Linguistic steganography with knowledge-poor paraphrase generation

Katia Lida Kermanidis
Department of Informatics, Ionian University, Greece

Abstract

Paraphrasing is very useful for many applications that normally involve deep linguistic alterations of a sentence, like summarization, textual entailment and question answering, and usually require sophisticated external resources, pre-processing, and semantic thesauri. This article presents a methodology for generating shallow linguistic alterations of Modern Greek sentences making use of only a low-resource chunker and taking advantage of the freedom in phrase ordering of the language. A statistical significance testing process is applied for extracting 'swappable' phrase bigrams. A supervised filtering phase follows, which helps to remove erroneous paraphrasing schemata, taking into account the context in which the alteration is to take place. Unlike most previous approaches to paraphrasing, the proposed process is knowledge-poor (and thus quite easily portable to other languages with a syntactic structure similar to Modern Greek), robust (applicable to any type of text), domain independent, and leads to the generation of a significant number of paraphrases, by allowing the application of more than one syntactic alterations per sentence. The significance of the automatically generated paraphrases is shown by their application in hiding secret information underneath a cover text in a steganographic communication channel. For this purpose, they need not be sophisticated linguistic alterations, but grammatically correct and significant in number, to ensure security. A steganographic security and capacity analysis of the presented implementation, as well as an explanatory description of the trade-off between them, is included to show the usefulness and the practical value of the methodology.

Correspondence:
Katia Kermanidis,
Department of Informatics,
Ionian University, 7 Tsirigoti Sq., 49100 Corfu, Greece.
E-mail:
kerman@ionio.gr

1 Introduction

Numerous Natural Language Processing applications require a component that, given a sentence, reformulates it (changes its syntactic structure and/or vocabulary), but preserves its meaning. In other words, it generates paraphrases of the original sentence. Example 1 shows paraphrase generation for a Modern Greek (MG) sentence using a different set of words, while in example 2 paraphrasing is achieved through syntactic restructuring.

(1) Θα πάω μια βόλτα μέχρι να βραδιάσει
(I'll go for a walk until it gets dark)
Θα περπατήσω μέχρι να πέσει ο ήλιος
(I'll walk until the sun sets)
(2) Πέρασα χτες από τον οδοντίατρο για έναν έλεγχο
(I went by the dentist's yesterday for a check-up)
Χτες πέρασα για έναν έλεγχο από τον οδοντίατρο

Paraphrasing is an important indicator of linguistic competence in first- and second-language learning (Milicevic, 2008), it provides significant

Literary and Linguistic Computing, Vol. 26, No. 4, 2011.
doi:10.1093/llc/fqr026 Advance Access published on 30 May 2011

resources for authoring support (Okamoto, 2003; Barreiro, 2009), it is essential for text summarization (Brockett and Dolan, 2005), for text realization in natural language generation (Harbusch *et al.*, 2007), as well as for question answering (Duclaye *et al.*, 2003), machine translation (Nakov, 2008), textual entailment, and semantic inference (Bar-Haim *et al.*, 2009).

The aforementioned applications usually make use of automatically extracted paraphrases that are intricate linguistic alterations of an original sentence, requiring sophisticated tools and resources that are not available for many languages. The extracted paraphrases are limited in number and the necessary tools are usually domain-specific and not particularly robust, i.e. they have limited coverage and can only be applied to text that is restricted in structure and domain.

The present work describes the process of automatically generating shallow MG paraphrases, and then, of using them to enable steganographic communication between two parties that wish to exchange secret information. The bits of the secret message are embedded within a cover text in an unremarkable way that does not arouse an eavesdropper's suspicion to the existence of hidden information. This application defines the two primary goals of the presented approach. The first goal is to produce as many correct paraphrases as possible for an original sentence. Steganographic security depends to a large extent on the number and the grammaticality of the paraphrases of each cover text sentence. Unlike previous work, where each paraphrasing rule may be applied once to a sentence (Meral *et al.*, 2007), the transformations proposed in the present approach may be applied multiple times (i.e. in multiple positions) to a sentence. Thereby, the number of extracted paraphrases increases. Security in steganography is always in a trade-off relation to steganographic capacity, i.e. the amount of secret information that may be embedded into the cover medium. An analysis of steganographic security and capacity of the proposed approach is included at the end of the article.

The second goal is to employ as limited linguistic resources as possible. This will first allow the portability of the proposed methodology to other languages that share certain syntactic properties with MG (e.g. Hungarian), and that are not necessarily equipped with sophisticated linguistic resources. Also, it will ensure robustness and domain independence (the proposed alterations are applicable to any MG text, regardless of its domain, genre or stylistics). The paraphrases need not be sophisticated syntactic or semantic alterations that presuppose the availability of high-level linguistic resources. The methodology requires a morphological case tagger (in the experiments presented here the corpus is manually tagged with morphological information, including the case), a phrase chunker that utilizes limited resources, a list of the most frequent copular MG verbs, and a list of the most common relative MG adverbs. Even the most elementary transformation is adequate for hiding secret information, as has been shown in previous work (Kermanidis and Magkos, 2009), unlike previous approaches that rely on more sophisticated syntactic alterations (Meral *et al.*, 2009; Chang and Clark, 2010).

To present the paraphrasing process briefly, the proposed methodology is a combination of two phases: a statistical process for generating an initial set of paraphrases and a filter that helps to remove erroneous (noisy) paraphrases. In other words, a statistical significance testing process is responsible for generating pairs of consecutive phrases (chunks) that are *swappable*, i.e. they are permitted to swap positions. *Swappability* is determined based on the phrases' co-occurrence statistics. Due to the low resource policy employed, the resulting set of chunk pairs is noisy and contains pairs that often lead to syntactically incorrect output when swapped. For this reason, a filtering phase follows that helps remove such error-prone pairs. Filtering is performed using supervised learning [a support vector machines (SVM) classifier], which identifies erroneous swaps by taking into account the context they appear in. To the author's knowledge, this is the first time a (partly) unsupervised approach to generating MG paraphrases is proposed.

The next section presents previous approaches to paraphrasing. Section 3 describes the MG corpus used in the presented experiments, its preprocessing, the paraphrase generation and filtering

process, as well as its evaluation. Section 4 presents a literature review on linguistic steganography, and describes the proposed message embedding and extraction process, including a discussion on security and capacity aspects. The article concludes in Section 5.

2 Paraphrasing: Related Research

Automatic paraphrasing entails paraphrase *generation*, i.e. the realization of a new sentence to express the meaning of another, and/or paraphrase *identification* (or acquisition), i.e. given two sentences, the classification of one as an (in)valid paraphrase of the other. Approaches to paraphrase generation vary from the application of handcrafted syntactic rules (Meral *et al.*, 2007) to dictionary-based synonym replacement techniques (Bolshakov, 2004). Paraphrase acquisition focuses on empirical methods to detect synonymous (or near-synonymous) sentences, resulting in resources that allow for paraphrase generation, such as the empirical construction of finite state automata that represent word lattices with synonymous arcs (Pang *et al.*, 2003), the use of statistical machine translation techniques (like word alignment on monolingual parallel corpora to build parallel interchangeable word/phrase pairs) (Quirk *et al.*, 2004), and using shared named entity markers in sentences of different articles to identify parallel utterances and then extract patterns of dependency relations from them (Shinyama *et al.*, 2002). Subsumption of two syntactic dependency graphs is checked, revealing equivalent graphs, in the work by Rus *et al.* (2008). Word lattices depicting patterns from clusters of structurally similar sentences are detected with multiple sequence alignment in the work described by Barzilay and Lee (2003). Viewing paraphrase identification as a tagging task, supervised machine learning (Kozareva and Montoyo, 2006) has also been proposed. In the series of PASCAL Recognizing Textual Entailment Challenges [the last of which is described in Bentivogli *et al.* (2009)] teams compete in implementing systems that perform semantic inference, i.e. given one sentence, they attempt to decide upon the meaning of another.

