CHAPTER 4

PARAPHRASE EXTRACTION USING FUZZY CLUSTERING

One of the major requirements for reducing information overload and focusing on relevant content is the identification of redundancy. Vast amounts of Natural language text are available on the Web as well as in large-scale repositories, much of which is redundant. The task of Paraphrase extraction focuses on identification of text units which convey the same meaning, from text corpora. The rich variability of natural language and the huge size of the corpora are two major challenges faced by Paraphrase Extraction systems. An effective Paraphrase Extraction system will benefit various Natural Language Processing applications such as Multi-document Summarization, Plagiarism detection, Question Answering and Document (Blog/Tweet/News-article) Clustering.

4.1 ARCHITECTURE OF PARAPHRASE EXTRACTION SYSTEM

This chapter reports the design of the proposed two-phase Paraphrase Extraction system using Fuzzy Hierarchical Clustering (PEFHC). In the first phase, shown in Figure 4.1, a Fuzzy Hierarchical Clustering approach has been adopted. Sentences from the corpus were preprocessed and clustered on the basis of verbs and nouns present in them. A sentence may describe multiple events and involve several entities and can therefore be placed within multiple clusters. Consequently, a Fuzzy Clustering approach

has been preferred. Further, since the number of clusters is not known, the Hierarchical Clustering technique has been used. All sentences which contain the same or similar verb were clustered together in the first step. Divisive Clustering was then used to split the clusters into sub-clusters, such that each sub-cluster relates to the same / similar set of nouns.

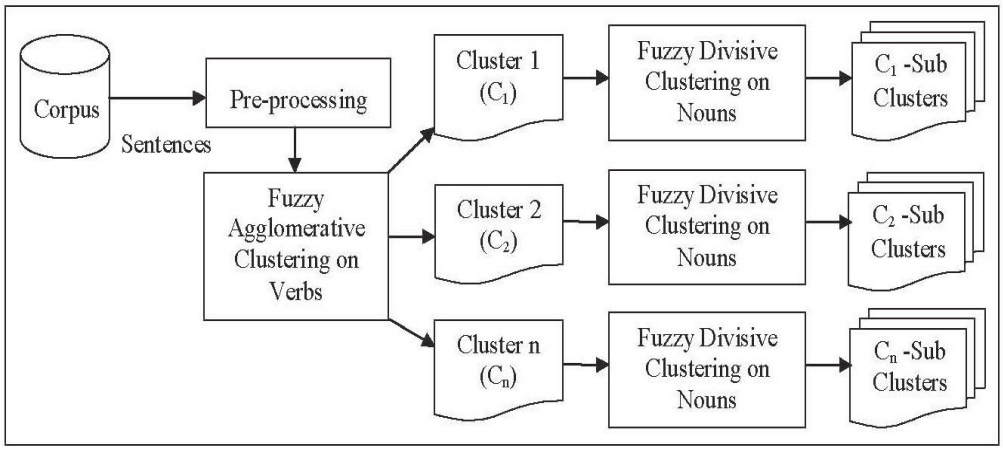


Figure 4.1 Fuzzy Hierarchical Clustering for Paraphrase Extraction

After forming clusters of sentences, in the second phase a Paraphrase Recognizer was used to identify the paraphrases within each cluster as shown in Figure 4.2. Various lexical, syntactic and semantic features were extracted from pairs of sentences which were then used by an SVM Classifier to classify each pair as positive or negative cases of Paraphrasing. The positive pairs have then been grouped together to produce a collection of sentence-level paraphrases.

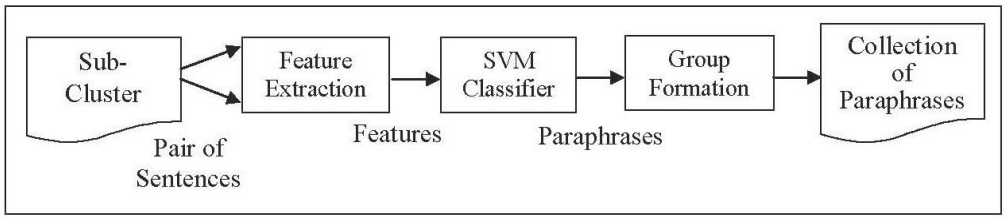


Figure 4.2 Collecting Paraphrases using a Paraphrase Recognizer

4.2 PREPROCESSING

The corpus may contain either full length documents or a collection of sentences. During preprocessing, the corpus was first split into sentences and unique identifiers were assigned to each sentence. Since the Clustering process was centered on the verbs and nouns within each sentence, the words of each sentence were assigned Parts-Of-Speech (POS) tags. For tagging, the Tree Tagger developed by Schmidt (1994) has been employed. Tree Tagger uses Probabilistic tagging to assign POS tags and also identifies the root form or lemma for each word of the sentence. The Tree Tagger can handle input from several languages. For English, tagging has been performed using the Penn Treebank tag set (Marcus et al 1993).

4.3 FUZZY AGGLOMERATIVE CLUSTERING BASED ON VERBS

In the first stage, Fuzzy Agglomerative Clustering based on verbs has been carried out. The reasons for choosing this approach for extraction of sentence-level paraphrases are as follows:

- Due to the ambiguous nature of natural language input, Fuzzy Clustering is better suited than Crisp techniques.
- Partitioning based approaches require the number of clusters to be pre-specified; therefore a Hierarchical clustering approach has been preferred.
- Typically in a large scale corpus, several sentences focus on the same or similar actions. Hence the sentences involving same verbs are first placed in the same cluster and these clusters are then merged depending on the similarity between the verbs.

- Hierarchical approaches have the ability to handle incremental data. New sentences can be added to the most similar cluster during any stage of the Clustering process.

The process of Fuzzy Agglomerative Clustering on Verbs starts with initial cluster formation, followed by grouping of similar clusters and finally merging of the clusters within each group. The Grouping and merging steps were repeatedly carried out, until no further merging was possible. The algorithm in pseudo-code is given in Figure 4.3.

Algorithm for Fuzzy Agglomerative Clustering based on Verbs
Initial Cluster Formation
For every sentence:

- The main verbs in the sentence are identified based on the POS tag
  - If there are no main verbs, the sentence is placed in the very first cluster
  - If there are one or more main verbs, the sentence is placed in the corresponding cluster(s) with membership = 1 / (number_of_main_verbs_in_sentence)
  - For each main verb in the sentence:
    - Root form of the verb is detected
    - If Word Sense Disambiguation(WSD) option is enabled, appropriate sense of the verb is chosen from WordNet by using the Lesk Algorithm; if WSD option is not enabled the first sense is used
    - the sense number is appended with root form of the verb to generate the cluster label
    - If a cluster with the generated label exists, the sentence is added to the cluster; if no such cluster exists, a new cluster is created and the sentence is added to the cluster.

