Chapter 4

Transfer Learning-based Semi-supervised Pseudo-corpus Generation

4.1 Introduction

In the previous chapter, we have discussed leveraging the features of similar languages by simply converting them into an intermediate Latin-based multilingual notation, which is helpful in improving the translation quality of low-resource languages. However, the problems faced by MTs become more challenging when the availability of training data is almost none or zero. We call such kinds of issues a Zero-Resource Problem *(ZRP)* as described in Figure 4.1. Techniques to translate such zero-resource language pairs are known as Zero-Shot Translation *(ZST)*. Some examples of zero-resource language pairs are Magahi↔Hindi, Bhojpuri↔Hindi and Russian↔Hindi [101], which have no or almost no publicly available parallel corpus for training. Creating parallel corpora for such languages is time-consuming and expensive process due to manual involvement of many language experts. A great deal of work has been done on NMT for zero-resource language pairs, e.g., multilingual and pivot-based translations [15, 16, 19, 28].

Multilingual models usually generalize in a better way due to inclusion of multiple languages [102]. However, sometimes this is not valid for morphologically rich languages due to differences in morphological complexity. Pivot-based MT is also one of the traditional approaches for producing translation of zero-resource language pairs. However, training the model via pivot-based approach leads to fluency issues [103]. To address above listed problems of ZST, we propose a Transfer Learning-based Semi-supervised Pseudo-corpus Generation *(TLSPG)* approach for translation of zero-resource languages that uses semi-supervised learning to exploit similarities between low and zero-resource language pairs.

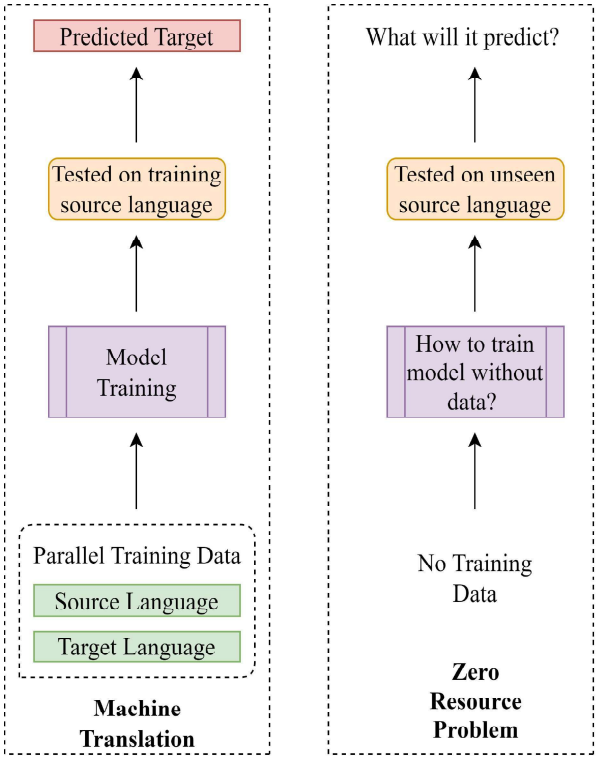

Figure 4.1: Zero Resource Problem in MT

The proposed TLSPG approach is motivated by the work of [27] built on the hybrid architecture of SMT and NMT. TLSPG generates the pseudo corpus by leveraging the relatedness between low and zero-resource language pairs and learns the context of sentences in a semi-supervised way using Transfer Learning *(TL)*. Unlike multi-

ple HRLs and LRLs in multilingual-based ZST, we use only a single LRLs parallel corpus to develop an MT system for zero resource languages. We demonstrate the experiments on Nepali (NE)↔Hindi (HI), Bhojpuri (BHO)↔Hindi (HI) and Magahi (MAG)↔Hindi (HI) language pairs. In our experiments, Nepali↔Hindi is used to generate the zero-resource language pairs (Bhojpuri↔Hindi and Magahi↔Hindi) by leveraging their relatedness via TL. All the demonstrated languages are mainly spoken in South Asian countries. Applications of ZST can be supported in different fields such as smart healthcare [104, 105], military and defence [106], finance [107] and e-commerce [108]. For instance, use of developed model can be helpful in removing the communication barrier between medical practitioners and local language speakers in a healthcare domain [109].

Based on sharing common characteristics described in Table 4.1, Bhojpuri, Magahi, Nepali and Hindi are considered related languages.

