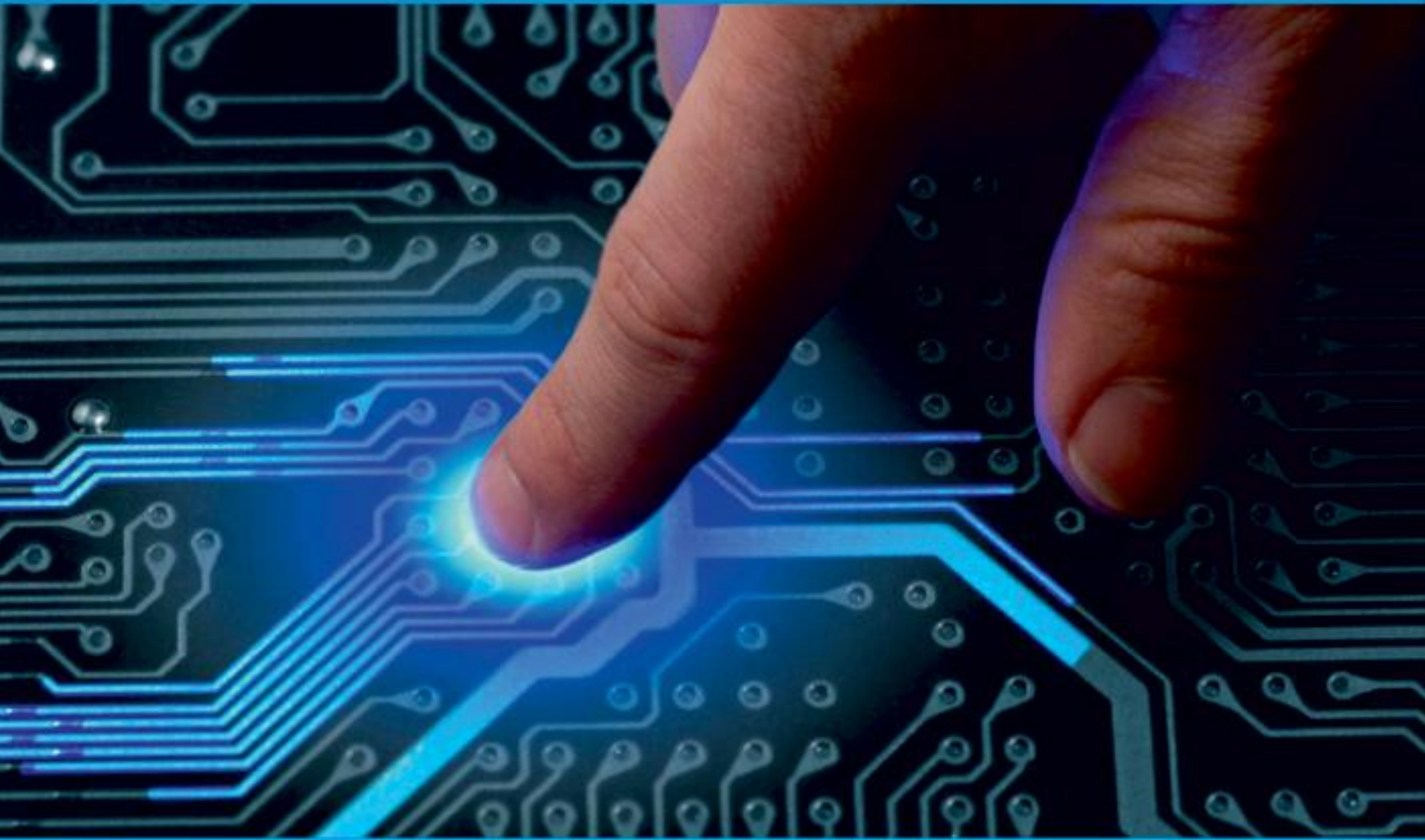
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907222 |

