JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Israr, H., Anwar, S., Zolkipli, M. F., & Ahmed, A. (2026). Non-autoregressive and iterative refinement non-autoregressive transformer with additional graph layer for Urdu-to-English machine translation. *Journal of Information and Communication Technology*, *25*(4), 120-144. https://doi.org/10.32890/jict2026.25.4.6

Non-Autoregressive and Iterative Refinement Non-Autoregressive Transformer with Additional Graph Layer for Urdu-to-English Machine Translation

[1]Huma Israr, [2]Shahid Anwar, [3]Mohamad Fadli Zolkipli & [4]Arslan Ahmed
[1&2]Department of Computing and Emerging Technology,
National Skills University Islamabad, Pakistan
[3]School of Computing, Universiti Utara Malaysia, Malaysia
[4]Sheffield Hallam University, United Kingdom

[*1]huma.israr@nsu.edu.pk
[2]shahidanwar@nsu.edu.pk
[3]m.fadli.zolkipli@uum.edu.my
[4]arslan.ahmed@shu.ac.uk

Received: 26/4/2026 Revised: 7/10/2026 Accepted: 17/10/2026 Published: 31/10/2026

ABSTRACT

Neural Machine Translation (NMT) has achieved state-of-the-art performance using autoregressive Transformer. However, the trade-off between translation quality and decoding speed remains a central challenge. This problem is further intensified for low-resource languages such as Urdu, where limited parallel data and complex linguistic structure result in poor translation quality and reduced model effectiveness. To overcome this problem Non-Autoregressive Transformer (NAT) and Non-Autoregressive Transformer with Iterative Refinement (iNAT) are proposed. This paper investigates NAT and iNAT for low resource language Urdu-English machine translation. We analyze translation accuracy and their effectiveness for low-resource language pairs. Experimental results show that both NAT and iNAT outperform the baseline Autoregressive Transformer (ART) in terms of translation accuracy. However, the overall translation quality remains limited due to the complex linguistic structure and resource scarcity of the Urdu-English pair. To address this limitation, we further refine these architectures by introducing an additional graph-based learning layer that captures latent relationships among tokens through attention-driven graph representations. The proposed graph-enhanced model learns contextual dependencies by dynamically constructing graphs over attention representations. Experimental evaluation demonstrates that the proposed architecture improves the translation performance of both NAT and iNAT models. On the Urdu-English translation task, the

proposed model achieves a 3.0% higher BLEU score than the baseline models. The findings highlight the significance of incorporating graph-based layer to enhance translation quality, making NMT systems more effective for low-resource and linguistically complex languages such as Urdu.

Keywords: Low-resource language, machine translation, non-autoregressive, transformer, Urdu.

INTRODUCTION

Neural Machine Translation (NMT) has transformed language translation using encoder-decoder architectures and attention mechanisms. Early NMT systems relied on recurrent and convolutional neural networks, including LSTM, GRU, and CNN-based models. With these architectures, NMT evolved rapidly over the past decade (El Maazouzi, 2017). In recent years, however, the field has shifted toward fully attention-based models, particularly the Transformer architecture introduced by Vaswani et al. (2017). The Transformer has become state-of-the-art approach for NMT across a wide range of translation tasks (Vaswani et al., 2017). The standard Transformer uses multi-head self-attention and position-wise feed-forward layers. The self-attention mechanism enables the model to capture long-range dependencies in a sequence. In addition, the Transformer performs computations in parallel, which significantly improves efficiency. As a result, it has achieved substantial performance gains across a wide range of translation tasks (Banerjee & Banik, 2025; Lakew et al., 2018; Zaki, 2024). These improvements have led Transformer-based models to become the dominant architecture in both academic research and industrial translation systems (Zaki, 2023).

The vanilla Transformer adopts an autoregressive strategy to translate. The Autoregressive models generate target tokens sequentially. To produce a current token the model uses the prediction from the previous time step. Each token depends on the source input and the token generated before it. As a result, the translation is generated sequentially in a left-to-right (LTR), word-by-word manner (Li et al., 2023). Because each token must wait for the prediction of the previous toke, this left-to-right decoding leads to high inference latency. The decoding speed and sequence generation latency remain significant bottlenecks for real-time applications.

In order to handle this speed and translation latency issue Non-Autoregressive Transformer (NAT) are introduced. The NAT models employ a more efficient parallel decoding then autoregressive/vanilla transformers (Gu et al., 2017). NAT models employ single pass decoding that attempt to generate all tokens in parallel. This significantly improves the translation speed. However, it often sacrifices the translation quality. The NAT-based on the Transformer generate each token independently and simultaneously. The NAT model assumes that the prediction of each target token is conditionally independent. It is based only on the source input. As a result, it does not consider previously generated words and generates all target tokens in parallel at once (Li et al., 2023).

One of the major limitations of NAT MT model is their low translation performance. This problem is mainly caused by incorrect prediction about target sentence length. In NAT, the model typically predicts the target sentence length before generating target tokens. If this predicted length is shorter than the true length, the output senetnce may miss some important words (Lim et al., 2022). To enhance the NAT model various improvement have been proposed at either encoder or decoder module (Ghazvininejad et al., 2019; Liu et al., 2021; Stern et al., 2019). However, their performance still remains far from optimal. Additionally, NAT models may also produce unclear phrases and repeated words in the translated output.. This is because in NAT models, each token in the target sentence is generated independently (Zhou et al., 2019). In order to overcome this problem a variation to NAT model called NAT with iterative refinement (iNAT) is proposed.

iNAT improve translation via multiple passes over predicted output (Xiao et al., 2025). iNAT models uses multiple decoding steps to generate the final results. The model keeps the non-autoregressive decoding approach in each step. The results from earlier steps provide partial target-side information. These intermediate results are iteratively refined in subsequent steps, leading to improved translation quality. Through this iterative refinements, the performance of these models achieved significant improvements. In certain cases, they surpass their ART counterparts.

In a separate approach to iterative refinement, Norouzi et al. (2023) proposed Distill Multiple Steps (DiMS) model (Norouzi et al., 2023). The DiMS uses a simple and effective distillation technique to generate quality translation. It reduces the number of steps needed to achieve a certain translation quality in DU-EN MT task. As a result of the architectural modification, the proposed model achieved notable gains in both efficiency and translation accuracy (Norouzi et al., 2023).

Another refinement model is proposed by Jason Lee. Lee et al. (2018) proposed Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. The model use conditional denoising auto-encoder. That substantially improved the translation quality (Lee et al., 2018). Like DiMS, this approach demonstrates that iterative refinement can enhance both translation accuracy and efficiency, providing a more effective solution for non-autoregressive translation. These models have recently demonstrated impressive performance in various translation and generation tasks, surpassing the ART counterparts. However, further research is still needed to build more efficient and competitive iNAT models that can achieve consistent and robust performance across different tasks (Xiao et al., 2025).

Urdu is a low-resource South Asian language with significant practical importance. The language has over 231 million speakers (Basit et al., 2024). It is one of the most widely spoken languages in South Asia and among the top ten languages in the world. Despite widely spoken, most NMT research has primarily focused on European and other high-resource languages (Zamin, 2013). Urdu remains relatively under-explored in systematic and controlled NMT studies (Khan & Khan, 2025). This imbalance highlights a clear research gap, emphasizing the need for more focused investigation into the challenges and development of effective NMT approaches for Urdu (Bilal, 2016).

Furthermore, Urdu is a difficult language for NMT systems. It is morphologically rich language. The linguistic complexity makes it difficult to effectively align source and target languages, particularly in low-resource conditions (Israr et al., 2025; Safder et al., 2024). These challenges highlight the need for systematic evaluations that examine architectural variations and analyze their effectiveness under controlled experimental conditions. In this context, it is useful to explore different neural translation paradigms, such as NAT and Iterative NAT models for UR-EN MT. This exploration will provide deeper insights into their suitability and performance for Urdu-English MT.

To overcome the limitations of existing work, this study focuses on two key contributions. First, it evaluates the performance of both autoregressive and iNAT models in Urdu-English machine translation. Second, it develops a model designed to produce more accurate, fluent, and contextually relevant translations. To address the limitations of existing approaches, this study makes several contributions to Urdu–English NMT. First, it provides a systematic comparison of Non-Autoregressive Translation (NAT) and Iterative Non-Autoregressive Translation (iNAT) models for the Urdu–English translation task, with particular emphasis on translation quality and accuracy. Second, the study proposes an architectural enhancement by incorporating a graph layer into the NAT and iNAT frameworks to improve the modelling of relationships and contextual information within

the translation process. Finally, a controlled and reproducible experimental evaluation is conducted to investigate the effectiveness of the proposed graph-enhanced architectures and to quantify their impact on Urdu–English NMT performance. Collectively, these contributions aim to develop translation models that generate more accurate, fluent, and contextually relevant Urdu–English translations while providing empirical evidence on the effectiveness of graph-based architectural enhancements for non-autoregressive machine translation.

