENSE-AGR because1" occurs about 2.6 times more frequently as compared to the converse tool "verb-NOUN-POSS because2". Therefore p_t and q_t have dimensionality $2T$, where T is the number of tools used. There are several ways to measure the distance between the two probability vectors, for example, chi-square distance or the Kullback–Leibler distance. When we use the Kullback–Leibler distance (Cachin, 2004), one has:

$$\left\{ \begin{array}{l} \sum_t p_t \log \frac{p_t}{q_t} < \delta \quad \text{no stego message} \\ \text{else text contains a stego message} \end{array} \right\}$$

We actually ran a watermarking experiment on a text containing 499 sentences. We have used all the nine tools from Table 2. The resulting Kullback–Leibler distance is found as 0.09. One way to qualify this distance figure is to assess it within a lossless source-coding framework. Consider the equality: $R_q = \sum_t p_t \log q_t = H(p) + D(p\|q)$, where $H(p)$ is the lossless coding rate of an encoder with true probabilities p_t, and where q_t are the guessed probabilities. The lossless source encoder would incur into a penalty of $D(p\|q)$ bit/symbol. If we normalize this loss term to the lossless coder rate $H(p)$, we obtain $\frac{D(p\|q)}{H(p)} = 0.03$, meaning that per symbol the encoder must spend 3% more bits. Obviously this is a negligible quantity.

Active warden case: In this mode, Wendy will occasionally try to disrupt the subliminal channel by random attacks just to thwart any possibility of secret communication between Alice and Bob. These random attacks consist of random watermarking of sentences. Actually, there is another type of attack outside the prison scenario, which is impersonation attack. In this mode, a pirate tries to invalidate the watermarked text by re-watermarking it to claim ownership. Since the pirate does not possess the secret key, the likelihood that he produces a correct m bit authorship signature is 2^{-m}, hence false positive probability is very small for $m \gg 1$ (Wu and Stinson, 2007). Notice both the active warden and the impersonator try to invalidate the watermark by increasing the false negative probability; therefore we can discuss both attack types under the umbrella of jamming attack. The security of the watermarking scheme then hinges on the likelihood that the jammer can destroy it, that is, invalidate the proof of authorship.

We assume that the text consists of n sentences, and that m of these sentences have been watermarked. In other words, the watermarking density is $\rho = m/n$. We assume that the watermark sequence is protected by some (m,k,t) block code, such that k information bits are encoded into m sentences, and the error-correcting code is capable of detecting and correcting up to t errors. The jammer mounts an attack by randomly selecting c sentences out of n and operating one of the available morphosyntactic tools. The jammer will succeed only if he hits and destroys at least t of the watermark bits in his c sentence attack. In other words, he must be lucky enough to have selected at least t sentences out of the m watermarked ones in his attack with c sentences and each time have hit the correct watermark tool within sentences. Let us denote the false negative probability as $P_{n,m,c,t}$, where for the sake of clarity we repeat: $n =$ #Total sentences, $m =$ #Watermarked sentences, $c =$ #Attacked sentences, $t =$ #Watermarks to be destroyed. The probability of the jammer's success, which is tantamount to the probability of a false negative, can be calculated recursively, since it is a case of sampling without replacement:

$$P_{n,m,c,t} = \frac{n-m}{n} P_{n,m,c-1,t} + \frac{m}{n} P_{n,m,c-1,t-1}$$

The first term $\frac{n-m}{n} P_{n,m,c-1,t}$ is the miss probability and the jammer is left with $(c-1)$ more shots; the second term $\frac{m}{n} P_{n,m,c-1,t-1}$ denotes the hit probability. In the latter case the jammer needs to hit $k-1$ more targets with $c-1$ available shots. We have also the following boundary conditions: $P_{n,m,1,1} = \frac{m}{n}$; $P_{n,m,c,t} = 0$ for $c < t$. For example, consider the case of a non-perfect code, BCH(63,36,5) code, where 36 information bits are embedded within 63 code bits, and up to and including $k = 5$ errors can be corrected. Fig. 4 displays the probability of jammer's success as a function of his attacks, and where $n = 1024$, $m = 63$, hence $\rho = 0.062$.