Paraphrase Detection in Indian Languages Using Deep Learning

Dr M Usha, Monisha D, Nivedidha K, Asmitha S

Assistant Professor, Dept. of CSE, Velammal Engineering College, Chennai, India

UG Student, Dept. of CSE, Velammal Engineering College, Chennai, India

UG Student, Dept. of CSE, Velammal Engineering College, Chennai, India

UG Student, Dept. of CSE, Velammal Engineering College, Chennai, India

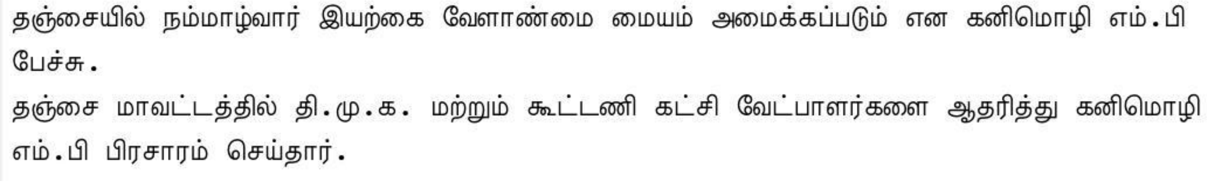
ABSTRACT: Paraphrases are different sentences conveying the same meaning. Paraphrase detection is the process of detecting whether the given sentences convey the same meaning. Paraphrase detection has a wide range of applications in areas such as plagiarism detection, test summarization, text mining, question answering, query ranking etc. Indian languages like Tamil, Hindi, Punjabi, Malayalam, etc., have a wide range of complex structure and vocabulary. So, it is difficult for a system to understand the semantics of these languages. Existing approaches for paraphrase detection have used machine learning techniques like multinomial logistic regression models and recursive autoencoders. These approaches lack hand crafted feature engineering whereas deep learning solves this problem. Many deep learning techniques have been introduced to address this problem. In our system two different deep learning algorithms namely BERT and USE have classified the sequences as paraphrase, semi paraphrase or non-paraphrase. Our system was evaluated on DPIL corpus and achieved the highest accuracy of 85.22% and 85.80% in task1 for Hindi and Punjabi languages respectively.

KEYWORDS: Deep learning, BERT, USE

I.INTRODUCTION

Languages are a universal means of communication for conveying the feeling of a human being to another human being. Paraphrase is one of the semantics of a language. A paraphrase is a restatement of the meaning of a text or passage using other words. A paraphrase detection system can be used in a variety of areas like plagiarism detection which can be applied in news articles and research papers to validate if the content is duplicated. Paraphrase detection can also be implemented in question answering systems to evaluate the correctness of answers as different sentences can convey the same meaning and different people express the same content using different sentences. In this work, the attempt to identify paraphrases in Indian languages using the DPIL corpus for four Indian languages namely Tamil, Malayalam, Hindi, Punjabi. Two tasks are involved in this system. Using deep learning algorithms namely BERT and USE, the attempt to classify the given sentences as paraphrases or not in Task1. In Task2, identifies the given sentences as either a paraphrase or semi-paraphrase or non-paraphrase. The following example improvises more on our proceedings.

Example: Consider the two sentences



தஞ்சையில் நம்மாழ்வார் இயற்கை வேளாண்மை மையம் அமைக்கப்படும் என கனிமொழி எம்.பி பேச்சு.

தஞ்சை மாவட்டத்தில் தி.மு.க. மற்றும் கூட்டணி கட்சி வேட்பாளர்களை ஆதரித்து கனிமொழி எம்.பி பிரசாரம் செய்தார்.

II.LITERATURE SURVEY

2.1 Paraphrase Detection in Indian Languages

a) Senthil Kumar et al. [2] performed the Paraphrase detection for Tamil language using Long-Short Term Memory (LSTM) neural networks. They adopted NMT architecture. The model they used consists of an embedding layer, encoders, decoders that used Bi-LSTM layers and attention mechanisms on top of the decoders to predict the class label.

b) Kamal Sarkar [6] used a paraphrase detection method that uses a multinomial logistic regression model trained with a variety of features which are basically lexical and semantic level similarities between two sentences in a pair. He did paraphrase detection for Tamil, Hindi, Malayalam and Punjabi languages. Similarity measures such as Cosine similarity, Word Overlap-exact match, N-gram based similarity and semantic similarity are used. He used a multinomial logistic regression classifier for paraphrase detection. The performance of this system evaluated using f1 measure is good for Punjabi and Hindi language and relatively low for Tamil and Malayalam languages due to the semantic complexities of their vocabulary.

III. WORKFLOW

To detect paraphrases in the sentences, two models are used namely, BERT (Bidirectional Encoder Representations from Transformers and USE (Universal Sentence Encoder). These models are built based on Recurrent Neural Network (RNN) using Deep Learning methods. These models are tested on four languages namely Tamil, Punjabi, Malayalam, Hindi and label the sentences of the languages as paraphrase or non-paraphrase or semi-paraphrase. After complete execution ofthese models the probability scores are obtained. Ensemble is done based on the confidence scores and the classification is performed using the prediction corresponding to the highest confidence score.

