CHAPTER 2

PARAPHRASE RECOGNITION USING SUPERVISED LEARNING

The objective of Paraphrase Recognition is to detect whether the input pair of entities form paraphrases or not. Due to the wide variability and context dependence of natural language text, efficient techniques are required for Paraphrase Recognition. PR systems constructed using supervised learning techniques such as Support Vector Machines, Naïve Bayesian technique and Decision tree classification have been found to exhibit good performance (Androutsopoulos & Malakasiotis 2010). This has motivated the design of Paraphrase Recognition systems using supervised learning. Such systems work by extracting features that quantify the extent of similarity from the input text units.

In this chapter, two different approaches for Paraphrase Recognition have been proposed. The first approach employs a Supervised Neural Network classification scheme, namely Radial Basis Function Neural Networks (RBFNN). Input sentence pairs have been classified by extracting Lexical, syntactic and semantic features from the sentences. Different schemes for designing the RBFNN network including Orthogonal Least Squares learning method, k-means and Fuzzy C-means clustering have been investigated. In the second approach, Support Vector Machine (SVM) classifiers have been employed by using the same set of features and various kernel functions such as linear, polynomial and radial-basis. Further the

performance of existing SVM based PR systems has been improved using wrapper based Feature Selection approach. Since the performance of the classifier depends on the features used, Genetic Algorithm strategy has been used to determine the ideal set of features by using the accuracy of the classifier as the objective function.

2.1 RBFNN BASED PARAPHRASE RECOGNIZER

The task of Paraphrase Recognition can be viewed as a binary classification problem. The choice of the classifier as well as the feature set used influences the efficiency of Paraphrase Recognition. Machine learning Classifiers have achieved considerable success in Paraphrase Recognition with SVM Recognizers being the most popular. Though Neural Network based learning is very popular in several applications, it has not been fully exploited in the domain of NLP. In areas related to Paraphrasing, very few systems have employed Neural Networks. Miikkulainen & Dyer (1989) have used modular neural networks for Paraphrase Generation and Socher et al (2011) have combined Recursive Autoencoders with dynamic pooling technique and a soft-max classifier for Paraphrase Recognition. The proposed design of a Paraphrase Recognition system using Radial Basis Function Neural Network has been presented.

2.1.1 Features used for Paraphrase Recognition

One of the crucial aspects of Paraphrase Recognition is the selection of suitable features. Lexical, syntactic and semantic features have been previously used both individually and in various combinations for detecting paraphrases as detailed in Section 1.5.1.2. This section describes the features used in the RBFNN recognizer.

2.1.1.1 Lexical Features

Lexical features are the simplest category of features and can be computed directly from the input text. These features have been used to assess the degree of word overlap between the candidate sentences. Various lexical features such as Unigrams, WER, PER, BLEU, Skip grams, Longest Common Subsequence (LoCoS) etc. have been used for the purpose of Paraphrase Recognition. Some of the most commonly used features which have yielded good performance such as BLEU, Skipgrams and LoCoS have been employed in this work.

BLEU - Bi-Lingual Evaluation Understudy metric (Papineni et al 2002) was originally used to evaluate the performance of Machine Translation systems by determining the word overlap between the reference and machine translations. The metric has been adapted for assessing similarity between sentences by Cordeiro et al (2007). The modified metric is based on the geometric mean of n-gram matches and is given in Equation (2.1).

$$\mathrm{BLEU}_{\mathrm{adapted}} = \exp \frac{1}{N} \sum_{n=1}^{N} \log C_n \tag{2.1}$$

$$C_n = \frac{\mathrm{count}_{\mathrm{match}}(\mathrm{ngram})}{\mathrm{count}(\mathrm{ngram})} \tag{2.2}$$

In Equation (2.2), count (ngram) gives the maximum number of n-grams in the reference sentence and $\mathrm{count}_{\mathrm{match}}$(ngram) is the number of common ngrams. N is the n-gram size in the range 1 to 4. When N=1, the metric is termed as BLEU1 and the modified BLEU1 metrics become equivalent to unigram metrics. BLEU precision and recall have been calculated by alternatively using both sentences as reference sentences similar to the approach used by Wan et al (2006). In this work, initially BLEU1 to BLEU4 were calculated and based on the results of experiments on the MSRPC, BLEU1 precision and recall metrics have been used. For the input pair:

"Spain beats Holland to lift Soccer World Cup", "Spain beats Holland in FIFA 2010 Final", the value of $count_{match}$(ngram) is 3 as the common unigrams are "Spain", "beats" and "Holland". The denominator count (ngram) takes the values eight and seven corresponding to the lengths of the input sentences.

Skipgram metrics – Skipgrams of an input sentence are formed by considering both contiguous and non-contiguous n-grams (Guthrie et al 2005). A Skip distance parameter is used to generate n-grams. A commonly used value for the skip distance 'k' is 4, which permits skips of length 0 to 4. Contiguous or simple n-grams formed from adjacent words result when the number of skips is equal to 0. In this work Skipgrams with n=1, 2, 3 have been formed using k=4. Two features, Skipgram precision and recall have been computed by dividing the number of common skipgrams by total possible number of Skipgrams constructed from first and second input sentences respectively.

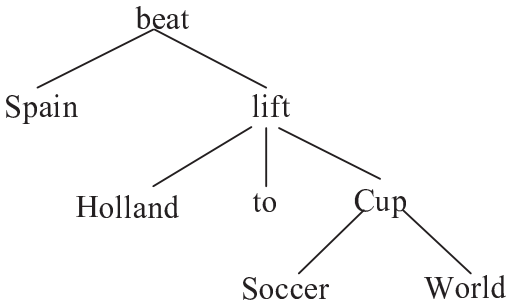
When n=2 and k=0, the skipgrams generated for the sentence "Spain beats Holland to lift Soccer World Cup" include: "Spain beats", "beats Holland", "Holland to", "to lift", "lift Soccer", "Soccer World", "World Cup". Setting k=2 generates "Spain to", "beats lift", "Holland Soccer", "to World" and "lift Cup". The common skipgrams for the given input sentences are: "Spain", "beats", "Holland", "Spain beats", "beats Holland", "Spain Holland" and "Spain beats Holland".

Longest Common Subsequence (LoCoS) metric – For a pair of sentences, the LoCoS feature has been obtained by dividing the length of the longest common in-sequence portion by the length of the shorter sentence. Since the in-sequence portion is considered, the feature may also be considered as 'Longest Common Substring'. For the given input pair the longest common subsequence is "Spain beats Holland".

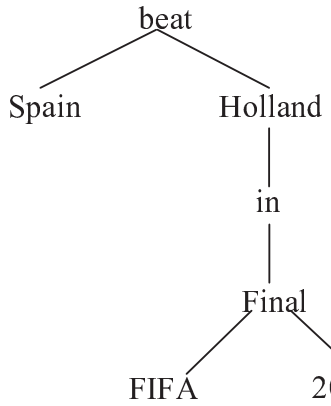
2.1.1.2 Syntactic Features

Syntactic Features reflect the grammatical structure of the text. These features have been used to assess the degree of structural similarity by parsing the input sentences. In addition to two purely syntactic measures, the Dependency tree edit distance and triple similarity function, this work has also used the Parts Of Speech based PER measure (Finch et al 2005).