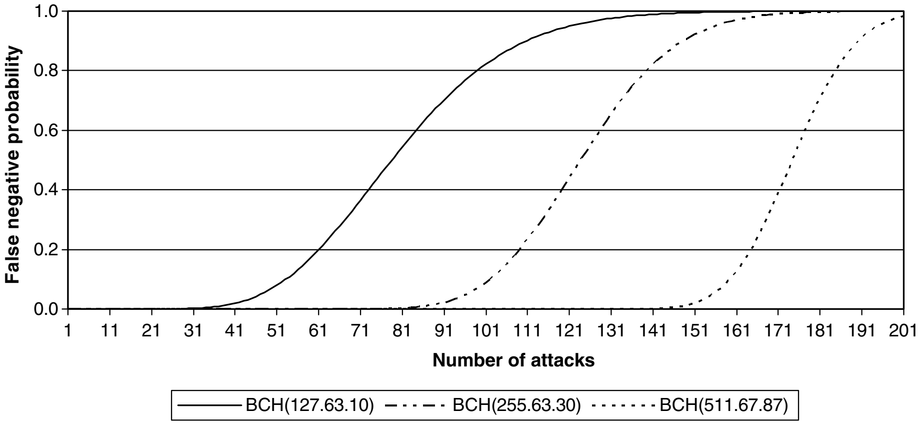

Fig. 4. False negative probability as a function of jammer attacks.

Table 1
Jammer resistance of the watermarking scheme as a function of the error-correcting code rate

Code type	Text length n	Codeword length M	Number of jammer attacks	Number of corrected errors k	Payload density
BCH(255,63,30)	1024	255	108	31	0.062
BCH(127,63,10)	1024	127	60	10	0.062
BCH(511,67,87)	1024	511	164	87	0.066

Probability of false negative is 0.2.

In Table 1 we show the performance results. The important quantity here is the number of attacks c that the jammer must mount in order to invalidate the message. We experimented with various codes of BCH and Goppa variety (Lin and Costello, 2004), but we report the BCH code results. We took the message length as 64 bits (this suffices for an ISBN number). We observe, for example, that when we use BCH(511,67,87) code, the jammer must attack at least 164 sentences on the average to destroy the watermark. Notice that this is a pessimistic result in that every watermarkable sentence sometimes offers more than one tool and the jammer has to pick up randomly one of them. The last column is the ratio of watermark source bits to the number of sentences of the text in the role of “carrier”.

Desynchronization via sentence deletions and insertions: Desynchronization of watermarked multimedia, such as resulting from geometric attacks on images, has proven the most difficult case to combat in the literature (Moulin, 2006). In a similar way NLW may suffer from loss of synchronization after sentence insertions and/or deletions, and the watermark bits may not be decoded correctly until synchronization is re-established. This situation resembles to a burst of errors in digital communication, since decoded bits in between two sync marks can be erroneous. There are two ways to mitigate the effects of desynchronization. First, the watermark sequence can be interleaved in the text so that a burst affects many fewer bits of a given sequence, that is, the burst is dispersed to appear like random channel errors. In other words the watermark sequence is randomly distributed to different portions of the text, say, over paragraphs. Consequently the probability that an error burst due to a desynchronization attack will destroy a group of bits in a sequence is much diminished. Second, error-correcting codes are used which, in association with the interleaver, can correct random errors. Finally, in order to gain synchronization, we can insert sync anchors in the text. Atallah et al. (2001) have used the lowest ranked sentences to function as sync markers. We propose to use n-tuple successive pass actions for synchronization. For example, three consecutive watermarkable sentences can be skipped to generate a synchronization mark, reminiscent of triple ACK mechanism in the TCP/IP protocol. If deletions/insertions happen to hit these particular sentences, one can hope to be synchronized in the next triple pass.

4. Watermarking performance

We tested our algorithm on two corpora consisting of a total of 5836 sentences. The first is METU-Sabancı Treebank (Oflazer et al., 2003; Oflazer 2003) that has been compiled from 16 main genres with a good balance across the genres. The second corpus is the İTÜ Treebank (Istanbul Technical University) (Eryiğit, 2007) that consists of three genres (20% newspaper articles, 20% novels, and 60% short stories). The sentences were processed using a dependency parser, which is usually superior to rule-based parsers for free word order languages. Each sentence in the treebank is represented in XML format from which the dependency relations between the words can be extracted.

Table 2 gives the statistics of watermarking manipulations on sample texts.