As already mentioned, most previous approaches aim at producing paraphrases that are relatively deep alterations of the original sentence, taking full advantage of their available resource potential, without worrying about the number of generated paraphrases. The focus of the present approach shifts to coverage, to the ability to apply the proposed shallow syntactic transformations to any MG text, to the fairly easy applicability of the methodology to languages that are not equipped with sophisticated resources and tools, and to the number of extracted paraphrases. As already mentioned the only required tools are a case tagger, a low-resource chunker, a list of the most frequent MG copular verbs, and a list of the most common MG relative adverbs. Resources like parallel corpora, semantic thesauri, syntactic parsers, etc. are not necessary.

3 Syntactic Transformations

This section describes in detail the process of generating shallow MG paraphrases. Certain linguistic properties of the language in question need to be presented first, as well as the corpus used and the shallow pre-processing tools that were applied to it.

3.1 Modern Greek and languages with similar syntax

A core aspect of the current work is the language in question. MG is highly inflectional. While the position 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 is a significant property of the language, and enables paraphrase generation merely by permutating the phrase order.

[Η *τράπεζα*] [*δανείζει*] [*σε πελάτες*] [*χρήματα*].
[The bank] [lends] [to clients] [money]

In the English translation, only very limited re-ordering of the phrases is permitted. The two objects ([to clients] and [money]) may swap places, but any other permutation leads to either

an ungrammatical or an unnatural sentence. The English language syntax follows to a large extent the subject–verb–object (SVO) order. However, all the permutations of the phrases in the Greek example above result in grammatically correct sentences, which are semantically identical to the original sentence. MG does not follow the SVO order. 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 next example) may not be common in everyday language, due to their stylistic properties (i.e. they are 'poetic' or 'theatrical'), but they remain perfectly grammatical.

[Σε πελάτες] [χρήματα] [δανείζει] [η τράπεζα].

Several languages are similar to MG regarding this phrase ordering freedom, like Hungarian (Kornai, 1992), Urdu (Ali and Hussain, 2010), Bengali (Ekbal and Bandyopadhyay, 2009), Arrernte (Levinson and Wilkins, 2006). A significant number of these languages is not adequately equipped with linguistic resources (Ekbal and Bandyopadhyay, 2009), thus increasing the importance of the knowledge-poor policy and the relatively easy portability of the proposed methodology.

3.2 Corpus and pre-processing

The ILSP/ELEFTHEROTYPIA corpus (Hatzigeorgiu *et al.*, 2000) used in the experiments consists of 5,244 sentences, is manually annotated with morphological information, and balanced in genre. Phrase structure information is obtained automatically by the chunker described in Stamatatos *et al.* (2000). During chunking, noun (NP), verb (VP), prepositional (PP), adverbial phrases (ADP), and conjunctions (CON) 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). Certain features of the extracted chunks need to be noted:

1 Chunks are non-overlapping, i.e. no chunk contains whole or part of other chunks.
2 Nominal modifiers in the genitive case are included in the same phrase with the main noun they modify.
3 Base nouns joined by a coordinating CON are grouped into one phrase.

3.3 Hypothesis testing

After splitting the corpus sentences into non-overlapping phrases, *phrase types* are identified by discarding content words from the phrases, and keeping only lexicosyntactic information that has an impact on the position of the phrases in the sentence. For *NP types* this information is the grammatical case of their headword. The headword is the noun in the nominative or accusative case. If there is no noun, it is the adjective, numeral or pronoun in this order, else the first phrase element. *VP types* are distinguished by the verb voice, by the CON introducing them (if any), and by their *copularity* (whether they are copular or not). Copular verbs denote a trait, a property, like *είμαι* (to be) and *γίνομαι* (to become). *PP types* are distinguished by the preposition introducing them, *CON types* by their CON-type (coordinating or subordinating), and *ADP types* are distinguished by their type (relative adverb or not). Copular verbs and relative adverbs are limited in number and easily identified. A total of 156 phrase types were formed.

Next, the statistical significance of the co-occurrence of two phrase types is measured using hypothesis testing: the *t*-test, the log likelihood ratio (LLR), the chi-squared metric (χ^2), and pointwise mutual information (MI) metrics have been experimented with. A detailed description of these metrics and their comparative evaluation can be found in many sources (Manning and Schuetze, 1999; Seretan, 2008). Phrase type pairs that occur in both orderings ([TYPE1][TYPE2] and [TYPE2][TYPE1]) among the top results with the highest rank (i.e. among the highest statistical significance values) are selected. These are considered permissible phrase swaps, as both orderings show significant correlation between the phrases forming them.

Table 1 Swap set size for every metric

	Size of phrase swap set			
	Top 50	**Top 100**	**Top 200**	**Top 300**
t-test	19	21	24	27
LLR	11	15	18	19
χ^2	12	20	25	29
MI	16	26	32	36

Table 1 shows the size of the set of selected phrase pairs for every statistical significance metric (only one of the two orderings is counted for every pair), and various values for the number of the *N*-best (top) results. For all *N*-values, statistical significance proved to be well above the threshold for α level 0.05. Table 2 shows the swap pairs for $N = 100$.

The process focuses on consecutive chunks in order to minimize the possibility of an erroneous swap: the longer the distance between the phrases to be swapped, the more probable 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.

Swap pairs that are a priori known to be incapable of producing legitimate swaps are removed from the sets ([Type][#], [Type][CON-coordinating], [Type][CON-subordinating], and their symmetrical pairs). '#' denotes end of sentence (the low-resources chunker sometimes groups punctuation marks into phrases).

For the remaining pairs, in case one is detected in an input sentence, the two phrases are swapped, and, thereby, a paraphrase is produced. The average number of swaps that are permitted per sentence for each phrase swap set in Table 1 is shown in Fig. 1. If more than one phrase swaps are applicable at different positions in an input sentence, all possible combinations of swaps are performed, and all respective paraphrases are produced, forming the *initial pool of paraphrases*. Figure 2 shows the frequency distribution of the initial pool size for the top 100 *t*-test metric, i.e. 482 sentences have 0 paraphrases, 1,468 have 1–2 paraphrases, etc.

Then, 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.

It is possible for a swap to result in a grammatically correct, but semantically different sentence compared to the initial one. This is not a problem in the present approach, as the cover text meaning itself is not important. Interexpert agreement exceeded 94% using the κ 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 Fig. 3 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 of the performance of the specific task.

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 (Seretan, 2008). 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.

