ASE@DPIL-FIRE2016: Hindi Paraphrase Detection using Natural Language Processing Techniques & Semantic Similarity Computations

Vani K
Department of Computer Science & Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Amrita University
Bangalore, India
k_vani@blr.amrita.edu

Deepa Gupta
Department of Mathematics
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Amrita University
Bangalore, India
g_deepa@blr.amrita.edu

ABSTRACT

The paper reports the approaches utilized and results achieved for our system in the shared task (in FIRE-2016) for paraphrase identification in Indian languages (DPIL). Since Indian languages have a complex inherent nature, paraphrase identification in these languages becomes a challenging task. In the DPIL task, the challenge is to detect and identify whether a given sentence pairs paraphrased or not. In the proposed work, natural language processing with semantic concept extractions is explored for paraphrase detection in Hindi. Stop word removal, stemming and part of speech tagging are employed. Further similarity computations between the sentence pairs are done by extracting semantic concepts using WordNet lexical database. Initially, the proposed approach is evaluated over the given training sets using different machine learning classifiers. Then testing phase is used to predict the classes using the proposed features. The results are found to be promising, which shows the potency of natural language processing techniques and semantic concept extractions in detecting paraphrases.

CCS Concepts

- **Computing methodologies-Natural language processing**
- **Information systems -Document analysis and feature selection; Near-duplicate and paraphrase detection**

Keywords

Paraphrase Detection; Semantic Concepts; POS Tags; Weka

1. INTRODUCTION

Paraphrasing is the process of restating the meaning of a text using other words or adopting the idea and completely rewriting the text information. Paraphrase detection is widely explored in English language. The Microsoft Research Paraphrase corpus (MSRP) is most commonly used benchmark database in English paraphrase detections [1].Vector space models (VSM), Latent Semantic Analysis (LSA), graph structures and semantic similarity based paraphrase detections are explored in English language [2-7]. Even in English language, detection of paraphrasing becomes more complex when the idea is adopted and rewritten. Effective techniques incorporating syntax-semantic techniques, deeper NLP techniques and soft computing approaches may be required to tackle such scenarios [8]. But when it comes to Indian languages, the task becomes more intricate. A paraphrase detection approach using deep learning for Tamil language was proposed in [9]. Paraphrase detection in twitter data and for SMS messages were explored in [10] and [11] respectively. A method for paraphrasing Hindi sentences by synonym and antonym replacements and substitutions was proposed in [12].In FIRE 2016;a shared task for Detecting Paraphrases in Indian Languages (DPIL) [13] is organized. The tasks are defined for 4 Indian languages: Tamil, Malayalam, Hindi and Punjabi. For each language two subtasks are defined as follows:

- Task -1: Given a pair of sentences from news paper domain in the specific language, the task is to classify them as paraphrases (P) or not paraphrases (NP).
- Task-2: Given two sentences from news paper domainin the specific language, the task is to identify whether they are completely equivalent (E) or roughly equivalent (RE) or not equivalent (NE). It is defined with three classes, viz., paraphrases (P), semi-paraphrase (SP) or not paraphrases (NP) respectively.

Task-1 is a binary classification problem, while Task-2 is a multi-class problem. The proposed work is carried out for identification of paraphrases in Hindi language. An approach that utilizes natural language processing (NLP) techniques with semantic similarity computations is adopted. The main focus is given to Task-1 and the same model is applied for evaluating Task-2.
The paper is organized as follows. Section 2 describes the proposed approach in detail. In Section 3, data statistics and evaluation measures are discussed. Section 4 discuss and analyze the results obtained. Section 4 concludes the paper with some insights to future work.

2. PROPOSED APPROACH

Fig.1 depicts the general work-flow of proposed approach. The three main modules and the sub modules are described in the following subsections.