– The column "tool" lists the nine tools we have used out of 20 feasible tools, as the rest had very low occurrence percentages.
– The column "occurrence in corpus" lists the occurrence probability of morphosyntactic tools per sentence in a corpus of 5836 sentences (the same occurrence frequencies as in Table A1). Notice that the occurrence probability of morphosyntactic tools is quite lopsided. For example, the three tools of active–passive alternation, adverb displacement and conjunct order change have respective forward probabilities of: 31.5%, 14.8%, and 12.4%. However, the active–passive transformation must be used with great parsimony as we discussed in Section 2.3.
– The last row in the table lists the unproductive cases, and corresponds to the percentage of sentences that do not possess any applicable morphosyntactic tools, as in a short sentence "John came".
– If we sum the occurrence frequencies of the watermarking tools, we obtain the average number of tools available per sentence, that is, 1.07 tools/sentence of a text bidirectionally. Since the percentage of watermarkable sentences is 76%, if we consider the tools available per "watermarkable sentence", this figure becomes $1.07/0.76 = 1.40$. In addition, if we also allow word order changes as a watermarking tool, we reach $1.07 + 0.97 = 2.04$ tools/sentence. Finally these frequencies indicate that the pass tool can be applied, when needed, in all watermarkable sentences.
– The column "edit percentage" shows the editing attempts for each tool, that is, the percentage of time that acting editors felt the urge to modify and correct the sentence bearing a specific watermarking tool. Section 4.1 discusses these points in a detailed way.

In the rest we report on two important performance measures for watermarking, namely, imperceptibility and capacity.

4.1. Imperceptibility of NLW

Imperceptibility of NLW can be interpreted as semantic and stylistic equivalence of the cover- and watermarked texts. Various metrics have been used for the preservation of the semantic content during watermarking transformations.

Table 2
The occurrence frequency (F: forward, B: backward) and edit-hit rate of syntactic watermarking tools

#	Tool	Occurrence % in corpus (5836 sentences)		Application % (499 sentences)		Edit percentage
		F	B	F	B	
1	Active/passive voice	31.5	0.3	3.2	2.0	12.5
2	Adverb displacement	14.8	11.6	8.7	6.7	7.4
3	Conjunct order change	12.4	12.4	8.2	8.2	6.4
4	Verb-NOUN-POSS when/verb-NOUN-POSS-LOC	2.9	1.4	0.2	1.8	1.7
5	Verb-TENSE-AGR because1/verb-NOUN-POSS because2	2.6	1.0	0.2	1.0	15.4
6	Verb-PARTICLE be-NOUN-POSS/verb-NOUN-POSS	0.1	1.5	0.4	2.2	1.2
7	Verb1 and verb2/verb+(y)Ip verb2	3.3	5.6	4.8	8.0	8.2
8	Verb-NOUN1-POSS although1/verb-NOUN2-POSS-DAT although2	0.3	0.07	0.2	0.4	0.0
9	If...verb-COND/verb-COND	0.7	4.9	0.2	0.8	2.4
10	Unproductive	12.0	12.0	13.0	13.0	8.4
	Average percentages	68.6	38.7	26.1	31.1	6.1

Topkara et al. (2005, 2006b) measure the watermarking imperceptibility in terms of machine translation software by which the two sentence varieties have been evaluated for the semantic equality. They resort to the MT (Machine Translation) evaluation Tool Kit of NIST to measure the correctness or the semantic equivalence of two different texts. They achieve a score of 0.47 on this toolkit where the top score is reported to be 0.51. This is measured by Topkara et al. (2006a) as the maximum distortion on the original text by using Semantic Concordance (SemCor2.1) (Landes et al., 1998) in which 1815 tagged forms are available for synonym substitution. The synonyms are ranked in distance from the reference word, and the distortion in a watermarked text is limited by allowing only substitutions up to rank order 15. Other imperceptibility measures are based on human judgments on the semantic equality of the two varieties. Murphy and Vogel (2007a) use judgments of native speakers, which is similar to the method of Meral et al. (2007), where in addition edit-hit statistics is used. In contrast, Grothoff et al. (2005) sacrifice the semantic equality of the watermarked text to the ability to deliver the secret message and the stealth of the communication and/or subliminal communication. SRILM (Stanford Research Institute Language Modeling) language model trainer[2] has also been used (Taskiran et al., 2006). SRILM is a statistical language model, which entails the estimation of model parameters from a collection of training text data.

Our imperceptibility or naturalness measure was based on subjective empirical evidence. Since syntactic watermarking transformations can perturb the naturalness and style of the text, we measured the palatability of the watermarking tools by having a set of subjects evaluate given texts and do editing corrections. In other words, subjects were given watermarked texts and asked to edit them for improved intelligibility and style, that is, to act like editors for articles submitted for publication to some journal. This is a blind test because the subjects were not told that watermarking had taken place nor did they in all likelihood know anything about NLW. When the subjects edited the "watermarked" article, some of their editing actions hit the watermarked morphosyntactic forms. The less a tool receives edit-hits, the more imperceptible it is deemed. Conversely, the forms that received many hits apparently are not to everybody's taste. In the experiment, three watermarked texts consisting of a total of 499 sentences were used and we received overall 32 processed texts from native speakers.

