Extracting Shallow Paraphrasing Schemata from Modern Greek Text Using Statistical Significance Testing and Supervised Learning

Katia Lida Kermanidis

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

Abstract. Paraphrasing normally involves sophisticated linguistic resources for pre-processing. In the present work Modern Greek paraphrases are automatically generated using statistical significance testing in a novel manner for the extraction of applicable reordering schemata of syntactic constituents. Next, supervised filtering helps remove erroneously generated paraphrases, taking into account the context surrounding the reordering position. The proposed process is knowledge-poor, and thus portable to languages with similar syntax, robust and domain-independent. The intended use of the extracted paraphrases is hiding secret information underneath a cover text.

Keywords: paraphrasing, statistical significance testing, supervised learning.

1 Introduction

Paraphrasing is expressing the meaning of a sentence using a different set of words and/or a different syntactic structure. Paraphrasing is useful in language learning, authoring support, text summarization, question answering, machine translation, textual entailment, and natural language generation. Significant research effort has been put into paraphrase identification [9, 6, 1], and generation [7, 2].

The present work describes the automatic inference of syntactic patterns from Modern Greek (MG) text for generating shallow paraphrases. The proposed methodology is a combination of a statistical significance testing process for generating 'swapable' phrase (chunk) pairs based on their co occurrence statistics, followed by a supervised filtering phase (a support vector machines classifier) that helps remove pairs that lead to erroneous swaps. A first goal is to produce as many correct paraphrases as possible for an original sentence, due to their intended use in steganographic communication [5], i.e. for embedding hidden information in unremarkable cover text [3, 8, 7, 11]. Among others, one way to insert hidden bits within a sentence is by taking advantage of the plural number of syntactic structures it can appear in, e.g. paraphrases. Steganographic security relies on the number and the grammaticality of produced paraphrases, not on their complexity [5]. Instead of focusing on few intricate alterations (common in previous work), the methodology aims at generating a significant number of paraphrases. Unlike the syntactic rules in previous

J.M. Sempere and P. García (Eds.): ICGI 2010, LNAI 6339, pp. 297–300, 2010.

work [7], each swapping schema (and different schemata simultaneously) may be applied multiple times (i.e. in multiple positions) to a sentence [5]. A second goal is to employ as limited external linguistic resources as possible, ensuring thereby the portability of the methodology to languages with similar syntax to MG, robustness and domain independence (the proposed alterations are applicable to any MG text).

2 Inferring Paraphrasing Schemata

MG is highly inflectional and allows for a large degree of freedom in the ordering of the chunks within a sentence. This freedom enables paraphrase generation merely by changing the chunk order. The ILSP/ELEFTHEROTYPIA corpus [4] used in the experiments consists of 5244 sentences and is manually annotated with morphological information. Phrase structure information is obtained automatically by a multi-pass parsing chunker that exploits minimal resources [10] and detects non-overlapping noun (NP), verb (VP), prepositional (PP), adverbial phrases (ADP) and conjunctions (CON). Next, *phrase types* are formed by stripping phrases from superfluous information. NP types retain the phrase case. VP types retain the verb voice, the conjunction introducing them and their copularity. PP types retain their preposition and CON types their conjunction type (coo/sub-ordinating). 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). Phrase type pairs that occur in both orderings ([TYPE1][TYPE2] and [TYPE2][TYPE1]) among the top results with the highest rank are selected. These are considered permissible phrase swaps, as both orderings show significant correlation between the phrases forming them. In case a swap pair is detected in an input sentence, the two phrases are swapped and a paraphrase is produced. The left column in Table 1 shows the size of the selected swap set and the average number of swaps that are permitted per sentence for each swap set for every metric (each pair is counted once), and various values for the N-best results. If more than one swap is applicable at different positions, all swap combinations are performed, and all respective paraphrases are produced.

As a first step towards evaluation, certain swap pairs that are incapable of producing legitimate swaps are removed from the sets, e.g. pairs like [Phrase][#] (# denotes end of sentence), [Phrase][CONcoo], [Phrase][CONsub] and their symmetrical pairs. Then, two native speakers judged the produced paraphrases of 193 randomly selected sentences, according to grammaticality and naturalness. Inter-expert agreement exceeded 96% using the kappa statistic. The percentage of paraphrases that required one or more manual swaps from the judges in order to become grammatical and/or natural is shown in the right column of Table 1. MI returns a smaller but more diverse set of infrequent swap pairs. Such phrase types are: copular VPs, genitive NPs, unusual PPs (e.g. PPs introduced by the preposition ως - until). This set leads to a small average number of swaps per sentence, and a high error rate. T-test returns a more extensive set of swap pairs that consist of more frequent phrase types and results in the smallest error rate. 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, inability to identify certain relative and adverbial expressions, to resolve PP attachment ambiguities, subordination dependencies etc.

