Paraphrase Recognition using Predicate Argument Structure Representation

Dr.A.Chitra[1], Anupriya Rajkumar[2]
{ctr.psg@gmail.com[1], anupriya_rajkumar@yahoo.co.in[2]}

Professor and Head, CA Department, PSG College of Technology, Coimbatore, Tamil Nadu, India – 641004[1], Professor, CSE Department, Dr.Mahalingam College of Engg. & Technology, Pollachi, Tamil Nadu, India – 642003[2]

Abstract. One of the tasks that make Natural Language Processing applications challenging is Paraphrase Recognition which is the establishment of semantic equivalence between two text units. A popular approach adopted in Paraphrase Recognition, is the usage of symbolic meaning representations as intermediate forms. In this work, Predicate Argument Structures (PAS) have been explored for the task of Paraphrase Recognition. The performance of the system was evaluated on the Microsoft Research Paraphrase Corpus and was found to be superior to existing approaches when the PAS based system was enhanced by using a table of equivalent phrases.

Keywords: Paraphrase Recognition, Predicate Argument Matching, Support Vector Machine.

1 Introduction

Human communication typically occurs through a multitude of natural language forms, all of which are characterized by rich semantic variability and ambiguity. The establishment of Semantic Similarity between text units is a pivotal task in applications such as Information Extraction, Question Answering and Summarization. Paraphrases and Entailment are two common forms of semantic similarity. Two text units are said to paraphrase each other, when exact semantic equivalence can be established between them. In text entailment, one of the inputs, usually the shorter one, also termed as the hypothesis may be inferred from the longer unit or text.

A logical solution for establishing semantic similarity would be to translate the input text units to anintermediate representation and then compare these. In this work, Predicate Argument Structure (PAS) basedintermediate representations have been employed for designing Paraphrase Recognition systems. This enables adeeper comparison of sentences by matching the semantic roles. In this work, a two stage approach has beendesigned by first pairing the Predicate Argument tuples. In the second stage, the sentences were grouped basedon the extent of paired and unpaired tuples and features extracted from the sentence pairs in each group were fedto a SVM classifier in order to recognize the paraphrases.

Section 2 of the paper describes previous work on Paraphrase and Text Entailment Recognition which relies onintermediate representations. Section 3 elaborates on the design of the Paraphrase Recognition system and Section 4 presents the performance evaluation. Section 5 concludes with future directions.

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314719

2 Literature Review

Paraphrase Recognition (PR) systems employ different techniques such as vector space models, surface string similarity, syntactic similarity, decoding and logic based approaches to establish semantic equivalence. The

Microsoft Research Paraphrase Corpus (MSRPC) has been used as a benchmark dataset to assess the performance of PR systems. Machine learning based Paraphrase Recognition systems have proved to be quite successful and use either machine translation metrics or a combination of lexical, syntactic and semantic features extracted directly from the input sentence pair [1]. The usage of representations such as FrameNet's frames or semantic roles from PropBank is another alternative which has been previously used for Paraphrase Recognition. In such systems, the input sentences are represented using intermediate representations from which similarity is assessed.

A popular intermediate representation which has been used in semantic similarity assessment is the Predicate

Argument Structure. Hickl et al [2] have designed a system for Recognizing Textual Entailment (RTE) by aligning Predicate Argument structures in addition to using lexical, syntactic and co-reference information. Rios and Gelbukh [3] have employed the TINE metric designed for automatic evaluation of machine translation for the RTE task. The metric combines lexical and semantic role matching by first aligning the verbs and then computing the cosine similarity between their arguments.

Qiu et al [4] have utilized a supervised framework focused on matching predicate argument tuples for detecting dissimilarities between sentences and detecting paraphrases. Initially the most similar predicate argument tuples were paired and then the unpaired tuples were examined by an SVM based dissimilarity classifier to judge the significance of extra information. The system labelled the input sentences as paraphrases, if there were very less or no unpaired tuples. The system has yielded an accuracy of 72% on the MSRPC. Yadav et al [5] have proposed an extension of Qiu et al's approach by distinguishing between paired, unpaired and loosely paired tuples. Liu et al [6] have used a sentence factorization approach where a sentence is factorized into its hierarchical form where each node represents a predicate-argument form. The authors have used a deep learning based Siamese network architecture and have obtained an accuracy of 74.09% on the MSRPC. Mohamed and Oussalah [7] have developed a hybrid system by combining sentence similarity assessment with named-entity semantic relatedness. This hybrid approach has resulted in an accuracy of 75.7% on the MSRPC.The present work is motivated by the observation that, though the PAS matching approach has been widely used in the RTE task, it is has been less explored in Paraphrase Recognition systems.