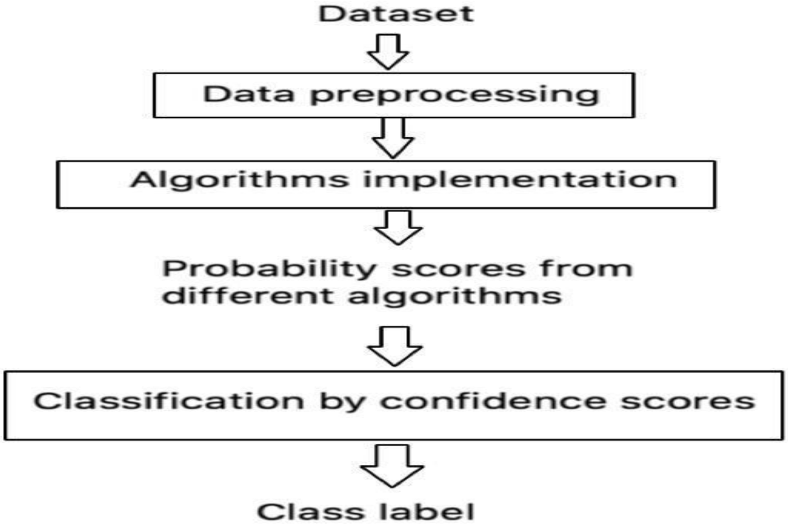

Dataset
Data preprocessing
Algorithms implementation
Probability scores from different algorithms
Classification by confidence scores
Class label

FIGURE 3.1: Workflow

IV. PROPOSED WORK

4.1 System Architecture

The System involves 2 models Bidirectional Encoder Representations from Transformers (BERT) and Universal Sentence Encoder (USE). The formulated data is given as an input to the two models which is processed by the algorithms and provides a probability score.The classifier outputs the class label according to the confidence values of the individual predictions. The classifier algorithm is implemented in such a way that the label corresponding to the prediction with the highest value is taken as the output.

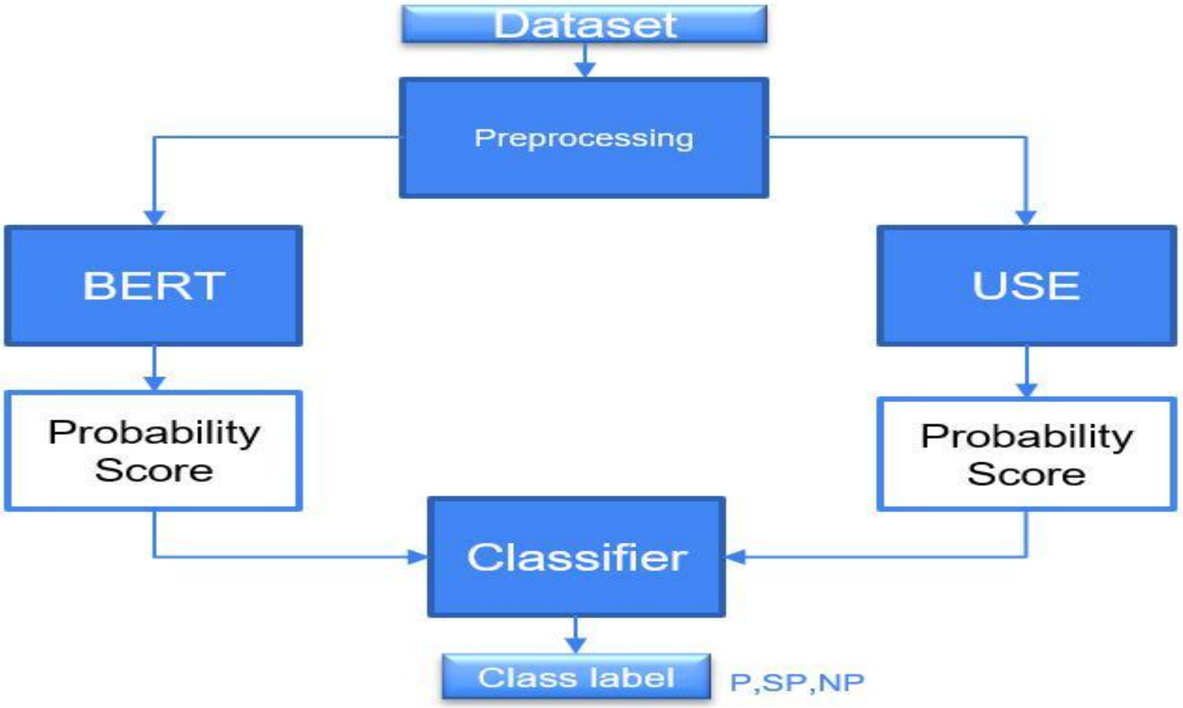


FIGURE 4.1: System architecture

4.2 System Description

a) **BERT Model:**

BERT is the first fine-tuning-based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many task-specific architectures. BERT makes use of Transformer, an attention mechanism that learns contextual relations betweenwords (or sub-words) in a text.

