Tamil Paraphrase Detection using Long-Short Term Memory Networks

B. Senthil Kumar, D. Thenmozhi, Chandrabose Aravindan, S. Kayalvizhi
SSN College of Engineering
Chennai

theni_d@ssn.edu.in

Abstract. Paraphrase detection, one of challenging task in NLP is to detect whether the given pair of sentences which are rephrased or with word reordering are preserving the meaning semantically. Paraphrase detection for Indian languages especially for Tamil, one of the Dravidian language which is agglutinative is a challenging task. In this paper, we present the Paraphrase detection for Tamil language using Long-Short Term Memory (LSTM) neural networks. Eventhough the state-of-art systems used traditional learning techniques, it suffers from carefull hand-crafted feature engineering which require pure lexical, syntactic or lexico-syntactic features of the language. To alleviate this problem, deep LSTM is used to train our system which considers the pair of sentences and predicts it as a Paraphrase (P) or Non-paraphrase (NP). Our system performs 15.2% better than the existing system using deep neural networks.

1. Introduction

The ability to detect similar sentences written in natural language is crucial for several applications, such as text mining, text summarization, plagiarism detection, authorship authentication, query ranking and question answering. Paraphrase can be identified, generated or extracted. Paraphrase identification can be used in generation system to choose the best from the list of candidates generated by the paraphrase generation system. It also plays vital role in validating the paraphrase extraction system and machine translation systems. Detecting redundancy is a very important issue for a multi-document summarization system because two sentences from different documents may convey the same semantic content and to make summary more informative, the redundant sentences should not be selected in the summary.

In this paper, the focus is to identify the paraphrase sentences from the DPIL corpus for Tamil language. The shared task on Detecting Paraphrases in Indian Languages (DPIL)@FIRE 2016 [1][2] was a good effort towards creating benchmark data for paraphrases in Indian Languages. Identifying the paraphrases in Indian languages especially for Tamil is a difficult task because evaluating the semantic similarity of the underlying content and understanding the morphological variations of the language are more critical.

Our main contributions in this work are:

(i) We propose a deep neural architecture for Tamil paraphrase detection using LSTM enhanced with attention mechanisms.

(ii) We evaluate this model in a monolingual setting, performing paraphrase detection on standard DPIL datasets, to assess the suitability of this model type for the paraphrasing task.

(iii) We compare the performance of our model to the state-of-the-art Tamil paraphrase detection model using deep learning, including the detailed analysis.

2. Related Work

Out of ten teams who submitted results in the DPIL shared task, five teams had submitted their results for Tamil paraphrase detection. Kong et al., [8] submitted results for all the four Indian languages – namely

Tamil, Malayalam, Hindi and Punjabi. They have used Cosine Distance, Jaccard Coefficient, Dice Distance and METEOR features and classification is done based on Gradient Tree Boosting. They achieved the overall best score across all the four languages. The Tanik et al., [9] used similarity based features, word overlapping features and scores from the machine translation evaluation metrics to find out the similarity scores between pair of sentences. They tried with three different classifiers namely Naïve Bayes, SVM and SMO.

The Tamil Shallow parser was used by Thangarajan et al., [10] to extract the morphological features of language and applied Support Vector Machine (SVM) and Maximum Entropy to classify Tamil paraphrases. They submitted results only for Tamil language with 82% accuracy. Kamal Sarkar [11] had submitted results for all the four languages. He used different lexical and semantic level (Word embeddings) similarity measures for computing features and used multinomial logistic regression model as a classifier. His model performed 78% accuracy for the Tamil language. Sarkar et al., [12] used the features based on Jaccard Similarity, length normalized Edit Distance and Cosine similarity. These feature-set are trained using Probabilistic Neural Network (PNN) to detect the paraphrases. For Tamil language, the system achieved 83.33% accuracy in subtask1.

All the above models applied the traditional learning techniques by using the lexical, syntactic or semantic similarity feature sets. Mahalakshmi et al., [3] used recursive auto-encoders (RAE) to represent the feature vectors for Tamil paraphrase detection. Initially the sentence pairs are parsed using the shallow parser and its word, phrase vectors are computed by RAE for learning feature vectors for phrases in syntactic trees in an unsupervised way. The model then used the Euclidean distances to measure the similarity and then softmax classifier is used to detect the paraphrases.