3. 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. In this work Predicate Argument matching approach has been used for recognizing sentential paraphrases. PA alignment is more relevant than surface level matching schemes when the sentences have considerable word overlap but convey dissimilar actions.

Previous approaches based on predicate alignment for the RTE task or paraphrase recognition have relied on score computation or a supervised approach. Hickl et al [2] and Rios and Gelbukh [3] have both used a supervised learning approach on features computed from the PA structures for the RTE task. In this work, similar to the approach employed by Qiu et al [4], 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 theClassification stage operates on features extracted from the PA representations of the input sentences. The proposed 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 are converted into Predicate Argument representation using a Semantic Role Labelling tool. In the PA matching stage, the PA tuples are matched by locating same or similar predicates/verbs and then matching their corresponding arguments. Similar to the method proposed by Yadav et al [5], the extent of similarity between the matched tuples is 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.

The novel aspect of this work is that after pairing the PA tuples, the sentence pairs are 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 [8] which rely on either a slicing feature or distance based clustering for partitioning the inputs. In the proposed system, for each category, separate classification models have been constructed using different features extracted from the sentence pairs. This approach has been proposed in order to handle the disparities in the nature of the paraphrases. In some cases though all the 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. The pseudo-code for the two stage approach has been given in Figure I.

```
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 Vi in Sentence 1 and verb Vj in Sentence 2
    o Compute the similarity between the verbs
    o If verbs are similar and neither one is negated:
        ▪ Identify the common arguments for Vi and Vj
        ▪ Match equivalent arguments to generate the argument scores for A0, A1, A2 and Aothers (for all other argument types except AM_NEG)
    o else if Vi and Vj are antonyms
        ▪ Match argument A0 of Sentence 1 with A1 of Sentence 2 and vice versa
        ▪ Match all other equivalent arguments
    o Generate a consolidated score Scoreij by combining the verb similarity and all the argument scores
    o Set similarity matrix entry to Scoreij
• For each Vi, determine the most similar verb Vmax with score Smax from Sentence 2 using the Similarity matrix
    o If Smax > 0.5 the verbs Vi and Vmax are considered as paired
    o else if 0.25 < Smax < 0.5 then Vi and Vmax are loosely paired
    o else Vi and Vmax 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 I

ALGORITHM FOR PARAPHRASE RECOGNITION USING PA MATCHING

3.1 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 [9] 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. 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 (AMMOD). The output produced by the SENNA parser is processed to pick the phrases and predicate argument tuples.

3.2 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 are matched by first checking if the verbs are identical or antonyms. Otherwise the similarity is computed by considering the distance between the verbs and also the WordNet synsets of the candidate verbs. The scores are assigned depending on the extent of similarity between the verbs.

Argument Matching

The arguments of similar verbs are 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 [10] in their work on determining entailment. Two arguments are said to match when there is considerable word overlap between them or there is high degree of similarity between the words. As proposed by Wu et al [11] in their work on detecting cross-language similarity, ARG0, ARG1 and ARG2 categories are given higher preference and generate individual scores. All other argument categories are clubbed together to generate a single score.

During matching, the common argument categories are first detected. In the default case, matching is carried out strictly between corresponding arguments only. The only exception to this general matching strategy is that in cases where ARG1 is missing in one sentence but present in the other - then ARG2 of the first sentence is matched against ARG1 of the second sentence. If two verbs are similar but either is negated, indicated by the presence of the AM-NEG argument, further matching is not carried out and the consolidated similarity score between the PAs is set to 0. In case the verbs are antonyms, matching is carried out between ARG0 of first PA and ARG1 / ARG2 of the second PA and vice-versa.The scores generated by verb matching and argument matching are consolidated by extending the approach proposed by Rios and Gelbukh [3]. The consolidated score for the matched PA tuples is in the range 0 – 1 and is computed adaptively by assigning the highest preference to the verb similarity score, followed by ARG0, ARG1 and ARG2 scores and least preference to scores of other argument categories such as ARG3 - ARG5 and AMMOD. A matrix of scores is 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 are identified as paired, loosely paired or unpaired based on the similarity value [5]. For every PA tuple, the closest matching tuple from the second sentence having the highest score in the similarity matrix is identified. The tuples are classified based on this maximum similarity value using the rules given below:

□□If similarity value >= 0.5 it implies tuples are ‘paired’
□□If similarity value is between 0.25 and 0.5 it implies tuples are ‘loosely paired’
□□Otherwise tuples are ‘unpaired’.

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

3.3 Classification of Sentence Pairs

In the Classification stage, various features based on phrase comparison as well as Named entity features are extracted from the sentence pairs. A supervised approach is adopted, where the extracted features are fed to an SVM Classifier which recognizes paraphrases. This process acts as an additional filter to distinguish the paraphrases from the non-paraphrases. The input sentence pairs are 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 is made with respect to the sentence containing the unpaired portion. The sixteen possible combinations have been grouped into eight different categories as shown in Table I.

TABLE I Categories of Sentence pairs

Category	**Description**	**Paired tuples**	**LP tuples**	**UP tuples**	
				Sent. 1	**Sent. 2**
I	Only unpaired tuples	NIL	NIL	Present in either one	
II	No unpaired tuples	Present in either one		NIL	NIL
III	Paired &UP tuples in at least onesentence	Present	NIL	Present in either one	
IV	Paired & 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 & LP tuples, UP tuples inat 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, various features are extracted from the sentence pairs in each category. 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 is opted for in the second stage to perform a finer level of comparison of the sentences than the PA level. Phrases are extracted from the output of the SENNA parse and a similarity matrix is 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 is determined. The phrase pair is classified as ‘Paired’ / ‘Loosely Paired’ / ‘Un-Paired’ depending on the similarity value. Table II lists the complete set of features used along with their description.

TABLE II Features used in the Classification stage of the Paraphrase Recognition system

Feature, type and 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 words (specifically 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.
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 are 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. The same rule applies if the unpaired phrasal portion of any one sentence contains alternation terms. 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. The thirteen features are extracted from each sentence pair. For each category, a different subset of the thirteen features may help to distinguish the paraphrases and non-paraphrases. The best performing feature set for each category has been identified by building separate SVM classification models for categories I to VIII listed in Table I. This approach has been followed since the sentences in each category differ in nature.

4. Performance Evaluation

The Microsoft Research Paraphrase Corpus (MSRPC) consisting of 5801 pairs of sentences has been used for evaluating the performance of the Paraphrase Recognition system. The corpus is divided into a training set with4076 sentence pairs and test set with 1725 pairs [12]. The system has been evaluated 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 [13]. The SVM classifier has been chosen as it avoids the problem of local minima and can produce stable and repeatable results. Further with respect to Paraphrase Recognition, SVM has been found to perform well consistently [1]. For 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 13 features listed in Table 2 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 III.

TABLE III Performance of PAS based Paraphrase Recognizer

Category	**Sentence pairs Count**	**Best Set of Features and Count**	**Accuracy %**	**F-measure %**
I	74	Word overlap, Similarity, Named Entity, Unpaired phrase length (5)	81.1	63.2
II	515	Word overlap, Similarity, Named Entity (3)	83.5	90.6
III	452	Word overlap, Similarity, Named Entity, Unpaired phrase length, Speech action (7)	76.8	83.9
IV	188	All (13)	76.6	84.3
V	105	Word overlap, Unpaired phrase length (3)	74.3	63.0
VI	53	Same as Category III	75.5	75.5
VII	242	Same as Category V	72.3	81.2
VIII	96	All (13)	75.0	81.3
Complete Test Set			**78.0**	**84.7**

Categories IV and VIII are the most complex as they contain both paired / as well as unpaired tuples in both sentences. For these two categories, the entire set of 13 features is required for classification. The overall accuracy and F-measure were calculated by consolidating the results from all categories, yielding 78% and 84.7% respectively. The proposed system has registered better performance than all other previous systems except that of Ji & Eisenstein [14] and Yin & Schutze [15].

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 below example:

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, a table of equivalent phrases was provided to the system. If the sentence pair possessed equivalent phrases available in the table, the phrase in one sentence was replaced by its equivalent present in the other sentence. The major categories of equivalent phrases include: Abbreviations and their expansions, Idioms and their equivalent phrases, Shortened versions and World knowledge or facts. A sample set of phrases has been listed in Table IV.

