Journal of Information Hiding and Multimedia Signal Processing ©2011 ISSN 2073-4212
Ubiquitous International Volume **2**, Number **3**, July **2011**

Capacity-rich Knowledge-poor Linguistic Steganography

Katia Lida Kermanidis

Department of Informatics
Ionian University
7 Tsirigoti Square, 49100 Corfu, Greece
kerman@ionio.gr

Received November 2010; revised March 2011

Abstract. *The present work describes a robust, high-capacity methodology for hiding secret information underneath a Modern Greek cover text by applying shallow syntactic transformations to it. Unlike similar approaches to linguistic steganography, the transformations are extracted automatically by making use of limited external resources, rendering the process easily portable to other free-phrase-order languages. Their shallow nature and restricted locality does not affect grammaticality, i.e. steganographic security, on the one hand, and, on the other, it ensures higher capacity values than the ones reported in the literature by steganographic systems that are based on syntactic transformations.*
Keywords: Linguistic Steganography, Syntactic Transformations, capacity, security, statistical significance testing, supervised learning, Modern Greek

1. **Introduction.** Linguistic steganography [22, 3, 7] studies the insertion of secret information underneath a cover text in order to enable its transmission in an unremarkable way, i.e. without it being detectable by a third party. Linguistic steganography is a relatively new, interdisciplinary field "at the intersection of natural language processing and information security" [15]. The linguistic redundancy, which arises due to the numerous different syntactic and semantic structures a certain meaning can be expressed with, allows for the assignment of a bit sequence to each structure and its insertion within a sentence simply by changing the syntax or choice of words.

The primary goal of steganography is the transmission of the hidden message without arousing any suspicion to its existence by a human or a computer warden. Steganographic communication [8] has succeeded if a third party is unable to detect anything unnatural to the transmitted text that points to it carrying any sort of extra hidden information. If the existence of a secret message is detected, steganographic communication has failed, even though there are ways to ensure that the message will not be extractable.

Previous approaches to linguistic steganography present two significant weaknesses. First, the available bandwidth for transmitting secret information, i.e. the amount of hidden information within a given size of cover text, or *steganographic capacity*, is limited. Approaches that perform syntactic transformations to the cover text take advantage of the multiple applicable syntactic rules to each sentence to hide secret bits within the text. The syntactic transformations may vary from shallow [17] to quite elaborate [15, 20]. Although many rules may be applicable, only one may be applied to the sentence at a given moment. Therefore, capacity is limited to at most one secret bit per cover sentence. Approaches that choose to alter the set of words of the cover text and express its original

meaning using semantically similar words (i.e. synonym substitution) take advantage of the number of synonyms of the words in the original (cover) sentence to hide the secret bits. Synonym sets may be hand-crafted [5, 23, 4], or automatically extracted, e.g. making use of WordNet synsets. The goal is to find the synonym that maximizes the probability as a substitute for the original word across all senses [17, 24, 21, 2]. Several words in a sentence may have one or more synonyms, and synonym replacement may be performed on more than one words simultaneously. Thereby, more than one secret bits may be hidden within a single sentence, leading to higher capacity than syntactic transformations.

A second weakness pertains to the required resources by the linguistic steganography systems. Approaches are knowledge-rich, i.e. they necessitate hand-crafted Grammars of syntactic transformation rules, lexica of paraphrases, parallel corpora, synonym thesauri (e.g. WordNet), and word sense disambiguation tools to perform the desired alterations. This prevents linguistic steganography approaches from being easily applicable to languages that are not adequately equipped with such resources. Furthermore, it affects the robustness of the generated system; sophisticated resources are normally domain-specific and genre-dependent. They perform decently only on specific types of text and limited thematic domains.

The present work describes a methodology for steganographic communication by performing shallow local syntactic transformations on Modern Greek (MG) text. Unlike previous approaches that make use of handcrafted syntactic rules to perform the necessary transformations (e.g. [11]), the transformations described here are learned automatically in two phases. First, statistical significance testing is used to identify phrase bigrams that are 'swappable', i.e. the two phrases forming the bigram may swap places (they appear in both orderings with a statistically significant frequency). In a second phase, erroneous bigrams are learned using supervised classification and filtered out. Finally, the final set of 'swappable' bigrams is used for robust, capacity-rich and secure information embedding.

The approach is knowledge-poor. In contrast to previous knowledge-rich approaches, it requires limited resources, i.e.

- a basic phrase chunker that utilizes a small keyword lexicon containing 450 closed-class words and a lexicon of 300 of the most common word suffixes in MG
- a list of the closed set of twelve relative adverbs and
- a list of the five most common copular verbs in MG.

No use of Grammars, syntactic parsers, paraphrase lexica, parallel corpora, semantic lexica and thesauri of any kind is made. Therefore, the alterations are easily portable to languages that have similar syntactic properties to MG, and domain- and genre-independent.

