©2010 International Journal of Computer Applications (0975 - 8887)
Volume 1 – No. 29

Paraphrase Recognition using Neural Network Classification

Anupriya Rajkumar
Assistant Professor, Department of PG CSE,
Dr.Mahalingam College of Engineering and
Technology, Pollachi, Tamil Nadu, India

Dr.A.Chitra
Professor, Department of CSE,
PSG College of Technology,
Coimbatore, Tamil Nadu, India

ABSTRACT

Paraphrasing refers to conveying the same content in several ways. The successful recognition of paraphrases is crucial to various natural language processing tasks such as Information Extraction, Document Summarization, Question Answering etc. Several techniques have been employed for paraphrase recognition using lexical, syntactic and semantic features. Many of these systems have been tested on the MicroSoft Research Paraphrase Corpus. But the performance of these systems has scope for further improvement. Since neural network architectures model the human brain structure which excels at natural language processing tasks, this paper presents a neural network classifier for recognizing paraphrases. A combination of lexical, syntactic and semantic features has been used to train a Back Propagation network. The system can be utilized for detecting similar sentences in applications such as Question Answering and detection of plagiarized content.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text Analysis; I.2.6 [Learning]: Connectionism and Neural nets

General Terms

Algorithms, Performance

Keywords

Paraphrase Recognition, Lexical, Syntactic, Semantic features, Neural Network Recognizer, Back Propagation Network

1. INTRODUCTION

Natural Language Processing (NLP) focuses on developing computer systems that can analyze, understand and generate natural human-languages. One of the major difficulties faced in natural language processing is ambiguity where the same text has several possible interpretations. Another equally challenging aspect is that the same content can be conveyed in different ways. This is termed as Paraphrasing. Paraphrases can occur at the word level, phrase level, sentence level or discourse level. A typical example of sentence level paraphrasing is the following pair of statements “Tata acquires Jaguar” and “Jaguar sold to Tata”.

Research problems related to paraphrasing are Paraphrase generation, Paraphrase extraction and Paraphrase recognition. Paraphrase generation which is a Natural language generation problem is the process of generating alternative forms of the input text. This finds application in areas such as document summarization and machine translation. Paraphrase Extraction involves the identification or discovery of paraphrases from a large corpus and finds application in Information Extraction tasks.

Paraphrase recognition is the task of recognizing the presence of paraphrases in a given corpus. A variant of this is the text entailment problem, which takes two sentences as input and decides whether one of them can be inferred from the other. Paraphrasing is considered as a case if bi-directional text entailment. Paraphrase recognition is applicable in domains such as Information Extraction, Plagiarism detection and Question Answering. Experiments in the English language paraphrasing domain have been carried out on various notable corpora such as the Microsoft Research Paraphrase Corpora (MSRPC), Recognizing Text Entailment Corpora and Machine Translation resources. Similar experiments have also been carried out for Spanish, Japanese and Chinese languages. Commonly used scoring metrics for rating the performance of a system are Precision, Recall and F-measure.

This paper presents the work carried out on paraphrase recognition using a Neural network classifier. Though various machine learning techniques such as Decision trees, Support Vector Machines have been employed for the task no reported work exists on Neural Network based paraphrase recognizers. Section 2 of this paper gives an overview of features suitable for Paraphrase recognition. Section 3 of the paper details several techniques used in Paraphrase recognition. Section 4 presents the work on the Neural Network based paraphrase recognizer.

2. FEATURES FOR RECOGNIZING SEMANTIC EQUIVALENCE

This section briefly discusses the various features of text [1] which help to recognize Paraphrases. The features can be classified as Lexical, Syntactic and Semantic. Composite features can be formed by combining two aspects such as the Lexical and Semantic attributes.

2.1 Lexical Features

These characterize the surface similarity or degree of word overlap between the candidate sentences. A list of lexical features used in paraphrase recognition is given below.

