A. Chitra and Anupriya Rajkumar*

Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer

DOI 10.1515/jisys-2014-0146
Received October 8, 2014; previously published online April 2, 2015.

Abstract: Plagiarism in free text has become a common occurrence due to the wide availability of voluminous information resources. Automatic plagiarism detection systems aim to identify plagiarized content present in large repositories. This task is rendered difficult by the use of sophisticated plagiarism techniques such as paraphrasing and summarization, which mask the occurrence of plagiarism. In this work, a monolingual plagiarism detection technique has been developed to tackle cases of paraphrased plagiarism. A support vector machine based paraphrase recognition system, which works by extracting lexical, syntactic, and semantic features from input text has been used. Both sentence-level and passage-level approaches have been investigated. The performance of the system has been evaluated on various corpora, and the passage level approach has registered promising results.

Keywords: Paraphrase recognition, passage-level plagiarism detection, support vector machine.

Classification according to MSC 2010: 68TXX – Artificial Intelligence, 68T50 – Natural Language Processing.

1 Introduction

The Internet has facilitated instant information access and has also led to the spawning of large amounts of unstructured data, especially text. A major drawback of the easy information access made possible by the Internet is the widespread prevalence of the phenomenon of copying and reusing information without permission. Plagiarism can be termed as the unauthorized reuse of content or ideas without giving due credit to the original authors. Plagiarism is a major threat to academics and has to be suitably addressed to ensure integrity and authenticity.

Literal plagiarism includes copy–paste operations and is usually easy to detect. More sophisticated forms of plagiarism may involve translation, summarization, and paraphrasing and are more difficult to recognize [1]. One of the most difficult to detect and relatively less addressed forms is paraphrased plagiarism in which the original content may be completely reworded and altered considerably [4]. Plagiarism detection systems focus on ensuring the originality of text content. Such systems are categorized as intrinsic and external detectors [15]. Intrinsic detectors attempt to identify plagiarism by analyzing the writing style variations within a single document. Extrinsic detectors compare a suspicious document against a large set of source documents and identify the plagiarized portions, if any, by first choosing a set of candidate source documents and then assessing similarity between the suspicious document and identified candidates.

The widespread usage of paraphrasing techniques for plagiarizing text has motivated the current work. The objective of this work is to investigate the suitability of utilizing a machine learning-based paraphrase recognition system for plagiarism detection. Various lexical, syntactic, and semantic features, which reflect

***Corresponding author: Anupriya Rajkumar,** Assistant Professor, CSE Department, Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India, e-mail: anupriya_rajkumar@yahoo.co.in
A. Chitra: Professor and Head, Computer Applications, PSG College of Technology, Coimbatore, India

the degree of similarity between the source and suspicious text, are extracted. These are used as input to a support vector machine classifier, which determines if the source text has been plagiarized. Section 2 of the paper describes related work, and Section 3 details the development of the plagiarism detector and the underlying paraphrase recognition system. Section 4 presents the results of experiments performed in order to evaluate the plagiarism detector. The conclusion and future work are discussed in Section 5.

2 Related Work

Plagiarism detectors usually characterize unstructured documents using various categories of textual features such as lexical, syntactic, and semantic. The most popular lexical features are character and word n-grams, while parts-of-speech (POS) information is used extensively to compute syntactic features. Semantic features depend on a thesaurus like WordNet to typify word relationships. Given a large document collection, in order to retrieve the candidate source documents for matching against the suspicious document, traditional information retrieval techniques based on cosine similarity, vector space model, and fuzzy retrieval may be used. Once the candidate documents are identified, they can be compared exhaustively using techniques based on string matching, vector similarity computation, syntax, semantic, fuzzy, and structural feature-based methods. Semantic and fuzzy methods are more effective in detecting complex types of plagiarism including paraphrasing and restructuring besides the simpler copy–paste forms [1].

