Applied Soft Computing 34 (2015) 426–437

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Paraphrase Extraction using fuzzy hierarchical clustering

A. Chitra [a,1], Anupriya Rajkumar [b,*]

[a] Department of Computer Applications, PSG College of Technology, Coimbatore 641004, Tamil Nadu, India
[b] Department of Computer Science and Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi 642003, Tamil Nadu, India

ARTICLE INFO

Article history:
Received 15 January 2014
Received in revised form 7 May 2015
Accepted 7 May 2015
Available online 19 May 2015

Keywords:
Fuzzy Agglomerative Clustering
Divisive clustering
Paraphrase Recognition

ABSTRACT

Paraphrase Extraction involves the discovery of equivalent text segments from large corpora and finds application in tasks such as multi-document summarization and document clustering. Semantic similarity identification is a challenging problem which is further compounded by the large size of the corpus. In this paper a two-stage approach which involves clustering followed by Paraphrase Recognition has been proposed for extraction of sentence-level paraphrases from text collections. In order to handle the ambiguity and inherent variability of natural language a fuzzy hierarchical clustering approach which combines agglomeration based on verbs and division on nouns has been used. Sentences within each resultant cluster are then processed by a machine-learning based Paraphrase Recognizer to discover the paraphrases. The two-stage approach has been applied on the Microsoft Research Paraphrase Corpus and a subset of the Microsoft Research Video Description Corpus. The performance has been evaluated against an existing *k*-means clustering approach as well as cosine-similarity technique and Fuzzy C-Means clustering and the two-stage system has consistently demonstrated better performance.

1. Introduction

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. The detection of similar text is complicated due to the rich variability of natural languages. The large scale of the corpora is another factor which poses hurdles in Paraphrase Extraction. Performing one-on-one matching of sentences is practically ruled out, even if it is assumed that a suitable Paraphrase Recognition system is available. Therefore, efficient techniques are required to identify possibly similar candidates from large scale corpora and then subject them to further processing to detect exact matches. An effective Paraphrase Extraction system will benefit various Natural Language Processing applications such as multi-document summarization, plagiarism detection, question answering and document clustering.

The significant aspect of this work is that a novel two-level fuzzy clustering technique has been proposed for sentence-level Paraphrase Extraction. As similar sentences tend to describe the same or similar actions, Fuzzy Agglomerative Clustering based on verbs is performed initially. Divisive clustering is then applied to identify sub-groups of sentences which center on the same nouns. A Support Vector Machine (SVM) based Paraphrase Recognizer is then used to identify the paraphrases within each cluster. The performance of the Paraphrase Extraction system has been evaluated using the Microsoft Research Paraphrase Corpus (MSRPC) and a subset of the Microsoft Research Video Description Corpus (MSRVDC).

The outline of the paper is as follows: Section 2 contains an overview of previous work related to Paraphrase Extraction and Hierarchical Clustering. Section 3 describes the methodology adopted for extracting paraphrases using a fuzzy hierarchical clustering approach and machine learning based Paraphrase Recognizer. Section 4 presents the results of experiments conducted using two different corpora. Possible directions for future work and applications are discussed in Section 5, which concludes the paper.

2. Related work

The task of Paraphrase Extraction or acquisition aims at extracting paraphrases from a given corpus. Several approaches exploit Harris's distributional hypothesis which states that words in similar contexts tend to have same meanings [1]. Bhagat and Ravichandran [2] have used this approach to extract phrase level entities, by constructing a feature vector for each phrase based on its context and computing the cosine similarity between the feature vectors of phrases. Metzler and Hovy [3] have deployed the

* Corresponding author. Tel.: +91 9443760000.
E-mail addresses: ctr.psg@gmail.co (A. Chitra), anupriya_rajkumar@yahoo.co.in (A. Rajkumar).
[1] Tel.: +91 9843222273.

http://dx.doi.org/10.1016/j.asoc.2015.05.017
1568-4946/

distributional hypothesis in a Hadoop-based framework to operate on large-scale corpora. Bootstrapping methods rely on seed patterns for acquiring paraphrases. Szpektor et al. have used terms from a domain-specific lexicon and coupled these with frequently co-occurring noun phrases to form seed slots and have then extracted templates [4]. Regneri et al. have extracted phrase level paraphrases from similar sentence pairs by assigning semantic role labels and then locating equivalent arguments or anchors. Dependency parse of the sentences has then been used to group anchors with their corresponding predicates, yielding matched predicate argument structures which were then extended to extract equivalent phrases [5].

Barzilay and Lee (2003) have applied hierarchical complete-link clustering to cluster the sentences describing the same type of event [6]. Sentence level paraphrases were discovered by applying multiple sequence alignment on pairs of sentences from each cluster. A similar sequence alignment technique has been employed by Regneri and Wang [7] to first extract sentence-level paraphrases and then phrase-level paraphrases. Yan et al. [8] have extracted multilingual phrasal paraphrases by aligning definition sentences extracted from Wikipedia articles. A minimally supervised approach was then used to extract paraphrases by computing global and local similarity measures between phrasal pairs from the aligned sentences. Wubben et al. have compared Clustering against pair-wise matching for extracting paraphrases from news corpora [9]. A *k*-means clustering approach was used to subdivide already existing clusters of headlines. Sentence-level paraphrases were then extracted by matching all possible sentence pairs within each cluster.

Traditional clustering algorithms create a hard partitioning of data in which each object is assigned to only one cluster. Fuzzy clustering is an alternate, wherein a soft partition is constructed with each object belonging to multiple clusters with different degrees of membership. Two popular variants of fuzzy clustering are Fuzzy C-Means and fuzzy hierarchical approach. The Fuzzy C-Means approach is a fuzzy variant of the original *k*-means partitioning approach and is limited by the necessity of specifying 'c'; though techniques for automatically detecting 'c' exist, they become infeasible in large corpora.

