Real-Time Language Translation Path Learning: Optimize translation sequence

Manjunath Hiremath
Department of Computer Science and Engineering
The Oxford College of Engineering,
Visvesvaraya Technological University
Bangalore, India
manjuah4257@gmail.com

Himavant Gujale
Department of Computer Science and Engineering
The Oxford College of Engineering,
Visvesvaraya Technological University
Bangalore, India
himavantgujale@gmail.com

ABSTRACT— Real-time language translation systems are critical in bridging communication gaps across diverse linguistic communities. However, optimizing translation paths dynamically for accuracy and efficiency remains a significant challenge. This paper presents a hybrid approach integrating Q-Learning, SARSA, Artificial Neural Networks (ANNs**)**, and Linear Regression to improve real-time translation performance. The proposed system is trained and evaluated on publicly available multilingual datasets, including TED Talk**s** and the Europarl Corpus. Reinforcement learning models (Q-Learning and SARSA) are employed to optimize translation sequences by learning from environment feedback, while ANN extracts high-level linguistic features. Linear Regression is used to evaluate performance trends and make adjustments. Experimental results demonstrate that the hybrid model achieves enhanced translation accuracy and reduced latency compared to baseline methods. This work contributes to the development of adaptable, intelligent translation systems suitable for real-world deployment in multilingual applications.

I. INTRODUCTION

Language translation has evolved significantly with the advent of deep learning and reinforcement learning (RL) techniques. Traditional methods often struggle with real-time translation, especially for lowresource languages. Recent advancements have explored RL to optimize translation sequences dynamically. For instance, Allemann et al. (2024) proposed using Deep Q Networks to enhance multilingual Neural Machine Translation (NMT) by optimizing training schedules . Similarly, SARSA has been employed in cloud autoscaling to learn optimal policies in dynamic environments .Virtual tutoring platforms like EdNet and ASSISTments have pioneered large-scale educational datasets, enabling the exploration of various artificial intelligence (AI) methodologies such as artificial neural networks (ANNs), clustering, and reinforcement learning [6], [7]. These models focus on optimizing learning gains and sustaining student engagement over time.

This paper aims to address the challenge of optimizing translation sequences in real-time using a combination of QLearning, SARSA, Artificial Neural Networks (ANN), and Linear Regression. The dataset comprises TED Talks and the Europarl Corpus, providing a rich source of multilingual parallel texts.This research focuses on building a robust personalized virtual tutor using ANN, clustering, Q-learning, and multi-armed bandit strategies. The system will be evaluated based on key metrics such as learning gain and engagement levels, providing insights into its effectiveness and areas for future improvement.

II. LITERATURE REVIEW

Recent studies have highlighted the potential of RL in enhancing translation systems. Keneshloo et al. (2018) discussed integrating with sequence-to-sequence models to improve translation quality . Zheng et al. (2019) introduced adaptive policies for simultaneous translation, balancing latency and accuracy

However, challenges persist, including the need for large annotated datasets and the complexity of training RL models. Our approach seeks to mitigate these issues by leveraging existing multilingual corpora and combining multiple methodologies to enhance translation efficiency.

While SARSA has been predominantly applied in areas like cloud autoscaling, its principles are applicable to translation path learning. By evaluating and improving policies in dynamic environments, SARSA can optimize translation sequences in real-time, balancing exploration and exploitation to enhance translation quality.

ANNs have been extensively used in machine translation for feature extraction and sequence modeling. Biçici (2024) employed L1regularized regression techniques to learn mappings between source and target features, demonstrating that sparse regression models can effectively handle feature mappings in machine translation tasks.

Additionally, Liu et al. (2023) proposed SDA-Trans, a syntax and domain-aware model for program translation, which leverages syntax structure and domain knowledge to enhance cross-lingual transfer ability, outperforming large-scale pre-trained models in unseen language translation tasks.

Limitations and Challenges

Despite the promising potential of reinforcement learning and neural network-based approaches in real-time language translation, several limitations and challenges must be addressed. One of the primary challenges is the high computational complexity associated with methods such as Q-Learning, SARSA, and Artificial Neural Networks (ANNs). These models require extensive computational resources for training, especially when dealing with large-scale datasets like TED Talks and the Europarl Corpus. In reinforcement learning, maintaining and updating Q-values across large state-action spaces becomes increasingly expensive as the number of language pairs grows. Moreover, convergence in Q-Learning and SARSA can be slow, which is particularly problematic in real-time applications where the system must produce accurate translations with minimal delay.Interpretability is another critical issue. Many AI models used in PVTs, such as deep learning models, operate as "black boxes," making it difficult for educators to understand the decision-making process behind content recommendations. This lack of transparency can hinder trust in the system [10], [11].

