# JU_NLP@DPIL-FIRE2016: Paraphrase Detection in Indian Languages - A Machine Learning Approach

Tanik Saikh Sudip Kumar Naskar Sivaji Bandyopadhyay
Computer Science and Engineering Department, Jadavpur University
Kolkata, India
tanik4u@gmail.com , {sudip.naskar, sbandyopadhyay}@cse.jdvu.ac.in

## ABSTRACT
This paper presents our system report on our participation in the shared task on "*Detecting Paraphrases in Indian Languages* (*DPIL*)" organized in the "*Forum for Information Retrieval Evaluation (FIRE)*"- *2016*, in both the tasks (*Task1 and Task2*) defined in this shared task in four Indian languages (Tamil, Malayalam, Hindi and Punjabi). We made use of different similarity measures and machine translation evaluation metrics as features and used machine learning framework to take paraphrase decision between a pair of text snippets. We obtained the accuracies of 97.08%, 94.2%, 97.32% and 98.29% in Task1 and 86.68%, 77.37%, 84% and 98.77% in Task2 for Tamil, Malayalam, Hindi and Punjabi respectively on the respective training sets using a 10-fold cross validation framework.

## Keywords
Paraphrase detection; Similarity Measures; Machine Learning.

## 1. INTRODUCTION
A paraphrase is a restatement of a text expressed using other words. Alternatively, two texts say text1 and text2 can be defined as paraphrases if they textually entail each other bi-directionally, i.e. if text1 entails text2 and text2 entails text1. Textual entailment and paraphrase relations between a pair of text snippets are highly correlated. There are two different tasks related to paraphrases - paraphrase identification and paraphrase generation. In this shared task [19] the focus is on identifying sentence level paraphrases in Indian languages namely Tamil, Malayalam, Hindi, and Punjabi. Two subtasks, Task1 and Task2, were defined in the shared task. Given a pair of sentences from newspaper domain, Task1 is to classify them as paraphrases (P) or not paraphrases (NP) and Task2 is to identify whether they are completely equivalent (E), not equivalent (NE) or roughly equivalent (RE). Task2 is similar to Task1 except that it is a ternary classification problem. Identifying paraphrase between a pair of sentence in Indian languages is not an easy task due to the scarcity of tools and resources in such languages. Semantic knowledge bases like WordNet [13] and BabelNet [10] are also very useful resources for knowledge based approaches to detecting paraphrases and textual entailment and Indian languages also suffer from very poor coverage in this respect. Therefore, for this study we adopted lexical level of analysis on the sentences, which is considered to be a shallow level of processing in any Natural Languages Processing task.

For both the tasks (i.e., Task1 and Task2) we made use of various kinds of lexical similarity measures namely Cosine similarity, unigram matching with respect to sentence1, unigram matching with respect to sentence2, Jaccard similarity [12], Dice coefficient [11], overlap, harmonic mean and machine translation (MT) evaluation metrics namely BLEU and METEOR. The scores of these measures were considered as feature values to build the models. The models were used to train the machine learning based classifiers. Naïve Bayes, SVM and SMO were employed for this purpose.

The work reported in [14] made use of same kinds of features in taking textual entailment decision between a pair of texts on the datasets released in the shared task on recognizing textual entailment in RTE-1, RTE-2 and RTE-3. In the present work we demonstrated that the same features and techniques are also effective in taking paraphrase decision between two sentences.

## 2. DATA
The shared task on detecting paraphrases in Indian languages (DPIL) defined two subtasks namely Task1 and Task2. Training datasets were provided for each of the subtasks in four Indian languages − Hindi, Panjabi, Tamil and Malayalam. The statistics of the training and test datasets are shown in table1 and table2 respectively.

**Table 1. Statistics of the training sets**

| Language | # of sentence pairs | | | | | | |
|---|---|---|---|---|---|---|---|
| | Task1 | | | Task2 | | | |
| | Total | P | NP | Total | E | NE | RE |
| Tamil | 2500 | 1000 | 1500 | 3500 | 1000 | 1500 | 1000 |
| Malayalam | 2500 | 1000 | 1500 | 3500 | 1000 | 1500 | 1000 |
| Hindi | 2500 | 1000 | 1500 | 3500 | 1000 | 1500 | 1000 |
| Punjabi | 1700 | 700 | 1000 | 2200 | 700 | 1000 | 500 |

**Table 2.Statistics of the test sets**

| Language | # of sentence pairs | |
|---|---|---|
| | Task1 | Task2 |
| Tamil | 900 | 1400 |
| Malayalam | 900 | 1400 |
| Hindi | 900 | 1400 |
| Punjabi | 500 | 750 |

## 3. Features
Features play a pivotal role in machine learning based frameworks. Therefore, analysis of features which take part in predicting the target class is very crucial. Features which have been used in this study can be broadly divided into two categories − similarity based features and MT evaluation metrics based features.