Grouping of clusters

- A similarity matrix is constructed by computing the similarity between a cluster-label and all other cluster-labels using the WordNet based Jiang Conrath similarity measure.
- A fuzzy partitioning of the clusters is constructed as follows:
  - For each cluster, a group consisting of itself and all other clusters which have a similarity greater than the specified threshold is formed
  - Duplicates, sub-groups and singleton groups are eliminated

Merging of groups
For each group consisting of two or more candidates:

- The parent verb for all the candidates is identified. The parent is the Lowest Common Ancestor in the WordNet hierarchy.
- If WSD option is enabled, the best sense of parent verb with respect to the candidate verb senses is chosen; if not the first sense is chosen.
- The sense number is appended to the parent verb to generate the label of parent cluster
- If the parent cluster exists already, all the candidate clusters are added as children of the parent. If not a new parent cluster is created and the candidate clusters are attached as children

Repeat the process of grouping and merging clusters until no further groups are formed, which implies that all similarity values are < threshold.

Figure 4.3 Algorithm for Fuzzy Agglomerative Clustering based on Verbs

4.3.1 Initial Cluster Formation

A Fuzzy Clustering of the sentences was initially established by placing each sentence in as many clusters as there are verbs. For this purpose, the verbs in each sentence were identified. To prevent the formation of generic clusters, the process targets only the main verbs and ignores the auxiliary verbs. Main verbs were identified by their POS tags which typically begin with "VV". Sentences which do not contain a main verb were placed in the first cluster with a membership value of 1. The first cluster was assigned the label "NO_VERB". In the presence of multiple verbs, sentence i has to be placed in each corresponding cluster j, with the membership value as given in Equation (4.1). The membership of sentence i in all other clusters is set to 0.

$$\mu_{ij} = 1 / (\text{number of main verbs in sentence i}) \quad (4.1)$$

In order to achieve finer clusters, Word Sense Disambiguation (WSD) can be used. The specific sense of the candidate verb is determined based on the context of the verb by using the Lesk disambiguation algorithm (Navigli 2009). The original version of Lesk algorithm is based on the principle of computing the gloss overlap between two target words. The simplified version of the Lesk algorithm has been used to determine the correct sense of a word, given its context. The Lesk algorithm determines all possible senses of the target word as well as their glosses from WordNet. A target word may have multiple senses, each of which has a corresponding gloss or textual definition. The sense whose gloss has the highest degree of word overlap with the context has been chosen as the best sense of the word. Here the context refers to the sentence containing the target word.

In the absence of Word Sense Disambiguation, the default or most common sense was chosen for each word. In either case, the selected sense

number was concatenated to the target verb to generate the equivalent cluster label. In case the cluster label matched any of the existing clusters exactly, the sentence was placed in the matching cluster; otherwise a new cluster was created with the generated label and the sentence was added to it. The pseudo-code for the simplified Lesk algorithm (Vasilescu et al 2004) is given in Figure 4.4.

Simplified Lesk Algorithm

Input: A word w and its context C which is the sentence containing the word
Output: Best sense of the word based on its context

```
best-sense = the most frequent sense of the word as identified by WordNet
which is usually sense number 1
best-score = 0
for each sense Si of the word
        score = number of overlapping words between gloss of sense Si and
                context C
        if score > best_score
              best-score = score
              best-sense = Si
 return best_sense
```

Figure 4.4 Simplified Lesk Algorithm

4.3.2 Grouping of Clusters

Once the initial clusters were formed, similar clusters were then identified and combined. The Fuzzy Agglomerative Clustering approach used in this thesis differs from previous work with respect to the formation of overlapping fuzzy partitions or groups. Centroid Clustering technique has been followed for merging clusters, as each cluster has been labeled with a representative verb and similarity between clusters can be computed between their representative verbs. Similarity computation between words, most often uses WordNet (Fellbaum 1998). The Jiang-Conrath measure which assesses the similarity between two words in terms of the information content of the

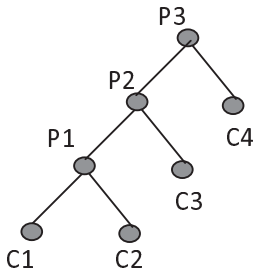
given words and their lowest common subsumer in the WordNet hierarchy has been used (Jiang & Conrath 1997). The Jiang-Conrath score computed using Equation (4.2) ranges between 0 and 1, with the value being closer to 1 as the similarity increases.

$$\mathrm{sim}(w_1, w_2) = \frac{1}{\mathrm{IC}(w_1) + \mathrm{IC}(w_2) - 2 * \mathrm{IC}(\mathrm{LCS})} \tag{4.2}$$

LCS represents the Least Common Subsumer or lowest common ancestor for the two words w_1, w_2 in the Wordnet hierarchy. IC represents the Information Content assessed using Equation (4.3) where P(w) is the probability of encountering an instance of word/concept w in a large corpus.

$$\mathrm{IC}(w) = -\log \mathrm{P}(w) \tag{4.3}$$

The traditional Centroid Clustering technique proceeds by identifying the pair of clusters which have the greatest similarity and then merges these clusters (Manning et al 2008). This pair of clusters will then be replaced by their parent cluster, resulting in a dendrogram with several levels as shown in Figure 4.5. In this work, a grouping strategy was utilized to first identify all clusters which have a similarity greater than or equal to a pre-specified threshold. Initially each cluster C_i was placed in a group of its own. Other clusters whose similarity with C_i were greater than or equal to the given threshold were then merged with C_i. The resultant clusters may be redundant. Hence before merging, all sub-groups, duplicate groups and groups containing a single element were removed from further consideration. The merging of multiple clusters in each step results in flatter dendrograms as shown in Figure 4.6.



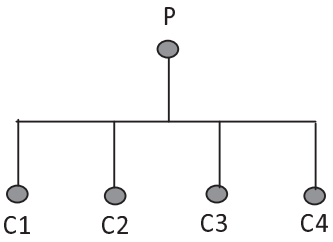


Figure 4.5 Binary Cluster merging Figure 4.6 Merging of multiple Clusters

The merging of clusters is illustrated using the pair-wise similarity matrix between five clusters shown in Figure 4.7. For a threshold of 0.4, the groups are {1, 4}, {2}, {3, 5}, {4, 1}, {5, 3}. After elimination of duplicate groups and singletons the remaining groups are {1, 4} and {3, 5}. When the threshold value is 0.2, the groups are {1, 2, 4}, {2, 1, 3, 4}, {3, 2, 5}, {4, 1, 2}, {5, 2, 3}. After elimination of duplicates and sub-groups, the remaining groups are: {2, 1, 3, 4} and {3, 2, 5}. It can be observed that cluster 2 is a part of two different groups. This strategy ensures a fuzzy partitioning of the clusters.

