# Chapter 5

# Incorporating Reinforcement Learning in Domain Adaptation

## 5.1 Introduction

Domain Adaptation *(DA)* is a type of TL technique that applies the model trained on source distribution in the context of different but related target domains [79]. Since a large corpus is beneficial for NMT, increasing training data by creating a synthetic corpus is considerably good for performance enhancement. However, increasing related datasets, also known as in-domain corpora, only improves the performance. Increasing unrelated datasets, also known as out-of-domain corpora, does not enhance or degrade the NMT's translation quality, such as for TED [1] and some IWSLT [2] tasks.

DA has been well studied in NMT for many language pairs [42, 45]. Most of the existing methods focus on using DA at model-level and sentence-level approaches [42–45]. However, the existing methods such as [54] suffer from domain shift because of mismatch of training and test domain data. Also, no specific approach focuses on reward-based learning and exploiting similarity or relatedness between the languages to handle

[1] https://www.ted.com/
[2] https://iwslt.org/

| English | Hindi | Nepali | Marathi |
|---|---|---|---|
| Ram is reading a book. | राम किताब पढ़ रहा है।<br>rAma kiwAba paDZa rahA hE. | राम किताब पढ्दैछ।<br>rAma kiwAba paDxECa. | राम एक पुस्तक वाचत आहे।<br>rAma eka puswaka vAcawa Ahe. |
| Mohan goes to school. | मोहन स्कूल जाता है।<br>mohana skUla jAwA hE. | मोहन स्कूल जान्छ।<br>mohana skUla jAnCa. | मोहन शाळेत जातो।<br>mohana SAlYewa jAwo. |
| Children are playing cricket. | बच्चे क्रिकेट खेल रहे हैं ।<br>bacce kriketa Kela rahe hEM. | बच्चाहरू क्रिकेट खेलिरहेका छन्।<br>baccAharU kriketa KelirahekA Can. | मुले क्रिकेट खेळत आहेत।<br>mule kriketa KelYawa Ahewa. |
| He has gone home. | वह घर गया है।<br>vaha Gara gayA hE. | ऊ घर गयो।<br>U Gara gayo. | तो घरी गेला आहे।<br>wo Garl gelA Ahe. |
| We enjoy playing. | हमें खेलना अच्छा लगता है।<br>hameM KelanA acCA lagawA hE. | हामी खेल्न चाहन्छौं।<br>hAml Kelna cAhanCOM. | आम्हाला खेळण्यात मजा येते।<br>AmhAlA KelYaNyAwa majA yewe. |

**Figure 5.1**: Examples of sentences in different languages.

the DA problems. Therefore, there is a need to propose an approach that leverages the relatedness between the languages and handles the DA problems via reward-based learning to improve translation quality.

In this chapter to address the DA issue, we propose a REINFORCE-based Sentence Selection and Weighting *(RSSW)* method that selects the data based on the received rewards. RSSW uses REINFORCE algorithm for learning the rewards [111]. The reason behind using REINFORCE algorithm is its policy gradient-based nature. Policy Gradient methods directly try to maximise the expected return by taking small steps in the direction of the policy gradient. It tends to converge better and gives a near-optimal solution. In DA, the hypothesis is that out-of-domain corpora possess some sentences close to the in-domain data, which resolves the domain shift issues up to some extent. After training NMT on out-of-domain data via the RSSW method, we fine-tune the NMT model on in-domain corpora with Maximum Likelihood Estimation *(MLE)* and Minimum Risk Training *(MRT)*.

We demonstrate the RSSW trained NMT model on two low-resource language pairs: Hindi↔Nepali (HI↔NE) and Hindi↔Marathi (HI↔MR) to show the robustness of our approach. Fig. 5.1 contains some sentences of demonstrated language pairs along with the WX transliterated form (Latin script) of the sentences [6]. Furthermore, RSSW leverages the relatedness between source and target languages via a WX encoding script. The contributions of this chapter are summarized below:

1. We propose *RSSW* method for sentence selection and weighting of out-of-domain sentences based on REINFORCE algorithm.
2. Using the common encoding *(Latin)*-based WX notation to train the language model helps in better learning the context of sentences for different languages.
3. In addition, we also show the scalability of the proposed method on different NMT training techniques such as MLE and MRT, towards solving the DA problem.
4. Proposed method outperforms the existing state-of-the-art and baseline approaches

by ∼2 BLEU points.