3. Dataset

We have used the Tamil paraphrase pair of sentences from the DPIL@FIRE2016 shared task. The shared task required participants to identify sentential paraphrases in four Indian languages – Hindi, Punjabi, Malayalam and Tamil. In this shared task, there were two sub-tasks: task1 is to classify a given pair of sentences as paraphrases (P) or not paraphrases (NP) and task2 is to identify whether a given pair of sentences are completely paraphrases (P) or semi-paraphrases (SP) or not paraphrases (NP). The evaluation dataset is mainly obtained from the newspaper. The details of this corpus can be found in http://nlp.amrita.edu/dpil_cen/. The corpora are divided into two different subsets. We used Task-1 subset for Tamil language, which categorizes 2500 paraphrase Tamil sentence pairs into one of binary class and with 900 test pairs.

The corpus statistics showed that the vocabulary sizes for Hindi & Punjabi languages are less than Tamil and Malayalam. This is because the Dravidian languages, Tamil and Malayalam are agglutinative in nature. Due to this phenomenon, Dravidian languages end up by having more unique words and hence larger vocabulary. The size of vocabulary for Tamil is around 17K for 2500 pairs of train sentences. This agglutinative phenomenon of Tamil language increases the complexity in detecting the paraphrases.

4. Proposed Methodology

To detect the paraphrases, we adopted the NMT architecture [4]. The model consists of embedding layer, encoders, decoders which uses Bi-LSTM layers and attention mechanism on top of encoders, projection layer on top of decoders to predict the class label. We treated this model as classification system to predict the class label as Paraphrase (P) or Non-Paraphrase (NP) for the given pair of input sentences. The embedding layer creates the vocabulary for both the input and the output. Here the input is the pair of sentences and the output is the binary class label P and NP. The dataset annotated the paraphrases using XML format for each language. The following is the sample paraphrase in Tamil:

<Paraphrase pID="TAM0001">

<Sentence1> சங்கராபுரம் தொகுதியில் போட்டியிடும் ஸ்டாலின் நடைபயணமாக சென்று பிரசாரம் செய்தார். </Sentence1>

<Sentence2> தி.மு.க., வேட்பாளர் ஸ்டாலின் போட்டியிடும் சங்கராபுரம் தொகுதியில் சின்ன சேலம் பகுதியில் நடைபயணமாக சென்று ஓட்டு சேகரித்தார். </Sentence2>

<Class> P </Class>

</Paraphrase>

Each paraphrase was assigned a unique ID followed by the two sentences marked by sentence number tags and the gold class label tag. This input pair of sentences are extracted from the dataset, preprocessed and presented as input sequences to encoder in the model as below:

<s> சங்கராபுரம் தொகுதியில் போட்டியிடும் ஸ்டாலின் நடைபயணமாக சென்று பிரசாரம் செய்தார். <eol> தி.மு.க., வேட்பாளர் ஸ்டாலின் போட்டியிடும் சங்கராபுரம் தொகுதியில் சின்ன சேலம் பகுதியில் நடைபயணமாக சென்று ஓட்டு சேகரித்தார் .</s>

The input pair of sentences are delimited with <eol> which acts as a boundary marker between the sentences pair and given as input to the Bi-LSTM units. The time-based Bi-LSTM units which act as an encoder generates a context vector for the given input pair of sentences which is then mapped to the corresponding class in output during the training. The attention mechanism on top of encoder Bi-LSTM units calculates the weights which select the set of inputs that contribute in predicting the output class label. During the testing, the unseen paraphrase sentences are given as input to the model. The encoding Bi-LSTM units that generate the context vector is given as input to decoder and act as an initializer to the decoding Bi-LSTM units. The context vector, the previous output and current input are given as input to the decoder Bi-LSTM units to predict the class label. From our earlier experiments, we resorted to use one Bi-LSTM layer as encoder and decoder. The performance of the system varies with the number of layers of Bi-LSTM units, number of epochs, type of attention mechanisms, the data size which generated the vocabulary and other hyper parameters to the model.

5. Result

To evaluate the system, we have considered 5-fold cross validation on the training data. We tried with two types of attention mechanisms – Normed Bahdanau (NB) and Scaled Luong (SL) on top of encoders. For the given 2500 pair of training sentences, the model accuracy is measured by dividing the training dataset into 5 folds. For each fold 500 pairs are considered for testing and the remaining 2000 pairs are considered as training data. The overall performance of the system for 5 folds is 65.2% accuracy and is shown in Table 1.