## 3.1 Similarity Based Features

For the present study, we considered different similarity based features such as vector based (cosine similarity, dice similarity), lexical based (unigram matching with respect to sentence1 and sentence2), set based (Jaccard, overlap and harmonic) which are discussed below.

### *3.1.1 Cosine Similarity*

Cosine similarity measures the similarity between two vectors of an inner product space that measures the cosine of angle between them. The lower the angle between the two vectors the more similar the two vectors are.

### *3.1.2 Unigram Matching*

Unigram (i.e., word) matches between two sentences are taken into consideration. Here two variations of the unigram matching are considered, i.e. unigram matching with respect to sentence1 and sentence2. These are calculated by the number of unigram matching between two sentences normalized by the number of unigrams in sentence1 and sentence2 respectively.

### *3.1.3 Jaccard*

Jaccard similarity is a set based measure which can be defined as

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

where A and B are two sets of element. It provides number of common elements between two sets.

### *3.1.4 Dice*

Dice is a vector based similarity measure the value of which lies between 0 to 1. It can be calculated by the following equation

$$Dice(A,B) = \frac{A \cap B}{(|A| + |B|)}$$

where A and B are two sets.

### *3.1.5 Overlap*

Overlap is a set based text similarity metric where a text can be represented as a set and the set elements are words. It is similar to *Dice* with a minor difference that it assumes a full match between two strings if one is subset of another. The similarity of this measure lies in the range of 0 to 1. It can be measured by the following equation.

$$Overlap(A,B) = \frac{|A \cap B|}{min(|A|,|B|)}$$

where A and B are two sets.

### *3.1.6 Harmonic*

Harmonic is also a set based similarity measure. It can be calculated by employing the following equation.

$$Harmonic(A,B) = \frac{|A \cap B|.(|A| + |B|)}{2.|A|.|B|}$$

where A and B are two sets.

## 3.2 Machine Translation Evaluation Metrics

Machine translation (MT) evaluation metrics are generally used to measure the closeness between the MT translation hypotheses and the reference translations. The closer a translation hypothesis is to the reference translation, the better the translation quality is. There are several MT evaluation metrics available like Word Error Rate (WER) [7], Position Independent word error rate (PER) [8], BLEU [1], METEOR [2] [3], Translation error/edit rate (TER) [4], NIST [6], General Text Matcher (GTM) [5] etc. Among these BLEU and METEOR are perhaps the most popular and widely used ones. In this present work, we made use of these two MT evaluation metrics as features to predict paraphrase relation between a pair of sentences.

### *3.2.1 BLEU*

*BiLingual Evaluation Understudy* (BLEU) is an algorithm to evaluate the quality of a machine translated text. It compares n-grams of the translation hypothesis with the n-grams of the reference translation(s) and counts the number of n-gram matches. It is essentially a modified n-gram precision measure. To avoid bias towards shorter translation candidates, BLEU uses a brevity penalty that penalizes candidate translations whose length differs significantly from that of the reference translation.

### *3.2.2 METEOR*

*Metric for evaluation of translation with explicit ordering* (METEOR) calculates n-gram overlaps between a translation hypothesis and the reference translation(s). If multiple reference translations are available for an MT output, the translation is scored against each reference translation independently and the best scoring pair is used for evaluation. Given a pair of sentences to be compared, METEOR creates a word alignment between the two sentences. An alignment is a mapping between the sentences such that every word in each sentence maps to at most one word in the other sentences. This alignment is incrementally produced by a sequence of word mapping modules which considers exact matching, stem matching and synonymy matching. Based on the number of mapped unigrams found between the two strings (m), the total number of unigrams in the translation (t) and the total number of unigrams in the reference (r), unigram precision is calculated as $P = {}^{m}/_{t}$ and unigram recall is calculated as $R = {}^{m}/_{r}$. Finally, it computes the F score as a parameterized harmonic mean of P and R as $F_{mean} = \frac{P*R}{\alpha*P+(1-\alpha)*R}$, where $\alpha$ is a constant. To address word ordering, METEOR calculates a reordering penalty based on how many chunks in the translation hypothesis need to be moved around to get the reference text. Finally the METEOR score for the alignment between the two sentences is calculated as follows.

$$score = (1 - Penaltyscore) * F_{mean}$$

## 4. System Description

We extract sentence pairs (say *sentence1* and *sentence2*) from the XML format training dataset for each language. An example training sentence pair is shown below from the Hindi dataset.

*<Paraphrase pID="HIN0001"><Sentence1>भारतीयमुस्लिमोंकीवजहसेनहींपनपसकताआईएस|</Sentence1><Sentence2>भारतमेंकभीवर्चस्वकायमनहींकरसकताआईएस|</Sentence2><Class>P</Class></Paraphrase>*.