## 5.2 Proposed RSSW Method

In this section, we discuss the proposed working architecture of RSSW method in NMT to solve the DA problem and improve the translation quality by selecting and weighting the sentences via Reinforcement Learning *(RL)*. RSSW consists of three modules: language model, policy network and translation model training. The language model helps in computing the initial sentence weight for each training sentence (Fig. 5.2). Then, the RL agent helps in generating the modification actions according to the environment states via the policy network, and the sentence weight modification module revises the sentence weights according to action values (Fig. 5.3). Finally, NMT training is performed on highly weighted out-of-domain (pseudo in-domain) training sentence pairs and fine-tune on original in-domain data. Detailed working of each module is described in following section.

### 5.2.1 Language model

Language Model *(LM)* is an important component of many natural language processing tasks. It is a probabilistic technique to determine the probability of a given sequence of words in a sentence. We exploit the LM to compute the initial weight score of each sentence as shown in Fig. 5.2. The weight score of each sentence represents the likelihood of occurrence of sentence in the in-domain corpus. For LM, we transliterate the source $X$ and target $Y$ sentences of in-domain corpus to WX representations and merge the source and target language pairs. In order to train the LM, we employ the transformer architecture on monolingual data of these merged language pairs. Cross-entropy loss function used to train the LM is defined as follows:

$$\mathcal{L}_{LM} = -\sum_{Z \in D} \hat{p}(Z) \log P(Z) \tag{5.1}$$

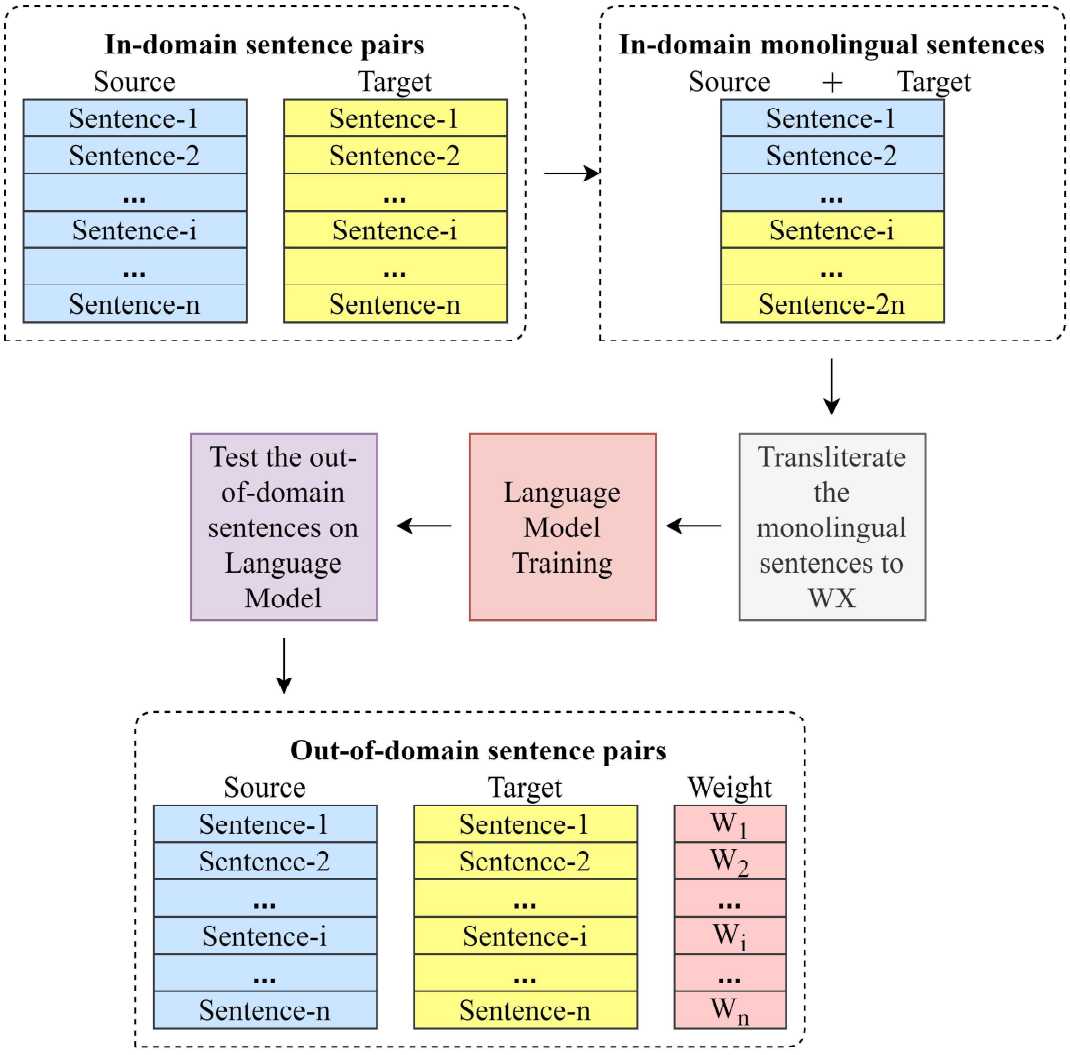



**Figure 5.2**: Weight initialization by language model.

where $Z$ is a sentence in the in-domain monolingual training corpus $D$ and $\hat{p}(Z)$ is truth label of $Z$.