Table 4.1: Relatedness features between languages

Languages	Language Family	Script	Word Order
Bhojpuri	Indo-Aryan	Devanagari	S-O-V
Hindi	Indo-Aryan	Devanagari	S-O-V
Magahi	Indo-Aryan	Devanagari	S-O-V
Nepali	Indo-Aryan	Devanagari	S-O-V

NOTE- S: Subject, O: Object, V: Verb

Specifically, the contributions of this chapter are summarized as follows:

- Propose TLSPG approach for ZST to overcome parallel data limitation of existing NMT models.
- Unlike the existing multilingual-based ZST models [15,16,28], proposed approach leverages the relatedness of single LRLs pair as a semi-supervised TL technique to improve the performance and validate it through empirical analysis.

4.2 Transfer Learning-based Semi-supervised Pseudo-corpus Generation

This section discusses the framework of our proposed model to handle the ZRP. We propose a framework based on transfer learning and name it as Transfer Learning-based Semi-supervised Pseudo-corpus Generation *(TLSPG)* approach. It consists of three modules: Transformer-based Semi-supervised Learning *(TSL)*, Moses-based Semi-supervised Learning *(MSL)* and TL-based pseudo-corpus generation. TSL and MSL modules pretrain the model for zero-resource language pairs based on a semi-supervised learning. TL-based pseudo-corpus generation module generates the parallel aligned corpus for zero-resource language pairs via pre-trained TSL and MSL modules. Then synthetic parallel corpus is generated by merging the parallel corpus of related language with pseudo-corpus generated data and training the translation model via Transformer or Moses based translation system.

4.2.1 TSL

TSL is a semi-supervised transfer learning approach based on training the NMT model via transformer architecture. We train the transformer with five number of encoder and decoder stacks. In order to fill the gap of training language pair in ZST, TSL takes zero-resource related language pair as input to train the transformer. Before training, TSL pre-processes the training sentences via sentencepiece unsupervised tokenizer and converts the sentences into subword tokens. To train the model, generated subword tokens are added to positional encoding in the forms of subword embedding and given as input to encoder and decoder layers as shown in Figure 4.2. TSL computes the attention in Transformer as follows:

$$attn = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \tag{4.1}$$

where Q, K, V and d_k represent query, key, value and dimension of the key generated from input sequences, respectively.