Clough et al. [10] have carried out some of the earliest experiments in the domain of text reuse detection and have also constructed the METER (MEasuring TExt Reuse) corpus. The METER corpus consists of text articles collected from the UK Press Association (PA) and nine British newspapers with the PA articles providing a source for the newspaper articles. The extent of text reuse has been assessed using n-gram overlap, Greedy String Tiling and sentence alignment. Clough and Stevenson have developed a corpus of plagiarized short answers [9] labeled as the Wikipedia Rewrite Corpus. The corpus was created for five questions from the Computer Science domain using candidate answers generated by participants, either independently or through the modification of the reference answer extracted from Wikipedia by various degrees. The similarity was assessed in terms of n-gram overlap and longest common subsequence.

More recently the PAN-PC competitions have generated considerable interest in this domain and have led to the development of several successful systems, which work on large-scale document collections. Some of the approaches used include winnowing, hash function computation, finger-printing, and exact matching at various levels such as character-n-grams, word-n-grams, sentences [15, 16].

Despite the large number of plagiarism detection alternatives, the identification of paraphrased plagiarism has not been fully addressed [4]. As the amount of lexical variation between the text units increases, plagiarism detection becomes tougher. Barron-Cedeno et al. have defined a typology of paraphrases comprising of 22 types based on the nature of changes such as morpho-lexicon based, structural, semantic, and miscellaneous. The authors of [4] have annotated a subset of the PAN-PC 2010 corpus according to their typology to create the Paraphrasing for Plagiarism (P4P) corpus. The authors have also analyzed the performance of the PAN-PC 2010 competitors on the P4P corpus and have observed that though the systems perform well on the entire PAN-PC 2010 corpus, they perform poorly on the P4P corpus, which involves extensive paraphrasing.

In an effort to focus on paraphrased plagiarism, subsequent PAN competitions have introduced multiple cases of simulated plagiarism, which were created by workers on Amazon's Mechanical Turk by rewriting original text content. While paraphrase recognition systems usually work on phrase-level or sentence-level inputs to determine semantic similarity, plagiarism detection systems operate at the passage level [5]. Burrows et al. have adopted the crowdsourcing approach to create paraphrased versions of text passages for constructing the Webis Crowdsourced Paraphrase Corpus (CPC). The corpus was originally developed as a part of the PAN 2010 competition to test the efficiency of plagiarism detection systems. The authors have also assessed the performance of various paraphrase similarity metrics for automatically filtering the generated paraphrases. The metrics include normalized edit distance, n-gram comparison-based measures such as simple word n-gram overlap, BLEU metric, and longest common prefix n-gram overlap, besides the Sumo metric and various asymmetrical paraphrase detection functions proposed by Cordeiro et al. [11]. Burrows et al. have concluded that using a combination of these metrics with a machine-learning classifier yields the best results [5].

Bar et al. have combined three categories of features based on the content, structure, and style for measuring text reuse [3]. Content-based features were generated by comparing the text content and include string similarity measures, greedy string tiling, word and character n-gram features, Wordnet-based semantic similarity measures besides latent semantic analysis and explicit semantic analysis. Structural similarity was assessed in terms of word pair order, distance, as well as stop-word and POS n-grams. Stylistic similarity was determined using sentence, token length properties, function word frequencies, and vocabulary richness measures such as sequential type-token ratio. The approach was tested on three different corpora, namely, Webis CPC, Wikipedia Rewrite Corpus, subset of METER corpus, and combining the three categories was found to yield the best results in two out of three cases.

From the study of related work, it is observed that paraphrased plagiarism, though common, has not been addressed satisfactorily. Hence, there is a need for efficient plagiarism detection approaches, which can handle paraphrased plagiarism.

3 Methodology

In this work, a machine learning-based paraphrase recognizer, which operates by extracting lexical, syntactic, and semantic features, has been used to detect plagiarism in text passages. The sentence-level paraphrase recognition system reported in [8] has been adapted for determining if two passages have been plagiarized. Two different approaches have been investigated: in the first, the input source and suspicious passages have been split into sentences, and the original sentential paraphrase recognition system has been applied. In the second approach, the input passages have been retained as it is, and various features extracted from the passages have been used to judge whether the suspicious passage is a plagiarized version of the source.