A set of nine manually created rules that govern bigram and trigram chunk swaps has been applied to the same data set (Kermanidis and

Table 2 The swap pair sets for every metric and top 100 statistical significance values

	LLR	*t*-test	χ^2	MI
1	[NPacc][NPacc]	[NPacc][NPacc]	[NPacc][NPacc]	[NPacc][NPacc]
2	[NPacc][VPact]	[NPacc][NPnom]	[NPacc][VPact]	[NPacc][NPnom]
3	[NPacc][VPpass]	[NPacc][VPact]	[NPacc][VPpass]	[NPacc][VPact]
4	[NPacc][PP$\gamma\iota\alpha$]	[NPacc][PP$\alpha\pi\acute{o}$]	[NPacc][PP$\gamma\iota\alpha$]	[NPacc][VPpass]
5	[NPacc][PP$\mu\varepsilon$]	[NPacc][PP$\mu\varepsilon$]	[NPacc][PP$\mu\varepsilon$]	[NPacc][VPcop]
6	[NPacc][ADP]	[NPacc][PP$\sigma\varepsilon$]	[NPacc][PP$\sigma\varepsilon$]	[NPacc][PP$\sigma\varepsilon$]
7	[NPacc][$\nu\alpha$VPact]	[NPacc][ADP]	[NPacc][ADP]	[NPacc][ADP]
8	[NPnom][NPnom]	[NPacc][$\nu\alpha$VPact]	[NPnom][NPnom]	[NPacc][$\acute{o}\tau\iota$VPact]
9	[NPnom][VPact]	[NPnom][NPnom]	[NPnom][VPact]	[NPgen][VPcop]
10	[NPnom][VPpass]	[NPnom][VPact]	[NPnom][VPpass]	[NPacc][NPnom]
11	[NPnom][PP$\alpha\pi\acute{o}$]	[NPnom][VPpass]	[NPnom][VPcop]	[NPnom][PP$\sigma\varepsilon$]
12	[NPnom][PP$\sigma\varepsilon$]	[NPnom][VPcop]	[NPnom][ADP]	[NPnom][ADPrel]
13	[NPnom][ADP]	[NPnom][PP$\alpha\pi\acute{o}$]	[VPact][VPact]	[NPnom][$\nu\alpha$VPact]
14	[VPact][PP$\sigma\varepsilon$]	[NPnom][PP$\sigma\varepsilon$]	[VPact][VPpass]	[VPact][VPact]
15	[VPact][ADP]	[NPnom][ADP]	[VPact][PP$\gamma\iota\alpha$]	[VPact][VPpass]
16		[VPact][PP$\sigma\varepsilon$]	[VPact][PP$\sigma\varepsilon$]	[VPact][PP$\omega\varsigma$]
17		[VPact][ADP]	[VPact][ADP]	[VPact][PP$\kappa\alpha\tau\acute{a}$]
18		[VPpass][PP$\sigma\varepsilon$]	[VPpass][VPpass]	[VPact][ADPrel]
19		[VPpass][ADP]	[VPpass][ADP]	[VPpass][VPpass]
20		[PP$\gamma\iota\alpha$][PP$\sigma\varepsilon$]	[$\nu\alpha$VPact][$\nu\alpha$VPact]	[PP$\alpha\pi\acute{o}$][PP$\sigma\varepsilon$]
21		[PP$\sigma\varepsilon$][ADP]		[PP$\alpha\pi\acute{o}$][$\nu\alpha$VPpass]
22				[PP$\gamma\iota\alpha$][PP$\gamma\iota\alpha$]
23				[PP$\gamma\iota\alpha$][ADP]
24				[VP$\sigma\varepsilon$][ADPrel]
25				[$\nu\alpha$VPpass][ADP]
26				[$\acute{o}\tau\iota$VPact][ADP]

The words appearing are $\gamma\iota\alpha$ (for), $\nu\alpha$ (to), $\mu\varepsilon$ (with), $\sigma\varepsilon$ (in), $\acute{o}\tau\iota$ (that), $\alpha\pi\acute{o}$ (from), $\kappa\alpha\tau\acute{a}$ (against), $\omega\varsigma$ (until). *act, pass, nom, acc, cop, rel, c* and *s* stand for active voice, passive voice, nominative case, accusative case, copular verb, relative adverb, coordinating conjunction, and subordinating conjunction, respectively.

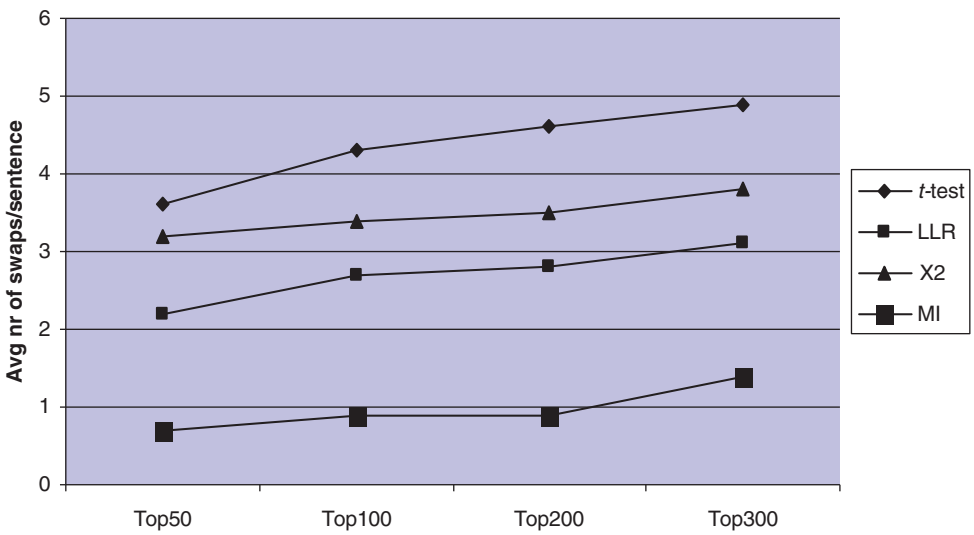

Fig. 1 Applicability of the phrase swap pairs

Magkos, 2009), and the authors reported an error rate of 8.2%. The rules are chunk swaps that represent shallow versions of subject–verb swapping, verb–secondary clause of result swapping, coordinated phrases swapping, swapping of an ADP with its preceding phrase, and verb–PP swapping. They are unification rules, i.e. morphological constraints determine their applicability. Attempting to extract 'swappable' phrase pairs statistically leads to a higher error rate, compared to the one achieved by the limited set of strict rules. However, at the same time, the number of generated paraphrases in the former case is significantly greater (there is a 15% increase in the number of generated paraphrases compared to the hand-crafted rules approach).

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 solve PP attachment ambiguities and subordination dependencies lead to swapping errors that would have been avoided by applying more sophisticated parsing.

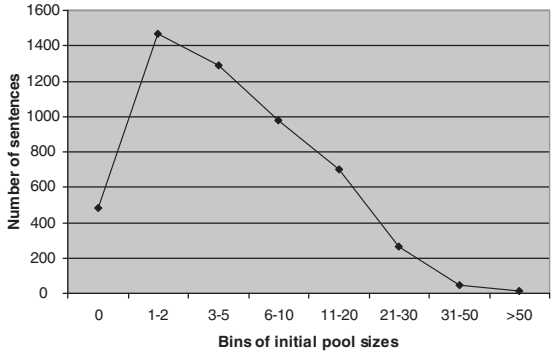

Fig. 2 Frequency distribution of the initial pool size for the Top 100 *t*-test metric

3.4 Filtering

To reduce the error rate, the extracted swap sets undergo a filtering process, where erroneous swap pairs are learned [pairs are classified as (in)valid] and withdrawn from the final pair sets. The positions of possible phrase swaps in the input sentences are identified according to the *t*-test swap set from the previous section. 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 fifty set.