A significant attribute of the proposed information insertion and extraction process is the shallow nature and the restricted locality of the employed alterations. It enables the simultaneous application of more than one alterations to the same sentence, i.e. the insertion of more than one secret bits within a single sentence, allowing thereby for higher steganographic capacity. The capacity value achieved is significantly higher compared to other approaches that utilize syntactic transformations for information hiding, and comparable to the one reported by synonym substitution approaches.

The rest of the paper is organized as follows. Section 2 describes the two-phase process of extracting the 'swappable' bigrams. Section 3 shows the use of the extracted alterations in information embedding and extraction for steganographic communication. A concrete step-by-step example of the message insertion/extraction process is presented in Section 4. Section 5 describes the experimental setup and the evaluation process for the presented methodology. A discussion regarding security and capacity aspects follows in section 6 and the paper concludes in section 7.

2. **Syntactic Transformations.** MG is a highly inflectional language. While the positioning of the words within a phrase is relatively strict, the rich morphology allows for a large degree of freedom in the ordering of the phrases within a sentence. This phrase ordering freedom enables paraphrase generation merely by changing the phrase order.

Example 2.1. *MG sentence: [Η εταιρία] [χάρισε] [στους υπαλλήλους] [από ένα δώρο] English translation: [The company] [gave] [to the employees] [a gift].*

All possible permutations of the phrases in the Greek example 2.1 above result in grammatically correct sentences that are semantically identical to the original sentence. Subject-verb and verb-object dependencies are determined by the morphology of the participating constituents rather than their position in the sentence. Certain permutations (see example 2.2) may not be common in everyday language, due to their stylistic properties (i.e. they are 'poetic' or 'theatrical'), but they remain perfectly grammatical, and only in rare occasions do they sound weird or suspicious.

Example 2.2. *MG sentence: [από ένα δώρο] [η εταιρία] [στους υπαλλήλους] [χάρισε].*

There are several other free phrase-order languages, like MG: Hungarian [12], Urdu [1], Bengali [9], Arrernte [14], etc. A significant number of these languages is not adequately equipped with linguistic resources [9], thus increasing the importance of the knowledge-poor policy and the relatively easy portability of the proposed methodology.

This idiosyncrasy led to the idea of generating paraphrases merely by swapping the position between two consecutive phrases (chunks). The process focuses on consecutive chunks in order to minimize the probability of an erroneous swap: the longer the distance between the phrases to be swapped, the more likely it is for long distance syntactic dependencies to be affected, and therefore for syntactic errors to appear. Long distance phrase swaps would be safer if the methodology employed linguistic tools for deep processing.

However, not all phrase pairs are 'swappable'. The statistical significance of the cooccurrence of two phrases in both orderings ([PhraseA][PhraseB] and [PhraseB][PhraseA]) is used to determine automatically phrase pairs that may be swapped. Actually, the cooccurrence significance of phrase *type* pairs is estimated. Phrases are abstracted into *phrase types* by performing de-lexicalisation, i.e. striping phrases from information that is not relevant to the task at hand. Noun phrase (NP) types retain the grammatical case of the head noun, verb phrase (VP) types retain the verb voice, the conjunction introducing them (if any) and their copularity, prepositional phrase (PP) types retain the preposition introducing them, while conjunctions (CON) and adverbial phrase (ADP) types retain their type (coordinating or subordinating conjunction, relative adverb or not).

The four most commonly used statistical significance metrics, i.e. the t-test, the log likelihood ratio (LLR), the chi-squared metric (χ^2) and pointwise mutual information (MI), were used to detect phrase type pairs that occur more often together in the ordering [PhraseA][PhraseB] than would be expected by chance, and the same for the ordering [PhraseB][PhraseA]. For each metric, the top N (e.g. N=50, 100, 200 etc.) scores are selected. Phrase type pairs that occur in both orderings among these top N results for a given metric, are considered permissible phrase swaps, as both orderings show significant correlation between the phrase types forming them. These phrase type pairs form the *initial swap set*, which differs depending on the metric.

2.1. **Filtering.** The automatic nature of the proposed procedure for identifying 'swappable' phrase type pairs, as well as its restricted visibility (only two consecutive chunks are taken into account, disregarding the surrounding context) lead inevitably to the selection of phrase bigrams that result in erroneous swaps. A Support Vector Machines

(SVMs) learner is used to classify the swaps of the high-statistical-significance phrase type pairs on input sentences as valid (grammatically correct) or invalid (erroneous), by taking into account the syntactic context surrounding the swap position. A swap position is the position between two 'swappable' phrase types in a sentence.

