Neural Processing Letters (2025) 57:23
https://doi.org/10.1007/s11063-025-11718-0

Character-Level Encoding based Neural Machine Translation for Hindi language

Divya Rathod[1] · Arun Kumar Yadav[1] · Mohit Kumar[1] · Divakar Yadav[2]

Accepted: 6 January 2025 / Published online: 18 February 2025
© The Author(s) 2025

Abstract
Neural Machine Translation (NMT) is one step ahead of traditional statistical phrase-based translation systems because of its better translation ability. But it requires a large amount of parallel training data, which can be challenging for languages with limited resources like many Indian languages. In the past, researchers have tried to address the issue using data augmentation. In this paper, we present a data augmentation technique for the Hindi language based on five phrases: noun phrases, verb phrases, prepositional phrases, adjective phrases, and adverb phrases. We augment the training corpus using parser-generated phrasal segments and evaluate the efficiency of the proposed work on the Hindi language. Further, the paper presents training in the NMT model at the character level instead of the word level. This approach can help overcome challenges associated with word-level translations, such as handling rare and out-of-vocabulary words and phrases, dealing with morphological complexity, and addressing languages with ambiguous word boundaries. The proposed work was evaluated on a low-resource language pair, Hindi-English, using the Google Transformer model as the baseline state-of-the-art. The experiments used two distinct datasets: WMT14 Hin-Eng and Samanantar Hin-Eng parallel corpus with character-level encoding for the translation task. The proposed model is able to surpass the cutting-edge baseline and saw an increase in BLEU scores for the WMT14 translation challenge with +2.52 on base paper using three phrase sentences with character-level encoding and +2.68 BLEU Score on base paper using five phrase sentences with character-level encoding. Further, character-level encoding is evaluated on non-augmented Samanantar dataset; it performs better in the baseline approach for translation purposes. It clearly shows that the proposed model outperforms in Hindi language translation.

Keywords Neural machine translation · Hindi neural machine translation · Data augmentation · Character-embeddings

✉ Mohit Kumar
mohit@nith.ac.in

[1] Department of Computer Science and Engineering, Hamirpur, New Delhi, India

[2] SOCIS, Indira Gandhi National Open University (IGNOU), New Delhi, India

1 Introduction

Machine Translation (MT), a branch of computational linguistics, automatically translates a text from a source language into a desired target language. MT has seen significant progress due to increased accuracy in machine learning, NLP and deep learning. Neural Machine Translation (NMT) is the most popular MT method and uses deep learning techniques for better accuracy [1–3]. NMT models use neural networks to comprehend language patterns and structures better, making them more precise and adaptable. Quality control may incorporate human reviewers with Machine Translation, which is entirely computer-generated. The most basic function of machine translation is word-by-word translation, which converts a word from the source language to its translation in the desired language. Since its humble origins, MT has developed significantly, and the precision and quality of translation are now much above expectations. In many language pairs, including those with intricate syntax and morphology, NMT dominates. This allows the model to capture the context of the entire sentence rather than just a fixed number of words, as in Statistical Machine Translation(SMT).

Deep neural networks and artificial intelligence are used in NMT to train neural models, which is a fundamentally different approach to the issue of language translation and localization [4–6]. In the last three years, there has been a significant shift from Statistical Machine Translation(SMT) [7] to Neural Machine Translation [8–10], making NMT the main machine translation methodology. The translations produced by neural machine translation are frequently of far high quality than those produced by statistical machine translation techniques due to their improved fluency and appropriateness. In the past, research has shown that NMT can achieve similar or better translation results than human translators in some cases [11] [12]. This indicates that if a machine can learn the grammar of a specific language, it is possible to translate it accurately with very little human input. Professionals may feel that machines will eventually do all aspects of their jobs due to these improvements in MT technology[13]. According to Bawa-Mason et al. [14], 38% of active translators are concerned that MT tools may eventually replace them. To have a thorough understanding of the performance of NMT in the present, it appears essential in this context to research modern MT systems.

While NMT has achieved considerable success, it is not without limitations. One of the most notable drawbacks is its sub-optimal performance for languages with limited resources, primarily due to the significant amount of training data required [15]. The language pairs with insufficient parallel samples are affected by this, which is undoubtedly an issue. Machine translation researchers have proposed several techniques to deal with the issue of sparse data. The majority of these methods center around enhancing the existing corpus to augment the quantity of parallel training samples. This includes incorporating monolingual data from both source and target sides during training, leveraging training data from other language pairs, and similar approaches. Back-translation is the most effective data augmentation technique that only uses the target monolingual data and can significantly increase the training corpus size [16], making it quite a straightforward process. Gupta et al. [17] present another intriguing method that enhances training data by incorporating syntactic phrasal segments extracted from the parse trees of the language pairs. The approach involves extracting three distinct types of syntactic phrases-noun phrases (NPs), verb phrases (VPs), and prepositional phrases (PPs)-from the constituency trees of the parallel training corpus. Subsequently, they retrieve the source translations of these extracted phrases using a target-to-source machine translation (MT) system. Finally, a new synthetic corpus is constructed from the obtained linguistically coherent source-target phrasal sequences.

