NLP-NITMZ@DPIL-FIRE2016: *Language Independent Paraphrases Detection*

Sandip Sarkar
Computer Science and Engineering
Jadavpur University
sandipsarkar.ju@gmail.com

Saurav Saha
Computer Science and Engineering
NIT Mizoram
me@sauravsaha.in

Jereemi Bentham
Computer Science and Engineering
NIT Mizoram
jereemibentham@gmail.com

Partha Pakray
Computer Science and Engineering
NIT Mizoram
parthapakray@gmail.com

Dipankar Das
Computer Science and Engineering
Jadavpur University
dipankar.dipnil2005@gmail.com

Alexander Gelbukh
CIC, Instituto Politécnico Nacional
Mexico
gelbukh@gelbukh.com

ABSTRACT

In this paper we describe the detailed information of NLP-NITMZ system on the participation of DPIL[1] shared task at Forum for Information Retrieval Evaluation (FIRE 2016). The main aim of DPIL shared task is to detect paraphrases in Indian Languages. Paraphrase detection is an important part in the field of Information Retrieval, Document Summarization, Question Answering, Plagiarism Detection etc. In our approach, we used language independent feature-set to detect paraphrases in Indian languages. Features are mainly based on lexical based similarity. Our system's three features are: Jaccard Similarity, length normalized Edit Distance and Cosine Similarity. Finally, these feature-set are trained using Probabilistic Neural Network (PNN) to detect the paraphrases. With our feature-set, we achieved 88.13% average accuracy in Sub-Task 1 and 71.98% average accuracy in Sub-Task 2.

Keywords

Probabilistic Neural Network (PNN), Plagiarism Detection, DPIL, Jaccard Similarity.

1. INTRODUCTION

Ambiguity is one of major difficulties in Natural Language Processing (NLP). In ambiguity, one text can be represented using many forms like lexical and semantic. This is known as paraphrasing. Here we consider only lexical level similarity for paraphrase detection. Paraphrase detection is a very important and challenging task in Information Retrieval, Question Answering, Text Simplification, Plagiarism Detection, Text summarization and even paraphrase detection on SMS [1]. In Information Retrieval, relevant documents are retrieved using paraphrase detection. Similarly, in Question Answering System, the best answer is identified using paraphrase detection. Paraphrase detection is also used in plagiarism detection to detect the sentences which are paraphrases of each other. Researcher used different type of approaches [2] [3] [4] like Lexical Similarity, Syntactic Similarity [5] and other approaches to detect paraphrases. Research problem based on paraphrasing can be divided into three categories: Paraphrase generation, Paraphrase extraction and Paraphrase recognition.

This paper describes the NLP-NITMZ system which participated in DPIL shared task [6]. DPIL (Detecting Paraphrases in Indian Languages) task is focused on sentence level paraphrase identification for Indian languages (Tamil, Malayalam, Hindi and Punjabi). DPIL shared task is divided into two sub-tasks. In Sub-Task 1, the participants have to classify sentences into two categories viz. Paraphrase (P) and Non-Paraphrase (NP).

Table 1. Sentences pair with classification Tag

Pair of Sentences	Tag
പിഞ്ചുകുഞ്ഞുങ്ങളെ വിഷം കൊ ടുത്തു കൊന്ന് യുവതി ആത്മഹത്യ ചെയ്തു. രണ്ടു മക്കളെ വിഷം കൊടുത്തു കൊ ന്നശേഷം യുവതി ആത്മഹത്യ ചെയ്തു.	P
மும்பை கண்டுவடெிப்பு வழக்கில் மலேம் ஒரவர் கதை. பிரசலெ்ஸ் கண்டுவடெிப்பு முக்கிய கு ற்றவாளி நஜீம் லாஷ்ராவி ஐ.எஸ் அ மை ப்பில் ஜயெிலராக இரந்தார்.	NP
ਹੁਣ ਵਭਿਾਗ ਨੂੰ ਬਣਦਾ ਕਰਿਾਇਆ ਅਦਾ ਕਰਨ ਲਈ ਕੇਸ ਬਣਾ ਕੇ ਭੇਜ ਦੱਤਾ ਹੈ ਤੇ ਜਲਦ ਹੀ ਕਰਿਾਇਆ ਅਦਾ ਕਰਦੱਤਾ ਜਾਵੇਗਾ। ਹੁਣ ਵਭਿਾਗ ਨੂੰ ਬਣਦਾ ਕਰਿਾਇਆ ਅਦਾ ਕਰਨ ਲਈ ਕੇਸ ਬਣਾ ਕੇ ਭੇਜ ਦੱਤਾ ਹੈ।	SP
क्रकिेट के भगवान सचनि को जन्मदनि मुबारक हो, दीजएि बधाई। के हुए सचनि तेंदुलकर जन्मदनि मुबारक हो, दीजएि बधाई।	P

Similarly in Sub-Task 2, the participants have to classify sentences into a three point scale i.e., three categories: Completely Equivalent (E), Roughly Equivalent (RE) and Not Equivalent

[1] http://nlp.amrita.edu/dpil_cen/

(NE) i.e. (Paraphrase, Non-paraphrase, and Semi-paraphrase). Table 1 describes the examples of DPIL training dataset.