The trained in-domain LM is used to compute the perplexity of out-of-domain sentences. So, the perplexity of each out-of-domain training sentence is computed as follows:

$$PP(Z') = \sqrt[n]{\frac{1}{P(w_1, w_2, ..., w_n)}} \tag{5.2}$$

where $P(w_1, w_2, ..., w_n)$ is probability of a sequence of words$\{w_1$,$w_2$,...,$w_n\}$ in a sentence $Z'$.

We use the perplexity of each sentence pair as weight to initialize the model training. So, initial weight $W_i$ of each sentence pair $S_i$ ($X_i$,$Y_i$) is computed as follows:

$$W_i = \frac{PP(X_i) + PP(Y_i)}{2} \tag{5.3}$$

where $X_i$ and $Y_i$ are the source and target sequences of sentence pair $S_i$.

This generated perplexity score of the sentence pairs is used as initial weight scores for further training of policy network, which is described in the following section.

### 5.2.2 Policy network

We represent the policy network as $\pi_\theta(c, a) = P(a|c; \theta)$, where $c$ stands for state, $a$ denotes the action and $\theta$ represents the model's parameters. We use the policy gradient algorithm to train the policy network and generate the actions based on the initial weight value from the language model.

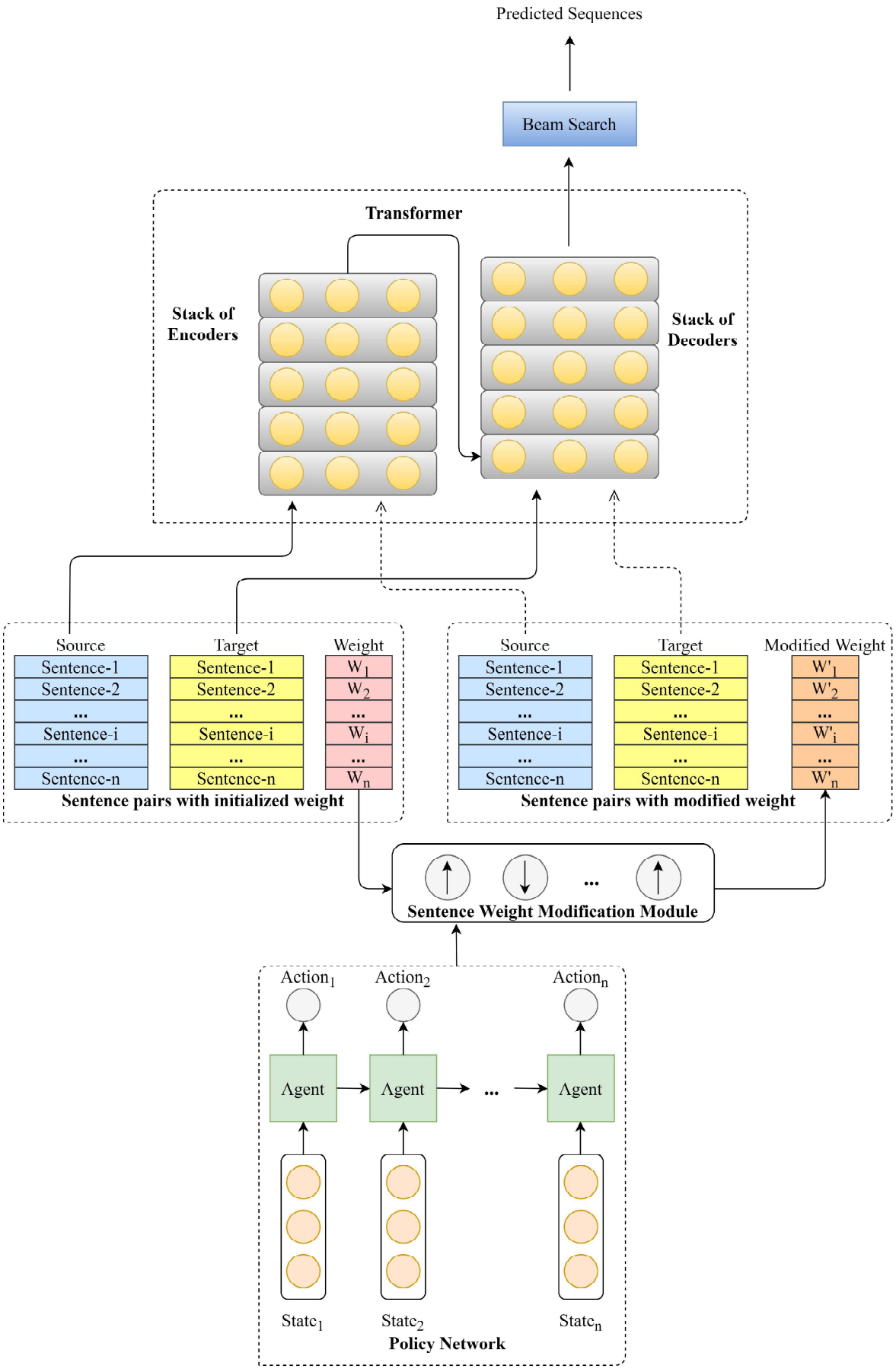



**Figure 5.3**: Architecture of RSSW.

#### 5.2.2.1 State

At each time step $t \in \{1,2,...,n\}$, a state $c_t$ consists of the sentence pair $S_t$ and sentence pair weight $W_t$. Formally, we define the state of the policy network as follows:

$$c_t = [S_t, W_t] \tag{5.4}$$

Thus, given a sequence of sentence pairs $S = \{S_1,S_2,...,S_n\}$, there will be corresponding sequence of environment states $C = \{c_1,c_2,...,c_n\}$. $C$ is passed to agent for producing the corresponding action $a_t$.

#### 5.2.2.2 Sentence weight modification

Given a sequence of states $C$, we generate the corresponding action sequence $A = \{a_1,a_2,...,a_n\}$ based on the probability density of Gaussian distribution and define the output of the agent at time step $t$ as follows:

$$a_t = \mathcal{N}(\mu(c_t, \theta), \sigma(c_t, \theta)) \tag{5.5}$$

where $\mu$ and $\sigma$ are the mean and standard deviation.

Then modified weight score is computed on the initial sentence weights $W = \{W_1,W_2,...,W_n\}$ and the generated actions $A = \{a_1,a_2,...,a_n\}$ as follows:

$$W_t^* = W_t + a_t \tag{5.6}$$

We linearly normalize the modified weights to ensure an equal-to-one sum via min-max normalization [112].

$$W_t' = \frac{W_t^* - W_{min}^*}{W_{max}^* - W_{min}^*} \tag{5.7}$$