The impact of using syntactic phrases as additional training examples has its advantages for any neural machine translation model. However, choosing the appropriate kinds of phrases is a further challenge to be explored and can be experimented with. In this study, we extend the augmentation strategy discussed earlier by Gupta et al. [17], including a different set of parser-generated phrases. Adjectives(ADJPs) and Adverb Phrases(ADVPs) along with the Noun Phrases(NPs), Verb Phrases(VPs), and Preposition Phrases(PPs). The justifications behind this are the following:

i. **Adjective phrases**: Adjective phrases include adjectives and modifiers, such as adverbs or adjectives. These phrases can help the model learn to handle different modifiers and their ordering in the sentence.
ii. **Adverb phrases**: Adverb phrases include an adverb and any modifiers, such as adverbs or prepositional phrases. These phrases can help the model learn to handle different adverb meanings and their position in the sentence.

This paper proposes a Google Transformer [18] based translation technique for the Indian low-resource language (Hindi) using character embedding for translation. To validate the proposed method, we have used HindEnCorp, the popular WMT14 Hindi-English dataset[19] and Samanantar Hin-Eng parallel corpus [20], and have evaluated the results using the widely used metric, the BLEU score [21]. The following are the primary contributions of this study:

- It has been proposed a data augmentation strategy using five synthetic phrases (Nouns, Verbs, Prepositions, Adjectives, and Adverbs) for Hindi-English Translation, earlier [17] presented a similar strategy using only three synthetic phrases.
- Character level encoding along with transformer architecture [18] based method for translation from Hindi to English has been proposed.
- Significant improvements have been reported in Hindi-to-English translation results over the state-of-the-art baseline models. Applying the augmentation strategy and character level encoding for translation on WMT14 [19] corpus, the proposed technique outperforms recent state-of-the-art methods and achieves a 17.85 BLEU-4 score. Furthermore, only character level encoding (without data augmentation) on Samanantar corpus [20] provides a BLEU-4 score of 19.65 that outperforms recent methods.

The rest of the paper is structured as follows. Section 2 includes a brief analysis of the relevant literature. Section 3 covers in depth the data augmentation technique. Section 4 outlines the baseline machine translation models and explains the experimental setup. Section 5 describes the proposed work carried out in the study. The specific details of the datasets used in the experiments are presented in Sect. 6. The results and analysis are discussed in Sect. 7. Section 7.4 shows some failure cases arising from the model, and we conclude the work and highlight future research scope in Sect. 8.

2 Literature Review

This section describes past data augmentation and translation research at the word and character levels, which is highly relevant to our current work. Firstly, we sketch out the applications and uses of different data augmentation strategies. Secondly, we review the various translation strategies and the level of embedding used in each.

2.1 Data Augmentation Strategies

Increasing the amount and diversity of the training dataset is one important strategy for improving the accuracy of machine translation models[22]. Several papers have been published over the years that have proposed different data augmentation techniques for machine translation[23–25], and we will offer a literature review of these strategies in this section. In the very first of the methods, Sennrich et al. [16] proposed back-translation, where a model is trained to translate from a specified target language to a source language, and the resulting translations are used to augment the original dataset. One limitation of this method is that it requires a pre-trained model for back-translation, which may only be available for some languages. Again, Zhang & Zong [26] took forward this approach by using back-translation and monolingual data to generate new training data. Initializing the parameters of the low resource language pair model with those of the high resource language pair model, Nguyen & Chiang [27] applied the transfer learning method. This method works better when the language pairs are related to one another. Transfer learning may show little progress when high and low-resource language pairs are distant or have different domains.

Fadaee et al. [28] produced synthetic training data by adding rare vocabulary words to the existing sentences. If a word's frequency falls below a preset level, it is classified as rare in the dictionary by the authors. The uncommon terms in the sentences were replaced using a language model. Wang et al. [29] gave a similar technique where they employed various sampling techniques to replace words randomly in both the source and destination phrases with words from their vocabulary. Further experimenting on the back-translation strategy, Zhao et al. [30] proposed a method that combines back-translation with a denoising autoencoder to improve the quality of the augmented data. Nevertheless, they do not combine their technique with any other strategy. [22] introduced a relatively new approach. They used semantic information for augmentation. In the paper [31], Shah & Barber proposed a method to use a generative model to obtain new synthetic samples, which are used to augment the training dataset. This approach may not be appropriate for languages with limited resources due to its high computational requirements. As mentioned above, the drawbacks of the research include the reliance on pre-trained back-translation models and the requirement for substantial amounts of high-quality monolingual data. Additionally, these methods may need to address the problem of overfitting fully and may not be robust to domain shift.

In this paper, we augment the training samples and increase them to quite a sizable degree using a linguistically motivated data augmentation technique. We build the constituency tree from the target language, English, and extract five kinds of words and phrases, viz. Noun Phrases, Verb Phrases, Adverb Phrases, Adjective Phrases, and Preposition Phrases. The phrase-based data augmentation technique is computationally cheaper and has no other complex requirements. Furthermore, since phrases induce only word-level knowledge to the training of the NMT model, it might fail to perform well for rare or unseen words. We address this issue by substituting the training technique for individual characters instead of words.

