CHAPTER 3

PARAPHRASE RECOGNITION USING INTERMEDIATE REPRESENTATIONS

The establishment of Semantic Similarity between text units is a pivotal task in several applications such as Information Extraction, Question Answering and Summarization. A logical solution for establishing semantic similarity would be to translate the input text units to an intermediate representation or Interlingua. Similarity measures can then be computed on these representations using additional resources such as WordNet. Such interlingua based approaches can also be utilized for determining if two sentences from different languages are similar.

Two different methods based on intermediate representations have been proposed for sentence level Paraphrase recognition. A scheme applicable for cross language inputs has been investigated, wherein the input sentences have been converted to intermediate representations in Universal Networking Language (UNL). Similarity measures were extracted from the UNL forms and a Support Vector Machine (SVM) classifier was used for Paraphrase Recognition. This approach is termed as UNLPR. The second approach has been designed to work on Predicate Argument Structure (PAS) representations of the input sentences and is termed as PASPR. In the initial Predicate Argument (PA) matching stage, the arguments of PA tuples which share the same or similar verb have been compared with each other to identify the equivalent parts in the input sentence pair. In the subsequent classification stage, features such as named entities match, word overlap and other features

were extracted from the PA structure representation and an SVM classifier was used to label the input pair as paraphrases or non-paraphrases.

3.1 UNL BASED PARAPHRASE RECOGNITION

The UNLPR approach based on Universal Networking Language (UNL) representation has been proposed for determining whether two input sentences are semantically similar. The advantage of the UNLPR system is that it can be used to establish cross-language similarity. Universal Networking Language has been proposed by United Nations University with the objective of developing universally usable computer interfaces. UNL represents the meaning of a sentence in the form of a semantic network with hyper-nodes. In the UNL semantic network, nodes represent concepts, and arcs represent relations between concepts (UNL Center, 2003). The three basic components in the UNL representation are: Universal Words, Relations and Attributes. Universal words are English words used to represent simple or compound concepts. Relations indicate the relationship between two universal words while attributes provide additional information about the Universal words. The process of converting a sentence given in natural language into UNL representation is termed as enconversion while the reverse process is termed as deconversion. The UNL representation of the sentence “Charles Babbage is considered as the father of computers.” is shown in Figure 3.1.

```
nam:01(charles(icl>name>abstract_thing,com>male,nam<person).@entry.@topic,babba
ge)
aoj:01(charles(icl>name>abstract_thing,com>male,nam<person).@entry.@topic,as(icl>
how,com>class,obj>thing,aoj<uw))
obj:01(as(icl>how,com>class,obj>thing,aoj<uw),father(icl>parent>living_thing,ant>mot
her,pos>child).@def)
pos:01(father(icl>parent>living_thing,ant>mother,pos>child).@def,computer(icl>machi
ne>thing).@pl)
obj(consider(icl>regard>be,cob>uw,obj>uw,aoj>person).@entry.@present,:01)
```

Figure 3.1 UNL Representation of sample sentence

Online servers which perform conversion and deconversion are available. One such server is the UNL Explorer which offers multi-lingual search, dictionary, UNL Ontology and UNL Talk services apart from conversion (Uchida et al 2012). Other popular servers are the Russian-UNL server and the Indian Institute of Technology Bombay-Center for Indian Language Technology (IITB-CFILT) UNL server. As the UNL representations for sentences in different languages are similar, UNL can be used as an intermediate language for finding semantic similarity between the sentences. A scheme which employs a machine learning classifier has been proposed for UNL matching. Various features extracted from the UNL forms of input sentences have been used to identify whether the two text units are semantically similar. The effects of Word Sense Disambiguation and Co-reference Resolution on UNL Matching have also been investigated. Figure 3.2 shows the stages of the proposed UNL matching system.

