Chapter 2

Theoretical Background and Literature Review

In this chapter, we will review some background concepts and previous work closely related to machine translation for low-resource languages.

2.1 Theoretical Background

2.1.1 Languages

A language is a structured system of communication. The structure of a language is its grammar, and the free components are its vocabulary. Humans' primary means of communication are languages, which can be expressed through speech (spoken language), signs, or writing. Based on the number of speakers and digitised form of resources, we have divided the languages into three types discussed as follows:

1. **High Resource Languages:** Languages for which numerous speakers and plenty of digitised resources are available are called High Resource Languages (HRLs).
2. **Low Resource Languages:** Languages for which digitised forms of resources are not available in sufficient amounts despite having native speakers are known as Low Resource Languages (LRLs).

3. **Endangered Languages:** Languages which face the risk of becoming extinct, i.e., without any native speakers in the near or medium term future.

2.1.2 Related or Similar Languages

The two terms related or similar languages together refer to a group of languages that share common ancestry or extensive contact for an extended period, or both, with each other, leading them to exhibit structural and linguistic similarities even across language families. Some similarities can also be due to chance. Examples of languages that share common ancestors are Indo-Aryan languages, Romance languages, and Slavic languages. Languages in contact for a long period lead to the convergence of linguistic features even if languages do not belong to common ancestors. Prolonged contact languages could lead to the formation of linguistic areas. Examples of such linguistic areas are the Indian subcontinent [10], Balkan [11], and Standard Average European [12] linguistic areas.

The similarity between languages depends on various factors; some of the factors are lexical similarity, structural correspondence, and morphological isomorphism to different degrees. Lexical similarity means that the languages share many words with similar forms (spelling/ pronunciation) and meaning, e.g. Sunday is written as रविवार (ravivar) in Hindi and रबिवार (rabivar) in Bhojpuri (both are proximate and related Indo-Aryan languages). These lexically similar words could be cognates, lateral borrowings, or loan words from other languages. Structural correspondence can mean that languages have the same basic word order, viz. SOV (Subject-Object-Verb), SVO Subject-Verb-Object). Similar languages also tend to possess structural correspondence to varying degrees. Morphological isomorphism refers to the one-one correspondence between inflectional affixes, at least to some degree. While content words are borrowed or inherited across similar languages, function words are generally not lexically similar across languages. However, function words in related languages (whether suffixes or

free words) tend to have a one-one correspondence to varying degrees.

2.1.3 Categories of machine translation

Broadly speaking, machine translation fits into two categories:

1. **Generic machine translation:** This kind of machine translation is intended to translate between any pair of languages and for data from any kind of domain. Google is one such example. Translations are not domain-specific and do not allow for any control over vocabulary, writing style, or domain context. Generic machine translation is useful for obtaining the gist or rough meaning of text written in another language, but it is less useful for publishing content. Generic machine translation raises a slew of privacy concerns, making your data available to the translating engine, where it might be used in ways you do not have control over.
2. **Custom machine translation:** Machine translation that has been trained to better meet specific requirements. Some machine translation service providers will provide us with domain-specific data to assist in customising our machine translation engine. Omniscien[1] takes it a step further with advanced data collection and processing tools. Data Synthesis and Bilingual Data Mining tools, for example, offer distinct advantages by providing millions of additional high-quality bilingual sentences. Regardless of translation technology, the best translations are usually produced by tailoring machine translation to a specific purpose and domain.

The five common most types of machine translation systems are:

2.1.3.1 Rule-Based Machine Translation

Rule-Based Machine Translation (RBMT) systems were the first commercial machine translation systems and are based on linguistic and grammatical rules that allow the

[1] https://omniscien.com/

words to be put in different places and to have different meanings depending on the context.

2.1.3.2 Statistical Machine Translation

Statistical Machine Translation (SMT) learns how to translate by analysing existing human translations (known as bilingual text corpora). Most modern SMT systems are phrase-based and assemble translations using overlapping phrases. In phrase-based translation, the aim is to reduce the restrictions of word-based translation by translating whole sequences of words where the lengths of phrases in the two languages may differ. Matching bilingual phrase patterns are used and the optimal target language sequences are selected by comparing them to patterns of monolingual data in the target language.