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

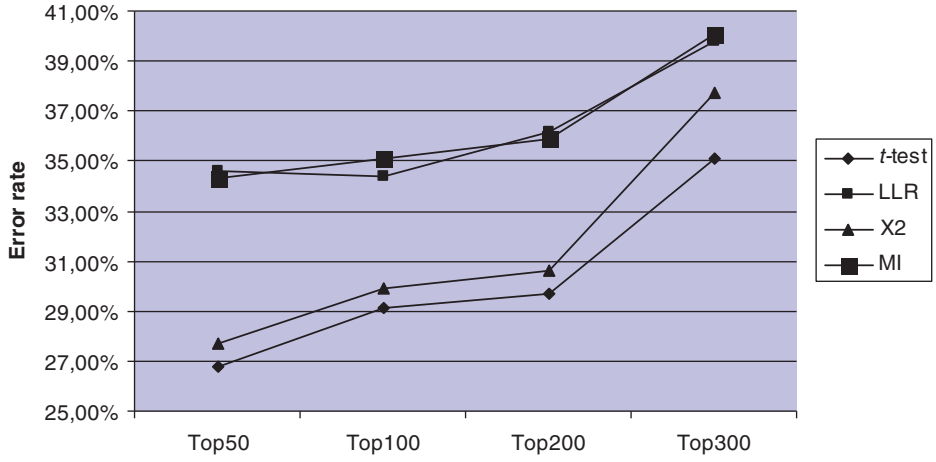

Fig. 3 Error rate for every swap set

Table 3 The features of the learning vector

	NP	VP	PP	CON/ADP
1	NP	VP	PP	CON/ADP
2	Case	–	–	–
3	Morph	–	–	–
4	Pronoun in NP	First word	Preposition	First word
5	Genitive element in NP	Copular verb	–	–
6	–	–	–	Num

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, shown in Table 3.

The first feature for every phrase is the phrase category (NP, VP, PP, etc.). *Case* is a character denoting the case of an NP headword. *Case* is decisive for several cases of swapping: a qualitative evaluation revealed that, while [NPnom][VPpass] is usually 'swappable', [NPacc][VPpass] usually is not. *Morph* is a three-letter code denoting whether an NP contains a definite or indefinite article and its case. The (in)definitiveness of an NP affects the accuracy of swaps containing NPs. The fourth feature varies according to the phrase category. For NPs it is the type of pronoun appearing in them, if any. It is the first word (non-verb) introducing a VP, the preposition introducing a PP, and the CON or the adverb in a CON and an ADP, respectively. The fifth feature is binary and encodes whether there is an element in the genitive case in an NP, or a copular verb in a VP. The presence of a genitive element is often decisive for the accuracy of swaps containing NPs. Finally, *num* is the number of tokens (words) within a CON or an ADP. This feature is very important, as a one-word ADP phrase usually denotes manner and a swap with its preceding phrase is permissible, while a multi-word ADP phrase very often does not.

In previous supervised learning approaches to paraphrase identification (Kozareva and Montoyo, 2006), 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 data set consists of learning examples, each one representing a single sentence. Features encode morphosyntactic information regarding the context surrounding a specific position of the sentence. There is a different learning example for each position. The goal 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 the low resource policy.

Native speakers have manually annotated the instances (vectors), corresponding to the 882 original sentences (5,104 instances) already used for the evaluation of the statistical significance testing process. A SVMs classifier (with a first degree polynomial kernel function, and the sequential minimal optimization algorithm for training) was trained to classify instances using 10-fold cross-validation. SVMs were selected because they are known to cope well with high data sparseness and multiple attribute problems, both valid in the present data set. Classification performance reached 82% precision and 86.2% recall. Kozareva and Montoyo (2006) (even though no direct comparison would be meaningful as their methodology and data set are very different) report 100% precision and 66.49% recall for knowledge-rich paraphrase identification with an SVMs classifier.

The correlation of each swap pair with the target class was estimated next. The swap pairs that appear more frequently in negative (invalid paraphrase) than in positive instances were removed from the final swap set (seven in number). Table 4 shows the percentage of appearance of each of the *t*-test swap pairs with the positive and negative class. The removed pairs are indicated in bold. Figure 4 depicts the full process of paraphrasing described so far.

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%. Given the 'knowledge poverty', the results are satisfactory when compared to those of approaches that utilize sophisticated resources (Meral *et al.*, 2007; Chang and Clark, 2010), which report an average error rate of 12.7% on the applied rules, and an error rate varying from 0 to 32.3% (depending on the value of *n* and the context size) when paraphrasing *n*-gram phrases, respectively. Table 5 shows the results for all approaches that make use of the same dataset, and therefore allow for direct comparison.

Table 4 Appearance of the co-occurrence of the pair with the positive and with the negative class

Pair	Positive class (%)	Negative class (%)
[NPacc][NPacc]	68	32
[NPacc][NPnom]	88	12
[NPacc][VPact]	80	20
[NPacc][PP$\alpha\pi\acute{o}$]	86	14
[NPacc][PP$\mu\varepsilon$]	84	16
[NPacc][PP$\sigma\varepsilon$]	**47**	**53**
[NPacc][ADP]	89	11
[NPacc][$\nu\alpha$VPact]	**38**	**62**
[NPnom][NPnom]	72	28
[NPnom][VPact]	89	11
[NPnom][VPpass]	91	9
[NPnom][VPcop]	94	6
[NPnom][PP$\alpha\pi\acute{o}$]	81	19
[NPnom][PP$\sigma\varepsilon$]	**46**	**54**
[NPnom][ADP]	83	17
[VPact][PP$\sigma\varepsilon$]	**44**	**56**
[VPact][ADP]	**42**	**58**
[VPpass][PP$\sigma\varepsilon$]	58	42
[VPpass][ADP]	**47**	**53**
[PP$\gamma\iota\alpha$][PP$\sigma\varepsilon$]	91	9
[PP$\sigma\varepsilon$][ADP]	**52**	**48**

The removed pairs are indicated in bold.

The filtering process helps to broaden the focus on the context surrounding the swap position. The context is very important when trying to decide upon a permissible swap. In the following example the fact that the phrase before the swap position (*) is an active VP (VP_1) links the following NP strongly with it, not allowing the swap. On several occasions, however, a swap between an accusative NP and a VP is permissible.

```
...[αυτό] [που] VP1[έχει] NP[σημασία] *
   VP2[είναι...]...
...[that] [what] VP1[has] NP[importance]
   VP2[is...]...
(...what's important is...)
```

It is interesting to study the pairs that tend to lead to correct vs. incorrect swaps. [PP$\gamma\iota\alpha$] (PP introduced by the preposition $\gamma\iota\alpha$—for) is usually attached to the sentence verb, and so may almost always be swapped with the phrase preceding it, while [PP$\sigma\varepsilon$] (a PP introduced by the preposition $\sigma\varepsilon$—to) is more problematic. ADPs may usually be swapped with preceding NPs, but things get more complicated when they are preceded by a VP. Certain secondary clauses (e.g. final clauses

Table 5 Collective, comparative results

Approach	Error rate (%)
Manually crafted rules (Kermanidis and Magkos, 2009)	8.2
Initial swap set	29.7
Reduced swap set—training set	13.8
Reduced swap set—held out test set	17.2

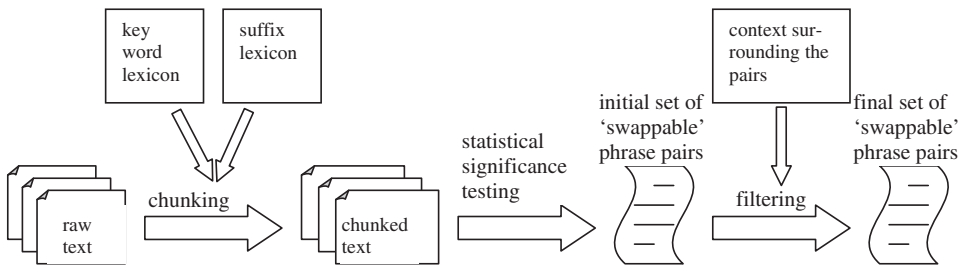

Fig. 4 The process of extracting the final reduced set of 'swappable' pairs of consecutive phrases

introduced by the particle *να*, or relative clauses) may often be swapped with their preceding main verb phrase, but not with a preceding NP. The set of paraphrases generated by the reduced swap set form the *final pool of paraphrases*.