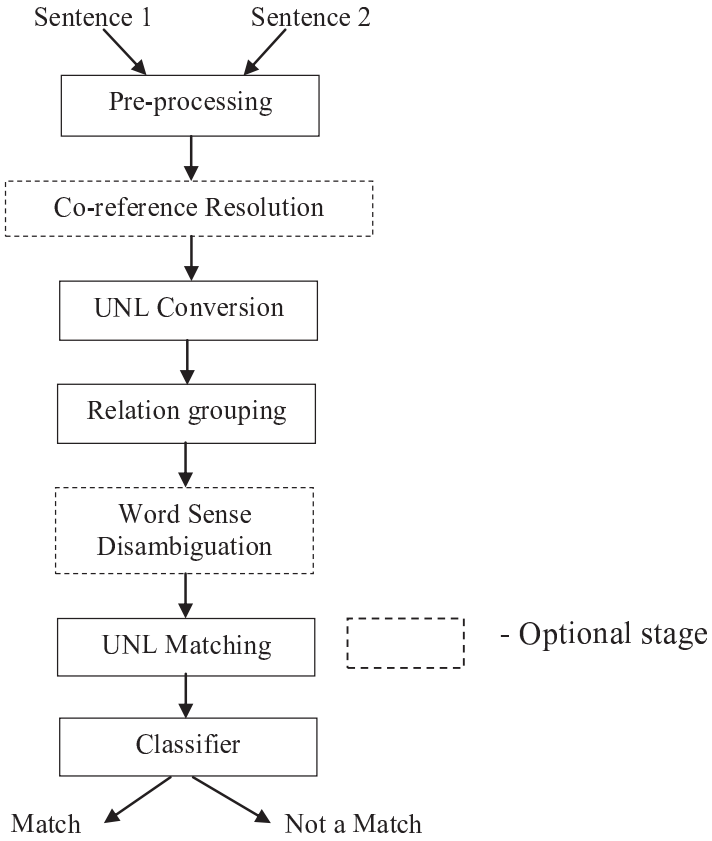

Figure 3.2 Stages of UNL based Paraphrase Recognizer

The input sentences were subjected to pre-processing to eliminate punctuations and word contractions. Co-reference resolution was applied to replace pronouns in the input sentence with their corresponding nouns. This step has been applied before UNL conversion to ensure that co-referring nouns and pronouns within a sentence map to the same Universal words. The sentences were then converted to UNL form. To facilitate the matching of UNL forms, the UNL relations which have distinct labels but are similar in terms of the underlying concept were grouped together. Further, for the Universal words corresponding to nouns and verbs, Word Sense Disambiguation (WSD) was applied to determine the specific sense in which a word is used. As this step requires the input sentence to be matched against the various senses of a word extracted from WordNet, it would not be feasible for cross-language inputs which do not have Wordnet like dictionaries. The UNL forms of the input sentences were then compared to extract various features which reflect the degree of similarity between the UNL forms. Since assessing semantic equivalence by using simple rules and thresholds is tedious as well as inaccurate, a machine learning classifier has been used. The extracted features were used by the classifier to detect the semantic similarity of the input sentences.

3.1.1 Pre-Processing and Co-Reference Resolution

During Pre-processing, punctuations were removed and contractions such as “aren’t”, “didn’t” were expanded. This was followed by Co-reference resolution which is the process of identifying all expressions that refer to the same entity. For example in the sentence “Ram went to the shop and he bought a book”, the pronoun ‘he’ refers to ‘Ram’. The Stanford Co-reference Resolution System (Lee et al 2011) has been used to resolve co-references in the input sentences.

3.1.2 UNL Conversion and Processing

UNL Enconversion has been carried out using the Russian UNL Converter (UNL Russian Module Server). The Russian UNL convertor is an online language server which converts English and Russian language sentences into UNL form. Due to its consistent availability and ability to handle longer sentences, the Russian Converter has been used in this work. Relations in UNL are also referred as links and indicate the relationship between the universal words. There are 39 distinct relation labels in UNL, with some of them being related. For instance agent('agt') refers to the initiator of an action on whom the sentence focuses whereas partner('ptn') indicates a non-focused initiator (UNL Center 2003). In this work, related UNL relations have been grouped together to facilitate better matching similar to Singh et al (2012). Sample relation groups have been shown in Table 3.1. All relations within the same group have been considered as approximate matches.

Table 3.1 Groups of related UNL relations

Time-related	Place-related	Event related
tim – time	plc – place	src – initial state
tif – initial time	plf – initial place	gol – final state
tmt – final time	plt – final place	via – intermediate state

3.1.3 WSD and Similarity Calculation

Word Sense Disambiguation is the process of determining the exact sense of a word based on its context. Though the Constraint list attached to a Universal Word aims to restrict the sense of the word, the same word may have different meanings in different contexts. The simplified Lesk algorithm

(Vasilescu et al 2004) has been used here. 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 highest 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 current work, the effect of WSD on UNL matching has also been studied.

In cases where the Universal Words do not match exactly, similarity between the words has been computed using WordNet. 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 ranges between 0 and 1, with the value being closer to 1 as the similarity increases. The score for the words “tree” and “plant” is 0.75 whereas for the pair “flower” and “bird” the score is 0.09. Word pairs with a score greater than 0.15 have been taken to be similar based on the scores obtained for word pairs classified as similar in Yang & Powers dataset (Yang & Powers 2013) and Miller Charles dataset (Bollegala et al 2011).