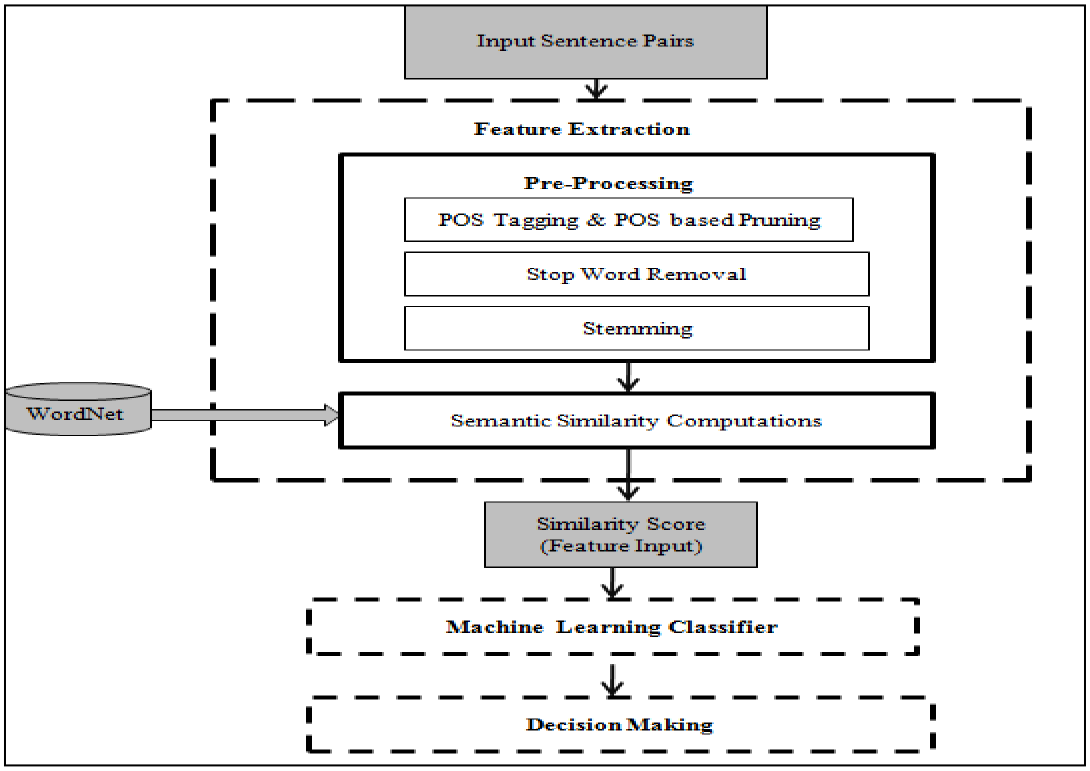

Input Sentence Pairs
Feature Extraction
Pre-Processing
POS Tagging & POS based Pruning
Stop Word Removal
Stemming
WordNet
Semantic Similarity Computations
Similarity Score (Feature Input)
Machine Learning Classifier
Decision Making

Figure 1. General Work-Flow of Proposed Approach

Table 1. Hindi Stop Word List

के	ने	का	हो	एस	थे	कुछ	थीं	के	क
था	कि	जो	कर	मे	गया	करने	किया	लिये	अपने
होता	द्वारा	हुआ	तक	साथ	करना	वाले	बाद	लिए	आप
वे	करते	बहुत	कहा	वर्ग	कई	करें	होती	अपनी	उनके
न	अभी	जैसे	सभी	करता	उनकी	तरह	उस	आदि	कुल
एक	में	की	है	यह	और	से	हैं	को	पर
बनी	नहीं	तो	ही	या	एवं	दिया	हो	इसका	इस
सकते	किसी	ये	इसके	सबसे	इसमें	थे	दो	होने	वह
यदि	हुई	जा	ना	इसे	कहते	जब	होते	कोई	हुए
रहा	इसकी	सकता	रहे	उनका	इसी	रखें	अपना	पे	उसके