The "edit percentage" column in Table 2 shows the editing attempts for each tool, that is, the percentage of time that acting editors felt the urge to modify and correct the sentence bearing a specific watermarking tool. Lack of editing efforts implies imperceptibility of the tool's operation. These percentage figures are relative, that is, with respect to the number of occurrences of the tool. To give a concrete example, a text of n ($n \gg 1$) sentences will encounter on the average $0.154 \times n$ adverb displacement opportunities. If S "editors" mark the text, then they would object statistically to 7.4% of these displacements according to Table 2. In our tests, we have exposed $0.154 \times n \times S = 0.154 \times 499 \times 32 = 2459$ adverb displacement cases where n indicates the number of sentences and S the number of evaluators.

Note that these imperceptibility statistics result from stylistic concerns rather than ungrammaticality or semantic anomaly of the sentences. A tool becomes viable only if it is grammatically applicable and can surmount all stylistic and semantic constraints that impede its usage. Active-to-passive tool for instance has 31.5% occurrence frequency but has been applied only to 3.2% sentences due to the fact that we had observed that in an uncontrolled text this tool received the highest edit-hit rate. This suggested that this tool must be applied with parsimony. It is also interesting to note that sentences that include null watermarks (sentences which have not been transformed) have also received edit-hits at a rate of 8.4%, in this blind test. We conjecture that, if proven on a much larger corpus, this rate can be interpreted as the "noise floor". In other words, only the morphosyntactic tools that receive reactions above this level can be labeled as objectionable.

4.2. NLW payload

The watermarking payload of NLW can be quantified in terms of bits per language unit, for example, per sentence or per lexeme.

The NLW payload figures attained in the literature are almost always less than one. For example, Topkara et al. (2005, 2006a) have achieved 0.67 bit/sentence with lexical substitutions, despite the plethora of synonyms

[2] Stanford research institute language modeling toolkit available at http://www.speech.sri.com/projects/srilm/.

in English. The payload achieved with syntactic transformations dwells in the [0.5–1.0] range. Both Atallah et al. (2001) and Topkara et al. (2006b) have attained embedding payload of 0.5 bit/sentence with syntactic methods. Atallah et al. (2002) in another work have pushed this embedding limit higher by resorting to ontological semantics, though this method remains as yet impractical. The method of interleaving typographical errors among lexical items (Topkara et al., 2007) has targeted 1 bit/word, though it remains fragile to adversary attacks. Finally, Stutsman et al. (2006) show that translation-based watermarking has a payload of 0.33 bit/sentence.

In this work we measure the watermarking payload, C, as the number of embeddable bits within a given text or the embedding rate per sentence. Let p_{unp} be the probability of sentences devoid of any watermarking tool, such as in the sentence "I ate"; p_{pass} the probability of a null watermark placed purposefully, i.e., simply skipping a sentence with watermarking opportunity; p_q the probability of morphosyntactic tool with q alternatives; and finally $r = m/n$ the rate of the error-correcting coding. The payload is then given by

$$C = r(1 - p_{\mathrm{unp}} - p_{\mathrm{pass}}) \sum_{q=2}^{\infty} p_q \log_2 q$$

However, since almost all morphosyntactic tools allow only up to two alternatives, this formula is simplified to $C = r(1 - \mathrm{p}_{\mathrm{unp}} - p_{\mathrm{pass}})$. The watermark skip probability p_{pass} is a user-defined parameter and it serves to balance the two conflicting criteria of payload and security: increasing p_{pass} makes the system more secure against malicious attacks at the expense of decreasing the bit payload. Assuming that there are no pass decisions and no error-correcting code has been used, that is $p_{\mathrm{pass}} = 0$ and $r = 1$, the watermarking payload is upper bounded by $C = 1.07$ bit/sentence without resorting to the free word order alternation stratagem. In our sample corpora, nearly one fourth of the sentences, typically short ones, do not admit any watermarking (i.e. $p_{\mathrm{unp}} = 0.24$). Hence, the watermarking payload of the proposed method was observed as $C = 0.76 * 1.07 = 0.81$ bit/sentence. In other words, on the average 0.81 bit can be embedded within a sentence. This would be payload rate in a data hiding application. On the other hand, in a watermarking application for proof of ownership, when an error-correcting code is used and the watermark embedding density is lowered, the payload rate diminishes by an order of magnitude.