The remaining paper is structured as follows: Section 2 depicts the Related Work on AR Transformer, NAT, and iNAT-based NMT models. Section 3 outlines the methodology, implementation and the proposed graph-based layer. Section 4 presents the experimental setup, while Section 5 reports the results and compares them with existing models. Section 6 concludes the study and suggests directions for future research.

RELATED WORKS

This part of the paper consists of related work to ART, NAT and iNAT using different datasets and neural translational approaches. It is organised into three subsections: Autoregressive Models, Non-Autoregressive Transformers with iterative refinement, and Urdu-English NMT. The models and datasets used are summarized in Tables 1, 2, and 3.

Table 1

Summary of Transformer-Based Models

Reference	Model	Language pair
(Vaswani et al., 2017)	Transformer	DU-EN
(Wang et al., 2025)	Transformer BPE	DU-EN
(Israr et al., 2024a)	Transformer with sub wording	UR-EN
(Gangar et al., 2021)	Transformer with BPE (word / sentence level)	HI-EN
(Mahsuli et al., 2023)	Transformer with LenM model	AR-EN, DE-EN
(Busto-Castiñeira et al., 2024)	Transformer for NLG	ER-EN
(Shahriar & Barbosa, 2024)	Bi- direction Transformer	BN-EN
(Dixit, 2023)	Auto-regressive Transformer	HI, TA, MR, OD, PA, and TE to EN

Autoregressive Models

Autoregressive Transformer models have significantly reshaped the landscape of machine translation. Since their introduction, the versatility of transformers has encouraged researchers to adopt it for different MT tasks across multiple languages (Kamatala et al., 2025).

Transformer architecture for Dutch-English (DU-EN) translation task model was proposed by Vaswani et al. (2017). Their finding demonstrated strong performance compared to earlier NMT models. However, the model faced challenges in handling rare and out-of-vocabulary (OOV) words. Transformer relies on a fixed word-level vocabulary, rare and unseen words are often mapped to an unknown token and lost in the output. To address this limitation, later studies incorporated Byte Pair Encoding (BPE) with Transformer. BPE segmenting words into smaller subword units, which lets the

model compose rare words from known pieces and reduces vocabulary mismatch and improving translation quality for DU-EN (Wang et al., 2025).

While, Transformer-based approach that uses subword units for representing text in Urdu-English machine translation was developed by Israr et al. (2024a). Their study demonstrated that the approach is effective for language pairs with different writing directions, such as left-to-right (LTR) and right-to-left (RTL) scripts. By integrating Byte Pair Encoding (BPE) with the Transformer architecture, the model achieved improved handling of rare and OOV words. Their results showed that subword modeling significantly reduce the OOV problem, leading to better translation quality. Gangar et al. (2021), used a Transformer model to translate texts from Hindi-to-English in a low-resource setting. They used the Hindi-English parallel corpus provided by the Center for Indian Language Technology (CFILT), IIT Bombay. They experimented with both word-level and subword-level tokenization using Byte Pair Encoding (BPE), and thus achieved a state-of-the-art BLEU score on test set (Gangar et al., 2021).

The Autoregressive Transformers generate target tokens sequentially, which often leads to high decoding latency. Mahsuli et al. (2023), implemented a Transformer model with a novel multiplicative residual connection. They also proposed the use of a length model (LenM) during decoding. The experiments were conducted on two language pairs: (1) German-English and (2) Arabic-English. The results showed that the ART, when combined with the length model, performs better than the baseline system (Mahsuli et al., 2023).

Busto-Castiñeira et al. (2024) implemented a bidirectional Transformer for a Spanish-English natural language generation (NLG) (Busto-Castiñeira et al., 2024). Similarly, Shahriar et al., applied a Transformer-based approach for Bengali-English translation (Shahriar & Barbosa, 2024). Dixit (2023), applied Transformer models to multiple Indian languages, including Hindi, Tamil, Marathi, Odia, Punjabi, and Telugu, for translation into English. They explored transfer learning and its variations using benchmark datasets such as IMDB, SQuAD 2.0, and RACE. Their study concluded that Transformer-based approaches achieve state-of-the-art results for South Asian languages (Dixit, 2023).

Despite demonstrating strong performance in translating these languages, the sequential decoding process of autoregressive models still leads to slow inference speeds, limiting their applicability in real-time translation. This has motivated researchers to explore non-autoregressive translation (NAT) models as a means to achieve faster decoding.

Table 2

Summary of Transformer-Based NAT and iNAT Models

Reference	Model	Language pair
(Gu et al., 2017)	NAT	DU-EN
(Lee et al., 2018)	NAT	DU-EN
(Wang et al., 2019)	NAT with Auxiliary Regularization	EN-DU
(Gu & Kong, 2021)	NAT with KD, CTC and GLAT	EN-RO, EN-JA and EN-DU
(Kasai et al., 2020b)	NAT models with CMLM	EN-DU, EN-ZH, and EN-FR
(Inaguma et al., 2021)	NAT	ES-EN
(Kasai et al., 2020a)	iNAT	EN-DU
(Ghazvininejad et al., 2019)	iNAT	DU-EN

Non-Autoregressive Transformers and Iterative Refinement on NAT

The non-autoregressive (NAT) variant of the Transformer model employs an optimized model design. It features a modified Transformer architecture and uses an alternative decoding algorithm. NATs enable parallel decoding of all tokens, resulting in reduced time complexity. Early work on NATs includes Gu et al. (2017) and Lee et al. (2018), for EU-DU translation task. In recent years, many studies have aimed to achieve the translation quality of autoregressive models using non-autoregressive approaches (Gu et al., 2017; Lee et al., 2018).

Wang et al. (2019) proposed a Non-Autoregressive MT model with Auxiliary Regularization (AUX-REG). They evaluated the model on the IWSLT14 German-to-English (IWSLT14 DE-EN) and IWSLT16 English-to-German (IWSLT16 EN-DE) datasets. By improving the quality of decoder hidden representations through two auxiliary regularization techniques, the model effectively addresses repeated and incomplete translations, achieving significantly lower latency (Wang et al., 2019). Schmidt et al. (2022) reviewed several techniques proposed to improve non-autoregressive translation models and compared their combined impact on translation quality and speed, using the WMT'13, WMT'14, and WMT'16 English-German (EN-DE) datasets (Schmidt et. al., 2022). The study also provides more realistic expectations for speed.

Gu et al. (2017) applied a NAT model for translation on the WMT'14 EN-RO, EN-JA, and EN-DE datasets. He explored variations of the core model using CTC, GLAT, RND, and KD (knowledge distillation), and provided concise results comparing them with autoregressive (AT) and existing NAT approaches. NAT models with a sufficient number of iterations generally outperform the fully autoregressive baseline (Gu & Kong, 2021). Inaguma et al. (2021), implemented NAT for Spanish-English translation, applying the model to improve decoding speed while maintaining competitive translation quality (Inaguma et al., 2021).

Whereas, Kasai et al. (2020b) evaluated NAT models with CMLM on the WMT'14, WMT'16, and WMT'17 datasets, including EN-DE, EN-ZH, and EN-FR. The study investigated autoregressive models with encoders and decoders of varying depths. Sequence-level knowledge was applied to both NAT and ART models during training. Experimental results show that a deep encoder and a single-layer autoregressive decoder perform very well on MT tasks. It can outperform strong non-autoregressive models while keeping similar inference speed. This study proposes a new approach for future research. Furthermore, It focuses on achieving fast and accurate machine translation (Kasai et al., 2020b).

Despite demonstrating strong performance in translating these languages their translation quality remains lower since tokens are produced non-conditionally in parallel in single pass decoding. Iterative refinement provides an alternative by updating predictions over multiple passes, reducing errors and improving output quality. Hence maintaining balances between speed and quality.

Further extending the idea of iterative refinement in decoding by Kasai et al. (2020a) proposed a variation of the Transformer called the DisCo Transformer (Kasai, 2020a). DisCo is a non-autoregressive (NAT) Transformer with attention masking. The proposed model generates all tokens simultaneously, significantly reducing decoding time on average. Each generated token is conditioned on an arbitrary subset of the other tokens, which leads to performance gains during training. Ghazvininejad et al. (2019) proposed a variation to NAT with parallel decoding. The proposed model generates subset of target translation conditioned on both the input text and a partially masked target

translation. The model predicts all of the target words non-autoregressively in first iteration. In subsequent iteration the decoder repeatedly mask out the produced subset of words. The decoder updates the words with low confidence (Ghazvininejad et al., 2019). Huang et al., proposed iterative refinement over NAT model using knowledge distillation technique (Huang et al., 2022). Their proposed model produces translations with fewer repetition errors, thereby significantly reducing the mismatch between training and inference.