Hierarchical clustering methods operate on the inter-cluster similarity matrix. Bank and Schwenker [10] have proposed a fuzzified version of the agglomerative algorithm where, on merging C_i and C_j to create C_{ij}, only the similarity value S_{ij} is set to zero and the remaining values corresponding to the clusters C_i and C_j are retained. This permits the clusters C_i and C_j to take part in subsequent mergers. When a cluster is a part of several parent clusters, normalized membership values are assigned. Rodrigues et al. [11] have proposed an Online Divisive Agglomerative Clustering where a semi-fuzzy approach has been used for assigning objects to clusters by computing the distance between the object and the candidate clusters during division. If the computed distance is greater than the Hoeffding bound, the object is assigned to the corresponding cluster otherwise it is assigned to multiple clusters. Diaz-Valenzuela et al. [12] have developed a fuzzy hierarchical clustering technique which works in a semi-supervised fashion for classifying scientific publications. The method outperforms the unsupervised approach but requires instance-level constraints to be specified to determine the optimal α-cut of the dendrograms produced by hierarchical clustering.

Clustering of sentences is a common task which finds several applications. Seno and Nunes [13] have used an incremental approach for clustering sentences from multiple documents belonging to Portuguese language. The first cluster was created with the first sentence, and each subsequent sentence was either added to an existing cluster based on cosine similarity and word overlap measures or assigned to a new cluster. Khoury has proposed a sentence clustering approach based on Parts-Of-Speech (POS) information [14]. A POS hierarchy has been used to compute the distance between two words or POS based on which existing clusters are split.

Fuzzy approaches have been previously applied to both document clustering as well as sentence clustering. Rodrigues and Sacks [15] have proposed Hierarchical Hyper-spherical Fuzzy C-Means (H^2-FCM) approach for clustering documents which uses cosine similarity for forming hyper-spherical clusters that are then merged. Skabar and Abdalgader [16] have proposed a fuzzy clustering algorithm for relational data termed as FRECCA, where sentence similarity values are used rather than the vector space representation of the sentences and expectation maximization has been used to determine cluster memberships. Wazarkar and Manjrekar have proposed an extension of the FRECCA approach which forms hierarchical clusters by using a divisive clustering approach [17].

From a study of related literature it has been concluded that, of the approaches applied for Paraphrase Extraction, the alignment approach is better suited for the extraction of sentence level paraphrases. Applying alignment approach directly on a large corpus is infeasible. Hence it is better to first apply clustering and then match sentences within the cluster. Fuzzy clustering approach is more applicable to natural language input than crisp approaches. Likewise hierarchical clustering approach which automatically establishes the number of clusters using thresholds on the similarity metric and also supports incremental clustering is better than partitioning approaches.

3. Fuzzy clustering for Paraphrase Extraction

In this work, Paraphrase Extraction is performed in two stages: fuzzy hierarchical clustering stage which focuses on grouping similar sentences followed by Paraphrase Recognition applied on each pair of sentences within a cluster. Since a sentence may describe multiple events and involve several entities and can therefore be placed within multiple clusters, fuzzy clustering approach has been preferred. Further, since the number of clusters is not known the hierarchical clustering technique has been used. The process of fuzzy hierarchical clustering shown in Fig. 1 consists of the following steps:

- Sentences from the corpus are subject to pre-processing where Parts-Of Speech tags are assigned.
- All sentences which contain the same verb are clustered together. These clusters are then merged based on the similarity between the verbs.
- Divisive clustering is then used to split the clusters into sub-clusters, such that each sub-cluster relates to the same/similar set of nouns.

In the second stage of the Paraphrase Extraction process, a Paraphrase Recognizer is used to identify the paraphrases within each cluster as shown in Fig. 2. Various lexical, syntactic and semantic features are extracted from pairs of sentences and an SVM classifier uses these features to classify each pair as positive or negative cases of Paraphrasing. The positive pairs are grouped together due to the transitive nature of paraphrases, to produce the output which consists of a collection of sentence-level paraphrases.

3.1. Preprocessing

During preprocessing, each sentence is assigned a unique identifier. Since the clustering process is centered on the verbs and nouns within each sentence, the words of each sentence are assigned POS tags using the Tree Tagger developed by Helmut Schmidt [18]. The

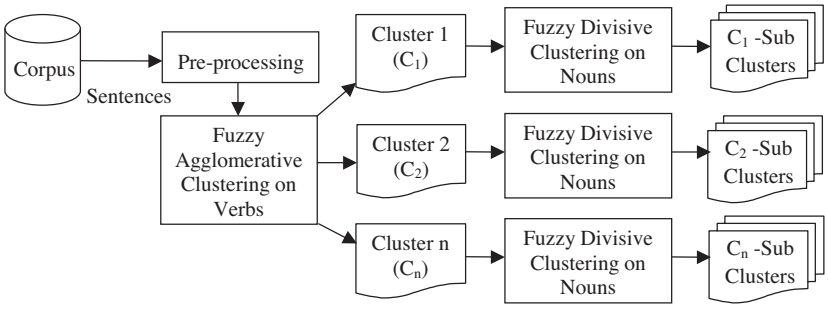


Fig. 1. Fuzzy hierarchical clustering for Paraphrase Extraction.

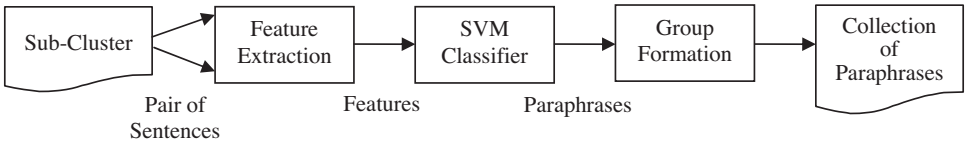


Fig. 2. Collecting paraphrases using a Paraphrase Recognizer.

Tree Tagger uses Probabilistic tagging to assign POS tags based on the Penn Treebank tagset [19] and also identifies the root form or lemma for each word of the sentence.

3.2. Fuzzy agglomerative clustering based on verbs

In order to cluster the sentences, a two-stage approach has been used. In the first stage fuzzy agglomerative clustering based on verbs has been carried out due to the following reasons:

- 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 same or similar actions. Hence sentences involving same verbs are first placed in the same cluster, these clusters are then merged depending on the similarity between the verbs.
- 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 and merging of clusters. The grouping and merging steps are repeatedly carried out, until no further merging is possible. The algorithm has been given in Fig. 3.