5. Conclusions

We have investigated syntactic tools for natural language watermarking. In line with Topkara et al. (2006b), we have shown the morphosyntactic approach provides a viable method of natural language watermarking, and that languages such as Turkish, rich in their morphosyntactic repertoire, are especially suitable.

The proposed text-watermarking algorithm works by successively transforming raw sentences into their treebank representation and then into their syntactic tree. A watermarking watchdog detects watermarking tools and marks the sentences by weaving through them in a pseudo-random occurrence order. We conclude that the syntactic data hiding achieves a payload between 0.5 and 1 bit/sentence, and that in security demanding application this rate is below 0.1 bit/sentence.

Future work will focus first on watermarking adaptation to the type or genre of texts. One can also envision developing stylistic watchdog software, which would consider the text as a whole and apply stylistic semaphores and/or corrections. This will take into account the correlations between watermarking transformations.

Acknowledgments

We would like to thank Ersin İhsan Ünkar for his contributions to this research by developing the tree transformer software. We would also like to acknowledge the support of EPRSC Research Project No. EP/F028938/1 on constraints on passivization in natural language watermarking.

Appendix A

A.1. Morphosyntactic tools of Turkish and English

We have gleaned 20 syntactic tools for watermarking from the Turkish language and listed those that are applicable in English (Meral et al., 2006). We think that the present set represents a fairly complete list of tools for Turkish, especially since the few more that we analyzed had vanishing probability of occurrence. In Table A1, the frequency column denotes the frequency *f* of occurrence of the tool per sentence as estimated from sample corpora. Finally, the tools marked with '*' are tools that are not available or require very complex and complicated syntactic manipulations in English. Note that English has more tools than the ones listed here (Topkara et al., 2006b; Murphy and Vogel, 2007a). We have included only those which have a correspondence in Turkish. For the tools applicable to English, relevant examples are also given in the table.

Table A1
The syntactic tools for NLW and their frequency: F: forward, B: backward

Tool	Frequency		Example (1st line: Turkish, 2nd line: gloss, 3rd line: English)
	F	B	
1. Active/passive voice	31.5	0.3	İşçiler **kumu** taşıdı/**Kum** işçiler **tarafından** taşı**n**dı workers sand carried/sand workers by was.carried "Workers carried the sand"/"The sand was carried by workers"
2. Adverb displacement	14.8	11.6	Ali **yarın** Istanbul'a gidecek/ Ali tomorrow to.Istanbul will.go/ "Ali will go to Istanbul tomorrow"/ **Yarın** Ali Istanbul'a gidecek tomorrow Ali to.Istanbul will.go "Tomorrow Ali will go to Istanbul"
3. Conjunct order change	12.4	12.4	Ali ve Ayşe/Ayşe ve Ali Ali and Ayşe/Ayşe and Ali
*4. Verb1 and verb2/verb+(y)Ip verb2	3.3	5.6	Ali eve gel**di** ve yattı/Ali eve gel**ip** yattı Ali to.home came and slept/Ali to.home came.and slept "Ali has come to home and slept"
*5. Verb-NOUN-POSS when/verb-NOUN-POSS-LOC	2.9	1.4	Eve geldiğim **zaman** uyuyordun/ to.home I.came when you.were.sleeping/ Eve geldiğim**de** uyuyordun to.home I.came-when you.were.sleeping "You were sleeping when I came home"
*6.Verb-NOUN1-POSS although1/verb-NOUN2-POSS-DAT although2	0.3	0.07	Baktığım **halde** göremedim/ I.look-NOUN1 although1 I.couldn't.see/ Bakmama **rağmen** göremedim I.look-NOUN2 although2 I.couldn't.see "Although I looked at it, I could not see"
*7. Verb-TENSE-AGR because1/verb-NOUN-POSS because2	2.6	1.0	O eve geç gel**di diye** ona kızdım/ s/he to.home late came because1 to.her/him I.got.angry/ O eve geç gel**diği için** ona kızdım s/he to.home late come-NOUN because2 to.her/him I.got.angry "I got angry because s/he came to home late"
*8. Subject-GEN verb-NOUN-POSS have to/subject-NOM verb-NECESSITY	1.0	1.02	Ali'**nin** çok çalış**ması gerek**/Ali çok çalış**malı** Ali-GEN hard study-NOUN necessary/Ali hard must.study "Ali has to study hard"
*9. Verb-PARTICLE be-NOUN-POSS/verb-NOUN-POSS	0.1	1.5	Geç gel**miş olması** beni kızdırdı/ late came be me made.angry/ Geç gel**mesi** beni kızdırdı late come-Noun me made.angry "Her/his coming late made me angry"