In more detail, a learning vector is created for every input sentence and each swap position. The features forming the vector encode syntactic information for the phrase right before the swap position, as well as two phrases to the left and two phrases to the right (a total of five phrases). Thereby, context information is taken into account. So even though the visibility of the swaps is limited to only two consecutive chunks, the filtering phase broadens the focus on the context surrounding the swap. Each of the five phrases is represented through a set of six features encoding the phrase category (NP, VP, PP, etc.), the morphological case, the presence of a pronoun or a genitive element in an NP, the copularity and the introductory conjunction of a VP, the preposition introducing a PP, the word introducing a CON or an ADP and their length (number of words they contain). Each swap position is represented by a vector of 30 features, and is manually annotated by language experts with the correct value of the binary target class (valid or invalid swap).

The correlation of each swap pair (vector) with the target class is estimated next. The swap pairs that appear more frequently in negative (invalid) than in positive (valid) vectors are to be removed from the initial swap set, forming thus the *final swap set*.

3. **Message Insertion and Extraction.** The final swap set is used for embedding and extracting secret information. To achieve this, the two parties that wish to communicate in secret need to agree on some crucial decisions beforehand:

- The two parties need to share a secret symmetric key, known only to them.
- The two parties share the same final swap set.
- Each pair in the final swap set needs to have its two sides marked, as all swaps are bidirectional (can be performed in both directions). Each side of the pairs is marked with the value of one bit ('0' or '1'). Unlike approaches [15] that apply uniform marking (e.g. all left hand sides are marked with '0' and right hand sides with '1'), the marking proposed here is performed using the key, and thus more difficult to form a pattern that may be detectable by a third party. A '1001' key, for example, could indicate that the left-hand side of the first and fourth swap pairs is marked with '1', while the right-hand side of the second and third swap pairs is marked with '1'.

The message insertion process is then performed following the steps described next.

1. For every sentence the applicable phrase swaps are selected from the swap set.
   - If the sentence does not allow for any swap, it remains unchanged, and is not used for information embedding.
   - If it does, a selection is possible either in a round-robin fashion, or using the secret symmetric key. For example, a secret key '1001' could indicate that the pairs in the first and fourth swap positions in the cover sentence are chosen to hide secret bits (to be considered for swapping), while the pairs in the second and third swap positions will be disregarded. It should be noted that at least two phrases need to intervene between two selected swap positions, in order to ensure that one swap does not affect the other.
2. If the bit to be hidden in a given swap position matches the marking of the selected applicable swap, the swap is not applied; otherwise it is applied. Assuming, for example, that the secret message is '10', and the pair [PhraseA][PhraseB] is observed

around the first selected swap position, and given that the a-priori marking of this pair is

$$\begin{array}{ll} [PhraseA][PhraseB] & 0 \\ [PhraseB][PhraseA] & 1 \end{array}$$

the marking of the observed pair ('0') does not match the first bit to be hidden ('1'), so the two phrases in the sentence are swapped.

On the other end, the extractor receives the final text. Having at his disposal the same swap set, he is able to identify the swaps that may be applied to each sentence. Sharing the same secret key, he is able to select the same swap positions used in the insertion process. For example, reading [PhraseB][PhraseA] around a selected position, and knowing that this sequence indicates a '1' marking, he correctly decides on '1' to be the first secret bit. Reading [PhraseA][PhraseB] would have meant a '0' marking and he would have decided on '0' to be the first secret bit.

4. **An Example.** This section presents an example for a better understanding of the secret message insertion and extraction process. Let the following three sentences of Figure 1 constitute the initial message. Each sentence is chunked and followed by its word-for-word English translation (the English words in brackets are implied and omitted in the Greek sentences)

(A) VP[είναι] 1 NP[βαρετός]
[(he) is] [boring]
(B) VP[πήγαμε] 1 PP[στον αγώνα] 2 ADP[χτες] CON[και] ADP[μάλιστα] 3 VP[κάτσαμε] 4 ADP[μπροστά]
[(we) went][to the game] [yesterday] [and] [also] [sat] [in the front]
(C) VP[κοιμήθηκε]
[(he) fell asleep]

FIGURE 1. Initial text.

Table 1 shows the marked final swap set that is used in the current example.

TABLE 1. Example of a marked final swap set.

	Final swap set	**Marking**
(1)	[VPpassive/copular][NPnominative]	'0'
(2)	[VPactive/non-copular][PP$\sigma\epsilon$]	'1'
(3)	[PP$\sigma\epsilon$][ADPnon-relative]	'1'
(4)	[ADPnon-relative][VPpassive/non-copular]	'0'

The applicable swap pairs for the given text are: for sentence A pair (1), for sentence B pairs (2), (3) and (4), and for sentence C no pair. Actually, pair (4) may be applied to sentence B twice. The respective swap positions are shown in the sentences with integers in boxes. However, due to the at-least-two-phrases-intervening requirement in sentence B, only swap positions 1 and 3, 1 and 4, 2 and 3, and 2 and 4 may be considered for bit insertion. So, even though there are four candidate swap positions, no more than two bits may be embedded in sentence B, so as to minimize the probability of grammatical errors.