3.2.1. Initial cluster formation

A fuzzy clustering of the sentences is initially established by placing each sentence in as many clusters as there are verbs. To prevent the formation of generic clusters, the process targets only main verbs and ignores auxiliary verbs. Sentences which do not contain a main verb are placed in the first cluster with a membership value of 1. In case multiple verbs are present, sentence i has to be placed in the corresponding clusters – j, with a membership value as follows:

$$\mu_{ij} = \frac{1}{\text{number\_of\_main\_verbs\_in\_sentence\_}i} \quad (1)$$

$$\mu_{ik} = 0 \quad \text{for all other clusters } k \quad (2)$$

In order to create 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. The simplified version of the Lesk algorithm is used to determine the correct sense of a word, given its context [20,21]. 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 is chosen as the best sense of the word. Here, context refers to the sentence containing the target word.

In the absence of WSD, the default or most common sense is chosen for each word. In either case, the selected sense number is concatenated to the target verb to generate the cluster label. In case the cluster label matches any of the existing clusters, the sentence is placed in the matching cluster; otherwise a new cluster is created with the generated label and the sentence is added to it. The pseudo-code for the simplified Lesk algorithm [21] is given in Fig. 4.

3.2.2. Grouping of clusters

Once the clusters are formed, similar clusters will have to be combined. The fuzzy agglomerative clustering approach used here differs from previous work with respect to the formation of overlapping fuzzy partitions or groups. Each cluster is labeled with a representative verb and similarity between clusters is computed between their labels. The Jiang–Conrath measure [22] which assesses the similarity between two words in terms of information content of the given words and their lowest common subsumer in the WordNet hierarchy as given in Eq. (3) has been used to compute the similarity.

$$sim(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 * IC(LCS)} \quad (3)$$

In Eq. (3) LCS represents the Least Common Subsumer or lowest common ancestor for the two words w_1, w_2 in the Wordnet hierarchy and IC represents the information content assessed using Eq. (4).

$$IC(w) = -\log\ P(w) \quad (4)$$

where $P(w)$ is the probability of encountering an instance of word w in a large corpus.

The traditional Centroid Clustering technique proceeds by identifying the pair of clusters which have greatest similarity and merges them; this, results in dendrograms with several levels as shown in Fig. 5a. In this work, a grouping strategy is utilized to

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 no parent is available the group is not merged.
- 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. If not a new parent cluster is created and the candidate clusters are attached to it.

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

Fig. 3. Algorithm for Fuzzy Agglomerative Clustering based on verbs.