- Unigrams – measures the number of shared words between the two sentences. Unigram precision and recall are the number of shared words divided by the length of the first sentence and second sentence respectively. Lemmatized unigram precision and recall are calculated after replacing words by their lemmas.
- Word error rate (WER) (Su et al., 1992) - a measure of the number of edit operations required to transform one sentence into another. It is also termed as Levenshtein Edit distance.
- Position-independent word error rate (PER) (Tillmann et al., 1997) - Similar to WER except that word order is not taken into account.
- Bi-Lingual Evaluation Understudy (BLEU) precision score (Papineni et al., 2001) - based on the geometric mean of n-gram matches. After reversing the order of the sentences, the BLEU recall score is calculated.
- Longest Common Substring and Subsequence – identify the longest common sequence of consecutive and non-consecutive words shared by the input sentence pair respectively.
- Modified N-gram precision - a variation of the BLEU measure which considers directional n-gram matches between the sentence pair.
- N-gram overlap measures – N-grams are sub-sequences of n-items from a given sequence. N-gram overlap measures identify the number of shared n-grams between the sentences.
- Skip-gram overlap measures – Skip-grams are non-consecutive sequences of words using a skip distance k. Skip-gram overlap measures are calculated by dividing the number of common skip-grams by the number of word combinations in the sentences.
- Exclusive longest common prefix N-gram overlap – This measure extends the simple n-gram overlap measure. It disregards all lower order subgrams of a maximal n-gram when the number of overlapping n-grams is calculated.

2.2 Syntactic Features

These analyse the degree of structural similarity between the pair of sentences. Some of the commonly used Syntactic features are:

- Dependency tree edit distance - A dependency tree is a syntactic representation of a sentence. Dependency tree edit distance measures the similarity of dependency trees.
- Dependency relation overlap features - A dependency relation is a pair of words with a parent-child relationship within the dependency tree. Dependency relation overlap features measure the extent of overlap of dependency relations between the two sentences.
- The morphological variants feature - identifies the Co-occurrence of morphological variants in sentence pairs. The words “compute” and “computing” are morphological variants.

2.3 Semantic Features

Several Semantic similarity features exist based on the WordNet database. These measures are termed as Knowledge based measures as they rely on additional resources such as the Wordnet dictionary. In the WordNet taxonomy, nodes represent concepts or words and edges represent the relations between the concepts. The Knowledge based measures [13] include:

- Leacock and Chodorow (1998) measure is calculated in terms of the length of the shortest path between two concepts using node counting and the maximum depth of the taxonomy.
- Lesk (1986) measure is a function of overlap between corresponding dictionary definitions.
- Wu and Palmer(1994) measure is based on the depth of two given concepts in the WordNet taxonomy and the depth of the Least Common Subsequence (LCS).
- Resnik (1995) measure assesses the information content of the LCS of two concepts. Information content of a concept c, is the probability of encountering it in a large corpus.
- Lin(1998) measure extends Resnik’s measure by considering the Information content of two concepts besides the Information content of the LCS.
- Jiang and Conrath(1997) measure is assessed as the inverse of the Information content of the two concepts and also their LCS.

2.4 Features used in Paraphrase Recognition

Fernando and Stevenson (2008) have used semantic features to measure the similarity between a pair of sentences [5]. The Jiang and Conrath measure was found to be superior to other metrics. The authors have suggested the incorporation of syntactic features to improve performance. Zhang and Patrick (2006) have used a variety of initial syntactic transformations along with lexical features to decide whether the input sentences are paraphrases [16]. Some of the syntactic transformations used were replacement of number entities with generic tags and passive-to-active voice change. The lexical features used were Longest Common Substring and Edit Subsequence, Edit distance and Modified N-gram precision. The results of the experiments show that pure lexical matching could be improved by including even preliminary syntactic transformations. Zhang et al have also suggested the inclusion of Lexical Semantic features to further improve performance.

Brockett and Dolan (2005) have used a combination of Lexical, Syntactic, Semantic and Composite features to perform paraphrase identification using Support Vector Machines [2]. Lexical features such as unigrams, word based edit distance and edit distance calculated after converting the sentences to alphabetized strings have been used. The syntactic morphological variants feature and WordNet mapping features based on Synonymy and Hypernymy have also been used. Edit distance and the co-occurrence of morphological variants features were found to be the most effective features.

Wan, Dras, Dale and Paris (2006) in their work on paraphrase generation have employed a machine learning classifier to identify paraphrases [15]. Both lexical, syntactic feature classes have been used. The Lexical Features include N-gram Overlap features based on Unigrams, Lemmatized Unigrams, BLEU and lemmatized BLEU. Dependency relation overlap feature and dependency tree edit distance are the syntactic measures that have been used. Dependency based features clubbed with bi-gram features were found to exhibit the best performance.