4 Application to Steganography

The significance of the produced paraphrases is shown by their use in linguistic steganography, i.e. the art of embedding hidden information in unremarkable cover text in a way that does not arouse an eavesdropper's suspicion to the existence of hidden content underneath the surface message (Atallah *et al.*, 2000; Cox *et al.*, 2002; Provos and Honeyman, 2003). Linguistic steganography is a relatively new, interdisciplinary field 'at the intersection of natural language processing and information security' (Meral *et al.*, 2009).

4.1 Linguistic steganography: related research

Previous approaches to the use of natural text as the cover medium for hiding secret information may be categorized into two groups. The first group, usually referred to as *text steganography*, focuses on alterations made to the physical formatting of the cover text (Bennett, 2004), like the insertion of spaces, deliberate orthographical errors, font resizing, line shifting, creating random or statistical character or word sequences (Wayner, 2002; Raskin and Nirenburg, 2003), or taking advantage of the special features that certain characters contain in some languages (Gutub and Fattani, 2007). The main drawback of these approaches is that they are easily detectable by human 'wardens' and they are not robust against attempts to reformat the text. The second group, referred to as *linguistic steganography*, embeds the message to be hidden within the linguistic structure of the cover text, by performing a set of alterations including synonym substitution, syntactic transformations and semantic transformations (Topkara *et al.*, 2005).

Synonym substitution replaces words in a sentence with their synonyms (Bolshakov, 2004; Topkara *et al.*, 2006a,b). Regarding synonym selection, some approaches employ WordNet synsets (Murphy and Vogel, 2007; Wouters *et al.*, 2007) and attempt to find the synonym that maximizes the probability as a substitute for the original word across all senses requiring a word sense disambiguation tool, while others make use of dictionary-based synonym lists and apply mixed radix number encoding and Huffman trees in order to perform the selection (Winstein, 1998; Bergmair, 2004). Wouters *et al.* (2007) propose human intervention for synonym selection by the sender, so as to ensure imperceptibility without having to make use of a word sense disambiguation tool. Semantic transformations (Atallah *et al.*, 2002) identify noun phrases that refer to the same entity (coreferences). Upon coreference detection, a repeated noun phrase is replaced by a referring expression (e.g. anaphora). Again, the needed tools are sophisticated resources for deep semantic analysis.

Finally, approaches performing syntactic transformations (Meral *et al.*, 2007; Murphy and Vogel, 2007; Kermanidis and Magkos, 2009; Chang and Clark, 2010), including the one presented in this article, modify the syntactic structure of a sentence (while preserving its original meaning). Typical syntactic alterations include active–passive transformations, extraposition, clefting and they normally require language-specific parsers. The plural number of syntactic structures a sentence may appear in allows for the embedding of information within the syntactic structure itself. Another syntactic approach, which does not rely on already existing text, is the use of context-free grammars (CFGs) for generating text (mimicry) that may hide secret bits wherever the grammar presents a syntactic ambiguity. The generated text is syntactically correct, as it has been produced by applying the grammar rules, but its semantics, discourse structure and style can be very problematic, making it easily detectable by a human third party. This is even more the case, if the grammar is restricted, i.e. it covers only a small set of syntactic phenomena.

Another set of previous approaches has focused on *natural language watermarking*. While in steganography the cover text itself is of no actual importance, in watermarking the cover text is important and needs to remain intact. Text watermarking is

more intricate than text or linguistic steganography, as it is challenging to ensure robustness (indestructibility of the cover text). Shallow (Murphy and Vogel, 2007) and more elaborate (Topkara *et al.*, 2006a; Meral *et al.*, 2007) syntactic transformations have been proposed for hiding secret information within the cover text in a way that does not allow the extraction of the hidden message (mark) without destroying the cover text. Synonym substitution approaches replace a word with a semantically similar one, in a way that is not safely reversible, i.e. the initial word cannot be safely identified. Topkara *et al.* (2006b) perform synonym substitution using WordNet to replace the original word with a more ambiguous synonym, so that the process is irreversible by a third party without damaging the meaning of the original text. Atallah *et al.* (2000) propose the use of the quadratic residue theory to detect the text places where meaning-preserving linguistic modifications should be performed in order to embed the secret message.

Other approaches make use of machine translation tools that generate possible translations of a source sentence and choose one of them to embed the secret message (Grothoff *et al.*, 2005; Stutsman *et al.*, 2006). At the orthographical level, Topkara *et al.* (2007) insert typographical errors in the cover text, and take advantage of the correction ambiguity to insert the secret bits. The tools needed for this are negligible, but the typos are instantly visible to a human or a machine warden, presenting thus a security problem.

Recently, approaches have been focusing also on the stegoanalyst's end, proposing attacking scenarios on linguistic steganographic systems. Taskiran *et al.* (2006) train two language models, one using cover (unmodified) and one using steganographic text, and use the output to train an SVM classifier to identify text as (un)modified. Chen *et al.* (2008a) take advantage of the difference in the statistical characteristics of correlations between certain words in normal and stego text in order to distinguish between the two text types. Zhi-li *et al.* (2008) detect differences between the distributions of words in the two text types. Chen *et al.* (2008b) also use an SVM classifier to classify between modified and unmodified text by taking into account an information entropy variable and statistical variance. Meng *et al.* (2008) compare the perplexities of normal and steganographic text that derive from a stego text language model, and find them very different.

4.2 Embedding the hidden message

Once the final pool of paraphrases for every sentence in the input (cover) text is formed, a secret message, i.e. a sequence of bits, is to be embedded within the cover text in three stages. As the phrase swaps are bi-directional, one direction is chosen (by convention, or using a secret symmetric cryptographic key) by the two communicating parties, and each side of the swap is marked with a 1-bit value (e.g. '0' marks the left and '1' marks the right-hand side of a specific swap). A cryptographic key is a secret bit string shared beforehand between the two parties. So, for example, if the first bit of the key equals to '1', this could mean that the left-hand side of the first phrase pair in the set is marked with '1'.

In the next stage, for every text 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 cryptographic key.

In the last stage, a secret bit is embedded as follows: if the bit to be hidden matches the marking of the selected applicable swap, the swap is not applied and the sentence remains unchanged, otherwise it is applied and the sentence is paraphrased.