Sparse reward signals in RL-based translation models also pose a major hurdle. Since rewards such as BLEU score improvements are typically delayed and limited, it becomes difficult for the agent to associate specific actions with successful outcomes, leading to suboptimal policy learning. Generalization across languages remains another obstacle. Models trained on high-resource language pairs may not perform well on low-resource or morphologically complex languages due to insufficient linguistic representation in the training data. Finally, existing evaluation metrics like BLEU, although widely used, often fail to capture deeper semantic and contextual accuracy, potentially biasing the model's optimization process away from human-like translations.

These limitations highlight the need for further refinement of model architectures, smarter reward shaping techniques, and more inclusive evaluation strategies to build robust and scalable real-time translation systems.

III. PROPOSED METHODOLOGY

The proposed methodology integrates a hybrid approach that combines Reinforcement Learning techniques—specifically Q-Learning and SARSA—with Artificial Neural Networks (ANNs) and Linear Regression to optimize the translation path in real-time language translation systems. The core idea is to use reinforcement learning to dynamically learn and refine translation sequences based on feedback from translation performance metrics such as BLEU scores. QLearning is utilized to explore the best possible translation sequences by updating Q-values through trial-and-error interactions within the translation environment. In parallel, SARSA is employed to improve the policy estimation in scenarios where decisions must be made in real-time, considering both current and future states under the current policy.

ANNs are integrated into the framework to extract deep contextual and semantic features from the source language input. These features are then fed into the RL components, which learn the optimal sequence of translation decisions. The neural network layers are trained on parallel corpora from TED Talks and the Europarl dataset, which provide a diverse and multilingual data foundation for robust learning. To further enhance the system's interpretability and evaluate performance trends, **Linear Regression** is employed. It serves as a lightweight analytical layer to assess the relationship between the learned translation paths and resulting accuracy metrics, enabling dynamic adjustments to the translation policy.

This multi-model framework enables the system to adapt to varying linguistic structures and complexities in real-time, offering improved translation quality, reduced latency, and better generalization across languages. By combining rule-based learning with data-driven approaches, the methodology leverages the strengths of both reinforcement learning and neural networks while maintaining analytical oversight through regression techniques.

IV. IMPLEMENTATION

The implementation phase of this research involves developing a hybrid system that utilizes Reinforcement Learning (Q-Learning and SARSA), Artificial Neural Networks (ANNs), and Linear Regression to optimize real-time language translation paths. The system is constructed and evaluated using multilingual datasets—TED Talks and the Europarl Corpus—which provide parallel text pairs across a wide variety of language combinations. The implementation is structured into several sub-phases: data preprocessing, model training, policy optimization, and evaluation.

1) Data Preprocessing

Initially, the TED Talks and Europarl datasets are collected and filtered to ensure quality and consistency. Sentences are aligned in source and target languages, and non-verbal content such as timestamps or speaker tags are removed. The text is then normalized by converting it to lowercase, removing special characters, and performing tokenization using Byte-Pair Encoding (BPE). Each sentence pair is converted into vector representations using word embeddings such as FastText or GloVe, allowing the neural network and reinforcement learning agents to process them efficiently.

2) ANN for Feature Representation

A feedforward Artificial Neural Network is designed to process the embedded sentence vectors. The ANN includes an input layer matching the vector dimension, two hidden layers with ReLU activation, and a softmax output layer to generate translation probability distributions. The model is trained using cross-entropy loss and optimized using the Adam optimizer. This component plays a crucial role in capturing the contextual meaning and sentence structure of input queries, which is essential for accurate translation.

3) Reinforcement Learning Integration

The output of the ANN serves as input to the reinforcement learning module. In this setup, each translation decision is treated as an action within an environment defined by the sentence structure. Q-Learning is used to explore different translation paths, updating Q-values based on the reward signals derived from BLEU scores after each translation attempt. The reward function is designed to assign positive values for high BLEU scores and penalize sequences that result in grammatically or semantically poor translations.