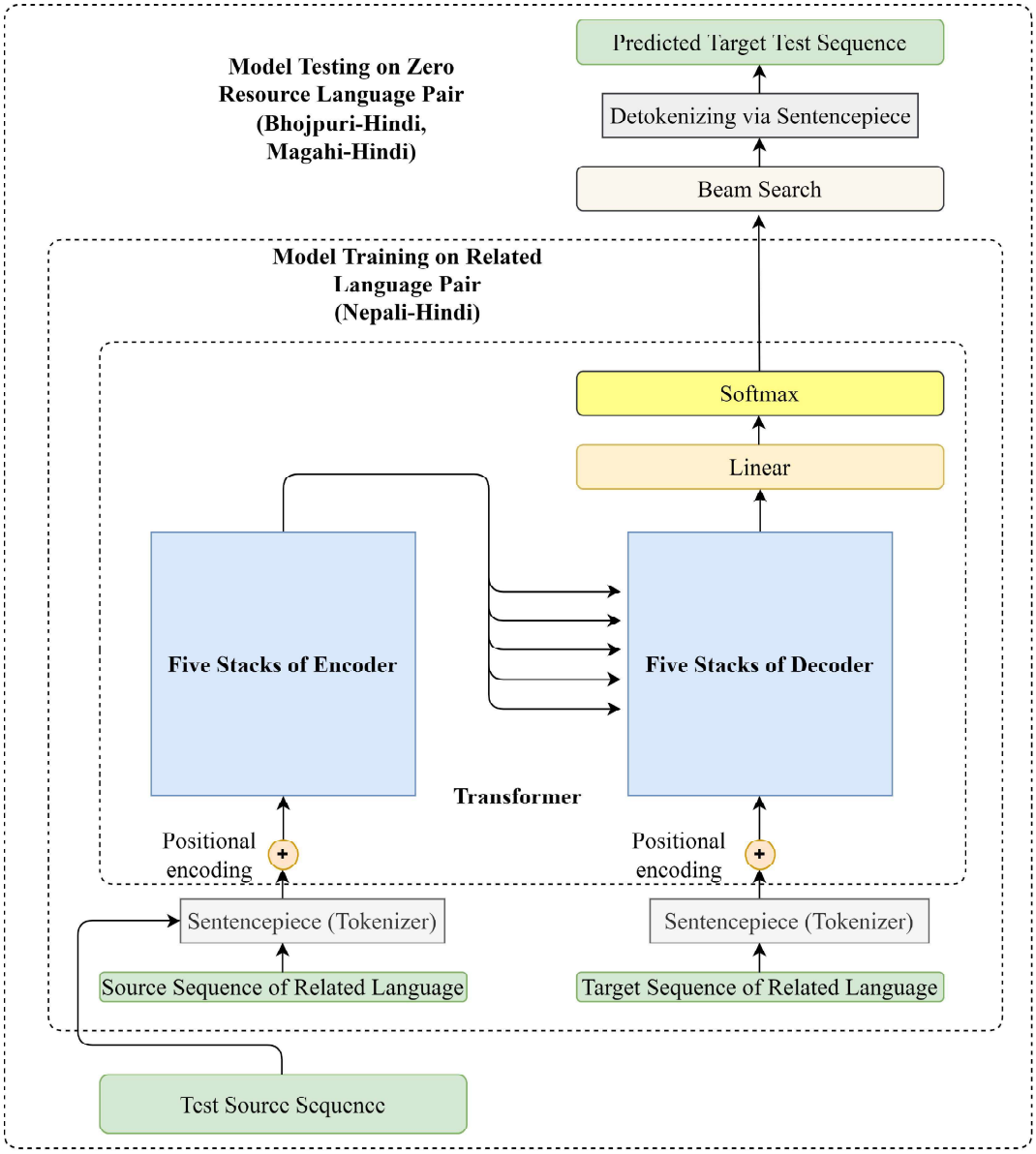

Figure 4.2: TSL for zero-resource language pair.

The cross-entropy loss function $\mathcal{L}_r$ used to train the transformer-based NMT model is defined below:

$$\mathcal{L}_r = - \sum_{(X_r, Y_r) \in D_r} \hat{p}_{(X_r, Y_r)} \log_{10} P(Y_r | X_r), \tag{4.2}$$

where X_r and Y_r are source and target sentences belonging to zero-shot related training language pairs D_r, respectively. Moreover, $\hat{p}_{(X_r,Y_r)}$ is the gold distribution of X_r. The softmax function used to convert the predicted subword embeddings into probabilities is defined as follows:

$$p(y_t) = \frac{\exp(y_t)}{\sum_{j=1}^{M} \exp(y_j)}, \tag{4.3}$$

where, M denotes the total number of unique words known by the model for the generated subword vector y_t at time step t.

The decoder decodes the predicted target probabilities and passes them to the beam search (Figure 4.2). Beam search gives the best predicted target-side subword tokens. Then detokenization is performed on predicted target-side subword tokens, and the model predicts the target sequences. For prediction, we give the zero-resource test sentence as input to the model. Finally, we get the ZST model as an output of the TSL approach.

4.2.2 MSL

MSL is a semi-supervised TL approach rely on training a phrase-based SMT system via Moses [88] framework. In order to fill the gap of training language pairs in ZST, MSL takes related language pairs as input to train Moses as shown in Figure 4.3. Before training, MSL preprocesses the training sentences via Moses tokenizer and limits the sentences upto 80 length. MSL, a Moses-based (log-linear) framework relies on two modules: translation model and language model. We use KenLM [90] to train the language model on the target side monolingual corpus of related language pairs. Translation model consists of phrase translation and distortion probabilities. For translation, MSL uses GIZA++ [89] for training on related language parallel corpus. We train overall MSL on Moses decoder. The decoder decodes the predicted target tokens. Then detokenization is performed on tokenized sequences and final target sequences are predicted. MSL computes the best target translation Y_{best} for a source input sentence

X as follows:

$$\begin{aligned} Y_{best} &= argmax_{Y} p(Y|X) \\ &= argmax_{Y} p(X|Y)\ p_{LM}(Y), \end{aligned} \tag{4.4}$$

where $p_{LM}(Y)$ and $p(X|Y)$ are language and translation models, respectively.