2.1.3.3 Syntax-Based Machine Translation

The idea of Syntax-Based Machine Translation (SBMT) is quite old and is based on the idea of translating syntactic units rather than single words or strings of words (as in phrase-based/SMT), i.e. (partial) parse trees of sentences/utterances. Syntax-Based Machine Translation techniques aim to incorporate an explicit representation of syntax into the statistical machine translation systems.

2.1.3.4 Neural Machine Translation

Neural Machine Translation (NMT) is a cutting-edge machine translation technique that employs neural network architecture to predict the likelihood of a sequence of words, which can be a text fragment, a complete sentence, or with the latest advances, an entire document.

2.1.3.5 Hybrid Machine Translation

Hybrid Machine Translation (HMT) is a machine translation method characterised by the use of multiple machine translation approaches within a single machine translation system to deliver a higher quality translation. There are several approaches to Hybrid MT, such as multi-engine, statistical rule generation, multi-pass, and confidence-based. Hybrid MT can also combine rule-based approaches with statistical MT or NMT, at the level of modules, or at the level of the whole system.

2.1.4 NMT

This section covers the fundamentals of three NMT architectures: Recurrent Neural Network *(RNN)*, attention and Transformer.

2.1.4.1 RNN-based NMT

Such NMT uses an encoder-decoder architecture mainly consisting of an RNN unit, where the encoder takes source word embedding x_i of a sequence $X = \{x_1, x_2,...,x_i,..., x_n\}$ as input to each RNN unit. Each RNN unit produces a hidden state h_i and passes it to the next RNN unit in sequence. Finally, the last RNN unit of sequence produces a context vector u that contains context information of all previous RNN units. The decoder uses this context vector u to generate the predicted sequence as follows [13]:

$$P(Y|X) = \prod softmax(f(y_j|y_{j-1}, h_j, u)) \tag{2.1}$$

where Y is predicted target sequence, y_j is word at position j in target sequence and h_j is the hidden state at j^{th} position in decoder side.

2.1.4.2 Attention-based NMT

In attention-based NMT, bi-directional (forward and backwards) one RNN acts as an encoder, and another RNN acts as a decoder. In encoding, each source word x_i of a sequence $X = \{x_1, x_2,...,x_i,...,x_n\}$ is represented as $h_i = [\text{concat}(\overrightarrow{h_i}, \overleftarrow{h_i})]$ where $\overrightarrow{h_i}$ is forward hidden state and $\overleftarrow{h_i}$ is the backward state. During decoding the hidden state q_t at time t is computed as follows [13]:

$$q_t = f(q_{t-1}, y_{t-1}, U_t). \tag{2.2}$$

Furthermore, U_t is computed as:

$$U_t = \sum_{i=1}^{T} \alpha_{ti} h_i \tag{2.3}$$

where α_{ti} is attention weight between source word x_i and target word y_t.

2.1.4.3 Transformer-based NMT

A transformer in NMT is a sequence transduction model entirely based on an attention mechanism and positional encoding, having multi-headed self-attention and thus not relying on RNN layers earlier used in encoder-decoder architectures [14]. A Transformer also comprises encoder and decoder stacks with the same number of layers (Fig. 5.3). Each encoder and decoder layer is divided into two sublayers, a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. In encoding, each source word embedding x_i of a sequence $X = \{x_1, x_2,...,x_n\}$ is added to the positional encoding and given to the encoder as input. Decoder layers have one extra sub-layer named Masked multi-head attention, which performs multi-head attention over the output of the encoder stack. Similar to an encoder, target word embedding y_i of a gold target sequence $Y = \{y_1, y_2,...,y_n\}$ is added to the positional encoding and given to decoder as target input embedding for decoding purpose.

Unlike traditional attention-based NMT systems, the Transformer defined attention as mapping a query and a set of key-value pairs to an output. The output is a weighted sum of the values, with the weight assigned to each value based on the query's compatibility function. So the attention is computed as:

$$\mathbf{attn} = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.4}$$

where **Q**, **K** and **V** are query, key and value respectively, and d_k is the dimension of key.