Table 2. Hindi Suffix List used for Stemming Process

"◌ो", "◌े", "◌ू", "◌ु", "◌ी", "◌ि", "◌ा","कर", "◌ाओ", "◌िए", "◌ाई", "◌ाए",
"ने", "नी", "ना", "ते", "◌ीं", "ती", ता", "◌ाँ", "◌ां", "◌ों", "◌ें",
"◌ाकर", "◌ाइए", "◌ाईं", "◌ाया", "◌ेगी", "◌ेगा", "◌ोगी", "◌ोगे", "◌ाने",
"◌ाना", "◌ाते", "◌ाती", "◌ाता", "तीं", "◌ाओं", "◌ाएं", "◌ुओं", "◌ुएं", "◌ुआं",
"◌ाएगी", "◌ाएगा", "◌ाओगी", "◌ाओगे", "एंगी", "◌ेंगी", "एंगे", "◌ेंगे",
"◌ूंगी", "◌ूंगा", "◌ातीं", "नाओं", "नाएं", "ताओं", "ताएं", "◌ियाँ", "◌ियों",
"◌ियां", "◌ाएंगी", "◌ाएंगे", "◌ाऊंगी", "◌ाऊंगा", "◌ाइयाँ", "◌ाइयों", "◌ाइयां"

2.1. Feature Extraction

Initially feature extraction is applied for extracting the traits from given sentence pairs. These features are given as the input to the classifier. For extracting the feature, sentences are processed using various pre-processing procedures with the incorporation of NLP techniques.

2.1.1. Pre-processing

Initially the input sentence pairs are tokenized. Then part of speech tagging is carried out.

POS Tagging &POS based Pruning: The word tokens are tagged with their respective classes using NLTK[1] POS tagger [14]. The word classes include noun, verb, adjective, adverb, preposition, conjunction etc. This is followed by POS based pruning. In this pruning process, the tags that can possibly convey some meaning or semantics are only retained while others are pruned out. The retained tags include Noun, Verb, Adjective and Adverb. The tag for cardinality which includes numbers and indicates years, cost etc. are also retained. The remaining tags are pruned out and not considered in further proceedings. This is followed by stop word removal and stemming.

Stop Word Removal: Stop words are the frequent and irrelevant words appearing within the document. The Hindi stop word list used in reported work is given in Table 1.Prior to stemming, punctuation removal is done. As punctuations play an important role in structural composition of documents, their removal can alter the results of NLP applications. The scenario becomes more affected with NLP techniques that operate at document level such as parsing, chunking, semantic role labeling (SRL) etc. Considering the dependence of NLP techniques on these structures of a document, punctuation removal is applied after POS tagging in our approach.

Stemming: Stemming is the process of removal of affixes from the given word. Stemming of the words is done using the suffix list given in Table 2. An example illustration for all these processing’s is done using a sample sentence *S*.

S:	43 केहुएसचिनतेंदुलकरजन्मदिनमुबारकहो,दीजिएबधाई|
After Tokenization:	[43,के,हुए,सचिन,तेंदुलकर,जन्मदिन,मुबारक,हो,,, दीजिए,बधाई, |]
After POS Tagging:	[43(CD),के(IN),हुए(IN),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन(NNP),मुबारक(NNP), हो (IN)(,),दीजिए(VP), बधाई(NN), | (|)]
After POS based Pruning:	[43(CD),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन (NNP),मुबारक(NNP),दीजिए(VP), बधाई(NN)]
After Stop Word Removal:	[43(CD),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन (NNP),मुबारक(NNP),दीजिए(VP),बधाई(NN)]
After Punctuation Removal :	[43(CD),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन (NNP),मुबारक(NNP),दीजिए(VP),बधाई(NN)]
After Stemming:	[43(CD),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन (NNP),मुबारक(NNP),दीज(VP), बधाई(NN)]

The processed sentences are then passed on for pair wise semantic similarity computations and comparisons.

2.1.3. Semantic Similarity Computations

Once the basic pre-processing and NLP techniques based processing is done, the semantic similarity between the processed sentences pairs are computed. The metric used extracts the semantic concepts in the form of synonyms of given word. Instead of considering just surface-level word matching, synonym–level matching is also done. This facilitates paraphrase detection, since in many cases paraphrasing is done by replacing the words with their synonyms. The synonyms are extracted using Word Net[2] lexical database [15-18].The steps for computing the semantic similarity is explained in following steps.