4.3 Extracting the hidden message

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 used in the insertion process. For example, reading [NPnom][VPact], and knowing that this sequence indicates a '0' marking, he decides on '0' to be the first secret bit. Reading [VPact][NPnom] would have meant a '1' marking and he would have decided on '1' to be the first

secret bit. The message insertion/extraction algorithm is shown step-by-step in Figs 5 and 6.

4.4 An example of information hiding

Let the following three sentences constitute the initial message.

```
(A) VP[γίνομαι] NP[ενήλικας]
    [I become] [an adult]
(B) VP[φάγαμε]PP[στο μεξικάνικο εστιατόριο] ADP[χτες]
    [we ate][at the Mexican place] [yesterday]
(C) VP[κουράστηκε]
    [he's tired]
```

Let's assume the following final swap set:

```
[VPpass][NP]     (1)
[VPact][PPσε]    (2)
[PPσε][ADP]      (3)
```

The applicable swap pairs for the given text are: For sentence A, pair (1); for sentence B pairs (2) and (3); and for sentence C no pair. So the final pool of paraphrases is:

```
(A1)                    [ενήλικας] [γίνομαι]
[B1-after swap (3)]     [φάγαμε][χτες]
                        [στο μεξικάνικο εστιατόριο]
[B2-after swap (2)]     [στο μεξικάνικο εστιατόριο]
                        [φάγαμε][χτες]
```

Suppose that the hidden message is the bit sequence '10'. For embedding the secret message, the marking of the pair that is applicable to the first sentence is being checked. Assuming that a [VPpass][ADP] sequence corresponds to swap marking '0', the two bits don't match (as the first bit to be hidden is '1'). Therefore, swap (1) is applied, the paraphrase is activated and the first sentence to be sent is A1.

Given the secret key, or in a round-robin fashion, the message sender decides next on one of the two applicable swap pairs for sentence B. Suppose that pair (2) is chosen and that, according to this pair, a [VPact][PP$\sigma\varepsilon$] sequence corresponds to swap marking '0'. The next message bit to be embedded is '0'. The two bits match, so swap (2) is not applied, and the second sentence is sent as it is. So the sent message is A1 B C.

The receiver gets this text. He looks at the swap set to decide which swap may possibly have been applied to sentence A1. It is only swap (1). In A1 he detects the sequence adverb–verb. According to pair (1), this indicates swap marking '1'. So, he chooses '1', which is the first hidden bit. The applicable swaps for the second sentence are (2) and (3). Using his secret key, he chooses pair (2). According to this pair, a sequence of a verb and a PP introduced by the preposition $\sigma\varepsilon$ corresponds to the marking '0'. He chooses '0', the second hidden bit.

4.5 Security, capacity, and robustness

There are three important aspects to steganography: security, capacity, and robustness. The security level determines the (in)ability of an eavesdropper to 'wiretap' the hidden message. Capacity refers to the amount of information that can be hidden in the cover medium, and robustness defines the amount of modification the cover medium can withstand without destroying the hidden information, or even the cover text in the case of watermarking.

In the presented approach, security is addressed in a number of ways:

(1) The number of permissible swaps. The average number of permissible syntactic alterations per sentence is greater compared to similar previous approaches (Meral *et al.*, 2009) 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, increasing the number of generated paraphrases. The greater-than-average number of legitimate alterations makes it difficult for an eavesdropper to decide upon the correct one.

(2) Unlike similar previous approaches that perform static marking on the left- and right-hand side of their syntactic alterations (Meral *et al.*, 2007), 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

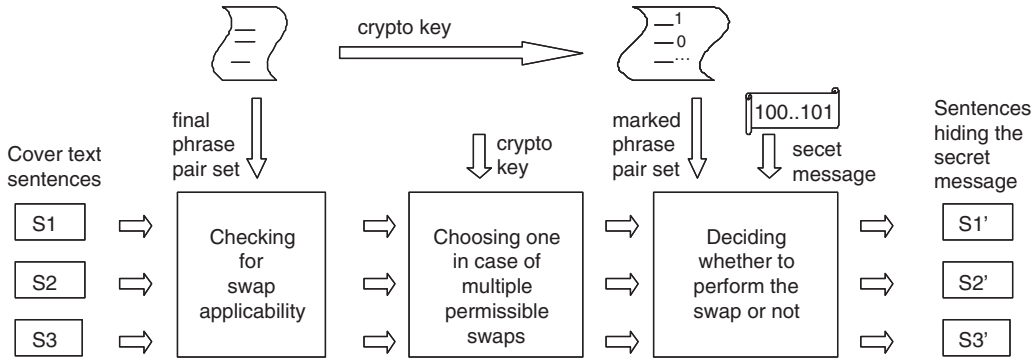

Fig. 5 The steps of the message insertion process. For each step the required input is indicated by the vertical arrows pointing to it

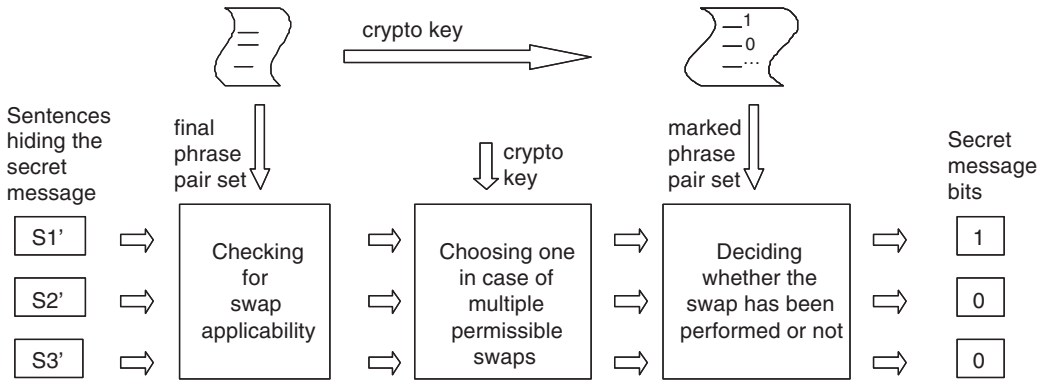

Fig. 6 The steps of the message extraction process. For each step the required input is indicated by the vertical arrows pointing to it

possible swap positions of the transmitted text, a process of significant complexity.

(3) The random manner of choosing the swap to be performed in case of multiple permissible 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.

(4) The choice between applying a swap or not. Even given multiple swap choices for a sentence, the message sender may decide to not perform any swap at all, and send the sentence as it is. This adds another degree of freedom to the steganographic process and another layer of confusion to the eavesdropper.

(5) 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.

It has to be noted at this point that 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) (Bolshakov, 2004). 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.

Another distinction is between a passive and an active attacker. A passive attacker will wiretap transmitted messages and try to detect those that contain hidden messages. If he suspects the existence of hidden information he may try to destroy it by performing changes to the transmitted text. The defense against passive attacks is the imperceptibility of the secret message, so that the attacker cannot suspect its existence. An active attacker will randomly attack the communication channel and change the transmitted text at arbitrary time points in order to impede any secret communication between the two parties. The insertion of control bits along with the secret message bits is the defense mechanism against such attacks. The control block is an error correction code that detects erroneously transmitted bits (bits transmitted by an illegitimate party), and it may correct up to a specific number of error bits.