	1	2	3	4	5
1	1.0	0.3	0.15	0.4	0.05
2	0.3	1.0	0.2	0.3	0.1
3	0.15	0.2	1.0	0.18	0.5
4	0.4	0.3	0.18	1.0	0.15
5	0.05	0.1	0.5	0.15	1.0

Figure 4.7 Sample Similarity matrix between Clusters

The similarity threshold imposed during merging has been chosen based on a study conducted using the benchmark verb dataset consisting of 130 verb pairs developed by Yang & Powers (2013). The Jiang-Conrath scores for word pairs in the categories ‘inseparably related’ and ‘strongly

related' were examined. It was observed that 75% of scores for 52 word pairs in these two categories was greater than 0.15. Increasing the threshold to 0.2 brought down this percentage to 67% whereas reducing it to 0.1 resulted in the inclusion of several word pairs from the 'indirectly related' category. Hence it was decided that 0.15 and 0.2 are suitable similarity threshold values for grouping the clusters.

4.3.3 Merging of Clusters

Once the clusters are grouped, the candidate clusters within each group have to be merged together. The first step in the merging process, is the identification of the parent verb for the representative verbs from each cluster. The parent of two words in the WordNet hierarchy is the lowest common ancestor of the words in the WordNet hierarchy. This concept has been extended to multiple candidates using the following steps:

Step 1: Identify the Parent – P, or lowest common ancestor of the first two candidate cluster labels

Step 2: For each subsequent candidate cluster label L_i, determine the parent – P_i of P and L_i. If no parent exists, which happens when the words may not be directly related, the process terminates

Step 3: Set $P = P_i$ and repeat Step 2.

If no parent verb was identifiable, the candidates within the group were not merged. If a parent verb existed, and in case of WSD, the correct sense of the parent verb was determined by using the original Lesk algorithm (Navigli 2009). Here instead of matching the target word and the context, overlap was determined between each of the n senses of the parent and the

best sense of each candidate verb. In the absence of WSD, the default sense was used.

If the parent cluster happened to be any one of the existing clusters, the candidate clusters were all attached as child clusters of the parent; if not a new parent cluster was created. When a child cluster C_i was added to a parent cluster P_j its membership was assigned as in Equation (4.4) using the method proposed by Bank & Schwenker (2010). In the Equation, p represents all the parent clusters for the child cluster i, s_{ij} and s_{ip} are the Jiang Conrath scores calculated between the representative verbs of the child cluster and the parent cluster.

$$\mu_{C_iP_j} = s_{ij} / \sum_{p} s_{ip} \tag{4.4}$$

The membership value for a sentence k of the child cluster i in the parent cluster j-μ_{kj} has been calculated using Equation (4.5). It depends on μ_{ki} –membership of sentence k in cluster i and μ_{CiPj} – the membership of child cluster C_i in parent P_j.

$$\mu_{kj} = \mu_{ki}\mu_{C_iP_j} \tag{4.5}$$

Since all the membership values range between 0 and 1, the sentence k will have a higher membership in its own cluster i than in the parent cluster j. An example of the merging process is shown in Figure 4.8. The verb 'fling' is present in two different groups. The first group contains three verbs with the parent 'impel'; whereas the second group consists of seven verbs with 'displace' as the parent.

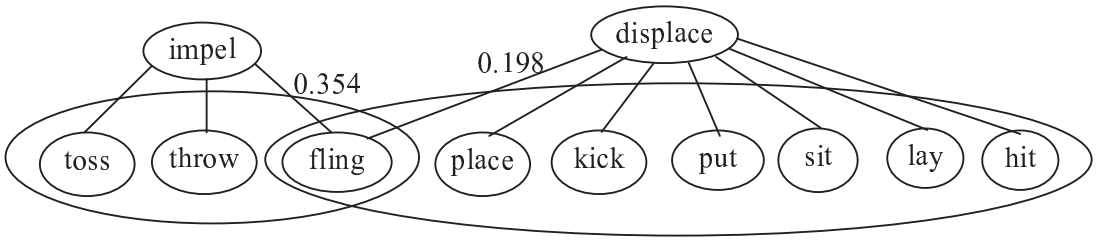


Figure 4.8 Example of Fuzzy Agglomerative Clustering

The similarity score between ‘fling’ and ‘impel’ is 0.354 whereas the score between ‘fling’ and ‘displace’ is 0.198. The cluster represented by 'fling' has membership values of 0.641 and 0.359 in the parent clusters 'impel' and 'displace' respectively computed as per Equation (4.4). Therefore a sentence which has a membership of 0.5 in the cluster 'fling' has a membership of 0.3205 in 'impel' and 0.1795 in 'displace' according to Equation (4.5).

After all groups were processed, the total number of clusters was reduced as the child clusters were merged within the parent clusters. A new similarity matrix was then constructed between the current set of cluster labels and the process of grouping and merging clusters was repeated until no further groups were formed. This results when all the similarity values are lesser than the specified threshold. At the end of this phase, several clusters each containing sentences with the same or similar verb were formed.

4.4 FUZZY DIVISIVE CLUSTERING

In the second stage of the Clustering process, the clusters formed by grouping sentences with the same or similar verbs were then divided such that all sentences within the same sub-cluster deal with same or related nouns. Fuzzy division was applied since a sentence may contain multiple nouns and may therefore belong to more than one cluster. The pseudo code for the Fuzzy division is given in Figure 4.9.

Algorithm for Fuzzy Divisive Clustering based on Nouns

For every cluster C:

- The nouns in each sentence are identified
- A similarity matrix of size S x S is constructed, where S is the number of sentences in cluster C. The matrix records the noun similarity for every pair of sentences in C.
- A fuzzy partitioning of the sentences is constructed:
  - For each sentence t:
    - Two most similar sentences t1 and t2 with similarity > threshold are chosen
    - The sentence is placed in the groups corresponding to t1 and t2
- Sub-groups are eliminated
- The cluster is split by creating child clusters.
  - Each sentence t is assigned to the respective child cluster with membership = to the original membership / number of child clusters to which t is assigned.