where $W_{min}^*$ and $W_{max}^*$ are the minimum and maximum modified weights of sentence

pairs.

Finally, revised sentence weight score $W'_t$ is used to select the sentences for training the NMT model on out-of-domain data as shown in Fig. 5.3.

#### 5.2.2.3 Objective

We use training loss function of NMT model as a reward $R_L$ to guide the policy network. Reward is computed on different samples of data obtained via revised sentence weight score. We train the policy network using the Reinforce algorithm with an objective to maximize the reward.

$$\begin{aligned} J(\Theta) &= E_\pi R(c_1 a_1 ... c_n a_n) \\ &= \sum_{c_1 a_1 ... c_T a_T} \prod_t p(a_t|c_t;\theta) R_L \end{aligned} \tag{5.8}$$

where $p(a|c;\theta)$ represents the probability of a generated action $a$ and $T$ is the total number of sentence pairs.

### 5.2.3 Translation model training

In this module, we discuss the training of the NMT model. We train the NMT model on different samples of training data via transformer network. Different training samples are obtained by extracting top 70% of sentences from out-of-domain corpus via revised sentence weight score $W'_t$. We train the translation model in following ways:

#### 5.2.3.1 Multi-Domain approach *(MDA)*

In this approach, we train the NMT model on a combination of pseudo selected and original in-domain data via objective function as follows:

$$\mathcal{L}_E = - \sum_{(X,Y)\in D} W_{(X,Y)} \log P(Y|X) \tag{5.9}$$

where $D$ is the domain of training sentences, $X$ and $Y$ are source and target parallel sentences, and $W_{(X,Y)}$ is the revised weight associated with each sentence pair. The weight for original in-domain sentences are taken as 1.

#### 5.2.3.2 Fine-tuning the translation model

In NMT, the standard way of optimizing the training objective is to minimize the negative log-likelihood $\mathcal{L}_E$ of the training data $D$ as shown in Eq. (5.9). This is known as MLE in NMT. It is typically performed with teacher forcing where ground truth label data is given to decoder, which causes mismatch to inferences. To avoid the problem of mismatching, we also use RSSW with MRT apart from MLE for fine-tuning on in-domain data. The objective training function of MRT is defined as [54]:

$$\mathcal{L}_{MRT} = \sum_{(X,Y)\in D} \sum_{\tilde{Y}\in\mathcal{Y}(X)} P(\tilde{Y}|X)\Delta(\tilde{Y},Y) \tag{5.10}$$

where $\Delta(\tilde{Y},Y)$ represents the divergence between gold translation $Y$ and model prediction $\tilde{Y}$ and $\mathcal{Y}(X)$ is the posterior distribution.