3.1 Paraphrase Recognition System

A support vector machine-based paraphrase recognizer [8] has been used to classify the input sentence pair as paraphrases by extracting various lexical, syntactic, and semantic features from the sentence pair. An initial version of the paraphrase recognizer was constructed using a total of 113 features as listed in Table 1.

Table 1: Paraphrase Recognition Features.

Category	Feature (Count)	Description
Lexical	BLEU-1 precision and recall (2)	Extent of unigram match with respect to each of the input sentences
	Skipgram precision and recall (2)	Ratio of number of common skipgrams to total possible skipgrams constructed from each of the input sentences
	Longest Common Subsequence (1)	Ratio of the length of the longest common in-sequence portion to the length of the shorter sentence
Syntactic	Tree Edit distance (2)	Ratio of minimum number of operations required to transform one dependency tree into another to the number of nodes in the input trees
	Triple similarity function (2)	Ratio of number of shared triples to the number of triples in the input sentences
	POSPER features (96)	POS position independent word error rate features formed by considering the degree of matches and non-matches for 48 POS tags
Semantic	Word similarity (4)	Jiang-Conrath score calculated on nouns, verbs, adjectives, and adverbs
	Proper noun match (1)	Extent of proper noun matches
	Negation features (3)	Presence of antonyms and explicit negation terms

Skipgrams of an input sentence are formed by considering both contiguous and noncontiguous n-grams. In this work, skipgrams with n = 1, 2, 3 have been formed with a maximum skip length of 4. The TreeTagger [17] has been used to stem the words in the input sentence and assign POS tags. The Stanford parser, developed by Klein and Manning [13], was used to construct dependency trees, which are syntactic representations of a sentence. From the dependency tree representation, triples, consisting of parent, child, and relationship between them were formed, and the triple similarity function was assessed. Word similarity has been determined using the Jiang–Conrath score based on WordNet [12]. The score quantifies the distance between two words in terms of the information content of the words and their least common ancestor in the WordNet hierarchy.

After feature extraction, for classifying the input sentences as paraphrases, a support vector machine classifier was used. The LibSVM tool [6] has been used for performing SVM classification. The 113 features extracted from the training set of the Microsoft Research Paraphrase Corpus (MSRPC) consisting of 4076 sentences [7] were used to train the SVM Classifier. Evaluation using the MSRPC test set of 1725 sentences yielded an accuracy of 75.58% [8]. Wrapper-based feature selection using genetic algorithms (GA) was then employed to improve the performance of the paraphrase recognizer. By using 57 of the original 113 features, an accuracy of 76.97% was achieved. Previous paraphrase recognizers have registered accuracy values in the range of 70%–77.4% with respect to MSRPC [2, 14]. The best features identified by GA-based feature selection include [8]:

- All five lexical features
- Verb, adverb similarity, and two of the negation features from the semantic category
- Dependency tree edit distance, triple similarity function from the syntactic category in addition to POSPER features corresponding to simple and comparative adjectives, singular and plural nouns, “be” forms of the verb.

3.2 Plagiarism Detection

As the input for the task of plagiarism detection is passage-level text, the sentence-level paraphrase recognition system has been modified to handle passages. The source and suspicious passages are split into sentences. In order to determine the closest matching source sentence for the suspicious passage sentences, the extent of unigram overlap is computed between the sentences in both the passages. For every sentence in the suspicious passage, the source sentence, which has the highest word overlap, is paired with it. The best set of 57 features identified for paraphrase recognition, as described in Section 3.1, are then extracted from the sentence pair. An SVM classifier is then used to label the sentence pairs as positive or negative cases of paraphrases. The decisions obtained for individual sentence pairs are then aggregated by computing the percentage of paraphrases among the total number of matched pairs. If the percentage exceeds a given threshold, the entire suspicious passage is declared to be plagiarized. The sentence-level processing algorithm is given below:

Algorithm: Plagiarism Detection by Sentence-level processing

```
Input: Pairs of Original and Candidate Suspicious passages
       Threshold T in the range (0, 1)
Output: Yes (Plagiarized)/No (Not-Plagiarized)
Procedure:
  – Split the passage pairs into training set (two-thirds) and test set (one-thirds)
  – For each pair of passages $P_o$ and $P_c$:
    Split the passages into sentences
    For each sentence $S_c$ in the candidate passage:
      Determine the word overlap with each sentence of the original passage
      Identify the sentence from the original passage – $S_o$ having the highest overlap
      Extract the set of 57 features from the sentence pair ($S_c$, $S_o$)
  – Train the SVM Classifier using features extracted from the training set
  – Classify the sentences extracted from the test-set passages as ‘Plagiarized’ or ‘Not-Plagiarized’
```

```
For each candidate suspicious passage $P_c$:
  Determine the total number of plagiarized sentences-$np_c$
  Compute the aggregate score by dividing $np_c$ by number of sentences in $P_c$
  If the score exceeds the given threshold T, the candidate passage is declared as a plagiarized version of
  the original passage $P_o$
```

Another scheme termed as “PassagePlag,” which operates directly on passage-level text, has also been adopted. In this method, paraphrase recognition features are extracted directly from the source and suspicious passage as a whole. Out of the best set of 57 features identified for paraphrase recognition, the text edit distance, triple similarity function features have been eliminated for passage-level inputs. This is due to the reason that these features are extracted from dependency parse trees, which are constructed for sentences and not lengthy passages. In this case, the SVM classifier is used to directly decide whether the passages are paraphrased and, hence, plagiarized.

4 Results

The suitability of the proposed approaches for the task of plagiarism detection has been investigated using four different corpora: the Webis CPC corpus [5], which is a subset of the PAN PC 2010 corpus, a subset of the METER corpus [10], Wikipedia Rewrite corpus [9], and a subset of the P4P corpus [4]. The Webis corpus contains a total of 7859 pairs of source and suspicious passages out of which 4067 have been labeled as positive cases of paraphrasing and the rest as negative. The positive cases vary in length from 28 to 954 words. The corpus has been constructed by crowdsourcing on Amazon’s Mechanical Turk. Volunteers were asked to paraphrase passages of text extracted from Project Gutenberg. The generated passages were reviewed to reject as nonparaphrases those cases that were exactly the same or very similar to the original. Of the remaining cases, grammatically correct versions, which conveyed the same meaning as the source, have only been accepted as paraphrases [5].

The METER corpus consists of 1716 text articles extracted from the UK Press Association releases and different newspapers [10]. Each newspaper article is a rewritten version of the corresponding PA source(s). The entire corpus collected over a period of 12 months has been grouped into two major domains – courts and show business. In order to reflect the extent of text reuse, each newspaper article has been manually categorized as “Wholly Derived,” “Partially Derived,’” or “Not Derived.” For the purpose of the current study, only a subset of 253 articles, which have a single source, have been chosen similar to the approach followed by Bar et al. [3] and Sanchez-Vega et al. (2010). Further, the three-way classification has been converted into two classes: Derived and Not Derived.

The Wikipedia Rewrite Corpus consists of 95 short answers collected from 19 participants for five different questions [9]. The collected answers have originally been labeled as “Near Copy,” “Light Revision,” “Heavy Revision,” and “Non-plagiarism” depending on their similarity to the reference answer. This four-way split has been converted into two classes: Plagiarized and Non-Plagiarized.

In order to investigate the performance of the paraphrase recognition system in handling various types of paraphrases, experiments were conducted using the P4P corpus. The P4P corpus consists of 847 pairs of fragments each containing <50 words [4]. The corpus has been manually annotated at various levels such as words, phrases, clauses, and sentences using the paraphrase typology. As the original paraphrase recognizer described in Section 3.1 operates at the sentence level, types involving only word-level or phrase-level changes have been eliminated. These include all the subtypes falling under the morpho-lexicon and miscellaneous categories. Only 10 subtypes falling under the structural and semantic categories have been considered for evaluation purposes.