Figure 4.9 Algorithm for Fuzzy Divisive Clustering based on Nouns

The Divisive Clustering stage has focused on the nouns present in each sentence of the cluster. The nouns present in each sentence were extracted and the normalized noun similarity $nsim_{xy}$, between every pair of sentences x and y, has been computed as in Equation (4.6). n_x and n_y represent the number of nouns in the sentences x, y respectively. Likewise $noun\_x_i$ and $noun\_y_i$ refer to the nouns from sentences x and y respectively. The Jiang Conrath measure was used to compute the noun similarities.

$$nsim_{xy} = \sum_{i=1}^{n_x} (\max_{j=1}^{n_y} \{similarity(noun\_x_i, noun\_x_j\}) / n_x \qquad (4.6)$$

The similarity values determined using Equation (4.6) were then used to create a partition of the sentences within the cluster. A threshold value was used for controlling the division of clusters. The Miller Charles dataset (Bollegala et al 2011) was used to identify the ideal threshold value. 78% of the related Noun-pairs from the dataset had Jiang-Conrath scores greater than 0.15 and 71% had scores greater than 0.2. Hence the values 0.15 and 0.2 were identified as suitable threshold values.

In the next step, the sentences were partitioned using the threshold. Computing the alpha-cut of the threshold creates a hard partitioning (GhasemiGol et al 2010). Hence a strategy analogous to the one adopted for grouping similar clusters has been followed. A group was formed for each sentence, comprising of itself as well as other sentences which have a similarity greater than the threshold. Duplicate groups and sub-groups were eliminated as before. In practice, when the number of sentences is high, different groups which have a high degree of overlap may be formed. This, results in the presence of the same sentence in several groups, which ultimately increases the processing time and reduces the advantage gained due to clustering of sentences. Eliminating a smaller group which has considerable overlap with another group may lead to sentences which are present only in the smaller group being dropped altogether. This problem was overcome by permitting a sentence to be a part of only two groups to which it was most similar. In Figure 4.10, the 11 sentences in Cluster C have been split into four partitions: {S1, S2, S3, S4} {S5} {S2, S4, S6, S7, S8, S9} and {S8, S9, S10, S11}. Sentence S2, S4, S8 and S9 have been placed within two groups.

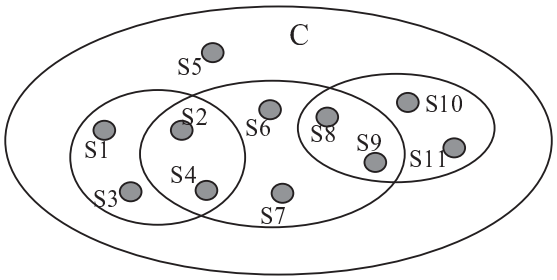


Figure 4.10 Example of Fuzzy Divisive Clustering

Once the groups were identified, the original cluster was split by creating child clusters and distributing the sentences according to the partitioning scheme. The membership value μ_{ij}, of a sentence i in the child cluster j, was computed using Equation (4.7).

$$\mu_{ij} = \mu_{iC} / (\text{number\_of\_groups\_containing\_i}) \quad (4.7)$$

Here μ_{iC} is the membership of the sentence in the original cluster C. The splitting of the clusters can be continued recursively until all the sentences within a cluster have a similarity not less than the given threshold. This continued splitting results in highly fragmented clusters and may produce several clusters with a single sentence. Hence, in this work recursive splitting has not been applied. After divisive clustering, each cluster contains sentences related to same/similar nouns and verbs.

4.5 IDENTIFICATION AND GROUPING OF PARAPHRASES

In the second phase of Paraphrase Extraction, the sentences within each cluster must be processed to identify the paraphrases. For this, the Support Vector Machine based Paraphrase Recognizer with features selected using Genetic Algorithm approach has been used (Section 2.3.1). Typically, all the sentences within a cluster are candidates for the Paraphrase Recognition process. In order to extract only the sentences which are very similar, the membership value of a sentence can be considered. A simple strategy would be to choose all sentences with membership greater than a threshold. Alternatively the top 50% of the sentences ranked by membership value can be chosen. The Classifier model which yielded a high performance of 77 % (Section 2.4.2) on the test set of the MSRPC has been used for the current work. Candidate sentences extracted from each cluster were fed to the Paraphrase Recognizer and classified. With respect to the MSRVDC there are no annotations available on whether all the sentences describing the same video are paraphrases or not. As the MSRPC is a standard corpora suitable for paraphrase recognition evaluation, in this work the classifier model constructed from the MSRPC training set has been used for the MSRVDC also.

Since the objective of this work was to extract all the equivalent sentences within a cluster, once the Paraphrase Recognizer categorizes the sentences, they have been collected together. Paraphrasing is transitive in nature, that is if A $\Leftrightarrow$ B and B $\Leftrightarrow$ C, then A $\Leftrightarrow$ C. Hence a chaining model was used to group the paraphrases within a cluster. The ensuing groups represent the paraphrases extracted from the original corpus.

4.6 EVALUATION METRICS

Precision is the common measure used in the evaluation of Paraphrase Extraction systems. Since Paraphrase Extraction was achieved through Clustering, measures which are used to evaluate the goodness of clusters have also been used here. The traditional extrinsic measures are Precision, Recall, Accuracy or Rand Index and F-measure. In the case of Paraphrase Extraction, since the number of possible pairings is very high, annotating the pairs as 'equivalent' or otherwise becomes difficult. Given a paraphrase corpus such as the MSRPC which has 5801 pairs, labeled as positive or negative, determining the labels for all the other possible pairs drawn from the collection is tedious. Hence two extrinsic measures Precision and Relative-Recall have been used.

Precision is defined as the number of true paraphrases out of the total number of sentence pairs declared as paraphrases as given in Equation (4.8).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.8}$$

where TP refers to True Positives and FP stands for False Positives. Due to the non-availability of decisions (paraphrase / non-paraphrase) for all the

possible sentence pairs from the MSRPC, TP and FP were assessed only with respect to the labeled sentence pairs.

The other metric Relative Recall (RR) has been adapted from the Information Retrieval version (Agosti & Melucci 2000) and assesses the recall of the current system S with respect to other systems as given in Equation (4.9). This measure has been used due to limited information on all the possible relevant paraphrase pairs.

$$\text{relative\_recall}_S = \frac{\text{Number of relevant pairs extracted by S}}{\text{Cumulative set of relevant pairs extracted by all systems}} \tag{4.9}$$

Purity, Entropy and V-measure are the popular measures used to assess the quality of clustering. Purity and Entropy assess the homogeneity of a cluster that is the degree to which clusters contain only objects from a single class (Reichart & Rappoport 2009). V-measure focuses on both homogeneity as well as completeness, which is the extent to which all objects from a single class are assigned to a single cluster.