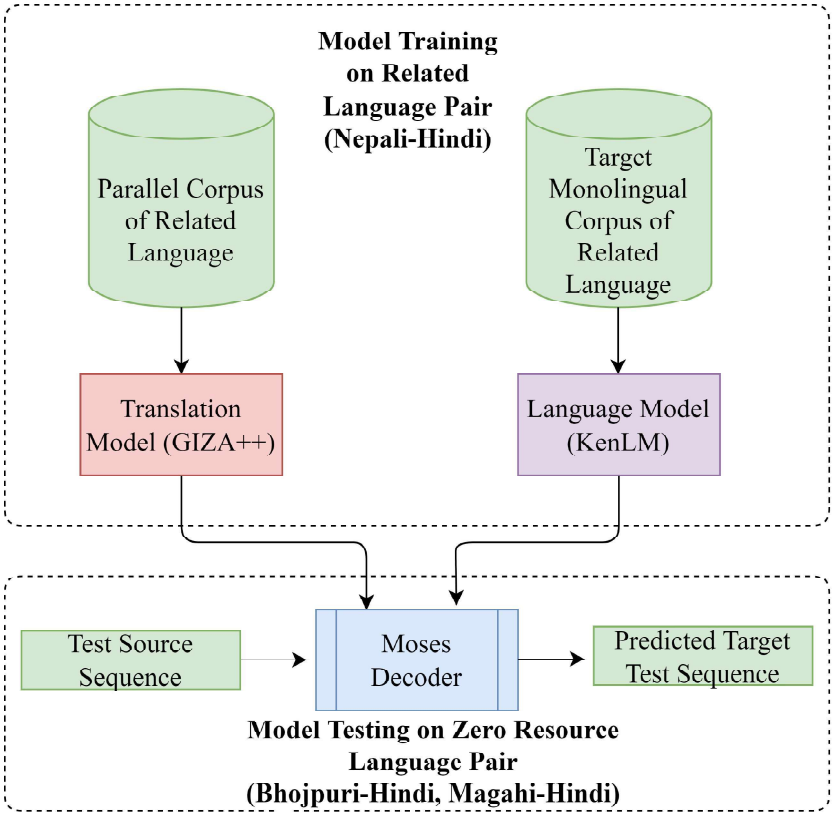

Figure 4.3: MSL for zero-resource language pair.

A phrase-based log-linear MSL model decomposes $p(X|Y)$ into $p(\bar{X}_1^I|\bar{Y}_1^I)$ as given in the following [75]:

$$p(\bar{X}_1^I|\,\bar{Y}_1^I) = \prod_{i=1}^{I} \phi(\bar{X}_i|\,\bar{Y}_i)\ d(start_i - terminate_{i-1} - 1), \tag{4.5}$$

where, I is the number of phrases $\bar{X}_i$ broken from X, ϕ is phrase translation probability, $d(.)$ is distortion probability, $start_i$ is the position of the first word of the source input

phrase that translates to the i^{th} target phrase and $terminate_i$ is the position of the last word of that source phrase.

Finally, in order to test the model, we give the zero-resource test data as input for prediction and get Moses-based ZST model as an output.