1. For all processed sentence pair,*(S1, S2)* Repeat steps 2 to 8.
2. Initialize *Count* =0.
3. For each word *w*in *S1,* do steps 4 to 7.
4. If *w* is in *S2*, then *Count* = *Count* +1, else go to step 5.
5. Extract synonyms of the word from WordNet.
6. For each synonym *syn* for word *w,* do step 7.
7. If s*yn* is in *S2*, then *Count* = *Count* +1 and goto step 6, else go to step 3.
8. Compute similarity, *sim* using Equation (1).

$$sim = \frac{Count}{max(|S1|,|S2|)} \quad (1)$$

Equation (1) computes similarity between the processed sentences (S1, S2) as the ratio of *Count* value, to the maximum among the lengths of given sentence pair. For illustration consider two sentences S1 and S2.

S1:43 केहुएसचिनतेंदुलकरजन्मदिनमुबारकहो,दीजिएबधाई|

S2:क्रिकेटकेभगवानसचिनकोजन्मदिवसमुबारकहो, दीजिएबधाई|

The sentences after doing tokenization, POS tagging, pruning, stop word removal and stemming are given below. The procedure is same as explained in subsection 2.1.1.

Processed S1:	[43(CD),सचिन(NNP),तेंदुलकर(NNP),जन्मदिन(NNP),मुबारक(NNP),दीज(VP), बधाई(NN)]
Processed S2:	[क्रिकेट(NN),भगवान(NNP),सचिन(NNP),जन्मदिवस(NNP),मुबारक(NNP), दीज(VP), बधाई(NN)]

In these sentences, each word in S1 in checked for its presence in S2. If word is not present, then synonym checking is done. In the given example, 4 exact matches are found, viz., सचिन,मुबारक,दीजand बधाई. One word is identified as synonym; viz. जन्मदिवस is a synonym of जन्मदिन. Thus the count value

[1]http://www.nltk.org/

[2]http://wordnet.princeton.edu/

will be, $Count$ =5. The similarity is computed using Equation (1) which will be:=5/(max(7,7)=0.7142.

The similarity output obtained is considered as the feature input from a sentence pair. This is the input to machine learning classifier.

2.2. Machine Learning Classifiers

Machine learning (ML) based classifiers are used for the paraphrase identification task. The similarity score which is the feature input is fed to the classifier and classification task is done. In the proposed work, different classifiers are tested and the best among them is selected based on accuracy.

2.3. Decision making

Using the training data, initially training phase is implemented. This is followed by testing, where decision making is done. Decision is made on whether a given sentence pair is paraphrased or not in Task-1. In Task-2,multi-class classification is done to decide whether the sentence pair is paraphrased, semi-paraphrased or non-paraphrased.

Section 3 describes the data statistics used in evaluation (training and test data) and the evaluation measures.

3. DATA STATISICS & EVALUATION MEASURES

In DPIL,Task-1 provides 2500 sentence pairs for training. The sentences are labeled as either paraphrased (P) or Non-Paraphrased (NP). The set include 1000 instances for P class and 1500 instances for NP class.Task-2 provides 3500 sentence pairs out of which 1000 are paraphrased (P), 1000 semi paraphrased (SP) and 1500 non-paraphrased (NP). Our main focus was Task-1 while we implemented the same model for Task-2 as well. The feature input is the semantic similarity computed, i.e., sim, using Equation (1). Result evaluation is carried out using the classification measures, viz., recall, precision, F-measure and % accuracy.

	P	**NP**
P	TP	FN
NP	FP	TN

(2)

$$Total\ Population = TP + TN + FP + FN \tag{3}$$

$$Accuracy = \frac{TP + TN}{Total\ Population} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F - measure = 2 * \frac{recall * precision}{recall + precision} \tag{7}$$

Confusion matrix is mainly used to evaluate classification problems. The true positives (TP), false negatives (FN), true negatives (TN) and false positives (FP) are obtained from this matrix. General confusion matrix for binary class problem is shown in Equation (2). In the proposed work, TP indicates the number of paraphrased documents correctly classified as paraphrased. FN indicates the number of paraphrased documents misclassified as non-paraphrased. TN is the number of non-paraphrased documents correctly classified as non-paraphrased and FP indicates the number of non-paraphrased documents misclassified as paraphrased. The total population is computed using Equation (3). Accuracy is measured using Equation (4) which is the fraction of number of correctly classified instances to the total number of instances in the population. Precision, Recall, and F-measure are computed using Equation (5), (6) and (7) respectively. Recall is defined as the number of correctly classified documents to the actual number of correct documents to be identified with respect to a particular class. Precision is defined as the number of correctly classified documents to the total number of documents identified as belonging to that class by the system. F-measure defines the harmonic mean of precision and recall.