To obtain a feeling of steganographic capacity, 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. Steganographic capacity (the available bandwidth) may be evaluated as follows: using the current implementation (with the initial swap set), which allows for the embedding of one bit per paraphrase-able sentence, 4,762 (5244−482) secret bits may be embedded in the corpus (the total number of corpus sentences—the number of sentences that cannot be paraphrased). In other words, one bit may be embedded every 1,667 bits of cover text. Capacity drops slightly after filtering, i.e. with the reduced swap set, to one embeddable bit every 1,733 cover text bits. This is still a significant improvement over the manually crafted rules approach (Kermanidis and Magkos, 2009), where the number of embeddable bits is 4,142 (one secret bit every 2,000 cover text bits). These calculations do not include control bits for defense against attacks, which would lower embedding density by an order of magnitude (Meral *et al.*, 2009).

It has been claimed (Chang and Clark, 2010) and verified here once more that there is a trade-off between security and capacity: the stricter the syntactic schemata employed, the more accurate (high security) and the less applicable they are (low capacity), and vice versa. This bandwidth may be further increased by exploiting the possibility of embedding more than 1 bits per sentence. This can be achieved by modifying the information embedding process to allow bit insertion at every possible swap position in a sentence. Ideally, a sentence that has X swap positions, will allow for hiding X secret bits. This is an innovative potential offered by the proposed methodology, as all syntactic transformation approaches in the literature lead to a capacity of 0.5–1 bits/sentence (Meral *et al.*, 2009), and constitutes an interesting future research direction. Approaches adopting synonym substitution achieve higher capacity values, due to the possibility of multiple word substitutions within a sentence. For example Bolshakov (2004) reports a capacity factor of one hidden bit for every 250 cover text bits. However, as mentioned earlier, they are very resource demanding. The presented approach is robust and adequately secure to be used for hiding without too much fuss (sophisticated pre-processing) small secret messages underneath any MG text, when defense against destruction attacks is not a primary concern, but only transmitting a non-detectable message.

5 Conclusion

This work presented a novel methodology for the automatic generation of MG shallow paraphrases.

The approach utilizes limited external linguistic resources, making the process easily portable to other languages that have a similar syntactic freedom to MG. The methodology is robust, i.e. it can be applied to any MG text, and domain independent. The statistical significance of the co-occurrence of phrase bigrams is measured, and phrase pairs that are highly correlated in both orderings form the initial swap set. To improve paraphrasing performance, a supervised SVMs filter, taking into account the morphosyntactic context in which the alteration is to take place, reduces the number of generated swapping schemata by removing schemata that are strongly correlated to erroneous syntactic transformations.

The correctness and the significant number of the extracted paraphrases render them useful for steganographic communication. The syntactic transformations make it possible to hide secret bits underneath a cover text, and a third party is unable to detect the applied transformations and decide upon the correct ones (and therefore unable to extract the hidden message). Apart from the innovative way of generating correct and significant in number paraphrases, the ability to perform multiple alterations (swaps) within a single sentence offers novel potential for enhancing steganographic security and capacity.

It would be interesting to explore the use of other filters (other than supervised learning) in order to remove erroneous candidate phrase swaps from the sets derived using the statistical significance metrics. Another challenging perspective would be to increase the depth of the transformations, e.g. to enlarge the window size between the phrases to be swapped, instead of focusing only on two consecutive chunks. This would increase paraphrasing accuracy (coordinating schemata would be dealt with, more distant dependencies would be addressed, etc.) and make it more complicated for a malicious party to detect the underlying syntactic transformations. In this case, however, correct swapping would require carefully set morphosyntactic restrictions on the context surrounding the swap. Regarding steganography, an interesting future research objective is taking advantage of the multiple swap positions in most sentences to hide more than one bits within a single sentence, thus increasing steganographic capacity.

The application areas of the presented paraphrasing generation methodology are not limited to linguistic steganography. An initial version of the methodology, that makes use of a small number of handcrafted rules for producing permissible swaps, has been proposed to be applied to the area of language learning (Kermanidis, 2009) for partial syntax checking or as a support tool for teachers and exam designers. It may also be employed for authoring support that provides the author with suggestions about how to form his/her text, for language generation, summarization (create a summary with an altered syntax), and question answering.

References

Ali, W. and Hussain, S. (2010). A Hybrid Approach to Urdu Verb Phrase Chunking. *Proceedings of the 8th Workshop on Asian Language Resources (ALR-8), COLING-2010*. Beijing, China, pp. 137–143.

Atallah, M., McDonough, C., Raskin, V., and Nirenburg, S. (2000). Natural language processing for information assurance and security: an overview and implementations. *Proceedings of the Workshop on New Security Paradigms*. New York: ACM, pp. 51–65.

Atallah, M., Raskin, V., Hempelmann, C. F. *et al.* (2002). Natural language watermarking and tamperproofing. In Petitcolas, F. A. P. (ed.), *Proceedings of the Fifth Information Hiding Workshop. Lecture Notes in Computer Science (2578)*. The Netherlands: Springer, pp. 196–212.

Bar-Haim, R., Berant, J., and Dagan, I. (2009). A compact forest for scalable inference over entailment and paraphrase rules. *Proceedings of the Conference of Empirical Methods in Natural Language Processing*. Singapore, pp. 1056–1065.

Barreiro, A. (2009). *Make it Simple with Paraphrases: Automated Paraphrasing for Authoring Aids and Machine Translation*. PhD thesis, New York University.

Barzilay, R. and Lee, L. (2003). Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. *Proceedings of the Conference on Human Language Technology*. (HLT-NAACL), Edmonton. pp. 16–23.

Bentivogli, L., Dagan, I., Dang, H., Giampiccolo, D., and Magnini, B. (2009). *The Fifth PASCAL Recognizing*

Textual Entailment Challenge, Proceedings of the Text Analysis Conference. Gaithersburg, Maryland, USA.

Bennett, K. (2004). *Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text*. CERIAS Technical Report 2004–13.

Bergmair, R. (2004). *Towards Linguistic Steganography: A Systematic Investigation of Approaches, Systems and Issues*. B.Sc. Thesis. University of Derby.

Bolshakov, I. A. (2004). A method of linguistic steganography based on collocationally-verified synonymy. In Fridrich, J. J. (ed.), *Proceedings of the Sixth International Workshop on Information Hiding*. Lecture Notes in Computer Science (3200), Berlin: Springer, pp. 180–91.

Brockett, C. and Dolan, W. B. (2005). Support vector machines for paraphrase identification and corpus construction. *Proceedings of the Third International Workshop on Paraphrasing*. Korea.

Chang, C. Y. and Clark, S. (2010). Linguistic steganography using automatically generated paraphrases. *Proceedings of the NAACL-HLT Conference*. Los Angeles.

Chen, Z., Huang, L., Yu, Z. *et al.* (2008a). Linguistic steganography detection using statistical characteristics of correlations between words. *Proceedings of the Tenth International Workshop on Information Hiding*. Santa Barbara, CA: Springer, pp. 224–7.

Chen, Z., Huang, L., Yu, Z, Zhao, X, and Zhao, X. (2008b). Effective linguistic steganography detection. *Proceedings of the Eighth IEEE International Conference on Computer and Information Technolog, CIT Workshops*. Sydney, Australia, pp. 224–9.

Cox, I., Miller, M. L., and Bloom, J. A. (2002). *Digital Watermarking*. San Francisco, USA: Morgan Kaufmann.

Duclaye, F., Yvon, F., and Collin, O. (2003). Learning paraphrases to improve a question-answering system. *Proceedings of the 10th Conference of EACL Workshop of Natural Language Processing for Question-Answering*. Budapest.