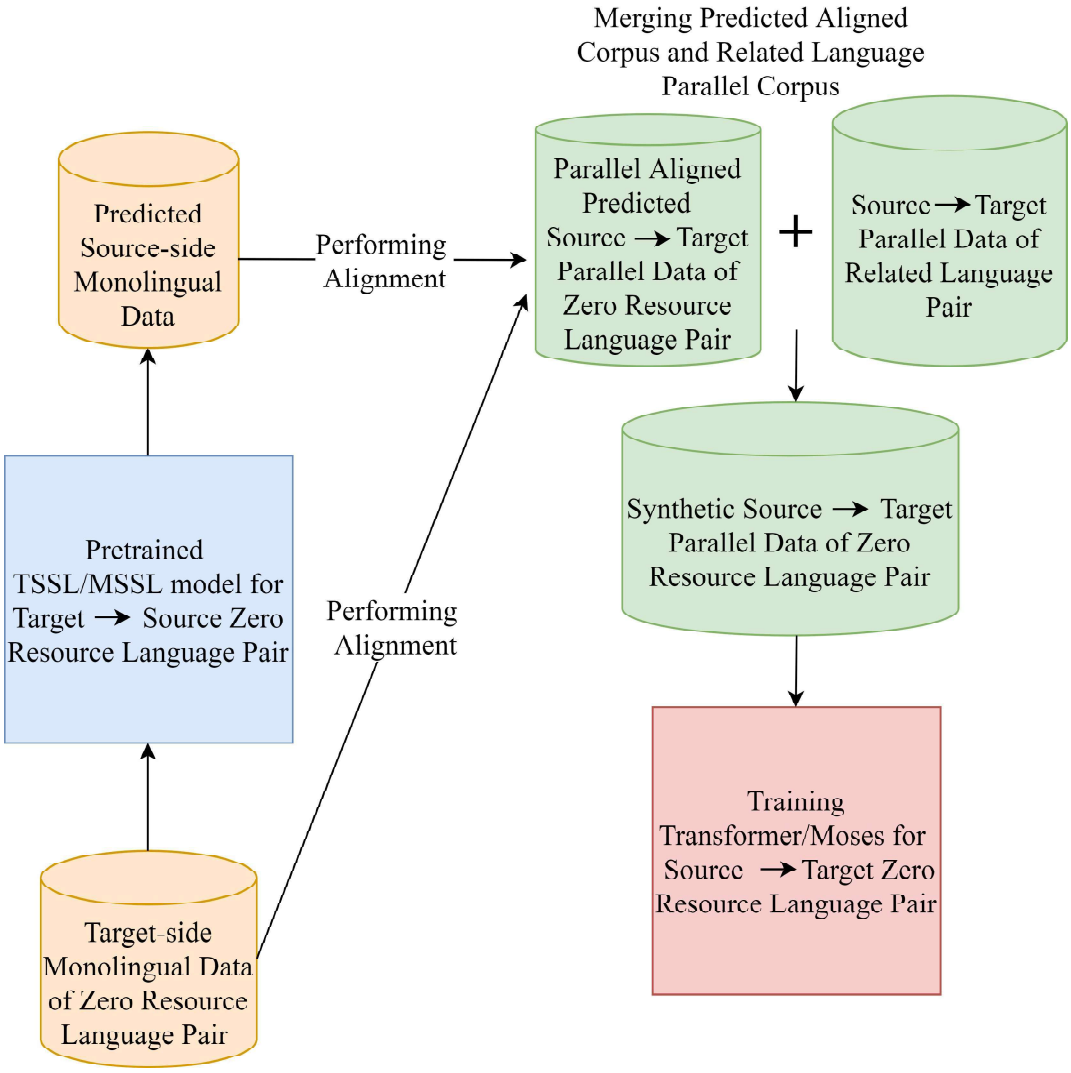

Figure 4.4: TLSPG approach.

4.2.3 TL-based pseudo-corpus generation

In this section, we discuss the pseudo-corpus generation method based on the pre-trained TSL and MSL models. Pseudo-corpus generation module with TLSPG ap-

proach is demonstrated in Figure 4.4. TLSPG first applies the pretrained TSL or MSL model on target-side monolingual data of zero-resource language pairs to generate the predicted source-side monolingual sentences. Then both the target-side monolingual sentences of zero-resource language pairs and predicted source-side monolingual sentences are aligned parallelly in the direction of Source→Target. TLSPG merges the generated aligned parallel data with Source→Target parallel corpus of related language pairs to create synthetic Source→Target parallel corpus for zero-resource language pairs.

4.2.4 Model Training

In this section, we discuss the training of the final ZST model. We train the final ZST via the Transformer and the Moses models. For Transformer, we define the cross entropy loss function to train ZST model as follows:

$$\mathcal{L}_{zst} = - \sum_{(X_{syn}, Y_{syn}) \in D_{syn}} \hat{p}_{(X_{syn}, Y_{syn})} \log_{10} P(Y_{syn} | X_{syn}), \tag{4.6}$$

where, X_{syn} and Y_{syn} represent source and target synthetic generated parallel sentences belonging to synthetic generated training corpus D_{syn}, respectively. Moreover, $\hat{p}_{(X_{syn}, Y_{syn})}$ is the gold distribution of X_{syn}.

For Moses, we use the same objective function defined in Eq. (4.4) for synthetic generated corpus. In order to get final ZST model, we train the translation model on pseudo generated corpus with following variations:

A **Transformer model training on TSL generated corpus architecture:** We train the Transformer on TSL-based generated synthetic corpus with five layers of encoder and decoder to train the Source→Target ZST model.

B **Moses training on MSL generated corpus architecture:** We train the Moses on MSL-based generated synthetic corpus with 6-gram KenLM language

model to train the Source→Target ZST model.

C **Transformer training on MSL generated corpus architecture:** We train the Transformer on MSL-based generated synthetic corpus with five layers of encoder and decoder to train the Source→Target ZST model.

D **Moses training on TSL generated corpus architecture:** We train the Moses on TSL-based generated synthetic corpus with 6-gram KenLM language model to train the Source→Target ZST model.

4.3 Data and Experimental Setup

In this section, we discuss the datasets and the experimental settings required to execute the models and analyze the results.

4.3.1 Data Preparation

We evaluate our proposed model on two language pairs (four translation directions): Hindi→Bhojpuri, Bhojpuri→ Hindi, Hindi→Magahi and Magahi→Hindi. Since, all the used language pairs have zero training data, we employ the model training on Nepali↔Hindi parallel corpus for semi-supervised TL learning. Training and development dataset for Nepali-Hindi parallel corpus are obtained from WMT 2019 similar language shared task [81], Opus [86], and TDIL[1]. In addition, the LoResMT2020[2] shared task provided a monolingual corpus as well as the development and test sets for the Bhojpuri and Magahi [101]. Table 4.2 summarises data statistics. All datasets are preprocessed using SentencePiece[3] tokenizer. The proposed model learns 5000 merge operations and restricts the source and target vocabulary to the most frequent 5000

[1]http://www.tdil-dc.in/index.php?lang=en
[2]https://sites.google.com/view/loresmt/loresmt-2020
[3]https://github.com/google/sentencepiece

tokens for the Transformer architecture.