In Section 2 we provide the detailed architecture of our system like feature-set and machine-learning technique. Section 3 describes the detailed statistics of test and training data which are used by our system. The result on test data is described in Section 4. Section 5 describes the conclusion and future work.

2. SYSTEM ARCHITECTURE

In this section, we elaborate our proposed architecture. As shown in Figure 1, our system NLP-NITMZ is based on three language-independent features: Jaccard Similarity, Levenshtein Ratio and Cosine Similarity. To find the Jaccard Similarity, first we calculate the number of similar unigram between two texts. After that, similarity score is obtained by dividing the count by the total unigram of those two sentences. Next one is Levenshtein Ratio which calculates total number of operations required to change one string to another form. Final feature is Cosine Similarity where each word of sentences is represented using Vector Space model.

For machine learning portion we have used Probabilistic Neural Network to predict the class. Probabilistic Neural Network (PNN) is derived from Bayesian network. PNN is normally used in classification problem and it has 4 layers. Those layers are namely Input layer, Pattern layer, Summation layer and Output layer. The advantage of PNN is that, that are much faster than feed forward Neural Network.

2.1 Features

Our system NLP-NITMZ used three types of features which are Language Independent. We used lexical based features which are mainly used to find the similarity between sentences for all Languages [7] [8].

2.1.1 Jaccard Similarity

The similarity and difference of two sets is calculated using Jaccard Similarity coefficient. For our task, Jaccard similarity coefficient between two sentences is the ratio between the numbers of unigram match to the total number of unique words in those two sentences. If S_1 and S_2 are two sets, then the Jaccard similarity is defined using following equation.

$$\boldsymbol{Jaccard\ Similarity}(\boldsymbol{S_1}, \boldsymbol{S_2}) = \frac{\mathbf{S_1} \cap \mathbf{S_2}}{\mathbf{S_1} \cup \mathbf{S_2}}$$

Table 2 shows the example of Jaccard Similarity.

Table 2. Jaccard Similarity

	Sentences	Score
Sentence 1	भारतीय मुस्लिमों की वजह से नहीं पनप सकता आईएस।	0.2
Sentence 2	भारत में कभी वर्चस्व कायम नहीं कर सकता आईएस।	

2.1.2 *Levenshtein Ratio*

The most common feature to compare two strings is the Levenshtein Distance which is obtained by minimum number of operations required (i.e. replacements, insertions, and deletions) to convert one string to another [9]. In our task we assign same weight, e.g. 1 to all operations. Here we consider character level distance between words of sentences. The probability of two sentences to be paraphrases is high when the edit distance of those two sentences is small.

$$\boldsymbol{EditRatio}(\boldsymbol{a}, \boldsymbol{b}) = \mathbf{1} - \frac{\boldsymbol{EditDistance}(\boldsymbol{a}, \boldsymbol{b})}{|\boldsymbol{a}| + |\boldsymbol{b}|}$$

Example of Levenshtein Ratio is given in Table 3.

Table 3. Levenshtein Ratio

	Sentences	Score
Sentence 1	भारतीय मुस्लिमों की वजह से नहीं पनप सकता आईएस।	0.7712
Sentence 2	भारत में कभी वर्चस्व कायम नहीं कर सकता आईएस।	

2.1.3 Cosine Similarity

Cosine similarity is another widely used feature to measure the similarity between two sentences. In this feature, each sentence is represented using word vectors. Here word vectors are mainly the frequency of words in the sentences. After that cosine similarity is calculated using the dot product of those two word vectors divided by the product of their lengths.

$$\boldsymbol{Cosine\ Similarity}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A}.\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|}$$

Table 4 describes the operation of cosine similarity on Hindi sentence pair.