2.2 Word-Level Neural Machine Translation

Word-level encoding has been a popular approach in NMT since its inception. Every machine translation model creates a numerical representation of the input sentence, called an encoding, employed to generate the target translation[32]. The word-level encoding approach involves encoding every word of the input sentence as a unique vector. Many NMT models use word-level encoding, and it has achieved state-of-the-art results on various machine translation

benchmarks[1, 4, 18]. Several studies have compared the effectiveness of word-level encoding to other techniques, including character-level and subword-level encoding. These research findings suggest that the efficacy of the NMT model can be significantly influenced by the encoding method used.

In the paper [18], Vaswani et al. discuss the self-attention-based encoding method. They converted the input sentence into a numerical representation using a self-attention technique. The authors demonstrated using word-level encoding in combination with the transformer architecture led to cutting-edge outcomes on many machine translation tasks. However, the self-attention technique has quadratic compute and memory overhead concerning the input sequence length. This may make training and inference computationally costly, especially for long sequences, limiting scalability and the capacity to translate in real-time in some situations. Sennrich et al. [33] use Byte-Pair Encoding (BPE) technique to evaluate the performance and accuracy of word-level and sub-word-level encoding. The authors found that sub-word-level encoding outperformed word-level encoding regarding computational efficiency and translation quality. Also, the authors claimed that the BPE technique is unsuccessful for low-resource languages due to the need for more training data. A technique for hybrid encoding with both word-level and character-level encoding was suggested by Luong & Manning [34]. They commented that the hybrid encoding strategy performed better on several machine translation benchmarks than individual word-level or character-level models-one major issue in the paper is that it takes large memory and processing power to train every character.

The review of past research demonstrates that the choice of encoding technique can significantly impact the effectiveness of NMT models. The use of word-level encoding is still a common strategy in NMT, and recent research has shown that it can be enhanced by combining it with other techniques, such as character-level or sub-word-level. Nevertheless, with word-level encoding, each word in the source sentence is encoded as a dense vector, which is then used as input to the NMT model. However, this strategy may have drawbacks when dealing with uncommon or Out-of-vocabulary (OOV) words because the NMT model might find it difficult to produce correct representations. This can lead to less-than-ideal translation performance. It motivates us to move towards machine translation at the character level.

2.3 Character-Level Neural Machine Translation

A method that has gained popularity in recent years for NMT is character-level encoding, which portrays the source text as a sequence of characters rather than words[35]. Since the model can be trained to generate meaningful representations for individual characters, which can then be combined to form representations for words. This encoding approach can address some of the shortcomings of word-level encoding, as discussed in section 2.2, such as handling unfamiliar words and morphological variations. As a result, the model can produce word representations in the source language more accurately, leading to improved translation performance. Some of the translation work using character-level encoding is discussed herewith. Character-level models use simple characters as the fundamental building blocks of sentences rather than an explicit segmentation of the input sequence. Translation models are subtly required to learn the internal structure of complicated words. Therefore, these models can better translate noisy and non-domain text[36].

Kim et al. [37] discuss the character-level convolutional neural network (CNN) for NMT. Based on the prior words and characters, the model develops the ability to anticipate the following word. The authors concluded their character-level model outperformed traditional

word-level models on language modelling tasks. However, the model was computationally expensive and required a significant amount of parallel training samples. Costa-Jussa & Fonollosa [38] used a convolutional network to identify local dependencies from character embedding before using a Highway network to construct segmented embedding. Although this approach produced encouraging results, it still vitally relied on word-level segmentation at the encoder and decoder sides. Using character-level information, a word-based model's out-of-vocabulary difficulties were reduced by Luong & Manning [34]. While the encoder continued to operate at the sub-word level, Chung et al. [39] compared character-level and sub-word-level decoders. They reported that the character-level decoder could outperform the sub-word level. Lee et al. [40] proposed a fully character-level model with similar computational requirements to sub-word-level models to convert a source language's character sequence into a target language's character sequence. They demonstrated that the model is superior to a baseline with a sub-word-level encoder on German-English and Czech-English language pairs. They also showed similar performance in Finnish-English and Russian-English. In [41], Gupta et al. investigated the effects of character-based input and output on the transformer architecture. They showed that character-based transformer models outperformed their BPE counterparts during translating noisy or out-of-domain text on German-English language pairs. A Char Transformer model was developed by Banar et al. [42], aiming to speed up the training on a single Graphics Processing Unit (GPU) while maintaining the accuracy of the baseline Transformer.

Nevertheless, for all of this work, English has been used in combination with a European language. Moreover, character-level neural machine translation research has not been conducted to our knowledge and awareness of the Indian languages. Hence, in our experiment, we focus on the Hindi-English language pair. We augment the training data using the technique given by Gupta et al. [17] and back-translate the extracted phrases using the built-in transformer library available as a Python package. We apply character-level encoding on the original translation task.

3 Motivation, Objectives and Common approaches

The current status of machine translation research for low-resource languages serves as the inspiration for the work done in this paper. Also, we aim to study the impact of applying character embeddings to improve the accuracy of the translation.