Graph Neural Networks and Graph-Enhanced Transformers

The Sequential Transformer-based model like ART, NAT and iNAT have demonstrated the effectivness in NMT. However, their self-attention mechanisim does not explicitly encode linguistic structure and remain inefficient for long sentences. To address this limitation, recent work has explored graph neural networks (GNNs) and graph-enhanced Transformers. These methods sentences are represented as graphs and information is propagated between connected words through message passing. In this line of research, dependency (syntactic) graphs are used to model grammatical relations between words, whereas semantic graphs aim to capture meaning-level relations beyond word order. In NMT, Bastings et al. incorporated syntactic dependency graphs into the encoder, while Marcheggiani et al. integrated semantic graph representations to enrich word embeddings, both reporting improvements in translation quality. Nevertheless, GNN-based approaches can be sensitive to noisy or unreliable graph structures, which degrade generalization performance.

To mitigate this, recent architectures have proposed Graph-based encoders and graph-aware Transformer variants, including graph attention networks and syntax-aware attention/Transformers are proposed. The transformer with GNN explicit structural signals can strengthen contextual representations and better support long-range dependency modeling compared to purely sequential encoders. Through graph attention networks, graph transformers, or syntax-aware attention formulations, where attention is constrained or re-weighted by graph neighborhoods. That result into more long-range dependency modeling.

However, these GNN based approaches primarily focus on ART architectures and high-resource language pair. Consequently, the integration of graph representations into efficient NAT and iNAT models remains largely unexplored.

Table 3

Research Summary on Urdu MT

Reference	Model	Language pair
(Safder et al., 2024)	Transformer	UR-EN
(Israr et al., 2025)	Transformer	UR-EN
(Israr et al., 2023)	Transformer with subword	UR-EN
(Javed et al., 2025)	Transformer curriculum learning, back-translation, BERT	ZN-UR
(Zuhra et al., 2026)	Transformer with different word embeddings	UR-EN

Urdu-English NMT

In recent years, Transformer models have been widely studied for Urdu-English NMT tasks. Safdar et al. (2024) conducted experiments on English-Urdu translation using the Transformer (Safder et al., 2024). Their proposed method demonstrated strong performance on the Urdu-English translation task.

Israr et al. (2023) has conducted extensive experiments on Urdu-English NMT using LSTM, GRU, CNN, and Transformer models. They proposed a refinement by introducing an attention-based dropout layer in the Transformer model, which resulted in a little improvement in translation accuracy. They also investigated a Transformer that utilizes subword units for Urdu-English MT. Their result demonstrated that subword modeling solve the OOV words problem. When combined with sub wording transformer performs better than the baseline ART transformer. It is experimented with dual-encoder-based Transformer and LSTM models for Urdu-English machine translation. Their study concluded that combining LSTM with a Transformer-based model improves overall translation performance (Israr et al., 2025).

Transformer-based NMT model was introduced by Javed et al. (2025) for Chinese-Urdu translation. The proposed model integrated curriculum learning, back-translation, BERT-based contextual embeddings in the transformer. To address data sparsity and OOV issues, the author employed a variant of the loss function. Their findings demonstrate measurable gains in BLEU scores and improved contextual coherence. Zuhra et al. (2026) proposed a novel English-to-Urdu NMT model based on the Transformer architecture, which leverages the average of three types of word embeddings: Urdu word2vec (skip-gram-based), part-of-speech n-gram (POS-Ngram) embeddings, and POS-POS embeddings. The resultant model generates translation with high accuracy (Zuhra et al., 2026).

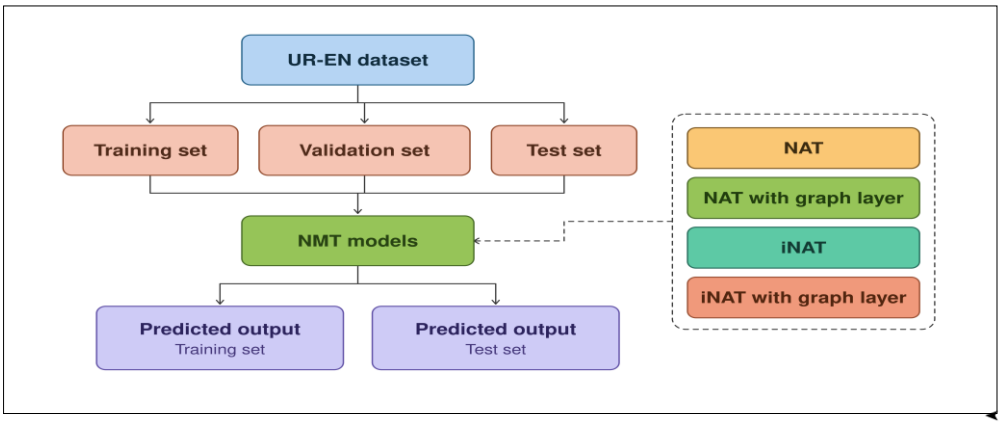
The literature review shows that substantial research has been done on translating different language pairs using autoregressive Transformer (ART) models. Many studies have focused on translating resource-rich languages such as English (EN) , French (Fr), and Chinese (Zn). However, less work has focused on translating the morphologically rich, under-resourced Urdu-English pair. A few studies have explored using autoregressive Transformer models for Urdu-English translation, but NAT and iNAT models have not been applied to this language pair. The present research focuses on addressing limitations highlighted in previous studies by exploring NAT and iNAT for Urdu-English translation and proposes a graph-enhanced NAT and iNAT model for the UR-EN language pair.

THE PROPOSED METHOD

This section of the paper presents the details of the tools used and the implemented NAT and iNAT encoder. The research methodology followed in this study is depicted in Figure 1, which outlines the step-by-step process of the experiments. To clearly describe the methodology and improve comprehension, a list of variables and their descriptions is provided in Table 4.

Figure 1

Model Training and Evaluation Workflow



The workflow of the implemented NAT encoders is illustrated in Figure 2. Figure 3 illustrates the proposed NAT model with an additional graph layer.

Figure 2

Transformer Based Encoder

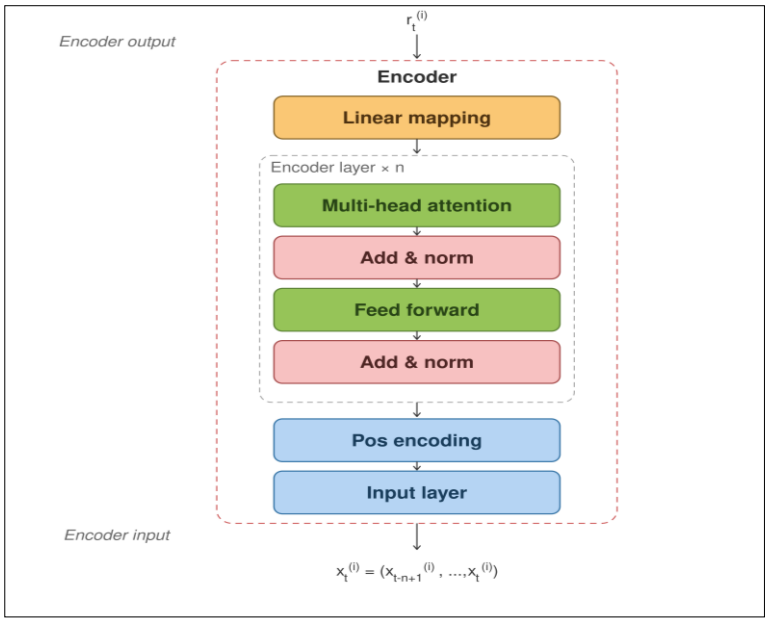


Table 4

List of Variables Used to Describe MT Model

Symbol	Description	Symbol	Description
X_M	Source sentence of length M	Y_N	Target sentence of length N
K	Value for self-attention	q	Query for self-attention
K	Key for Multi-Head Attention	PE	Positional encoding
V	Values for Multi-Head Attention	Q	Query for Multi-Head Attention

Autoregressive Transformer

Autoregressive Transformer is an encoder-decoder architecture for MT that takes source sentence X = (x_1, x_2,...., x_M), represented as word embedding and positional embedding and generates the corresponding target sentence $Y = \{y_1, y_2, \ldots, y_N\}$, as defined in Equation 1. Each word in the input sentence is represented by a real vector called word embedding. As shown in Fig. 1, Transformer models estimate the probability of producing a target sequence Y given the source sequence X, conditioning on the input and earlier target tokens. Equation 2 expresses the joint probability of the target tokens given X. This can be expressed as a chain of conditional probabilities, as illustrated in Equation 3:

$$f(X \rightarrow Y) = P(Y|X) \tag{1}$$

$$P(Y|X) = P(y_1, y_2, y_3 \ldots y_N \mid X; \theta_{AT}) \tag{2}$$

$$\prod_{i=1}^{N} P(\, y_i \mid y_{<i}, X; \, \theta_{AT}\,) \tag{3}$$

Such that y_i represents the token generated at i_{th} time step. The $y<i$ indicate they represent the previously generated tokens, respectively. N represents the length of the entire target sentence. θ_{AT} is the parameter of the auto-regressive transformer model. The $P(y_i \mid X)$ is independent of Y . This property helps the decoder to generate all tokens in parallel.