TABLE IV List of Equivalent Phrases

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

Including phrasal pairs which indicate either linguistic equivalence or embed world knowledge was found to improve the accuracy of the PAS based system from 78% to 81.2% which is higher than that of previous approaches.

TABLE V Performance Comparison of PR systems

System	**Accuracy on MSRPC**
PAS based Paraphrase Recognition System	78.0%
Yin & Schutze [13]	78.7%
Ji & Eisenstein [12]	80.4%
PAS system extended using phrasal pairs	81.2%

The results have been summarized in Table V and the proposed system which employs a two-stage approach based on PA matching has registered competitive performance when compared to previous work. The reason for good performance of the PAS based approach is that it enables a deeper comparison based on matching of semantic roles. The results also indicate that the PAS based system enhanced by the inclusion of phrasal pairs is a promising approach for Paraphrase Recognition.

5. Conclusion

The major contribution of this paper is the design of a two stage Paraphrase Recognition system using Predicate Argument Structures as an intermediate notation. The PAS based system extended using a table of phrasal pairs is found to be suitable for the PR task as it has exhibited better performance than other existing approaches. Possible directions for future work include the identification of suitable features for judging the significance of unmatched portions and the deployment of the Paraphrase Recognition system in practical applications such as multi-document summarization and plagiarism detection.

References

[1] Androutsopoulos I, Malakasiotis P (2010) A Survey of Paraphrasing and Textual Entailment Methods.Journal of Artificial Intelligence Research 38: 135-187.

[2] Hickl A, Bensley J, Williams J, Roberts K, Rink B, Shi Y (2006) Recognizing Textual Entailment withLCC's GROUNDHOG System. In: Second PASCAL Challenges Workshop, Venice, Italy, pp 80-85.

[3] Rios M, Gelbukh A (2012) Recognizing Textual Entailment with Similarity Metrics. In: 11th MexicanInternational Conference on Artificial Intelligence, San Luis Potosi, pp 15-20.

[4] Qiu L, Kan MY, Chua T (2006) Paraphrase Recognition via dissimilarity significance classification. In:Conference on Empirical Methods on Natural Language Processing (EMNLP), Sydney, Australia, pp 18–26.

[5] Yadav R, Kumar A, Kumar AV, Kumar P (2012) Conceptualization of Sentence Paraphrase Recognition with Semantic Role Labels. In: International Conference on Data Mining, Las Vegas, pp 172-180.

[6] Liu B, Zhang T, Han F.X., et al (2018) Matching Natural Language Sentences with Hierarchical Sentence Factorization. In: World Wide Web Conference, Lyons, pp. 1237 - 1246.

[7] Mohamed M, Oussalah M (2019) A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics. Lang Resources & Evaluation 54, pp.457–485.

[8] Gayathri Monicka, J & Jamuna, V 2015, 'Hybrid Cascaded MLI topology using Ternary Voltage Progression Technique with Multicarrier Strategy', Journal of Electrical Engineering &Technology (JEET) vol.10,pp.1610-1620,2015

[9] Gayathri Monicka, J Jamuna.V, and Hemalatha.V, (2014), "A new approach for torque ripple minimization of PMBLDC motor drive" in the Lecture Notes in Electrical Engineering (LNEE), ISSN 1876-1100, Vol. 326, pp 285-294.

[10] Andreevskaia A, Li Z, Bergler S (2006) Partial Predicate Argument Structure Matching for EntailmentDetermination. In: Machine Learning Challenges – Evaluating Predictive Uncertainty, Visual ObjectClassification and Recognizing Textual Entailment, Springer Lecture Notes in Computer Science,3944:332-343.

[11] Wu S, Choi JD, Palmer M (2010) Detecting Cross-lingual Semantic Similarity Using Parallel PropBanks.In: 9th Conference of the Association for Machine Translation in the Americas.

[12] Dolan B, Quirk C, Brockett C (2004) Unsupervised construction of large paraphrase corpora: Exploitingmassively parallel news sources. In: Int. Conf. on Comp. Linguistics, Geneva, pp 350–356.

[13] Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Transactions onIntelligent Systems and Technology, vol.2, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] Ji Y, Eisenstein J (2013) Discriminative Improvements to Distributional Sentence Similarity. In:Conference on Empirical Methods on Natural Language Processing (EMNLP), Seattle, USA, pp 891-896.

[15] Yin W, Schutze H (2015) Discriminative Phrase Embedding for Paraphrase Identification. In: HumanLanguage Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, Denver,Colorado, pp 1368–1373.

Author Information

Dr. A. Chitrais working as Professor and Head of the Department of Computer Applications, PSG College of Technology, Coimbatore. She has published three books and around 75 papers in International / National Journals and Conferences. Dr.Chitra is Principal Investigator for several sponsored R&D projects. She is a recipient of Tamil Nadu Young Women Scientist Award and ISTE National Award for Outstanding Academician. Her areas of interest include: Soft Computing, Information Retrieval, Compiler Design, Agent Technology and Machine Intelligence.

AnupriyaRajkumar is working as Professor in the Department of Computer Science and Engineering, Dr.Mahalingam College of Engineering andTechnology, Pollachi. Her areas of interest include: Natural Language Processing, Machine Learning and Information Retrieval.