Purity of a cluster is the ratio of the highest number of common objects between the cluster and any one class to the number of objects in the cluster. Given C clusters and K reference classes, purity of a cluster C_i is given by Equation (4.10).

$$P_i = \frac{1}{|C_i|} \max_j (|C_i \cap K_j|) \tag{4.10}$$

The overall Purity (P) for a set of clusters is the weighted sum of the Purity of the individual cluster as given by Equation (4.11).

$$P = \frac{1}{N}\sum_{i=1}^{|C|}(|C_i|\,P_i) \tag{4.11}$$

Entropy of a cluster E_i is a measure of disorder and measures how the classes are distributed within a cluster as given in Equation (4.12). The Overall Entropy E is the weighted average of the individual cluster entropies (Skabar & Abdalgader 2013) as given in Equation (4.13).

$$E_i = -\frac{1}{\log|K|}\sum_{j=1}^{|K|}\frac{|C_i \cap K_j|}{K_j}\log\frac{|C_i \cap K_j|}{K_j} \tag{4.12}$$

$$E = \frac{1}{N}\sum_{i=1}^{|C|}(C_i E_i) \tag{4.13}$$

V-measure focuses on both homogeneity as well as completeness, which is the extent to which all objects from a particular class are assigned to a single cluster. As shown in Equation (4.14) V-measure is defined as the harmonic mean of homogeneity h - Equation (4.15) and completeness c – Equation (4.16). The parameters h and c are calculated in terms of entropies H(K), H(C) and conditional entropies H(K|C), H(C|K) given in Equations (4.17) to (4.20) where C_i is a cluster and K_j represents a reference class. By varying β, the preference given to h and c can be controlled; if both are equally important, β is set to 1(Geiss 2011).

$$V = \frac{(\beta+1)hc}{(\beta h + c)} \tag{4.14}$$

$$h = 1 - \frac{H(K|C)}{H(K)} \tag{4.15}$$

$$c = 1 - \frac{H(C|K)}{H(C)} \tag{4.16}$$

where

$$H(K) = -\sum_{j=1}^{|K|} \frac{|K_j|}{N} \log \frac{|K_j|}{N} \tag{4.17}$$

$$H(C) = -\sum_{i=1}^{|C|} \frac{|C_i|}{N} \log \frac{|C_i|}{N} \tag{4.18}$$

$$H(K \mid C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{|C_i \cap K_j|}{N} \log \frac{|C_i \cap K_j|}{|C_i|} \tag{4.19}$$

$$H(C \mid K) = -\sum_{j=1}^{|K|} \sum_{i=1}^{|C|} \frac{|C_i \cap K_j|}{N} \log \frac{|C_i \cap K_j|}{|K_j|} \tag{4.20}$$

Besides the above measures, the Partition Coefficient (PC) has been used specifically with reference to fuzzy clustering. The Partition Coefficient measure (Bezdek 1981) given in Equation (4.21) quantifies the fuzziness of a partition with higher values which indicate least fuzzy clustering, being preferable.

$$PC = \frac{1}{N} \sum_{i=1}^{|C|} \sum_{j=1}^{N} \mu_{ij}^2 \tag{4.21}$$

In Equation (4.21), N represents the total number of objects and μ^{ij} is the membership value of object j in cluster i.

4.7 EXISTING EXTRACTION SYSTEM

The proposed system – Paraphrase Extraction using Fuzzy Hierarchical Clustering (PEFHC) has been evaluated against the techniques used by Wubben et al (2009) for Paraphrase Acquisition from Newspaper headlines. Wubben et al's system has been chosen for comparative evaluation

since the focus of their work is very much similar to the proposed one. Wubben's system works on clusters of Google News headlines. The originally available Google News headlines have been first re-clustered into finer sub-clusters by using the k-means algorithm from the CLUTO software package (CLUTO Toolkit). The PK1 Cluster Stopping Criterion has been used (Pederson & Kulkarni 2006). After clustering, all the sentences within a cluster were aligned pair-wise. A second approach has also been used by Wubben et al without clustering. All the sentences in the originally available Google News clusters were matched pair-wise and the cosine similarity score was computed. If the similarity exceeded the upper threshold, the matching was accepted as paraphrases; if the similarity was less than a lower threshold, the pair was rejected. When the similarity was between the thresholds, the context in which the headline occurred was used to decide the result. The Cosine similarity approach has been used by Wubben as a baseline.

For the current performance evaluation, both the systems proposed by Wubben et al (2009) have been used with minor modifications. In the k-means clustering approach, the PK3 cluster stopping criteria has been used in place of PK1 due to the following reasons: PK3 has been reported to be more efficient than PK1 (Kulkarni 2006) and PK1 requires additional work to fix a suitable threshold. Also Wubben et al have applied the clustering approach to existing clusters of similar sentences in order to create sub-clusters. But in the current context, since no clusters exist initially, k-means clustering has straight-away been applied on the entire corpus. The other variation is with respect to the pair-wise cosine similarity computation where due to the lack of context when dealing with stand-alone sentences, a single threshold value has been used. If the similarity exceeds the threshold then the sentence pair is accepted as equivalent and rejected otherwise.

Besides Wubben et al's systems, a Fuzzy C-Means (FCM) clustering approach has also been adopted. Wubben et al (2009) have attempted k-means clustering, while the proposed system uses fuzzy clustering. In order to obtain a comprehensive performance evaluation the FCM approach has also been implemented. The optimal number of clusters has been identified by choosing the partition which yields highest partition coefficient and lowest classification entropy.

4.8 RESULTS AND DISCUSSION

This section describes the performance evaluation of the Fuzzy Clustering approach for Paraphrase Extraction. Some of the factors which influence the evaluation of Paraphrase Extraction systems are as follows:

- A one-on-one comparison of Paraphrase Extraction systems may not be possible as the systems may work on different types of corpora and extract units of different sizes.
- The systems may be evaluated either in a stand-alone manner or in the context of specific tasks such as information extraction, query expansion etc.
- The efficiency of the PR system used. The best Paraphrase Recognition systems today have an accuracy of about 77% [Table 1.1 – Section 1.5.1.4] which will definitely impact the task of Paraphrase Extraction.
- Availability of bench-mark corpora is yet another challenge. At present, though there are a few large-scale sentence-level paraphrase collections such as the Microsoft Research Video

Description Corpus, they have not yet been completely annotated.

- Choosing suitable evaluation metrics is another hurdle because of the difficulty in identifying the set of all Positives and Negatives in large scale corpora.

The performance of the Fuzzy Clustering based Paraphrase Extraction system has been evaluated on three different datasets: the Microsoft Research Paraphrase Corpus, a small subset of the Microsoft Research Video Description Corpus (MSRVDC) and the complete set of English language sentences from MSRVDC. The results have been compared with that of the Automatic Paraphrase Acquisition system developed by Wubben et al (2009) as well as FCM Clustering.