3.1.4 UNL Matching

In order to assess semantic similarity, various features were computed from the UNL forms of both the sentences by matching the Relations, Universal Words and UNL attributes. Additionally the Named Entities in both sentences have been identified using the Stanford Named Entity Recognizer (Finkel et al 2005) and compared to generate a pair of features. All the features proposed are applicable for cross-language inputs. WSD has been provided as an additional option which can be disregarded in the case of cross-language inputs. In the case of WSD option, when two words do not match exactly, the following steps were performed:

- the specific sense of each Universal word was determined by applying Word Sense Disambiguation
- the synsets corresponding to the specific senses were obtained
- If the synsets overlapped, the Universal words were considered to match semantically

The features extracted from UNL forms and the rules for their computation are listed in Table 3.2 where S1, S2 are the input sentences, R1, R2 represent relations and UW1 to UW4 are the four universal words from R1, R2.

Table 3.2 UNL Matching Features

Feature	Rules for Computation
Simple Relation Precision	Number of common relations between S1 and S2 divided by the number of relations in S1, S2 respectively.
Simple Relation Recall	
UW Precision	Number of matching Universal word pairs between S1 and S2 divided by Universal word pairs in S1, S2.
UW Recall	
Overall Relation Precision	If R1=R2 or if both are in the same group, the pairs UW1, UW3 and UW2, UW4 are compared. If they match exactly or if they are similar R1 and R2 are said to match. The Number of matching relations is divided by the number of relations in S1 and S2 respectively.
Overall Relation Recall	
Named Entity Precision and Recall	Number of matching named entities divided by the number of named entities in S1, S2 respectively.

A Support Vector Machine (SVM) Classifier has been used to classify the sentences as positive or negative cases of paraphrases using the features extracted from the input sentence pair. The LibSVM tool (Chang & Lin 2011) has been employed to implement a nu-Classifier with a radial basis function kernel.

3.2 PARAPHRASE RECOGNITION USING PA MATCHING

Predicate Argument (PA) representations of a sentence indicate the various semantic roles in a sentence. PA structures help to clearly convey the meaning of the sentence by identifying each predicate or verb and each of its arguments and their corresponding roles. The PA representation of the sentence "John hit Jack" is given below:

Relation / predicate	:	hit
ARG0	:	John
ARG1	:	Jack

where ARG0 represents the agent and ARG1 indicates the patient. In this work Predicate Argument matching approach has been used for recognizing sentential paraphrases. PA alignment is more relevant than surface level matching schemes as in the case of the sentences "John hit Jack and "Jack hit John" which have complete word overlap but convey two opposite actions.

Previous approaches based on predicate alignment for the RTE task or paraphrase recognition have relied on score computation or a supervised approach. Wang & Zhang (2009) have proposed a Textual Relatedness score for recognizing entailment. Hickl et al (2006) and Rios & Gelbukh (2012) have both used a supervised learning approach on features computed from the PA structures for the RTE task. In this work - PASPR, similar to the approach employed by Qiu et al (2006), a two-stage approach has been used for Paraphrase Recognition. The PA matching stage which is an unsupervised one focuses on pairing PA tuples, whereas the Classification stage operates on features extracted from the PA representations of the input sentences. The PASPR system differs from Qiu et al's work, in the strategies used for PA tuple pairing as well as the classification methodology and features used.

Initially, both the sentences in the input pair have been converted into Predicate Argument representation using a Semantic Role Labeling tool. In the PA matching stage, the PA tuples were matched by locating same or similar predicates/verbs and then matching their corresponding arguments. Similar to the approach proposed by Yadav et al (2012), the extent of similarity between the matched tuples was used to classify them as equivalent or paired, more or less equivalent or loosely paired and not-equivalent or unpaired. In the Classification stage, a supervised learning strategy has been used to classify the sentence pair based on features extracted from the PA representation. A divide and conquer approach has been used to partition the inputs into disjoint subsets for which prior approaches have used either random partitioning or Clustering techniques (Chawla et al 2001). The pseudo-code for PASPR approach has been given in Figure 3.3.

Algorithm for Paraphrase Recognition using PA matching

- Represent each sentence of the input pair using Predicate Argument Structures

Stage 1 – PA matching

- Extract the verbs present in each sentence.
- for each verb V_i in Sentence 1 and verb V_j in Sentence 2
  - Compute the similarity between the verbs
  - If verbs are similar but neither one is negated:
    - Identify the common arguments for V_i and V_j
    - Match equivalent arguments to generate the argument scores for A0, A1, A2 and Aothers (all other argument types except AM_NEG)
  - else if V_i and V_j are antonyms
    - Match argument A0 of Sentence 1 with A1 of Sentence 2 and vice versa
    - Match all other equivalent arguments
  - Generate a consolidated score $Score_{ij}$ by combining verb similarity and all the argument scores
  - Set similarity matrix entry to $Score_{ij}$
- For each V_i, determine the most similar verb V_{max} with score S_{max} from Sentence 2 using the Similarity matrix
  - If $S_{max} > 0.5$ the verbs V_i and V_{max} are considered as paired
  - Else if $0.25 < S_{max} < 0.5$ V_i and V_{max} are loosely paired
  - Else V_i and V_{max} remain unpaired