The evaluation measures considered here are Accuracy, Precision, Recall, and F-measure, which are calculated as given in eqs. (1)–(4), where a case of plagiarism refers to a pair of original and candidate passages. True Positive (TP) refers to a plagiarized passage being labeled as plagiarized and True Negative (TN) is a correctly identified case of nonplagiarism. False Positive (FP) refers to nonplagiarized cases labeled as plagiarized, while False Negative (FN) is the vice versa.

$$\begin{aligned}\text{Accuracy }(\text{A}) &= \text{number of correctly labeled cases / total number of cases}\\ &= \text{TP}+\text{TN}/(\text{TP}+\text{TN}+\text{FP}+\text{FN})\end{aligned} \tag{1}$$

$$\begin{aligned}\text{Precision}(\text{P}) &= \text{number of correctly detected cases of plagiarism / reported cases of plagiarism}\\ &= \text{TP}/(\text{TP}+\text{FP})\end{aligned} \tag{2}$$

$$\begin{aligned}\text{Recall}(\text{R}) &= \text{number of correctly detected cases of plagiarism / actual cases of plagiarism}\\ &= \text{TP}/(\text{TP}+\text{FN})\end{aligned} \tag{3}$$

$$F-\text{measure}(F) = (2*\text{Precision}*\text{Recall})/(\text{Precision}+\text{Recall}) \tag{4}$$

The evaluation of the passage-level approach – PassagePlag has been carried out using a 10-fold cross validation approach. On the other hand, as the evaluation of the sentence-level approach requires the decisions made on sentence pairs to be aggregated, the corpora were partitioned into three folds of equal number of passages. Similar to the cross-validation technique three trials were carried out, in each of which two folds were used for training, and one fold was used for testing. Paraphrase recognition features extracted from paired sentences in the training set were fed to an SVM classifier and used to build a model. This was then used to classify the sentence pairs of the test set. The output decisions produced by the classifier for sentence pairs from the passages were aggregated to arrive at passage-level decisions as described in Section 3.2. The results of the sentence-level as well as passage-level approaches on the three corpora are shown in Table 2.

The threshold used to arrive at the passage-level decision was varied, and the best results were obtained when the threshold was set at 60% for the Webis corpus and 50% for the other two corpora. In the case of the sentence-level approach, the reported values have been arrived at by averaging the results obtained in the three trials. From Table 2, it can be observed that the overall performance of the passage-level approach is better than that of the sentence-level approach. Precision and recall exhibit the traditional tradeoff with precision being better in the passage-level approach, whereas the sentence-level approach has better recall. The better performance of the passage-level approach can be attributed to the reason that when passages are paraphrased, a sentence-by-sentence approach may not always be adopted as observed in [5]. Two sentences in the original passage may either be combined, or a single sentence maybe split. Hence, establishing a one–one correspondence between sentences of the source and suspicious passages becomes difficult. In the current approach, the degree of word overlap between the sentences has been used to pair the sentences for further comparison. Other alternatives such as semantic similarity can be considered, as two sentences that do not have a high word overlap may be paraphrases. When the entire passage is considered as a single entity, this disadvantage is overcome leading to better performance.

In order to benchmark the performance of PassagePlag on the three different corpora, the obtained results have been compared with that of Bar et al. and are tabulated in Table 3. From the results, it can be observed that for the Webis corpus, the proposed passage level paraphrase recognition approach – PassagePlag has slightly lower performance than that of Bar et al., which is the best performing system on the Webis corpus [3].

Table 2: Performance of Sentence-Level and Passage-Level Approaches.

Corpus	Webis corpus		METER corpus subset		Wikipedia Rewrite corpus	
Approach	Sentence level	Passage level	Sentence level	Passage level	Sentence level	Passage level
Accuracy %	79.08	83.48	78.66	81.42	95.83	96.84
Precision %	73.41	78.72	82.90	88.07	98.15	100
Recall %	93.41	93.31	88.40	85.64	94.81	94.74
F-measure %	82.21	85.40	85.56	86.83	96.43	97.29

Table 3: Performance Comparison on Various Corpora.