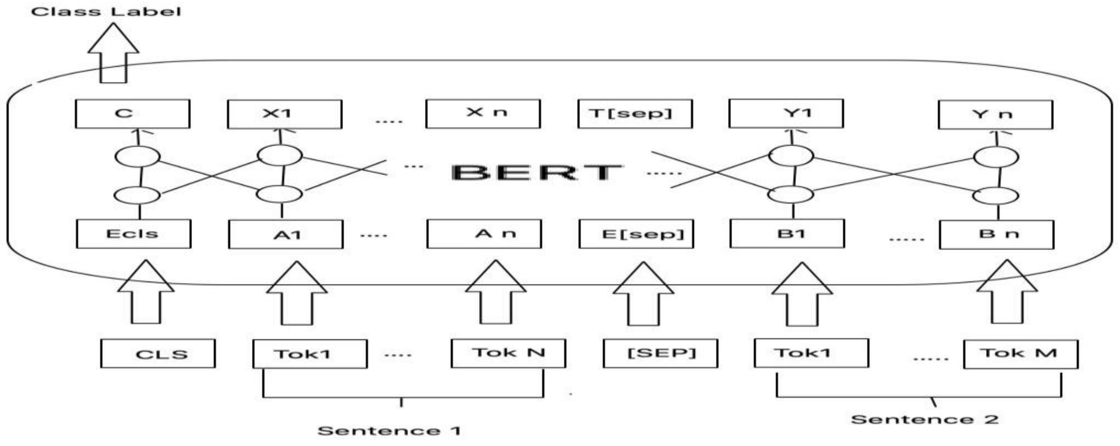


FIGURE 4.2: BERT architecture

b) Universal Sentence Encoder (USE) Model:

The Universal Sentence Encoder encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. It comes with two variations i.e., one trained with Transformer encoder and other trained with Deep Averaging Network (DAN). Transformer architecture targets high accuracy at the cost of greater model complexity and resource consumption. DAN targets efficient inference with slightly reduced accuracy. The transformer-based sentence encoding model constructs sentence embeddings using the encoding subgraph of the transformer architecture. In Deep averaging network (DAN) input embeddings for words and bi-grams are first averaged together and then passed through a feedforward deep neural network (DNN) to produce sentence embeddings.