### 5.2.4 End-to-end training

We list the overall training description of the proposed model in Algorithm 6.1 to understand the described module in a synchronized way. RSSW takes two types of corpus: out-of-domain and in-domain as input for training. First, RSSW merges the source $X$ and target $Y$ sentences of in-domain corpus and create a single monolingual in-domain corpus and pretrains the LM on the generated monolingual in-domain corpus via Eq. (5.1). It computes the perplexity score as the weight for each sentence of the out-of-domain corpus by testing each sentence on pretrained LM via Eq.(5.3). RSSW use the computed perplexity of each sentence for weight initialization. Then revised weights of each sentence are computed using Eqs. (5.5), (5.6) and (5.7) for $n$ number of

Gaussian iterations via policy network training. RSSW selects the top 70% of pseudo in-domain sentences. Finally, the translation module trains the NMT via MDA or fine-tune the RSSW pretrained NMT on in-domain corpus using MLE or MRT objective function.

**Algorithm 5.1:** Overall model training procedure

**Input**: in-domain($X$,$Y$) and out-of-domain($X$,$Y$),$\forall$X$\in$Source language,$\forall$Y$\in$Target language
**Output**: RSSW-based NMT model

Merge the $X$ and $Y$ sentences of in-domain($X$,$Y$) into single monolingual in-domain corpus;
Pretrain the LM on monolingual in-domain corpus via Eq. (5.1);
Test the each sentence of out-of-domain($X$,$Y$) on pretrained LM;
Initialize the weight for each sentence of out-of-domain($X$,$Y$) via perplexity score;
**for** $k$=1 *to* $n$ **do**
    Generate modified weights via Eqs. (5.5), (5.6) and (5.7);
    Select top 70% of sentences from out-of-domain($X$,$Y$);
    Train the policy network via Eq. (5.8);
    Train the translation module via MDA using Eq. (5.9) or fine-tune RSSW trained NMT model on in-domain($X$,$Y$) via MLE or MRT;

## 5.3 Experiments

This section discusses the corpus statistics, experimental settings, results, and ablation study by demonstrating the experiments.

### 5.3.1 Corpus description

We evaluate RSSW model on two low-resource language pairs (Four translation directions): Hindi↔Nepali (HI↔NE) and Hindi↔Marathi (HI↔MR). In Hindi↔Nepali translation task, we use the parallel corpora of Agriculture *(AGRI)* and Entertainment *(ENT)* domain of TDIL [113]. In Hindi↔ Marathi translation task, we use the parallel corpora of different domains from WMT 2020 shared task to train the translation models [114]. For HI↔MR, we use corpora of two domains: News *(NEWS)* and PM India *(PMINDIA)*. Table 5.1 displays statistics on the corpora of different domains.

**Table 5.1**: Corpus statistics

| Pairs | Domain | Train | | Dev | | Test | |
|---|---|---|---|---|---|---|---|
| | | Sentences | Words | Sentences | Words | Sentences | Words |
| HI↔NE | Agriculture | 12000 | 40965 | 1000 | 6973 | 1000 | 6895 |
| | Entertainment | 22000 | 77413 | 2500 | 16281 | 2500 | 13483 |
| HI↔MR | News | 10,349 | 46858 | 1000 | 10810 | 1000 | 10625 |
| | PM India | 23,897 | 78811 | 1000 | 10538 | 1000 | 10558 |

### 5.3.2 Experimental setup

Different frameworks and parameters required to train the model are as follows:

#### 5.3.2.1 Our approach

We implement the proposed method on Fairseq, a sequence modelling toolkit and execute the experiments on an *Nvidia V100 GPU* [115]. We use a transformer architecture to train the LM and NMT model. We pre-process the data with *SentencePiece* library [69]. Our model trained on the 5 number of the decoder and encoder layers with 512 embedding dimension for each and learns the joint vocabulary by sharing dictionary and embedding space. The feed-forward network has encoder and decoder embedding dimensions equal to 2048. The number of attention heads used for the decoder and encoder is 2. We use weight decay, label smoothing and dropout for regularisation, with the corresponding hyper-parameters to 0.0001, 0.2 and 0.4, respectively. We use Adam optimizer for the model having $\beta 1 = 0.9$ and $\beta 2 = 0.98$, and keep the *patience* value equal to 10. $\beta 1$ and $\beta_2$ are the hyperparameters used by Adam optimizer for controlling the exponential decay rates of moving averages of the gradient and the squared gradient [116].