Suppose that the message to be hidden is the bit sequence '101', and that the shared key is '1001'. The two communicating parties have agreed on a '0' bit in the key to indicate

disregarding the position, while a '1' bit indicates using the position for information insertion. So, for embedding the secret message, position 1 in sentence A is considered. The marking of the pair that is applicable at this position is being checked, according to Table 1. The applicable swap pair is pair (1), which appears in the sentence in the ordering [VPpassive/copular][NPnominative] (i.e. marking '1'). The first secret bit is '1', the two bits match, so the swap does not take place. Similarly, according to the key, position 1 in sentence B will also be considered for bit insertion. The applicable swap pair is pair (2), which appears in the sentence in the ordering [VPactive/non-copular][PP$\sigma\epsilon$] (i.e. marking '0'). The second secret bit is '0', the two bits match, so the swap does not take place. Positions 2 and 3 in sentence B are disregarded. The next position to be considered is position 4 in sentence B. The applicable swap pair is (4), which appears in the ordering [VPpassive/non-copular][ADPnon-relative] (i.e. marking '0'). The last secret bit is '1', the two bits do not match so the swap takes place. The final text to be sent is shown in Figure 2.

(A1) VP[είναι] 1 NP[βαρετός]

(B1) VP[πήγαμε] 1 PP[στον αγώνα] 2 ADP[χτες] CON[και] ADP[μάλιστα] 3 ADP[μπροστά] 4 VP[κάτσαμε]

(C1) VP[κοιμήθηκε]

FIGURE 2. Transmitted text.

The receiver gets this text. Applying the reverse process, and having at his disposal the same marked swap set and the same secret key, he knows position 1 in sentence A1 is hiding a secret bit. The applicable swap is swap (1). He reads the sequence [VPpassive/copular][NPnominative] (i.e. marking '1'), and chooses '1' to be the first secret bit. The second position hiding a secret bit is position 1 in sentence B1. The applicable swap is swap (2). Reading [VPactive/non-copular][PP$\sigma\epsilon$] (i.e. marking '0'), he chooses '0' to be the second secret bit, and so on.

The example shows clearly the potential of the proposed algorithm to embed more than one secret bits within a single sentence (i.e. sentence B), even though the use of syntactic transformations for linguistic steganography has so far been known to allow for at most one bit insertion per sentence.

5. **Experimental Setup.** The ILSP/ELEFTHEROTYPIA corpus [10] used in the experiments consists of 5244 sentences, is manually annotated with morphological information, and balanced in genre. Phrase structure information is obtained automatically by the chunker described in [19]. During chunking, non-overlapping NPs, VPs, PPs, ADPs and CONs are detected via multi-pass parsing. The chunker exploits minimal linguistic resources: a keyword lexicon containing 450 closed-class words (articles, prepositions etc.) and a lexicon of 300 of the most common word suffixes in MG. The chunker identifies basic phrase constructions during the first passes (e.g. adjective-nouns, article-nouns), and combines smaller phrases into longer ones in later passes (e.g. coordination, inclusion of genitive modifiers, compound phrases).

De-lexicalisation leads to 156 phrase types. Applying statistical significance testing to type pairs, using all four metrics, led to the pair sets shown in Figure 3 for various top N values. The average number of swaps that are permitted per sentence for each phrase swap set in Figure 3 is shown in Figure 4.

Two native speakers judged 882 randomly selected sentences and their produced paraphrases, according to grammaticality. The judgment process was blind, i.e. the experts

were not familiar with the original sentence. They were simply shown a set of sentences and asked to decide whether they were grammatical, or they required a phrase swap to become grammatical. Inter-expert agreement exceeded 94% using the kappa statistic. The percentage of paraphrases (sentences) that required one or more manual phrase swaps from the human judges in order to become grammatical is shown in Figure 5 for every swap set. It should be noted that an average of 6% of the reported errors were on the original sentences, an indication of an upper bound for the performance of the specific task.

It is possible for a swap to result in a grammatically correct, but semantically different sentence compared to the initial one. While in steganography the cover text meaning itself is not important (unlike watermarking), discourse cohesion is, i.e. the sequence of transmitted sentences needs to make sense, so as not to arouse any suspicion that something is wrong. A second 'non-blind' series of experiments was conducted, where the experts where shown the original sentence and its paraphrases, and were asked to judge whether the latter present a difference in meaning, compared to the original sentences. An average of 1.4% of the paraphrases presented a difference in meaning: a very low rate, that does not affect the given task. It should be noted that a difference in meaning does not necessarily imply a discourse cohesion inconsistency, making this phenomenon even less problematic for the task. Furthermore, the style or naturalness of all the original sentences remained intact when they were paraphrased, due to the locality of the syntactic alterations, thus not affecting imperceptibility in the least. But even if the style were affected, it would not necessarily mean that a third party's suspicion would be aroused to the existence of an anomaly in the text in front of them. It should be kept in mind that the corpus is comprised of texts of varying genre and style. In other words, the proposed process is 'trained' to cope robustly with any text style.