Table 1. Swap set size and error rate for every metric

	Swap Set Size/Avg nr of swaps				Error rate			
	Top50	Top100	Top200	Top300	Top50	Top100	Top200	Top300
Ttest	21/3.8	38/4.2	67/4.6	92/4.9	27.8%	29.1%	29.7%	36.9%
LLR	11/2.2	31/2.5	49/2.8	77/3.0	34.8%	35.5%	37.1%	41.2%
χ^2	12/3.1	30/3.4	47/3.6	71/3.8	28.1%	29.9%	30.6%	37.7%
MI	16/0.6	19/0.6	36/0.9	60/1.4	33.1%	35.1%	35.4%	39.9%

To reduce the error rate, the extracted swap sets undergo a filtering process, where erroneous swap pairs are learned using supervised classification and withdrawn from the final pair sets. The positions of possible swaps are identified according to the T-test swap set for the top 200 results. A learning vector is created for every input sentence and each swap position for the 193 sentences. The features forming the vector encode syntactic information for the phrase right before the swap position, two phrases to the left and two phrases to the right. Thereby, context information is taken into account. Each of the five phrases is represented through six features (Table 2). Unlike previous supervised learning approaches to paraphrase identification [6], the presented dataset does not consist of candidate sentence-paraphrase pairs, but of single sentences that in certain positions allow (or not) the neighboring phrases to be swapped. So commonly employed features like shared word sequences and word similarity [6] are out of the scope of the methodology and not abiding by the low resource policy. A support vector machines (SVM) classifier (first degree polynomial kernel function, and SMO for training) classified instances using 10-fold cross validation. SVM were selected because they are known to cope well with high data sparseness and multiple attribute problems. Classification reached 82% precision and 86.2% recall. The correlation of each swap pair with the target class (valid/not valid paraphrase) was estimated next. 28 swap pairs that appear more frequently with the negative than with the positive class value were removed from the final swap set.

Table 2. The features of the learning vector

	NP	VP	PP	CON/ADP
1	NP	VP	PP	CON/ADP
2	case of phrase headword	-	-	-
3	NP is (in)definite	-	-	-
4	pronoun in NP (if any)	conjunction in VP	preposition	1st word lemma
5	contains (not) genitive element	verb is (not) copular	-	-
6	-	-	-	nr of words in phrase

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 17.6%. Against the 193-sentence training set, the error rate dropped to 14.3%. Given the 'knowledge poverty' of the approach, the results are satisfactory when compared to those of approaches that utilize sophisticated resources [7].

It is interesting to study the pairs that tend to lead to correct vs. incorrect swaps. PPs introduced by the preposition *για* (for) are usually attached to the sentence verb, and so may almost always be swapped with the preceding phrase. PPs introduced by the preposition *σε* (to) are more problematic. ADPs may usually be swapped with preceding NPs, but preceding VPs are confusing. Consecutive main verb phrases are rarely 'swapable'. Certain secondary clauses (e.g. final or relative clauses) may often be swapped with their preceding main verb phrase, but not with a preceding NP.

The use of other filters, the set of features for supervised learning, and the context window size should be further explored. Another challenging perspective would be to enlarge the window size between the phrases to be swapped, instead of focusing only on two consecutive chunks. This would increase paraphrasing accuracy.

References

1. Barzilay, R., Lee, L.: Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In: Proceedings of the Conference on Human Language Technology (HLT-NAACL), Edmonton, pp. 16–23 (2003)
2. Bentivogli, L., Dagan, I., Dang, H., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the Text Analysis Conference. Gaithersburg, Maryland (2009)
3. Cox, I., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2002)
4. Hatzigeorgiu, N., et al.: Design and Implementation of the online ILSP Greek Corpus. In: Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, pp. 1737–1742 (2000)
5. Kermanidis, K.L., Magkos, E.: Empirical Paraphrasing of Modern Greek Text in Two Phases: An Application to Steganography. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 535–546. Springer, Heidelberg (2009)
6. Kozareva, Z., Montoyo, A.: Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 524–533. Springer, Heidelberg (2006)
7. Meral, H.M., Sevinc, E., Unkar, E., Sankur, B., Ozsoy, A.S., Gungor, T.: Syntactic Tools for Text Watermarking. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505 (2007)
8. Provos, N., Honeyman, P.: Hide and Seek: An Introduction to Steganography. IEEE Security and Privacy, 32–44 (2003)
9. Rus, V., McCarthy, P.M., Lintean, M.C., McNamara, D.S., Graesser, A.C.: Paraphrase Identification with Lexico-syntactic Graph Subsumption. In: Proceedings of the Florida Artificial Intelligence Research Society, pp. 201–206 (2008)
10. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: A Practical Chunker for Unrestricted Text. In: Christodoulakis, D.N. (ed.) NLP 2000. LNCS (LNAI), vol. 1835, pp. 139–150. Springer, Heidelberg (2000)
11. Topkara, M., Taskiran, C.M., Delp, E.: Natural Language Watermarking. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, San Jose (2005)