In the above example the relation between *sentence1* and *sentence2* is paraphrase which is tagged as "*P*" in the training dataset. We calculate different similarity scores between the sentences pairs extracted from the training datasets released in the shared task. These scores are used as feature values to build the models for the four languages. Four models were prepared by combining different features for every language. Model1 considers only lexical features (LF), i.e., cosine similarity, unigram matching with respect to sentence1, unigram matching

with respect to sentence2, jaccard, dice, overlap and harmonic. Model2 considers LF and BLEU. LF and METEOR were considered for Model3. Model4 considers all the features, i.e., LF, BLEU and METEOR. These models were used to train three machine learning classifiers namely naïve bayes [17], support vector machine (SVM) [16] and sequential minimal optimization algorithm (SMO) [9]. Weka[1] [15] tool was used for this purpose which is freely available in web; the java implementations of various machine learning algorithms are available in this tool including the classifiers used for our experiments. We carried out experiments based on 10-fold cross validation on the training set. So our system can predict paraphrase class of an unknown sentence pair based on these learning. Finally, the classifier−model combinations producing the optimized results were applied on the corresponding test sets.

## 5. Results and Discussion

We took part in both the tasks (*Task1 and Task2*) defined in the shared task in all the four languages (Hindi, Punjabi, Tamil and Malayalam). Accuracies obtained on the training sets (using 10-fold cross validation) on Task1 and Task2 using the different classifiers are shown in Table 3 and Table 4 respectively.

**Table 3. Accuracies on training sets in Task1.**

| | Classifiers | Models | | | |
|---|---|---|---|---|---|
| | | LF | LF+B | LF+M | LF+B+M |
| Hindi | Naïve Byes | 90.04 | 92.52 | 90.16 | 74.97 |
| | SVM | 90.36 | 92.52 | 90.36 | 82.85 |
| | SMO | 90.84 | 97.32 | 90.84 | 84 |
| Tamil | Naïve Byes | 92.44 | 94.16 | 92.2 | 93.92 |
| | SVM | 93.24 | 94.44 | 93.24 | 94.24 |
| | SMO | 76.71 | 97.04 | 93.36 | 97.08 |
| Malayalam | Naïve Byes | 83.48 | 88.36 | 83.56 | 87.88 |
| | SVM | 84.24 | 86.88 | 83.96 | 86.56 |
| | SMO | 84.68 | 94.2 | 84.72 | 94.2 |
| Punjabi | Naïve Byes | 97.64 | 97.64 | 97.70 | 97.58 |
| | SVM | 97.76 | 97.82 | 97.70 | 97.7 |
| | SMO | 98.17 | 98.23 | 98.23 | 98.29 |

In Task1 the best accuracy 97.32% was obtained for Hindi in SMO with LF+B model. Tamil achieved the highest accuracy of 97.08% with the LF+B+M model in SMO. Malayalam resulted in the highest accuracy of 94.2% with both LF+B and LF+B+F models in SMO, whereas in Punjabi we got the highest accuracy of 98.29% with LF+B+M model by SMO. Thus SMO provided the optimum results in all four languages on the training datasets.

In Task2, the SMO classifier and LF+B+M model combination produced the best accuracies for all the four languages − Hindi (84%), Tamil (86.68%), Malayalam (77.37%) and Punjabi (98.77%). For Punjabi the LF+B model (on SMO) also achieved the highest accuracy along with the LF+B+M model.

The features which are used in these experiments are independent of each other. Naïve Bayes makes the simplification assumption that features are independent to given class [17].

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

**Table 4. Accuracies on training sets in Task2.**

| | Classifiers | Models | | | |
|---|---|---|---|---|---|
| | | LF | LF+B | LF+M | LF+B+M |
| Hindi | Naïve Byes | 73.14 | 75.37 | 73.45 | 74.97 |
| | SVM | 82.6 | 82.82 | 82.71 | 82.85 |
| | SMO | 82.88 | 83.17 | 83.37 | 84 |
| Tamil | Naïve Byes | 71.6 | 78.82 | 71.25 | 78.54 |
| | SVM | 76.57 | 77.77 | 76.54 | 77.77 |
| | SMO | 76.71 | 86.4 | 77.11 | 86.68 |
| Malayalam | Naïve Byes | 65.02 | 72.65 | 65.54 | 72.22 |
| | SVM | 67.31 | 68.51 | 67.25 | 68.48 |
| | SMO | 68 | 77.25 | 68 | 77.37 |
| Punjabi | Naïve Byes | 93.81 | 97.5 | 94.45 | 97.31 |
| | SVM | 96.22 | 98.04 | 97.09 | 98.04 |
| | SMO | 97.18 | 98.77 | 97.72 | 98.77 |

SVMs are comparatively new machine learning approaches for solving two class pattern recognition problems. In the field of NLP, SVMs have been employed for many tasks including text classification and are reported to have obtained high accuracy without falling into overfitting even with a large number of features [18]. Since SVMs perform better in binary classification problems, our results in Task1 (binary classification problem) also outperforms the results obtained in Task2 (ternary classification problem). SMO is essentially another way of expressing SVMs which implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. It makes use of heuristics to partition the training problem into smaller problems.

