2010 22nd International Conference on Tools with Artificial Intelligence

Hiding Secret Information By Automatically Paraphrasing Modern Greek Text With Minimal Resources

Katia Lida Kermanidis
Department of Informatics
Ionian University
Corfu, Greece
kerman@ionio.gr

Abstract— **Paraphrasing normally involves sophisticated external resources and semantic thesauri. This paper describes the automatic generation of Modern Greek paraphrases using statistical significance testing for extracting applicable syntactic reordering schemata. Next, supervised filtering helps remove erroneous schemata minding their context. As the paraphrases will be used in steganographic communication, they need not be sophisticated alterations, but significant in number. The proposed process is therefore knowledge-poor (portable to other languages with similar syntax), robust and domain-independent.**

paraphrasing; statistical significance testing; supervised learning; Modern Greek; linguistic steganography

I. Introduction

Linguistic steganography is a new field aiming to create systems for hiding information unremarkably underneath a cover text [2, 6]. A sentence can be expressed in many ways. This redundancy allows the insertion of bits within a sentence by transforming it syntactically [8] and/or semantically [3].

This work describes the automatic generation of Modern Greek (MG) shallow paraphrases to be used in steganographic communication. Unlike most previous approaches to paraphrasing [1, 3, 10] that focus on generating a few intricate alterations of a sentence and require sophisticated resources, the goals of this approach are different. The first is to produce as many correct paraphrases as possible, as security depends largely on the grammaticality and the number of extracted paraphrases. The second goal is to employ as limited resources as possible. This will allow the portability of the methodology to other languages that share certain syntactic properties with MG. Also, it will ensure robustness and domain independence.

II. Syntactic Transformations

The rich morphology of MG allows for freedom in the ordering of the chunks within a sentence. This enables paraphrase generation by changing only the chunk order.

The ILSP/ELEFTHEROTYPIA corpus [12] used in this work consists of 5244 sentences, is manually annotated morphologically, and balanced. Phrase structure information is obtained automatically by a low-resource chunker [11], that detects noun (NP), verb (VP), prepositional (PP), adverbial phrases (ADP) and conjunctions (CON) via multi-pass parsing.

A. Hypothesis Testing

Phrase types are formed by stripping text phrases from morphosyntactic information that is not essential for the task. The type patterns are [NP<case>], [<conjunction>VP<voice>], [PP<preposition>], [CON<sub/coordinating>], [ADP<(not)relative>]. 156 phrase types were formed.

The statistical significance of the cooccurrence of two phrase types is measured; the t-test, the LLR, the χ^2 metric and pointwise MI metrics have been experimented with. Phrase type pairs that occur in both orderings ([TYPE1][TYPE2] and [TYPE2][TYPE1]) among the top results with the highest rank are considered permissible phrase swaps, as both orderings show significant correlation between the phrases forming them.

The left columns in Table 1 show the size of the selected swap set (each pair is counted once) for various values for the number of the N-best results. For all N values, statistical significance proved to be well above the threshold for a=0.05.

In case a pair is detected in an input sentence, the two phrases are swapped and a paraphrase is produced. The average number of swaps per corpus sentence are shown in the center columns. If more than one phrase swaps are applicable at different positions in a sentence, all possible combinations of swaps are performed, and all respective paraphrases are produced, forming the *initial pool of paraphrases*.

TABLE I. Swap Set Size, Applicability and Error Rate

	Swap set size – Avg nr of swaps/sentence – Error rate(%)								
	Top50			Top100			Top200		
Ttest	21	3.8	27.8	38	4.2	29.1	67	4.6	29.7
LLR	11	2.2	34.8	31	2.5	35.5	49	2.8	37.1
χ^2	12	3.1	28.1	30	3.4	29.9	47	3.6	30.6
MI	16	0.6	33.1	19	0.6	35.1	36	0.9	35.4

Two native speakers judged 193 randomly selected sentences and their paraphrases on grammaticality. Inter-expert agreement exceeded 96% (kappa statistic). The percentage of paraphrases that required one or more phrase swaps to become grammatical is in the right columns.

MI returns a smaller but more diverse set of swap pairs that contains certain 'exclusive' phrase types, that are not included in the other sets. T-test returns a more extensive set of more frequent phrases in its pairs, and the smallest error rate.

B. Filtering

To reduce the error rate, the extracted swap sets undergo a filtering process (supervised classification), where erroneous swap pairs are learned (classified as (in)valid) and withdrawn.

The positions of possible phrase swaps in the input sentences are identified according to the top 200 T-test set. A learning vector is created for every input sentence and each swap position. The features forming the vector encode

1082-3409/10 $26.00 © 2010 IEEE
DOI 10.1109/ICTAI.2010.135

IEEE computer society

morphosyntactic information (phrase type, case, (in)definitiveness, preposition, conjunction type, adverb type) for the phrase right before the swap position, as well as two phrases to the left and two to the right. 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.