4.8.1 Experiments on MSRPC

The Microsoft Research Paraphrase Corpus (MSRPC) is best suited for evaluating the performance of Paraphrase Recognition systems. In order to use it for Paraphrase Extraction, the corpus has been viewed as a collection of individual sentences. The 10,948 unique sentences out of the 11,602 sentences were taken up for further processing after assigning unique IDs. Since MSRPC contains pairs of paraphrases rather than groups, only Precision and Relative Recall have been calculated.

For testing the performance of the PEFHC system, the sentences were subject to Fuzzy Agglomerative Clustering followed by Divisive Clustering both with and without WSD. As described in Sections 4.3 and 4.4, the thresholds for merging and splitting of clusters were chosen as 0.15 and 0.2. Candidate sentences were chosen from each cluster based on two

schemes: all sentences and 50% of sentences from each cluster based on the membership values. Using these choices, eight variants of the PEFHC system were framed using three sets of options, namely: With and Without WSD, threshold of 0.15 and 0.2, using all sentences in a cluster and only the top 50%. The best set of features (57 features) identified in the Feature Selection process (Section 2.4.2) were extracted from each pair of sentences and passed on to the Paraphrase Recognizer to classify the sentences. Since the MSRPC contains only paraphrase pairs the chaining of paraphrases described in Section 4.5 has been omitted.

Equation (4.8) has been used to assess the precision by first picking the set of sentence pairs with a decision from the complete set of retrieved pairs. The positive cases retrieved were counted as True Positives whereas the negative cases were considered as False Positives. Table 4.1 presents the performance of the Existing system as well as PEFHC variants in terms of pairs with a known decision, that is True Positives and False Positives.

The system which involves direct computation of cosine similarity was tested with three different thresholds starting from 0.5 and an increment of 0.1. The value of T represents the cut-off threshold used in the Cosine Similarity system and the semantic similarity threshold for the PEFHC variants. The Cosine Similarity based system with low thresholds was found to retrieve a larger number of known pairs followed by the k-means approach. Though it has reported a very large number of possible pairs, the FCM approach was found to retrieve the least number of known pairs. The WSD based variants of the proposed system retrieved more candidates due to the tighter clusters based on specific word senses.

Table 4.1 Statistics of retrieved Paraphrase pairs

System		Number of known pairs	TP	FP
k-means Clustering (Wubben et al 2009)		3926	2770	1156
Cosine Similarity (Wubben et al 2009)	T=0.5	4935	3607	1328
	T=0.6	3902	3041	861
	T=0.7	2677	2154	523
FCM Clustering		564	372	174
No WSD, T=0.15, Top 50%		2998	2409	589
No WSD, T=0.15, All		3177	2542	635
No WSD, T=0.2, Top 50%		2844	2285	559
No WSD, T=0.2, All		2973	2380	593
WSD, T=0.15, Top 50%		3177	2572	605
WSD, T=0.15, All		3238	2620	618
WSD, T=0.2, Top 50%		3140	2544	596
WSD, T=0.2, All		3252	2633	619

The Precision of the existing approaches: Cosine Similarity, k-means Clustering, FCM Clustering as well as all the proposed PEFHC variants have been presented in Table 4.2. The PEFHC system outscores the existing systems in terms of precision by combining the Fuzzy Clustering approach with a Paraphrase Recognizer which exploits lexical, syntactic and semantic features. The best precision is registered by the most rigorous system: With WSD, threshold = 0.2 and extracting only top 50% sentences. Since these options result in finer clusters, precision is better.

Table 4.2 Precision of Existing and PEFHC approaches

<table>
<tr><th colspan="3">PEFHC Variants</th><th>Precision %</th><th>Existing System Variant</th><th>Precision %</th></tr>
<tr><td rowspan="4">No WSD</td><td rowspan="2">Threshold 0.15</td><td>Top 50%</td><td>80.4</td><td rowspan="2">k-means Clustering– (Wubben et al 2009)</td><td rowspan="2">70.6</td></tr>
<tr><td>All</td><td>80.0</td></tr>
<tr><td rowspan="2">Threshold 0.2</td><td>Top 50%</td><td>80.3</td><td>FCM Clustering</td><td>68.1</td></tr>
<tr><td>All</td><td>80.0</td><td colspan="2" rowspan="2">Cosine Similarity (Wubben et al 2009)</td></tr>
<tr><td rowspan="4">WSD</td><td rowspan="2">Threshold 0.15</td><td>Top 50%</td><td>80.9</td></tr>
<tr><td>All</td><td>80.9</td><td>Threshold = 0.5</td><td>73.0</td></tr>
<tr><td rowspan="2">Threshold 0.2</td><td>Top 50%</td><td>81.0</td><td>Threshold = 0.6</td><td>77.9</td></tr>
<tr><td>All</td><td>80.9</td><td>Threshold = 0.7</td><td>80.5</td></tr>
</table>

From Table 4.2 the following inferences can be drawn: Using WSD option yields better precision across all cases. This can be attributed to the fact that, by using WSD a distinction can be made among the sentences having the same verb. This results in finer clusters and hence during Paraphrase Extraction the number of true positives is higher. With respect to similarity thresholds, since both the thresholds yield comparable performance, any one of the thresholds can be used in the PEFHC system. This observation is in line with the study conducted on the Miller-Charles dataset and Yang-Powers dataset for fixing the thresholds. In terms of the membership values of sentences within a cluster, the strategy of using only the top ranking 50% yields slightly better precision than using all the sentences. This shows that the precision is affected only by the overall Clustering and not by the specific membership values.

Relative Recall has been assessed with respect to four systems namely:

- Best proposed system variant using WSD option, threshold of 0.2 and top 50% of sentences.
- k-means approach (Wubben et al 2009)
- Cosine Similarity variant (Wubben et al 2009) using threshold = 0.7
- FCM Clustering approach

The Relative Recall computation has also been carried out according to Equation (4.9) with respect to known relevant pairs or True Positives (Table 4.1) retrieved by the systems. The results have been presented in Table 4.3 and the k-means approach exhibits the highest relative recall. This is followed by the proposed PEFHC variant whereas the FCM Clustering approach has least relative recall. The system which retrieves most pairs is found to have higher relative recall as there is a greater scope of containing the relevant pairs.

Table 4.3 Relative Recall Evaluation

System	Relative Recall
k-means Clustering	0.78
Proposed system variant –WSD, Threshold = 0.2, Top 50%	0.73
Cosine Similarity – Threshold = 0.7	0.64
FCM Clustering	0.11

From the results it is obvious that the proposed system has yielded the best precision and also possesses reasonable relative recall when compared to the other variants.