Stage 2 – Classification of Sentence pairs

- Segregate the sentences into various categories based on the presence of paired, loosely paired and unpaired tuples
- Extract features from the sentences and build separate classification models for each category
- Use the relevant classification model to determine if the sentences are paraphrases

Figure 3.3 Algorithm for Paraphrase Recognition using PA Matching

The novel aspect of this work is that after pairing the PA tuples, the sentence pairs have been segregated into various categories based on the extent of paired, loosely paired and unpaired tuples. This is a variation of the directed diversity approaches employed by ˇZliobait˙e (2011) which rely on either a slicing feature or distance based clustering for partitioning the inputs. In PASPR, for each category, separate classification models have been constructed using different features extracted from sentence pairs. This approach has been proposed in order to handle the disparities in the nature of paraphrases. In some cases though all PA tuples in the input sentences are paired the sentences turn out to be non-paraphrases. Therefore additional features based on word overlap, named entity matching and presence of cue words indicating negation or alternation have been included in the second stage.

3.2.1 Predicate Argument Representations

The first step involves the conversion of the input sentence pairs to Predicate Argument representation. For this purpose, the Semantic/syntactic Extraction using a Neural Network Architecture (SENNA) parser developed by Collobert et al (2011) has been used. SENNA uses neural networks for POS tagging, chunking, named entity recognition and Semantic Role Labelling (SRL) and has been shown to exhibit competitive performance and produce quick results. The Penn Treebank dataset has been used for tagging, and the Noun Phrase (NP) and Verb Phrase (VP) chunks have been identified. Three major categories of Named Entities – Person, Organization and Location are identified by SENNA. In the SRL task, the IOB / IOBES (Inside Other Begin End Single) formats are used in association with the Propbank annotation guidelines for arguments A0-A5 and other modifying arguments (AM-MOD). In this work, the output produced by the SENNA parser has

been processed to identify the phrases and predicate argument tuples in the input sentences.

A sample output produced by the SENNA parser for the sentence "Charles Babbage is considered as the father of computers." is given in Table 3.3 where PSG refers to Phrase Structure Grammar.

Table 3.3 Sample output of SENNA parser

Word	POS Tag	Chunk	Named Entity	Verb 1	Arguments	PSG format
Charles	NNP	B-NP	B-PER	-	B-A1	(S1(S(NP*
Babbage	NNP	E-NP	E-PER	-	E-A1	*)
is	VBZ	B-VP	O	-	O	(VP*
considered	VBN	E-VP	O	considered	S-V	(VP*
as	IN	S-PP	O	-	B-A2	(PP*
the	DT	B-NP	O	-	I-A2	(NP(NP*
father	NN	E-NP	O	-	I-A2	*)
of	IN	S-PP	O	-	I-A2	(PP*
computers	NNS	S-NP	O	-	E-A2	(NP*))))))
.	.	O	O	-	O	*))

3.2.2 Predicate Argument Structure Matching

In order to pair the Predicate Argument tuples in the input sentences, a three step process has been adopted. In the first step, the similarity between the verbs of the two sentences was computed to identify which PA tuples have to be compared. In the second step, the corresponding arguments of the PA structures were matched and a consolidated score was calculated for each PA tuple pair. Finally a pairing of the tuples was carried out based on the scores.

Verb Matching

The verbs in the two sentences have been matched by first checking if the verbs are identical or antonyms. Otherwise the similarity was computed by considering the distance between the verbs and also the WordNet synsets of the candidate verbs. Different scores have been assigned depending on the extent of similarity between the verbs. In case of exact match the highest score of 1 was assigned. The next level score of 0.8 was assigned if one of the verbs was present in the synset of the other and a still lower score of 0.6 was assigned when there was at least a single common word between the synsets. The scores of 0.8 and 0.6 were fixed after experimenting with various values between 1.0 and 0.1. The distance between the two words in the Wordnet hierarchy has also been considered. Two verbs were considered similar if their similarity score exceeds the threshold value of 0.15 identified based on similarity scores of similar word pairs in the Yang and Powers dataset (Yang & Powers 2013). The algorithm used for Verb matching is given in Figure 3.4.