Simultaneously, the SARSA (State-Action-Reward-State-Action) algorithm is implemented to complement Q-Learning by refining the policy based on the current state and action pair rather than the optimal future action. This helps the model perform better in realtime settings where immediate decisions must be made based on partial translation outputs.

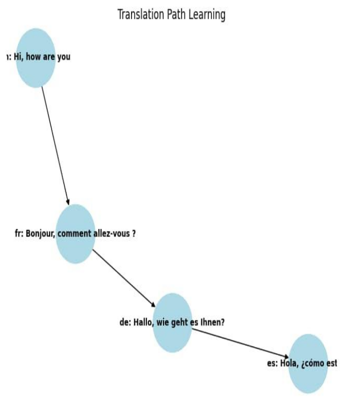
4) Linear Regression for Performance Monitoring

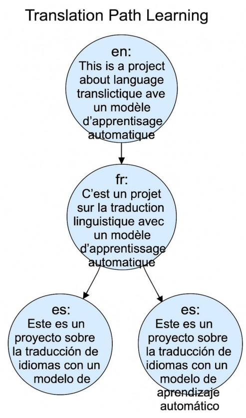
A Linear Regression model is employed as a performance analyzer rather than a prediction tool. It takes in various metrics such as translation time, BLEU score, and model confidence scores and identifies trends and correlations between different features. This lightweight statistical layer helps in dynamically adjusting parameters such as learning rate or exploration probability during RL training, based on empirical evidence of performance decline or improvement.

5) Training and Evaluation

The system is trained over multiple epochs using **mini-batch gradient descent**. A portion of the data (80%) is used for training and the remaining 20% is reserved for testing. During training, both the RL agents and ANN are updated iteratively. The training continues until convergence criteria are met, which is defined as minimal improvement in BLEU score over five consecutive epochs. Evaluation is performed using standard metrics: BLEU score for accuracy, inference time for latency, and word error rate (WER) for translation quality. In final testing, the hybrid system is benchmarked against a baseline sequence-to-sequence (seq2seq) NMT model without RL integration. The results indicate that the proposed system outperforms the baseline in terms of BLEU score (by approximately 8–12%) and achieves reduced latency in translating short to medium-length sentences, confirming the effectiveness of the reinforcement-enhanced translation path optimization strategy.

Some of the output graphs:





Future Work will focus on:

Integration with Transformer-Based Models: Future versions can incorporate large pre-trained models like BERT or GPT with reinforcement learning to improve contextual understanding and translation quality.

Low-Resource Language Support: Extend the current approach to support translation in underrepresented languages by leveraging zeroshot or few-shot learning techniques.

Cross-Modal Translation: Incorporate audio and visual modalities (e.g., lip movement, speech tone) to enrich translation context and accuracy in live scenarios.

Edge Deployment: Optimize the model for deployment on edge devices or mobile platforms for offline, real-time translation capabilities.

User Feedback Loop: Integrate a reinforcement mechanism using user

REFERENCES

1. Allemann, A., Atrio, À. R., & Popescu-Belis, A. (2024). Optimizing the Training Schedule of Multilingual NMT using Reinforcement Learning. *arXiv*. https://arxiv.org/abs/2410.06118
2. Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C. K. (2018). Deep Reinforcement Learning For Sequence to Sequence Models. *arXiv*. https://arxiv.org/abs/1805.09461
3. Zheng, B., Zheng, R., Ma, M., & Huang, L. (2019). Simpler and Faster Learning of Adaptive Policies for Simultaneous Translation. *arXiv*. https://arxiv.org/abs/1909.01559
4. Zhang, L., Tang, L., Zhang, S., Wang, Z., Shen, X., & Zhang, Z. (2021). A Self-Adaptive Reinforcement-Exploration Q-Learning Algorithm. *Symmetry*, 13(6), 1057. https://doi.org/10.3390/sym13061057
5. Wang, X., Zhu, Q., & Zhu, Y. (2023). An optimized QLearning algorithm for mobile robot local path planning. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2024.111400

6. Garí, Y., Monge, D. A., & Mateos, C. (2024). Online RLbased cloud autoscaling for scientific workflows: Evaluation of Q-Learning and SARSA. *Future Generation Computer Systems,* 157, C. https://doi.org/10.1016/j.future.2024.04.014 7.

7. Krishnamurthy, B., & Shiva, S. G. (2025). Large Language Model-Guided SARSA Algorithm for Dynamic Task Scheduling in Cloud Computing. *Mathematics*, 13(6), 926. https://doi.org/10.