4.8.2 Experiments on MSRVDC

The Microsoft Research Video Description Corpus (MSRVDC) was constructed by Chen & Dolan (2011). The corpus consists of 1,20,000 sentences collected from multi-lingual descriptions of short video snippets supplied by workers on Mechanical Turk. 85,550 English sentences were contributed by 733 workers. The workers were classified as Tier 1 and Tier 2 depending on the quality and consistency of their video descriptions, with Tier-2 being the better category. Out of the 85,550 English sentences, 33,855 were from 50 Tier-2 workers. For evaluating the performance of PEFHC, two different datasets were constructed from MSRVDC.

The first dataset was constructed with the objective of judging the performance of the PEFHC system in extracting groups or clusters of paraphrases. It was observed that many of the clusters described the same actions and involved the same entities. Hence in order to test the performance of the system in a controlled environment consisting of clusters with little or no overlap it was decided to handpick a set of clusters and to limit the size of the dataset to around 2000 sentences. 143 clusters involving distinct verbs were manually chosen. The sentences were extracted from the 33,855 sentences contributed by Tier-2 workers. Descriptions of the same video were treated as belonging to a single cluster. Duplicate sentences were eliminated from within each cluster and the major or repeated verbs in each cluster were identified. Each cluster was inspected by two judges to eliminate sentences which did not agree with the overall theme of the cluster; disagreement between judges was resolved by a third judge.

The PEFHC system, the existing systems (Wubben et al 2009) and FCM Clustering approach were tested on this dataset. For implementing the proposed system, after fuzzy clustering, all the candidate sentences within a cluster were classified by the paraphrase recognizer and the positive pairs

were chained together as described in Section 4.5 to generate clusters of paraphrases. As in the case of the MSRPC, the various implementations of PEFHC with respect to similarity threshold, sentence membership and WSD were investigated. Since the focus is on extracting groups of paraphrases, the second set of performance evaluation measures namely entropy, purity and v-measure were computed as shown in Table 4.4. The 143 clusters identified were fixed as reference classes against which the created clusters have been judged.

Table 4.4 Performance of PEFHC variants on MSRVDC Dataset 1

	Without WSD				**With WSD**			
	Threshold =0.15		**Threshold = 0.2**		**Threshold = 0.15**		**Threshold = 0.2**	
	Top 50%	**All**	**Top 50%**	**All**	**Top 50%**	**All**	**Top 50%**	**All**
Entropy %	5.7	11.0	5.6	10.9	5.1	6.9	**4.8**	6.9
Purity %	64.8	**79.4**	65.6	78.8	63.1	70.8	61.3	70.1
v-measure %	83.9	83.1	84.0	83.0	84.0	84.1	**84.3**	**84.3**

In terms of entropy, the best performance is registered by the rigorous system, with WSD, threshold of 0.2 and using only the top 50% in terms of membership values. Using WSD forms several smaller clusters and therefore the entropy is lesser. Likewise using a threshold of 0.2 and only top 50% of sentences is more restrictive, hence the entropy is lower. On the other hand, the most lenient system that is the one without WSD, threshold = 0.15 and considering all sentences within a cluster, has the highest purity. This can be attributed to the fact that this variant, results in the formation of lesser number of large clusters which tend to have a greater degree of overlap with reference classes. Decreasing the threshold, increases the entropy as sentences belonging to several classes are placed within the same cluster. At the same

time, purity also increases as larger clusters which have a higher degree of overlap with a given class are formed. With respect to the v-measure, which is more comprehensive as it considers both homogeneity as well as completeness, both the variants with WSD and threshold of 0.2 have registered the best performance of 84.3%. Using a higher threshold with WSD improves the homogeneity. The rigorous system was chosen as the best out of the eight variants as it has moderate performance with respect to purity, low entropy and high v-measure. Table 4.5 records the performance of the existing systems. It can be observed that similar to the results obtained on MSRPC, the cosine similarity method achieves better performance than the k-means clustering technique.

Table 4.5 Performance of Existing systems on MSRVDC Dataset 1

Metric	k-means Clustering (Wubben et al 2007)	FCM Clustering	Cosine Similarity with varying Thresholds (T)		
			T = 0.5	T =0.6	T = 0.7
Entropy %	19.7	32.6	93.5	63.9	**10.3**
Purity %	39.0	47.7	11.3	33.6	**55.0**
v-measure %	80.0	67.3	11.4	33.9	**84.0**

Comparing PEFHC and existing systems, it can be seen that almost all the eight variants of the PEFHC system perform better with respect to all three parameters. Additionally the rigorous variant of the proposed Fuzzy hierarchical Clustering approach and the FCM approach were compared in terms of the Partition Coefficient yielding values of 0.44 and 0.16 respectively. This indicates that the quality of fuzzy clustering is better in the proposed system. Hence it can be concluded that the proposed system is better at identifying groups of paraphrases due to refinement of fuzzy clusters by using the paraphrase recognizer.

The second dataset was also extracted from MSRVDC and consists of 27,291 unique sentences from 33,855 Tier-II English sentences. All sentences describing the same video were grouped into a cluster and a total of 1931 clusters were formed. Due to the large size of the corpus, only the lenient and rigorous variants of PEFHC have been evaluated against Wubben's Clustering approach, Cosine Similarity systems with threshold=0.7 and FCM approach as shown in Table 4.6.

Table 4.6 Performance Evaluation on MSRVDC Dataset 2

Metric	PEFHC without WSD Threshold 0.15, All	PEFHC with WSD Threshold 0.2, Top 50%	k-means Clustering	Cosine Similarity Threshold 0.7	FCM Clustering
Entropy %	34.9	**19.4**	68.0	66.9	86.8
Purity %	**44.1**	41.3	2.4	7.0	0.8
v-measure %	69.0	**77.1**	42.7	43.2	19.6
Partition Coefficient	0.8	**0.9**	-	-	0.3

The rigorous variant of the PEFHC system performs much better than the other systems in terms of Entropy as well as v-measure. In terms of purity, the lenient variant performs well as in the case of Dataset 1, but the improvement in purity is less when compared to the increase in entropy. In terms of Partition Coefficient also the rigorous variant performs better than the lenient variant as well as FCM Clustering. Therefore the rigorous variant was chosen as the better of the two PEFHC variants.