Each of the four metrics has idiosyncratic properties that affect the resulting swap sets. MI, due to its relation to Information Theory, returns a more diverse set of swap pairs, i.e. a set that contains 'exclusive' ('surprising', not very frequent) phrase types, that are not included (or included scarcely) in the sets returned by the other metrics. Such phrase types are relative ADPs, genitive NPs, unusual PPs (e.g. PPs introduced by the preposition $\omega\varsigma$ - until). This set leads to a small average number of swaps per sentence, and a high error rate. T-test returns an extensive set of swap pairs that consist of more frequent (usual) phrase types and results in the smallest error rate. The use of the T-test for testing the significance of word co occurrence has been contested, due to its assumption that the data is normally distributed [18]. The good results in the current approach are attributed to the fact that the statistical significance of phrase types'- rather than words'- co occurrence is tested, and the distribution of phrase types is not as heavily tailed as the Zipfian distribution (the distribution of words), due to the 'de-lexicalisation' process. The χ^2 metric relies heavily on the sample size. Our corpus, though balanced and varying in style is not large enough for the metric to produce as small an error rate as the T-test. The LLR, like the MI, is again biased towards rare, extreme events, i.e. unusual phrase types, that, when included in a swap set, tend to result in erroneous transformations.

A significant part of the errors is attributed to the automatic nature and the low level of the chunking process: Erroneous phrase splitting, incorrect attachment of punctuation marks, and the inability to identify certain relative, adverbial and idiomatic expressions, and to resolve PP attachment ambiguities and subordination dependencies lead to swapping errors that would have been avoided by applying more sophisticated parsing.

To evaluate the filtering impact on the error rate, the positions of possible phrase swaps in the input sentences were identified according to the T-test swap set. The swap set for the top 100 results was selected, as its error rate turned out to be significantly lower than

that of the top 200 and top 300 swap sets, and the average number of paraphrases it returned higher than the top 50 set. The experts manually annotated (assigned the class label value: valid or invalid paraphrase) the instances (vectors), corresponding to the 882 original sentences (5104 instances) already used for the evaluation of the statistical significance testing process. The parameters of the SVMs classification algorithm were set to a first degree polynomial kernel function, and the sequential minimal optimization algorithm for training. 10-fold cross validation was chosen as the evaluation method. SVMs were selected because they are known to cope well with high data sparseness and multiple attribute problems, both valid in the present dataset. Classification performance reached 82% precision and 86.2% recall. Swap pairs that occur more frequently with the negative (invalid paraphrase) than with the positive (valid paraphrase) class are removed, seven in number.

The reduced swap set was evaluated against a held-out test set (100 new corpus sentences, not included in the training data of the filtering phase) and reached an error rate of 17.2%. Against the 882-sentence training set, the error rate dropped to 13.8%.

Comparing these results with previous supervised learning approaches to paraphrase identification is not straightforward. In [13] a learning example represents a pair of sentences through a set of features that denote lexico-semantic similarity between the two sentences, like shared word sequences, word similarity etc. The goal is to decide whether one of the two sentences is a paraphrase of the other. In the current approach, the presented dataset consists of learning examples, each one representing a single sentence. Each learning example corresponds to a swap position, and the example's features encode morphosyntactic information regarding the context surrounding the position. The goal here is to decide whether the two phrases surrounding the given position may or may not be swapped. Lexico-semantic features like the ones mentioned previously are out of the scope of the present methodology and not abiding by its low resource policy. The authors in [13] (even though no direct comparison would be meaningful as their methodology and dataset are very different) report 100% precision and 66.49% recall for knowledge-rich paraphrase identification with an SVMs classifier.

When comparing these results to previous approaches, one needs to take into account all aspects of each work in question, including the required resources. In the most recent linguistic steganography system for English that employs syntactic transformations [6], a paraphrase dictionary (based on a parallel corpus and statistical machine translation techniques), a Combinatory Categorial Grammar parser, as well as the Google N-gram Data are required for generating and confirming the grammaticality of paraphrases. The authors in [6] report an error rate (1 minus accuracy) between 32% and 59%, depending on the value of N (the number of words in the phrase to be replaced by its paraphrase) and the phrase's context size to be taken into account. Another approach that performs syntactic transformations on Turkish for hiding secret bits by applying twenty hand-crafted rules and making use of rather sophisticated resources (Turkish Treebank, WordNet, Dictionary) is described in [16]. The authors report an average error rate of 12.7% on the applied rules.

6. **Security and Capacity.** In the presented approach, security is addressed in a number of ways:

- The number of permissible swaps. The average number of permissible syntactic alterations per sentence is greater compared to similar previous approaches [15] due to their shallow nature, and the linguistic properties of MG that allow for relatively free phrase swapping. Unlike approaches that allow for the application of at most one rule to a sentence, the proposed methodology allows for the application of more

than one phrase swaps at various positions to a sentence. The greater-than-average number of legitimate alterations makes it difficult for an eavesdropper to decide upon the correct one.