Algorithm for Verb Matching

Input: Verbs V1 and V2
Output: Verb Similarity score

- If V1 = V2 Similarity score VSim = 1
- else
    - Stem the verbs to obtain VS1 and VS2
    - If VS1 and VS2 are Antonyms VSim = -1
    - Else
        - VSim1 = 1 – WordNet distance between VS1 and VS2
        - If synsets of VS1 contain VS2 or vice versa VSim2 = 0.8
        - else if synsets of VS1 and VS2 overlap VSim2 = 0.6
        - VSim = Max(VSim1, VSim2)

Figure 3.4 Algorithm for matching verbs

Argument Matching

The arguments of similar verbs have been matched to generate a consolidated score by combining the verb score as well as individual argument scores, similar to the strategy employed by Andreevskaia et al (2006) in their work on determining entailment. Two arguments were considered to match when there was considerable word overlap between them or there was high degree of similarity between the words. As proposed by Wu et al (2010) in their work on detecting cross-language similarity, ARG0, ARG1 and ARG2 categories were given higher weightage and generated individual scores. All other argument categories were grouped together to generate a single score.

In the matching process, the common argument categories were first detected. In the default case, matching was carried out strictly between corresponding arguments only. The only exception to this general matching strategy was that in cases where ARG1 was missing in one sentence but present in the other; ARG2 of the first sentence was matched against ARG1 of the second sentence. One such case is shown in Table 3.4.

Table 3.4 Example for matching arguments of different categories

Sentence	The case puts the Supreme Court back into the debate over the separation of church and state.	The case marks the court's second major separation of church and state case in two years.
Verb	puts	marks
ARG0	The case	The case
ARG1	-	the court's second major separation of church and state case in two years.
ARG2	Supreme Court back into the debate over the separation of church and state	-

If two verbs were found to be similar but either one has been negated, indicated by the presence of the AM-NEG argument, further matching was not carried out and the consolidated similarity score between the PAs was set to 0. This is in line with the technique used by Snow et al (2006) in their work on recognizing false entailment. For instance the following sentences are not paraphrases despite the high word overlap, because the verb is negated.

- "The food service business does not fit the company's policy"
- "The food service business fits the company's policy"

In case the verbs were antonyms, matching was carried out between ARG0 of the first PA and ARG1 or ARG2 of the second PA and vice-versa as called for in the example of Table 3.5.

Table 3.5 Example for matching opposite arguments

Sentence	John bought the car from Jack	Jack sold the car to John
Verb	bought	sold
ARG0	John	Jack
ARG1	car	car
ARG2	Jack	John

The scores generated by verb matching and argument matching have been consolidated by extending the approach proposed by Rios et al (2011) for framing a metric to assess Machine Translation quality. The consolidated score for the PA tuples has been calculated using Equation (3.1).

$$\text{overall\_score} = \sum_{i \in \text{Comp}} x_i . w_i . \text{score}_i \tag{3.1}$$

where Comp refers to the set of components {Verb, ARG0, ARG1, ARG2, AO} with AO indicating all other arguments ARG2-ARG5 and AM-MOD. The boolean value x_i denotes the presence or absence of component i and w_i is

the weight assigned to the component i. For instance, the verb component is assigned a weight of 0.4, whereas ARG0, ARG1 components are assigned weights of 0.2 and ARG2, AO categories have weights of 0.1. The weights were fixed after experimenting with various values between 0.5 and 0.1. When any of the arguments is not present the corresponding weight is added to the weight of the verb component. This ensures that the summation of all weights is equal to 1 and the verb component retains the highest weight. The term $score_i$ refers to the scores computed by the verb and argument matching processes. A matrix of scores was generated by matching each PA tuple from the first sentence with every PA tuple of the second sentence.

Pairing PA tuples

In the last step of the PA matching process, pairs of PA tuples were identified as paired, loosely paired or unpaired based on the similarity value (Yadav et al 2012). For every PA tuple, the closest matching tuple from the second sentence, having the highest score in the similarity matrix was identified. The tuples have been classified based on this maximum similarity value using the below rules:

- If similarity value >= 0.5 it implies tuples are 'paired'
- If value is between 0.25 and 0.5 tuples are 'loosely paired'
- Otherwise tuples are 'unpaired'.

For each sentence pair, the number of paired, loosely paired and unpaired tuples was recorded and used in the Classification stage of the Paraphrase Recognition process to segregate the sentence pairs.

3.2.3 Classification of Sentence pairs

In the Classification stage, the input sentence pairs were first segregated into different groups based on the presence of Paired (P), Un-

Paired (UP) and Loosely Paired (LP) tuples. With respect to unpaired tuples, distinction has been made with respect to the sentence containing the unpaired portion and eight categories have been formed as shown in Table 3.6. After categorization, various features based on phrase comparison as well as Named entity features were extracted from the sentence. A supervised approach has been adopted, where the extracted features were fed to an SVM Classifier which recognizes paraphrases. This process acts as an additional filter to distinguish the paraphrases from the non-paraphrases. The sentence pairs having all their tuples paired may turn out to be non-paraphrases and sentences with no unpaired tuples may be paraphrases.