Receiver Operating Characteristic Curve (ROC) is also plotted for better understanding. ROC curve plots sensitivity Vs 1-specificity. Sensitivity is same as recall or true positive rate (TPR) while specificity is the true negative rate (TNR) which is defined by the fraction of documents correctly rejected to the total number of documents to be rejected. 1-specificity is termed fall-out, which is the false positive rate (FPR) defined as the fraction of documents misclassified or incorrectly rejected to the total number of documents to be rejected.ROC curves help us to understand the discriminative power of the classifier. Using these measures, the performance of proposed approach is evaluated over Task-1 and Task-2.

4. EXPERIMENTAL RESULTS & ANALYSIS

Initially the proposed approach is evaluated using different classifiers in Weka. Weka[3] is an open source machine learning suite. The accuracy obtained using 10 fold cross-validations over Task-1 ad Task-2 by the tested classifiers is reported in Table 3.It is observed that decision tree exhibits the maximum accuracy in both tasks. Thus for further evaluations decision tree is considered. The Weka implementation of C4.5 decision tree, viz., J48 is used in proposed work.

For better understanding, the ROC curves obtained using J48 onTask-1 and Task-2 is also plotted.Figure.2 and 3 plots the ROC curve for class P in Task-1 and Task-2 respectively. From Figure 2 and 3, it is observed that area under ROC curve (AUC) is 0.9 and 0.799 respectively for Task-1 and 2. The values show that the J48 classifier is able to discriminate the classes significantly in Task-1 and it is not so low in Task-2.

[3]http://www.cs.waikato.ac.nz/ml/weka/

Table 3. Comparison of Accuracy between Multiple Classifiers

Classifier	%Accuracy	
	Task-1	Task-2
Naïve Bayes (NB)	90.38%	64.21%
Support Vector Machine (SVM)	90.36%	64.45%
Decision Tree (J48)	**90.52%**	**66.77%**
Logistic Regression	90.25%	64.89%
Multilayer Perceptron	90.38%	65.09%