Table 4. Cosine Similarity

	Sentences	Score
Sentence 1	भारतीय मुस्लिमों की वजह से नहीं पनप सकता आईएस।	0.523
Sentence 2	भारत में कभी वर्चस्व कायम नहीं कर सकता आईएस।	

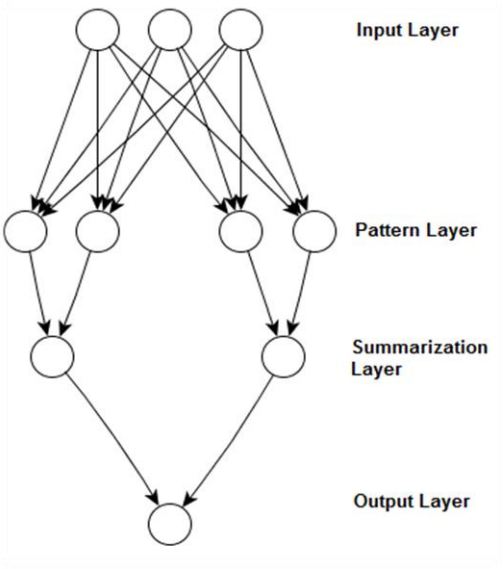

Figure 1. Architecture of PNN

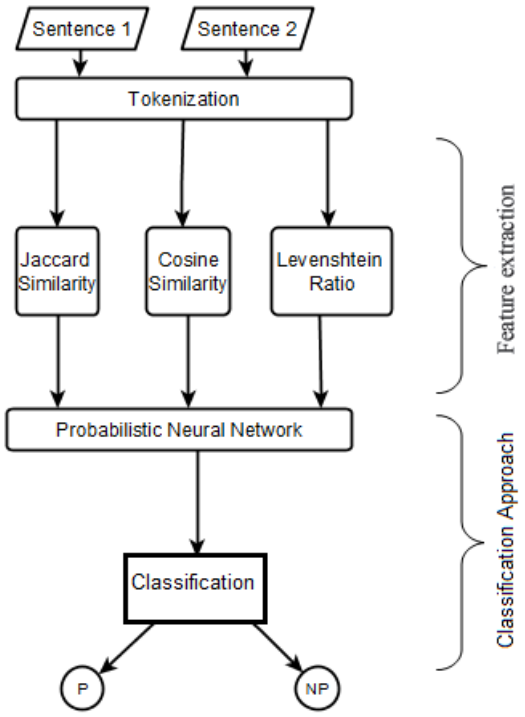

Figure 2. System Architecture of NLP-NITMZ.

2.2 CLASSIFICATION APPROACH

For this classification task we used Probabilistic Neural Network (PNN) to classify those sentences. The PNN was first introduced by Specht [10], and it is mainly based on Bayes Parzen classification. The PNN is one of the supervised learning networks. It is implemented using the probabilistic model, such as Bayesian classifiers. In this network we don't require to set the initial weights of the network. The overall structure of the probabilistic neural network is illustrated in Figure 2. The PNN [11] has four layers: the Input layer, Pattern layer, Summarization layer and Output Layer. PNN have many advantages like it is much faster than well-known back propagation algorithm and has simple structure, PNN networks generate accurate predicted target probability scores, PNN approach Bayes optimal classification [12]. In the same time, it is robust to noise examples.

A simple probabilistic density function (pdf) for class k is as follows where X = unknown (input), X_k = "K^{th}" sample, σ = smoothing parameter and p = length of vectors

$$f_k(X) = \frac{1}{(2\pi)^{\frac{p}{2}}.\sigma^p} e^{\frac{-||x-x_k||^2}{2\sigma^2}}$$

The accuracy of PNN classification depends mainly on probability density function. The probability density function for single population is described using the following equation where n = no of samples in the population.

$$g_i(X) = \frac{1}{(2\pi)^{\frac{p}{2}}.\sigma^p} \frac{1}{n_i} \sum_{k=1}^{n_i} e^{\frac{-||x-x_k||^2}{2\sigma^2}}$$

If there are two classes i, j then classification criteria is decided using the following comparison:

$$g_i(X) > g_j(X) \text{ for all } j \neq i$$

The advantage of PNN networks is that the training process is easy and quick. They can be used in real time. For our experiment we used existing MATLAB toolkit to classify test data[2].

3. Dataset

DPIL shared task includes sentence pairs of four languages: Tamil, Malayalam, Hindi, and Punjabi. This shared task is divided into two sub-tasks. In Sub-Task 1, the main aim was to classify those four sentences as paraphrases (P) or not paraphrases (NP). Similarly Sub-Task 2 is to assign those sentences into three categories completely equivalent (E) or roughly equivalent (RE) or not equivalent (NE). Table 5 describes the details statistics of training and test dataset.