2.1.5 Reinforcement learning

Reinforcement learning is a machine learning training method that involves rewarding desired behaviours and/or punishing undesirable ones. A reinforcement learning agent, in general, can interact with its environment, act, and learn through trial and error, as shown in Fig. 2.1. Developers have to devise a method of rewarding desired behaviours while punishing negative behaviours in reinforcement learning. This method assigns positive values to desired actions and negative values to undesirable behaviours to encourage the agent. In order to achieve an optimal solution, the agent is programmed to seek long-term and maximum overall reward.

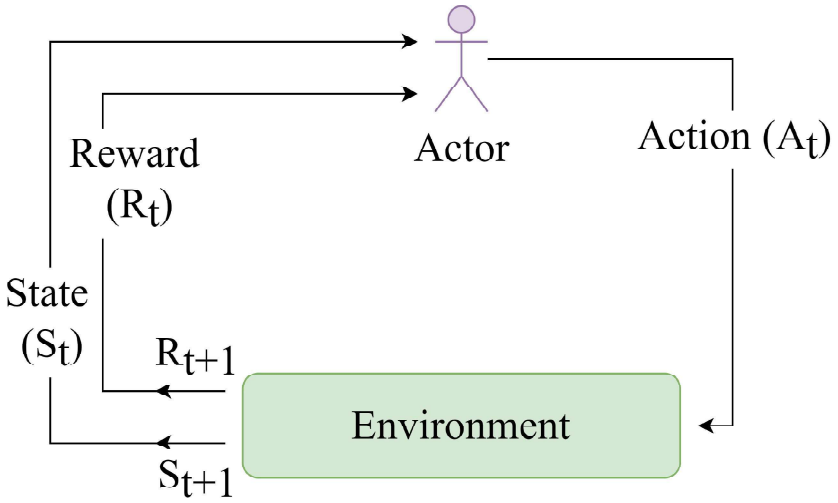

Figure 2.1: Reinforcement Learning.

These long-term objectives prevent the agent from stalling on shorter-term objectives. With time, the agent learns to avoid the negative and seek the positive. This learning method has been adopted in Artificial Intelligence (AI) to direct unsupervised machine learning through rewards and penalties.

2.2 Literature Review

In this section, we categorise the literature review into four sub-problems, which are discussed as follows:

2.2.1 Zero-shot Problem

In this part, we review the existing ZST systems shown in Table 2.1. In [15], authors proposed a finetuning algorithm for a multiway, multilingual NMT model to translate zero-resource language pairs. In [16], authors fed all training data into a single NMT engine and trained the model. In the work of [17], authors demonstrated a zero-shot system consisting of reinforcement and dual learning.

In the work of [18], authors have suggested a multilingual NMT on a zero-shot direction based on monolingual data and demonstrated that the self-learning technique improves the efficiency of multilingual zero-shot directions by using bilingual parallel corpora for training. In [19], authors demonstrated a multilingual encoder-decoder NMT architecture with an explicit neural interlingua for performing direct ZST.

In [20], authors designed a setup by setting a "Chain" of languages for 12 language pairs on the standard IWSLT 2017 multilingual benchmark. In [21], authors presented a multilingual MT system for ZST on 110 unique translation directions trained on WMT 2019 shared parallel task datasets and evaluated by creating gold sets for zero-shot pairs in TED talks multi-parallel datasets.

In [22], authors addressed the degeneracy issue by quantitatively analysing the mutual information between the language of the source and decoded sentences. In [23],

authors performed a zero-shot experiment on 103 languages trained on 25 billion examples. In [24], authors proposed an auxiliary loss forcing the model to learn the source language invariant representations that improve generalisation. In [25], authors focused on ZST generalisation and proposed a consistent agreement-based learning approach for zero-shot translation. In [26], authors demonstrated the feasibility of back-translation to allow for massively ZST and conduct the experiments on the multilingual dataset.

Kumar et al. [27] proposed a bilingual-based ZST system for Bhojpuri↔Hindi and Magahi↔Hindi language pairs. It is based on an unsupervised domain adaptation approach. In [28], authors presented $mBART-$ a sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages using the BART objective. In [29], authors proposed a novel zero-shot NMT approach, which includes three stages: initialisation, augmentation, and training for constructing a self-learning cycle of zero-shot pair.

