Chapter 6

Incorporating Reinforcement Learning in Adversarial Network

6.1 Introduction

As we have seen in the previous chapter, RL works as a good approach in selecting pseudo-in-domain sentences from out-of-domain corpora by optimising the weights of sentences. However, for improving MTs, there is a need to focus on phonetic features in addition to textual data, which mitigates large corpus requirements and improves context. So, in order to mitigate corpus requirements and improve context learning on extremely LRLs, this chapter introduces a novel improvement by retrofitting a deep reinforcement-based optimized attention network and joint embedding of textual and phonetic information of languages in the GAN-NMT. In GAN-NMT, one pretrained NMT model is used as generator which generates translations based on source sentences and another neural network is used as discriminator that distinguishes whether a generated translation sentences is real or fake, i.e., machine-generated translation sentences is close to human translated sentences or not.

Since, Reinforcement Learning (RL) acts as an effective approach in various Natural Language Processing (NLP) tasks [117], thus, we use it as a deep RL for obtaining

better attention weights by designing a novel GAN model that consist of the Deep RL Guided Attention (DRGA) [57] as generator model and Convolution Neural Network (CNN) [118] as discriminator model. We train the GAN model on a novel Joint Embedding (JE) instead of traditional word embedding representations. Our proposed JE is created by the concatenation of orthographic subword embedding and phonetic subword embedding of input vectors. We use International Phonetic Alphabet (IPA)[1] for converting the textual information into phonetic information. Some examples of IPA conversion are listed in Fig. 6.1. We also design the novel hypothesis selection method to select the best sentences generated by GAN model between textual and phonological representations.

Language	Sentences	IPA
English	Ram is reading a book.	ræm ɪz ˈriːdɪŋ ə bʊk.
Hindi	आज, मुझे शिक्षा के इस मंदिर में जाने का सम्मान प्राप्त हुआ है।	ɑːd͡ʒə, mud͡ʒʰeː ʃɪkʂɑː keː ɪʂə mə̃ɖɪrə mɛ̃ː d͡ʒɑːɲeː kɑː ʂəmmɑːɲə prɑːpt̪ə ɦuɑː ɦəɪ.
Gujarati	આજે, મને આ શિક્ષણના મંદિરની મુલાકાત લેવાનું સન્માન પ્રાપ્ત થયું છે.	ɑːd͡ʒeː, mənɲeː ɑː ʃɪkʂəɳəɳɑː mə̃ɖɪrəɲiː mulɑːkɑːt̪ə leːvɑːɲũ ʂənmɑːɲə prɑːpt̪ə t̪ʰəjũ t͡ʃʰeː.
Nepali	राम िकताब पढ्दैछ।	rɑːmə ɪkət̪ɑːbə pəɖʰɖ̪əɪt͡ʃʰə.
Punjabi	ਇਸ ਨਾਲ ਫਾਇਲ ਪਹਿਲਾਂ ਵੀ ਮੌਜੂਦ ਹੈ ਅਤੇ ਤੁਹਾਨੂੰ ਫਾਇਲ ਦੇ ਉੱਪਰ ਲਿਖ ਲਈ ਅਧਿਕਾਰ ਨਹੀਂ ਹਨ।	ɪʂə ɲɑːlə pʰɑːɪlə pəɦɪlɑ̃ː vɪː məud͡ʒuːɖ̪ə ɦəɪ ət̪eː t̪uɦɑːɲũː pʰɑːɪlə ɖ̪eː uppərə lɪkʰə ləiː əɖ̪ʰɪkɑːrə ɲəɦĩː ɦəɲə.
Maithili	ओडेसा एकटा बाढ़िया खेल अछि।	oːɖeːʂɑː eːkət̪ɑː bəɽʰɪjɑː kʰeːlə ət͡ʃʰɪ.oːɖeːʂɑː eːkəɭɑː bɔɽʰɪjɑː kʰɛːlə ət͡ʃʰɪ.
Urdu	اسے نرمی سے سمجھاؤ کہ شاید وہ سمجھ لے یا ڈر جائے	ɑːʂeː نرme: ʂe: سمd͡ʒʰɑː_oː کہّک ʃəeːد oːسمجھ ہ leː jɑː ڈر d͡ʒɑː_jeː

Figure 6.1: Some examples of IPA.

[1]https://www.internationalphoneticassociation.org/content/full-ipa-chart

Our proposed architecture improves the GAN model by modifying the original attention in the generator with deep RL-guided attention, increasing the likelihood of fooling the discriminator model. The use of joint information from textual and phonological representations to train the optimized GAN model aids in learning additional phonetic context.

The contributions of this chapter are summarized as:

1. To best of our knowledge, we are the first to use the JE of orthographic sub-word and phonetic sub-word information instead of traditional word and sub-word embedding and optimize the GAN-NMT using the deep RL guided attention network for LRLs.
2. Our model generates the hypothesis that contains phonological and textual information and selects the best of them for final predicted sequences.
3. We also retrofit our proposed model in multilingual and unsupervised settings and achieve improvement of >2 BLEU points over existing multilingual and unsupervised baselines.

6.2 System Model and Problem Discussion

Data scarcity in LRLs restricts the NMT models to perform better by loosing context due to insufficient information about sentences. One of the solutions to handle such missing context problem is to optimize the original attention generated by model as discuss in the following sections.

6.2.1 System model

In order to optimize the attention, we retrofit the DRGA in GAN-NMT. DRGA is made up of three modules: Transformer network, policy network, and adjusting module as discuss below.