Table 1. Performance of Models

Sl No	n-Fold	Model Accuracy(%)	
		NB	SL
1	1-Fold	60.8	67.0
2	2-Fold	64.4	62.0

3	3-Fold	68.6	68.4
4	4-Fold	63.0	63.6
5	5-Fold	68.2	65.0
	Overall	65.0	**65.2**

It is observed from the results that both the two models – NB and SL – performed almost similar. We have compared the results of Mahalakshmi et al., [3] who have reported on deep learning approach for paraphrase detection in Tamil. They have also considered 2000 pairs for training and 500 pairs for evaluating the performance. The result comparison of our approach with the existing approach related to deep learning method for Tamil paraphrase is shown in Table 2.

Table 2. Performance Comparison

Methodology	**Accuracy (%)**
	Overall
Mahalakshmi et al., [3]	50
Our method - NB	65
Our method - SL	**65.2**

The major focus of work by Mahalakshmi et al., is to detect the Tamil Paraphrases correctly and they reported 65.17% of accuracy in detecting paraphrases and 34.83% for detecting non-paraphrases. On overall the accuracy of the system [3] is 50%. It is observed from Table 2 that our two approaches have improved the overall accuracy by 15.2%.

5.1. Detailed Analysis

Here is the deep analysis of our model. The confusion matrix for both of our approaches NB and SL are given in Table 3 and Table 4 respectively.

Table 3. Confusion matrix (NB)

Folds / Evaluation	**Fold 1**		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
	P	**NP**	**P**	**NP**	**P**	**NP**	**P**	**NP**	**P**	**NP**
P	119	131	129	121	**150**	100	130	120	149	101
NP	65	185	57	193	57	**193**	65	185	58	192

Table 4. Confusion matrix (SL)

Folds / Evaluation	**Fold 1**		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
	P	**NP**	**P**	**NP**	**P**	**NP**	**P**	**NP**	**P**	**NP**

P	127	123	132	118	**141**	109	145	105	148	102
NP	42	208	72	178	49	**201**	77	173	73	177

From Table 3 and Table 4, it is obvious that the accuracy of fold 3 is maximum in NB and SL systems. The average accuracy is slightly better in SL system because of an improvement in precision by 0.25 and recall by 1.28 when compared with NB system as shown in Table 5 and Table 6.

Table 5. Performance of NB

Folds/Metrics	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Avg**
Precision (%)	64.67	69.35	72.46	66.67	71.9	69.01
Recall (%)	47.6	51.6	60	52	59.6	54.16
F1 (%)	54.84	59.17	65.65	58.43	65.2	60.66
Accuracy (%)	60.8	64.4	68.6	63	68.2	65

Table 6. Performance of SL

Metrics/Folds	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Avg**
Precision (%)	75.15	64.71	74.21	65.32	66.9	69.26
Recall (%)	50.8	52.8	56.4	58	59.2	55.44
F1 (%)	60.62	58.15	64.09	61.4	62.8	61.41
Accuracy (%)	67	62	68.4	63.6	65	65.2

The system with scaled-luong attention mechanism (SL) performed slightly better than normed-bahdanau attention system (NB).

Several research works have reported on paraphrase detection for Indian languages including Tamil. However they have used traditional learning techniques with lexical, syntactic and lexico-syntactic and word embedding features to develop their systems on DPIL@FIRE2016 dataset. The shared task reported that the traditional learning techniques yield 83.33% of accuracy for Tamil language, whereas our deep learning approach gives 56.4% & 49% of accuracies on test data for NB and SL models respectively. This is due to the limitations in the size of the corpus in Tamil.

5.2. Error Analysis

The models suffer from the data sparseness problems because specific paraphrase instances occur only a handful of times in the training set. Consider the following instance from the test data:

Sentence number: 11

இந்திய விடுதலைப்போராட்டம் 1857ல் வேலூர்ப்புரட்சியின் போதே ஆரம்பித்துவிட்டது eol 1857ல் நடைபெற்ற வேலூர்ப்புரட்சியிலிருந்தே இந்திய விடுதலைப்போராட்டமானது உயிர்கொள்ள ஆரம்பமானது

The above is classified as P (Paraphrase) in gold target, whereas our systems (both NB and SL) predicted it as NP (Non paraphrase). Because NMT learns word representations in continuous space, it tends to translate (map) the words that are frequent in context [13]. Most of the words like விடுதலைப்போராட்டம், வேலூர்ப்புரட்சி, ஆரம்பித்துவிட்டது, உயிர்கொள்ள, ஆரம்பமானது, etc., in the test instance 11 are of type UNK where there is not even a single occurrence in the training data. Whereas the word இந்திய has occurred 52 times in P, 125 times in NP sentences in training data. Similarly another word நடைபெற்ற has its appearance of 28 times in P and 44 in NP sentences. Both these words appeared more frequently in the context of NP sentences rather than P sentences of the training data.