Table 2.1: Comparative overview of the closely related existing models for zero-shot problem

Papers	Types of MT		Techniques		Training model	Zero-resource Language pairs
	Bilingual	Multilingual	Finetuning	Pivot	Neural	
[15]		✓	✓		GRU	ES→FR
[17]		✓			LSTM	ES↔FR, ES↔RU, RU↔FR
[16]		✓			LSTM	PT→ES, ES→JA, EN↔{BE, RU, UK}
[18]	✓	✓			RNN and Transformer	IT↔RO
[19]		✓			LSTM	FR↔RU, ES↔ZH, ES↔FR
[20]		✓			Transformer	EN↔RO, DE↔IT, EN↔NL, NL↔IT, DE↔RO, NL↔RO
[21]		✓			Transformer	CS, DE. FI, GU, KK, LT, RU, TR, ZH , FR (88 directions)
[22]		✓			Transformer	DE→IT, DE→NL, AR→RU, AR→ZH, RU→ZH
[23]		✓			Transformer	DE→FR, BE→RU, Yi→DE, FR→ZH, HI→FI, RU→FI
[24]		✓	✓	✓	Transformer	DE↔FR
[25]		✓	✓		LSTM	ES↔DE, ES↔FR, DE↔FR
[26]		✓	✓	✓	Transformer	OPUS-100
[27]	✓				Transformer	BHO↔HI, MAG↔HI
[28]		✓	✓		mBART	NL↔EN, AR↔EN, NL↔DE
[29]	✓	✓		✓	Transformer	AZ↔EN, BE↔EN, GL↔EN, SK↔EN

Note: EN: English, ES: Spanish, DE:German, FR: France, RU: Russian, PT: Portuguese, JA: Japanese, ZH: Chinese, GU: Gujarati, HI: Hindi, IT: Italian,, RO: Romanian, BHO: Bhojpuri, MAG: Magahi, BE: Belarusian, UK: Ukrainian, CS: Czech, NL: Dutch, FI: Finnish, KK: Kazakh, LT: Lithuanian, TR: Turkish, AR: Arabic, AZ: Azerbaijani, GL: Galician, SK: Slovak, GRU: Gated Recurrent Unit, LSTM: Long Short-Term Memory, RNN: Recurrent Neural Network.

2.2.2 Morphological Richness

This section summarizes closely related works (Table 2.2) based on language similarity, morphological richness, statistical and neural models, and language pairs used as discussed below.

Although there had been work in the past, the recent sharper focus on machine translation for similar languages is also due to the shared tasks on this topic organized as part of the WMT conferences from 2019 to 2021. In [30], authors demonstrated that pre-training could help even when the language used for fine-tuning is absent during pre-training. In [31], authors experimented with attention-based recurrent neural network architecture (seq2seq) on HI↔MR and explored the use of different linguistic features like part-of-speech and morphological features, along with back translation for HI→MR and MR→HI machine translation. In [32], authors ensembled two Transformer models to try to allow the NMT system to learn the nuances of translation for low-resource language pairs by taking advantage of the fact that the source and target languages are written using the same script. In [33], authors' work relied on NMT with attention mechanism for the similar language translation in the WMT19 shared task in the context of NE↔HI language pair.

In [34], the authors conducted a series of experiments to address the challenges of translation between similar languages. Out of which, the authors developed one phrase-based SMT system and one NMT system using byte-pair embedding for the HI↔MR pair. In [35], authors used a Transformer-based NMT with *sentencepiece* for subword embedding on HI↔MR language pair [36]. In [37], authors used the Transformer-NMT for multilingual model training and evaluated the result on the HI↔MR pair. In [38], authors focused on incorporating monolingual data into NMT models with a back-translation approach. In [39], authors introduced NLP resources for 11 major Indian languages from two major language families. These resources include: large-scale sentence-level monolingual corpora, pre-trained word embeddings, pre-trained language

models, and multiple NLU evaluation datasets. In [40], authors presented IndicBART, a multilingual, sequence-to-sequence pre-trained model focusing on 11 Indic languages and English. IndicBART utilized the orthographic similarity between Indic scripts to improve transfer learning between similar Indic languages.

Table 2.2: Comparison of existing works for morphological richness issue. ✓ and ✗ represent presence and absence of particular feature, respectively.