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
```

Fig. 4. Simplified Lesk algorithm [21].

first identify all clusters which have a similarity greater than or equal to a pre-specified threshold. Initially each cluster C_i is placed in a group of its own. Each group is then expanded, by including all other clusters whose similarity with C_i exceeds the threshold. Before merging, all sub-groups, duplicate and singleton groups are removed from further consideration. The merging of multiple clusters in each step results in flatter dendrograms as shown in Fig. 5b.

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 and Powers [23]. The Jiang–Conrath scores for word pairs in the categories 'inseparably related' and 'strongly related' were examined and it was concluded that 0.15 and 0.2 are suitable similarity threshold values.

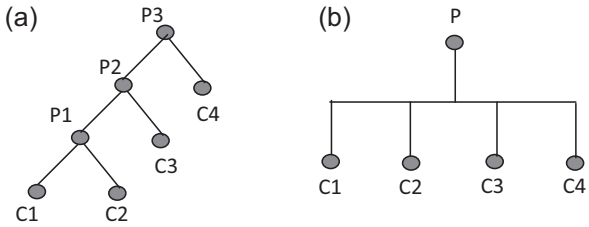


Fig. 5. (a) Binary merging of clusters. (b) Merging of multiple clusters.

Algorithm for Fuzzy Divisive Clustering based on Nouns

For every cluster:

- 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 the cluster. The matrix records the noun similarity for every pair of sentences in the cluster.
- A fuzzy partitioning of the sentences is constructed:
  - For each sentence t:
    - The 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 is assigned to the respective child cluster with membership equal to the original membership / the number of child clusters to which the sentence is assigned.

Fig. 6. Algorithm for fuzzy divisive clustering based on nouns.

3.2.3. Merging of clusters

Once the clusters are grouped, the candidate clusters within each group have to be merged together by first determining 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 is 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 can be identified the candidates within the group are not merged. If a parent verb exists, and in case of WSD, the correct sense of the parent verb is determined by using the original Lesk algorithm [20]. Here instead of matching the target word and context, overlap is 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 is used.

If the parent cluster happens to be any of the existing clusters, the candidate clusters are all attached as child clusters of the parent; if not a new parent cluster is created. When a child cluster C_i is added to a parent cluster P_j its membership is assigned according to Eq. (5), using the method proposed by Bank and Schwenker [10]:

$$\mu_{C_i P_j} = \frac{s_{ij}}{\sum_p s_{ip}} \tag{5}$$

where p represents all the parent clusters for the child cluster i, s_{ij} and s_{ip} are the Jiang–Conrath scores calculated between representative verbs of the child cluster and the parent cluster. The membership value for a sentence k of the child cluster i in parent cluster j is computed using Eq. (6).

$$\mu_{kj} = \mu_{ki}\mu_{C_i P_j} \tag{6}$$

After all groups have been processed, a new similarity matrix is constructed between the current set of cluster labels and the process of grouping and merging clusters is repeated until no further groups are formed. This results when all similarity values are lesser than the specified threshold.

3.3. Fuzzy divisive clustering

The clusters formed by grouping sentences with the same or similar verbs are then divided such that all sentences within the same sub-cluster deal with the same or related nouns. Fuzzy division is applied since a sentence may contain multiple nouns and may therefore belong to more than one cluster. The algorithm used is given in Fig. 6.

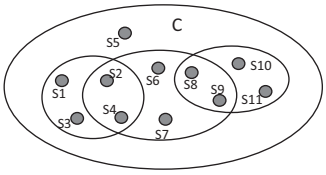


Fig. 7. Example of fuzzy divisive clustering.

The normalized noun similarity *nsim*, between every pair of sentences x and y, is computed as in Eq. (7). n_x and n_y represent the number of nouns in the sentences x and y, respectively. The Jiang–Conrath measure is used to compute the noun similarities.

$$nsim_{xy} = \sum_{i=1}^{n_x} \frac{\max_{j=1}^{n_y}\{similarity(noun\_x_i, noun\_x_j)\}}{n_x} \tag{7}$$

These similarity values are used to create a partition of the sentences within the cluster. A threshold value is used for controlling the division of clusters. The similarity scores for the Miller–Charles dataset [24] were used to identify suitable threshold values of 0.15 and 0.2.

In the next step the sentences are partitioned based on the threshold. A group is 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 are eliminated as before. In practice, when the number of sentences is high, different groups which have a high degree of overlap are formed. This, results in the presence of the same sentence in several groups, which ultimately increases processing time and reduces the advantage gained due to clustering of sentences. This problem is overcome by permitting a sentence to be a part of only two groups to which it is most similar to. In Fig. 7, the 11 sentences in Cluster C are split into four partitions: {S1, S2, S3, S4} {S5} {S2, S4, S6, S7, S8, S9} and {S8, S9, S10, S11}. Sentences S2, S4, S8 and S9 are placed within two groups.

Once groups are identified, the original cluster is split by creating child clusters and distributing the sentences according to the partitioning scheme. The membership value μ_{ij}, of a sentence i in child cluster j, is computed using Eq. (8).

$$\mu_{ij} = \frac{\mu_{iC}}{\text{number\_of\_groups\_containing\_}i} \tag{8}$$

where μ_{iC} is the membership of the sentence in the original cluster C. Splitting of clusters can be continued recursively until all 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 with the same/similar nouns and verbs.

3.4. Paraphrase Recognition

In the second phase of Paraphrase Extraction, the sentences within each cluster must be processed to identify paraphrases. Sentences which share the same verbs and nouns need not necessarily be paraphrases. Therefore an SVM based Paraphrase Recognizer [25] has been used to identify the paraphrases within each cluster. Of the existing approaches for Paraphrase Recognition, machine learning techniques, notably SVMs have achieved considerable success [1] by extracting various features from the input sentences. Lexical, syntactic and semantic features have been previously used both individually and in various combinations for detecting paraphrases [1,25]. In order to extract only 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. A better tactic is to rank sentences by membership and choose only the top 50% as it avoids the usage of a threshold.

3.4.1. Lexical features

The adapted Bi-Lingual Evaluation Understudy (BLEU) precision and recall metrics [26] have been calculated by considering the extent of unigram match with respect to each of the input sentences. Skipgrams of a sentence can be formed by considering both contiguous and non-contiguous n-grams [27]. Skipgram precision and recall have been computed by dividing the number of common skipgrams by total possible number of Skipgrams constructed from each of the input sentences. The Longest Common Subsequence feature was evaluated by dividing the length of the longest common in-sequence portion by the length of the shorter sentence.