Table 5. Statistics of Training and Test datasets

LANGUAGE	TASK	Count(Train)	Count(Test)
Hindi	Task 1	2500	900
Hindi	Task 2	3500	1400
Malayalam	Task 1	2500	900
Malayalam	Task 2	3500	1400
Punjabi	Task 1	1700	500
Punjabi	Task 2	2200	750
Tamil	Task 1	2500	900
Tamil	Task 2	3500	1400

4. RESULT

The individual accuracy and F1 score is describe in Table 6. At the same time the comparison between winner's score and our score is also described in Table 6. We can see that our proposed method achieved very good result on Panjabi and Hindi language whereas our system struggles on Malayalam and Tamil language. F1 score is the harmonic mean of Precision and Recall. Macro F1 score is used for Task 2 score evaluation. Precision, Recall, F1 score, F1 Macro and accuracy can be described using the following equations where True Positive = (TP), True Negative = (TN), False Positive = (FP), False Negative = (FN).

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

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

$$F1 = \frac{2TP}{2TP + FP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

[2] http://in.mathworks.com/help/nnet/ref/newpnn.html

Table 6. Comparison between Winners's Score and Our System Score

LANGUAGE	TASK	Our System		Winner's System	
		Accuracy	F1 Score	Accuracy	F1 Score
Hindi	Task 1	0.91555	0.91	0.92	0.91
Hindi	Task 2	0.78571	0.76422	0.90142	0.90001
Malayalam	Task 1	0.83444	0.79	0.83777	0.81
Malayalam	Task 2	0.62428	0.60677	0.74857	0.74597
Punjabi	Task 1	0.942	0.94	0.946	0.95
Punjabi	Task 2	0.812	0.8086	0.92266	0.923
Tamil	Task 1	0.83333	0.79	0.8333	0.79
Tamil	Task 2	0.65714	0.63067	0.755	0.73979

5. CONCLUSION AND FUTURE WORK

In this paper, we presented our NLP-NITMZ system used for DPIL shared task. Overall, our approach looks promising, but needs some improvement. There are some disadvantages of PNN like: require large memory, slow execution. In future we want to overcome those problems using better machine learning approach and also want to implement semantic features for all languages to increase performance. We can also identify stop words of all four languages so that we can omit them from the corpus. Since our approach is based on language independent feature set so our methodology can be extended to various languages.

6. ACKNOWLEDGMENTS

This work presented here is under the Research Project Grant No. YSS/2015/000988 and supported is by the Department of Science & Technology (DST) and Science and Engineering Research Board (SERB), Govt. of India. Authors are also acknowledges the Department of Computer Science & Engineering of National Institute of Technology Mizoram, India for providing infrastructural facilities and support.

7. REFERENCES

[1] Wu W.,Ju Y., Li X. and Wang Y. 2010. Paraphrase detection on SMS messages in automobiles. In *Acoustics Speech and Signal Processing (ICASSP)*, 2010 IEEE International Conference on (pp. 5326-5329).

[2] Dolan, B., and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*

[3] Sundaram, M. S., Madasamy, A. K., and Padannayil, S. K. 2005. AMRITA_CEN@SemEval-2015: Paraphrase Detection for Twitter using Unsupervised Feature Learning with Recursive Autoencoders. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 45-50

[4] Mahalakshmi, S., Anand Kumar and M., Soman, K.P. Paraphrase detection for Tamil language using deep learning algorithm, *In (2015) International Journal of Applied Engineering Research*, 10 (17), pp. 13929-13934

[5] Socher R., Huang E., Pennin J., Manning C. D. and And Ng A.Y. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in Neural Information Processing Systems* (pp. 801-809).

[6] Anand Kumar M., Singh, S., Kavirajan, B., and Soman, K P. 2016. DPIL@FIRE2016: Overview of shared task on Detecting Paraphrases in Indian Languages. In *Working notes of FIRE 2016 – Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings, CEUR-WS.org

[7] Pakray P. and Sojka P. 2014. An Architecture for Scientific Document Retrieval Using Textual and Math Entailment Modules. In *RASLAN 2014: Recent Advances in Slavonic Natural Language Processing*, Karlova Studánka, Czech Republic, December 5-7, 2014.

[8] Lynum, A., Pakray, P., Gamback, B. and Jimenez, S. 2014. NTNU: Measuring semantic similarity with sublexical feature representations and soft cardinality. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 448–453, Dublin, Ireland, August 23-24, 2014.

[9] Sarkar S, Das D, Pakray P., Gelbukh A., 2016. JUNITMZ at SemEval-2016 Task 1: Identifying Semantic Similarity Using Levenshtein Ratio. In *Proceedings of SemEval-2016*, pages 702–705, San Diego, California.

[10] Specht, D.F., 1990. Probabilistic neural networks. *Neural Networks* 3, 109 – 118

[11] Donald F. S. 1990. Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification. In *IEEE Transactions on Neural Networks*, vol. I. No. I, march 1990

[12] Hajmeer, M and Basheer, I. 2002. A probabilistic neural network approach for modeling and classification of bacterial growth/no-growth data. In *Journal of microbiological methods*, vol. 51, No. 2, 217—226.