Dependency Tree Edit Distance - A dependency tree is a syntactic representation of a sentence. Tree Edit Distance (TED) has been defined as the minimum cost incurred in transforming one tree into another, using elementary operations such as insertion, deletion and substitution (Bille 2005). In this work, Stanford Parser (Klein & Manning 2003) was used to construct dependency trees for the input sentences. The trees constructed for the example sentences are shown in Figure 2.1. The dependency tree edit distance was calculated using the approach proposed by Zhang & Shasha (1989) after assigning equal costs for insertions, deletions and substitutions. Two tree edit distance features were calculated by normalizing tree edit distance using the number of nodes in the dependency trees of first and second sentence.



Sentence 1: Spain beats Holland to lift Soccer World Cup



Sentence 2: Spain beats Holland in FIFA 2010 Final

Figure 2.1 Dependency trees for example sentences

Triple Similarity Function (TSF) – For every pair of words connected by a link in the dependency tree, a corresponding dependency relation can be formed. Each dependency relation is in the form of a triple, consisting of head, modifier or dependent and relationship between them. In this work, as proposed by Wan et al (2006) the number of shared triples between the sentences was divided by the number of triples in the first and second sentence to obtain a pair of triple similarity measures. The dependency relations or triples produced for the example sentences are shown in Figure 2.2. The only shared triple for the example case is 'beat Spain nsubj'.

Sentence 1:	Spain beats Holland to lift Soccer World Cup
Triples:	nsubj(beats, Spain), nsubj(lift, Holland), aux(lift, to), xcomp(beats, lift), nn(Cup, Soccer), nn(Cup, World), dobj(lift, Cup)
Sentence 2:	Spain beats Holland in FIFA 2010 Final
Triples:	nsubj(beats, Spain), dobj(beats, Holland), nn(Final, FIFA) nn(Final, 2010), prep_in(Holland, Final)

Figure 2.2 Dependency relations for example sentences

Parts Of Speech based PER (POSPER) - Position-independent word error rate (PER) (Tillmann et al 1997) assesses the number of edit operations required to transform one sentence into another without taking the word order into account. Non-matching words are treated as substitutions, additional words as insertions and missing words as deletions. According to Finch et al (2005), the PER 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.

Given texts T1 and T2, w^- is the collection of words from T1 which do not match those in T2 and w^+ is the set of matching words. For non-

matches the contribution of each class of POS tags t, is given in Equation (2.3).

$$f_t^- = \frac{\sum_{w \in w^-} count_t^-(w)}{|s_i|} \tag{2.3}$$

where $count_t^-(w)$ is the number of times word w occurs in w^- with tag t and $|s_i|$ is the length of the shorter sentence. Likewise for matches, Equation (2.4) gives the contribution of each class of POS tags.

$$f_t^+ = \frac{\sum_{w \in w^+} count_t^+(w)}{|s_i|} \tag{2.4}$$

A total of 48 POS tags were generated by the TreeTagger (Schmid 1994) which was used to tag the input sentences. Two sets of 48 features were used, one for matches and the other for non-matches. This set of features termed as POSPER, combine lexical and syntactic information. The w^+ set for the input sentences consists of the words Spain(NP), beat(VVZ) and Holland(NP). The feature f_{NP}^+ is therefore 2 / 7 as there are two words with the "NP" tag and the length of the shorter sentence is seven.

2.1.1.3 Semantic Features

Semantic Similarity features focus on the semantic meaning and are usually based on resources such as WordNet (Fellbaum 1998). As proposed by Kozareva & Montoyo (2006a), the Noun/Verb Similarity, degree of Cardinal number and Proper Noun matches as well as presence of antonyms have been used as Semantic features in this work.

Noun / Verb Similarity – The degree of similarity between the nouns and verbs of the input sentences is indicative of the overall sentence similarity. In order to assess the similarity between a pair of nouns / verbs the WordNet based Jiang and Conrath measure (Jiang & Conrath 1997) was used as it has been shown to exhibit superior performance in similarity assessment (Fernando & Stevenson 2008). The Jiang and Conrath measure gives the distance between two words in terms of the Information Content of the words and their Lowest Common Subsumer in the Wordnet hierarchy. A portion of the Wordnet hierarchy extracted from the LabelMe Online tool (Russell et al 2008) developed at MIT is shown in Figure 2.3. The lowest common ancestor of the words 'Computer' and 'monitor' is 'device'. In this work, the cumulative similarity between nouns was normalized using the number of noun pairs across the two sentences. The same procedure was used for verbs and can be extended for adverbs and adjectives as well. For example, the distance between verbs 'lift' and 'beat' from the considered sentence pair can be used to assess the similarity measure.

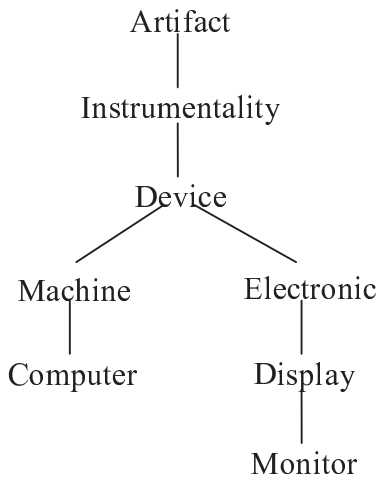


Figure 2.3 Portion of Wordnet hierarchy

Cardinal Number and Proper Noun match –Cardinal number matches have been detected by transforming phrases involving cardinal numbers into equivalent numerical values and standard relationships (Kozareva & Montoyo 2006a) such as 'less than', 'greater than', 'equal to' and 'not equal to'. For instance, the phrases such as 'greater', 'more' and 'higher' are all mapped to the '>' symbol. Using this procedure, phrases such as 'not more than twenty' and 'greater than 15' were identified as matches. The matching proper nouns in the two sentences have also been identified. The number of cardinal number and proper noun matches have been normalized using the length of the shorter sentence. The matching Proper Nouns in the given example are 'Spain', 'Holland' and there are no Cardinal number matches.

Negations – The technique used to handle negations is motivated by the work of Kozareva & Montoyo (2006b) and Herrera et al (2006). Implicit negations characterized by the presence of a word in T1 and its antonym in T2 or vice-versa can be detected using the Wordnet Antonymy relationship. The number of such antonym pairs is normalized by the length of both sentences to generate two antonym features. In order to handle explicit negations such as 'not', 'never' and 'no', the corresponding dependency relations were analyzed and the associated head word was treated as being in negative form. For the sentence "He was not sad", the dependency relations are: nsubj(sad, He), cop(sad, was), neg(sad, not). The word 'sad' which is the headword for 'not' is considered as being negated. Once such a negative term was detected the other sentence in the pair was checked for the presence of the original word viz. 'sad' or one of its synonyms. In case, such a term was present a boolean feature was toggled to indicate that the sentence pairs have opposite terms. The three categories of features extracted from the sentence pairs have been passed as input to two different classification schemes, RBFNN and SVM to detect the paraphrases.

2.1.2 Design of RBFNN based PR System

Artificial Neural Networks (ANNs) consist of interconnected computational units which try to simulate the working of the neurons present in the human brain. Multi-Layered Perceptron (MLP) networks have been traditionally used in classification problems especially involving linearly inseparable data. MLP networks are trained using the Back Propagation Algorithm which is an iterative method that consumes a large amount of time. Radial Basis Function Neural Networks are considered as an alternative to the traditional MLP network as they can be trained quickly when compared to the iterative training method of MLP networks and require lesser training data. RBFNNs are feed forward networks with a single hidden layer made up of radial units and a linear output layer (Haykin 2003). These networks are used extensively in time-series prediction and classification (Orr 1996).