In particular, the encoder first maps an input sentence $X = \{x_1, x_2, \ldots, x_M\}$ into a continuous hidden representation $H = \{h_1, h_2, \ldots, h_M\}$. The hidden state is then forwarded to the decoder. The decoder decodes the hidden representations and generates the target output $Y = \{y_{1,2}, \ldots, y_N\}$ one element at a time.

At each time step, the prediction is autoregressive. The decoder uses previously predicted tokens as additional inputs while predicting next token. As a result, the generation process is sequential. The output is produced from left-to-right (LTR), generating one word at a time.

Non-Autoregressive Neural Machine Translation

To speed up autoregressive decoding, Gu et al. (2017) proposed the Non-Autoregressive Transformer (NAT). This model generates target tokens independently and simultaneously, which is known as non-autoregressive neural machine translation (Gu et al., 2017). The NAT uses an encoder-decoder framework similar to the Transformer architecture. However, NAT is different from ART. The NAT decoder ignores dependencies among target tokens and assumes that each token is predicted independently. Therefore, all tokens are generated simultaneously in parallel.

For a given source sentence $X = \{x_1, x_2, \ldots x_M\}$, Non-autoregressive Transformer produce corresponding target sentence $Y = \{y_1, y_2, \ldots, y_N\}$ based on assumption of conditional independence. NAT produce the target translation Y from source input X as:

$$f(X \rightarrow Y) = P(Y|X) \tag{4}$$

$$P(Y|X) = P(y_1, y_2, y_3 \ldots y_N \mid X; \theta_{NAT}) \tag{5}$$

$$P(y_1, y_2, y_3 \dots y_N \mid X; \theta_{NAT}) = P(\mathrm{N} \mid X; \theta_{NAT}) \prod_{i=1}^{N} P(y_i \mid y_{<i}, X; \theta_{NAT}) \quad (6)$$

y_i refers to the i-th time step token. N is the length of the target sentence. θ_{NAT} is the parameter of NAT. Equation 4 and Equation 5 in general define the NAT mapping and its joint probability, while Equation 6 factorises the probability.

Proposed Graph Layer (Post-Encoder Implicit Graph Message Passing)

To enhance the model's learning and representation capability, as shown in Fig. 2, a new layer is incorporated above the attention layer (i.e. on top of the standard Transformer encoder stack). This additional graph refinement layer generalizes the model to unseen data by explicitly modeling token-to-token relationships through an attention-derived similarity mechanism. Graph Layer is applied, once after the final encoder layer. The graph layer takes input from the last layer and produced enhanced output.

$$H' = \mathrm{Graph}(H_{encoder}) \quad (7)$$

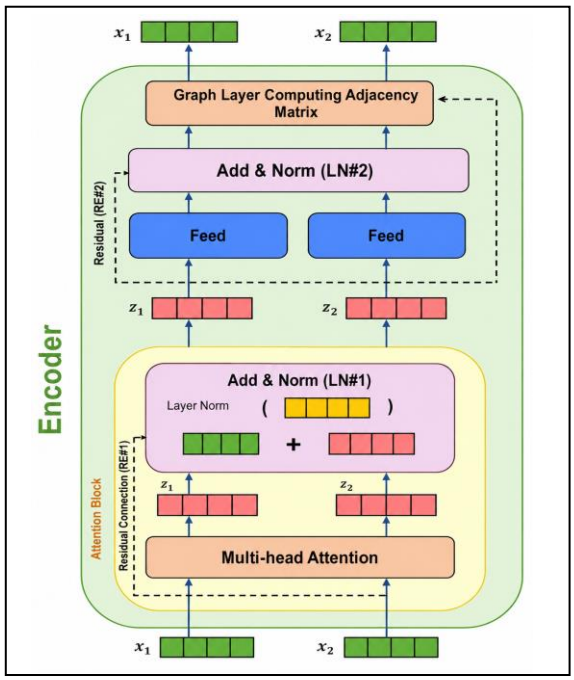
Where $H_{encoder}$ is output of encoder, and $A_{learned}$ graph adjacency computed from attention. H′ is the enhanced output of the encoder. The graph layer refines the encoder output as in Equation 7. Inside GraphLayer, an attention-based adjacency matrix $A_{learned}$ is constructed dynamically from learned Query/Key projections computed from encoder H_{encode} , i.e.,

$$A_{learned} = softmax(\frac{QK^T}{\sqrt{C}}) \quad (8)$$

The graph refinement is applied only once, after the standard Transformer stack. The refined output H′ is then passed to the decoder (through the normal encoder-decoder attention mechanism).

Figure 3

Transformer-Based Encoder with Graph Layer



The output of the last encoder layer is $X \in R^{T \times B \times C}$, where T is sequence length, B is batch size, and C is embedding dimension. The layer first transforms the input into batch-first format. The layer first reshape $X \in R^{T \times B \times C}$ to $X' \in R^{B \times T \times C}$. In the next step, it computes three projections, i.e. Query (Q), Key (K), and Value (V), using linear transformations on the input. The GraphLayer computes projected node features as:

$$Q = X'W_q\ ,\ K = X'W_k\ ,\ V = X'W_v \tag{9}$$

Where $W_q, W_k, W_v \in R^{C \times C}$. Then, the scaled dot product is applied between Q and K, and the calculated score is termed attention scores. The adjacency matrix, given compactly in Equation 8, is built from the query and key projections of Equation 9. Instead of the standard Transformer multi-head mechanism, it forms a single implicit adjacency matrix via scaled dot-product affinities.

$$S = \left(\frac{QK^T}{\sqrt{C}}\right)\ ,\qquad A = softmax(S)\ \in R^{BxTxT} \tag{10}$$

Here, $A_{b,i,j}$ can be interpreted as the (soft, directed) edge weight from token i to token j in the implicit graph for sample b. The single-head adjacency matrix computed from Q and K is without multi-head splitting/concatenation. The scaled dot-product scores and the softmax-normalized adjacency matrix A are defined in Equation 10. In contrast, Transformer self-attention is implemented as multi-head attention with head-wise projections, concatenation, and is integrated inside every encoder layer together with the position-wise FFN and masking mechanisms.

In the last step, a softmax operation is applied to generate an adjacency matrix as shown in Figure 3. This adjacency matrix can be interpreted as a fully connected graph. This captures the global dependencies beyond standard positional encoding. In the adjacency matrix, each token attends to all other tokens.

A message passing (graph propagation) is then performed H = AV, followed by an output projection and stabilization with residual connection and LayerNorm as:

$$Y = LayerNorm(H'W_o\ +\ X')\ \tag{11}$$

Residual connection and layer normalization, is given in Equation 11. Finally, Y is reshaped back to T×B×C and fed to the decoder. Unlike multi-head self-attention, this graph layer uses a single-head implicit adjacency matrix and is applied once as a post-encoder refinement module. The novelty of this layer is the way it interprets the attention weights, representing it as graph connections and applying a graph-based aggregation mechanism. The computed adjacency matrix is used to propagate information by multiplying it with the value vectors, producing enriched node representations. The result is then passed through a linear projection and combined with the original input via a residual connection. The process is then followed by layer normalization. Finally, the output is reshaped back to T×B×C and forwarded to subsequent components. Unlike the standard Transformer, the attention is confined within encoder layers. This additional Graph Layer provides an explicit global relational refinement step. The progressive integration of graph-derived static preferences with transformer-encoded dynamic representations enhances model robustness (Badouch, 2026). It will enable better modelling of complex dependencies in Urdu-English translation tasks.

The overall system is trained using standard sequence-to-sequence NMT training with cross-entropy loss in Equation 12, conditioned on the graph-refined encoder output.

$$\mathcal{L} = -\sum_i log\ p(yt \mid y < t, H') \quad (12)$$

Where H′ is the graph-refined encoder output.

EXPERIMENTAL SETUP

This section describes the dataset, trained models, and tools used. It also explains the training setup, hyperparameters, and evaluation metrics.

Dataset

This study uses the relevant UR-EN dataset collected to verify the effectiveness of the different NMT models. The researcher used the standard training, development, and test splits provided by Israr et al. (2024b) which allow reproducible results and meaningful comparisons with existing studies. The train set consists of 90,000 parallel sentences in UR-EN. The validation and test sets contain 5,000 sentences each. Despite its small size, we use this dataset because it is a clean, versatile, and publicly accessible Urdu-English corpus. Table 5 provides sample sentences from the dataset.