Table 3.6 Categories of Sentence pairs

Category	Description	Paired tuples	Loosely Paired tuples	Unpaired tuples	
				Sentence 1	Sentence 2
I	Only unpaired tuples	NIL	NIL	Present in either one	
II	No unpaired tuples	Present in either one		NIL	NIL
III	Paired and UP tuples in at least one sentence	Present	NIL	Present in either one	
IV	Paired and UP tuples in both sentences	Present	NIL	Present	Present
V	LP and UP tuples in at least one sentence	NIL	Present	Present in either one	
VI	LP and UP tuples in both sentences	NIL	Present	Present	Present
VII	Paired and LP tuples, UP tuples in at least one sentence	Present	Present	Present in either one	
VIII	Paired, LP tuples and UP tuples in both sentences	Present	Present	Present	Present

In order to distinguish the paraphrases from the non-paraphrases in each category, various features have been extracted from the sentence pairs. These include surface-level features such as word overlap, presence of positive / negative cue words as well as those computed by matching the phrases in the sentences. Phrase matching was opted for in the second stage to perform a finer level of comparison of the sentences. Phrases were extracted from the output of the SENNA parse and a similarity matrix was constructed for all the phrases similar to the approach used for PAs. For each phrase of the first sentence, the closest matching phrase in the second sentence was determined. The phrase pair was classified as 'Paired' / 'Loosely Paired' / 'Un-Paired' depending on the similarity value. Finally, the length of unpaired phrases has been used as a feature. Table 3.7 lists the complete set of features used along with their description.

Table 3.7 Features used in the Classification stage of PASPR system

Feature, type, number	Description
Word_overlap, Numeric, single	Extent of word overlap between the two sentences.
Word_Similarity, Numeric, single	The similarity between the sentences assessed in terms of WordNet distance between the nouns, verbs, adverbs and adjectives.
Named_Entity match, Numeric, single	Ratio of matching named entities to the maximum number of named entities in the two sentences. Named entities were detected from the parse produced by SENNA.
Unpaired_phrase lengths, numeric, pair	Ratio of the number of words in the unpaired portion (after phrase matching) to the total number of words.
Positive cue words, boolean, pair	Indicate presence of positive cue words such as "rise", "gain", "win" in the unpaired portion.
Negative cue words, boolean, pair	Indicate presence of negative cue words such as "fall", "loss", "loose" in the unpaired portion.
Alternation, boolean, pair	Indicate presence of alternation cue words such as "but", "despite", "although" in unpaired portion.
Speech action, boolean, pair	Signal presence of speech action words such as "say", "report", "announce" in unpaired portion.

The positive and negative cue words in the unpaired portions were used to check for the presence of antonyms. There is very high probability of the input pair being non-paraphrases, if one sentence of the pair has positive or negative cue words in its unpaired portion as in the case of Table 3.8-Example 1.

Table 3.8 Examples of cue word presence

Example	Category	Sentence 1	Sentence 2
1 – Presence of positive / negative cue words	Non paraphrases	A divided Supreme Court ruled Monday that Congress can force the nations public libraries to equip computers with anti pornography filters.	The Supreme Court said Monday the government can require public libraries to equip computers with anti pornography filters **rejecting** librarians complaints that the law amounts to censorship.
2 – Presence of alternation terms	Non paraphrases	And of those, 149, or 55%, "claimed to treat, prevent, diagnose or cure specific diseases."	And of those, 55 percent, or 149, claimed to treat, prevent, diagnose or cure specific diseases **- despite** the regulations prohibiting that kind of statement.
3 – Presence of words describing speech action	Paraphrases	**In a statement, Microsoft said** the dividend is payable Nov. 7 to shareholders of record on Oct. 17.	The dividend, the company's second this calendar year, is payable on Nov. 7 to shareholders of record at the close of business Oct. 17.

The same rule applies if the unpaired phrasal portion of any one sentence contains alternation terms as shown in Table 3.8 - Example 2. On the other hand, the presence of cue words corresponding to the speech action in the unpaired portion indicate additional portions which do not contribute significantly to the sentence meaning and therefore imply paraphrases as in Table 3.8 - Example 3. The thirteen features extracted from the sentence pairs of each of the eight categories were fed separately to an SVM classifier which then classified the input pair as paraphrases / non-paraphrases by modeling the behavior of each category.