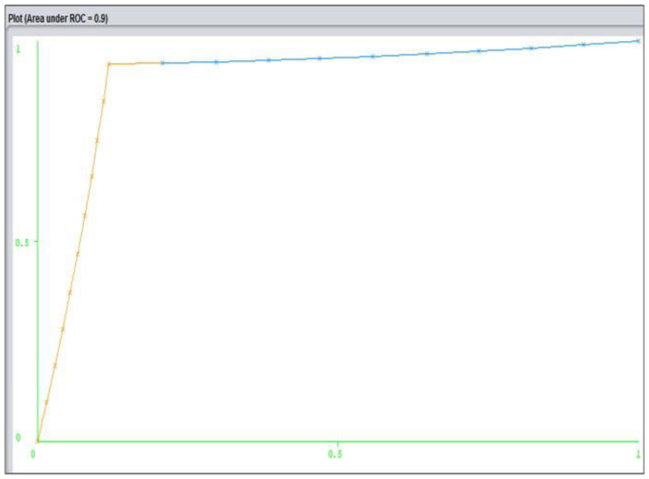


Figure 2. ROC Curve with Decision Tree (J48) for Class P in Task-1

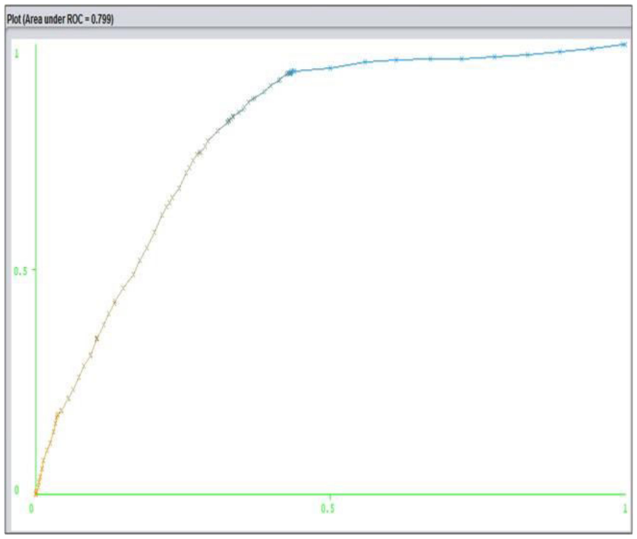


Figure 3. ROC Curve with Decision Tree (J48) for Class P in Task-2

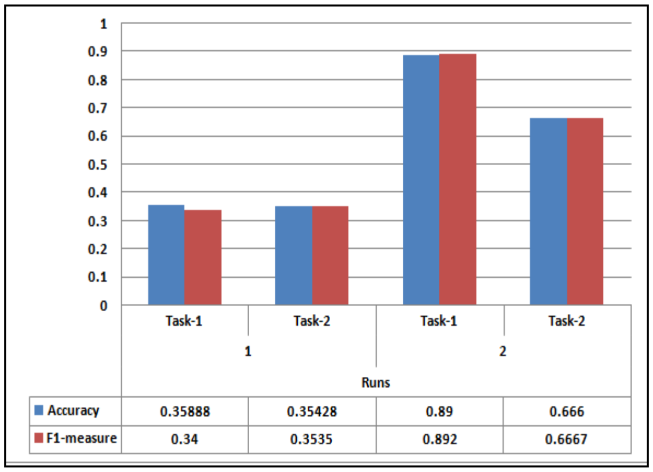
Table 4 and Table 5 reports the classification performance with each class for Task-1 and Task-2 respectively using J48 classifier. In Task-1, an accuracy of 90.52% is obtained at training phase while in Task-2, an accuracy of 66.77% is presented.

Table 4. Classification Measures using J48 Classifier in Training Set-Task 1

Class	Recall	Precision	F-measure
P	0.942	0.84	0.888
NP	0.881	0.958	0.918
Weighted Average	0.905	0.911	0.906

Table 5. Classification Measures using J48 Classifier in Training Set-Task 2

Class	Recall	Precision	F-measure
P	0.518	0.540	0.529
SP	0.515	0.478	0.496
NP	0.869	0.892	0.880
Weighted Average	0.668	0.673	0.670



	Task-1	Task-2	Task-1	Task-2
🟦 Accuracy	0.35888	0.35428	0.89	0.666
🟥 F1-measure	0.34	0.3535	0.892	0.6667

Figure 4. Results with Run 1 and Run 2 Submission on Test Data

Compared to the training results, during the testing phase, our results exhibited significant variation. Figure 4 plots the test results obtained using the proposed approach. Test results presented a considerable drop. In contrast to the 90.52% accuracy (Task-1) on training set, test set presented only 35.88% accuracy. Similar drop is noted in Task-2 also (Run-1 in Figure 4). On rechecking the submission, we found that the results were submitted wrongly. In Run-1 submission, the first sentence pair was not written to the final output file and hence making the second pair as first, third as second etc. and thus completely altering our results.

On request to DPIL, our results were reevaluated. The results of Run 2 are the final results of proposed approach. It is observed from Figure 4, that an accuracy of 89% in Task-1 and 66.6% in Task-2 is obtained on test sets for Run-2. This matches the training results also.

The proposed approach was originally developed for plagiarism identification and classification in English language. The results obtained in Task-1 reflect the potency of our model to be extended to other languages also. Task-2 can be further improved by extracting significant features and using advanced NLP techniques.

5. CONCLUSIONS & FUTURE WORK

In the proposed approach natural language processing techniques and semantic similarity computations are used to classify a Hindi sentence pair as paraphrased or not. Part of speech tagging is used for comparing only relevant tags within each sentence pair. A semantic similarity metric is employed which extracts the word synonyms from WordNet to check whether the compared words are synonyms or not. This facilitates in detailed analysis and comparisons and helps in unmasking paraphrasing imposed by synonym replacements. The metric as a whole helps in detecting and classifying paraphrased and non-paraphrased sentence pairs effectively. In future, deeper natural language processing techniques and intelligent computing techniques can be explored. These advanced techniques are very less explored in Indian language paraphrase detections.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge Department of Science and Technology, Govt. of India (www.dst.gov.in), for sponsoring this research project, Sanction No.SB/FTP/ETA-0212/2014-2016.