Table A1 (*continued*)

Tool	Frequency		Example (1st line: Turkish, 2nd line: gloss, 3rd line: English)
	F	B	
*10. If … verb-COND/verb-COND	0.7	4.9	**Eğer** erken gelir**se** gideriz/Erken gelir**se** gideriz if early come-COND we.go/early come COND we.go "If s/he comes early, we will go"
11. Sentence 1: subject-predicate-DIR/ Sentence 2: predicate subject-DIR	0.5	0.63	**Damgalama** önemli çalışma alanlarından biri**dir**/ watermarking important study area.of.the one.is/ "Watermarking is one of the important areas of study"/ Önemli çalışma alanlarından biri **damgalamadır** important study area.of.the one watermarking.is "One of the important areas of study is watermarking"
*12. Noun do/Noun be	<0.1	<0.1	Size **yardım ed**ebilir miyim?/Size **yardımcı ol**abilir miyim? to.you make.help can.I/for.you be.helper can.I "Can I help you?"
*13. Noun-without1/Noun-without2	2.5	< 0.1	Toplantıya Ahmet'**siz** başlayabiliriz/ to.meeting Ahmet.without we.can.start/ Toplantıya Ahmet **olmadan** başlayabiliriz to.meeting Ahmet without we.can.start "We can start to the meeting without Ahmet"
*14. Verb1-ArAk/Verb1-A+ verb1-A	4.3	0.22	Ali eve koş**arak** geldi/ Ali to.home by.running came/ Ali eve koş**a koşa** geldi Ali to.home running running came "Ali came home by running"
*15. Emotional Verb-NOM-POSS-A according to1/Emotional Verb-NOM-POSS according to2	<0.1	<0.1	Duyduğuma **göre** Ankara'ya gidiyormuşsun/ what.I.hear according.to to.Ankara you.are.going/ Duyduğum **kadarıyla** Ankara'ya gidiyormuşsun as.far.as I.heard to.Ankara you.are.going "According to what I heard, you are going to Ankara"
16. Subject-GEN verb- NOUN-POSS obvious/Obvious that subject-NOM verb-TENSE-AGR	0.5	0.1	Bu işin o.kadar kolay olmayacağı **belli**/ this work that easy will.not.be obvious/ "That this work will not be so easy is obvious"/ **Belli ki** bu iş o.kadar kolay olmayacak it.is.obvious that this work that easy will.not.be "It is obvious that this work will not be so easy"
*17. More1 verb-AOR-CON-AGR more2/ verb-DIKçA	<0.1	0.44	**Ne kadar** çalışırsak **o kadar** iş çıkıyor/ more we.work more.work we.get done/ Çalıştıkça iş çıkıyor as.we.work.more more.work we.get "As we work more, we get more work done"
*18. Maybe verb-TENSE-AGR/Verb-POSSIBILITY-TENSE-AGR	9.8	1.07	Ali **belki** bu.akşam gelir/Ali bu.şam gel**ebil**ir Ali maybe tonight comes/Ali tonight may.come "Ali may come tonight"
*19. Verb-NEG-IMP-AGR so that/Verb-NEG-NOUN-POSS because	0.7	1.0	Yorulmasın **diye** az iş verdim/ he.will.not.get.tired so.that little work I.assigned/ Yorulmaması **için** az iş verdim he.will.not.get.tired because little work I.assigned "I have assigned little work, so that he will not get tired"
*20. SOV/OSV/VSO/VOS/SVO/OVS	97.2		Ali kitabı okudu/Kitabı Ali okudu/Okudu Ali kitabı/Okudu kitabı Ali/Ali okudu kitabı/Kitabı okudu Ali Ali: Ali, kitabı: the book, okudu: read "Ali read the book"
*21. Unproductive	1.8 (with Tool #20) 24.0 (without Tool #20)		This is the case where none of the watermarking tools is applicable. Geldi came "S/he came"

The tools marked with '*' are tools that are not available or require very complex and complicated syntactic manipulations in English.