3.4.2. Syntactic features

Stanford Parser developed by Klein and Manning [28] was used to construct dependency trees which are syntactic representations of a sentence as shown in Fig. 8. Tree edit distance has been computed as the minimum number of operations required to transform one dependency tree into another [29]. For every edge in a dependency tree a triple can be formed in terms of parent/head-word, child/dependent and relationship between them. 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 [30]. A total of 96 POSPER (Parts-Of-Speech Position independent word Error Rate) features [31] have been generated by taking into account the degree of matches and non-matches for each tag.

3.4.3. Semantic features

Four word similarity features have been evaluated by computing the Jiang–Conrath score between nouns, verbs, adverbs and adjectives. As in [32], features which assess the extent of cardinal number and proper noun match have been used. Additional negation features which consider the number of antonym occurrences as well as the presence of explicit negation terms have also been used.

From earlier work carried out using a SVM classifier on the MSRPC, it was observed that the choice of features has a significant effect on the performance of the Paraphrase Recognizer as shown in Table 1 [25]. For discovering the best subset of features, Wrapper method of feature selection was employed by combining genetic algorithms with SVM classifiers. The best performance was obtained using 57 features out of the original set of 114 features. The selected features include:

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

After feature extraction, for classifying the input sentences as paraphrases, a SVM classifier has been used. The LibSVM tool [33] has been used for performing SVM classification. A nu-classification scheme with a Radial Basis Kernel function was employed. The classifier model which yielded a high performance of 76.97% [25] on the test set of the MSRPC has been used for the current work. Candidate sentences extracted from each cluster are 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.

3.5. Grouping of paraphrases

Since the objective of this work is to extract all the equivalent sentences within a cluster, once the Paraphrase Recognizer categorizes the sentences they must be 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 is used to group the paraphrases within a cluster.

4. Results

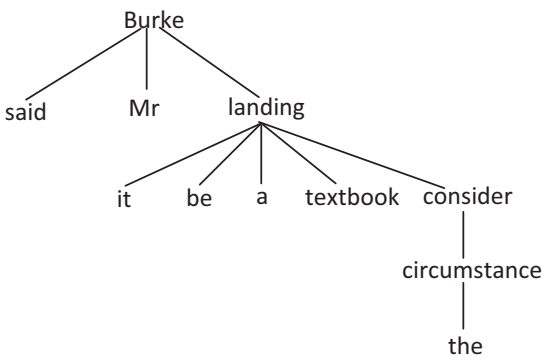
Some of the factors which influence the evaluation of Paraphrase Extraction systems are:

- 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 underlying Paraphrase Recognition system. The best Paraphrase Recognition systems today have an accuracy of only 77% [1] which will definitely impact the task of Paraphrase Extraction.
- Availability of bench-mark corpora is yet another challenge. Though there are a few large-scale sentence-level paraphrase collections such as the MSRVDC, annotating the entire corpus is difficult.
- Choosing suitable evaluation metrics is another hurdle because of the difficulty in identifying set of all Positives and Negatives in large scale corpora.

The fuzzy hierarchical clustering approach has been implemented in Java and the WordNet electronic dictionary has been used as an additional resource. The performance of the proposed approach has been evaluated on two different corpora: the MSRPC and the MSRVDC. The obtained results have been compared with that of the Paraphrase Acquisition system developed by Wubben et al. [9].

4.1. Evaluation metrics

In addition to two traditional Paraphrase Extraction evaluation measures – precision and relative-recall [2], clustering measures



nn(Burke-3, Said-1)
nn(Burke-3, Mr-2)
nsubj(landing-8, it-4)
cop(landing-8, be-5)
det(landing-8, a-6)
nn(landing-8, textbook-7)
dep(Burke-3, landing-8)
partmod(landing-8, consider-9)
det(circumstance-11, the-10)
dobj(consider-9, circumstance-11)

Fig. 8. Dependency parse and triples for the sentence – "Mr Burke said it was a textbook landing considering the circumstances".

Table 1
Performance of Paraphrase Recognition system on MSRPC [25].

Features	Number of features	Accuracy %	Precision %	Recall %	*F*-Measure %
Lexical	5	73.68	73.40	94.77	82.72
Syntactic	100	73.33	74.38	91.37	82.00
Semantic	9	68.41	69.03	95.20	80.03
Lexical, POSPER, NVSim, proper noun	104	75.36	76.2	91.54	83.17
All lexical, syntactic and semantic features except cardinal number match	113	75.58	79.83	86.59	83.07
Features selected using GA-SVM method	57	76.97	80.47	88.09	84.11

which are used to evaluate the goodness of clusters have also been used here. Precision is defined as the number of true paraphrases out of the total number of sentences declared as paraphrases. It is calculated using Eq. (9).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

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 are assessed only with respect to the annotated sentence pairs.

The other metric Relative Recall (RR) has been adapted from the Information Retrieval version [34] and assesses the recall of the current system S with respect to other systems as given in Eq. (10). This measure has been used due to limited information on all the possible relevant paraphrase pairs.

$$relative\_recall_S = \frac{\text{Number of relevant pairs extracted by } S}{\text{Cumulative set of relevant pairs extracted by all systems}} \tag{10}$$

Purity, entropy and v-measure are the popular measures used to assess the quality of clustering. Purity of a cluster is the fraction of the highest number of common objects between the cluster and any one class, to the number of objects in the cluster [16]. Given that C represents the set of clusters and K the set of reference classes purity of a cluster C_i is:

$$P_i = \frac{1}{|C_i|}\max_j(|C_i \cap K_j|) \tag{11}$$

The overall purity (P) for a set of clusters is the weighted sum of the purity of the individual clusters, where N is the total number of sentences.

$$P = \frac{1}{N}\sum_{i=1}^{|C|}(|C_i| \cdot P_i) \tag{12}$$

Entropy of a cluster E_i is a measure of disorder and measures how the classes are distributed within a cluster [16]. The overall entropy E is the weighted average of the individual cluster entropies.

$$E_i = -\frac{1}{\log |K|}\sum_{j=1}^{|K|}\frac{|C_i \cap K_j|}{K_j}\log\frac{|C_i \cap K_j|}{K_j} \tag{13}$$

$$E = \frac{1}{N}\sum_{i=1}^{|C|}(|C_i| \cdot E_i) \tag{14}$$

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. It is defined as the harmonic mean of homogeneity h and completeness c which are calculated in terms of entropies $H(K)$, $H(C)$ and conditional entropies $H(K|C)$ and $H(C|K)$. By varying β, the preference given to h and c can be controlled; if both are equally important, β is set to 1 [35].

$$V = \frac{(\beta + 1)hc}{(\beta h + c)} \tag{15}$$

$$h = 1 - \frac{H(K|C)}{H(K)} \tag{16}$$

$$c = 1 - \frac{H(C|K)}{H(C)} \tag{17}$$

Besides the above measures, the Partition Coefficient (PC) has been used specifically with reference to fuzzy clustering. The partition coefficient measure given in Eq. (18) quantifies the fuzziness of a partition with higher values which indicate least fuzzy clustering, being preferable.