Table 5

UR-EN Dataset

Source sentence (Urdu)	Target translation (English)
ٹام خوش ہے۔	Tom is happy .
میرا پسندیدہ مضمون کیمسٹری ہے ۔	My favorite subject is chemistry .
دونوں ٹیموں کے کپتان پچ پر ملتے ہیں اور یہ فیصلہ کرنے کے لیے ایک سکہ ٹاس کرتے ہیں کہ کون سی ٹیم پہلے بیٹنگ کرے گی یا فیلڈنگ کرے گی۔	The two team captains meet on the pitch and toss a coin to decide which team will bat or field first.
سخت سزائیں اور ان کی خودکار نوعیت قومی پارلیمانوں کی اقتصادی پالیسی پر اثر انداز ہونے کی طاقت کو محدود کرتی ہے۔	The strict penalties and their automatic nature limit the power of national parliaments to influence economic policy .

Model Trained

The researcher trained two variation of ART, namely NAT and iNAT. Then, the researcher adopted the model proposed by the Gu et al., (2017) as the NAT model. For the iNAT model, the researcher followed the approach proposed by Lee et al. (2018), which uses iterative refinement to improve translation quality. Finally, incorporated the proposed graph layer into both the NAT and iNAT models (Gu et al., 2017; Lee et al., 2018).

1. Non-autoregressive Transformer (NAT)
2. Non-autoregressive Transformer with iterative refinement (iNAT)
3. NAT with graph layer
4. iNAt with graph layer

The result of the produced model was compared with standard Transformer model implemented by Israr et al. (2023). The main goal of this experiment is to ensure a fair and consistent comparison of all models. This is achieved by analyzing and evaluating their performance on the Urdu--English dataset using the same experimental settings.

Model Configurations

NAT and iNAT models are implemented based on the Transformer model using Fairseq (Ott et al., 2019). We follow the Fairseq examples to train the Transformer models. Both models are trained for 10 epochs. After training, both models were used to translate the Urdu sentences in the validation and test sets. The encoder and decoder in our transformer-based model are each composed of eight layers. This architecture helps the model extract rich features from the input and improves translation quality. The model uses a word vector size of 512. During training, the Adam optimizer was employed with a learning rate of $5e^{-4}$ to update model parameters. The researcher used drop out rate 0.3 to reduce over fitting during training. The hyper-parameter used in our experiments is shown in Table 6.

Table 6

Common Parameters Used for Training Transformer-Based Models

Serial	Parameter	Specification	Serial	Parameter	Specification
1.	Number of Layers	512	6.	adam-betas	(0.9, 0.98)
2.	Embedding size	512	7.	Attention head	8
3.	Epoch	10	8.	criterion	nat loss
4.	Learning rate	5e-4	9.	dropout	0.3
5.	Optimizer	adam	10.	weight-decay	0.0001

Tools

The proposed model is trained and evaluated using Fairseq. Fairseq is a Py-Torch based toolkit designed for developing and evaluating sequence modeling NMT models. This research employed the Fairseq-py implementation for NAT and iNAT models. The models are trained and tested on a computer system equipped with an NVIDIA GeForce GTX 1650 SUPER GPU and an Intel® Core™ i9-9900K CPU at 3.60 GHz.

Evaluation Metrics

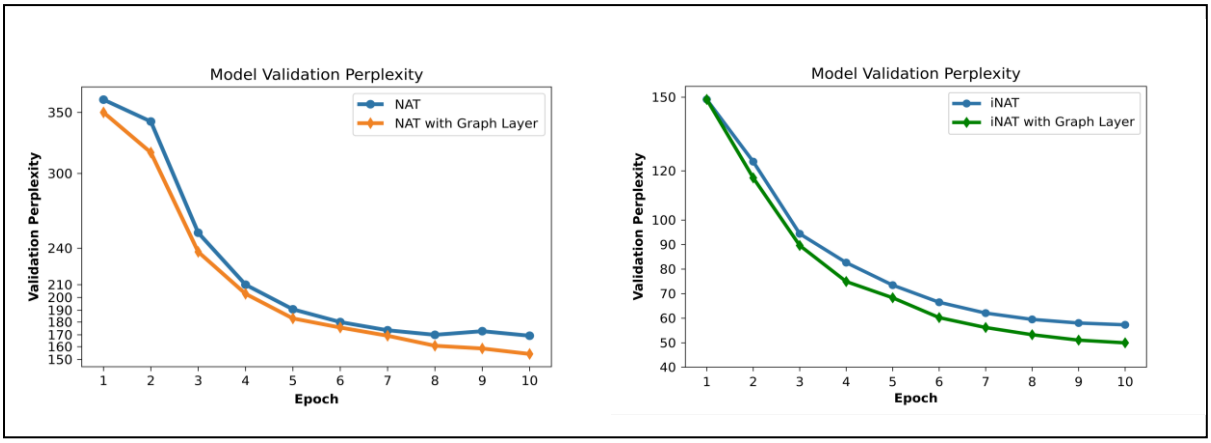
Perplexity was used to assess how well the model learns during training. To conduct quantitative verification of the produced translation, the researcher used the BLEU score metric. The results with different metrics has been evaluated together with the BLEU score. Five additional metrics was being considerd in the experiments: METEOR, WER, TER, GLUE and ROUGE score . The performance of the models is finally analyzed by applying BLEURT. Based on the BERT model, BLEURT compares candidate text to reference text with high accuracy (Sellam et al., 2020). BLEURT analyses the semantic fidelity and fluency of translations compared to the reference text, producing scores that closely correlate with human judgments.

EVALUATION AND RESULTS

This part of the study evaluates and compares the performance of NAT and iNAT models with the Transformer model proposed in the paper "Neural Machine Translation with Attention-based Dropout Layer" (Israr et al., 2023). The results are evaluated on the same Urdu-English dataset. This research introduces an "additional layer called a graph layer" in the Transformer encoder architecture. The proposed models include NAT with a graph layer and iNAT with a graph layer. The additional layer enhances the quality of the learned feature representations (Garmastewira, 2019). These models are compared with several baseline models.

Figure 4

Validation Perplexity of NAT, iNAT and Proposed Model with Graph Layer



Transformer with Attention-based Dropout Layer. The purpose of this evaluation is to analyse the effectiveness of the proposed framework: validation perplexity of NAT, NAT with graph layer, validation perplexity of iNAT, and iNAT with graph. To gain a better insight into how well each model generalized beyond the training data, the learning behaviour of the proposed model was analyzed by tracking step-wise perplexity (PPL) during training and validation (Jelinek et al., 1977).

Figure 4 illustrates the perplexity values recorded during the validation phase after each epoch. In Figure 4, the validation perplexity of NAT with the graph layer is consistently lower than that of the standard NAT model across all epochs. This indicates that the graph layer helps the model learn better contextual relationships between tokens. As training progresses, the NAT with the graph layer converges to a lower perplexity value. This shows that the proposed graph-enhanced NAT model performs better than the baseline NAT. Figure 4 shows that iNAT with the graph layer achieved lower perplexity compared to the standard iNAT model. The improvement is visible across most training epochs. The graph layer helps the model capture additional contextual dependencies. As a result, the proposed graph-enhanced iNAT model demonstrates better validation performance.

Table 7

Each Model Training and Validation Perplexity on the Last Epoch

Statistics	Transformer	Transformer-ADL	NAT	iNAT
Training Perplexity	112	98	82.62	32.05
Validation Perplexity	192	183	169.3	57.2

Table 7 presents the results from the final training epoch, where the iNAT model reached a minimum training perplexity of 32.05 and a validation perplexity of 57.2. The NAT model, on the other hand, reached a training perplexity of 82.62 and a validation perplexity of 169.3, indicating comparatively lower performance than the iNAT model. The Transformer and Transformer-ADL models reported higher perplexity values on both training and validation sets. This indicates that the iNAT model learns the data distribution more effectively than the other models. In neural machine translation, lower perplexity indicates better predictive capability and improved generalization performance.

Table 8

Model with New Graph Layer Training and Validation Perplexity on Last Epoch

Statistics	NAT with graph layer	iNAT with graph layer
Training Perplexity	76.57 ↓	27.79 ↓
Validation Perplexity	154.17 ↓	49.95 ↓

Notes. The upward and downward arrows indicate increases and decreases, respectively, in the evaluation metrics.

As shown in Table 8, adding the graph layer further improves the performance of both NAT and iNAT models. The NAT with graph layer achieves a training perplexity of 76.57 and a validation perplexity of 154.17, which are lower than the baseline NAT model. Similarly, iNAT with a graph layer achieves a training perplexity of 27.79 and a validation perplexity of 49.95. This shows a clear improvement compared to the standard iNAT model.