Turkish is an Altaic language that possesses rich morphosyntactic structure and differs in quite a number of structural properties from Indo-European languages (Comrie, 1989). Non-agglutinative languages like English make use mostly of distinct lexical items to express linguistic notions such as temporality, actionality, complex sentence formation (i.e. clausal complements, relative clauses), while an agglutinative language like Turkish employs suffixes rather than root words. For instance, to express a temporal modification in a sentence, Turkish makes use of either a lexical item 'zaman' (which corresponds to English 'when'), or suffixes '-DA' or 'IncA' with the same function "when". This creates three varieties such as "Ahmet geldiği zaman", "Ahmet gelince" and "Ahmet geldiğin-de" (when Ahmet has arrived) with similar semantic interpretations.

References

Atallah, M.J., McDonough, C., Nirenburg, S., Raskin, V., 2000. Natural language processing for information assurance and security: an overview and implementations. In: Proceedings of 9th ACM/SIGSAC New Security Paradigms Workshop. pp. 51–65.

Atallah, M., Raskin, V., Crogan, M., Hempelmann, C., Kerschbaum, F., Mohamed, D., Naik, S., 2001. Natural language watermarking: design, analysis and a proof-of-concept implementations. In: Moskowitz, I.S. (Ed.), Proceedings of the 4th Information Hiding Workshop LNCS 2137. Springer, pp. 185–199.

Atallah, M., Raskin, V., Hempelmann, C.F., Karahan, M., Sion, R., Topkara, U., Triezenberg, K.E., 2002. Natural language watermarking and tamperproofing. In: Petitcolas, F.A.P. (Ed.), Proceedings of the 5th Information Hiding Workshop LNCS 2578. Springer, The Netherlands, pp. 196–212.

Bailer, W., Rathner, L., 2001. Linguistic information hiding. <http://www.wbailer.com/wbstego>.

Bennett, K., 2004. Linguistic steganography: survey, analysis, and robustness concerns for hiding information in text. M.S. Thesis, Purdue University. <http://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2004-13.pdf>.

Bergmair, R., 2004. Natural language steganography and an "AI-complete" security primitive. <http://www.ccc.de/congress/2004/fahrplan/files/284-aicomplete-pres.pdf>.

Bergmair, R. 2007. A comprehensive bibliography of linguistic steganography. In: Delp III, E.J., Wong, P.W., (Eds.), Security, Steganography and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging SPIE 6505. pp. 65050W-1–65050W-6.

Bergmair, R., Katzenbeisser, S., 2004. Towards human interactive proofs in the text-domain. In: Proceedings of the 7th Information Security Conference 3225. Springer-Verlag, pp. 257–267.

Bergmair, R., Katzenbeisser, S., 2005. Content-aware steganography: About lazy prisoners and narrow-minded wardens. Tech. Rep. fki-252-05, Technische Universitat München, Institut für Informatik AI/Cognition group.

Bilgin, O., Çetinoğlu, Ö., Oflazer, K., 2004. Building a wordnet for Turkish. Romanian Journal of Information Science and Technology 7 (1–2), 163–172.

Bolskhakov, I.A., 2004. A method of linguistic steganography based on collocationally-verified synonym. In: Fridric, J.J. (Ed.), Information Hiding: 6th International Workshop LNCS 3200. Springer, pp. 180–191.

Cachin, C., 2004. An information-theoretic model for steganography. Information and Computation 192, 41–56.

Chapman, M., Davida, G., 1997. Hiding the hidden: a software system for concealing ciphertext as innocuous text. In: Proceedings of the International Conference on Information and Communication Security. pp. 335–345.

Comrie, B., 1989. The World's Major Languages. Oxford University Press, London.

Cox, I., Miller, M.L., Bloom, J.A., Kaufman, M., 2002. Digital Watermarking. Morgan and Kaufmann, San Francisco.

Eryiğit, G., 2007. ITU validation set for Metu-Sabancı Turkish Treebank. Unpublished paper, ITU.

Eryiğit, G., Oflazer, K., 2006. Statistical dependency parsing of Turkish. In: Proceedings of EACL. pp. 89–96.

Eryiğit, G., Nivre, J., Oflazer, K., 2006. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In: Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages (ICCPOL) LNAI 4285. Springer, Singapore, pp. 498–507.

Göksel, A., Özsoy, A.S., 2003. d*A* as a focus/topic associated clitic in Turkish. Lingua, special edition on focus in Turkish. pp. 1143–1167.

Grothoff, C., Grothoff, H., Alkhutova, L., Stutsman, R., Atallah, M.J., 2005. Translation based steganography. In: Proceedings of Information Hiding Workshop (IH). Springer, pp. 213–233.

Khankhalli, M.S., Hau, K.F., 2002. Watermarking of electronic text documents. Electronic Commerce Research 2, 169–187.

