Efficient use of machine learning models to evaluate the parametric performance of the DL models for language translation from Telugu to Hindi.

Shail garg
Department of Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham,
Bengaluru, India.
bl.en.u4cse22254@bl.students.amrita.edu

Yerukola Gayatri
Department of Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham,
Bengaluru, India.
bl.en.u4cse22267@bl.students.amrita.edu

A. Surya Kausthub
Department of Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham,
Bengaluru, India.
bl.en.u4cse22287@bl.students.amrita.edu

***Abstract*—Due to the language differences involved, there are lots of issues while translating Telugu which is a low resource language to Hindi, which is one of the most widely used languages in India. This research presents the different models of advanced machine translation such as OpenNMT, Fairseq machine translation, Helsinki-NLP/OPUS-MT, and BART or Bidirectional and Auto-Regressive transformers to solve these challenges. It is for this reason that the study expects to engage in systematic experimentation of these models under various parameter conditions with the overall aim of ascertaining their parametric ability to produce contextually correct translations. The primary aim here is to determine which models and approaches lead to improvement of the quality of translations, thus helping improve on the existing solutions for underrepresented languages such as Telugu. Dubbed as The Roadmap for enhancing accessibility and communicational opportunities in low-resource languages with MT, the findings of this research will positively impact the particular field.**

***Index Terms*—Telugu to Hindi Translation,Low-Resource Language,OpenNMT,Fairseq,Helsinki-NLP/OPUS-MT,BART**

I. Introduction

Language translation is fundamental in NLP since it helps in language barriers hence assists in enhancing global relations in terms of culture, economy, language among other relations. The process of translating from Telugu to Hindi, or for any other pair of languages that are phonetically, syntactically and, to an extent semantically dissimilar entails much more than mapping twenty-six letters onto homophones, fingers onto typewriter keys, or concepts onto a correlate array of symbols. Though both the Telugu, a Dravidian language of Andhra Pradesh in India, and Hindi – one of the most extensive languages in India – share no similitude in their script, grammar, syntax, semiotic systems, translating from one to the other is a challenge.

The first difficulty which appears when translating from Telugu to Hindi is the absence of large parallel corpora required to train deep learning models. Telugu, which is a low resource language having limited availability of digital text data is a major hurdle as most of today's MT systems require large amount of parallel data for achieving high level results. Even fewer involve real subject matter, let alone academic, research, legal or medical subject matter, making the translation task even more challenging especially when facing low resources settings.

It is interesting to note that in recent years deep learning has imposed more flexible approaches to address the imbalance between the languages. Players as OpenNMT, Fairseq, Helsinki-NLP/OPUS-MT, and BART (Bidirectional and Auto-Regressive Transformers) have emerged in this area, with each of them carrying its peculiarities into this area. Fairseq among them is famous for its solid sequence-to-sequence structures and flexibility of the framework, which makes both models fit for fine-tuning on particular translation tasks. Helsinki-NLP/OPUS-MT, for instance, uses a range of multilingual models which means that low resource languages, such as Telugu to Hindi translations can also be made. Since BART is both bidirectional and autoregressive it presents strong capabilities for translations, especially for more complicated language pairs due to its fluency and contextual relevance.

However, like any other research, each model has its pros and cons when it comes to low-resource languages especially the Telugu language. It is therefore important to know how these models behave depending on different conditions of use, the different parameters and adjustments that one can make in an aim to produce the best translations. This study is carried on by analyzing the architectural details, tuning techniques, and computational complexity integral to these models to improve the Telugu-to-Hindi translation. The knowledge obtained will be helpful to create more effective machine translation approaches for low-resource languages, hence providing paths toward enhanced interaction in different settings.

II. Literature survey

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, Fu Lee Wang [1] Did a Critical Review and Assessment While preserving performance, parameter-efficient fine-tuning (PEFT) techniques use less memory and fewer fine-tuning parameters. They are utilized in cross-lingual transfer, backdoor attack protection, and multi-task learning. For best results, they incorporate new parameters or combine different PEFT methodologies. Future studies will examine PEFT approaches in multi-modal learning and computer vision, as well as improve their efficiency and explainability.