6.2.1.1 Transformer network

In transformer network, NMT grabs an input sequence $Z = \{z_1,z_2,...,z_n\}$ (where z_i is a word) in the form of orthographic subword embeddings $\mathbf{sw_i}$ $(1 \leq i \leq n)$ and generates attention $\mathbf{attn_i}$ $(1 \leq i \leq n)$ for each $\mathbf{sw_i}$ using following [119]:

$$\mathbf{attn_i} = softmax\left(\frac{\mathbf{Q_i}\mathbf{K_i}^T}{\sqrt{d_k}}\right)\mathbf{V_i}, \tag{6.1}$$

where $\mathbf{Q_i}$, $\mathbf{K_i}$, $\mathbf{V_i}$ and d_k are query, key, value and the dimension of key, respectively. Attention $\mathbf{attn_i}$ is passed through layer-normalization steps to feed-forward layer. The output of the top encoder is then transformed into a set of attention vectors. Finally, this set of attention vectors is given as input to the encoder-decoder attention layer for loss calculation and sequence prediction.

6.2.1.2 Policy network

We design the *RL* agent as policy network $\pi_\theta(s,a) = P(a|s;\theta)$, where s represents the state, a stands for action and θ denotes the model's parameters. Policy network is trained based on the policy gradient algorithm [117]. We define the *state* s_t at each time step $t \in \{1,...,n\}$ as:

$$s_t = [\mathbf{sw_t}; \mathbf{Q_t}; \mathbf{K_t}; \mathbf{V_t}], \tag{6.2}$$

where $\mathbf{sw_t} \in \mathbb{R}^{D_{sw}}$ is an orthographic subword vector, $\mathbf{Q_t} \in \mathbb{R}^{D_Q}$ is query vector, $\mathbf{K_t}$ $\in \mathbb{R}^{D_K}$ is key vector and $\mathbf{V_t} \in \mathbb{R}^{D_V}$ is value vector at time step t. The orthographic subword vector $\mathbf{sw_t}$ is retrieved from a lookup table, and $\mathbf{Q_t}$, $\mathbf{K_t}$ and $\mathbf{V_t}$ are computed from transformer network. As a result, for every sequence $\mathbf{Z} = \{\mathbf{sw_1},..., \mathbf{sw_t},..., \mathbf{sw_n}\}$, there is an associated state sequence $\mathbf{S} = \{\mathbf{s_1},..., \mathbf{s_t},..., \mathbf{s_n}\}$ which is pumped into the agent and the agent generates the appropriate action $\mathbf{a_t}$.

6.2.1.3 Adjusting module

It consists of three parts– Action, Reward, Transition and Observation, as described below:

- **Action:** In this part, we sample the corresponding action sequence $\mathbf{A} = \{\mathbf{a_1},..., \mathbf{a_t},..., \mathbf{a_n}\}$ from a given state sequence $\mathbf{S}$ using Gaussian distribution as given in the following (see Fig. 6.3) [57]:

$$\begin{aligned} \pi(s_t;\theta) &= \mathcal{N}(\mu(s_t,\theta), \sigma(s_t,\theta)), \\ \mathbf{a_t} &\sim \pi(s_t;\theta), \end{aligned} \tag{6.3}$$

  where μ and σ are the mean and standard deviation, respectively.

- **Transition:** When the actions for all the input sequences are sampled, the adjusting module raises or lowers the corresponding attention score based on the action values. Given an original attention vector $\mathbf{attn} = \{\mathbf{attn_1},...,\mathbf{attn_t},...,\mathbf{attn_n}\}$ and the actions $\mathbf{A} = \{\mathbf{a_1},...,\mathbf{a_t},...,\mathbf{a_n}\}$, adjustment for new weight score $\mathbf{attn_t^r}$ at each time step $t \in \{1,...,n\}$ is computed as:

$$\mathbf{attn_t^r} = \mathbf{attn_t} + \mathbf{a_t}\beta, \tag{6.4}$$

  where β is trainable parameter lies between 0 and 1.

  Once we obtain the attention values, we normalize them by putting in the softmax function to ensure their sum is equal to one.

$$\mathbf{attn'_t} = softmax(\mathbf{attn_t^r}). \tag{6.5}$$

  Finally, the modified attention scores $\mathbf{attn'} = \{\mathbf{attn'_1},...,\mathbf{attn'_n}\}$ are passed to the feed-forward layer.

- **Observation:** Once the transformer generates the prediction based on $\mathbf{attn'}$,

the posterior output probability $P(y|\mathbf{Z})$ is calculated. Policy network updates its parameters only after actions of all the states in a sequence are generated.

- **Reward:** In order to optimize the policy, we compute the reward using posterior output probability obtained from revised attention in transformer network. Based on [57], we add an additional term to regulate the number of highly weighted elements to encourage the agent to make proper adjustments. We add the unimodal function ($f(x) = x + T_o/x$) in the reward to give more weights to salient words in the sentence rather than stop or unsalient words. Mathematically, reward R_T is computed as follows:

$$R_T = \log P(y|\mathbf{Z}) - \gamma(T/M + MT_0/T), \tag{6.6}$$

where T represents the length of the sequence, M states the highly weighted elements, γ is a harmonic factor to regulate the reward and the unimodal function [120] $f(x) = x + T_o/x$ have a minimum value at $x_o = \sqrt{T_o}$ (T_o should be decided according to the development data set) and encourages M to be $\sqrt{T_o}T$.