Kozareva and Montoyo (2006) have used purely lexical and semantic features for Paraphrase identification [11]. The lexical features used were the Longest Common Subsequence and Skip-gram overlap. The semantic features include the Jiang and Conrath noun/verb semantic similarity measure, proper name matches and cardinal number coincidences. The lexical features correctly determined the non-paraphrase pairs whereas the semantic features were found to be good at identifying the paraphrases. The authors have suggested that including syntactic information will be beneficial.

Finch, Hwang and Sumita (2005) have extended basic lexical measures such as WER, BLEU and PER to incorporate both semantic and syntactic information [6]. The purely lexical edit distance has been extended by considering the semantic distance between words calculated using the Jiang and Conrath similarity measure. The PER was also calculated for each Part Of Speech(POS) separately. Extending the PER feature based on POS information was found to improve the performance.

Rus, McCarthy, Lintean, McNamara and Graesser (2008) have utilized Lexical, Syntactic and Semantic information for paraphrase identification [14]. The significant aspects of this work are the usage of syntactic information, enhancing the lexical component using Synonymy relations from WordNet and Negation handling using Antonymy relations. The authors have suggested that weighting words with their specificity value will help to improve the performance.

3. TECHNIQUES FOR PARAPHRASE RECOGNITION

This section presents a study of various techniques used in Paraphrase Recognition. Machine learning, graph based approach and matrix similarity method have been utilized by various researchers.

3.1 Machine Learning Techniques

Some of the commonly used machine learning techniques in paraphrase identification are Decision trees, Support Vector machines, Naïve Bayesian method and the K-Nearest Neighbour technique. Zhang and Patrick (2006) have used a decision tree based classifier to identify paraphrases after transforming the input sentences using canonicalization rules [16]. The rules employed were replacement of number entities with generic tags, passive-to-active voice change and replacement of specific future tense usages with more generic ones. Lexical features extracted from the transformed sentences were fed to the decision tree classifier. The authors have experimented on the Microsoft Research Paraphrase Corpus (MSRPC) and have reported a maximum accuracy of 71.9%.

Brockett and Dolan (2005) have employed Support Vector Machines for Paraphrase Identification and Corpus Construction [2]. The authors have reported precision and recall values of 86.76% and 86.39% respectively. Finch et al (2005) have also employed a Support Vector Machine Classifier with radial basis function kernels for identifying paraphrases based on machine learning evaluation features and have reported an accuracy level of 74.96% on the MSRPC [6].

Wan et al (2006) have employed various machine learning classification techniques such as Naïve Bayesian Learner, Decision tree based classifier, SVM and K-Nearest Neighbour technique to rule out inconsistent paraphrases [15]. The best performance was exhibited by Support Vector Machines on the MSRPC. The maximum observed accuracy was 75% when a combination of several lexical and syntactic features was used.

Kozareva and Montoyo (2006) have studied the behaviour of three machine learning classifiers for identifying paraphrases, namely SVMs, k-Nearest Neighbour technique and Maximum Entropy method [11]. The SVM technique was found to perform better than the other techniques. But the best performance has been exhibited by a voting system which involved the three machine learning classifiers. The system of [11] has registered the highest accuracy level of 76.64% on the MSRPC.

3.2 Graph Based Approach

In [14] paraphrases have been recognized using a graph subsumption approach. The input sentences are mapped to graph structures and subsumption is detected by evaluating graph isomorphism. Text A is entailed from B if and only if B subsumes A. The entailment score for Text A with respect to Text B and B with respect to A have been averaged to determine whether A and B are paraphrases. In tests carried out on the MSRPC an accuracy value of 70.61% has been observed.

3.3 Matrix Similarity Method

Fernando and Stevenson (2008) have utilized a matrix similarity method for paraphrase detection [5]. In this work the semantic similarity values between all pairs of words have been computed using the knowledge based measures [13] and an accuracy of 74.1% has been reported.

4. NEURAL NETWORK BASED PARAPHRASE RECOGNITION

Neural networks are computational models inspired by the human nervous system and are one of the foremost machine learning techniques. Neural architectures are suitable candidates for language processing tasks because of their robustness to noisy input and their similarity to cognitive thought processes. The task of Paraphrase Recognition can be viewed as a binary classification problem. Given a pair of sentences the Recognizer must provide a response as to whether the sentences can be considered as paraphrases or not. Automatic Paraphrase Recognition can be tackled using Machine Learning algorithms. Though Neural Networks have been employed for Paraphrase Generation [Miikkulainen 1989], there are no known systems employing neural networks for the task of Paraphrase Recognition

This paper presents the work on Paraphrase Recognition using a Neural Network Classifier. The architecture of the proposed system as shown in Figure 1 consists of a Feature Extractor module which identifies the various features from the sentence pairs present in the training corpus. A feature vector is constructed and passed on to the Neural Network Classifier. Once the network is trained its performance can be evaluated on test data using standard measures such as precision, recall and F-measure.