Sample clusters produced by the various systems have been shown in Table 4.7. The clusters produced by the PEFHC system and Cosine similarity approach tend to be more homogeneous. It can be observed that in the cluster produced by the PEFHC system all sentences involve the same

verb – 'reading' and related nouns – 'teacher' or 'woman'. Though the cosine similarity clustering is better than the k-means approach, it has also considered only word matching and not the concepts. FCM Clustering has resulted in large-sized clusters of low purity and high entropy and has therefore not been included in Table 4.7.

Table 4.7 Sample Clusters on MSRVDC Dataset 2

PEFHC system clustering (Purity = 0.82, Entropy = 0.08)	**k-means Clustering (Purity -0.11 (highest), Entropy – 0.37(lowest))**	**Cosine Similarity (Purity -0.78, Entropy – 0.07)**
• A woman is reading something, while another woman measures the first woman's ankle with a tape measure. • A female teacher reads to the class. • A female teacher is reading out loud from a piece of paper. • A teacher reads a paper to the class. • A teacher reads aloud from a piece of paper. • A teacher reads to her class. • A woman is holding a sheet of paper and reading the text. • A woman is reading a piece of paper. • The teacher is reading the paper. • The teacher reads from a paper to the class.	• A woman is applying a lotion to her hair. • Workers are tending to the field. • A woman is dancing by the water. • A woman is baking a fish in the pan. • People sing and dance in a surreal scene. • A woman trims a flower plant. • A cat is jumping into a cardboard box. • A woman is straightening her hair. • The boxers fought in the ring. • A whale rises out of the water. • A whale surfaces in the water. • Two dogs are wrestling. • The man did acrobatic jumps in his routine. • Two women eat hamburgers in a cafe and chat. • Two men are standing in a kitchen and one is talking on a cell phone. • A boy and woman are playing catch with a pumpkin. • Someone is putting sauce into a pan. • A little kid dances.	• A group of cars are driving down a road. • Several different kinds of racing cars are driving down a road. • A group of deer are crossing a road. • A group of deer cross the road in a forest. • A group of deers are crossing road. • A herd of caribou are crossing a road. • A herd of deer are crossing a road. • A herd of deer are crossing the street. • A herd of deer cross a road.

4.8.3 Discussion

The proposed system - PEFHC has consistently exhibited better performance with respect to all the three datasets as shown in Figure 4.11. Of the four systems, the two stage PEFHC approach which uses Fuzzy Clustering followed by Paraphrase Recognition performs best. The rigorous variant which uses a threshold of 0.2, WSD option during Fuzzy Clustering and only the top 50% of sentences within each cluster as candidates for Paraphrase Recognition has been chosen for comparison against the existing systems. The second best performance is demonstrated by Wubben's Cosine similarity approach.

The rigorous variant has been chosen for comparison against the existing systems. Of the four systems, the two stage proposed PEFHC system performs best. The second best performance is demonstrated by Wubben's Cosine similarity approach. A significant aspect is that, the improvement in performance of the proposed system is more with respect to MSRVDC Dataset 2, which is the biggest of the three datasets.

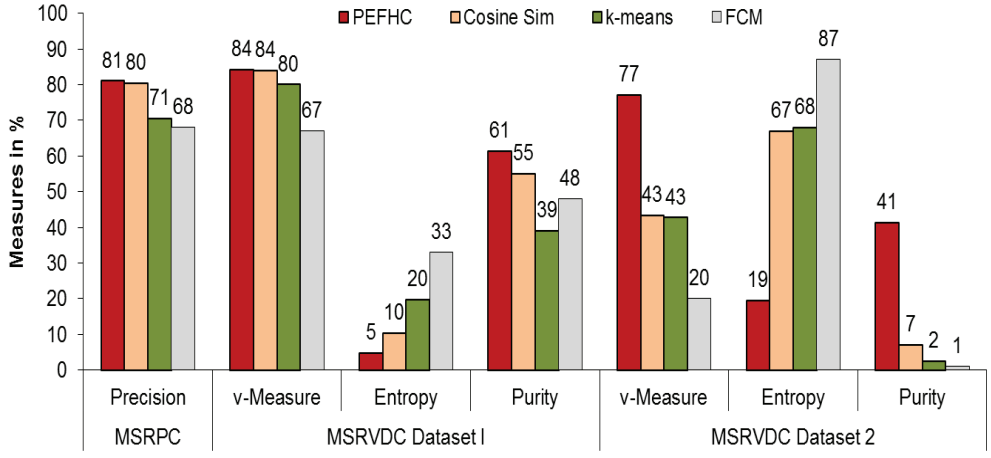


Figure 4.11 Performance Evaluation of Proposed and Existing systems

With respect to k-means clustering, the determination of ideal k value for Clustering is tedious for large corpora. Cosine similarity computation has a complexity of $O(N^2)$ where N is the number of sentences in the corpus. In Fuzzy Hierarchical Clustering approach the number of clusters is determined automatically. Also, once the sentences are initially assigned to clusters in O(N) time all further operations are either within the clusters or between the cluster representatives. Therefore the efficiency of the Fuzzy Hierarchical Clustering approach is better. The proposed approach also supports incremental clustering as a new sentence can be added to the clusters labeled with the most similar verbs.

The major contribution of this work is the design of a novel Fuzzy Hierarchical Clustering approach and its application for the task of Paraphrase Extraction. Though several sentence clustering approaches exist previously in Information Extraction and multi-document summarization applications, the uniqueness of this approach is its usage of a Fuzzy Hierarchical technique based on Sentence similarity. The approach has been tested on two different corpora and the following inferences can be drawn from the performance evaluation.

- PEFHC approach results in meaningful and cohesive clusters when compared to other clustering approaches such k-means and FCM.
- It is computationally efficient when compared to the Cosine similarity approach and at the same time considers semantic similarity also.
- The approach is incremental and can also be parallelized as Divisive clustering can be carried out on each cluster independently.

- Therefore the Fuzzy Hierarchical Clustering approach is a viable alternative to existing distributional and bootstrapping approaches for Paraphrase Extraction and can be employed for Sentence-level paraphrase extraction.

4.9 SUMMARY

A two-stage approach centered on Fuzzy Clustering followed by Paraphrase Recognition has been proposed for Paraphrase Extraction. The significant aspects of the approach are the usage of a soft hierarchical clustering scheme which has scope for parallelism and the ability to perform incremental clustering. Further a novel fuzzy grouping strategy has been utilized for merging clusters which ensures faster clustering and flatter hierarchies. The system has been evaluated on two different paraphrase corpora and has exhibited good performance in comparison to a Cosine Similarity technique as well as k-means and FCM Clustering approaches. The effect of applying WSD has also been investigated and was found to improve the performance. The approach can also be adapted for tasks such as News Headline / Tweet Clustering, Plagiarism detection and Multi-document Summarization.