6.2.2 Problem statement

In GAN-NMT, the pretrained NMT model is used as a generator that generates translation from source sentences to fool the discriminator. For LRLs, using traditional NMT systems as generator degrades the translation quality due to a shortage of training data and increases the data sparseness that leads to missing the context of sentences. To overcome this problem, there is a need of optimized generator that can enrich the sentences with context. To achieve the above stated objective, we use JE that contains textual and phonetic information of data in the NMT model and train the model based on the DRGA framework as discussed in the following section.

6.3 Proposed Approach

In this section, we discuss our proposed model in order to handle the resource problem faced by LRL-NMT. First, our model generates JE as shown in Fig. 6.2. Then model takes this JE as input vector to train the generator and outputs the two types of sentences: textual and phonetic as shown in Fig. 6.3. Finally, the model jointly trains the generator and discriminator based on architecture as shown in Fig. 6.4.

6.3.1 Joint embedding

Projecting embedding with orthographic subword units is one of the promising approaches to represent document vocabulary. However, orthographic subword encoding technique leverages linguistic and textual features of words, ignoring the phonetic knowledge that degrades model performance. Therefore, we introduce a phonological-based embedding that favours the model in learning the phonetic features and improves the context vector. To give phonetic representation to words, we use the IPA notation. IPA is an alphabetic system of phonetic notation based on Latin script. We use the Byte Pair Encoding (BPE) technique [121] to segment the phonetic representation of words and called these segmented token as phonetic sub-word tokens. Then we compute JE as shown in Fig. 6.2. For JE, $\mathbf{u_i}$, $(1 \leq i \leq n)$, we represent a word as z and its phonetic form as o and compute as follows:

$$\mathbf{u_i} = concat(\mathbf{sw_i}, \mathbf{so_i}), \tag{6.7}$$

where $\mathbf{sw_i}$ is orthographic subword vector of word z, and $\mathbf{so_i}$ is the phonetic sub-word vector of token o.

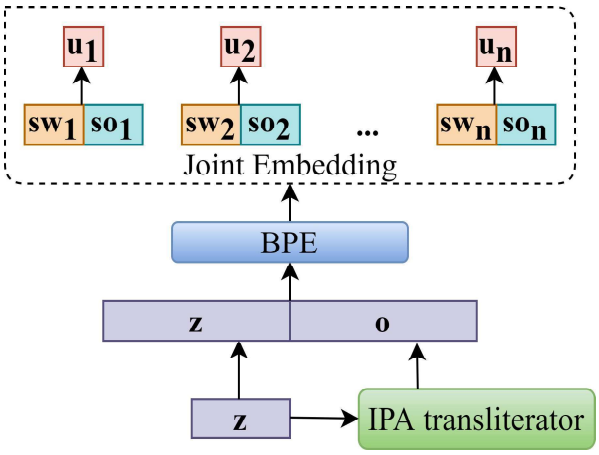

Figure 6.2: Joint embedding.

6.3.2 Generator

Fig. 6.3 contains the optimized architecture of our generator model. The generator model (GEN) is modified form of transformer-NMT. Firstly, we give bilingual corpus of language pairs as input to the model. Bilingual corpus contains sentence-wise parallel corpus of source and target language pairs. Model takes the JE as input and trains the GEN based on DRGA framework explained in Section 6.2.1. Then, for decoding the target words from the generated vectors, model uses Eqs. (6.8) and (6.9) after the final linear layer of decoder. After decoding the generated vector, we get two types of hypothesis vectors, textual (**sw**) and phonetic (**so**). Then diverse beam search is applied on them to get the best predicted sentences for both vectors. Finally both types of predicted sentences have to go through the best hypothesis selection method for better prediction.

6.3.2.1 NMT decoding

JE computed for the target-side inputs given to decoder for loss calculation between target-side input ($\mathbf{u}_t$) and the generated logits vector ($\mathbf{l}_t$) at time step t. This logits vector contain the joint information of each orthographic subword and phonetic sub-word vector. To decode each orthographic subword and phonetic sub-word vector,