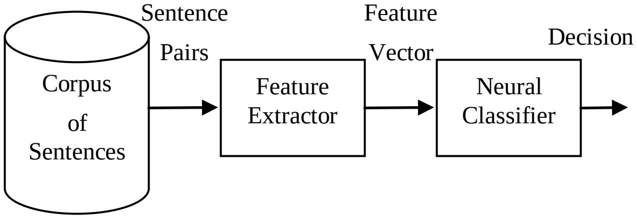

Figure 1 Paraphrase Recognition using a Neural Classifier

4.1 Experimental Data

Experiments have been carried out using the Microsoft Research Paraphrase Corpora (MSRPC). This corpora was constructed from a collection of Internet news articles and is partitioned into the training set and test set. The sentence pairs in the corpora were labelled as Positive and Negative cases of Paraphrasing by Multiple human annotators. Out of the total collection, 67% of paraphrases were found to be present. The training set consists of 4076 sentence pairs and the test set has 1726 sentence pairs. Of these the number of paraphrases in the training set and test set are 2753 and 1147 respectively [6].

4.2 Feature Extraction

The Feature Extraction module is responsible for extracting various features from the input pair of sentences. Recognition of semantic equivalence has been found to require processing at the lexical level, syntactic level and the sentence semantic level. In this work a combination of purely lexical, syntactic, lexical-semantic and lexical-syntactic features have been used for paraphrase recognition as described below.

4.2.1 String Edit distance extended to permit lexical Variations

The Levenshtein distance also known as Edit distance or Word Error Rate (WER) (Su et al. 1992) is a purely lexical measure that computes the number of insertions, deletions and substitutions required to transform one string into another. With respect to processing input sentences S_1, S_2 if the words in the positions i of S_1 and j of S_2 are the same the cost is 0 else it is 1. Usually a dynamic programming approach is used to compute the edit distance. A disadvantage of the Edit distance measure is that sentences with high lexical alternations or different syntactic structures have a high edit distance and are hence not considered to be paraphrases even though they may actually be paraphrases.

In an attempt to overcome this, modified edit distance has been used in [6], which instead of looking for exact matches between a pair of words uses semantic similarity measures to decide whether the words are similar. For determining whether a pair of words was semantically similar the Jiang and Conrath measure [8] was used in [6].

$$\mathrm{Dist}(\mathrm{word}_i, \mathrm{word}_j) = \mathrm{IC}(\mathrm{word}_i) + \mathrm{IC}(\mathrm{word}_i) - 2 * \mathrm{IC}(\mathrm{LSuper}(\mathrm{word}_i,\mathrm{word}_j))$$

Here $\mathrm{IC}(\mathrm{word}_1) = -\log P(\mathrm{word}_1)$ where $P(\mathrm{word}_1)$ is the probability of occurrence of word_1 in the corpus and $\mathrm{LSuper}(\mathrm{word}_i, \mathrm{word}_j)$ denotes the lowest super-ordinate of both the words in the WordNet taxonomy. This measure has been shown to exhibit superior performance in similarity assessment [5]. Hence in this work the modified string edit distance computed using the Jiang and Conrath measure which combines both lexical and semantic aspects has been used.

4.2.2 Skip-grams

An n-gram is a sub-sequence of n-items from a given sequence. Similar sentences are expected to have a greater percentage of shared n-grams[7]. But simple n-grams tend to overlook non-contiguous word associations [3]. Skip grams detect non-contiguous word associations along with the contiguous word associations identified by n-grams. Skip-grams are usually formed using a skip distance k and allow a total of k or less skips. It is a purely lexical measure. A commonly used value for k is 4. To assess the degree of similarity between two sentences the number of common skip-grams between them has been used.