Paper	Similar Language	Reducing Redundancy Complexity	Statistical	Neural	WX	Language Pair
[30]	✓	✗	✗	✓	✗	HI↔MR, ES↔PT
[31]	✓	✗	✗	✓	✗	HI↔MR
[32]	✓	✗	✗	✓	✗	HI↔MR
[33]	✓	✗	✗	✓	✗	NE↔HI
[34]	✓	✗	✓	✓	✗	HI↔MR
[35]	✓	✗	✗	✓	✗	HI↔MR
[37]	✓	✗	✗	✓	✗	HI↔MR
[38]	✓	✗	✗	✓	✗	ES↔PT, CS↔PL, NE↔HI
[39]	✓	✗	✗	✓	✗	11 Indian languages
[40]	✓	✗	✗	✓	✗	11 Indic languages and English
Proposed approach	✓	✓	✗	✓	✓	{GU,MR,NE,MAI,PA,UR}↔HI

Note- HI: Hindi, MR: Marathi, ES: Spanish, PT: Portuguese, NE: Nepali, CS: Czech, PL:Polish, GU: Gujarati, MAI: Maithili, PA: Punjabi, UR: Urdu

2.2.3 Domain Shift Problem

DA has been applied to statistical and neural-based machine translation models in a variety of ways [41–44]. These works are categorized into data-centric [42, 45] and model-centric [46–48]. Data-centric techniques include fetching training samples from out-of-domain parallel data based on a language model or generating pseudo parallel data [42, 45, 49–51]. In contrast, the model-centric approach focuses on the injection of out-of-domain and in-domain models at the model or instance level [46–48]. The following sections go into detail about works that are closely related to data-centric and model-centric approaches.

2.2.3.1 Data-centric approach

In [45], authors proposed a data selection approach that computes the language model cross-entropy difference for each sentence in a monolingual corpus. [50] developed

an invitation model based on iterative weighted invitations using the Expectation-Maximization *(EM)* algorithm and presented a data selection approach for machine translation on a large parallel corpus comprised of a rather diverse set of domains. In [51], authors used the data selection techniques and EM-based mixture modelling with two joint models: the neural network joint model and the operation sequence model to carry out DA for machine translation. However, these approaches did not talk about using the reward-based method for data selection.

In [13], authors talk about sentence-level DA for NMT. They exploited the internal sentence embedding of NMT and used the similarity between sentence embeddings to fetch pseudo in-domain sentences from the out-of-domain corpus. They also proposed a dynamic training and multi-domain sentence weighting method that balances training samples' domain distributions and reduces the domain shift between training and testing data. However, these approaches give good contributions but do not discuss leveraging the orthographic features of languages and reward-based learning strategy in training the model.

[44] proposed a dynamic approach for selecting and weighting sentences in iterative back-translation. In [52], authors presented an effective approach for filtering the synthetic parallel corpus. Using a convolutional neural network, they trained two binary classifiers for source and target language by merging the out-of-domain and in-domain corpora. They generated synthetic corpus using iterative back-translation and performed classifier-based domain filtering. These approaches focused on extending the selection approaches to iterative back-translation, but there is no focus on reinforcement strategy for selecting the sentences.

2.2.3.2 Model-centric approach

In [46], the authors proposed the fill-up adaptation method and tested it on a speech translation task. [47] described sentence-weight-based adaptation approach depending

on the closeness between sentences in the target domain text and the training set. In [48], authors proposed the Feature Decay Algorithm-5 *(FDA5)*, a parameterization, optimization, and implementation framework for a type of instance selection algorithm that employ feature decay.

[53] proposed vocabulary adaptation method for fine-tuning. They replace the NMT model's embedding layers by projecting word embeddings of monolingual corpus in a target domain onto a source-domain embedding space. Vocabulary adaptation helps the model to adapt the vocabulary of a pretrained NMT model. In [54], authors reduced the domain shift by linking exposure bias to hallucinations. In [55], authors pruned the model and only kept the essential parameters for in-domain and general domain translation. They further trained the pruned model with knowledge distillation based on the original model. Finally, they resize the model to its original size and fine-tune the new parameters for in-domain translation. However, all the discussed approaches do not consider reward-based learning in selecting the sentences of out-of-domain data.