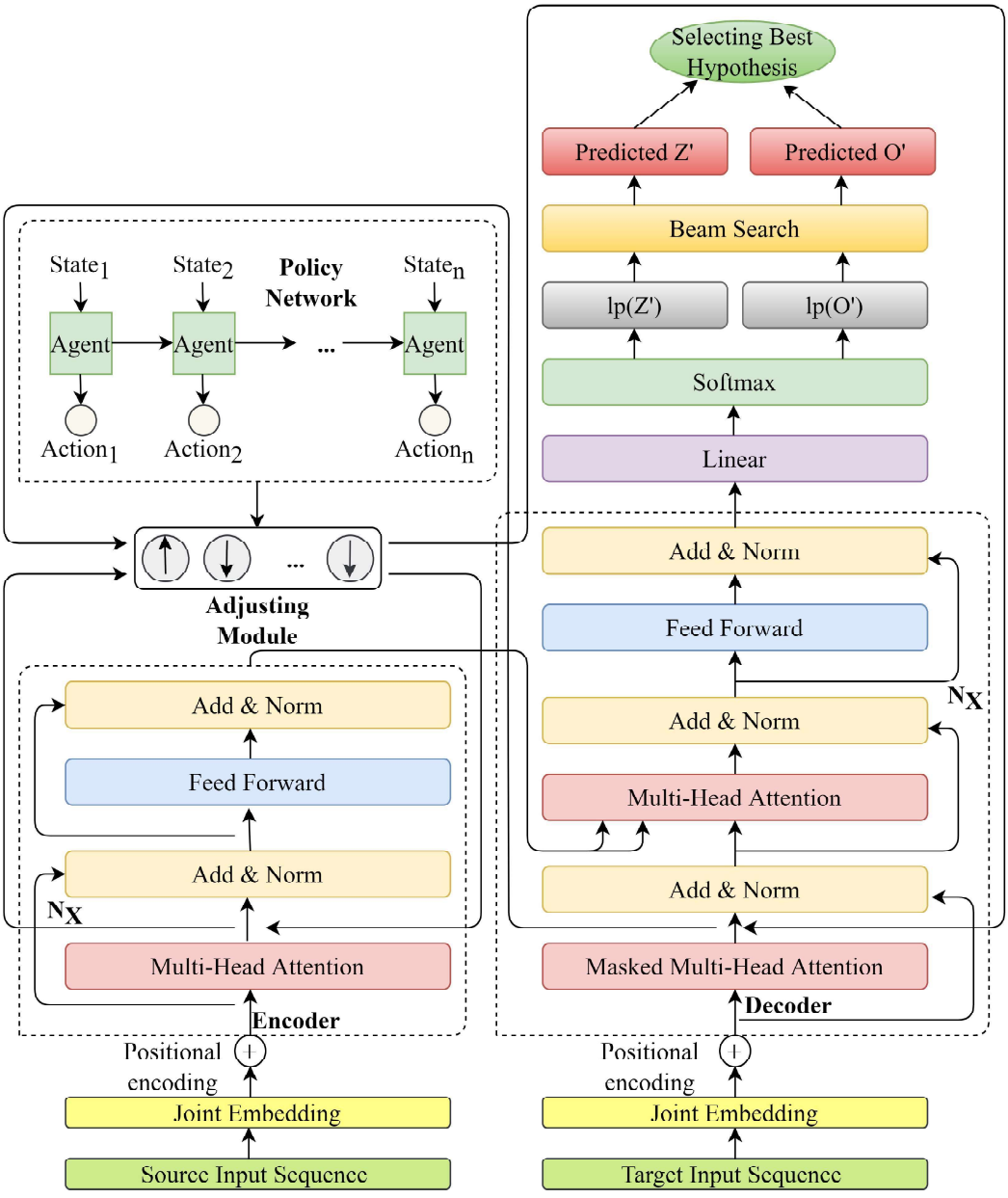

Figure 6.3: Optimized generator model in GAN-NMT.

we split the $\mathbf{l_t}$ into two vectors ($\mathbf{sw'_t}$ and $\mathbf{so'_t}$). Model uses Eqs. (6.8) and (6.9) for orthographic subword and phonetic sub-word probabilities. Then diverse beam search is applied to get the best possible hypothesis for both orthographic subword and phonetic sub-word vectors.

$$p(\mathbf{sw'_{t_l}}) = \frac{exp(\mathbf{sw'_{t_l}})}{\sum_{j=1}^{M_{sw}} exp(\mathbf{sw'_{tj}})}, \tag{6.8}$$

$$p(\mathbf{so'_{t_l}}) = \frac{exp(\mathbf{so'_{t_l}})}{\sum_{j=1}^{M_{so}} exp(\mathbf{so'_{tj}})}, \tag{6.9}$$

where M_{sw} and M_{so} are the number of generated orthographic subword and phonetic sub-word vectors, respectively.

6.3.2.2 Best hypothesis selection

As shown in Fig. 6.3, we get two types of hypothesis: predicted target sequences contain textual knowledge (Z') and predicted target sequences contain phonetic knowledge (O'). Then model performs sentence-wise BLEU comparison between the two predicted hypotheses for each predicted sequence pairs. Predicted sequences with better BLEU is selected by the model for final sequence generation (Y').

$$Y' = (1 - \lambda)Z' + \lambda O', \tag{6.10}$$

where λ is parameter. When $\lambda = 0$, model gives more weightage to Z' and when $\lambda = 1$, O' gains more weightage.

6.3.3 Generator training procedure

We train the GEN following the Algorithm 6.1. First, pretrain the modified transformer network until the model achieves its optimal performance. Next, pretrain the policy network by keeping the parameters of the attention model fixed. Finally, jointly train the whole network until its convergences. In joint training, first model calls each source

and target sequences (Z and Y) and computes the original attention weights for both Z and Y. Then, model generates the actions a_t corresponding to each state s_t for both sequences. Further, model modifies the original attention weights via Eqs. (6.4) and (6.5) and performs its training using loss function in decoder via Eq. (6.11) and make predictions. Finally, model computes reward for each sequences via Eq. (6.6) and updates the parameters of network and RL agent. A cross-entropy loss function is used to train the generator as given in the following:

$$\mathcal{L} = -\sum_{\mathbf{Z}} \sum_{1}^{C} \hat{p}(y|\mathbf{Z}) \log P(y|\mathbf{Z}), \tag{6.11}$$

where $\hat{p}(y|\mathbf{Z})$ is the gold distribution of $\mathbf{Z}$ and C is number of class labels.