Yasir Abdelgadir Mohamed; Akbar Khanan; Mohamed Bashir; Abdul Hakim H. M. Mohamed; Mousab A. E. Adiel [2] Worked on machine translation developments, with a special emphasis on neural machine translation (NMT) systems. It emphasizes how deep learning and artificial neural networks can be used to increase the accuracy, efficiency, and quality of translation. The study highlights the need for additional research to improve translation quality and address cultural nuances, as well as the significance of assessing machine translation systems using both automated metrics and human evaluations.

Manuel Eugenio Morocho Cayamcela,Wansu Lim [3] highlights advances in neural machine translation and deep learning, as well as ongoing conversations about inclusivity and cultural sensitivity, in order to examine how AI is changing language translation. Fuzzy algorithms tackle semantic ambiguity and linguistic variability, while feature extraction, intelligent recognition, and maximum-entropy principles improve language context awareness. AI is transforming translation. .

Kexin Zhang [4] Explains how back-translation and cross-lingual embeddings, along with creative strategies, enhance translation quality. It highlights the significance of resolving issues with NMT training techniques for more effective and precise translation systems by showing notable gains over conventional unsupervised systems. With assessments concentrating on evaluating translation quality through automated and human-based methodologies, the study also emphasizes machine translation as a useful tool in increasing efficiency and lessening the workload on translators.

Teddy Mantoro; Jelita Asian; Media A. Ayu [5] Applied sequence IRSTLM translation parameters and pruning to enhance a statistical machine translator's translation performance. It talks about how difficult it is to translate and offers a method that can still create accurate translations without requiring a deep understanding of the target language. The study highlights the significance of user interface, customisation, and pruning in machine translation while evaluating 28 distinct IRSTLM language modeling factors. The suggested strategy outperforms conventional approaches that rely on linguistic expertise, and the results reveal promising outcomes.

Kahler, B., Bacher, B. and Jones, K.C. [6] proposed a technique which addresses character conversion problems and enhances machine translation (MT) accuracy by employing ISO character mapping to improve reliability. It minimizes mistranslations and absurd results by drastically lowering errors when converting characters from languages like Arabic, Asian, and Cyrillic to Western scripts. Users consequently gain a deeper comprehension of foreign-language online information. The authors emphasize how easily open-source MT tools may be integrated with their suggested developer solution. This advancement in translating technology has wider ramifications for improving cross-cultural relationships and lowering obstacles to communication. The study recommends more investigation to hone these techniques and look at other ways to enhance MT systems.

Sun, S., Hou, H.X., Yang, Z.H. and Wang, Y.S[7] created an innovative method to improve translation for languages that are less well-known and have little data. To increase translation accuracy, especially when a substantial quantity of bilingual data is absent, it leverages the powerful pre-trained model CeMAT. How to avoid the model from repeatedly making the same mistakes is one of the primary problems that is mentioned. They tackle this by presenting an approach that lets the model grow from its errors. Additionally, they offer a clever training strategy that modifies based on the data and the confidence level of the model, particularly helpful for languages with limited resources. Their experiments show that these approaches translate substantially better, proving the effectiveness of pre-training in conjunction with this innovative learning strategy for low-resource languages.

III. Methodology

1. **Pseudo-Inverse Solution** Objective: To solve the linear classification problem using the pseudo-inverse method. The goal is to classify high-value transactions based on the features: Candies, Mangoes, Milk_Packets, and Payment.

Steps:

1. Data Preparation:

- Feature Matrix (X): Includes Candies, Mangoes, Milk_Packets, and Payment. A bias term (column of ones) is added to X.
- Target Vector (y): Binary values indicating whether a transaction is of high value.

2. Pseudo-Inverse Calculation:

- compute the pseudo-inverse of the feature matrix X.
- Calculate the weight vector W using the pseudo-inverse.
- Make predictions by applying a threshold to the weighted sum.

2. **Backpropagation for AND Gate** Objective: To train a neural network to simulate the AND gate logic using backpropagation. The neural network will learn the weights that map inputs to the correct outputs. Steps:

1. Data Preparation:

- Inputs: Truth table for AND gate.
- Outputs: Expected results of AND gate.

2. Neural Network Configuration:

- Architecture: 2-input neurons, 2 hidden neurons, 1 output neuron.
- Activation Function: Sigmoid function.
- Learning Rate: 0.05
- Epochs: 1000
- Convergence Criteria: Error threshold of 0.002.