Parallel training samples and domain-specific corpora are unavailable in low-resource language scenarios. All Indian languages, including Hindi, fall under this category, even though many corpora have been created for it. Due to the stark differences between the quantity of parallel training data available for Hindi and the majority of European languages, we try to tackle the data scarcity issues by using the data augmentation strategy given by Gupta et al [17] and further explore with character-level encoding applied for the translation task. The literature shows that syntactic information generated (data augmentation) using syntactic phrasal segments using NMT might enhance the performance. This can be especially advantageous in resource-poor languages, where there is a limited amount of parallel data and a short vocabulary. The following points provide the rationale:

i. **Improving model robustness and generalization:** Syntactic phrasal segments can supplement the training data, giving the model additional instances of sentence structures and phrasing. This can lessen the likelihood that the model will overfit the sparse training data and increase the model's robustness and generalization capability.

ii. **Increasing the size and diversify the training data:** The training data can be made more extensive and diverse by adding syntactic phrasal segments, which may result in a more thorough learning process. Also, the additional diversity may aid the model's ability to handle more difficult translation tasks.
iii. **Improving Translation Quality and Diversity:** The additional training data can improve translations with more variety by giving the model a wider choice of sentence structures and phrasing. When there is a lack of original training data, this can be extremely helpful.
iv. **Introducing Linguistic Knowledge to the Model:** Syntactic phrasal segments can provide the model with linguistic information, enabling it to learn the syntax and grammar of the source and target languages for more accurate and high-quality translations.

In addition, various reasons favour character-level encoding for machine translation over the more traditional word-level encoding, according to a number of research that has looked into this strategy:

i. **Handling Out-of-Vocabulary (OOV) words**: The model can handle unknown words that are absent from the training data using the character level since it can build the representation of a word based on the characters[40].
ii. **Handling low-resource languages**: For low-resource languages, where there is a limited quantity of parallel data and a short vocabulary, character-level encoding can be especially helpful [43].
iii. **Handling morphological variations**: Separating the words into their constituent characters to manage morphological variations. As demonstrated by Sennrich et al. [33], this can be helpful for languages with rich morphological features, like Arabic and Hebrew. They discovered that character-level encoding can enhance the model's capacity to handle morphological variations.
iv. **Handling noisy data**: Models trained on characters can construct representations of words based on their underlying characters rather than relying on precise spelling, making them robust to spelling mistakes and other sorts of noise in the input data. [44].
v. **Handling languages with no spaces or no clear word boundaries**: This encoding method can handle languages like Chinese and Japanese that don't have distinct word boundaries or spaces between words. [45].

As outlined in Sect. 2 earlier, NMT can combine character-level encoding with various state-of-the-art translation models to represent the input and output sequences. In brief, there are the following popular methods that have been proposed so far:

i. **Byte Pair Encoding (BPE)**: BPE [33] is a popular sub-word encoding technique to segments words into smaller sub-word units based on their frequency in the training data. BPE is widely used in NMT because it can efficiently capture morphological and semantic information from the input and output sequences.
ii. **Character-level Convolutional Neural Networks (CNN)**: CNNs [43] can be applied to learn characters that represent the input and output sequences. This can assist the NMT model in capturing finer-grained data, such as spelling changes and inflections, word-level embeddings might not be able to do.
iii. **Recurrent Neural Networks (RNN) with character-level embeddings**:The character sequences in the input and output sequences can be modelled using RNNs [5, 7]. This method feeds an RNN model with a learned embedding vector for each character as its representation.
iv. **Convolutional Sequence-to-Sequence (ConvS2S)**: ConvS2S is a sequence-to-sequence model that use convolutional neural networks to encode the input sequence at the character

level [2, 46]. It has been demonstrated that this method works well for managing lengthy sequences and obtaining morphological data.

v. **Transformer models with character-level embeddings**: Character-level embeddings can be utilised with transformer models to express the input and output sequences [12, 40, 42]. There is less reliance on pre-trained word embeddings thanks to this method, which has been proven to handle OOV words.

The type of input and output sequences, the training corpus size, and the target language's difficulty each play a role in the selection of the encoding method. In addition, using Transformer models for NMT can result in better translation quality, quicker training and inference, better handling of large sequences, and more flexibility in managing variable-length input and output sequences. Due to mentioned numerous advantages, this study uses the transformer design with character-level embeddings.

4 Methodology

This section describes the methods used for data augmentation and translation in this study. The first part of the study describes the encoder and decoder architecture of NMT, and the second part describes the Google transformer for machine translation.