Approach	Webis corpus		METER corpus		Wikipedia Rewrite corpus	
	Accuracy %	F-measure %	Accuracy %	F-measure %	Accuracy %	F-measure %
PassagePlag	83.5	85.4	81.4	86.8	96.8	97.3
Bar et al. [3]	85.3	86.2	80.2	85.8	96.8	97.3

With respect to the METER sub-corpus and Wikipedia Rewrite Corpus, our approach was compared with that of Bar et al., which is again the best performing approach [3]. The passage-level approach was found to exhibit better or comparable performance. Both of these corpora contain samples belonging to nonplagiarized category as well as varying degrees of plagiarism. The data was folded to two classes: Plagiarized and Non-Plagiarized.

Table 4 presents the statistics of true, false positives, and negatives for the various corpora. The experimental results indicate that the paraphrase recognition approach used in this work reports a higher number of false positives and fewer false negatives when compared to Bar et al.'s system with respect to Webis corpus. The difference is less pronounced with respect to the METER corpus, and there is very little difference in the case of the Wikipedia Rewrite corpus. The increased number of false positives reported could be due to the excessive dependence on lexical features. As the MSRPC has been found to be lenient toward paraphrases with greater word overlap, lexical features were found to be a good choice. But the annotation of the Webis corpus has been carried out to overcome this bias by specifically labeling duplicates and near duplicates as nonparaphrases [5]. In the case of the METER corpus, even if the candidate passages have a considerable overlap, the presence of additional content in one passage has led to the passages being labeled as nonplagiarized.

In addition to the above experiments, the performance of the paraphrase recognition system was assessed on various categories of fragments available in the P4P corpus. As all the fragments classified under the syntactic and semantic categories are positive cases of paraphrasing, for evaluation purposes, samples extracted from the corpus of 1999 negative samples created by Cordeiro et al. (2007) in their work on sentence compression [11] have been used. For each subcategory containing Y-positive samples, an equal number of negative samples N was added. The results are presented in Table 5 and are ranked based on accuracy.

Most of the categories exhibit a good performance with accuracy >95%. The best performing categories are coordination, punctuation, and format where the paraphrased versions are very much similar to the original. The two categories with the lowest performance are "ellipsis" and "semantic-based changes." The reduced performance for the "semantic changes" fragments is due to the reason that this is the toughest category to detect as it involves considerable variation from the original. For the ellipsis category, the omission of words or phrases as in the pair "the long initial vowel of area" and "area" results in less overlap and, therefore, reduced performance. A study of the inputs, which have been wrongly classified as nonparaphrases, indicate the following reasons:

– Very low lexical overlap between the original and candidate inputs
– Presence of phrases

Table 4: Performance Statistics for Various Corpora.

Approach	Webis corpus		METER corpus		Wikipedia Rewrite corpus	
	PassagePlag	Bar et al. [3]	PassagePlag	Bar et al. [3]	PassagePlag	Bar et al. [3]
True positives	3795	3654	160	151	54	55
True negatives	2766	3033	46	52	38	37
False positives	1026	759	26	20	0	1
False negatives	272	413	21	30	3	2

Table 5: Performance on a Subset of the P4P Corpus.

Type	Description	Number of positive pairs	Accuracy %
Coordination	Change in coordinated linguistic units	210	97.8
Punctuation and format	Changes in punctuation and format	538	97.5
Syntax/discourse Structure	Syntax/discourse reorganizations	313	97.2
Sentence modality	Change in sentence modality	35	97.1
Subordination and nesting	Changes in subordinated or nested units	597	96.4
Direct and indirect style	Direct and indirect style variations	36	95.8
Diathesis	Alternations such as voice change	130	95.8
Negations	Changing the position of negation	33	95.4
Ellipsis	Omission of words or phrases	87	94.3
Semantics based	Different lexicalization of same content	340	91.6

These are demonstrated in the following examples:

- Example 1: “it has been endeavored” and “the attempt has been made”(Semantic changes)
- Example 2:“inspire to better attentiveness” and “excite us to greater diligence” (Ellipsis)
- Example 3: “let’s stop emptying our heads” and “don’t let us split hairs”(Negations)

The good performance on various corpora, as well as the different categories of the P4P corpus, indicates the suitability of the current approach for detecting paraphrased plagiarism. Some of the possible directions for future work include:

- Improving the underlying paraphrase recognition system to handle semantically similar inputs having greater lexical variation.
- Handling exactly similar or very similar inputs, which do not qualify as paraphrases but can be grouped under the copy–paste plagiarism category.