3. Training:

- Perform feedforward operations to calculate predictions.
- Apply backpropagation to update weights.

3. **Backpropagation for XOR Gate** Objective: To train a neural network to simulate the XOR gate logic, which requires non-linear separation. Steps:

1. Data Preparation:

- Inputs: Truth table for XOR gate.
- Outputs: Expected results of XOR gate.

2. Neural Network Configuration:

- Architecture: 2-input neurons, 2 hidden neurons, 1 output neuron.
- Activation Function: Sigmoid function.
- Learning Rate: 0.05
- Epochs: 1000
- Convergence Criteria: Error threshold of 0.002.

3. Training:

- Perform feedforward operations and backpropagation.

4. **XOR Gate with Two Output Nodes** Objective: To extend the XOR problem to output two binary values, emulating a more complex classification problem. Steps:

1. Data Preparation:

- Inputs: Truth table for XOR gate.
- Outputs: Extended to two binary output values.

2. Neural Network Configuration:

- Architecture: 2-input neurons, 2 hidden neurons, 2 output neurons.
- Activation Function: Sigmoid function.
- Learning Rate: 0.05
- Epochs: 1000
- Convergence Criteria: Error threshold of 0.002.

3. Training:

- Perform feedforward and backpropagation processes.

5. **Text Classification Using MLPClassifier** Objective: To classify textual data using an MLPClassifier to understand its performance on real-world data. Steps: 1. Data Preparation:

- Dataset: Textual data with features and labels.
- Preprocessing: TF-IDF vectorization and label encoding.

2. Model Training:

- Architecture: MLPClassifier with 10 hidden neurons.
- Activation Function: ReLU.
- Epochs: 1000.

3. Evaluation:

- Calculate accuracy and compare predictions with actual values.

IV. Results

1. Pseudo-Inverse Solution

Table 1: Predicted vs Actual Results Using Pseudo-Inverse

TABLE I

SHOWS THE COMPARISON BETWEEN ACTUAL VALUES AND PREDICTIONS MADE USING THE PSEUDO-INVERSE METHOD. THE PREDICTIONS PERFECTLY MATCH THE ACTUAL VALUES, INDICATING ACCURATE CLASSIFICATION

Actual	Predicted Pseudo Inverse
1	1
1	1
1	1
0	0
1	1
0	0
1	1
1	1
0	0
0	0

2. Backpropagation for AND Gate

Table 2: Final Weights and Error for AND Gate

TABLE II

SUMMARIZES THE FINAL WEIGHTS OBTAINED AFTER TRAINING THE NEURAL NETWORK FOR THE AND GATE. THE FINAL ERROR IS VERY LOW, INDICATING SUCCESSFUL TRAINING.

Epoch	Weights (Input to Hidden)	Weights (Hidden to Output)
-	[-0.2172, 0.6485], [0.1030, 0.3320]	[-1.0818], [-0.6480]

3. Backpropagation for XOR Gate Table 3: Final Weights and Error for XOR Gate

TABLE III

PRESENTS THE FINAL WEIGHTS AND ERROR AFTER TRAINING THE NEURAL NETWORK ON THE XOR GATE. THE LOW ERROR INDICATES EFFECTIVE LEARNING OF THE XOR FUNCTION.

Epoch	Weights(Input to Hidden)	Weights(Hidden to Output)
-	[[0.37, 0.96], [0.72, 0.62]]	[[-0.05], [0.08]]

4. XOR Gate with Two Output Nodes

Table 4: Final Weights and Error for XOR Gate (Two Output Nodes) 5. Text Classification Using MLPClassifier

TABLE IV

DISPLAYS THE FINAL WEIGHTS AND ERROR AFTER TRAINING THE XOR GATE MODEL WITH TWO OUTPUT NODES. THE MODEL ACHIEVED A SATISFACTORY ERROR RATE.

Epoch	Weights(Input to Hidden)	Weights(Hidden to Output)
-	[[0.276, 1.099], [0.67, 0.844]]	[[0.153, -0.449], [-0.177, 0.434]]

Table 5: Performance of MLPClassifier on Text Classification

TABLE V

SUMMARIZES THE PERFORMANCE METRICS OF THE MLPCLASSIFIER ON THE TEXT CLASSIFICATION TASK. THE ACCURACY IS NOTABLY LOW, SUGGESTING ROOM FOR IMPROVEMENT.