$$PC = \frac{1}{N}\sum_{i=1}^{|C|}\sum_{j=1}^{N}\mu_{ij}^2 \tag{18}$$

In Eq. (18), N represents the total number of objects and μ_{ij} is the membership value of object j in cluster i.

4.2. Existing models

The proposed system has been evaluated against the techniques used by Wubben et al. for Paraphrase Acquisition from Newspaper headlines [9]. Wubben et al.'s system has been chosen for comparative evaluation due to reason that the focus of the work is very much similar to the proposed one. In [9], originally available Google News headlines have first been re-clustered into finer sub-clusters by using *k*-means algorithm from CLUTO software package [36]. The PK1 Cluster Stopping Criterion specified by Pederson and Kulkarni (PK) [37] has been used to identify the '*k*' value. After clustering, all sentences within a cluster are aligned pair-wise. Wubben et al. have used a second approach, wherein all sentences in the originally available Google News clusters are directly matched pair-wise and the cosine similarity score is computed. If the similarity exceeds the upper threshold, the pair was accepted as paraphrases; if similarity was less than a lower threshold, the pair was rejected. In case the similarity was between thresholds, the context in which the headline occurred was used to decide. The cosine similarity approach has been used by Wubben as a baseline.

For the current performance evaluation, both systems proposed by Wubben have been used with modifications. In *k*-means clustering approach, PK3 cluster stopping criteria has been used in place of PK1, as PK3 has been reported to be more efficient than PK1 [38]. Since no clusters exist initially, *k*-means clustering has straightaway been applied on the entire corpus. In the pair-wise cosine similarity computation, due to 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. 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 using the R package. The optimal number of clusters has been identified by choosing the partition which yields highest partition coefficient and lowest classification entropy.

4.3. Experiments on MSRPC

The MSRPC consists of 5801 pairs of sentences and is best suited for evaluating the performance of Paraphrase Recognition systems. It consists of 3900 positive cases of paraphrases and 1901 negative cases [39]. 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 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 proposed system, the sentences were subject to Fuzzy Agglomerative Clustering followed by divisive clustering, both with and without WSD. As described in Sections 3.2 and 3.3, 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 the two schemes: all sentences and 50% of sentences from each cluster based on the membership values. Using these choices, eight variants of the proposed system are framed, 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 3.4) 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 3.5 has been omitted.

Table 2
Statistics of paraphrase pairs retrieved from MSRPC.

System	Number of known pairs	TP	FP
k-Means clustering (Wubben et al. [9])	3926	2770	1156
Cosine similarity (Wubben et al.)			
$T = 0.3$	5747	3891	1856
$T = 0.4$	5516	3837	1679
$T = 0.5$	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

Eq. (9) 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 are counted as True Positives whereas the non-paraphrased pairs retrieved are considered as False Positives. Table 2 presents the performance of the existing system as well as the proposed system variants in terms of pairs with a known decision, True Positives and False Positives. The system which involves direct computation of cosine similarity was tested with five different thresholds starting from 0.3 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 proposed system variants.

The cosine similarity based system with low thresholds is 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 retrieve more candidates due to the tighter clusters based on specific word senses. The precision of the existing approaches: cosine similarity, *k*-means, FCM Clustering as well as all the proposed system variants have been presented in Table 3.

The proposed system outscores the existing system variants 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.

From Table 3 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 proposed 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.

Table 3
Precision of existing and proposed approaches.

Proposed system variants		Precision %	Existing system variants	Precision %
No WSD				
Threshold 0.15	Top 50%	80.35	*k*-Means clustering (Wubben et al. [9])	**70.55**
	All	80.01	FCM clustering	**68.13**
Threshold 0.2	Top 50%	80.34	Cosine similarity (Wubben et al. [9])	
	All	80.05	Threshold = 0.3	67.70
WSD				
Threshold 0.15	Top 50%	80.96	Threshold = 0.4	69.56
	All	80.91	Threshold = 0.5	73.09
Threshold 0.2	Top 50%	**81.02**	Threshold = 0.6	77.93
	All	80.96	**Threshold = 0.7**	**80.46**

The results of the four systems which are considered for further evaluation (Fig. 9) have been given in bold.

Table 4
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

The 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 [9].
- Cosine similarity variant [9] using threshold = 0.7.
- FCM Clustering approach.

The relative recall computation has also been carried out according to Eq. (10) with respect to known relevant pairs or True Positives (Table 2) retrieved by the systems. The results are presented in Table 4 and the *k*-means approach exhibits the highest relative recall. This is followed by the proposed system 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 for containing the relevant pairs.

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.

Since the objective of the proposed system is to extract paraphrases from large corpora, an unsupervised clustering approach has been used to first cluster the sentences. In the second stage a supervised approach has been used to filter the paraphrases from within each cluster. Prior work using MSRPC have either relied on a purely supervised approach or in the case of unsupervised learning, have computed scores between each specific sentence pair only. In contrast this work considers all the sentences within the MSRPC corpus for determining paraphrases. Further, when the unsupervised approach is used a large number of sentence pairs are reported as paraphrases but the output decision is available only for 5801 pairs. Hence the evaluation results presented in Tables 3 and 4 have been computed using only the known pairs.

Due to the above reasons a direct comparison of the results with that of previous work on MSRPC would definitely be biased with the unsupervised approaches which use specific pair-wise comparison approaches having an advantage. However to identify the performance level of the proposed system, a brief comparison with prior work using unsupervised approaches on MSRPC has been presented in Table 5. It can be seen that the proposed system gives good results only with respect to precision. The results can be attributed to the fact that the previous unsupervised approaches – Fernando et al. [40] and Mihalcea et al. [41] judge only the given sentence pair and focus on Paraphrase Recognition only. However the proposed system undertakes the responsibility of first identifying possible candidates through clustering and then judging them, which would be the requirement of a Paraphrase Extraction system.