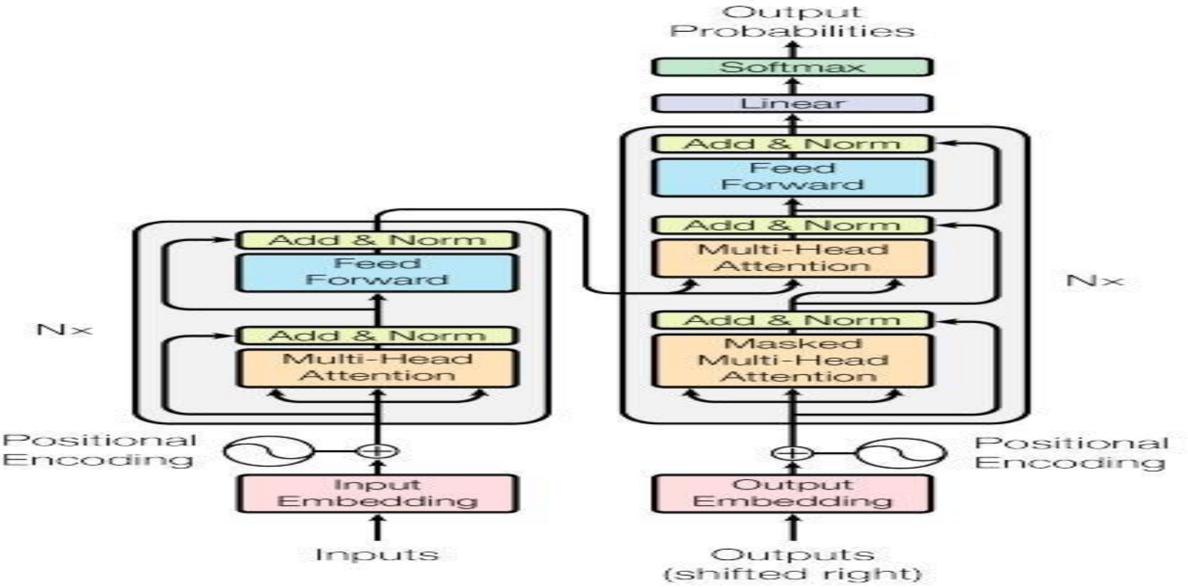


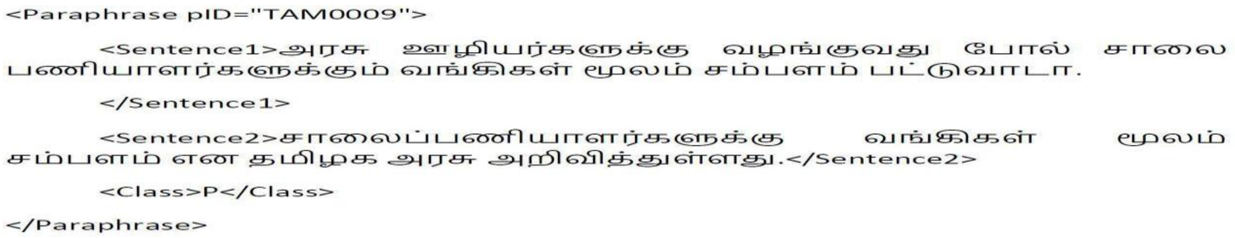
FIGURE 4.3: USE architecture

4.3 Dataset

The dataset from DPIL@FIRE2016 [1] is used. The dataset contains sentence pairs in four Indian languages namely Tamil, Malayalam, Hindi and Punjabi. Task1 is to identify if the sentences are paraphrases (P) or non paraphrases (NP). Task2 is to identify if the sentences are paraphrases (P) or non paraphrases (NP) or semi paraphrases (SP). The evaluation dataset is obtained mostly from news 16 articles. The details of this corpus can be found in http://nlp.amrita.edu/dpilcen/.

The data set is available in XML format. Every sentence pair is assigned an ID and is classified as P or SP or NP.

Sample Training Data:

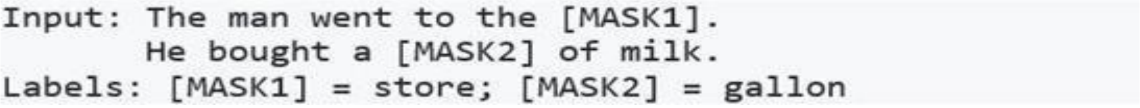


```
<Paraphrase pID="TAM0009">
        <Sentence1>அரசு ஊழியர்களுக்கு வழங்குவது போல் சாலை பணியாளர்களுக்கும் வங்கிகள் மூலம் சம்பளம் பட்டுவாடா.
        </Sentence1>
        <Sentence2>சாலைப்பணியாளர்களுக்கு வங்கிகள் மூலம் சம்பளம் என தமிழக அரசு அறிவித்துள்ளது.</Sentence2>
        <Class>P</Class>
</Paraphrase>
```

V. ALGORITHM AND TECHNIQUES

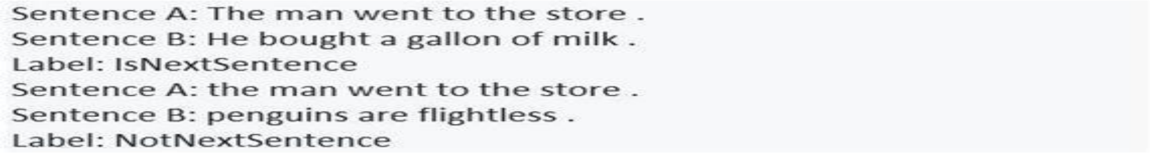
5.1 BERT Algorithm:

BERT is a natural language processing pre-training approach that can be used on a large body of text. It handles tasks such as entity recognition, part of speech tagging, and question-answering among other natural language processes. The BERT algorithm (Bidirectional Encoder Representations from Transformers) is a deep learning algorithm related to natural language processing. It helps a machine to understand what words in a sentence mean, but with all the nuances of context. BERT uses a simple approach for this: mask out 15% of the words in the input, run the entire sequence through a deep bidirectional Transformer encoder, and then predict only the masked words. For example:



```
Input: The man went to the [MASK1].
       He bought a [MASK2] of milk.
Labels: [MASK1] = store; [MASK2] = gallon
```

Given two sentences A and B, is B the actual next sentence that comes after A, or just a random sentence from the corpus.



```
Sentence A: The man went to the store .
Sentence B: He bought a gallon of milk .
Label: IsNextSentence
Sentence A: the man went to the store .
Sentence B: penguins are flightless .
Label: NotNextSentence
```

USE Algorithm:

The Universal Sentence Encoder encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. The pre-trained Universal

Sentence Encoder is publicly available inTensorFlow-hub. It comes with two variations i.e., one trained with Transformer encoder and other trained with Deep Averaging Network (DAN). The two have a trade-off of accuracy and computational resource requirement. While the one with a Transformer encoder has higher accuracy, it is computationally more intensive.