Paraphrasing involves a wide range of possibilities in terms of lexical and syntactic variations. Two sentences which are lexically and syntactically very different may in fact be semantically equivalent. On the other hand sentences which have the same structure or even the same set of words may not be paraphrases. Thus Paraphrase Recognition becomes a linearly inseparable problem and hence requires the usage of efficient classification techniques. Since RBFNNs transform linearly inseparable data to a separable form by mapping data to a higher dimensional space, they have been investigated for the Paraphrase Recognition problem.

RBFNNs have a three layered architecture as shown in Figure 2.4. Given a set of m, n-dimensional training vectors the input layer neurons pass on the input vector X to the hidden layer. The number of input units depends on the dimensionality of the input vectors. The size of the hidden layer is variable and is determined during the training of the network.

The activation function of the units in the hidden layer is a radial basis function such as a Gaussian function. The hidden layer neurons are responsible for performing a non-linear mapping of the input to a higher dimensional space. The size of the output layer is determined by the number of outputs in the training vector. The network is fully connected, with the input to hidden layer links being non-weighted whereas the hidden to output layer connections are assigned weights.

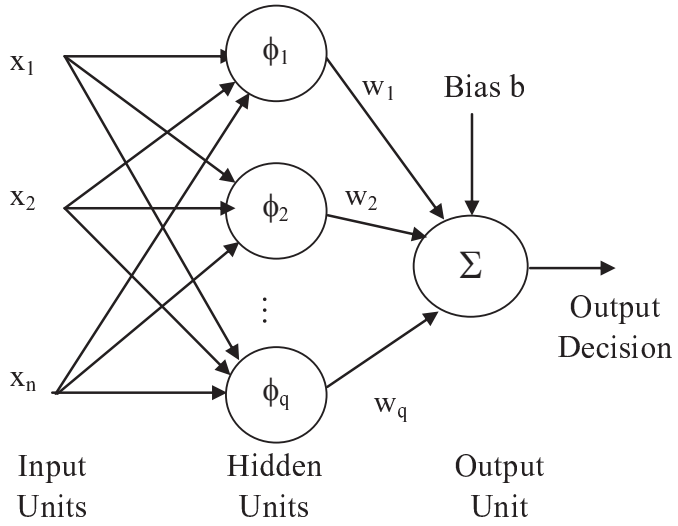


Figure 2.4 Architecture of the RBF Neural Network

The process of training the RBFNN involves the determination of the number of hidden layer neurons, their centers as well as the weights from the hidden to the output layer. The training is carried out in two stages. In the first, an unsupervised strategy is used to determine the activations of the hidden layer. The second stage uses a supervised learning approach to determine the weights between the hidden layer and output layer. The number of hidden layer neurons is fixed using any one of the following strategies (Jayawardena et al 1997):

- In the simplest case the number of hidden layer neurons is equal to the number of inputs. This strategy is unsuitable for large-scale input.

- The other alternate is to first cluster the input data using k-means algorithm and assign one hidden layer neuron corresponding to each cluster's centroid. This approach requires repeated computations using all the inputs to determine the number of clusters k.

- The third uses an Orthogonal Least Squares Learning algorithm (Chen et al 1991) where the hidden layer neurons are added iteratively. In each iteration, a hidden layer neuron is added with center corresponding to the input which produces a maximal reduction in the Sum of Squared errors. The process is repeated until the error of the network meets the specified error goal or the number of hidden layer neurons reaches the maximum limit specified.

During the unsupervised learning stage, the activation of a hidden layer neuron is computed in terms of distance between the input and prototype vector associated with the neuron. The hidden layer applies a non-linear transformation from input space to a high dimensional space by using Kernel functions. Each radial unit of the hidden layer computes its activation in terms of the Euclidean norm between the hidden layer neuron's center μ and the input X by using the Gaussian function of Equation (2.5). σ is the spread factor or smoothing parameter which is equivalent to the radius of the neuron and controls its response. When the value of the spread factor is less, the neuron is very selective and responds only to inputs very close to its center.

$$\phi(x) = \exp(-\frac{\| X - \mu \|^2}{2\sigma^2}) \quad (2.5)$$

Once the number of hidden layer neurons is determined, the weights from the hidden layer neurons to output neuron are adjusted using a supervised learning paradigm. The output neuron has a linear activation function and computes f(x) as in Equation (2.6) where w_i represents the weight from the i^{th} hidden layer neuron to the output layer neuron and q is the number of hidden layer neurons. The output f(x) is compared with the target output and the weights are adjusted using the least mean square rule.

$$f(x) = \sum_{i=1}^{q} w_i \phi_i(X) \quad (2.6)$$

In this work, the network has been designed with input layer size equal to number of input features and a single output layer neuron corresponding to the binary decision of whether the input sentences are paraphrases or not. The number of neurons in the hidden layer was fixed using three different schemes:

- k-means Clustering : In this approach the input vectors were clustered using the partitioning based k-means approach. Various cluster patterns were produced by altering the value of k – the number of clusters. One hidden layer neuron was associated with the center of each cluster and its activation was calculated using Equation (2.5). The weight vector - w between the hidden layer and output layer was determined using the least squares method as per Equation (2.7) where t is the target output and ϕ is the activation of the hidden layer. The weight vector and the centroids of the clusters can be stored for predicting the class labels of the test data.

$$w = (\phi^T \phi)^{-1} \phi^T t \quad (2.7)$$

- Fuzzy C-Means Clustering: the only difference when compared to the previous scheme was that a Fuzzy Clustering approach was used to produce a soft partitioning of the data. Both 'C' the number of clusters as well as the exponent of the partitioning matrix were changed to produce different cluster patterns.

- Orthogonal Least Squares Learning algorithm: This approach starts with an empty hidden layer and progressively adds neurons. The input which produces the maximum error was identified and a corresponding neuron was added by setting its representative as the input. For each input vector, the activation of the hidden layer was calculated using Equation (2.5). A neuron produces a maximum response when its representative matches the input. The process was repeated until the desired error level was reached or the number of neurons reached the pre-specified limit. The maximum number of neurons in the hidden layer and the spread parameter were varied to create different network architectures. After training, the network was used to predict class labels for the test data set.

The performance evaluation of the PR system using RBFNN has been presented in Section 2.4.1. The RBFNN Recognizer has registered a better accuracy and precision then the SVM Paraphrase Recognizer. In terms of recall, the vice-versa was observed and the F-measure of both systems was found to be comparable. The marginal overall improvement of the RBFNN recognizer was obtained at the expense of increased training time. The

performance of both systems was found to be dependent on the underlying features.

2.2 PARAPHRASE RECOGNITION USING SVM

Support Vector machines are one of the popular and efficient methods used for Pattern recognition tasks (Kotsiantis 2007). SVM technique has also been widely used in Paraphrase Recognition (Androutsopoulous & Malakasiotis 2010). Therefore, with a view to benchmark the performance of RBFNN oriented PR systems and to establish the effectiveness of the features described in Section 2.1.1 for the PR task, an SVM based system has also been designed.