#### 5.3.2.2 Baseline approaches

a In-domain NMT: In this model, we train the transformer model on in-domain training data and test it on the in-domain test set. Training and model parameters

for NMT are the same as used in our approach.

b MDA: Here, we train the transformer model on a combination of out-of-domain and in-domain training datasets and test the model on an in-domain test set. Training and model parameters for NMT are the same as used in our approach.

c Fine-Tuning Approach *(FTA)*: In this part, we first train the model on the out-of-domain training dataset and fine-tune this out-of-domain trained model on in-domain training dataset via MLE. Training and model parameters for NMT are the same as used in our approach.

#### 5.3.2.3 State-of-the-art approach

We also compared our proposed approach with [54]. For training NMT, we use *NEMATUS*; a sequence modelling toolkit and execute the experiments on an *Nvidia V100 GPU*. Training and model parameters for NMT are the same as used in [54]. Since state-of-the-art model [54] is implemented using NEMATUS, we perform our experiments with NEMATUS in place of Fairseq for better comparison.

### 5.3.3 Results and ablation study

We conduct the experiments on two language pairs of different domains and report the $BLEU$ [72, 110], $chrF2$ [93] and $TER$ [92] metrics to analyze the performance of model as discussed below.

#### 5.3.3.1 Effect of RSSW on different training techniques

Tables 5.2 and 5.3 contain the translation results on different method. Compared to the traditional approach such as in-domain NMT, MDA and FTA give better performance due to extra data. Additional training data is beneficial for NMT, but it limits the performance of NMT when it contains out-of-domain data or unrelated corpora. Our proposed method (RSSW) overcomes this limitation of MDA and FTA by selecting the

data close to in-domain data from out-of-domain data via Reinforce technique in the best possible ways. Our RSSW method gives an improvement of $\sim$ 2 BLEU points on all translations tasks in each direction compared to existing baseline approaches. The primary reason for such improvement is that our model optimizes the translation performance by selecting the pseudo in-domain data close to original in-domain data via a reward-based approach. This reward-based approach better selects the sentences by checking each possible probability of selecting the sentence from the out-of-domain data.

**Table 5.2**: Results on HI↔NE translation task

| Methods | BLEU | | | | chrF2 | | | | TER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AGRI | | ENT | | AGRI | | ENT | | AGRI | | ENT | |
| | → | ← | → | ← | → | ← | → | ← | → | ← | → | ← |
| In-domain | 48.45 | 48.93 | 29.97 | 31.43 | 0.76 | 0.73 | 0.60 | 0.59 | 0.292 | 0.319 | 0.426 | 0.536 |
| MDA | 49.51 | 49.81 | 32.31 | 33.54 | 0.76 | 0.74 | 0.61 | 0.60 | 0.281 | 0.297 | 0.421 | 0.528 |
| FTA | 49.48 | 49.95 | 31.51 | 32.21 | 0.76 | 0.74 | 0.60 | 0.61 | 0.285 | 0.296 | 0.423 | 0.524 |
| RSSW + MDA | **52.50** | **53.12** | **35.70** | **36.91** | **0.79** | **0.76** | **0.63** | **0.64** | **0.259** | **0.289** | **0.418** | **0.519** |
| RSSW + FTA | 50.9 | 51.66 | 33.68 | 34.85 | 0.78 | 0.75 | 0.62 | 0.62 | 0.274 | 0.294 | 0.422 | 0.530 |

**Table 5.3**: Results on HI↔MR translation task

| Methods | BLEU | | | | chrF2 | | | | TER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NEWS | | PMINDIA | | NEWS | | PMINDIA | | NEWS | | PMINDIA | |
| | → | ← | → | ← | → | ← | → | ← | → | ← | → | ← |
| In-domain | 5.88 | 8.64 | 13.92 | 19.59 | 0.31 | 0.30 | 0.46 | 0.47 | 0.911 | 0.834 | 0.780 | 0.689 |
| MDA | 4.56 | 9.01 | 14.01 | 18.14 | 0.31 | 0.31 | 0.46 | 0.48 | 0.912 | 0.814 | 0.745 | 0.678 |
| FTA | 6.03 | 9.56 | 15.03 | 18.91 | 0.32 | 0.31 | 0.47 | 0.48 | 0.908 | 0.816 | 0.742 | 0.674 |
| RSSW + MDA | 6.21 | 11.21 | 16.32 | **21.54** | **0.33** | 0.31 | **0.48** | **0.49** | 0.904 | **0.810** | **0.739** | **0.667** |
| RSSW + FTA | **7.91** | **11.31** | **17.85** | 20.34 | 0.32 | **0.32** | **0.48** | **0.49** | **0.901** | 0.812 | **0.739** | 0.669 |