Algorithm 6.1: Generator training procedure

Input: $Bilingual(Z, Y)$,$\forall Z$ $\in$Source language, $\forall Y$ $\in$Target language
Output: Optimized encoder-decoder attention model

Pretrain the encoder-decoder attention network
Fix the parameters of the attention network and pretrain the policy network
for $\forall Z_i,\ \forall Y_i \in Bilingual(Z, Y)$ **do**
    Feed Z_i and Y_i into Encoder and Decoder respectively;
    Compute original attention weights for both Z_i and Y_i via Eq. (6.1);
    for $\forall t \in Z_i$ **do**
        Sample the action a_t for states s_t via Eqs. (6.2) and (6.3);
    Adjust original attention values by adjusting module for Z_i via Eqs. (6.4) and (6.5);
    for $\forall t \in Y_i$ **do**
        Sample the action a_t for states s_t via Eqs. (6.2) and (6.3);
    Adjust original attention values by adjusting module for Y_i via Eqs. (6.4) and (6.5);
    Make prediction in decoder, and minimize loss via Eq. (6.11);
    Compute reward for RL agent via Eq. (6.6);
    Update parameters for attention network and RL agent;

6.3.4 Discriminator

Discriminator (DIS) is classification model based on CNN to discriminate between real sentences and GEN-translated sentences (see Fig. 6.4). It is a CNN architecture starting with two convolution layers, where each one has 20 filters of size 3×3 and each one is followed by a max-pooling layer with filter size 2×2. After the convolution and

pooling layers, one multi-layer perceptron layer is added with 20 hidden nodes, and eventually sigmoid activation is used to give the probability. The dimension of word embedding used in *DIS* is the same as that for the corresponding GEN.

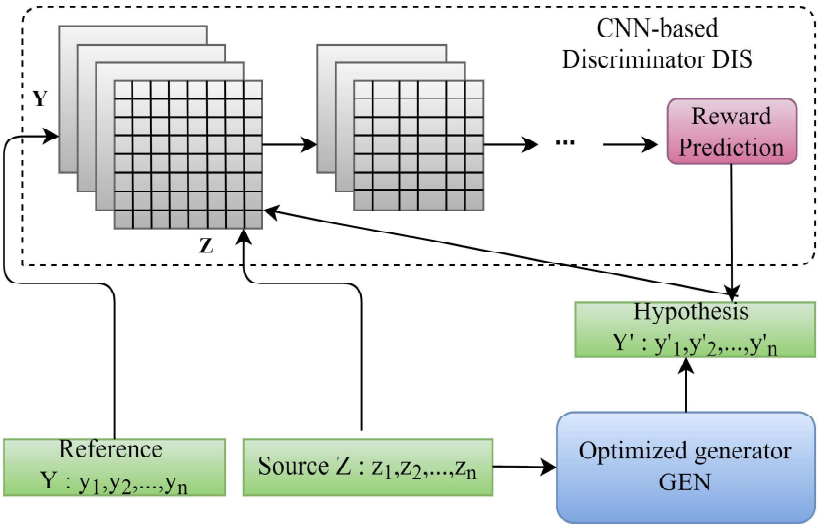

Figure 6.4: Overall architecture of GAN NMT.

6.3.5 Model training

In this section, we explain the training procedure of our proposed model. Firstly, we train the generator based on DRGA framework to achieve the best translation performance. Secondly, we generate the translated sentence using pretrained generator on greedy sample training set. Then, we train the CNN-discriminator on the combined dataset of training and machine translated sentences F. Finally, we jointly train the generator and discriminator until convergence. The elaborated procedure of training the model is shown in Fig. 6.5.

6.4 Performance Study

In this section, we discuss the corpus statistics, experimental setup and result analysis on various parameters to execute the experiments.

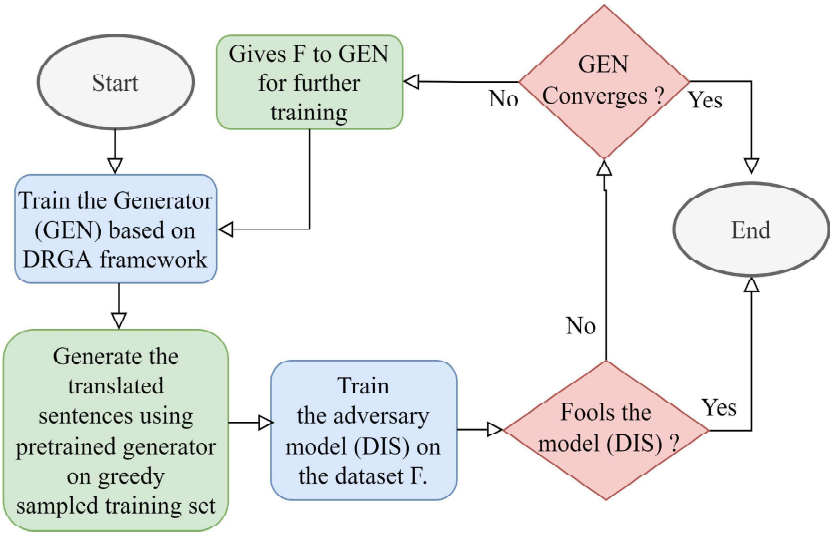

Figure 6.5: Flow chart of proposed training procedure.

6.4.1 Datasets

We conduct the experiments on GU↔HI, NE↔HI, PA↔HI, MAI↔HI, and UR↔HI language translation pairs. In GU↔HI translation task, we extract the bilingual datasets from CVIT-PIB which consists of 15K training, 1973 test and 1973 validation sentence pairs [84]. For NE↔HI translation task, training, test, and validation data of about 0.133M, 3K, 3K respectively collected from WMT19 task [81], Opus [86], and TDIL repositories [122]. For PA↔HI, MAI↔HI, and UR↔HI, we use the bilingual data from Opus [86]. For unsupervised experiments, test set of BHO↔HI and MAG↔HI are obtained from the zero-shot translation task at LoResMT2020 [101]. In PA↔HI and MAI↔HI translation task, Opus contains lot of noise in data. For better performance of model, we manually filter the data and remove the unwanted noise. After filtering, we use 0.2M and 93K training sentences for PA↔HI and MAI↔HI, respectively. Testing and validation are done on 7K sentences for PA↔HI and 3K sentences for MAI↔HI translation tasks. For UR↔HI, we train the model on 0.1M sentences. Testing and validation are done on 3K sentences.