Table 4.2: Description of corpus statistics

Languages	Types	Sentences
NE↔HI**	Training	136991
	Development	3000
HI*	Training	473605
BHO*	Training	91131
MAG*	Training	148606
BHO↔HI**	Development	500
	Test	500
MAG↔HI**	Development	500
	Test	500

* Monolingual data.
** Parallel data.

4.3.2 Experimental Setup

This part discusses the experimental settings required to train the TLSPG and baseline models.

4.3.2.1 TLSPG

For TSL, we use the NMT model based on Transformer architecture. Transformer has been trained and evaluated on the open-source Fairseq toolkit [87]. We have trained the model on the default parameters of Kumar et al. [27] as described in Table 4.3, for better comparison. For MSL, we use Moses[4], a phrase-based statistical MT model. We apply GIZA++ and KenLM to train the translation and language models of Moses, respectively. Moreover, GIZA++ is employed for phrase alignment based on the Markov model. We also use MERT (Minimum Error Rate Training) to tune the model. We train the KenLM on the different setups of 1 to 6-gram and consider 6-gram in our

[4]http://www.statmt.org/moses/

experiments for a critical analysis of models. In the pseudo-corpus generation module, Transformer and Moses are trained on the same settings as described in TSL and MSL models.

Table 4.3: Experimental setup used to train the TSL model

Parameter	**Value**
Model	Transformer
Encoder and Decoder layers	5
Encoder embedding dimension	512
Decoder embedding dimension	512
Encoder attention heads	2
Decoder attention heads	2
Dropout	0.4
Attention dropout	0.2
Optimizer	Adam
Learning rate scheduler	inverse sqrt
Learning rate	1e-3
Minimum learning rate	1e-9
Adam-betas	(0.9, 0.98)
Number of epochs	100

4.3.2.2 Baselines

We use mBART [28], a state-of-the-art multilingual and zero-shot MT approach to compare our proposed TLSPG model. We employ the NE↔HI component of the multilingual NMT method from the pre-trained *mbart.cc*25 [28] model to directly evaluate it on BHO↔HI and MAG↔HI test sets in zero-shot conditions due to similarity among NE↔HI, BHO↔HI and MAG↔HI language pairs. In addition to mBART, we also compare our model performance with the work done by [27] on the same training and test dataset.

4.4 Results and Analysis

We evaluate our model on three metrics: BLEU [72], chrF2 [93], and TER [92]. To compute each metric, we use SacreBLEU [110] tool. From the obtained scores of each metric listed in Table 4.4 on different models, we see that the proposed approach outperforms the existing state-of-the-art models in all three metrics. The reported scores of each metric also show a lot of co-relation between each other. We can conclude that the Moses-based system performs better than Transformer. The similarity (relatedness) factors shared by Bhojpuri, Magahi, Nepali and Hindi account for the Moses-based system's superior performance over a transformer-based approach. This is because SMT is known to perform better than NMT for small amounts of training data.

Table 4.4: Experimental results for different language pairs

Language Pairs	**Scores**	**mBART**	[27]	**MSL**	**TSL**	**A**	**B**	**C**	**D**
Bhojpuri→Hindi	BLEU	2.63	19.5	32.78	17.80	19.44	35.06	19.49	32.94
	chrF2	0.46	-	0.53	0.56	0.58	0.57	0.59	0.57
	TER	1.000	-	0.459	0.656	0.609	0.549	0.595	0.565
Magahi→Hindi	BLEU	3.50	13.71	16.67	12.58	14.94	20.54	16.14	21.84
	chrF2	0.43	-	0.44	0.43	0.46	0.43	0.49	0.48
	TER	1.000	-	0.601	0.725	0.662	0.652	0.654	0.618
Hindi→Bhojpuri	BLEU	0.16	2.54	1.16	2.61	3.78	4.77	2.56	6.52
	chrF2	0.08	-	0.16	0.15	0.17	0.24	0.16	0.25
	TER	1.000	-	1.313	0.995	0.971	0.803	0.982	0.780
Hindi→Magahi	BLEU	0.19	3.16	1.17	2.89	3.18	3.50	2.39	5.15
	chrF2	0.07	-	0.16	0.14	0.17	0.22	0.15	0.24
	TER	1.000	-	1.376	1.027	1.023	0.850	1.025	0.896

[A:] Transformer model training on TSL generated corpus architecture
[B:] Moses training on MSL generated corpus architecture
[C:] Transformer training on MSL generated corpus architecture
[D:] Moses training on TSL generated corpus architecture

4.4.1 Impact of TSL and MSL

Without using the pseudo-corpus generation approach, TSL and MSL in TLSPG get an improvement of +15 and +30 BLEU on BHO→HI respectively, +9 and +13 BLEU on MAG→HI, respectively, +2 and +1 BLEU on HI→BHO respectively, and +2 and

+1 BLEU on HI→MAG respectively compared to mBART model. One of the possible reasons behind improvement is the high relatedness between the language pairs.