#### 5.3.3.2 Effects on different domains

We demonstrate the experiments with two language pairs: HI↔NE and HI↔MR on different domains. Our approach gives a consistent improvement in all the domains. We test the model on an in-domain test of agriculture and entertainment for HI↔NE, and news and PMINDIA domain for HI↔MR translation task. Out-of-domain training data for each translation task are just the opposite of in-domain training and testing

data. For example, out-of-domain training data for in-domain agriculture training and testing data of HI↔NE translation task belong to the entertainment domain and vice-versa for in-domain entertainment training and testing data. Baseline approaches (such as MDA and FTA) gives further improvement compared to NMT on in-domain data. However, they did not give optimal performance. Our approach optimizes the model performance by selecting the pseudo in-domain sentences via Gaussian distribution. In Tables 5.2 and 5.3, the observation are as follows:

a) Training the model on a combination of in-domain and out-of-domain training data hampers the performance of the translation model (e.g., NEWS domain for HI→MR and PMINDIA domain for MR→Hi translation task in MDA).

b) RSSW improves the performance of the MDA model by filtering the best possible sentences close to in-domain data from out-of-domain data.

c) Fine-tuning the out-of-domain data on in-domain data also decreases the translation quality of the FTA model (e.g., PMINDIA domain of MR→HI translation task).

d) Fine-tuning pseudo in-domain corpus by selecting the sentences via RSSW method on original in-domain data improves the translation quality of the FTA model.

#### 5.3.3.3 Effects of selected pseudo in-domain data

The experiment results for changing the amount of the selected pseudo in-domain data on the agriculture development dataset of the NE↔HI translation task are shown in Fig. 5.4. We plot the graph for selected pseudo in-domain data on X-axis and BLEU score on Y-axis. We represent the selected pseudo in-domain data as a percentage of out-of-domain training data. We see that our model improves the performance up to 70 % of selected pseudo in-domain data, and after 70 %, it decreases the performance. This is the reason we extract 70 % of pseudo in-domain data for NMT training.

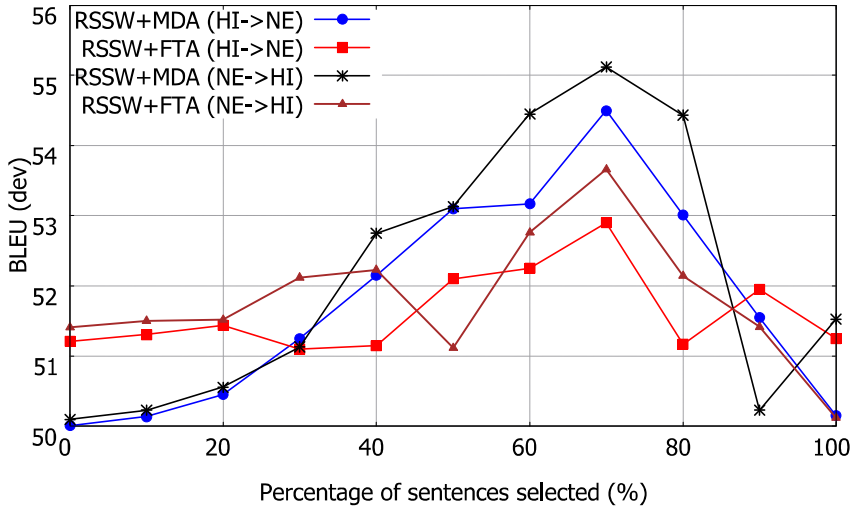
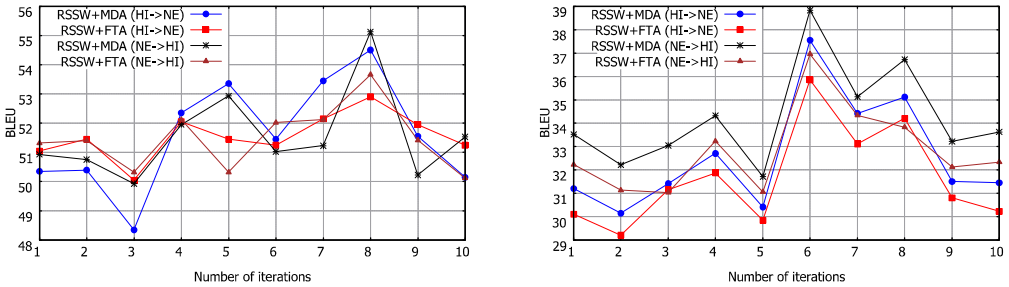