4.1 Encoder-Decoder Architecture of NMT

A recurrent neural network(RNN) is used to develop neural machine translation, which comprises an encoder and a decoder[47]. An RNN can handle a sequence of data, such as text, generally a sequence of words. The encoder-decoder architecture is depicted in Fig. 1. The fundamental benefit of RNN is that it considers the unlimited history of prior observations. It stores crucial sentence information, including its structure, in a memory unit called the internal state, updated at each time step. For instance, if the statement has five words, there are five-time steps. RNNs fall into two main categories: Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The structure of the memory unit is where they diverge most from one another [48]. This model was used in various NLP applications, including text classification, translation, speech recognition, question-answer systems, etc. However, the encoder and decoder network have trouble processing lengthy phrases since it must compress all the words in the source sentence into a fixed-length vector. The attention mechanism presented as a new method to address the issue of long sentences allows "soft" segments by automatically searching for source phrase components pertinent to the anticipated target word [1, 7]. The source sentence $X = (x_1, ..., x_{T_x})$ of length T_x is used by neural machine translation to create the target sentence $Y = (y_1, ..., y_{T_y})$ of length T_y, where x_t and y_t cite the lexicon entries for the source and target sentences, respectively. Context vectors are created from the input source sentence by the attention model's encoder [49].

$$C = h_1, h_2....., h_{T_x}$$

Recurrent neural networks operating in both directions, forward and backward, produce this context set. The forward RNN's goal is to read the source input from the first token to the final one [48], which produces the forward context vectors $\overrightarrow{h_1}, \overrightarrow{h_{T_x}}$ in this case, where

$$\overrightarrow{h_t} = \overrightarrow{\psi_{enc}}(\overrightarrow{h_{t-1}}, E_x[x_t])$$

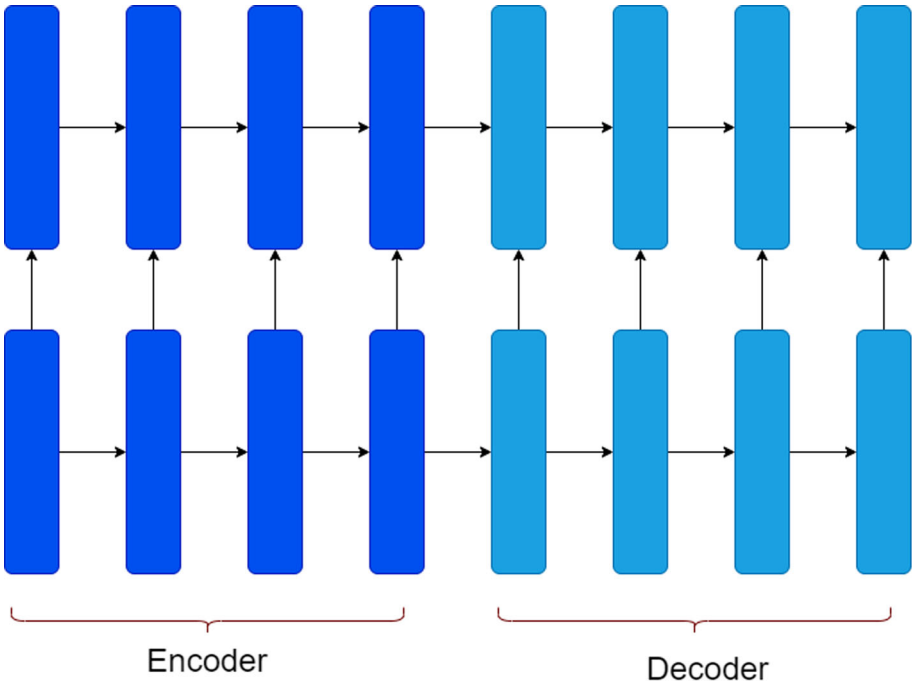

Fig. 1 Enocoder-Decoder Architecture

$E_x \in R^{|V_x| \times d}$ this is the embedding matrix to contains row vectors of source symbols [49]. The reversing RNN is the opposite direction $\overleftarrow{h_1}, \overleftarrow{h_{T_x}}$ where

$$\overleftarrow{h_t} = \overleftarrow{\psi_{enc}}(\overleftarrow{h_{t+1}}, E_x[x_t])$$

Both $\overrightarrow{\psi_{enc}}, \overleftarrow{\psi_{enc}}$ refer to recurrent activation functions such as gated recurrent units (GRU) or long short-term memory unit (LSTM) [49]. Whole context vectors are created for each place in the source phrases by concatenating the forward and reverse context vectors. For example,

$$h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right]$$

Based on the context vectors in the encoder has returned, the decoder generates one symbol at a time [49].

4.2 Google's Transformer Model

To eliminate the RNNs, and CNNs models and replace them with attention-only models, transformer neural network architectures were introduced in the paper [18]. Transformers would allow for more parallelization during training, making it much faster and more accurate. The key idea in Transformers is the attention mechanism. The evaluation of the connection between each input and each outcome is what attention is. Next is self-attention, which involves assessing how each input item relates to another. It explains how one word in a sentence relates to other words in the same sentence. Multi-headed attention is used frequently