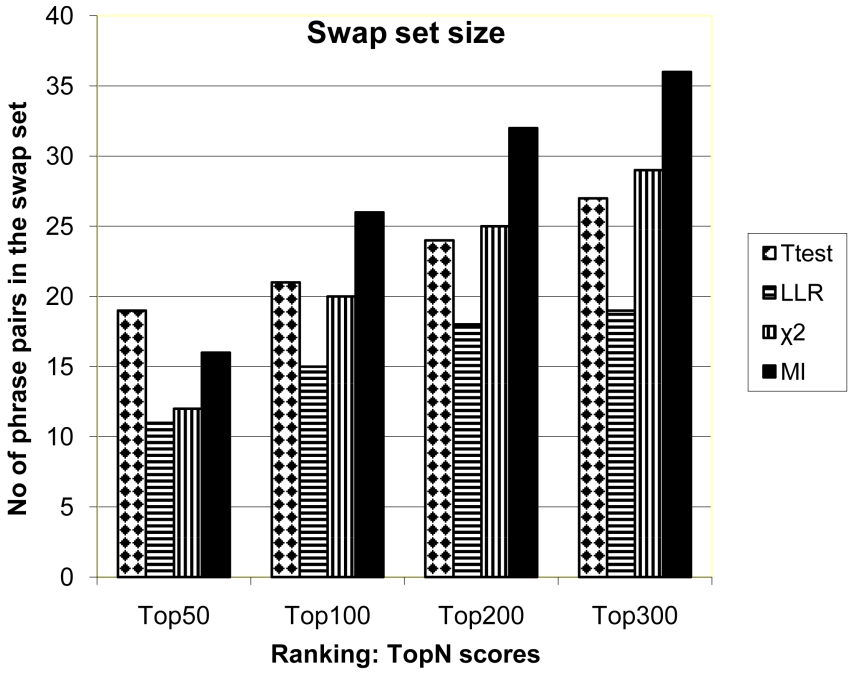

FIGURE 3. The size of the swap sets for various statistical significance metrics.

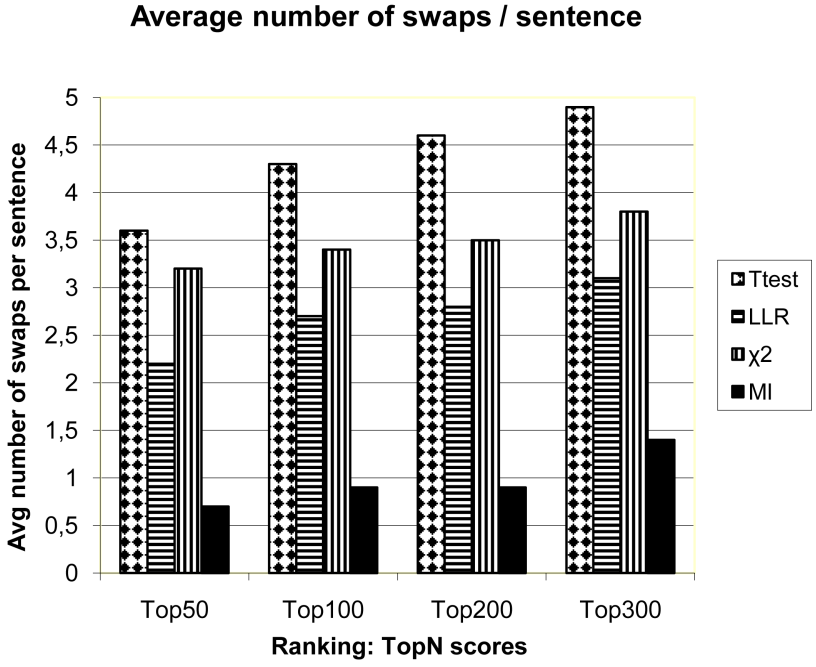

FIGURE 4. The applicability of the various swap sets.