A Support Vector Machine (SVM) is a supervised learning algorithm originally developed by Vapnik in 1995 to solve classification problems (Shawe-Taylor & Cristianini 2000). It performs classification by constructing a hyper-plane that divides the data. Given a set of m, n-dimensional training vectors D = $\{(x_1, y_1)\ldots(x_m, y_m)\}$ where $x \in Rn$ and $y = \{1,-1\}$, SVMs construct a maximum margin hyperplane $f(x) = w.x + b$ where b is the bias and $<w.x>$ is the inner product between weight vector w and input vector x (Zhang & Lee 2003).

SVMs handle data that cannot be fully linearly separated by using a soft margin as shown in Figure 2.5. In such cases a slack variable ξ_i, is used which measures the degree of misclassification along with a user specified cost parameter C, which is used to control the tradeoff between permitting training errors and forcing rigid margins. Such classifiers are termed as C-Classifiers.

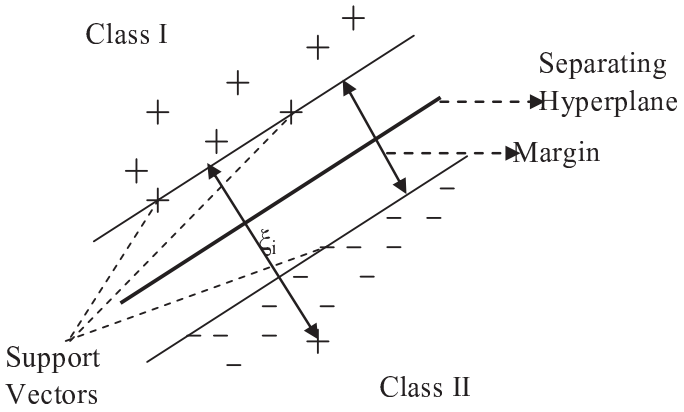


Figure 2.5 Support Vector Machine classification

In order to limit the number of support vectors, another variant of SVM Classification termed as nu-Classifier has been used in this work. The nu-Classifier uses the parameter ρ to control the width of the margin and ν acts as a lower bound on the number of support vectors and upper bound on the misclassifications. The objective of nu-Classifier is to minimize the classification error given by Equation (2.8) subject to the condition $y_i(w.x_i + b) \geq \rho - \xi_i$ where ξ_i is the slack variable, w is the weight vector, b is the bias, x_i is the input with output y_i (Chen et al 2005).

$$\frac{1}{2} \| w \|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \tag{2.8}$$

when the given data is not linearly separable, SVM attempts to achieve a better representation of the data by mapping the input data x to a higher dimensional space $\phi(x)$ termed as Feature space. Given l support vectors denoted as x_i and their corresponding Lagrangian multipliers α_i, the decision function of the hyperplane is given in Equation (2.9) where $K(x_i, x) = \phi(x_i)$. $\phi(x)$ is termed as the Kernel function.

$$f(x) = \sum_{i=1}^{l} \alpha_i \phi(x_i).\phi(x) + b \tag{2.9}$$

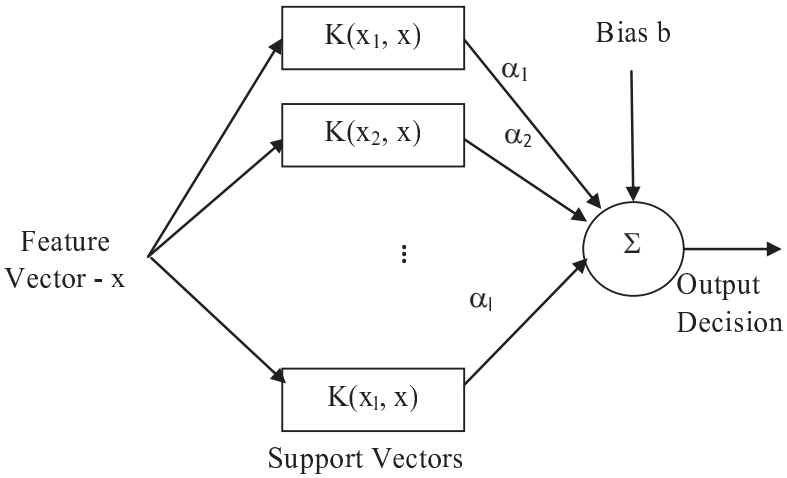


Figure 2.6 Architecture of an SVM Classifier

The architecture of an SVM Classifier is shown in Figure 2.6 where x is the input feature vector and x_i, i=1, 2 …l are the support vectors. The input x is compared with each of the support vectors by using the kernel function K. The output of each comparison is then consolidated using Equation (2.9) to yield the final decision.

Different kernel functions such as Linear, Radial Basis, Polynomial and Sigmoid exist. In this work experiments have been conducted using all the four standard kernels. The simplest kernel function is the linear kernel given in Equation (2.10). The linear kernel works well for noise-free data and when the dimensionality of the dataset is high in comparison to its numerosity (McCue 2009).

$$K(x_i, x) = x_i^T x \tag{2.10}$$

The polynomial and sigmoid kernels are also commonly used for non-linear data. The polynomial kernel is given in Equation (2.11) and involves multiple parameters with γ which is the inverse of the number of dimensions; d and r being the polynomial degree and degree coefficient respectively.

$$K(x_i, x) = (\gamma x_i^T x + r)^d \tag{2.11}$$

$$K(x_i, x) = \tanh(\gamma x_i^T x + r) \tag{2.12}$$

The sigmoid kernel given in Equation (2.12) is also known as hyperbolic tangent kernel and renders the SVM equivalent to a two-layer neural network. The Radial Basis function kernel is given in Equation (2.13) where σ is the width of the kernel. This kernel is popular as it can handle non-linear relationships better and the number of parameters involved is less when compared to sigmoid and polynomial kernels (Gunn 1998).

$$K(x_i, x) = \exp(-\frac{1}{2\sigma^2} \| x_i - x \|^2) \tag{2.13}$$

An alternative formulation is given in Equation (2.14) where γ is set as 1 / number of features or dimensions.

$$K(x_i, x) = \exp(-\gamma \| x_i - x \|^2) \tag{2.14}$$

In this work, the LibSVM tool (Chang & Lin 2011) has been used for performing SVM classification. The SVM classifier was trained using features extracted from training set of the Paraphrase corpus. Features from the test set were then used to evaluate the performance. The nu-Classification

scheme with Radial Basis Function kernel was found to be most successful as detailed in Section 2.4.1.

2.3 FEATURE SELECTION FOR PARAPHRASE RECOGNITION

The choice of features was found to have a significant effect on the performance of the Paraphrase Recognition system. When the feature space is large, determining the ideal set of features becomes challenging. This has motivated the application of Feature Selection technique to determine the best set of features. Though Feature selection approaches have previously been used for Paraphrase Recognition by Malakasiotis (2009) the focus was solely on feature reduction. This work focuses on identifying the set of features which improve the performance of Paraphrase Recognition.

2.3.1 Genetic Algorithm based Feature Selection

In problems involving large number of features, the application of Feature selection techniques helps to identify the ideal set of features for the problem at hand. This leads to benefits in terms of improved performance due to the elimination of irrelevant features, reduced storage requirements and better response time due to dimensionality reduction. Two major approaches for feature selection are the filter method and wrapper method. The filter method is a pre-selection method wherein feature selection is done independently of any classification system. On the other hand, wrapper methods receive feedback from the classifier. Feature subsets identified by a search on the feature space are evaluated using a classifier to determine their performance. Despite the additional cost involved in receiving feedback from the classifier for each subset, wrapper methods exhibit better classification performance (Jarmulak & Craw 1999, Vafaie & De Jong 1992, Zhuo et al 2008).