Table 2.3: Comparison of related work for domain shift problem

Paper	Neural	REINFORCE	MLE	MRT	Language pair
[45]	✗	✗	✗	✗	EN-FR
[46]	✗	✗	✗	✗	AR-EN, EN-FR
[42]	✗	✗	✗	✗	ZH-EN
[50]	✗	✗	✗	✗	EN-ES
[49]	✗	✗	✗	✗	EN-DE, FR, RU, ES
[51]	✗	✗	✗	✗	DE-EN, AR-EN
[47]	✗	✗	✗	✗	ZH-EN
[48]	✗	✗	✗	✗	EN-DE, EN-TR
[13]	✓	✗	✓	✗	EN-FR, EN-DE
[53]	✓	✗	✓	✗	EN-JA, DE-EN
[44]	✓	✗	✓	✗	EN-DE, EN-IT
[54]	✓	✗	✓	✓	DE-EN
[55]	✓	✗	✓	✗	EN-FR, EN-DE, ZH-EN
[52]	✓	✗	✓	✗	DE-EN
Proposed Method	✓	✓	✓	✓	HI-NE, HI-MR

Note- MRT: Minimum Risk Training, MLE: Maximum Likelihood Estimation, FR: French, EN: English, ZH: Chinese, ES: Spanish, DE: German, RU: Russian, AR: Arabic, TR: Turkish, JA: Japanese, IT: Italian

2.2.4 Missing Context and Rare-word Problem

As shown in Table 2.4, the following section examines the closely related works with context information and the rare word problem encountered by NMT.

2.2.4.1 Context information

In [14], authors proposed the Transformer, the first sequence transduction model based mainly on attention and positional encoding, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention. In [56], authors presented a GRU-GAtt for NMT. Instead of using decoding-invariant source representations, GAtt produces new source representations that vary across different decoding steps according to the partial translation to improve the discrimination of context vectors for translation. In [57], authors used deep reinforcement learning to automatically optimize attention distribution via DRGA framework for various NLP task. Our model is based on this DRGA framework. In [58], authors introduced Nepali–English and Sinhala–English datasets and demonstrated that existing methods perform rather poorly on these datasets, posed a challenge to the research community working on low-resource NMT. In [59], authors proposed a GAN based NMT model by exploiting morphological word embedding as input of the model training and using multiple reference translations instead of a single one to optimize the parameters of discriminator. However, there is less work on NMT that exploits phonetic features of languages or uses common phonetic and orthographic space for languages with different writing systems.

In [60], authors examined approaches for multi-source NMT using incomplete multilingual corpora. In [61], authors proposed the RelateLM that explores relatedness between the LRLs and a Related Prominent Language (RPL) already present in the language model. In [62], authors explored effective methods to exploit parallel data from multiple related languages to improve the translation between Indian languages

and English. In [63], authors proposed a multilingual extension of the machine oriented-reinforce algorithm able to work in zero-shot settings. In [64], authors showed that it is possible to leverage monolingual corpora of assisting languages to pre-train NMT models for language pairs that lack parallel as well as monolingual data. In [65], authors proposed to use both textual and phonetic information in NMT by combining them in the input embedding layer of neural networks. In [66], we worked on the combined approach of domain adaptation and back-translation, and achived promising results for Bhojpuri↔Hindi and Magahi↔Hindi zero-shot MT system. However, this work did use phonetic information at a low granularity level.

2.2.4.2 Rare word problem

In [67], authors proposed an effective technique to address the rare word problem with the help of dictionary. In [68], authors showed that NMT systems can handle open-vocabulary translation by representing the unseen and rare words with subword units. However, there is a need to focus on phonetic information along with text. In [69], authors proposed SentencePiece, a language-independent subword tokenizer and detokenizer designed for neural-based text processing, including NMT. In [70], authors incorporated a class-specific copy network in NMT to handle the rare word problem. In [71], authors proposed three different strategies to handle rare words in NMT, in which the combination of methods brings significant improvements to the NMT systems on LRLs pairs. However, these techniques do not use the phonetic context of languages.