Landes, S., Leacock, C., Tengi, R.I., 1998. Building semantic concordances. In: Fellbaum, C. (Ed.), WordNet: An Electronic Lexical Database, Cambridge, Mass.

Lin, S., Costello, D.J., 2004. Error Control Coding: Fundamentals and Applications. Prentice Hall, Englewood Cliffs, NJ.

Liu, Y., Sun, X., Wu, Y., 2005. A natural language watermarking based on Chinese syntax. In: Advances in Natural Computation 3612. Springer, pp. 958–961.

Meral, H.M., Sankur, B., Özsoy, A.S., 2006. Watermarking tools for Turkish texts. In: Proceedings of National Conference on Signal Processing and Application (SIU).

Meral, H.M., Sevinç, E., Ünkar, E., Sankur, B., Özsoy, A.S., Güngör, T., 2007. Syntactic tools for text watermarking. In: Delp III, E.J., Wong, P.W., (Eds.), Security, Steganography and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging SPIE 6505. pp. 65050X-1–65050X-12.

Mihalcea, R., Csomai, A., 2005. Sense learner: word sense disambiguation for all words in unrestricted text. In: Proceedings of 43rd Annual Meeting of the Association of Computational Linguistics, Ann Arbor. pp. 53–56.

Moulin, P., 2006. On the optimal structure of watermark decoders under desynchronization attacks. In: Proc. Int. Conf. Image Proc. (ICIP), Atlanta.

Murphy, B., 2001. Syntactic information hiding in plain text. M.S. Thesis, CLCS, Trinity College.

Murphy, B., Vogel, C., 2007a. The syntax of concealment: reliable methods for plain text information hiding. In: Delp III, E.J., Wong, P.W. (Eds.), Security, Steganography and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging SPIE 6505. pp. 65050Y-1–65050Y-12.

Murphy, B., Vogel, C., 2007b. Statistically-constrained shallow text marking: techniques, evaluation paradigm and results. In: Delp III, E.J., Wong, P.W. (Eds.), Security, Steganography and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging SPIE. pp. 65050Z-1–65050Z-9.

Oflazer, K., 2003. Dependency parsing with an extended finite state approach. Computational Linguistics 29 (4), 515–544.

Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G., 2003. Building a Turkish treebank. In: Abeille, A. (Ed.), Building and Exploiting Syntactically-Annotated Corpora. Kluwer Academic Publishers.

Raskin, V., Nirenburg, S., 2003. Ontological Semantics. MIT Press, Cambridge.

Simmons, G.J., 1983. The prisoners' problem and the subliminal channel. In: Proceedings of Advances in Cryptology (CRYPTO '83). Springer, USA, pp. 51–67.

Slobin, D.I., Bever, T.G., 1982. Children use canonical sentence schemas: a crosslinguistic study of word order and inflections. Cognition 12, 229–265.

Stutsman, R., Atallah, M.J., Grothoff, C., Grothoff, K., 2006. Lost in just the translation. In: Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC). ACM Press, Dijon, pp. 338–345.

Taskiran, C.M., Topkara, U., Topkara, M., Delp, E.J., 2006. Attacks on linguistic steganography systems using text analysis. In: Delp III, E.J., Wong, P.W. (Eds.), Security, Steganography, and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging 3&4. pp. 313–336.

Topkara, M., Taskiran, C.M., Delp, E.J., 2005. Natural language watermarking. In: Delp, III E.J., Wong, P.W. (Eds.), Security, Steganography, and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging 5681.

Topkara, U., Topkara, M., Atallah, M.J., 2006a. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In: MM&Sec' 06, Proceedings of 8th Workshop on Multimedia and Security. ACM Press, Geneva, pp. 164–174.

Topkara, M., Topkara, U., Atallah, M.J., 2006b. Words are not enough: sentence level natural language watermarking. In: Proceedings of ACM Workshop on Content Protection and Security (in Conjunction with ACM Multimedia) (MCPS). ACM Press, Barbara, California, pp. 37–46.

Topkara, M., Topraka, U., Atallah, M.J., 2007. Information hiding through errors: a confusing approach. In: Delp III, E.J., Wong, P.W. (Eds.), Security, Steganography, and Watermarking of Multimedia Contents IX. Proceedings of SPIE-IS&T Electronic Imaging SPIE 6505. pp. 65050V-1–65050V-12.

Turkish Dictionary. 10th ed., Turkish Language Association, Ankara, 2005.

Wu, J., Stinson, D.R., 2007. Authorship proof for textual document. Cryptology ePrint Archive, Report 2007/042.