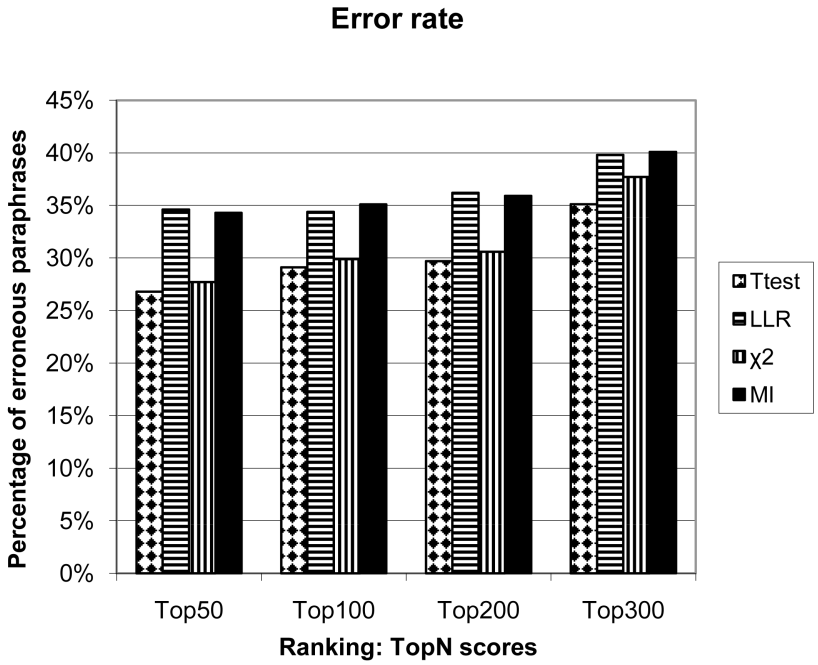

FIGURE 5. Error rate for all swap sets.

- Unlike similar previous approaches that perform static marking on the left and right hand side of their syntactic alterations [16], the swap set marking presented here is based on a cryptographic key. Thereby only the two communicating parties can 'interpret' the presence of a specific phrase bigram as indicating a bit value '0' or '1'. A third party, not familiar with the key, even if he got a hold of the swap set, would have to try out all possible markings, in all possible swap positions of the transmitted text, a process of significant complexity.
- The random manner of choosing the swaps to be performed, as well as the not-to-be-performed swaps. The choice is again based on a cryptographic key, forcing the eavesdropper (if he is familiar with the swap set) to test all possible alternatives (perform all possible alterations to a transmitted sentence). Furthermore, this 'randomness' does not allow for any kind of pattern in the insertion process (the set of performed syntactic alterations) to be detectable by an outsider.
- The grammaticality of the swap set. The swap set evaluation in the previous section proved to be comparable to state-of-the-art approaches, ensuring the generation of correct paraphrases. Thereby, looking at a transmitted message, it is very difficult for an eavesdropper to suspect whether it contains a hidden message or not.

Security has to be defined in relation to the profile of the attacker. An initial question is whether the attacker is familiar with the resources, i.e. the utilized swap set. The degree of freedom might make the informed attacker's job difficult, but it is not impossible to decode the hidden information, even if the possibilities are numerous. A way to improve security further and address this shortcoming is the use of a separate secret bit string (key), that has comparable length to that of the hidden message, to encode the message, before embedding it, using a bitwise logical operation of equivalence (e.g. OR) [5]. After extracting it, the recipient decodes the message by performing the reverse logical operation. The transformed (operated upon) secret message is now very difficult to extract by a third party that is not aware of the keys employed.

One significant aspect of the presented approach, as already mentioned, is the increased bandwidth it offers, compared to other linguistic steganography approaches that utilize syntactic alterations. Assuming an average word size of 6 bytes/word, and given that the corpus consists of 166,000 words, the corpus size equals roughly 1 Million bytes. If only one bit per paraphrase-able sentence were allowed (with the initial swap set), 4762 (5244 minus 482) secret bits would be able to be embedded in the corpus (the total number of corpus sentences minus the number of sentences that cannot be paraphrased). In other words, 1 bit would be able to be embedded every 1667 bits of cover text. Capacity drops slightly after filtering, i.e. with the reduced swap set, to 1 embeddable bit every 1733 cover text bits.

Using the current implementation, however, where, depending on the secret key, more than one sentence positions may hide a secret bit, capacity increases significantly. Ideally, capacity would reach a maximum value if all swap positions were allowed to hide secret bits. Given that the total number of swap positions in the text is around 32,600, this means 1 embeddable bit every 246 cover text bits. Allowing, however, every swap position to hide a secret bit threatens security. Too many alterations on the same sentence are highly likely to affect grammaticality, semantics, and discourse cohesion. The trade-off between capacity and security has been claimed and verified in previous work also [6]. The stricter the syntactic schemata employed, the more accurate (high security) and the less applicable they are (low capacity), and vice versa. So, choosing random swap positions and taking care not to violate the at-least-two-phrases-intervening criterion, still results in a higher capacity than all previous approaches performing syntactic alterations that report a capacity of 0.5-1 bits/sentence [15]. The only line of approaches that lead to higher capacity are the ones employing synonym substitution, due to the possibility of multiple word substitutions within a sentence. For example the author in [5] reports a capacity factor of 1 hidden bit for every 250 cover text bits. However, as mentioned earlier, these approaches are very resource-demanding.