Genetic Algorithms (GA) are evolutionary computing procedures based on the human evolution process (Mitchell 1998). They are best suited for optimization problems which involve searching through the solution space. GA operates on an initial set of randomly generated solutions termed as the initial population. The solution to the problem is typically encoded as a binary string referred as a chromosome. The quality of the solutions is progressively refined by applying the 'Survival of the Fittest' principle and is assessed in terms of a Fitness function. During every iteration, alternatively called as generation, some members of the current population are selected using a suitable selection strategy; these are termed as parents. Genetic operators such as Crossover and Mutation are applied on the parent chromosomes to generate offspring. An example of two-point crossover has been shown in Figure 2.7.

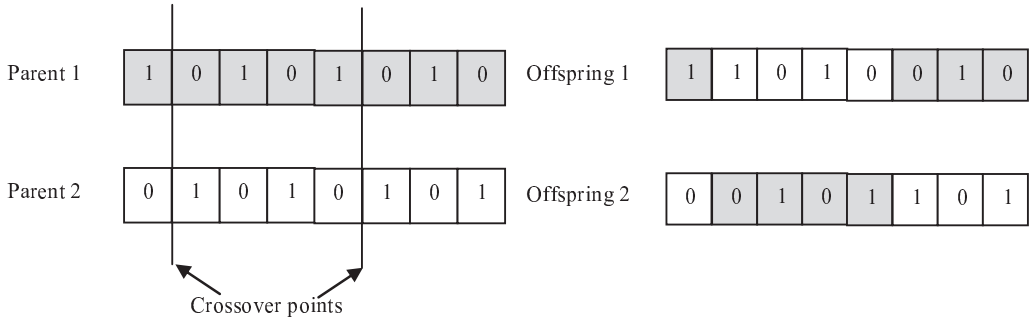


Figure 2.7 Two-point Crossover

Replacement policies are used to reconstitute the population by considering the fitness of the offspring. This procedure is repeated for several generations until the defined convergence condition is met. Convergence criteria are usually fixed in terms of the maximum number of generations, elapsed time or as no significant change in the fitness function in subsequent generations. The steps in GA based Feature Selection have been shown in Figure 2.8.

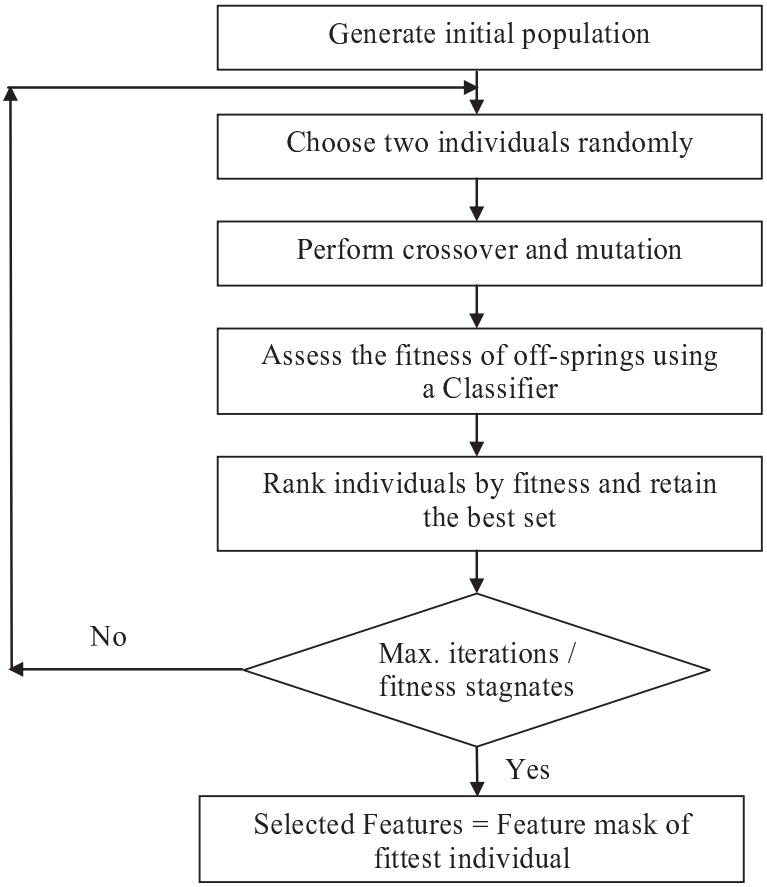


Figure 2.8 Process of Feature Selection using Genetic Algorithms

Genetic Algorithms have proven to be one of the most efficient search techniques for searching through the search space (Zhuo et al 2008). Being a random search methodology, GA strives to attain global optimization and hence performs better than local search techniques such as the greedy method. GA based feature selection performs well when large feature sets are involved. They are also domain independent search techniques which can perform well even when domain knowledge is unavailable (Vafaie & De Jong 1992). Hence Genetic Algorithm based Feature selection has been adopted considering the following facts: the size of the feature set involved (114 features) and the difficulty in encoding domain knowledge in Natural Language Processing applications.

The feature set influences the result of the Paraphrase Recognition task. Genetic Algorithm based Feature Selection has been attempted with the objective of identifying the ideal feature set and thereby improving the performance of Paraphrase Recognition systems. The Wrapper method has been adopted by combining the GA procedure with the SVM Classification scheme.

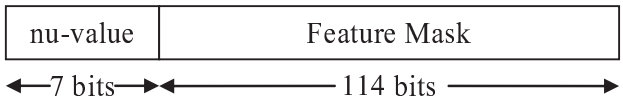


Figure 2.9 Chromosome Structure

Since the performance of the SVM Classifier was found to be dependent on both the features used and the value of the nu parameter, the chromosome structure for the Genetic Algorithm has been designed with two parts: the nu-value and the feature mask as shown in Figure 2.9. In earlier experiments the SVM classifier was found to exhibit best performance for nu values greater than 0.52 and less than 0.65. The base value of nu was fixed as 0.521 and 7 bits were used to encode the offset. The offset value ranging from 0 to 128 was divided by 1000 and added to 0.521 to get nu values between 0.521 and 0.649, for which the SVM classifier performed best. The size of the fitness mask was chosen to be 114 which is the total number of features extracted originally.

Even though obtaining reduced number of features is one of the major aims of feature selection, here the objective was to achieve improved classification performance. So the fitness of the chromosome was directly fixed as the accuracy of the SVM classifier for the selected features. Adopting Feature Selection has improved the performance of the Paraphrase Recognizer and has resulted in a significant reduction in the number of features as described in Section 2.4.2. The process has also yielded insight into the type of features that significantly influence the performance of the Paraphrase Recognizer.