3.3 RESULTS

The performance of the two systems based on intermediate language representations has been assessed using the MSRPC and augmented KMC, in terms of measures such as Accuracy, Precision, Recall and F-measure.

3.3.1 Performance Evaluation of UNLPR system

The performance of the proposed UNL matching scheme has been compared with that of Singh et al's system (2012) which has recorded a correlation of 0.1936 when the IITB-CFILT UNL converter has been used. For benchmarking the proposed system, the existing system has been re-implemented by using the Russian UNL converter and various measures such as accuracy, precision, recall and F-measure have been assessed. Two variations of the existing system were tried; in the first the originally prescribed threshold of 0.5 was used. In the second case, the system has been tested using a Support Vector Machine Classifier which avoids the need for using a threshold. The results have been presented in Table 3.9.

Table 3.9 Performance Evaluation of UNLPR system

System Variant		Accuracy %	Precision %	Recall %	F-measure %
Existing System	Threshold = 0.5	59.7	60.6	74.0	66.6
	SVM Classification	66.5	67.6	92.2	78.0
Proposed System	All features	**71.0**	**71.9**	92.6	**81.0**
	Without Named Entity features	70.6	71.1	**93.7**	80.9
	After Co-reference resolution	66.5	68.5	91.8	78.5
	Word Sense Disambiguation	70.8	71.5	93.1	80.9

With respect to the proposed system, experiments were conducted by considering various combinations of the features listed in Table 3.2 and options such as Co-reference resolution, Word Sense Disambiguation. Of the features listed in Table 3.2, the best individual performance of 70.84% accuracy was registered by Universal Word features. This can be attributed to the fact that similar sentences or paraphrase pairs in MSRPC exhibit a considerable degree of word overlap. This was followed by the Overall Relation Precision and Recall and finally Simple Relation features with accuracies of 68.23% and 67.53% respectively. This difference is due to the fact that the overall relation features permit the matching of relations within the same group and do not require the candidate relations to be exactly the same. Since Named Entity features do not include any of the UNL entities such as UWs or relations, they have not been used as stand-alone inputs.

From the results of the experiments it can be observed that the proposed UNL matching system which uses all the eight features has the best overall performance. An increase in accuracy of more than 11% is observed when compared to the existing approach of Singh et al (2012). A notable aspect is that combining the scoring mechanism of the existing system with a machine learning approach serves to improve the accuracy considerably.

Experiments were also conducted by performing Co-reference Resolution and WSD independently. Other aspects that can be inferred from the results are that:

- The usage of Named Entity features improves the performance of the system as paraphrases tend to share more Named entities
- Applying Word Sense Disambiguation on universal words (specifically nouns and verbs) during UNL matching, leads to a slight drop in accuracy due to the reason that considering specific senses of words is more restrictive.

- When Co-reference resolution was carried out on pronouns before computing the features listed in Table 3.2, a drop in performance was observed. This can be attributed to the fact that in some cases the references are wrongly resolved as shown below:

  Original Sentence: But under cross-examination by O'Donnell's attorney, Lorna Schofield, Toepfer conceded **she** had ignored many of O'Donnell's suggestions and projects.

  After Co-reference resolution: But under cross-examination by O'Donnell's attorney, Lorna Schofield, Toepfer conceded **O'Donnell's** had ignored many of O'Donnell's suggestions and projects.

Additionally with respect to semantic similarity assessment, co-references across sentences rather than within the same sentence is more of an issue as can be seen from the example given below:

Sentence 1: The two had argued that only a new board would have had the credibility to restore El Paso to health.

Sentence 2: He and Zilkha believed that only a new board would have had the credibility to restore El Paso to health.

The inclusion of word overlap features resulted in an increased accuracy of 73%. But this requires both the input sentences to belong to the same language. Since the objective here is to develop a system for measuring similarity between inputs from different languages, such features have been disregarded.

Despite the improvement in performance over the existing UNL oriented PR system, the results obtained on the MSRPC are less when compared to those reported in Table 2.6 where lexical, syntactic and semantic features have been directly extracted from the input sentences to result in a maximum accuracy of 77%. A similar result is observed with respect to the KMC where the accuracy drops to 97.3% for the UNL matching approach,

which is less than the best value of 98.6% obtained when selected features are used as discussed in Section 2.4.2. The better performance of the non-UNL approaches can be attributed to the use of word overlap and dependency parse based features.

3.3.2 Performance Evaluation of PASPR system

The PASPR system has been evaluated on two different corpora by first pairing the PA tuples in each input sentence pair and then segregating the input pairs into eight categories. The features based on word overlap, phrase matching and occurrence of cue words, were used to construct a classification model for each category separately. SVM classification has been adopted by using the LibSVM tool. For the MSRPC, the classification model for each of the eight categories was constructed from the training set, and evaluation was carried out using the test set. Experiments were conducted using the thirteen features listed in Table 3.7 to determine the best set of features for each category. The best performing feature set as well as the accuracy and F-measure have been given in Table 3.10.