6.4.2 Settings

This section covers the different framework and hyper-parameter used to train the NMT models.

6.4.2.1 Proposed approach

We implement GAN model based on the DRGA, transformer-NMT and CNN architecture. We execute the experiments on Paramshivay supercomputer[2] having configuration *Nvidia V100 GPU*. The decoder and encoder layers have been set to 5. The encoder and decoder embedding dimensions in the feed-forward network are set to 2048. The embedding dimensions of the encoder and decoder are 512. The attention heads for the decoder and encoder are set to 2. The models are regularized with dropout, label smoothing and weight decay, with the corresponding hyper-parameters being set to 0.4, 0.2 and 0.0001, respectively. Models are optimized with Adam using learning rate of 0.0004 and have patience value 10. The beam size for decoding is 5. Each NMT model is trained for 100 epochs. For the attention regulating factor β and harmonic factor γ, we used the values between 0 and 1, out of which we get better performance when $\beta = 0.4$ and $\gamma = 0.3$. We terminate the training of GEN and GAN when the performance is not improved for continuous 10 iterations on the validation set. We also stop training the GAN model when the discriminator fails to discriminate between real and fake sentences.

6.4.2.2 Baseline approaches

To demonstrate the effectiveness of our approach, we compare it with the following baseline models:

- **Transformer-NMT:** We apply the same settings given in [58] for Transformer-NMT.

[2]https://www.iitbhu.ac.in/cf/scc

- **Transformer-NMT+Phonetic Embedding (PE):** We apply the same settings as used in Transformer-NMT. However, difference is that the model takes PE as input rather than a word embedding.
- **Transformer-NMT + GAN:** We use the Transformer-NMT as the generator and CNN as the discriminator to train the model on orthographic subword embedding.
- **Transformer-NMT+DRGA:** We also compare our proposed approach with the framework used in [57] for the transformer-NMT. We follow the same settings of the model used in [57] and train it on our language pairs.
- **Unsupervised NMT:** We use the [66] to compare our approach under zero-shot condition on language pairs: BHO↔HI and MAG↔HI.

Table 6.1: Results on different methods ("→": X→HI and "←": HI→X)

Model	X=GU		X=NE		X=PA		X=MAI		X=UR	
	→	←	→	←	→	←	→	←	→	←
Transformer-NMT [58]	22.8	15.7	30.5	32.8	78.5	80.0	79.4	86.5	28.7	22.7
Transformer-NMT + PE	23.1	15.9	31.8	32.6	79.1	80.1	79.5	86.8	28.9	22.5
Transformer-NMT + GAN	24.2	16.1	31.1	33.6	79.8	81.4	80.4	86.5	29.1	23.4
Transformer-NMT + DRGA [57]	24.5	17.2	31.6	33.4	78.8	81.6	80.1	88.1	30.1	23.9
Transformer-NMT + JE (Proposed)	27.6	20.7	35.3	35.7	79.7	81.6	80.1	88.2	30.1	24.6
Transformer-NMT + GAN + DRGA +JE (Proposed)	29.5	22.6	36.5	36.9	80.9	82.5	81.5	89.6	32.3	26.7

Table 6.2: Results on different methods under multilingual settings ("→": X→HI and "←": HI→X)

Model	X=GU		X=NE		X=PA		X=MAI		X=UR	
	→	←	→	←	→	←	→	←	→	←
Transformer-NMT [58]	20.8	21.8	28.5	30.1	57.1	59.3	64.3	66.2	13.4	12.5
Transformer-NMT + PE	21.1	20.9	28.6	30.4	57.4	57.1	64.1	65.6	13.5	12.7
Transformer-NMT + GAN	21.3	22.3	28.4	30.6	57.4	59.7	65.2	66.9	13.8	13.1
Transformer-NMT + DRGA [57]	22.6	22.9	29.1	31.8	58.3	61.2	65.9	68.1	14.2	14.5
Transformer-NMT + JE (Proposed)	26.0	23.9	30.8	32.1	59.5	60.1	67.9	69.4	15.3	13.6
Transformer-NMT + GAN + DRGA +JE (Proposed)	27.8	24.8	32.1	34.2	61.5	62.5	69.6	71.8	17.1	15.8

6.4.3 Results and analysis

In the following section, we describe the results and analyze the proposed approach on bilingual NMT, multilingual settings, and unsupervised settings.

6.4.3.1 Effects on bilingual NMT

Table 6.1 presents the translation results on bilingual NMT. Our model outperforms all the baseline approaches. Our model gains an improvement of > 6, > 6, > 2, > 2 and > 2 BLEU points on GU↔HI, NE↔HI, PA↔HI, MAI↔HI and UR↔HI, respectively, compared to Transformer-NMT and also surpasses the DRGA result with > 2 BLEU points. The reason behind improvement is the baseline models have not considered any phonetic-related features. Our model incorporates both phonetic sub-word and orthographic subword embedding in GAN-NMT and achieves the best performance among all the baselines. Another reason for improvement is the use of optimized attention in GAN-NMT that leads to learning of better context vectors.

6.4.3.2 Effects of JE

We demonstrate the experiments on JE with optimized GAN and without optimized GAN as shown in Tables 6.1 and 6.2. On transformer-NMT, JE without GAN also gives good results and outperforms many baseline approaches. This shows that use of extra phonetic context helps improving the models performance. So, we notice that phonetic information along with textual knowledge contributes largely in models' performance.