Table 9 (a)

Baseline Transformer Model and the Selected NAT & iNAT Models Training and Validation Perplexity

Result on Test-set	Transformer	Transformer-ADL	NAT	iNAT
BLEU Score	1.27	2.02	0.069	15.25*
GLUE-Corpus	0.039	0.40	0.118	0.207 *
GLUE-sentence average Score	0.049	0.059	0.163	0.265*
ROUGE_L Score	0.156	0.181	0.373	0.506*
METEOR	0.048	0.060	0.159	0.230*
TER	1.005	1.257	0.749	0.613*
WER	24.00	31.00	19.18	16.07*
Precision:	0.172	0.138	0.357	0.492*
Recall:	0.107	0.141	0.364	0.493*
f1:	0.132	0.139	0.361	0.493*
fMean:	0.114	0.140	0.363	0.493*
Bleu_1:	12.80	16.20	0.346	0.461*
Bleu_2:	04.20	06.60	0.181	0.307*
Bleu_3:	02.10	03.50	0.107	0.212*
Bleu_4:	01.27	02.20	0.072	0.151*

Notes. The * symbol highlights the best-performing model among the four.

Table 9 (a) presents a comparative evaluation of the baseline Transformer model and the selected NAT and iNAT models on the Urdu-English test set. Specifically, it includes the Vanilla Transformer, Transformer-ADL, NAT, and iNAT models. The performance is evaluated using BLEU, GLUE, ROUGE-L, METEOR, TER, and WER metrics. Higher BLEU, GLUE, ROUGE-L, and METEOR scores indicate better translation performance, while lower values of TER and WER represent improved translation quality (Israr et al., 2023).

The experiments indicate that the iNAT model achieves a clear performance improvement over the baseline (vanilla) Transformer and Transformer-ADL, achieving a BLEU score of 15.25, which highlights the advantage of iterative refinement for Urdu-English translation. The iterative refinement allows model to progressively correct and enhance initial translations, thereby producing more accurate and fluent outputs.

Table 9 (b) presents a comparative evaluation of the baseline NAT and iNAT models and the proposed models with an additional graph layer on the Urdu-English test set. The models are compared with the proposed graph-enhanced NAT and iNAT models to analyse the impact of the additional layer.

Table 9 (b)

Proposed Model Training and Validation Perplexity

Result on Test-set	NAT	iNAT	NAT with graph layer	iNAT with graph layer
BLEU Score	0.069	15.25	0.092 ↑	18.26 ↑
GLUE-Corpus	0.118	0.207	0.148 ↑	0.227 ↑
GLUE-sentence average Score	0.163	0.265	0.176 ↑	0.325 ↑
ROUGE_L Score	0.373	0.506	0.401 ↑	0.523 ↑
METEOR	0.159	0.230	0.181 ↑	0.380 ↑
TER	0.749	0.613	0.721 ↓	0.603 ↓
WER	19.18	16.07	18.36 ↓	15.64 ↓
Precision:	0.357	0.492	0.402 ↑	0.768 ↑
Recall:	0.364	0.493	0.409 ↑	0.777 ↑
f1:	0.361	0.493	0.406 ↑	0.772 ↑
fMean:	0.363	0.493	0.408 ↑	0.776 ↑
Bleu_1:	0.346	0.461	0.402 ↑	0.504 ↑
Bleu_2:	0.181	0.307	0.222 ↑	0.344 ↑
Bleu_3:	0.107	0.212	0.135 ↑	0.247 ↑
Bleu_4:	0.072	0.151	0.092 ↑	0.186 ↑

Notes. The upward and downward arrows indicate increases and decreases, respectively, in the evaluation metrics.

The results show that the NAT model with the graph layer outperforms the standard NAT model across all evaluation metrics. Similarly, the proposed iNAT model with the graph layer achieves better performance compared to the baseline iNAT model. Among all six models, the graph-enhanced iNAT model demonstrates the best overall performance, showing consistent improvements across all metrics. The improvements are reflected in translation-quality and overlap-based scores (BLEU, ROUGE_L, METEOR), as well as task-level alignment metrics (Precision/Recall/F1, Bleu_n, WER/TER reductions).

Furthermore, the proposed graph-enhanced models provide additional improvements over their baseline counterparts. In particular, iNAT with the graph layer achieves the best performance, obtaining the highest BLEU score of 18.26, improved ROUGE-L (0.52251), and a higher METEOR score (0.37989). The proposed graph layer also reduces the TER and WER in output translation. Similarly, NAT with the graph layer shows consistent improvements across several metrics compared with the classical NAT model.

BLEU-1 represents unigram (1-gram) word overlap, achieving 50.0%, which indicates a strong lexical match. BLEU-2 represents bigram (2-gram) overlap, achieving 34.4%, showing significant improvement in phrase-level matching. BLEU-3 represents trigram (3-gram) overlap, achieving 24.6%, reflecting better capture of longer phrases. BLEU-4 represents four-gram overlap, achieving 18%, which indicates improved fluency and more coherent sentence generation.

Overall, the results demonstrate that incorporating a graph learning layer helps the model capture richer contextual relationships between tokens, leading to more accurate and fluent translations. The proposed graph layer explicitly models relationships between non-adjacent tokens by constructing a structured representation of the input sequence. This allows the model to reason beyond the immediate sequential context. Meanwhile, adjacency metrics are learned through the attention mechanism, which captures local and sequential dependencies. Together, these components allow the model to integrate both local sequential context and long-range relational context, resulting in better translation quality. The graph-enhanced iNAT model achieves the best overall performance, indicating that combining iterative refinement with graph-based contextual modelling is effective for improving Urdu-English MT.

Table 10

Over/Under-Translation Score of Test Set

Model	OTEM/2	UTEM/4
NAT	0.436	0.873
NAT with graph layer	0.409 ↓	0.853 ↓
iNAT	0.401	0.740
iNAT with graph layer	0.389 ↓	0.737 ↓

Table 10 reports the OTEM/2 (over-translation evaluation metric) and UTEM/4 (under-translation evaluation metric) scores obtained by the baseline and graph-enhanced models (Yang et al., 2018). These metrics evaluate the degree of over-translation and under-translation in the unseen test set. The lower values indicate better translation quality. The results show that incorporating a graph layer improves translation behaviour in addition to improving the BLEU score. In particular, the NAT with graph model achieved a lower OTEM score (0.409) compared to the baseline NAT model (0.436), indicating a reduction in over-translation. Similarly, the UTEM/2 score decreased from 0.873 to 0.853, suggesting fewer under-translation errors.

A similar improvement trend can be observed in other models as well. The graph-enhanced iNAT model maintains competitive UTEM/2 performance while improving translation consistency. The results demonstrate that integrating graph-based representations helps the model to better capture the contextual relationships in the source sentence. It reduces both over-translation and under-translation errors.

Three example sentences are selected from the test set to evaluate translation quality. The assessment compares the NAT and iNAT models with the proposed graph layer against the standard NAT and iNAT models for the Urdu-to-English translation task. These example sentences have been previously examined for translation quality in prior work (Israr et al., 2023, 2025; Israr et al., 2024a).

The translations produced by each model for the sample sentences are shown in Table 11, Table 12, and Table 13. We evaluate translation quality using BLEURT-20 score as the primary success criterion (Sellam et al., 2020). Comparing the results of the NAT, iNAT, and the same models with an additional graph layer shows that using an additional graph layer on the encoder layer improves translation performance.

Table 11

Sentence No. 01 Translation

SENT:1	" حکومت نے ملک میں لانگ مارچ روکنے کے لئے ہزاروں پولیس کارکنوں کو تعینات کیا ۔"	BLEURT-20 Score
Gold Reference	"The Government deployed thousands of Police workers to stop long march in the country ."	--
iNAT with graph layer	The government thousands thousands of police police deployed deployed to prevent prevent in in in the March .	0.455
iNAT	The government police order thousands workers workers workers workers preventing March March March March March March March March .	0.404
NAT with graph layer	The government has thousands thousands thousands thousands of to in March prevent in March Marc .	0.401
NAT	The governent has more police thousands to March March March March again again in in in the work .	0.384
TRF	I would like to thank you for your support .	0.288
TRF with ADL	It is not enough to interfere in society .	0.347

The propose model generated a translation for source sentence, "حکومت نے ملک میں لانگ مارچ روکنے کے لئے ہزاروں پولیس کارکنوں کو تعینات کیا ۔ " , "The government deployed thousands of police workers to stop a cooperative March in the country ". "A detailed analysis of sentence No.01's translation is provided in Table 11, which indicates that the model iNAT with graph layer produced comparatively better translations with high accuracy than the other models. In particular, for sentence No. 01, the iNAT with graph model achieved a BLEURT score of 0.4546, which is higher than the scores obtained by the iNAT (0.4045), NAT with graph (0.4005), and NAT (0.3839) models, respectively. The generated output still contains some repeated words. However, the iNAT with graph layer preserves more relevant words from the source sentence than the baseline NAT models.