Ekbal, A. and Bandyopadhyay, S. (2009). Voted NER system using appropriate unlabeled data. *Proceedings of the Named Entities Workshop: Shared Task on Transliteration*. Suntec, Singapore, pp. 202–10.

Grothoff, C., Grothoff, H., Alkhutova, L., Stutsman, R., and Atallah, M. J. (2005). Translation based steganograph. *Proceedings of Information Hiding Workshop* (IH). Germany: Springer Verlag, pp. 213–33.

Gutub, A. and Fattani, M. (2007). A novel Arabic text steganography method using letter points and extensions. *World Academy of Science, Engineering and Technology*, **21**: 28–31.

Harbusch, K., van Breugel, C., Koch, U., and Kempen, G. (2007). Interactive sentence combining and paraphrasing in support of integrated writing and grammar instruction: a new application area for natural language sentence generators. *Proceedings of the 11th European Workshop in Natural Language Generation*. Germany: Association for Computational Linguistics, pp. 65–8.

Hatzigeorgiu, N., Gavrilidou, M., Piperidis, S. *et al.* (2000). Design and implementation of the online ILSP Greek corpus. *Proceedings of the Second International Conference on Language Resources and Evaluation*. Athens, Greece: European Language Resources Association, pp. 1737–42.

Kermanidis, K. L. (2009). *A Modern Greek Syntactic Toolset for Language Learning, Procedings of the Workshop on Informatics in Education (WIE 2009)*. Corfu, Greece.

Kermanidis, K. L. and Magkos, E. (2009). Empirical paraphrasing of modern Greek text in two phases: an application to steganography. *Proceedings of the 10th International Conference on Intelligent Text Processing and Computational Linguistics,* Mexico City. Lecture Notes in Computer Science (5449), Germany: Springer Verlag, pp. 535–46.

Kornai, A. (1992). Frequency in morphology. In Kenesei, I. (ed.), *Approaches to Hungarian*, Vol. 4. Jate, Szeged, pp. 246–68.

Kozareva, Z. and Montoyo, A. (2006). *Paraphrase Identification on the Basis of Supervised Machine Learning Techniques*. Lecture Notes in Artificial Intelligence (4139). Germany: Springer Verlag, pp. 524–33.

Levinson, S. and Wilkins, D. (2006). *Grammars of Space*. Cambridge: Cambridge University Press.

Manning, C. and Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Meng, P., Huang, L., Chen, Z., Yang, W., and Li, D. (2008). Linguistic steganography detection based on perplexity. *Proceedings of the International Conference on Multimedia and Information Technology*. Three Gorges, China, pp. 217–20.

Meral, H. M., Sevinc, E., Unkar, E., Sankur, B., Ozsoy, A. S., and Gungor, T. (2007). Syntactic tools

for text watermarking. In Delp, E. J.III and Wong, P. W. (eds), *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*.

Meral, H. M., Sankur, B., Ozsoy, A. S., Gungor, T., and Sevinc, E. (2009). Natural language watermarking via morphosyntactic alterations. *Computer Speech and Language*, **23**: 107–25.

Milicevic, J. (2008). Paraphrase as a tool for achieving lexical competence in L2. *Proceedings of the Symposium on Complexity, Accuracy and Fluency in Second Language Use, Learning and Teaching*. Brussels, Belgium: KVAB Belgium, pp. 153–67.

Murphy, B. and Vogel, C. (2007). Statistically-constrained shallow text marking: techniques, evaluation paradigm and results. *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX,* 6505. San Jose, USA, pp. 65050Z.

Nakov, P. (2008). Improved statistical machine translation using monolingual paraphrases. *Proceedings of the European Conference on Artificial Intelligence*. Patras, Greece.

Okamoto, H., Sato, K., and Saito, H. (2003). Preferential presentation of Japanese near-synonyms using definition statements. *Proceedings of the Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (IWP 2003). Sapporo, Japan.

Pang, B., Knight, K., and Marcu, D. (2003). Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. *Proceedings of the Conference on Human-Language Technology*. Edmonton, pp. 102–9.

Provos, N. and Honeyman, P. (2003). Hide and seek: an introduction to steganography. *Proceedings of the IEEE Conference on Security and Privacy*, **1(3)**: 32–44.

Quirk, C., Brockett, C., and Dolan, W. B. (2004). Monolingual machine translation for paraphrase generation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics. Stroudsburg, USA, pp. 142–9.

Raskin, V. and Nirenburg, S. (2003). *Ontological Semantics*. Cambridge: MIT Press.

Rus, V., McCarthy, P. M., Lintean, M. C., McNamara, D. S., and Graesser, A. C. (2008). Paraphrase identification with lexico-syntactic graph subsumption. *Proceedings of the Florida Artificial Intelligence Research Society*. Coconut Grove, Florida, pp. 201–6.

Seretan, V. (2008). *Collocation Extraction Based on Syntactic Parsing.* Ph.D. Thesis, University of Geneva.

Shinyama, Y., Sekine, S., and Sudo, K. (2002). Automatic paraphrase acquisition from news articles. *Proceedings of the Conference on Human-Language Technology*. San Diego, USA: Morgan Kaufmann. San Francisco, USA, pp. 313–8.

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2000). A practical chunker for unrestricted text. *Proceedings of the Conference on Natural Language Processing*. Greece: Patras, pp. 139–50.

Stutsman, R., Atallah, M. J., Grothoff, C., and Grothoff, K. (2006). Lost in just the translation. *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC)*. Dijon: ACM Press, pp. 338–45.

Taskiran, C., Topkara, U., Topkara, M., and Delp, E. (2006). Attacks on lexical natural language steganography systems. *Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents VIII,* 6072. San Jose, USA, pp. 97–105.

Topkara, M., Taskiran, C. M., and Delp, E. (2005). Natural language watermarking. *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*. San Jose, USA.

Topkara, M., Riccardi, G., Hakkani-Tuer, D., and Atallah, M. (2006a). Natural language watermarking: challenges in building a practical system. *Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents VIII,* 6072. San Jose, USA, pp. 106–17.

Topkara, U., Topkara, M., and Atallah, M. (2006b). *The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. Proceedings of ACM Multimedia and Security Workshop*. Geneva, Switzerland.

Topkara, M., Topkara, U., and Atallah, M. J. (2007). Information hiding through errors: a confusing approach. In Delp, E. J.III and Wong, P. W. (eds), *Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents IX,* 6505. San Jose, USA, pp. 1–12.

Wayner, P. (2002). *Disappearing Cryptography: Information Hiding: Steganography and Watermarking.* 2nd edn. San Francisco: Morgan Kaufmann.

Winstein, K. (1998). *Lexical Steganography Through Adaptive Modulation of the Word Choice Hash.* http://

alumni.imsa.edu/~keithw/tlex/lsteg.pdf (last accessed 9 May 2011).

Wouters, K., Wyseur, B., and Preneel, B. (2007). Lexical natural language steganography systems with human interaction. *Proceedings of the Sixth European Conference on Information Warfare and Security*. Shrivenham, UK, pp. 303–312.

Zhi-li, C., Liu-sheng, H., Zhen-shan, Y., Ling-jun, L., and Wei, Y. (2008). A statistical algorithm for linguistic steganography detection based on distribution of words. *Proceedings of the Third International Conference on Availability, Reliability and Security* (ARES). Barcelona, Spain, pp. 558–63.