6. Conclusion

We used the LSTM-based deep neural network model to detect the Tamil paraphrase from the DPIL corpus. We evaluated our system for 5 folds on the given training data of 2500 instances. We developed two variations with respect to attention techniques on the deep neural network – system using scaled-luong (SL), normed bahdanau (NB) as attention mechanisms. Among these, SL showed the overall accuracy of 65.2% on the training data of the DPIL corpus for Tamil paraphrase detection. Eventhough the state-of-the-art systems for Tamil paraphrase detection have used the traditional learning techniques, it suffers from heavy hand-crafted feature engineering. Whereas deep neural network systems alleviate this need of lexico-syntactic features. The performance of this system can be improved further with more number of training instances. Since the deep neural network systems require more data to be trained, the DPIL training instances for Tamil paraphrase – 2500 – was not enough for the deep neural network model to capture and learn the syntactic feature of the language automatically from the given instances.

References

1. Anand Kumar, M., Singh, S., Kavirajan, B., Soman, K.P., DPIL@FIRE 2016: Overview of shared task on detecting paraphrases in Indian Languages (DPIL), CEUR Workshop Proceedings, 1737, pp. 233-238, 2016.
2. Anand Kumar M., Singh S., Kavirajan B., Soman K.P., Shared Task on Detecting Paraphrases in Indian Languages (DPIL): An Overview. In: Majumder P., Mitra M., Mehta P., Sankhavara J. (eds) Text Processing. FIRE 2016. Lecture Notes in Computer Science, vol 10478. Springer, 2016.
3. Mahalakshmi, S., M. Anand Kumar, and K. P. Soman., Paraphrase detection for Tamil language using deep learning algorithm, International Journal of Applied Engineeing Reseseach, Volume 10, no. 17, pp: 13929-13934, 2015.
4. Minh-Thang Luong, Eugene Brevdo, Rui Zhao, Neural Machine Translation (seq2seq) Tutorial, https://github.com/tensorflow/nmt, 2017.
5. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, Neural machine translation by jointly learning to align and translate, ICLR, 2015.
6. Minh-Thang Luong, Hieu Pham, and Christopher D Manning, Effective approaches to attention-based neural machine translation. EMNLP, 2015.
7. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le., Sequence to sequence learning with neural networks, NIPS, 2014.
8. Kong, Leilei, Kaisheng Chen, Liuyang Tian, Zhenyuan Hao, Zhongyuan Han, and Haoliang Qi. HIT2016@ DPIL-FIRE2016: Detecting Paraphrases in Indian Languages based on Gradient Tree Boosting, In FIRE (Working Notes), pp. 260-265, 2016.

9. Saikh, Tanik, Sudip Kumar Naskar, and Sivaji Bandyopadhyay, JU_NLP@ DPIL-FIRE2016: Paraphrase Detection in Indian Languages-A Machine Learning Approach, In FIRE (Working Notes), pp. 275-278, 2016.
10. Thangarajan, R., S. V. Kogilavani, A. Karthic, and S. Jawahar, KEC@ DPIL-FIRE2016: detection of paraphrases on Indian languages, In FIRE (Working Notes), pp. 282-288, 2016.
11. Sarkar, Kamal. KS_JU@ DPIL-FIRE2016: detecting paraphrases in Indian languages using multinomial logistic regression model, In FIRE (Working Notes), pp. 250-255, 2016.
12. Sarkar, Sandip, Saurav Saha, Jereemi Bentham, Partha Pakray, Dipankar Das, and Alexander F. Gelbukh, NLP-NITMZ@ DPIL-FIRE2016: Language Independent Paraphrases Detection, In FIRE (Working Notes), pp. 256-259, 2016.
13. Nguyen, Toan and Chiang, David, Improving Lexical Choice in Neural Machine Translation, In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, ACL, pp. 334-343, 2018.