The official results of our submissions released by the task organizers on the test sets for the two subtasks in four languages are reported in Table 5.

**Table 5. Results on test sets in Task1 and Task2.**

| Languages | Task | Count | Accuracy | F1 Measure/ Macro F1 Measure |
|---|---|---|---|---|
| Hindi | Task1 | 900 | 0.8222 | 0.74 |
| Hindi | Task2 | 1400 | 0.68571 | 0.6841 |
| Malayalam | Task1 | 900 | 0.59 | 0.16 |
| Malayalam | Task2 | 1400 | 0.42214 | 0.3078 |
| Punjabi | Task1 | 500 | 0.942 | 0.94 |
| Punjabi | Task2 | 750 | 0.88666 | 0.88664 |
| Tamil | Task1 | 900 | 0.57555 | 0.09 |
| Tamil | Task2 | 1400 | 0.55071 | 0.4319 |

## 6. Conclusions

The paper presents our submissions in the *DPIL* shared task organized in *FIRE 2016*. We took part in both the subtasks in all four languages. Different lexical level similarity measures and two machine translation evaluation metrics namely BLEU and METEOR were employed as features to find similarity scores between pair of sentences. Four different models were built combining these features. The models were used to train three classifiers namely naïve bayes, SVM and SMO. We carried out experiments using 10-fold cross validation framework on the

training sets. The SMO classifier produced the optimum results when all the features were combined.

## 7. Acknowledgements

The research work has received funding from the project "Development of Tree Bank in Indian Languages" funded by The Department of Electronics and Information Technology (DeitY), Ministry of Communication and Information Technology, Government of India.

## 8. REFERENCES

[1] Papineni, K., Roukos, S., Ward, T., and Zhu, W.J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, pp. 311–318.

[2] Banerjee, S. and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, Michigan, pp. 65–72.

[3] Lavie, A. and Agarwal, A. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the Second ACL Workshop on Statistical Machine Translation, Prague, Czech Republic, pp. 228–231.

[4] Snover, M., Dorr, B., Schwartz, R., Micciulla, and L., Makhoul, J. 2006: A study of translation edit rate with targeted human annotation. In Proceedings of Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp. 223–231.

[5] Turian, J.P., Shen, L., and Dan Melamed, I. 2003. Evaluation of Machine Translation and Its Evaluation. In Proceedings of MT Summit, New Orleans, Luisiana, pp. 386–393.

[6] Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In Proceedings of the Second International Conference on Human Language Technology Research, Morgan Kaufmann Publishers Inc, pp. 138–145.

[7] Vidal, E.1997. Finite-State Speech-to-Speech Translation. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 111–114.

[8] Tillmann, C., Vogel, S., Ney, H., Sawaf, H., and Zubiaga, A. 1997. Accelerated DP based Search for Statistical Translation. In Proceedings of the 5th European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 2667–2670.

[9] John C. Platt. 1999. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In Bernhard Schölkopf, Christopher J. C. Burges and Alexander J. Smola (eds.). 1999. Advances in Kernel Methods: Support Vector Learning. The MIT Press, Cambridge, MA, pp. 185-208.

[10] Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence, pp. 193:217– 250.

[11] Dice, L. 1945. Measures of the amount of ecologic association between species. Ecology, vol 26, no 3, pp. 297-302.

[12] Jaccard, P. 1901. Étude comparative de la distribution floraledansune portion des Alpeset des Jura. Bulletin de la SociétéVaudoise des Sciences Naturelles 37, pp. 547-579.

[13] Fellbaum, Christine, ed.1998.WordNet: An Electronic Lexical Database, Cambridge, MA: MIT Press.

[14] T Saikh, SK Naskar, C Giri, and S Bandyopadhyay. 2015. Textual Entailment Using Different Similarity Metrics in Computational Linguistics and Intelligent Text Processing, pp. 491-501.

[15] I.H. Witten and E. Frank. 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition.

[16] Vapnik, Vladimir N. 1995. The Nature of Statistical Learning Theory. Springer.

[17] I. Rish. 2001. An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence, vol 22, pp. 41-46.

[18] AEkbal and S Bandyopadhyay. 2008. Bengali Named Entity Recognition Using Support Vector Machine. In International Joint Conference on Natural Language Processing (IJCNLP), pp. 51-58

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