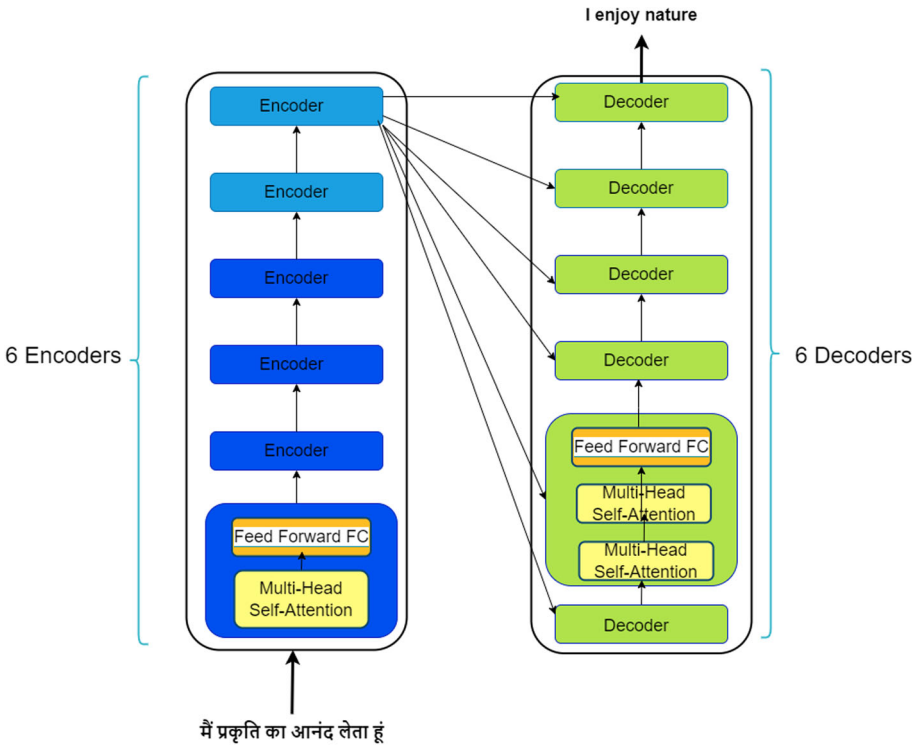

Fig. 2 Transformer Architecture

in transformer networks. Each head independently learns attention relationships. Transformers is an encoder- and decoder-based sequence-to-sequence architecture as shown in Fig 2. The encoders and decoders operate very similarly to RNNs. It employs a number of stacked multi-headed attention modules. The encoder assesses each word's level of importance to the other words in the phrase to be translated. The decoder captures both language interactions, such as those between Hindi and English. We obtain the connections between the terms in the two languages. The decoder predicts the following word generally, and we carry out this prediction through a number of time steps until the end of the phrase. These encoders and decoders are linked by feed-forward layers, which turn the outputs into forms of encoder or decoder block can process. Using RNNs, input sequences are sent simultaneously rather than one at a time, which is significantly faster. For instance, in natural language processing, transformers learn the entirety of a sentence, while RNNs can only learn one word at a time.

5 Proposed Work

This section discusses the proposed work to address the issues identified in Sect. 1 and 3 regarding implementing character-level NMT for Hindi to English. Literature shows that syntactic parse trees are a useful source of information for data augmentation [50].In the past, various methods were used to include this linguistic knowledge in the machine translation

model. Inducing the phrases taken from the constituency tree is one of the powerful strategies. The following phrases were used for augmentation in the past:

i. **Noun phrases**: A noun with any modifiers, such as adjectives or articles, make up a noun phrase. These words frequently contain crucial semantic information and can be used to add more interesting expressions to the training data.
ii. **Verb phrases**: A verb with any modifiers, like adverbs or prepositional phrases, are considered verb phrases. The model can pick up on these expressions as it learns how to deal with various sentence structures, verb tenses, and aspects.
iii. **Prepositional phrases**: Prepositional phrases are composed of a noun phrase or pronoun and a preposition which can help the model understand how to use these expressions to manage various prepositions and their meanings and the proper case of pronouns.

Apart from all these primary phrases, other phrases can be considered potentially good candidates for augmenting a dataset and making it further comprehensive and a cover-all approach. Therefore, we don't eliminate the Adjective and Adverb Phrases, which are extracted from the constituency tree of the target side language(English). Later, we back-translate those phrases and add them to our source language(Hindi). We explain the reason for selecting these two additional phrases in our study for data augmentation as follows.

Adjective Phrases: An adjective phrase is a syntactic phrase that consists of an adjective and any words that modify or describe that adjective. The modifiers may include adverbs, prepositional phrases, and other adjectives. For example, in the sentence "The big, brown dog barked loudly," the phrase "big, brown" is an adjective phrase that modifies the noun "dog," and the adverb "loudly" modifies the verb "barked."

Adjective phrases are used as a component of the data augmentation technique in neural machine translation to aid the model's ability to learn how to deal with various modifier types and their placement in sentences. This is crucial because different languages have different rules for the placement of adjectives and other modifiers, which can change how a statement is understood.

During training, the model can learn to recognize and produce acceptable adjective phrases in the target language by being exposed to a wide variety of adjective phrases. Particularly for sentences containing several adjectives or complex modifier structures, this can aid in improving the fluency and accuracy of the translations generated by the model. Adjective phrases can be generated for a wide range of nouns and adjectives, which may only appear occasionally in the training data. This may help the model better generalize to new, unseen examples and improve the overall quality of the translations.