4.2.3 BLEU

The BiLingual Evaluation Understudy (BLEU) metric was proposed by Papineni et. al as a method for automatic evaluation of machine translation. It is based on the concept of a weighted average of similar length phrase matches (n-grams). The BLEU metric has been adapted for assessing similarity between sentences by Cordeiro and Dias [4]. The metric is given by:

$$BLEU_{adapted} = \exp \frac{1}{N} \sum_{n=1}^{N} \log C_n$$

$$\text{where } C_n = \sum_{ngram} \frac{count_{match}(ngram)}{count\,(ngram)}$$

Here count(ngram) gives the maximum number of n-grams in the shorter sentence and N is the maximum n-gram size taking values between 1 to 4. The brevity penalty originally used in BLEU metric to penalize shorter outputs can be omitted here as suggested in [6]. The adapted BLEU metric which is purely lexical in nature has been used as one of the features here.

4.2.4 Dependency tree Edit distance

A dependency tree is a syntactic representation of a sentence. The edit distance between dependency trees has been used to detect text entailment by Kouylekov and Maginini [10]. The dependency trees for input sentences have been constructed using the Stanford Parser. The dependency tree edit distance has been calculated using the approach proposed by Zhang and Shasha (1989). The number of insertions, deletions and substitutions required for transforming one dependency tree to another was calculated by assigning equal costs for insertions, deletions and substitutions. The computed cost was normalized as in [10] to form one of the purely syntactic features for Paraphrase recognition.

4.2.5 Parts of Speech enhanced Position Error Rate measure

Position Error Rate(PER) is similar to Levenshtein distance except that the positions of the words in the sentences are ignored. It has been determined that edits involving certain classes of words cost more than other classes and that enhancing the PER measure with Parts Of Speech(POS) information yields good results for detecting semantic similarity in [6]. Given the input texts T_1, T_2 the PER is calculated as:

$$PER(T1, T2) = \max[diff(T_1,T_2), diff(T_2,T_1)] / |T_2|$$

where $diff(T_1,T_2)$ is the number of words observed only in T_1. The total edit distance can be split into components corresponding to each POS tag which reflects the contribution of words belonging to the respective POS tag to the overall edit distance. Two sets of features are used, one for matches and the other for non-matches.

4.2.6 Negations

Negations maybe present explicitly or implicitly through the usage of words such as ‘not’, ‘no’, or through the usage of antonyms respectively [12]. Explicit negations are handled using a binary negation attribute as in [12]. Implicit negations are more challenging. The presence of a word in T_1 and its antonym in T_2 or vice-versa indicates a negation. In our work the number of negations detected has been used as one of the features. But when there is more than one such situation interpretation becomes difficult. In the same way the presence of both explicit and implicit negations together also leads to complications. These issues are proposed be handled in the future.

4.3 Neural Network Classifier

Various machine learning techniques such as Memory based Classifiers [12], Support Vector Machines [2, 11 and 15], Decision Trees [16] and k-Nearest Neighbour [11, 15] techniques have been utilized in paraphrase identification and text entailment detection. Some of the most successful systems for Paraphrase Recognition [6, 11] have employed Support Vector Machines. Support Vector Machines are found to have similar operational characteristics as Neural Networks [9]. Though Neural Networks are inferior to Support Vector machines in aspects such as intolerance to irrelevant attributes and over-fitting they score better in the ability to perform incremental learning.

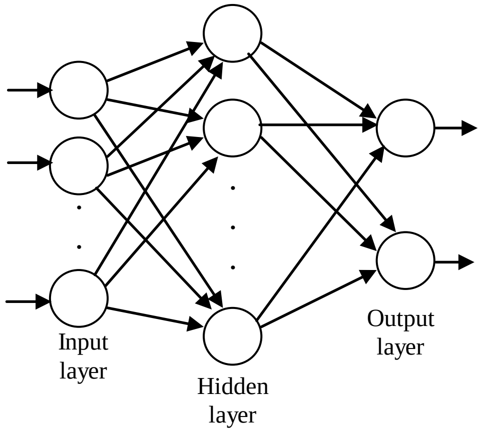

Figure 2 Architecture of a Back Propagation network

Some of the most successful neural classifiers are multi-layer feed forward networks such as the Back Propagation Network [9]. Hence a Back Propagation Network has been employed for Paraphrase Recognition.

4.3.1 Architecture

A Back Propagation Network with three layers (Figure 2) has been utilized. The input layer has n number of neurons, where n is the number of extracted features. The neurons in the input layer are fully inter-connected with those in the hidden layer. A logistic activation function has been employed. Since Paraphrase recognition is a two class problem the output layer contains two neurons which perform a weighted sum of all hidden unit outputs.