4.4.2 Impact of TLSPG

Our proposed method, TLSPG, gets an improvement of +32.43, +18.34, +6.36 and +4.96 BLEU points for BHO→HI, MAG→HI, HI→BHO and HI→MAG, respectively compared to mBART model. This shows that relatedness can play a major role in improving the ZST systems. Apart from these improvements, we also noticed a large variation of BLEU between X→HI and HI→X (where X are BHO and MAG). Such a large variation of BLEU while changing the language direction depends on the complexity of the languages described in the next section.

4.4.3 Relatedness between languages

In this part, we perform some empirical analysis on the relatedness factor of languages to analyze the large improvement in score. We use a corpus based approach, SSNGLM-Score [94], to measure the similarity (relatedness) between languages.

4.4.3.1 Cross-lingual similarity between languages using SSNGLMScore

We use the similarity metric given by [94], called as SSNGLMScore, to measure the relatedness between the languages, defined as follows:

$$S_{sl,tl} = \sum_{tl=1}^{m} p_{sl.tl}(w_n|w_1^{n-1}), \tag{4.7}$$

where S stands for Scaled Sum of n-gram language model scores.

$$MS_{sl,tl} = \frac{S_{sl,tl} - \min(S_{SL,TL})}{\max(S_{SL,TL}) - \min(S_{SL,TL})}, \tag{4.8}$$

where, sl and tl represent the source language and the target language, respectively. Moreover, $sl \in$ SL(Nepali, Bhojpuri, Hindi, Magahi) and m is the total number of sentences in the target language $tl \in$ TL(Nepali, Bhojpuri, Hindi, Magahi). We train the language model using a 6-gram character-level KenLM model on the source monolingual corpus (sl). Each language model is tested on target language (tl), and the scores are reported.

Table 4.5 lists the cross-lingual similarity scores of Bhojpuri, Magahi, Nepali and Hindi with each other. The values in Table 4.5 indicate how closely languages are related to one another. It empirically justifies the relatedness between languages that show highly co-relation with the results described in Table 4.4. This relatedness between languages aids the performance of our proposed model for zero-resource languages, as shown in Table 4.4. We see the performance of Hindi→Bhojpuri is close to Hindi→Magahi. The models for Hindi→Bhojpuri and Hindi→Magahi are built by applying a transfer learning approach on Hindi→Nepali language pair. Moreover, based on Table 4.6, Nepali is morphologically more complex than Bhojpuri and Magahi. The initial pair of translations for Hindi→Bhojpuri and Hindi→Magahi was Hindi→Nepali, with Nepali as the target language, which decreases MT performance [59]. The complexity between languages hurts the transferable parameters of Bhojpuri and Magahi. Hence, the differences in the BLUE scores between Hindi→Bhojpuri and Hindi→Magahi are very close. The reason for the better score of Bhojpuri→Hindi than Magahi→Hindi may be that the difference in out-of-vocabulary words between Magahi and Bhojpuri with Nepali. Thus, we analyze out-of-vocabulary in the following Section.

4.4.4 Impact of out-of-vocabulary

Out-of-vocabulary is the collection of words present in test data but absent in training data. Reason for the better score of Bhojpuri→Hindi than Magahi→Hindi could be that there are more out-of-vocabulary present in Magahi test data than that of Bhojpuri

Table 4.5: SSNGLMScore

Score	BHO	MAG	NE	HI
BHO	-	0.3015	0.2823	0.2996
MAG	0.3015	-	0.3635	0.3366
NE	0.2823	0.3635	-	0.4845
HI	0.2996	0.3366	0.4845	-

test data. Since we use Nepali→Hindi as training data because of its relatedness with Bhojpuri→Hindi and Magahi→Hindi language pairs, we compute the out-of-vocabulary ratio between the source languages such as between Nepali and Bhojpuri, and between Nepali and Magahi. For this, we perform the set operations on training data of Nepali with test data of Bhojpuri and Magahi. Out-of-vocabulary ratio of test data with training data of Bhojpuri is 45.79% and for Magahi it is 69.98% concerning Nepali. We also find that Magahi has 3.16 times more number out-of-vocabulary words compared to Bhojpuri. The reason behind Magahi having the highest OOV ratio with Nepali may be the nature of the corpus. Nepali corpus consists of Entertainment, Agriculture, Bible and WMT2019 datasets and Magahi contains the sentences consisting of speech delivered by PM of India (General domain). This may also be the reason behind the better BLEU score of Bhojpuri→Hindi compared to Magahi→Hindi despite Magahi being more similar to Nepali than Bhojpuri.