7. REFERENCES

[1] Dolan, W.B., Quirk, C., and Brockett, C. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *In Proceedings of the 20th International Conference on Computational Linguistic*s, Geneva, Switzerland.

[2] Mihalcea, R., Corley, C., and Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity, *Proceedings of the National Conference on Artificial Intelligence (AAAI 2006)*, Boston, Massachusetts, 775-780.

[3] Rus, V., and McCarthy, P.M. and Lintean, M.C. and McNamara, D.S. and Graesser, A.C. 2008. Paraphrase identification with lexico-syntactic graph subsumption, *FLAIRS 2008*, 201-206.

[4] Fernando, S., and Stevenson, M. 2008. A semantic similarity approach to paraphrase detection, *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium.*

[5] Blacoe, W., and Lapata, M. 2012. A comparison of vector-based representations for semantic composition, *Proceedings of EMNLP*, Jeju Island, Korea, 546-556.

[6] Islam, A., and Inkpen, D. 2007. Semantic similarity of short texts, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*, Borovets, Bulgaria, 291-297.

[7] Ul-Qayyum, Z., and Altaf, W. 2012. Paraphrase Identification using Semantic Heuristic Features.*Res. J. of Appld. Sci., Engg. and Tech.*, 4(22), 4894-4904.

[8] Vani,K., and Gupta,D. 2016. Study on Extrinsic Text Plagiarism Detection Techniques and Tools,*J. of Engg. Sci. and Tech. Review.*, 9(4), 150-164.

[9] Mahalakshmi, S., Anand Kumar, M., and Soman, K.P 2015. Paraphrase detection for Tamil language using deep learning algorithm. Int. J. of Appld. Engg. Res., 10 (17), 13929-13934.

[10] Mahalakshmi, S., Anand Kumar, M., and Soman, K.P. 2015.AMRITA CEN@ SemEval-2015: Paraphrase Detection for Twitter using Unsupervised Feature Learning with Recursive Autoencoders, SemEval-2015, 45.

[11] Wei Wu., Yun-Cheng Ju., Xiao Li and Ye-Yi Wang. 2010. Paraphrase detection on SMS messages in automobiles.2010. *In Acoustics Speech and Signal Processing (ICASSP),* 2010, 5326-5329.

[12] Sethi,N., Agrawal,P., Madaan,Vishu and Kumar Singh S.2016.A Novel Approach to Paraphrase Hindi Sentences using Natural Language Processing.*Ind. J. of Sci. and Tech.*, 9(28).

[13] Anand Kumar, M., Singh, S., Kavirajan, B., Soman,KP. 2016. DPIL@FIRE2016: Overview of shared task on DetectingParaphrases in Indian Languages. *In Working notes of FIRE 2016-Forum for Information Retrieval Evaluation,* Kolkata, India.

[14] Charniak, E., 1997.Statistical Techniques for Natural Language Parsing, *AI Magazine* 18 (4), 33–44.

[15] Miller, G.A.1995. WordNet: A lexical database for English, *Commun. of the ACM*, 38(11), 39-41.

[16] Bhingardive,S., Shukla,R., Saraswati,J., Kashyap, L., Singh,D., and Bhattacharyya, P. 2016. Synset Ranking of Hindi WordNet. *In Proceedings of theLanguage Resources and Evaluation Conference*, Portorož, Slovenia.

[17] Gupta, D., Vani,K., and Singh, C.K.2014. Using Natural Language Processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection. *In Proceedings of theInternational Conference on Advances in Computing, Communication and Informatics*, Noida, 2694-2699.

[18] Vani,K.,andGupta, D. 2015. Investigating the Impact of Combined Similarity Metrics and POS tagging in Extrinsic Text Plagiarism Detection System. *In Proceedings of the International Conference on Advances in Computing, Communication and Informatics,* Kochi, India, 1578-1584.