6.4.3.3 Effects on multilingual settings

We also conduct the translation experiments under multilingual settings to measure the performance of our model. Table 6.2 contains the results under multilingual settings. We conduct the multilingual experiments on GU↔HI, NE↔HI, PA↔HI, MAI↔HI and UR↔HI language pairs. We get an improvement of (> 2 BLEU) points over the all baseline approaches. Reason for improvement is the context learnt by our model consider both phonetic sub-word and orthographic subword level of information. Generating the final hypothesis based on sentence-wise BLEU comparison of both textual and phonetic hypothesis leads to further improvement.

6.4.3.4 Effect on unsupervised settings

Table 6.3 lists the result of experiments performed under unsupervised settings. For evaluating the model on unsupervised condition, we demonstrate the experiments on BHO↔HI and MAG↔HI zero-shot language pairs. We use the NE↔HI pretrained models to evaluate the results on zero-shot language pairs. Reason for opting these models is closely relatedness between NE, BHO, MAG and HI languages. Under unsupervised conditions, proposed models also achieve an improvement of (> 2 BLEU) points.

Table 6.3: Unsupervised results on X→HI and HI→X

Model	X=BHO		X=MAG	
	→	←	→	←
a [66]	19.5	2.5	13.7	3.1
a + b	19.7	1.8	13.4	2.9
a + c	19.7	2.7	14.1	3.1
a + d	20.5	2.9	14.9	3.7
a + e (Proposed)	20.4	3.1	15.6	4.7
a+c+d+e (Proposed)	22.3	4.5	17.1	5.9

Note– a: Unsupervised NMT, b: PE, c: GAN, d: DRGA, e: JE, "→": X→HI, "←": HI→X.

6.4.3.5 Morphological analysis

For morphological analysis, we measure our model on chrF2 metric as shown in Fig. 6.6. Since chrF2 is relied on character-based evaluation of F-measure and this character-based measure help in analysing whether the model preserve the morphological and syntactic characteristics of languages. More chrF2 corresponds to preserving better morphological and syntactic features of languages. Our experimental results demonstrate that our approach accomplishes good improvement of (> 4% chrF2) on all language pairs.

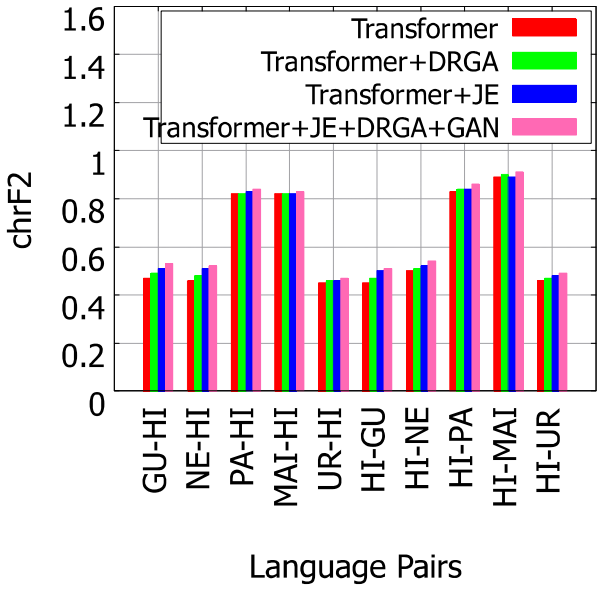

Figure 6.6: chrF2 scores.

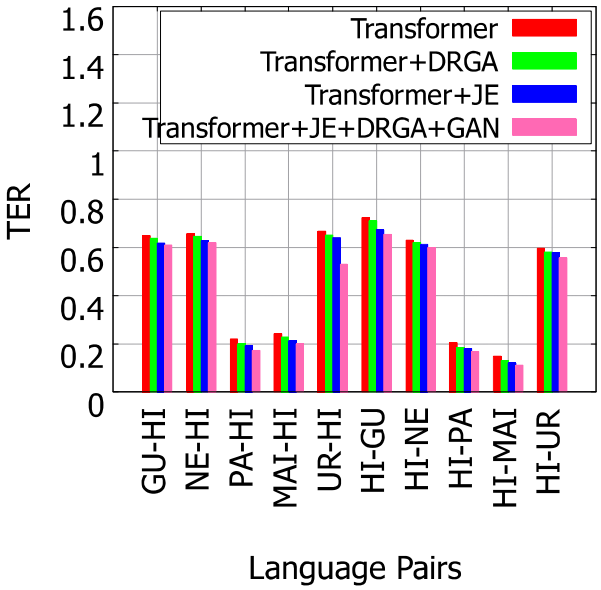

Figure 6.7: TER scores.

6.4.3.6 Error analysis

We also performed error analysis with the help of TER score shown in Fig. 6.7. TER is an error metric for machine translation that measures the number of edits required to change a system output into one of the references. Going through the Fig. 6.7, we observe that our model is effective in reducing the error compared to the existing approaches. Results show that our model required less number of edits operation on system output to match with reference sentences for every language pair. Reason behind this is that optimized GAN model better captures the attention compared to the baseline models.

6.4.3.7 Examples of systems output

We translate some sentences of different languages using different methods and present them in Fig. 6.8. On going through examples, we see that predicted sentence pairs in our model gives more semantic representation of sentences than existing baseline methods. Reason for improvement is the optimized attention used in generator model better captures the context of sentences.