5.2 Implementation

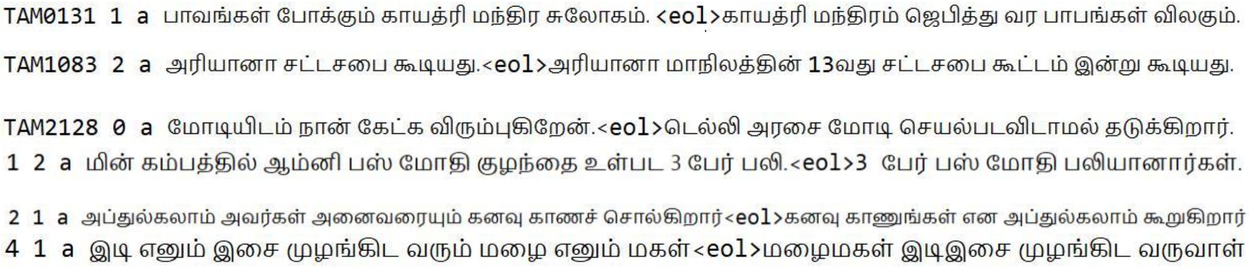
This section describes pre-processing the data as required by the algorithms and describes the train and test data formats required by each of the algorithms and their workflow.

Data pre-processing: A data set (or dataset) is a collection of data. Dataset plays a vital role in deep learning. In our project the system is trained and tested using a Dataset of four languages. Pre-processing is mainly done using python script. The train dataset is obtained in the form of .xml file and the test data is obtained in the form of .xlsx file. Python scripts are written to extract data from the xml file and convert it into tab separated variables format as required by the algorithms.

5.2.1 BERT

Test-Data

sentence1<eol>sentence2
sentence1<eol>sentence2



```
TAM0131 1 a பாவங்கள் போக்கும் காயத்ரி மந்திர சுலோகம். <eol>காயத்ரி மந்திரம் ஜெபித்து வர பாபங்கள் விலகும்.
TAM1083 2 a அரியானா சட்டசபை கூடியது.<eol>அரியானா மாநிலத்தின் 13வது சட்டசபை கூட்டம் இன்று கூடியது.

TAM2128 0 a மோடியிடம் நான் கேட்க விரும்புகிறேன்.<eol>டெல்லி அரசை மோடி செயல்படவிடாமல் தடுக்கிறார்.
 1 2 a மின் கம்பத்தில் ஆம்னி பஸ் மோதி குழந்தை உள்பட 3 பேர் பலி.<eol>3 பேர் பஸ் மோதி பலியானார்கள்.

 2 1 a அப்துல்கலாம் அவர்கள் அனைவரையும் கனவு காணச் சொல்கிறார்<eol>கனவு காணுங்கள் என அப்துல்கலாம் கூறுகிறார்
4 1 a இடி எனும் இசை முழங்கிட வரும் மழை எனும் மகள்<eol>மழைமகள் இடிஇசை முழங்கிட வருவாள்
```

Workflow:

- Dataset is loaded. 80% labelled data is used for training and 20% is used for validation.
- The text is split into tokens and are associated with token ids.

  - [CLS] token is prepended for each sentence and [SEP] token is appended.
  - All sentences must be padded or truncated to a single, fixed length
  - Attentions masks are generated, it indicates if a token is a padding token or not.
  - BERT is used for sequence classification from the transformer's library. This is the normal BERT model with an added single linear layer on top for classification that will use as a sentence classifier. As input data is feed, an entire pre-trained BERT model and the additional untrained classification layer is trained on our specific

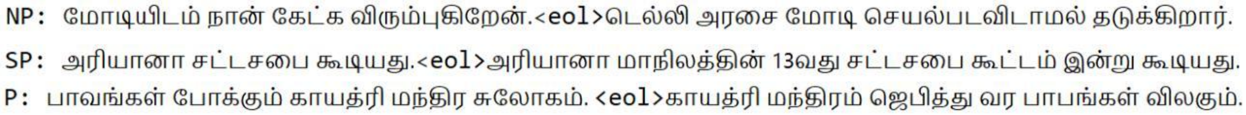
task. The pretrained model used is 'Bert-base-uncased' 12-layer, 768-hidden, 12-heads, 110M parameters, Trained on cased text in the top 104 languages with the largest Wikipedia.

- Training is parameters are defined,
  Batch size:32
  Learning rate:2e-5
  Number of epochs:4
- Training is done and a model is obtained.
- Test data is applied to the developed model, logits are returned.
- The logits are passed to SoftMax to obtain the probabilities.