7. **Conclusions.** The presented methodology takes advantage of the phrase-ordering freedom of MG in order to extract permissible shallow syntactic alterations that will enable the embedding of secret information underneath a cover text for steganographic communication. Unlike previous approaches that employ syntactic transformations for linguistic steganography, the proposed methodology relies on minimal external linguistic resources, and is therefore robust, domain independent, and easily applicable to other free-phrase-order languages. Despite the shallow nature of the extracted alterations, their grammaticality evaluation proved to be comparable to that of resource-demanding approaches, ensuring steganographic security. Furthermore, the presented approach is a first implementation of a steganographic system that is characterized by high capacity, even though it is based on syntactic transformations.

REFERENCES

[1] W. Ali, and S. Hussain, A hybrid approach to urdu verb phrase chunking, *Proc. of International Conference on Computational Linguistics* (COLING), Beijing, China, 2010.

[2] M. Atallah, C. McDonough, V. Raskin, and S. Nirenburg, Natural language processing for information assurance and security: an overview and implementations, *Workshop on New Security Paradigms*, Ballycotton, County Cork, Ireland, pp. 51-65, 2000.

[3] K. Bennett, *Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text*, CERIAS Technical Report 2004-13, 2004.

[4] R. Bergmair, *Towards Linguistic Steganography: A Systematic Investigation of Approaches, Systems and Issues*, BSc Thesis, University of Derby, 2004.

[5] I. A. Bolshakov, J. J. Fridrich (eds.), A method of linguistic steganography based on collocationally-verified synonymy, *Proc. of the 6th International Workshop on Information Hiding*, Springer Verlag, pp. 180-191, 2004.

[6] C. Y. Chang, and S. Clark, Linguistic Steganography Using Automatically Generated Paraphrases, *Proc. of NAACL-HLT Conference*, Los Angeles, 2010.

[7] I. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2002.

[8] A. Desoky, NORMALS: Normal linguistic steganography methodology, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 3, pp. 145-171, 2010.

[9] A. Ekbal, and S. Bandyopadhyay, Voted NER system using appropriate unlabeled data, *Proc. of Named Entities Workshop: Shared Task on Transliteration*, Suntec, Singapore, pp. 202-210, 2009.

[10] N. Hatzigeorgiu, et al., Design and implementation of the online ILSP greek corpus, *Proc. of the 2nd International Conference on Language Resources and Evaluation*, Athens, pp. 1737-1742, 2000. (http://www.elda.fr/catalogue/en/text/W0022.html)

[11] K. Kermanidis, and E. Magkos, Empirical paraphrasing of modern greek text in two phases: an application to steganography, *Proc. of the 10th International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing), Springer Verlag, pp. 535-546, ISBN: 978-3-642-00381-3, ISSN: 0302-9743, 2009.

[12] A. Kornai, I. Kenesei (eds.), Frequency in morphology, *Approaches to Hungarian*, vol. 4, pp. 246-268, 1992.

[13] Z. Kozareva, and A. Montoyo, Paraphrase Identification on the Basis of Supervised Machine Learning Techniques, *Proc. of International Conference on Natural Language Processing* (FinTAL), Springer Verlag, pp. 524-533, 2006.

[14] S. Levinson, and D. Wilkins, *Grammars of Space*, Cambridge University Press, 2006. (http://dx.doi.org/10.1017/CBO9780511486753.003)

[15] H. M. Meral, B. Sankur, A. S. Ozsoy, T. Gungor, and E. Sevinc, Natural Language Watermarking via Morphosyntactic Alterations, *Computer Speech and Language*, Elsevier, vol. 23, pp. 107-125, 2009.

[16] H. M. Meral, E. Sevinc, E. Unkar, B. Sankur, A. S. Ozsoy, and T. Gungor, Syntactic Tools for Text Watermarking, *Proc of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents* IX, 2007.

[17] B. Murphy, and C. Vogel, Statistically-constrained Shallow Text Marking: Techniques, Evaluation Paradigm and Results, *Proc. of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents* IX, 2007.

[18] V. Seretan, *Collocation Extraction Based on Syntactic Parsing*, Ph. D. Thesis, University of Geneva, 2008.

[19] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, A practical chunker for unrestricted text, *Proc. of Conference on Natural Language Processing*, Patras, pp. 139-150, 2000.

[20] M. Topkara, G. Riccardi, D. Hakkani-Tuer, and M. Atallah, Natural language watermarking: challenges in building a practical system, *Proc. of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents* VIII, pp. 106-117, 2006.

[21] U. Topkara, M. Topkara, and M. Atallah, The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym Substitutions, *Proc. of the 8th Workshop on Multimedia and Security*, Geneva, pp. 164-174, 2006b.

[22] M. Topkara, C. M. Taskiran, and E. Delp, Natural language watermarking, *Proc. of SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, USA, 2005.

[23] K. Winstein, *Lexical Steganography Through Adaptive Modulation of the Word Choice Hash*, 1998. (http://www.imsa.edu/ keithw/tlex/)

[24] B. Wyseur, K. Wouters, and B. Preneel, Lexical natural language steganography systems with human interaction, *Proc. of the 6th European Conference on Information Warfare and Security*, 2007.