2.3 Research gap

As mentioned above in the literature study, we are attempting to address four problems that low-resource machine translation for Indian languages encounters. In this section, we go through key research gaps that need to be filled regarding each problem in order

Table 2.4: Comparison of existing works for context missing and rare-word problem

Paper	SWE	SPE	IPA	BS	CTP
[14]	✓	✗	✗	✓	✗
[56]	✗	✗	✗	✓	✗
[67]	✗	✗	✗	✗	✗
[68]	✓	✗	✗	✓	✗
[57]	✗	✗	✗	✓	✗
[58]	✓	✗	✗	✓	✗
[59]	✗	✗	✗	✓	✗
[60]	✓	✗	✗	✓	✗
[61]	✓	✗	✗	✗	✗
[62]	✓	✗	✗	✗	✗
[63]	✓	✗	✗	✗	✗
[64]	✓	✗	✗	✓	✗
[65]	✓	✗	✗	✗	✗
[66]	✓	✗	✗	✓	✗
[69]	✓	✗	✗	✗	✗
[70]	✗	✗	✗	✗	✗
[71]	✓	✗	✗	✗	✗
Proposed approach	✓	✓	✓	✓	✓

Note- SWE: Orthographic Sub-Word Embedding, SPE: Phonetic Sub-Word Embedding, BS: Beam Search, IPA: International Phonetic Alphabet, CTP: Sentence-wise BLEU comparison between predicted textual and phonetic sequences.

to improve the current MTs which are discussed as follows:

2.3.1 Zero-shot problem

As mentioned above in Table 2.1, most of the existing methods for ZST are mainly outcomes of the multilingual NMT model and specifically trained on the combinations of HRLs and LRLs to improve the qualities of the translation of zero-shot languages. The insufficient availability of parallel corpora acts as a hindrance in developing the ZST systems. There is a need to focus on TL, especially for zero-resource languages that leverage the relatedness of LRLs pairs even without any help from HRLs.

2.3.2 Morphological richness

In most of the existing works on MT for related languages (e.g., [35], [37], [38]), authors have discussed about improving the NMT models using the extra monolingual corpora in addition to bi-lingual corpora. In addition, there is a need to exploit common orthographic and phonetic space wherever possible. For this, we use projection to common space, one method being conversion to the common WX encoding for multiple Indian languages, which can reduce the redundancy and complexity in language data to enable better learning of models.

2.3.3 Domain shift

As we can see in Table 2.3, methods using sentence selection and weighting having focused on the cross-entropy difference between the out-of-domain sentences and the in-domain corpora. They do not discuss the reward-based strategy for sentence selection which may be beneficial in improving NMT's performance. So, there is need to use reward-based approach in sentence selection which can play an important role in handling the domain shift in MT.

2.3.4 Missing context and rare-word problem

In most of the existing methods (e.g., [67], [68], [61]), authors talked about including textual and morphological information in training the NMT models. There is also a need to include phonological information along with textual and morphological features in NMT which favours the models in capturing extra phonological context and improves the performance of NMT for similar LRLs, such as Indian languages.

2.4 Evaluation metrics

This section discusses the three types of evaluation metrics that are used to assess the performance of the proposed models. These evaluation metrics judge the model's performance by focusing on overlaps between the translated and the reference sentences, and have been shown to have a good correlation with human evaluation [72].

2.4.1 BLEU

BLEU score is a standard metric accepted by NLP researchers to obtain the accuracy of predicted translated outputs compared to the human-translated reference sentences (gold labels). It is observed that the higher the value of the BLEU score, better the output of translations. The formula of the BLEU score is as follows [72]:

$$BLEU = min\left(1, \frac{output\_length}{reference\_length}\right)\left(\prod_{i=1}^{4} precision_i\right) \tag{2.5}$$

where, *output_length* and *reference_length* are the lengths of predicted sentence and the reference sentence, respectively.

2.4.2 chrF2

chrF2 is character n-gram F-score for automatic evaluation of machine translation output. Mathematically, chrF2 is computed as follows:

$$chrF2 = 5\frac{chrP \cdot chrR}{4chrP + chrR} \tag{2.6}$$

where, $chrP$ represents the percentage of n-grams in the hypothesis which have a counterpart in the reference and $chrR$ indicates the percentage of character n-grams in the reference which are also present in the hypothesis.

2.4.3 Translation Edit Rate (TER)

TER is the minimum number of edits needed to change a hypothesis so that it exactly matches one of the references, normalized by the average length of the references.