5.2.2 USE

Dev-Data format

label: *sentence*1<*eol*>*sentence*2



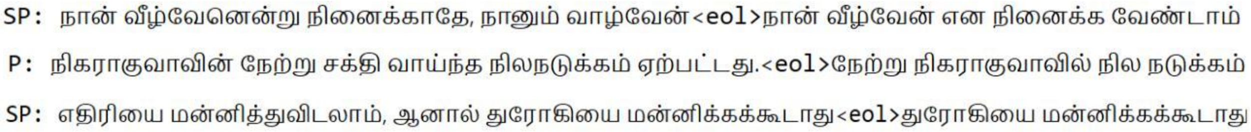
NP: மோடியிடம் நான் கேட்க விரும்புகிறேன்.<eol>டெல்லி அரசை மோடி செயல்படவிடாமல் தடுக்கிறார்.

SP: அரியானா சட்டசபை கூடியது.<eol>அரியானா மாநிலத்தின் 13வது சட்டசபை கூட்டம் இன்று கூடியது.

P: பாவங்கள் போக்கும் காயத்ரி மந்திர சுலோகம். <eol>காயத்ரி மந்திரம் ஜெபித்து வர பாபங்கள் விலகும்.

Test-Data format

label:*sentence*1<*eol*>*sentence*2



SP: நான் வீழ்வேனென்று நினைக்காதே, நானும் வாழ்வேன்<eol>நான் வீழ்வேன் என நினைக்க வேண்டாம்

P: நிகராகுவாவின் நேற்று சக்தி வாய்ந்த நிலநடுக்கம் ஏற்பட்டது.<eol>நேற்று நிகராகுவாவில் நில நடுக்கம்

SP: எதிரியை மன்னித்துவிடலாம், ஆனால் துரோகியை மன்னிக்கக்கூடாது<eol>துரோகியை மன்னிக்கக்கூடாது

Workflow:

- The first step is to turn the raw text file into a pandas Data Frame and set the" label" column to be a categorical column so as it can further access a label as a numeric value.
- Next step will prepare the input/output data for the model by the process of dataset preprocessing using a python script, the input as a list of pairs of sentences of a particular language, and output as a list of predicted values (P, NP or SP).
- Once the Dataset is pre-processed, the model is ready to be built.
- In the next step, train the model with the training datasets named as" language" -train.txt and validate it with the development dataset named as" language" -dev.txt. Validate the performance at the end of each training epoch with test datasets named as□ language" -test.txt.
- The final validation result shows the highest accuracy gets around 50-60% after training for 10 epochs.
- After the model is trained and its weights saved to a file, it is really to make predictions on a new set of sentence pairs.
- The final result is obtained as P, NP or SP.
- The predicted labels are compared with the true labels for the purpose of calculating f1 score, accuracy, precision and recall scores.

VI. EXPERIMENTAL RESULTS

6.1 Prediction Results

The statistics of the training and testing data set for each language are described. For task1, the confusion matrix for all languages for each of the algorithms is constructed. The confusion matrix describes the number of correct predictions and false predictions. Here Tp signifies true positives - number of paraphrases correctly identified as paraphrases, Fp signifies the false positive - number of non paraphrases identified as paraphrases, Fn signifies false negatives - number of paraphrases identified as non paraphrases, Tn signifies true negative - number of paraphrases correctly identified as paraphrases.

6.1.1 Tamil Prediction Results

Task1

TABLE1: Statistics of Training and Testing Data:

Category	**Training Data**	**Testing Data**
Number of Paraphrases	1000	400
Number of Non-Paraphrases	1500	500
Total	2500	900

TABLE 2: Confusion Matrix

Category		**Predicted labels**			
		BERT		**USE**	
		P	NP	P	NP
True labels	P	tp= 99	fn=301	tp= 34	fn=366
	NP	fp= 73	tn=427	fp=48	tn=452

Using BERT:
Number of True predictions: tp+tn=526
Number of False predictions: fp+fn=374
Accuracy: (tp+tn)/(tp+tn+fp+fn)=58.44

Using USE:
Number of True predictions: tp+tn=486
Number of False predictions: fp+fn=414
Accuracy: (tp+tn)/(tp+tn+fp+fn) =54

TABLE 3:
Statistics of Training and Testing Data

Category	**Training Data**	**Testing Data**
Number of Paraphrases	1000	400
Number of Non-Paraphrases	1500	500
Number of Semi-Paraphrases	1000	500
Total	3500	1400

TABLE 4:
Proportion of correct Predictions

Category	BERT		USE	
	Number of Predictions	No of correct predictions	Number of Predictions	No of correct predictions
Paraphrases	176	106	955	210
Non-Paraphrases	977	427	251	36
Semi-paraphrases	247	96	194	62

Accuracy = Number of correct predictions/Total number of predictions *100
Using BERT:
Total number of correct predictions: 629
Accuracy = 44.92
Using USE:
Total number of correct predictions: 308
Accuracy = 2

VI. ENSEMBLING

The results of the two algorithms are combined to improve the accuracy compared to individual algorithms. The prediction is done as follows:

- The probability scores of each of the algorithm is compared.
  The highest probability score is taken and the prediction corresponding to that probability score if considered as the output.
- Consider an example for task1, for a sequence, BERT score of a prediction is 0.12 for P and 0.88 for NP, USE score is 0.11 for P and 0.89 for NP. Since USE score for NP is the highest compared to the other scores, the output class label is taken as Non paraphrase. Similarly, the output is computed in task2.