Adverb Phrases: An adverb and any words alter or describe it make up an adverb phrase, a particular sort of syntactic phrase. Other adverbs, prepositional phrases, and other types of phrases can be used as modifiers. For example, in the sentence "She spoke softly to the audience," the phrase "softly" is an adverb phrase that modifies the verb "spoke," and the prepositional phrase "to the audience" modifies the verb as well.

Using boosting context modelling, strengthening fluency and naturalness, enabling improved word ordering, and addressing adverb-specific translation issues, adding adverb phrases taken from a language's constituency tree to the initial training data can improve the model's performance. Integrating adverb phrases, the model gains the ability to produce more accurate and natural-sounding translations and use suitable adverbs because of the emphasis on adverb phrases given during training.

After extracting all the phrases of interest from parse trees, we perform back-translation. We use Facebook's pre-trained mBART model.[1] mBART-50 is a multilingual Sequence-to-

[1] https://huggingface.co/docs/transformers/model_doc/mbart

Table 1 Back-translated phrases of Fig. 3

Extracted English Phrase	Corresponding Hindi Translation
They'll	वे
the things	चीज़ें
never be perfect but well be able to fix the things	कभी भी पूर्ण नहीं हो सकता लेकिन अच्छी तरह से चीजों को ठीक करने में सक्षम होना
be perfect but well be able to fix the things	पूर्ण होना लेकिन अच्छी तरह से चीजों को ठीक करने में सक्षम होना
be perfect	पूर्ण होना
be able to fix the things	चीज़ों को ठीक करने में सक्षम होना
to fix the things	चीज़ों को ठीक करने के लिए
fix the things	चीजों को ठीक करना
never	कभी नहीं
perfect	पूर्ण

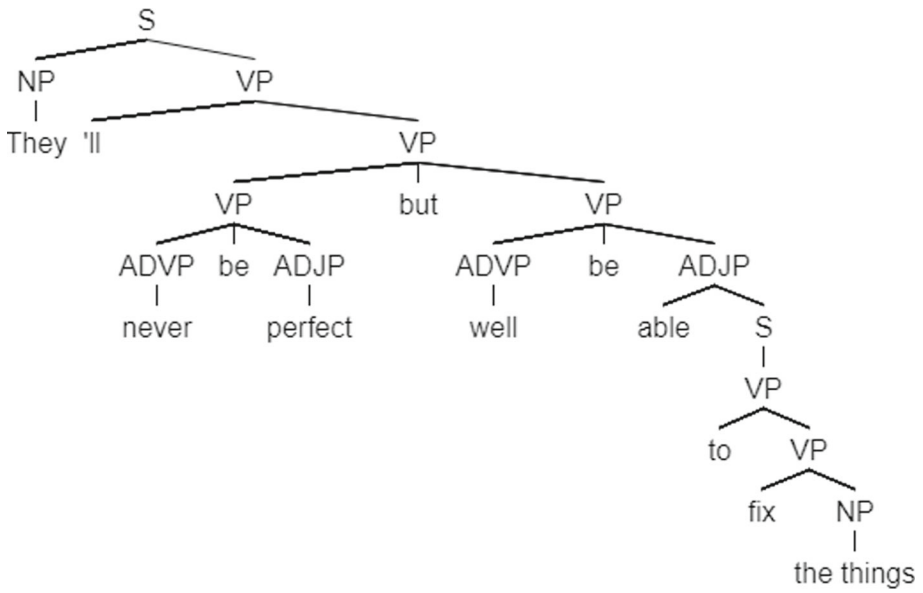

Fig. 3 Constituency parse tree with Adjective and Adverb Phrases

Sequence model. Its introduction aimed to demonstrate how multilingual fine-tuning could be used to develop multilingual translation models. A pre-trained model is tweaked simultaneously in several ways instead of only one. To accommodate multilingual machine translation models of 50 languages, the mBART-50 model was developed utilizing the original mBART model and enhanced to incorporate additional 25 languages. We obtain the corresponding Hindi translations and use them in the parallel corpus to generate the final augmented dataset.

Table 1 shows the extracted phrases and the corresponding translations obtained from the pre-trained mBART language model for the sentence structure shown in Fig. 3.

In Table 2, we show the back-translated Hindi phrases of the corresponding English ones for the example sentence shown in the constituency tree taken from the WMT14 corpus itself. The complete sentence goes like this, "This is India, and of course, it is all the more striking."