Metric	Value
Accuracy	0.0125
Sample Predictions	['Hyderabad...', '...', ...]
Actual	['Hyderabad...', '...', ...]

V. DISCUSSION

Pseudo-Inverse Solution:
The perfect classification for high-value transactions highlights the effectiveness of the pseudo-inverse method for linearly separable problems. The absence of misclassifications reinforces its utility for problems with well-defined linear decision boundaries. This supports the idea that pseudo-inverse methods, though simple, can be robust for tasks requiring precise and direct separation between classes.

Backpropagation for AND Gate The network's ability to learn the AND gate function with minimal error shows the power of neural networks, even with basic architectures. The low final error suggests that the network converged efficiently, reflecting the straightforward nature of the AND gate as a linearly separable problem. This experiment also underscores how even simple networks can handle basic logical operations.

Backpropagation for XOR Gate The success in learning the XOR gate function, a well-known non-linear problem, is a testament to the neural network's ability to model more complex patterns. The low error rate indicates that the network captured the inherent non-linearity of the XOR logic. This result reinforces the neural network's strength in handling tasks that cannot be solved by linear classifiers.

XOR Gate with Two Output Nodes Handling the XOR gate with multiple outputs further illustrates the flexibility of neural networks. By achieving a satisfactory error rate, the model demonstrates its ability to manage tasks with multiple output dimensions, highlighting the adaptability of neural networks when addressing more complex classification problems.

Text Classification Using MLPClassifier The poor performance in the text classification task points to the challenges of applying neural networks without sufficient preprocessing or appropriate architectural complexity. With an accuracy of only 0.0125, this suggests that either the model was underfitting or the input data was not prepared adequately. This experiment highlights the need for thorough data preprocessing (e.g., tokenization, stop-word removal, and vectorization) and perhaps deeper architectures. Further exploration of different feature extraction methods and hyperparameter tuning is necessary to enhance the model's performance.

VI. CONCLUSION

In conclusion, the experiments demonstrate the effectiveness of neural networks across a range of classification tasks. The pseudo-inverse solution performed excellently in linear classification problems, and backpropagation successfully learned both simple (AND) and non-linear (XOR) logical functions. The results show that neural networks can adapt to a variety of problems, including handling multiple outputs.

However, the underperformance in text classification highlights the importance of proper preprocessing, architecture selection, and model tuning for more complex, real-world data. Future work should focus on refining the approach to text classification by exploring more sophisticated architectures like recurrent neural networks (RNNs) or transformer-based models, along with enhanced feature engineering and extensive hyperparameter optimization. This will be essential to fully realize the potential of neural networks in diverse applications.

REFERENCES

[1] Xu, L., Xie, H., Qin, S.Z.J., Tao, X. and Wang, F.L., 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. arXiv preprint arXiv:2312.12148..

[2] Mohamed, Y.A., Khanan, A., Bashir, M., Mohamed, A.H.H., Adiel, M.A. and Elsadig, M.A., 2024. The impact of artificial intelligence on language translation: a review. Ieee Access, 12, pp.25553-25579.

[3] Cayamcela, M.E.M. and Lim, W., 2019, February. Fine-tuning a pre-trained convolutional neural network model to translate American sign language in real-time. In 2019 International Conference on Computing, Networking and Communications (ICNC) (pp. 100-104). IEEE.

[4] Zhang, K., 2021, May. Application of Pretrained Models for Machine Translation. In 2021 International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 849-853). IEEE.

[5] Mantoro, T., Asian, J. and Ayu, M.A., 2016, October. Improving the performance of translation process in a statistical machine translator using sequence IRSTLM translation parameters and pruning. In 2016 International Conference on Informatics and Computing (ICIC) (pp. 314-318). IEEE.

[6] Kahler, B., Bacher, B. and Jones, K.C., 2012, July. Language translation of web-based content. In 2012 IEEE National Aerospace and Electronics Conference (NAECON) (pp. 40-45). IEEE.

[7] Sun, S., Hou, H.X., Yang, Z.H. and Wang, Y.S., 2023, June. Multilingual Pre-training Model-Assisted Contrastive Learning Neural Machine Translation. In 2023 International Joint Conference on Neural Networks (IJCNN) (pp. 01-07). IEEE.