Comparison of algorithms:

From the above table, it can be seen that the performance of all the models for Tamil and Malayalam languagesare low compared to Hindi and Punjabi. This is due to the complex vocabulary of Tamil and Malayalam languages.
The performance of BERT is better compared to other algorithms due to its capability to understand the contextual meaning of the sentences.

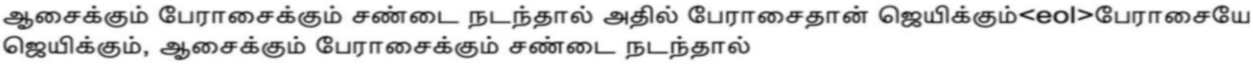
Consolidated Individual predictions of BERT and USE:

TABLE 6:

Language	**Task 1 Prediction Accuracy**		**Task 2 Prediction Accuracy**	
	BERT	**USE**	**BERT**	**USE**
Tamil	58.44%	54.00%	44.57%	22.00%
Malayalam	65.55%	56.11%	56.71%	35.78%
Hindi	85.22%	59.22%	54.14%	34.42%
Punjabi	85.80%	58.19%	63.06%	36.66%

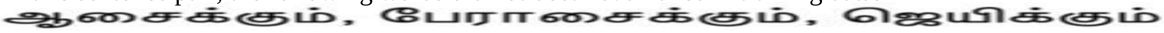
VII. ERROR ANALYSIS

The system encountered more new words that are not found in training data at least once and some words occurred only one or two times. The prediction accuracy is reduced due to the complex structure and vocabulary of Tamil and Malayalam languages. For example, consider the sentence pair



ஆசைக்கும் பேராசைக்கும் சண்டை நடந்தால் அதில் பேராசைதான் ஜெயிக்கும்<eol>பேராசையே ஜெயிக்கும், ஆசைக்கும் பேராசைக்கும் சண்டை நடந்தால்

In this sentence pair, the following words did not occur even once in training data:



ஆசைக்கும், பேராசைக்கும், ஜெயிக்கும்

These unknown words are assigned with UNK tokens. As a result, the system falsely predicted the paraphrase as non-paraphrase.

VIII. CONCLUSION AND FUTURE WORKS

A system that identifies paraphrases in a given sentence pair using two algorithms namely BERT and USE is developed. It consists of two tasks. In task1, the sentences are identified as paraphrases or not. In task2, the sentences are more precisely identified as paraphrase or semi-paraphrase or non-paraphrase. Data pre-processing is done as per the requirements of the different algorithms. The Vocabulary files are generated. Each algorithm outputs the probability scores using which the predictions are done. Ensemble is done based on the confidence score values. The system achieved higher accuracies of 85.22% and 85.80% in task1 for Hindi and Punjabi languages respectively. The system's performance is equivalent to that of the BERT performance as confidence values of BERT are higher than provided by the other algorithms. The increasing dataset size reduces the prediction accuracy. Future works may include character embedding of the sentences for USE and Seq2Seq models and repeating the algorithms and it can extend for cross-lingual paraphrases detection for more Indian languages.

REFERENCES

[1] Anand Kumar, M., Singh, S., Kavirajan, B., Soman, K.P. (2016), DPIL@FIRE 2016: Overview of shared task on detecting paraphrases in Indian Languages (DPIL)' , CEUR Workshop Proceedings, 1737, pp. 233-238.
[2]D.Thenmozhi and C. Aravindan (2016), ' Paraphrase Identification by Using Clause-based Similarity Features and Machine Translation Metrics' , In the Computer Journal, vol. 59, no. 9, pp. 1289–1302.

[3]DanielCer, Yinfei Yang, Sheng-yi Kong, Nan Hua,NicoleLimtiaco, Rhomni St John, Noah Constant,Mario Guajardo-Cespedes, Steve Yuan, Chris Tar,et al. (2018), ’ Universal sentence encoder’ , ar Xivpreprint arXiv:1803.11175.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2018) ,’ Bert: Pre-training of deep bidirectional transformers for language understanding’ , arXiv preprint arXiv:1810.04805.Kamal Sarkar (2016), ’ Detecting Paraphrases in Indian Languages Using Multinomial Logistic Regression Model’ ,In Shared-task on detecting paraphrases in Indian languages(DPIL), Forum for Information Retrieval.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details