**Figure 5.4**: Changing the amount of the selected pseudo in-domain data on the HI↔NE task.

#### 5.3.3.4 Effects of Gaussian iterations

Experimental results on the development dataset of the agriculture domain with the increase in Gaussian iterations are shown in Fig 5.5. Gaussian iterations are the number of times RL agents use the Gaussian distribution to generate the modified score of sentences. In each iteration, we extract the top 70 % of sentences based on the modified score to train the model on pseudo in-domain data. We observe that our RSSW method at $8^{th}$ and $6^{th}$ iterations gives better results for agriculture and entertainment domain data, respectively, on the NE↔HI translation task.




(a) Agriculture domain. (b) Entertainment domain.

**Figure 5.5**: BLEU score versus Gaussian iteration for HI↔NE task.

#### 5.3.3.5 Comparison with MRT fine-tuning method

We also compared our approach with state-of-the-art technique of [54] as shown in Table 5.4. [54] trained the model with MLE objective function on out-of-domain data and fine-tuned this trained NMT model on in-domain data via MRT objective function instead of MLE. We perform the same operation with our RSSW method. We train the model on pseudo in-domain data selected by RSSW with MLE objective function and fine-tune it on original in-domain data via MRT. We find that RSSW outperforms the state-of-the-art technique by +2 BLEU points. RSSW helps the model to learn better in-domain context by providing pseudo in-domain data semantically close to original in-domain training data.

**Table 5.4**: Comparison with state-of-the-art approach on HI↔NE translation task

| Methods | BLEU | | | | chrF2 | | | | TER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AGRI | | ENT | | AGRI | | ENT | | AGRI | | ENT | |
| | → | ← | → | ← | → | ← | → | ← | → | ← | → | ← |
| [54] | 49.56 | 50.10 | 32.23 | 33.43 | 0.76 | 0.62 | 0.59 | 0.60 | 0.280 | 0.291 | 0.418 | 0.521 |
| RSSW + MRT | 51.90 | 51.70 | 33.21 | 34.46 | 0.76 | 0.64 | 0.59 | 0.61 | 0.277 | 0.280 | 0.415 | 0.516 |

#### 5.3.3.6 LSTM as translation model

In addition to the above analysis, we have also conducted an ablation study by replacing the components of the proposed model with other architecture. We have replaced the transformer as translation model with LSTM and observed the performance on HI↔NE translation task as shown in Table 5.5. We have observed that RSSW also shows improvement of ∼1 BLEU points using the LSTM translation model. This illustrates the robustness of our method in comparison to different NMT architectures.

## 5.4 Summary

In this chapter, we have proposed the RSSW method to handle the domain shift problem faced by NMT for low resource languages. We used LM to calculate the weight

**Table 5.5**: Results of LSTM as translation model on HI↔NE translation task

| Methods | BLEU | | | | chrF2 | | | | TER | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AGRI | | ENT | | AGRI | | ENT | | AGRI | | ENT | |
| | → | ← | → | ← | → | ← | → | ← | → | ← | → | ← |
| In-domain | 44.60 | 44.81 | 26.54 | 25.77 | 0.74 | 0.70 | 0.57 | 0.54 | 0.316 | 0.334 | 0.455 | 0.584 |
| MDA | 45.15 | 45.70 | 26.74 | 26.12 | 0.75 | 0.70 | 0.57 | 0.55 | 0.305 | 0.321 | 0.452 | 0.579 |
| FTA | 44.89 | 45.31 | 26.51 | 25.61 | 0.74 | 0.70 | 0.57 | 0.54 | 0.312 | 0.338 | 0.457 | 0.581 |
| RSSW + MDA | **46.87** | **47.01** | **27.65** | **27.87** | 0.74 | **0.71** | **0.57** | **0.55** | **0.299** | **0.319** | **0.450** | **0.571** |
| RSSW + FTA | 45.78 | 45.64 | 27.34 | 26.12 | **0.75** | **0.71** | 0.56 | 0.53 | 0.309 | 0.334 | 0.454 | 0.576 |

of each training sentence and then used this calculated weight for the initialization of reinforce-based model training. We have used the proposed method for sentence selection and weighting of out-of-domain sentences for NMT training. In addition, we have used a minimum risk training approach for fine-tuning the model along with the maximum likelihood estimation function. The proposed method has been evaluated in the Hindi↔Nepali and Hindi↔Marathi low-resource language pairs translation task. The method outperformed the existing approaches by ∼2 BLEU points for both language pair translation tasks.