5 Conclusion

Plagiarism of text has become a common occurrence today with difficult to detect forms such as paraphrasing and summarizing being frequently practiced. Hence, there is a need to design effective mechanisms for automatic plagiarism detection. In this work, a paraphrase recognition approach has been used to detect the occurrence of plagiarism in source and suspicious passages. The system has been tested on three different corpora: Webis CPC, subset of METER, and Wikipedia Rewrite Corpus, at both the passage level as well as the sentence level, with the passage-level approach demonstrating better performance. The system has also exhibited comparable or better performance when compared to the best performing system on the three corpora. Further, the system was also tested on various subcategories of the P4P corpus with good results. This shows that employing paraphrase recognition techniques is a promising direction to explore in the development of plagiarism detection systems.

Bibliography

[1] S. Alzahrani, N. Salim and A. Abraham, Understanding plagiarism linguistic patterns, textual features and detection methods, *IEEE T. Syst. Man Cyb.* **42** (2011), 133–149.

[2] I. Androutsopoulos and P. Malakasiotis, A survey of paraphrasing and textual entailment methods, *J. Artif. Intell. Res.* **38** (2010), 135–187.

[3] D. Bar, T. Zesch and I. Gurevych, Text reuse detection using a composition of text similarity measures, in: *Proceedings of COLING 2012*, pp. 167–184, Mumbai, December 2012.

[4] A. Barrón-Cedeño, M. Vila, M. A. Martí and P. Rosso, Plagiarism meets paraphrasing: insights for the next generation in automatic plagiarism detection, *Comput. Linguist.* **39** (2013), 917–947.

[5] S. Burrows, M. Potthast and B. Stein, Paraphrase acquisition via crowdsourcing and machine learning, *ACM T. Intell. Syst. Technol.* **4** (2012), 1–21.

[6] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, *ACM T. Intell. Syst. Technol.* **2** (2011), 1–27, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] D. L. Chen and W. B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: *Proceedings of the 49th Annual Meeting of ACL*, pp. 90–200, Portland, USA, June 2011.

[8] A. Chitra and A. Rajkumar, Genetic algorithm based feature selection for paraphrase recognition, *Int. J. Artif. Intell. Tool.* **22** (2013), 1350007.1-17.

[9] P. Clough and M. Stevenson, Developing a Corpus of Plagiarized Short Answers, Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis **45** (2011), 5–24.

[10] P. Clough, R. Gaizauskas and S. L. Piao, Building and annotating a corpus for the study of journalistic text reuse, in: *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 1678–1691, Spain, May 2002.

[11] J. Cordeiro, G. Dias, and P. Brazdil, New functions for unsupervised asymmetrical paraphrase detection, *J. Software* **2** (2007), 12–23.

[12] J. Jiang and D. W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan, pp. 19–33, September 1997.

[13] D. Klein and C. D. Manning, Accurate unlexicalized parsing, in: *Proceedings of 41st Meeting of ACL*, pp. 423–430, 2003.

[14] N. Madnani, J. Tetreault and M. Chodorow, Re-examining machine translation metrics for paraphrase identification, in: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.182–190, Montreal, Canada, 2012.

[15] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño and P. Rosso, Overview of the 1st International competition on plagiarism detection, in: *Proceedings of Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09)*, B. Stein, P. Rosso, E. Stamatatos, M. Koppel and E. Agirre, eds., pp. 1–9. Spain, September 2009.

[16] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño and P. Rosso, Overview of the 2nd International competition on plagiarism detection, in: *Notebook Papers of CLEF 2010 Labs and Workshops*, M. Braschler, and D. Harman, eds., Italy, September 2010.

[17] H. Schmid, Probabilistic part-of-speech tagging using decision trees, in: *Proceedings of International Conference on New Methods in Language Processing*, pp. 44–49, Manchester, September 1994.