However, the proposed system performs better than Wubben et al.'s approaches as well as FCM Clustering for Paraphrase Extraction. Therefore, it can be concluded that the proposed system is suitable for the task of Paraphrase Extraction.

4.4. Experiments on MSRVDC

The MSRVDC was constructed by Chen and Dolan [42] and consists of 120,000 sentences collected from multi-lingual descriptions of short video snippets supplied by workers on Mechanical Turk. Out of the 85,000 English sentences, 33,855 were from Tier-2 workers who had a higher rating. For evaluating the performance of the proposed system, two different datasets were constructed from MSRVDC.

The first dataset consists of 2007 sentences extracted from the 35 K 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. 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 chosen and 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 proposed system, the existing systems (Wubben et al.) 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 3.5 to generate clusters of paraphrases. As in the case of MSRPC, various implementations of the proposed system 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 6. The 143 clusters identified were fixed as reference classes against which the created clusters have been judged.

In terms of entropy, the best performance was registered by the rigorous system, with WSD, threshold 0.2 and using only the

Table 5
Comparative performance on MSRPC.

Approach	Accuracy %	Precision %	Recall %	*F*-Measure %
Fernando et al. (2008) [40]	74.1	75.2	91.3	82.4
Mihalcea et al. (2006) [41]	70.3	69.6	97.7	81.3
Proposed system variant WSD, threshold = 0.2, top 50%	66.4	81.0	65.2	72.3
Cosine similarity, threshold = 0.7 (Wubben et al. [9])	62.5	80.5	56.6	66.5
k-Means clustering (Wubben et al. [9])	60.6	70.6	71.0	70.8
FCM clustering	36.2	68.1	9.5	16.7

Table 6
Performance of proposed system 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 %	5.68	10.98	5.55	10.9	5.06	6.88	**4.75**	6.86
Purity %	64.82	**79.42**	65.62	78.77	63.08	70.8	61.34	70.05
ν-Measure %	83.87	83.05	84.04	82.97	83.98	84.14	84.25	**84.3**

The highest value for each evaluation measure has been given in bold.

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. With respect to the ν-measure, the variant with WSD and threshold of 0.2 but considering all sentences has registered the best performance of 84.3%. Using a higher threshold with WSD improves the homogeneity, whereas including all sentences within a cluster increases the completeness. Since the rigorous system has very close ν-measure, it is chosen as the best out of the eight variants as it has moderate performance with respect to purity and low entropy and high ν-measure.

Table 7 records the performance of the existing systems on MSRVDC Dataset1. It can be observed that similar to the results obtained on MSRPC, the cosine similarity method with threshold = 0.7, achieves better performance than *k*-means and FCM clustering techniques.

Comparing proposed and existing systems, it can be seen that almost all the eight variants of the proposed 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 the proposed system have been evaluated against Wubben's Clustering approach, cosine similarity system with threshold = 0.7 and FCM approach as shown in Table 8.

The rigorous variant of the Proposed system, performs much better than the other systems in terms of entropy as well as ν-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 is chosen as the better of the two proposed system variants.

Sample clusters produced by the various systems have been shown in Table 9. The clusters produced by the proposed system and cosine similarity approach tend to be more homogeneous. It can be observed that in the cluster produced by the proposed 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 9.

5. Discussion

The proposed system has consistently exhibited better performance with respect to all the three datasets as shown in Fig. 9. The rigorous variant has been chosen for comparison against the existing systems. Of the four systems, the two stage fuzzy clustering followed by Paraphrase Recognition approach 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.

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.

Table 7
Performance of existing approaches on MSRVDC Dataset 1.

	k-Means clustering [9]	FCM clustering	Cosine similarity approach [9] with varying thresholds (*T*)				
			T = 0.3	*T* = 0.4	*T* = 0.5	*T* = 0.6	*T* = 0.7
Entropy %	19.68	32.6	99.56	99.58	93.48	63.86	**10.27**
Purity %	39.01	47.73	1.69	3.34	11.26	33.58	**54.95**
v-Measure %	80.03	67.33	0.17	0.15	11.36	33.89	**84.03**

The highest value for each evaluation measure has been given in bold.

Table 8
Performance evaluation on MSRVDC Dataset 2.

	Without WSD Threshold = 0.15, all	With WSD threshold = 0.2, top 50%	*k*-Means clustering	Cosine similarity Threshold = 0.7	FCM clustering
Entropy %	34.88	**19.41**	68.03	66.91	86.78
Purity %	**44.07**	41.30	2.42	6.97	0.78
v-Measure %	68.95	**77.09**	42.67	43.24	19.56
Partition coefficient	0.78	**0.98**	–	–	0.33

The highest value for each evaluation measure has been given in bold.

Table 9
Sample clusters on MSRVDC Dataset 2.

Proposed system clustering (sample clusterpurity = 0.82, entropy = 0.08)	*k*-Means clustering (sample cluster purity = 0.11 (highest), entropy = 0.37 (lowest))	Cosine sample cluster 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 read 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 read 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

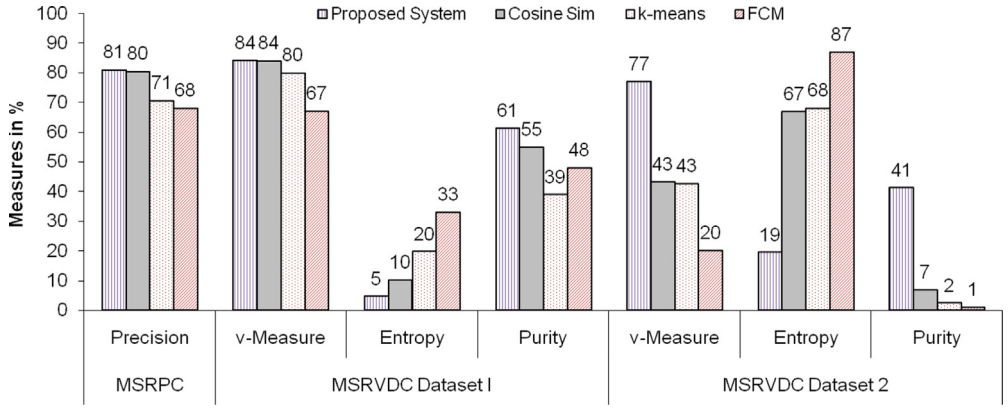


Fig. 9. Performance evaluation of proposed and existing systems.

- The 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.

6. Conclusions

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: usage of a soft hierarchical clustering 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 is found to improve the performance. In future, the performance of the two-stage approach on larger corpora can be probed. The supervised classifier used in the second stage for Paraphrase Recognition can be replaced by suitable paraphrase evaluation metrics enabling the entire process to work in an unsupervised fashion. The approach can also be adapted for tasks such as tweet clustering, plagiarism detection and multi-document summarization.

References

[1] I. Androutsopoulos, P. Malakasiotis, A survey of paraphrasing and textual entailment methods, J. Artif. Intell. Res. 38 (2010) 135–187.
[2] R. Bhagat, D. Ravichandran, Large scale acquisition of paraphrases for learning surface patterns, in: Proc. of the 46th Annual Meeting of ACL: HLT, Columbus, USA, 2008, pp. 674–682.