In contrast, the Transformer produced translations that are semantically less related to the source sentence. For instance, the Transformer generated the sentence "I would like to thank you for your support." with a BLEURT score of 0.288, while the Transformer with ADL produced "It is not enough to interfere in society." with a BLEURT score of 0.3465. These outputs deviate significantly from the original source meaning. Overall, the results suggest that incorporating the graph layer helps the NAT-based models better preserve the contextual information of the source sentence. It improved the translation quality as reflected by higher BLEURT scores.

The translations generated by the NAT and iNAT models contain excessive repetition, such as “thousands thousands” and “March March,” making the output incoherent and difficult to understand. In contrast, the graph-enhanced iNAT model, despite some repeated words, produces a more structured translation that better preserves the meaning of the source sentence.

Table 12

Sentence No. 02 Translation

SENT:2	“ کیا آپ تھوڑا سا اونچی آواز میں بول سکتے ہیں ؟”	BLEURT-20 Score
Gold Reference	“ Can you speak a little louder ?”	--
iNAT with graph layer	Can you speak in a bit a high loud ?	0.539
iNAT	Can you speak a speak out high ?	0.429
NAT with graph layer	speak in you a high loud ?	0.396
NAT	speak in a a a a high ?	0.287
TRF	I am sorry .	0.270
TRF with ADL	why ?	0.226

As illustrated by sentence No. 02 in Table 12, the BLEURT scores for the translations generated by the NAT and NAT with graph layer is 0.2867 and 0.3960, respectively, indicating low translation quality. The iNAT model achieved a higher BLEURT score of 0.4290, while the iNAT with graph model obtained the highest BLEURT score of 0.5386 for the same sentence, demonstrating a clear improvement in translation quality when the graph layer is incorporated. The Transformer-based models performed poorly on this sentence, with BLEURT scores of 0.2696 and 0.2262 for the baseline and ADL-enhanced versions, respectively.

In terms of translation quality, the outputs generated by the NAT and NAT with graph layer models are largely incoherent, repeating or misordering words, such as “a a a a” and “speak in you,” making the sentence difficult to understand. The iNAT model produces a more structured sentence, though some phrases like “speak out high” are unusual. In contrast, the iNAT with graph layer translation, “Can you speak in a bit a high loud?”, maintains grammatical structure and conveys the meaning of the source sentence more clearly, demonstrating that the graph layer helps preserve semantic and syntactic coherence. However, the generated sentence is not fully grammatically correct in its current form.

Table 13

Sentence No. 03 Translation and BLEURT Score

SENT:3	“ کلاسیکی معاشیات میں ، یہ خیال کیا جاتا ہے کہ معیشت بنیادی طور پر دولت کے علم کے بارے میں ہے . ”	BLEURT-20 Score
Gold Reference	“In classical economics , it is believed that the economy is primarily about the knowledge of wealth .”	--
iNAt with graph layer	In in economies it is is is primarily mainly about knowledge knowledge the economy . .	0.319
iNAT	In knowledge in is is is that is knowledge knowledge knowledge wealth . . .	0.250
NAT with graph layer	Just the economics economics economics economics economics is about knowledge knowledge knowledge of the principles	0.281
NAT	It is in in is is is is knowledge knowledge wealth wealth wealth economy .	0.279
TRF	I would like to congratulate the rapporteur .	0.095
TRF with ADL	It is also important for us to see the principles .	0.282

For sentence No. 03, the translations presented in Table 13 show that the iNAT with graph layer achieved the highest BLEURT score of 0.319, followed by the NAT with graph (0.281), NAT (0.279), and iNAT (0.250) models. The Transformer-based models performed poorly, with BLEURT scores of 0.282 for Transformer with ADL and 0.095 for the baseline Transformer. These results indicate that incorporating the graph layer improves the model's semantic similarity to the reference translation.

Qualitative analysis reveals that the NAT and iNAT outputs, although containing repeated words and phrases like "is is is" and "knowledge knowledge," still retain some key terms from the source sentence. In contrast, the iNAT with graph translation, despite minor repetitions, better captures the original meaning about classical economics and the knowledge of wealth, producing a translation that is more understandable and aligned with the source. The overall translations, however, largely fail to convey the intended meaning.

Table 14

Summary of Comparison

Reference	Model trained	BLEU score
Israr et al., (2023)	Transformer	01.27
	Transformer-ADL	02.02
Proposed Model	NAT	06.92
	iNAT	15.25
	NAT with graph layer	09.15 ↑
	iNAT with graph layer	18.26 ↑

The comparison between the results of the ART, NAT and iNAT models trained on the same dataset is illustrated in Table 14. All transformer-based models are trained on the identical Urdu-English dataset introduced in earlier studies (Israr et al., 2023). The BLEU differences in Table 14 reflect systematic variations in model inference mechanisms. Transformer models perform fully autoregressive decoding, whereas NAT/iNAT generate targets in a parallel and refinement-based manner, which changes how errors are introduced and corrected during inference. Because BLEU is sensitive to exact n-gram overlap, these decoding dynamics can produce large changes in lexical matching even when broader fluency and global syntactic quality remain limited. NAT/iNAT augmented with a graph layer further improves the BLEU score. This suggests that the graph layer enhances the model's ability to capture more structured relationships during refinement, leading to better target reconstruction and therefore stronger exact n-gram alignment.

In nutshell, the findings reported in this section emphasise the effectiveness of the proposed graph enhanced layer for NMT of low-resource language pairs such as Urdu-English. Proposed graph-enhanced model shows consistent performance gains over baseline models across all evaluation metrics on unseen test datasets. In addition, example translations showed that graph enhanced generate more fluent and coherent outputs.

CONCLUSION

This study evaluates ART, NAT and iNAT on Urdu-English NMT. iNAT enhances translation quality as compared to NAT. This makes it suitable for real-world applications that balance accuracy and performance requirements. The proposed work presents a novel approach to enhancing NMT for low-resource languages, with a specific focus on Urdu-English translation. We propose a novel model, iNAT, with an additional graph layer for UR-EN MT. The graph layer captures token relationships via an adjacency matrix. This helps the model produce more fluent translations with reduced repetition. The proposed graph layer improves the accuracy of the non-autoregressive transformer-based model and its variants. The proposed model achieves consistent improvements, showing reduced perplexity and higher BLEU scores. The proposed model achieved better accuracy as compared to classical ART and NAT. On the Urdu-to-English translation task, the trained model achieves better performance compared to the baseline models. Specifically, it surpasses classical NAT and iNAT models by +0.03 and +3.0 BLEU points, respectively. While these results are encouraging, one limitation of the approach is that iNAT with the additional graph layer can produce translations that lack fluency. The output often contains repetitive segments, which negatively affect readability and result in unnatural phrasing.

Although this study primarily investigates the effect of incorporating a graph layer into the iNAT model, future work will examine its impact on model complexity and translation speed. There is also a need to evaluate the proposed approach on additional datasets, including high-resource language pairs such as English–German (EN–DE), to assess its generalizability and robustness across different translation scenarios. The promising accuracy achieved on the small, low-resource Urdu–English dataset provides a foundation for further investigation of the proposed work.

ACKNOWLEDGMENT

The researchers gratefully acknowledge the Department of Computing and Emerging Technology (DCET), National Skills University Islamabad, for providing the computational resources and support necessary to conduct this research.

Non-funded

This study did not receive any specific funding from public, commercial, or nonprofit organizations.