6.5 Summary

In this chapter, we have proposed a novel optimized GAN-NMT model for low resource languages which use phonetic writing systems that works on the principle of DRGA framework. Our model extends the existing methods by exploiting the phonetic features along with orthographic subword features of languages. Proposed model generates the phonetic information along with textual data and selects the best one from them for final hypothesis prediction. The experiments performed on GU↔HI, NE↔HI, PA↔HI, MAI↔HI and UR↔HI translation tasks clearly demonstrate the effectiveness of our method.

Language Pair-1	Gujarati->Hindi
Source	તેમણે ઉમેર્યું હતું કે, નવા ભારત માટે નિર્માણ કરવામાં આવેલા આધુનિક ઈન્ફ્રાસ્ટ્રક્ચરની સલામતીની જવાબદારી સીઆઈએસએફના સવામત હાથમાં છે.
Transformer-NMT	उन्होंने कहा भारत की निर्मित आधुनिक ढांचे की जिम्मेदारी सीआईएसएफ की है।
Transformer-NMT+GAN	उन्होंने कहा की नए भारत की ढांचे की जिम्मेदारी सीआईएसएफ की है।
Transformer-NMT+DRGA	उन्होंने कहा की नए भारत की निर्मित आधुनिक ढांचे की जिम्मेदारी सीआईएसएफ की है।
Transformer-NMT+JE (Proposed)	उन्होंने कहा कि नए भारत के लिए बनाए गए बुनियादी ढांचे की सुरक्षा की जिम्मेदारी सीआईएसएफ के हाथों में है।
Transformer-NMT+ JE + DRGA + GAN (Proposed)	उन्होंने कहा कि नए भारत के लिए बनाए गए आधुनिक बुनियादी ढांचे की सुरक्षा की जिम्मेदारी सीआईएसएफ के सुरक्षित हाथों में है।
Reference	उन्होंने कहा कि नवीन भारत के लिए निर्मित आधुनिक अवसंरचना की सुरक्षा की जिम्मेदारी सीआईएसएफ के सुरक्षित हाथों में है।
Language Pair-2	**Nepali->Hindi**
Source	कृषि ऋणमा निरन्तर वृद्धिमाथि सरकारले आफ्नो काँध थपथपाइरहेको छ , तर यसको यथार्थ धेरै चिन्ताजनक छ ।
Transformer-NMT	कृषि ऋण में निरंतर वृद्धि का बोझ सरकार उठा रही है, लेकिन वास्तविकता चिंताजनक है।
Transformer-NMT+GAN	कृषि उधार में बढ़ोतरी पर सरकार पीठ थपथपा रही है , इसकी सच्चाई चिंता करने योग्य है ।
Transformer-NMT+DRGA	कृषि उधार में बढ़ोतरी पर सरकार थपथपा रही है , इसकी सच्चाई चिंता करने योग्य है ।
Transformer-NMT+JE (Proposed)	कृषि वृद्धि में बढ़ोतरी पर सरकार पीठ थपथपा रही है , इसकी सच्चाई चिंता करने योग्य है ।
Transformer-NMT+ JE + DRGA + GAN (Proposed)	कृषि वृद्धि में बढ़ोतरी पर सरकार पीठ थपथपा रही है , इसकी हकीकत चिंता करने योग्य है ।
Reference	कृषि कर्ज में लगातार वृद्धि पर सरकार अपनी पीठ थपथपा रही है , लेकिन इसकी हकीकत बेहद चिंताजनक है ।
Language Pair-3	**Punjabi->Hindi**
Source	ਇਸ ਨਾਲ ਲਗਾਤਾਰ ਪੁੱਛਣ ਨੂੰ ਰੋਕਿਆ ਜਾਂ ਚਾਲੂ ਕੀਤਾ ਜਾ ਸਕਦਾ ਹੈ, ਜਦੋਂ ਤੁਸੀਂ ਮੋਲਿੰਗ ਲਿਸਟ ਉੱਤੇ ਆਏ ਸੁਨੇਹਾ ਦਾ ਜਵਾਬ ਪ੍ਰਾਈਵੇਟ ਭੇਜ ਰਹੇ ਹੋਵੋ।
Transformer NMT	जब आप मोलिंग सूची में किसी संदेश का निजी तौर पर उत्तर दे रहे हों तो यह बार-बार होने वाली क्वेरी को रोक या सक्षम कर सकता है।
Transformer-NMT+GAN	यह क्वेरी को सक्षम करके आगाह है की डाक भेजे गए सन्देश का जवाब भेजने की प्रयोग कर रहे है ।
Transformer-NMT+DRGA	यह क्वेरी को सक्षम करके आगाह करता है की डाक द्वारा भेजे गए सन्देश का जवाब भेजने की प्रयोग कर रहे है ।
Transformer-NMT+JE (Proposed)	यह पुनरावृत्ति क्वेरी को निष्क्रिय/सक्षम करके आगाह करता है की डाक द्वारा भेजे गए सन्देश का जवाब भेजने की प्रयोग कर रहे है ।
Transformer-NMT+ JE + DRGA + GAN (Proposed)	यह पुनरावृत्त संवाद को निष्क्रिय/सक्षम करके चेतावनी देता है की आप डाक सूचि द्वारा भेजे गए सन्देश का निजी जवाब भेजने की कोशिश कर रहे है ।
Reference	यह पुनरावृत्त संवाद को निष्क्रिय/सक्षम करके चेतावनी देता है की आप डाक सूचि द्वारा भेजे गए सन्देश का निजी जवाब भेजने की कोशिश कर रहे है ।

Figure 6.8: Translation examples by different systems.