[3] D. Metzler, E. Hovy, Mavuno: a scalable and effective Hadoop-based paraphrase acquisition system, in: Proc. of the Third Workshop on Large Scale Data Mining: Theory and Applications, San Diego, USA, August, 2011.
[4] I. Szpektor, H. Tanev, I. Dagan, B. Coppola, Scaling web-based acquisition of entailment relations, in: Proc. of the Conf. on EMNLP, Barcelona, Spain, 2004, pp. 41–48.
[5] M. Regneri, R. Wang, M. Pinkal, Aligning predicate-argument structures for paraphrase fragment extraction, in: Proc. of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May, 2014, pp. 4300–4307.
[6] R. Barzilay, L. Lee, Learning to paraphrase: an unsupervised approach using multiple-sequence alignment, in: Proc. of the Human Language Technology Conf. of NAACL, Edmonton, Canada, 2003, pp. 16–23.
[7] M. Regneri, R. Wang, Using discourse information for paraphrase extraction, in: Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 2012, pp. 916–927.
[8] Y. Yan, C. Hashimoto, K. Torisawa, T. Kawai, J. Kazama, S. De Saeger, Minimally supervised method for multilingual paraphrase extraction from definition sentences on the web, in: Proc. of NAACL-HLT 2013, Atlanta, Georgia, June, 2013, pp. 63–73.
[9] S. Wubben, A. van Den Bosch, E. Krahmer, E. Marsi, Clustering and matching headlines for automatic paraphrase acquisition, in: Proc. of the 12th European Workshop on Natural Language Generation, Athens, Greece, March, 2009, pp. 122–125.
[10] M. Bank, F. Schwenker, Fuzzification of agglomerative hierarchical crisp clustering algorithms, in: Proc. of the 34th Annual Conf. of the GFKI, Karlsruhe, July, 2010, pp. 3–11.
[11] P. Rodrigues, J. Gama, Semi-fuzzy splitting in online divisive-agglomerative clustering Lecture Notes in Computer Science, vol. 4874, Springer, Berlin, 2007, pp. 133–144.
[12] I. Diaz-Valenzuela, M.J. Martin-Bautista, M.A. Vila, A fuzzy semisupervised clustering method: application to the classification of scientific publications Information Processing and Management of Uncertainty in Knowledge-Based System, vol. 442, Springer International Publishing, 2014, pp. 179–188.
[13] E. Rossi Marques Seno, M. das Graças Volpe Nunes, Some experiments on clustering similar sentences of texts in Portuguese Lecture Notes in Computer Science, vol. 5190, Springer, Berlin, 2008, pp. 133–142.
[14] R. Khoury, Sentence clustering using parts-of-speech, Int. J. Inf. Eng. Electron. Bus. 1 (2012) 1–9.
[15] M. Rodrigues, L. Sacks, A scalable hierarchical fuzzy clustering algorithm for text mining, in: Proc. of the 5th Int. Conf. on Recent Advances in Soft Computing, December 2004, pp. 269–274.
[16] A. Skabar, K. Abdalgader, Clustering sentence-level text using a novel fuzzy relational clustering algorithm, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 62–75.
[17] S.V. Wazarkar, A.A. Manjrekar, Text clustering using HFRECCA and rough K-means clustering algorithm, in: Proc. of International Conference on Advances in Computer Engineering & Applications, vol. 15 (40), 2014, pp. 44–47.
[18] H. Schmid, Probabilistic part-of-speech tagging using decision trees, in: Proc. of International Conference on New Methods in Language Processing, Manchester, UK, September, 1994, pp. 44–49.
[19] M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of English: the Penn Treebank, J. Comput. Linguist. 19 (2) (1993) 313–330.
[20] R. Navigli, Word sense disambiguation: a survey, ACM Comput. Surv. 1 (2) (2009) 1–69.
[21] F. Vasilescu, P. Langlais, G. Lapalme, Evaluating variants of the Lesk approach for disambiguating words, in: Proc. LREC 2004, the 4th International Conference on Language Resources and Evaluation, 2004, pp. 633–636.
[22] J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proc. International Conference on Research in Computational Linguistics, Taiwan, September, 1997, pp. 19–33.
[23] D. Yang, D. Powers, Verb pair similarity scores (Gold Standard 130 verb pairs), Flinders University, 2013 http://hdl.handle.net/2328.1/1112
[24] D. Bollegala, Y. Matsuo, M. Ishizuka, A web search engine-based approach to measure semantic similarity between words, IEEE Trans. Knowl. Data Eng. 23 (July (7)) (2011) 977–990.
[25] A. Chitra, A. Rajkumar, Genetic algorithm based feature selection for paraphrase recognition, Int. J. Artif. Intell. Tools 22 (2) (2013) 1350007, 1–17.
[26] J. Cordeiro, G. Dias, P. Brazdil, New functions for unsupervised asymmetrical paraphrase detection, J. Softw. 2 (4) (2007) 12–23.
[27] D. Guthrie, B. Allison, W. Liu, L. Guthrie, Y. Wilks, A closer look at skip-gram modeling, in: Proc. of the Fifth International Conference on Language Resources and Evaluation, 2005, pp. 1222–1225.
[28] D. Klein, C.D. Manning, Accurate unlexicalized parsing, in: Proc. 41st Meeting of ACL, 2003, pp. 423–430.
[29] P. Bille, A survey on tree edit distance and related problems, J. Theor. Comput. Sci. 337 (1–3) (2005) 217–239.
[30] S. Wan, M. Dras, R. Dale, C. Paris, Using dependency-based features to take the parafarce out of paraphrase, in: Proc. Australasian Language Technology Workshop, Sydney, Australia, 2006, pp. 131–138.
[31] A. Finch, Y.S. Hwang, E. Sumita, Using machine translation evaluation techniques to determine sentence-level semantic equivalence, in: Proc. 3rd International Workshop on Paraphrasing, Jeju Island, Korea, October, 2005, pp. 17–24.
[32] Z. Kozareva, A. Montoyo, Paraphrase identification on the basis of supervised machine learning techniques, in: Proc. Advances in Natural Language Processing: 5th International Conference on NLP, Turku, Finland, August, 2006, pp. 524–533.
[33] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27, Software available at http://www.csie.ntu.edu.tw/(cjlin/libsvm
[34] M. Agosti, M. Melucci, Information retrieval on the web, in: M. Agosti, F. Crestani, G. Pasi (Eds.), Lectures on Information Retrieval: Third European Summer-School, Springer, 2000, pp. 242–285.
[35] J. Geiss, Latent Semantic Sentence Clustering for Multi-Document Summarization (Ph.D. thesis), University of Cambridge, Cambridge, 2011.
[36] CLUTO. A Clustering Toolkit, http://www.cs.umn.edu/(karypis/cluto
[37] T. Pedersen, A. Kulkarni, Automatic cluster stopping with criterion functions and the gap statistic, in: Proc. of the Human Language Technology Conference of the NAACL, New York City, June, 2006, pp. 276–279.
[38] A. Kulkarni, Unsupervised context discrimination and automatic cluster stopping (M.Sc. thesis), University of Minnesota, Minnesota, July 2006.
[39] B. Dolan, C. Quirk, C. Brockett, Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources, in: Proc. 20th Int. Conf. on Comp. Linguistics, Geneva, Switzerland, 2004, pp. 350–356.
[40] S. Fernando, M. Stevenson, A semantic similarity approach to paraphrase detection, in: Proc. of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics, Oxford, England, 2008, pp. 45–52.
[41] R. Mihalcea, C. Corley, C. Strapparava, Corpus based and knowledge-based measures of text semantic similarity, in: Proc. of 21st Conference of American Association for Artificial Intelligence, Boston, 2006, pp. 775–780.
[42] D.L. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: Proc. of the 49th Annual Meeting of ACL, Portland, USA, June, 2011, pp. 190–200.