2.4 RESULTS AND DISCUSSION

In order to evaluate the performance of the Paraphrase Recognizers, two different corpora namely MSRPC and the extended KMC have been used. The traditional measures used in evaluation of unranked Information Retrieval such as Accuracy, Precision, Recall and F-measure (Manning et al 2008) given in Equations (2.15) – (2.18) have been used for evaluating the performance of PR systems.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.16}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.17}$$

$$F - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.18}$$

The notations used are:

- True Positives (TP) –Paraphrases categorized as Paraphrases
- True Negatives (TN) –Non-Paraphrases correctly categorized
- False Positives (FP) –Non-paraphrases classified as Paraphrases
- False Negatives (FN) – Paraphrases classified as Non-paraphrases.

There is a traditional trade-off between Precision and Recall. Hence Accuracy and F-measure are the more comprehensive out of the four measures.

2.4.1 Performance Comparison of RBFNN and SVM based PR Systems

Lexical, syntactic and semantic features were extracted from the training set and test set of MSRPC. Experiments were conducted by considering the features individually and in various combinations. Besides individual features and category-wise combinations such as Lexical (LoCoS, BLEU and Skipgrams), Syntactic (POSPER, TED, TSF) and Semantic (Noun Verb Similarity, Proper Noun, Cardinal number match, Antonyms), other combinations of features were also tried. The performance of the paraphrase recognizers was evaluated in terms of the four metrics: Accuracy, Precision, Recall and F-measure.

The RBFNN system for Paraphrase Recognition was designed using three different schemes for construction of the hidden layer namely: k-means clustering, Fuzzy C-Means Clustering and Orthogonal Least Squares (OLS) Learning method. The performance of the resultant paraphrase recognition systems was evaluated by varying the following parameters: k and C – the number of clusters in k-means and FCM approaches, the exponent of the partitioning matrix for FCM Clustering, the spread parameter - σ and the maximum number of hidden layer neurons for the Orthogonal Least Squares approach.

The OLS approach where the neurons are added iteratively has registered better performance than the other two approaches for all the different feature sets. Though the performance is only marginally better, the OLS method does not have the overhead of clustering the input data initially and hence the training process consumes lesser time. Of the two clustering approaches, the k-means technique has been found to perform better but still

requires k as input. The comparison of the above systems with respect to MSRPC in terms of Accuracy has been presented in Table 2.1.

Table 2.1 Accuracy of RBFNN schemes for Paraphrase Recognition

Features	Accuracy %		
	k-means Clustering	**FCM Clustering**	**OLS method**
LoCoS	**67.4**	66.8	**67.4**
Noun Verb Similarity (NVSim)	68.2	68.3	**68.4**
Semantic Features	68.3	68.1	**68.6**
Skipgrams	69.2	69.1	**69.6**
TSF	68.9	68.4	**69.7**
TED	**70.8**	70.7	**70.8**
POSPER	**73.2**	71.2	**73.2**
BLEU	73.1	71.0	**73.7**
Syntactic Features	73.7	73.7	**73.9**
Lexical Features	**74.1**	73.9	**74.1**
Lexical + POSPER, NVSim, Proper Noun, Antonym	75.1	73.6	**75.5**
Lexical + POSPER, NVSim, Proper Noun	74.8	73.3	**75.5**
All features	75.4	74.8	**75.5**
All features excluding Cardinal number match	75.4	75.1	**75.6**

With respect to the Paraphrase Recognition system based on Support Vector Machines, experiments were conducted using a nu-classification scheme and all the four standard kernels. The corresponding parameters for each kernel such as r, d as well as the nu-value were varied to determine the optimum performance. The comparative accuracy of the various kernels on the MSRPC has been presented in Table 2.2. The RBF kernel was found to

perform better than the other three kernels and also required lesser number of input parameters.

Table 2.2 Accuracy of various SVM Kernels for Paraphrase Recognition

Features	Accuracy %			
	Linear	**Polynomial**	**Sigmoid**	**RBF**
Skipgrams	62.5	62.6	**62.9**	62.7
LoCoS	66.0	66.7	66.9	**67.0**
Noun Verb Similarity (NVSim)	66.5	66.5	65.3	**67.4**
TSF	67.5	63.5	**68.8**	68.2
Semantic Features	66.6	66.5	66.5	**68.4**
TED	68.5	69.7	68.5	**70.5**
POSPER	**72.7**	72.6	72.5	72.3
BLEU	72.7	**73.2**	72.8	73.1
Syntactic Features	73.1	73.2	73.1	**73.3**
Lexical Features	73.5	73.6	73.0	**73.7**
Lexical + POSPER, NVSim, Proper Noun, Antonym	75.0	75.2	74.9	**75.3**
Lexical + POSPER, NVSim, Proper Noun	75.2	75.1	75.1	**75.4**
All features excluding Cardinal number match	74.8	75.0	74.6	**75.5**
All features	75.2	75.1	75.1	**75.5**

Based on the results reported in Tables 2.1 and 2.2, a comparison of the RBFNN system using Orthogonal Least Squares approach and the SVM Paraphrase Recognition system using RBF Kernel was carried out with respect to all the four evaluation metrics and the results have been presented

in Table 2.3. The RBFNN method was found to demonstrate better accuracy for all the attempted feature sets. Considering only individual features, BLEU precision and recall were found to exhibit the best result. In the combinations based on category, lexical features outperformed syntactic and semantic features.

Table 2.3 Performance Comparison of SVM and RBFNN Paraphrase Recognizers on MSRPC

Features	Accuracy %		Precision %		Recall %		F-measure	
	SVM	RBFNN	SVM	RBFNN	SVM	RBFNN	SVM	RBFNN
LoCoS	67.0	**67.4**	67.7	**68.1**	**96.3**	96.0	79.5	**79.7**
Noun Verb Similarity (NVSim)	67.4	**68.4**	68.6	**70.0**	**94.3**	91.8	**79.4**	**79.4**
Semantic Features	68.4	**68.6**	69.0	**69.5**	**95.2**	94.1	**80.0**	79.9
Skipgrams	62.7	**69.6**	67.5	**72.2**	84.7	**88.1**	75.1	**79.4**
TSF	68.2	**69.7**	68.6	**72.6**	**96.3**	87.4	**80.1**	79.3
TED	70.5	**70.8**	74.1	**76.1**	**85.4**	81.9	**79.4**	78.9
POSPER	72.3	**73.2**	74.4	**75.6**	**88.6**	88.2	80.9	**81.4**
BLEU	73.1	**73.7**	73.2	**76.6**	**94.0**	87.1	**82.3**	81.5
Syntactic Features	73.3	**73.9**	74.4	**76.2**	**91.4**	88.4	**82.0**	81.8
Lexical Features	73.7	**74.1**	73.4	**75.1**	**94.8**	91.3	**82.7**	82.4
Lexical + POSPER, NVSim, Proper Noun	75.4	**75.5**	76.2	**77.7**	**91.5**	88.7	**83.2**	82.8
Lexical + POSPER, NVSim, Proper Noun, Antonym	75.3	**75.5**	76.4	**77.2**	**91.0**	89.7	**83.1**	83.0
All features	**75.5**	**75.5**	77.9	**78.0**	**88.1**	**88.1**	**82.7**	**82.7**
All features excluding Cardinal number match	75.5	**75.6**	77.7	**77.8**	**88.8**	88.6	**82.9**	82.8

Accuracy greater than 75% was achieved by SVM and RBFNN for the two combinations of features consisting of lexical (BLEU, LoCoS, Skipgrams), syntactic (POSPER) and semantic features (Noun/Verb Similarity, Proper Noun, Antonyms). When all the features were used, both systems have the same accuracy of 75.5%. On excluding the cardinal number match feature, RBFNN reached its highest accuracy value of 75.6% which is better than that of SVM.