REFERENCES

Badouch, M., Boulmane, E. S., Boutaounte, M., & Mahmoud, H. (2026). Graph-augmented transformer framework for context-aware recommender systems on Amazon e-commerce data. *Journal of Information and Communication Technology, 25*(2), 50–69. https://doi.org/10.32890/jict2026.25.2.3

Banerjee, A., & Banik, D. (2025). A comprehensive survey on transformer-based machine translation: Identifying research gaps and solutions for large language models. *ACM Computing Surveys, 58*(5), 1–33. https://doi.org/10.1145/3773076

Basit, A., Azeemi, A. H., & Raza, A. A. (2024). Challenges in Urdu machine translation. *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*. https://doi.org/10.18653/v1/2024.loresmt-1.4

Bilal, M., Israr, H., Shahid, M., & Khan, A. (2016). Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, decision tree, and KNN classification techniques. *Journal of King Saud University–Computer and Information Sciences, 28*(3), 330–344. https://doi.org/10.1016/j.jksuci.2015.11.003

Busto-Castiñeira, A., González-Castaño, F. J., García-Méndez, S., & de Arriba-Pérez, F. (2024). Predictability and causality in Spanish and English natural language generation. *IEEE Access, 12*, 132521–132532. https://doi.org/10.1109/ACCESS.2024.3420710

Dixit, R. (2023). *A comprehensive review of transformer models and their implementation in machine translation specifically on Indian regional languages* (SSRN No. 4449023). https://doi.org/10.2139/ssrn.4449023

El Maazouzi, Z., El Mohajir, B. E., & Al Achhab, M. (2017). A systematic reading in statistical translation: From statistical machine translation to neural translation models. *Journal of Information and Communication Technology, 16*(2), 408–441. https://doi.org/10.32890/jict2017.16.2.8239

Gangar, K., Ruparel, H., & Lele, S. (2021). Hindi to English: Transformer-based neural machine translation. In *International Conference on Communication, Computing and Electronics Systems (ICCCES 2020)*. https://doi.org/10.1007/978-981-33-4909-4_25

Garmastewira, G., & Khodra, M. L. (2019). Summarizing Indonesian news articles using graph convolutional networks. *Journal of Information and Communication Technology*, *18*(3), 345–365. https://doi.org/10.32890/jict2019.18.3.4675

Ghazvininejad, M., Levy, O., Liu, Y., & Zettlemoyer, L. (2019). Mask-predict: Parallel decoding of conditional masked language models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/D19-1633

Gu, J., Bradbury, J., Xiong, C., Li, V. O. K., & Socher, R. (2017). *Non-autoregressive neural machine translation*. arXiv. https://doi.org/10.48550/arXiv.1711.02281

Gu, J., & Kong, X. (2021). Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. https://doi.org/10.18653/v1/2021.findings-acl.11

Huang, X. S., Perez, F., & Volkovs, M. (2022). Improving non-autoregressive translation models without distillation. In *International Conference on Learning Representations*.

Inaguma, H., Higuchi, Y., Duh, K., Kawahara, T., & Watanabe, S. (2021). *Non-autoregressive end-to-end speech translation with parallel autoregressive rescoring*. arXiv. https://doi.org/10.48550/arXiv.2109.04411

Israr, H., Khan, S. A., Tahir, M. A., Shahzad, M. K., Ahmad, M., & Zain, J. M. (2023). Neural machine translation models with attention-based dropout layer. *Computers, Materials & Continua, 75*(2). https://doi.org/10.32604/cmc.2023.035814

Israr, H., Khan, S. A., Tahir, M. A., Shahzad, M. K., Ahmad, M., & Zain, J. M. (2025). Knowledge grounded attention-based neural machine translation model. *Applied Computational Intelligence and Soft Computing, 2025*(1), Article 6234949. https://doi.org/10.1155/2025/6234949

Israr, H., Parvaz, N., & Khan, S. A. (2024a). Urdu-to-English neural machine translation using transformer with subword tokenization. *Information Sciences Letters, 13*(2). https://doi.org/10.18576/isl/130501

Israr, H., Shahzad, M. K., & Anwar, S. (2024b). Improved Urdu-English neural machine translation with a fully convolutional neural network encoder. *International Journal of Mathematical, Engineering & Management Sciences, 9*(5). https://doi.org/10.33889/ijmems.2024.9.5.056

Javed, A., Zan, H., Mamyrbayev, O., Abdullah, M., Ahmed, K., Oralbekova, D., Dinara, K., & Akhmediyarova, A. (2025). Transformer-based re-ranking model for enhancing contextual and syntactic translation in low-resource neural machine translation. *Electronics, 14*(2), 243. https://doi.org/10.3390/electronics14020243

Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (1977). Perplexity—A measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America, 62*(S1), S63. https://doi.org/10.1121/1.2016299

Kamatala, S., Jonnalagadda, A. K., & Naayini, P. (2025). Transformers beyond NLP: Expanding horizons in machine learning. *IRE Journals, 8*(7). https://doi.org/10.2139/ssrn.5112305

Kasai, J., Cross, J., Ghazvininejad, M., & Gu, J. (2020a). Non-autoregressive machine translation with disentangled context transformer. In *International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2001.05136

Kasai, J., Pappas, N., Peng, H., Cross, J., & Smith, N. A. (2020b). *Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation.* arXiv. https://doi.org/10.48550/arXiv.2006.10369

Khan, A. M., & Khan, R. (2025). *A comprehensive review on English-to-Urdu and English-to-Arabic machine translation techniques*. https://doi.org/10.29284/bnrrgt42

Lakew, S. M., Cettolo, M., & Federico, M. (2018). A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *Proceedings of the 27th International Conference on Computational Linguistics*.

Lee, J., Mansimov, E., & Cho, K. (2018). Deterministic non-autoregressive neural sequence modeling by iterative refinement. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/D18-1149

Li, F., Chen, J., & Zhang, X. (2023). A survey of non-autoregressive neural machine translation. *Electronics, 12*(13), 2980.

Lim, Y.-S., Park, E.-J., Song, H.-J., & Park, S.-B. (2022). A non-autoregressive neural machine translation model with iterative length update of target sentences. *IEEE Access, 10,* 43341–43350. https://doi.org/10.1109/ACCESS.2022.3169419

Liu, Y., Wan, Y., Zhang, J., Zhao, W., & Yu, P. S. (2021). Enriching non-autoregressive transformer with syntactic and semantic structures for neural machine translation. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. https://doi.org/10.18653/v1/2021.eacl-main.105

Mahsuli, M. M., Khadivi, S., & Homayounpour, M. M. (2023). LENM: Improving low-resource neural machine translation using target length modeling. *Neural Processing Letters, 55*(7), 9435–9466. https://doi.org/10.1007/s11063-023-11208-1

Norouzi, S., Hosseinzadeh, R., Perez, F., & Volkovs, M. (2023). DiMS: Distilling multiple steps of iterative non-autoregressive transformers for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*. https://doi.org/10.18653/v1/2023.findings-acl.542

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). Fairseq: A fast, extensible toolkit for sequence modeling. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations.* https://doi.org/10.18653/v1/N19-4009

Safder, I., Abu Bakar, M., Zaman, F., Waheed, H., Aljohani, N. R., Nawaz, R., & Hassan, S. U. (2024). Transforming language translation: A deep learning approach to Urdu–English translation. *Journal of Ambient Intelligence and Humanized Computing, 15*(10), 3651–3662. https://doi.org/10.1007/s12652-024-04839-2

Schmidt, R., Pires, T., Peitz, S., & Lööf, J. (2022). Non-autoregressive neural machine translation: A call for clarity. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2022.emnlp-main.179

Sellam, T., Das, D., & Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Shahriar, A., & Barbosa, D. (2024). Improving Bengali and Hindi large language models. *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. https://doi.org/10.63317/525zh4mzrj87

Stern, M., Chan, W., Kiros, J., & Uszkoreit, J. (2019). Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.1902.03249

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., & Liu, T.-Y. (2019). Non-autoregressive machine translation with auxiliary regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v33i01.33015377

Wang, Z., Song, R., Yu, Z., Mao, C., & Gao, S. (2025). DRA: Dynamic routing attention for neural machine translation with low-resource languages. *International Journal of Machine Learning and Cybernetics, 16*(10), 7147–7161. https://doi.org/10.1007/s13042-024-02206-3

Xiao, Y., Guo, P., Sun, Z., Li, J., Song, K., & Zhang, M. (2025). An empirical study of iterative refinements for non-autoregressive translation. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers). https://doi.org/10.18653/v1/2025.acl-long.1443

Yang, J., Zhang, B., Qin, Y., Zhang, X., Lin, Q., & Su, J. (2018). OTEM & UTEM: Over- and under-translation evaluation metrics for neural machine translation. In *CCF International Conference on Natural Language Processing and Chinese Computing.* https://doi.org/10.1007/978-3-319-99495-6_25

Zaki, M. Z. (2023). Transforming worlds: The intersection of translation technology and transformers. *Journal of Language and AI Technologies, 4*, 112–128. https://doi.org/10.5121/cseij.2024.14301

Zaki, M. Z. (2024). Revolutionising translation technology: A comparative study of variant transformer models—BERT, GPT, and T5. *Computer Science and Engineering – An International Journal*, *14*(3), 15–27. https://doi.org/10.5121/cseij.2024.14302

Zamin, N., Oxley, A., & Bakar, Z. A. (2013). Projecting named entity tags from a resource-rich language to a resource-poor language. *Journal of Information and Communication Technology, 12*, 121–146. https://doi.org/10.32890/jict.12.2013.8140

Zhou, C., Neubig, G., & Gu, J. (2019). *Understanding knowledge distillation in non-autoregressive machine translation*. arXiv. https://doi.org/10.48550/arXiv.1911.02727

Zuhra, F. T., Din, S. J. U., Ali, H., Naz, S., Rasool, S., & Idrees, F. (2026). English-to-Urdu neural machine translation using transformer with averaged word embeddings. *VAWKUM Transactions on Computer Sciences, 14*(1), 1–14. https://doi.org/10.21015/vtcs.v14i1.2213