4.4.5 Impact of language complexity

Our studies primarily include morphologically diverse languages. To correlate our findings with the morphological richness of languages, we have used corpus-based complexity scales.

4.4.5.1 Character-level entropy of languages

The average information content of words is represented by Entropy [98]. This metric would be higher for languages with a wider variety of word forms, i.e., languages that learn more details into word structure rather than a phrase or sentence structure.

A "word" is defined in our experiments as a space-separated token, i.e., a string of alphanumeric Unicode characters delimited by white spaces. The average information content of character types for words is then calculated in terms of Shannon entropy [99]:

$$H(T) = -\sum_{i=1}^{V} p(c_i) \log_2(p(c_i)) \tag{4.9}$$

where V is number of characters (c_i) in a word.

Table 4.6 lists the word (unigram) entropy of languages at character level, which indirectly represents languages' lexical richness, i.e., how complex – in terms of characters they are made up of – word forms are. Since we compute the unigram entropy based on characters, we can say that lexical richness also indicates morphological complexity, both derivational and inflectional. According to Table 4.6, Magahi and Hindi have entropy scores of 5.1480 and 5.1474, respectively, indicating that Magahi is morphologically close to Hindi based on the data used. Translation direction for Hindi→Bhojpuri is from high to low lexical rich language and translation direction for Hindi→Magahi is from high to high lexical rich language (Magahi and Hindi are morphologically close based on the data used). Thus, high lexical richness of languages is the reason behind the low scores of Hindi→Bhojpuri and Hindi→Magahi compared to Bhojpuri →Hindi and Magahi→Hindi as shown in Table 4.4.

4.4.6 Using LSTM in-place of Transformer in TLSPG

In addition to the above analysis, we have also conducted a study by replacing some components of the proposed approach. This replacement shows how the approach gets

Table 4.6: Entropy

Languages	Entropy
Hindi	5.1474
Nepali	5.6140
Bhojpuri	4.9658
Magahi	5.1480

affected without the particular part. Here, we have replaced the Transformer part of the TLSPG approach with Long Short Term Memory *(LSTM)* architecture. We have performed experiments on four models: LSSL (replacing transformer in TSL), A_{lstm} (replacing transformer in variant A), C_{lstm} (replacing transformer in variant C) and D_{lstm} (replacing transformer in variant D). Results on all the four models are listed in Table 4.7. We have observed that replacing the transformer with LSTM hurts the performance of all the models. This study shows the effectiveness of the transformer in the proposed TLSPG approach.

Table 4.7: Experiments on using LSTM instead of Transformer

Language Pairs	Metrics	LSSL	A_{lstm}	C_{lstm}	D_{lstm}
Bhojpuri→Hindi	BLEU	16.7	18.41	17.21	29.43
	chrF2	0.54	0.57	0.56	0.55
	TER	0.661	0.611	0.601	0.569
Magahi→Hindi	BLEU	11.0	12.28	15.10	18.41
	chrF2	0.41	0.46	0.48	0.48
	TER	0.695	0.662	0.657	0.622
Hindi→Bhojpuri	BLEU	2.52	2.91	2.51	5.48
	chrF2	0.12	0.15	0.11	0.19
	TER	1.061	0.975	0.991	0.7889
Hindi→Magahi	BLEU	2.91	2.99	2.28	5.09
	chrF2	0.16	0.16	0.15	0.19
	TER	1.121	1.027	1.029	0.899

A_{LSTM} : LSTM model training on LSSL generated corpus architecture
C_{LSTM} : LSTM training on MSL generated corpus architecture
D_{LSTM} : Moses training on LSSL generated corpus architecture

4.5 Summary

In this chapter, we have proposed TLSPG, a transfer learning-based semi-supervised pseudo-corpus generation approach for zero-shot translation systems. We have demonstrated the effectiveness of the proposed model on Bhojpuri ↔Hindi and Magahi↔Hindi language pairs in four different directions. The proposed approach outperformed the state-of-the-art models with +15.56 on Bhojpuri→Hindi, +8.13 on Magahi→Hindi, +3.98 on Hindi→Bhojpuri and +2 on Hindi→Magahi language pairs, respectively. We have also conducted various experiments using corpus-based approaches to support the performance of our model.