In terms of Precision also the RBFNN method has performed better than the SVM system for all the feature categories. On the other hand, with respect to Recall the SVM method has exhibited better performance except in the case of Skipgram features. The performance evaluation has demonstrated the classic trade-off between Precision and Recall. The RBFNN method has an upper hand with respect to reduction of False Positives whereas the SVM technique reduces the False Negatives. With respect to F-measure, in nine of the fourteen cases SVM has performed marginally better, whereas for the remaining cases RBFNN has recorded better performance. On the whole, the performance of the RBFNN recognizer was found to be better than SVM technique with improved Accuracy, Precision and comparable F-measure values.

The performance of both the paraphrase recognition systems was also evaluated using the augmented Knight and Marcu Corpus (KMC) which contains 1087 pairs of positive cases and the same number of negative pairs (Cordeiro et al 2007). Since there is no standard partitioning of the corpus into training set and test set, experiments have been conducted using 10-fold cross-validation approach. Similar to the results obtained on the MSRPC, BLEU and lexical features were the best performing in terms of accuracy for the individual and category-wise feature sets respectively. The best performance was obtained when all the features were used. In all the cases the

RBFNN recognizer has performed better than the SVM system. The results obtained on the KMC indicate that the positive and negative cases are more easily separable when compared to the MSRPC.

Another common observation is that the positive cases in the Knight and Marcu corpus have considerable word overlap as in the sentence pair given below:

- Much of ATM 's performance depends on the underlying application.
- Like FaceLift, much of ATM's screen performance depends on the underlying application.

Table 2.4 Accuracy of SVM and RBFNN systems on extended KMC

Features	Accuracy %	
	RBFNN using OLS method	**SVM using linear kernel**
Noun Verb Similarity (NVSim)	**92.5**	90.0
Skipgrams	**93.1**	90.8
Semantic Features	**93.6**	91.2
LoCoS	**94.8**	94.7
POSPER	**96.6**	94.2
TSF	**97.6**	93.9
TED	**97.9**	93.9
BLEU	**98.1**	94.9
Syntactic Features	**98.2**	96.0
Lexical + POSPER, NVSim, Proper Noun	**98.2**	96.3
Lexical Features	**98.3**	96.1
Lexical + POSPER, NVSim, Proper Noun, Antonym	**98.3**	96.5
All features excluding Cardinal number match	**98.4**	98.0
All features	**98.4**	98.2

A comparison of the accuracy obtained using the RBFNN approach and the SVM technique have been presented in Table 2.4. Since the KMC was originally used in sentence compression experiments, one sentence is usually contained within the other. On the other hand in the negative samples added by Cordeiro et al (2007) from news sources, one sentence is usually unrelated to the other as shown in the following pair of sentences.

- The Palestinians say reopening the crossings is essential to rebuilding Gaza's shattered economy after three decades of Israeli control, especially with the harvest season approaching.
- At the same time, officials are relying heavily on extended families and community organizations to give seniors the personalized, computer-based assistance they need.

Another common phenomenon observed with respect to the negative cases is that exactly same sentences are given as input. Due to these factors the Paraphrase Recognition performance on the KMC is much higher than that observed on MSRPC. The highest accuracy value of 98.4% achieved by the RBFNN based PR system equals that registered by Cordeiro et al (2007) on the extended KMC.

Since the OLS learning methodology dynamically adapts the number of neurons in the hidden layer until convergence is reached, RBFNN exhibits superior generalization. Due to this reason the RBFNN performs better than SVM for the Paraphrase Recognition task. The number of hidden neurons influences the generalization ability of Neural Networks. If the number of hidden neurons is very less, the network does not train adequately. On the other hand, too many hidden neurons lead to over-fitting and the network loses its generalization ability. The RBFNN achieves better response time than SVM because the number of hidden layer neurons used in RBFNN

is less than the support vectors constructed in SVM. However the training time for SVM is lesser, whereas since RBFNN gradually increases the number of hidden neurons it takes a longer time to learn than SVM.

From the results obtained on both the MSRPC and KMC, the following conclusions can be drawn:

- the RBFNN system exhibits consistently better performance when compared to the SVM technique but incurs additional training time
- the choice of features has a major impact on the performance of the recognition system
- Lexical features perform better than the other categories since the extent of lexical overlap continues to play a major role in Paraphrase Recognition

2.4.2 Evaluation of GA based Feature Selection Approach

In an effort to improve the performance of Paraphrase Recognition as well as to determine the best set of features, Genetic Algorithm based feature selection has been attempted. A wrapper based approach has been followed by deploying the SVM Classifier for evaluating the goodness of the feature subsets.

During initial experiments, the population size and the number of generations were varied between 20 and 100. At lower values, sufficiently fit individuals were not produced whereas at higher values the procedure took longer to converge and there was meagre or no improvement. Hence the population size and the maximum number of generations were both set at 50. The parameters chosen have been listed in Table 2.5.

Table 2.5 Parameters for GA based Feature Selection

Parameter	Value
Population Size	50
Maximum number of generations	50
Mutation rate	2%
Crossover scheme	Two-point

Two-point crossover was used with a mutation rate of 0.02. Even for the same set of features, different nu-values were found to influence the performance of the classifier. In order to ensure greater variability in the nu-value, one of the cross-over points was chosen in the nu-value region while the other was chosen from the feature mask region. The mutation rate was fixed as 2% in order to avoid stagnation. In each generation, two candidates were randomly selected for crossover and the crossover point was also chosen at random.

The entire feature selection procedure was repeated for twenty times. The fittest individual produced in an iteration was included as a chromosome during the next iteration. This helped the selection process to converge faster. The best accuracy achieved was 76.97% when only 57 features were used with a nu-value of 0.561. The comparison between the best performance of the paraphrase recognizers with and without feature selection on the MSRPC is shown in Table 2.6.

Table 2.6 Evaluation of Feature Selection procedure on MSRPC

Technique	Number of Features used	Accuracy %	Precision %	Recall %	F-measure %
SVM Classifier without Feature Selection	114	75.5	77.9	88.1	82.7
SVM Classifier with feature selection	57	**77.0**	**80.5**	**88.1**	**84.1**

However the improved performance of the Feature Selection approach could in some measure be credited to the fact that the accuracy on the test set has been directly used in subsequent generations. The selected feature set (listed in Appendix 2) includes all the five lexical features, namely LoCoS, Skipgram precision and recall, BLEU precision and recall. It is notable that all the lexical features have been selected. This is in agreement with the results reported in Section 2.4.1, indicating the significant contribution of lexical features, because in typical cases of paraphrasing, the set of words found in the original sentence are reused to a great extent.

Out of the original set of nine semantic features only the verb, adverb similarity and two of the negation features were selected. Verb similarity is another notable contributing factor, as paraphrases tend to describe the same actions. A simple ablation experiment was conducted to study the effect of including the noun similarity feature on the performance of the Paraphrase Recognizer. The performance of the recognizer was found to drop to 76.04%. This could be attributed to an increase in false positives particularly in classifying non-paraphrase sentences, in which there is a great extent of noun similarity such as the example pair given below: “Ratliff's daughters, Margaret and Martha Ratliff, were adopted by Peterson after their mother's death” and “Peterson helped raise Ratliff's two daughters, Margaret and Martha Ratliff, who supported him throughout the trial”.