Table 3.10 Performance of PASPR system on MSRPC

Category	Best Set of Features and Count	Accuracy %	F-measure %
I	Word overlap, Similarity, Named Entity, Unpaired phrase length (5)	81.1	63.2
II	Word overlap, Similarity, Named Entity (3)	83.5	90.6
III	Word overlap, Similarity, Named Entity, Unpaired phrase length, Speech action (7)	76.8	83.9
IV	All (13)	76.6	84.3
V	Same as Category I	74.3	63.0
VI	Same as Category III	75.5	75.5
VII	Same as Category III	72.3	81.2
VIII	All (13)	75.0	81.3

For Categories I and V which do not have any paired tuples, the best performance was recorded when the first five features including unpaired phrase length were used. As expected, with respect to Category II which has no unpaired tuples only the first three features excluding all features relying on unpaired portions yielded the best performance. For categories III, VI and VII including the features pertaining to speech action along with the first five features has yielded the best performance. Categories IV and VIII are the most complex as they contain both paired and / or loosely paired as well as unpaired tuples in both sentences. For these two categories, the entire set of 13 features was required for classification. The overall accuracy and F-measure were calculated by consolidating the results from all categories, yielding 78% and 84.7% respectively. This is marginally better than the current best performance of 77.4% and 84.1% on the MSRPC registered by Madnani et al (2012).

Categories VII and V have yielded the lowest performance. In Category VII which corresponds to sentences containing paired, loosely paired tuples as well as unpaired tuples in either sentence, the number of False Positives was found to be very high. An analysis of these cases indicates that though the sentence pairs exhibit considerable word overlap, there is an additional or extra portion available in either of the sentences as shown in the example given below:

> **Sentence 1:** He had been arrested twice before for trespassing and barred from the complex **home to his mother and two children**.
>
> **Sentence 2:** He had been arrested twice before for trespassing and was barred from the complex.

With respect to Category V sentences, which contain loosely paired tuples and unpaired tuples in either of the sentences the low

performance was due to a higher number of false negatives. The false negatives were found to have additional portions with very less word overlap. Such portions require extensive analysis or real world knowledge to establish equivalence as in the following example:

> **Sentence 1: Brendsel** and chief financial officer Vaughn Clarke resigned June 9.
>
> **Sentence 2:** The **company's chief executive** retired and chief financial officer resigned.

In order to improve the performance further, additional features which determine the significance of the unpaired portion are required to reduce the false positives and false negatives. The PASPR approach was also tested on the augmented KMC containing 1087 pairs of paraphrases and non-paraphrases. A cross-validation approach has been used to estimate the category-wise accuracy shown in Table 3.11.

Table 3.11 Performance of PASPR system on extended KMC

Category	Best Set of Features and Count	Accuracy %
I	Word overlap, Similarity, Named Entity, Unpaired phrase length (5)	98.3
II	Word overlap, Similarity, Named Entity (3)	97.5
III	Word overlap, Similarity, Named Entity, Unpaired phrase length, Speech action (7)	99.3
IV	All (13)	94.7
V	Same as Category I	98.9
VI	Same as Category III	99.1
VII	Same as Category I	99.0
VIII	All (13)	100.0

The best set of features was found to be similar to that of the MSRPC for all categories other than Category VII. In this case, good performance was achieved using Word similarity, Named Entity features in addition to word overlap and unpaired phrase length features. The overall accuracy on the extended KMC was found to be 98.6% which is better than that of 98.4% recorded by Cordeiro et al (2007).

3.4 SUMMARY

Two approaches – UNLPR and PASPR, based on Intermediate representations have been proposed for Paraphrase Recognition. In both approaches, supervised learning has been used for classifying the input sentence pair. The first approach – UNLPR which operates on UNL representations has the advantage of being applicable for cross-language inputs but has registered only moderate performance of 71% on the MSRPC. The second approach PASPR which relies on PA matching in addition to features extracted from the sentence pairs is promising and has yielded an accuracy of 78% on the MSRPC. Considering the fact that the MSRPC has a fixed partitioning as training set and test set which may predispose the results, the UNLPR and PASPR approaches have also been tested on the extended KMC with similar results. The accuracy of the UNLPR approach was found to drop to 97.3% whereas 98.6% has been registered for the PASPR approach. The higher accuracy of both approaches on the extended KMC can be attributed to the shorter sentence length in the corpus and greater word overlap for positive paraphrase cases.