Unlike previous supervised learning approaches to paraphrase identification [7], the presented dataset does not consist of sentence pairs that are candidate paraphrases, but of single sentences that in certain positions allow (or not) the neighboring phrases to be swapped. Therefore, commonly employed features like shared word sequences and word similarity [7] are out of the scope of the present methodology and not abiding by the low resource policy.

Native speakers manually annotated the instances (vectors), corresponding to the same 193 sentences (981 instances) of the previous section. A support vector machines (SVMs) classifier (first degree polynomial kernel function, the sequential minimal optimization algorithm for training) was trained to classify instances using 10-fold cross validation. Classification performance reached 82% precision and 86.2% recall.

The correlation of each swap pair with the target class was estimated next. Swap pairs that appear more frequently in negative (invalid paraphrase) than in positive instances were removed from the final swap set (28 in number).

The reduced swap set was evaluated against a held-out test set (100 new corpus sentences) and reached an error rate of 17.6%. Against the 193-sentence training set, the error rate dropped to 14.3%. These results are quite satisfactory compared even to knowledge-rich approaches. The reduced set forms the *final pool of paraphrases*.

III. Application to Steganography

Unlike previous work that also employs syntactic alterations for information hiding [4, 8], the ones proposed here are simple, they have increased applicability and they may be applied in multiple positions to a sentence. Thereby, the number of extracted paraphrases increases.

Once the final pool for every sentence in the input text is formed, secret bits may be embedded in the text in 3 stages.

1. Each side of the bidirectional swap rules is marked with a 1-bit value.
2. For every sentence, the applicable swaps are selected from the swap set. If the sentence does not allow for any swap it is not used for information embedding. If it does, a selection is possible either in a round-robin fashion, or using a secret symmetric cryptographic key.
3. If the bit to be hidden matches the marking of the selected swap, the swap is not applied and the sentence remains as it is. Otherwise it is applied and the sentence is paraphrased.

On the other end, the extractor receives the final text. Having at his disposal the same swap set, he can identify the applicable swaps for each sentence. Sharing the same secret key, he can select the same swap used for insertion. 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.

Steganographic security relies on two factors. First, it is very important for the final text to be correct and natural, so as not to raise an eavesdropper’s suspicion. The low error rate after filtering ensures that. A second important security aspect is the pool size. Even if the eavesdropper suspects something, if the pool size is large enough, it will be non-trivial to decide upon the correct paraphrase. The pool size is significantly larger than that of approaches that apply more sophisticated transformations [8], due to the ‘generic’ nature of the transformations. Even so, ‘non-trivial’ is not ‘impossible’. To improve security further, a separate secret key may be used (comparable in length to the hidden message) to perform a bitwise logical operation of equivalence and encode the message before embedding it [3]. After extraction, the recipient decodes it by performing the reverse logical operation.

Steganographic capacity is the bandwidth available for hiding information. 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. Using the current implementation (with the initial swap set) 1 bit may 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. There is a trade-off between security and capacity: stricter (more accurate) syntactic schemata enhance accuracy and thereby security, but have low applicability (and capacity), and vice versa. This bandwidth may be further increased by exploiting the possibility of embedding more than one bits per sentence (at every swap position), a potential offered by the methodology.

References

[1] L. Bentivogli, I. Dagan, H. Dang, D. Giampiccolo, and B. Magnini, “The 5th PASCAL recognizing textual entailment challenge,” Proc. Text Analysis Conf., Maryland, USA, 2009.

[2] R. Bergmair, “Towards linguistic steganography: a systematic investigation of approaches, systems and issues,” BSc Thesis, University of Derby, 2004.

[3] I. A. Bolshakov, “A method of linguistic steganography based on collocationally-verified synonymy,” Proc. 6th Int. Workshop on Information Hiding, LNCS, vol. 3200, Springer, 2004, pp. 180–191.

[4] C. Y. Chang, and S. Clark, “Linguistic steganography using automatically generated paraphrases,” Proc. NAACL-HLT Conference, Los Angeles, 2010.

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

[6] A. Desoky, “NORMALS: Normal linguistic steganography methodology,” Journal of Information Hiding and Multimedia Signal Processing, Vol. 1, No. 3, pp. 145-171, 2010

[7] Z. Kozareva, and A. Montoyo, “Paraphrase identification on the basis of supervised machine learning techniques,” LNAI, Springer, vol. 4139, 2006, pp. 524-533.

[8] 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, 2009, pp. 107-125.

[9] N. Provos, and P. Honeyman, “Hide and seek: an introduction to steganography,” IEEE Security and Privacy, 2003, pp. 32-44.

[10] C. Quirk, C. Brockett, and W. B. Dolan, “Monolingual machine translation for paraphrase generation,” Proc. Conf. on Empirical Methods in Natural Language Processing, Barcelona, 2004, pp. 142-149.

[11] E. Stamatatos, N. Fakotakis and G. Kokkinakis, “A practical chunker for unrestricted text,” Proc. Conf. on Natural Language Processing, Patras, Greece, 2000, pp. 139-150.

[12] http://www.elda.fr/catalogue/en/text/W0022.html