Only 48 Syntactic features were selected from among the original 100 syntactic features. Dependency tree edit distance and Triple Similarity function were two important syntactic features which were selected. POSPER features corresponding to simple and comparative adjectives, singular and plural nouns, particles, modals, ‘be’ forms of the verb have been selected. Several of the POSPER features corresponding to non-contributing though frequent classes such as Determiners, Symbols, Personal Pronouns, Interjections and List Markers were eliminated. The Feature selection process

sheds light not only on the features which contribute positively but also those that have a negative impact on the Paraphrase Recognizer.

Further, using only the selected set of 57 features obtained through the feature selection process on the extended KMC, resulted in an improvement, with an accuracy of 98.6% when compared to the previous best performance of 98.2% obtained when all features were used. There is an evident improvement in the performance of the recognizer with respect to all the evaluation parameters by using roughly half the number of features. The Classifier training and prediction processes are also faster since only half the number of features is used. The results are also better than that achieved by Malakasiotis (2009) who has reported an accuracy of 73.86% using Feature Selection by employing Forward Hill climbing and Beam search approaches. From the experiments it is evident that adopting Feature Selection has improved the performance of the Paraphrase Recognizer and has resulted in a significant reduction in the number of features.

2.4.3 Performance Improvement using Equivalent Phrases

Despite the performance enhancement achieved using Feature selection, there is scope for further improvement. With the complete test feature set as input, the SVM Recognizer yielded an accuracy of 75.5% on the MSRPC test set with 287 False Positive cases and 136 False Negative cases. In a further attempt to improve the performance of the Paraphrase Recognizers a detailed analysis of the error cases was carried out. An analysis of the False Positives indicates that the system misclassifies two sentences with considerably similar content as paraphrases even though one of them contains additional text as shown below:

- WorldCom's accounting problems came to light early last year, and the company filed for bankruptcy in July 2002, **citing massive accounting irregularities**.

- WorldCom's financial troubles came to light early last year, and the company subsequently filed for bankruptcy in July, 2002.

The incorporation of additional features which assess the significance of the dissimilar portions may help to overcome this problem. Another factor which was observed with respect to the MSRPC is the inconsistency in the annotation of the sentence pairs. Even in the presence of additional phrases, some cases were labeled as paraphrases while other such cases were marked as non-paraphrases as can be seen in the first two examples given in Table 2.7. Likewise, in the last two examples, sentence pairs which are in fact equivalent have been given different labels.

Table 2.7 MSRPC - Error Analysis

Label	Sentence 1	Sentence 2
Paraphrase	Buoyed by some of the advice imparted by Nicklaus Howell shot an 8 under 64 for a one stroke lead over Kenny Perry. (ID: 618359)	Buoyed by advice imparted by Nicklaus Howell shot an 8 under 64 **on Thursday to enter todays round** with a one stroke lead over Kenny Perry. (ID: 617945)
Non Paraphrase	This is America my friends and it should not happen here **he said to loud applause.** (ID: 3299227)	This is America my friends and it should not happen here. (ID: 3299188)
Paraphrase	NBC probably will end the season as the second most popular network behind CBS **which** is first among the key 18 to 49 year old demographic. (ID: 228847)	NBC probably will end the season as the second most popular network behind CBS **although it** is first among the key 18 to 49 year old demographic. (ID: 229207)
Non Paraphrase	NBC will probably end the season as the second most popular network behind CBS **although it** is first among the key 18 to 49 year old demographic. (ID: 228808)	NBC will probably end the season as the second most popular network behind CBS **which** is first among the key 18 to 49 year old demographic. (ID: 229298)

An analysis of false negatives indicates the presence of equivalent phrases which are difficult to identify. For instance in the paraphrase pair given below, the phrase “pulled back” is equivalent to “fell”.

- The Standard & Poor's 500 stock index pulled back by nearly 4 points to 1,066.62.
- The broad Standard & Poor's 500 Index <.SPX> fell 0.70 points, or 0.07 percent, to 1,069.42.

In other cases such as the pair given below, additional information is required for successful matching of inputs. Without embedding extra knowledge the Paraphrase Recognition system has no way of concluding that “the world’s two largest automakers” refers to “GM and Ford Motor Co.”.

- The world's two largest automakers said their U.S. sales declined more than predicted last month as a late summer sales frenzy caused more of an industry backlash than expected.
- Domestic sales at both GM and No. 2 Ford Motor Co. declined more than predicted as a late summer sales frenzy prompted a larger-than-expected industry backlash.

Therefore in order to improve the performance, a table of equivalent phrases was provided to the system. Before extraction of the various categories of features, pre-processing was performed to replace contractions such as “aren’t”, “didn’t” etc. by their equivalent longer phrases. Additionally if the sentence pair possessed equivalent phrases available in the table, the phrase in one sentence was replaced by its equivalent present in the other sentence. A sample set of phrases has been listed in Table 2.8.

Table 2.8 List of Equivalent Phrases

Phrase	Equivalent
Young woman	Female teen
NYPD cop	New York City police officer
From all sides	In and around
Air Transportation Stabilization Board	ATSB
Blackout	Power outage
Suddenly fresh slump	Surprise fall
From January to June	In the years first half
Compound the pain	Rubbing salt in the wound

The full set of such equivalent phrases identified for the MSRPC test set has been provided in Appendix 1. The phrases were identified by eliminating the overlapping portions in both sentences and then comparing the unmatched portions manually. The major categories of equivalent phrases include:

- Abbreviations and their expansions
- Idioms and their equivalent phrases
- Shortened versions
- World knowledge or facts

The effect of the pre-processing steps on the performance of the Paraphrase Recognition system was investigated by using the various features extracted from the MSRPC. A notable improvement in performance was observed as shown in Table 2.9. A significant reduction of False Negatives from 136 in the original case to 71 when using all features and 58 when using only the selected features was recorded. This shows that including phrasal pairs which indicate either linguistic equivalence or embed world knowledge proves beneficial to the process of Paraphrase Recognition.

Table 2.9 Performance Improvement using Equivalent phrases

Parameter	All Features (114)		Selected Features (57)	
	Original	Using Eqv. Phrases	Original	Using Eqv. Phrases
Accuracy %	75.5	**79.0**	77.0	**80.4**
Precision %	77.9	**78.7**	80.5	**79.5**
Recall %	88.1	**93.8**	88.1	**95.0**
F-measure %	82.7	**85.6**	84.1	**86.5**

2.5 SUMMARY

Two different machine learning approaches have been proposed for Paraphrase Recognition. The first approach based on Radial Basis Function Neural Networks has achieved good performance with an accuracy of 75.6% on the MSRPC. However it suffers from the disadvantage of extensive training requirements. The second approach has combined existing SVM classifiers with Genetic Algorithm based feature selection to improve the accuracy to 77% for the MSRPC with the added advantage of reducing the feature set size. Lexical features were found to have the best performance among the various categories of features. The provision of an external table of equivalent phrases as input to the Paraphrase Recognition system has further improved the accuracy to 80.4%. Since the MSRPC has a pre-defined training / test set split which may bias the results, the performance of both approaches have also been tested on a second corpus namely the extended KMC using a cross-validation approach. These experiments have also yielded similar results with improved accuracy values of 98.4% for the RBFNN method and 98.6% with feature selection.