4.3.2 Training

The network has been trained by presenting the sentence pairs from the training set of the MSRPC. The stages in training the neural network are:

1. Initialization – The weights on the connections between the input-hidden layers and hidden-output layers are initialized. The learning rate and momentum values are also initialized.
2. Presentation of training samples – The extracted feature vectors along with their class information is fed to the input layer for each pair of sentences.
3. Forward computation – The input is propagated to the hidden layer via weighted connections. Each hidden layer neuron calculates the weighted sum of its inputs. The activation of the hidden layer neurons are passed on to the output layer where a similar computation is performed to determine the actual output. The difference between the target and actual output gives the error.
4. Backward computation – The local gradient for each output layer neuron is computed in terms of its error and activation function value. This value is then propagated backwards to determine the gradient of the hidden layer neurons. Using the gradient values the weights are updated.

5. Iteration – The set of training samples are presented repeatedly in a random order and stages 3 and 4 are repeated until the error value becomes minimal.

The learning rate and momentum values are gradually decreased as iterations progress in order to achieve stable learning.

5. DISCUSSION

The Neural network based Paraphrase recognizer is currently under implementation. The performance of the system can be assessed by measuring the accuracy and precision and comparing it with the Paraphrase Recognizers of [6] and [11] which have reported an accuracy value of 75% and above. The system can be used in Question Answering systems and for plagiarism detection in document collections. Sentences present in the abstracts of document collections can be checked for the presence of paraphrases; these can a serve as an indicator for detecting plagiarised content.

6. REFERENCES

[1] Anupriya, R. and Chitra, A. 2009. “Analysis of Paraphrase Recognition Techniques” CSI Communications, Vol.33, Issue 9, 12-14.

[2] Brockett, C. and Dolan, B. 2005. Support Vector Machines for Paraphrase Identification and Corpus Construction. In Proceedings of the 3rd International Workshop on Paraphrasing, 1-8.

[3] Cheng, W., Greaves, C. and Warren, M. 2006. “From n-gram to Skip-gram to Conc-gram” International Journal of Corpus Linguistics, Vol. 11, Issue 4, 411-433.

[4] Cordeiro, J., Dias, G. and Brazdil, P. 2007. “New Functions for Unsupervised Asymmetrical Paraphrase Detection,” Journal of Software, vol. 2, Issue 4, 12-23.

[5] Fernando, S. and Stevenson, M. 2008. A Semantic Similarity Approach to Paraphrase Detection. In Proceedings of the Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium.

[6] Finch, A., Hwang, Y. and Sumita, E. 2005. Using Machine Translation Evaluation Techniques to Determine Sentence level Semantic Equivalence. In Proceedings of the Third International Workshop on Paraphrasing, 17-24.

[7] Guthrie, D., Allison, B., Liu, W., Guthrie, L. and Wilks, Y. 2005. A Closer look at Skip-gram modeling. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, 1222-1225.

[8] Jiang, J. and Conrath, D.W. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In Proceedings of the International Conference on Research in Computational Linguistics.

[9] Kotsiantis, S. 2007. “Supervised Machine Learning: A Review of Classification Techniques,” Informatica Journal, vol. 31, 249-268.

[10] Kouylekov, M. and Magnini, B. 2005. Recognizing Textual Entailment with Tree Edit Distance Algorithms. In Proceedings of the First PASCAL Challenges Workshop on Recognising Textual Entailment, 17–20.

[11] Kozareva, Z. and Montoyo, A. 2006. Paraphrase Identification on the basis of Supervised Machine Learning Techniques. In Proceedings of Advances in Natural Language Processing: 5th International Conference on NLP, 524-533.

[12] Kozareva, Z. and Montoyo, A. 2006. The Role and Resolution of Textual Entailment in Natural Language Processing Applications. In Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems, 186-196.

[13] Mihalcea, R., Corley, C. and Strapparava, C. 2006. Corpus based and Knowledge-based Measures of Text Semantic Similarity. In Proceedings of the 21st Conference of American Association for Artificial Intelligence, 775-780.

[14] Rus, V., McCarthy, P. M., Lintean, C., McNamara, D.S. and Graesser, A.C. 2008. Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, 201-206.

[15] Wan, S., Dras, M., Dale, R. and Paris, C. 2006. Using Dependency-based Features to take the "Para-farce" out of Paraphrase. In Proceedings of the Australasian Language Technology Workshop, 131-138.

[16] Zhang, Y. and Patrick, J. 2005. Paraphrase Identification by Text Canonicalization. In Proceedings of the Australasian Language Technology Workshop, 160–166.