Table 2 Back-translated phrases of Fig. 4

Extracted English Phrase	Corresponding Hindi Translation
this	यह
India	भारत
of course its all the more striking	निश्चय ही इससे भी अधिक उल्लेखनीय है
course	तालीम
its all the more striking	इससे भी अधिक उल्लेखनीय है
all the more striking	और भी आश्चर्यजनक
all the	सभी
is India and of course it is all the more striking	भारत है और निश्चित रूप से यह और भी अधिक उल्लेखनीय है
of course	अलबत्ता
striking	उल्लेखनीय

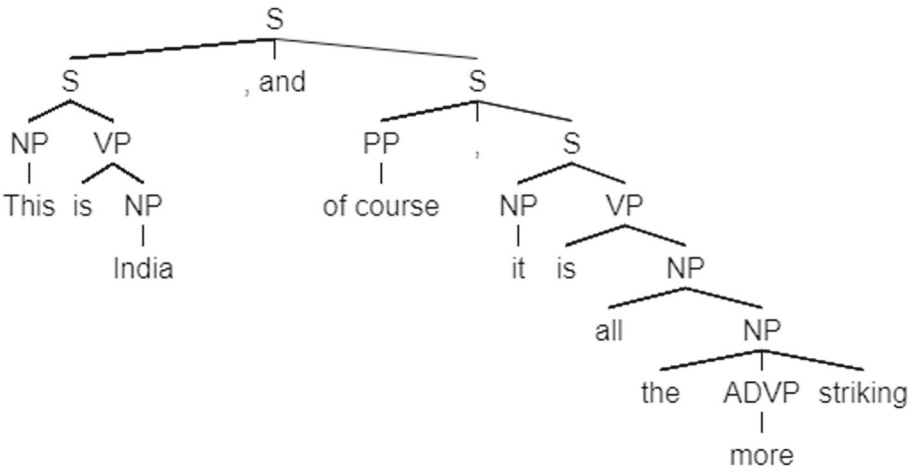

Fig. 4 Constituency parse tree with Adverb phrases

We only get one adverb phrase from this sentence. Figure 4 depicts the constituency tree obtained for this sentence. Including these syntactic phrasal segments for data augmentation in the proposed model improves the model's diversity, fluency, and performance, and reduces sparsity issues. It can be a valuable technique for improving the accuracy and naturalness of machine translation.

We conduct two separate experiments to assess how well the model performs with this augmentation technique, combined with the effect of character-level machine translation. We use this augmentation technique for the first experiment on the WMT2014 Hindi-English dataset [19] and apply character-level encoding for the forward translation.

In another experiment, we use the Samanantar Hindi-English dataset [20], however, without any augmentation. In Figure.5, we show the architecture of the proposed work-cum-flowchart for the WMT14 Hindi to English translation task, and Figure.6 shows the corresponding steps involved in the task for the Samanantar dataset.

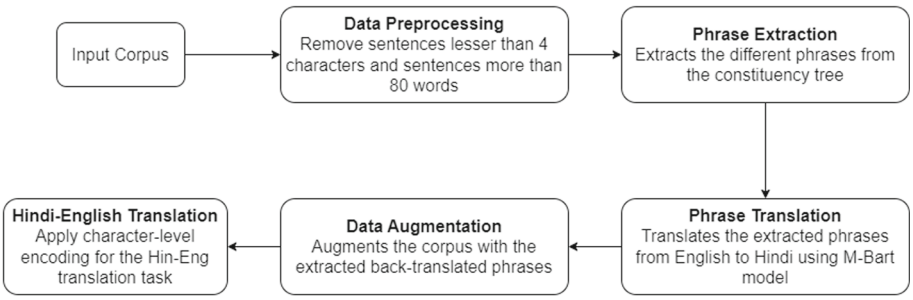

Fig. 5 Flowchart for the WMT14 Translation task

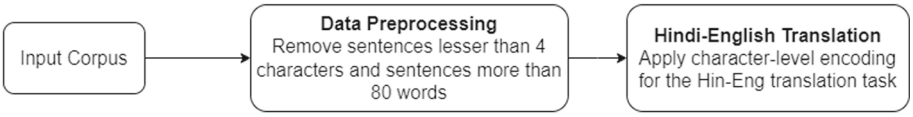

Fig. 6 Flowchart for the Samanantar Translation task

Table 3 Corpus Statistics

	WMT14		**Samanantar**	
	Train Set	Test Set	Train Set	Test Set
Sentences	307044	2507	238458	2501
Characters(Hindi)	17296521	237482	22250502	240970
Characters(English)	18977683	245477	23367193	248641
Unique Characters	162	144	168	153
Total characters	**36274204**	**482959**	**45617695**	**489611**

Bold represents the total characters in different sets of WMT14 and Samanantar datasets

6 Datasets

This section explains the datasets that were used in our experiments. The corpus statistics are outlined in Table.3. For the first task, i.e. we use the WMT 2014 Hindi-English dataset [19]. This is the dataset on which the augmentation strategy described earlier has been used. After the data has been augmented using the back-translated extracted phrases, the model is trained at the character level to obtain the translation results.

As part of another subtask for the experiment, we use the Samanantar Hindi-English corpus. We provide the training and test set sentence pairs for each task, as well as the number of characters trained on the model. The train and test set sentences for the WMT2014 dataset are taken from the standard WMT2014 sets. For the Samanantar dataset, the train and test set sentences are taken randomly from the corpus without augmentation. Character-level machine translation is applied during the training phase of the model.

7 Experiment and Results

This section describes the evaluation process of results based on the proposal given in section 5, experimental setup, experimental evaluation and analysis of data augmentation and translation.

7.1 Experimental Setup

The proposed experiment uses the state-of-the-art NMT model, the Google Transformer, given in the paper [18] for translating Hindi to English. As for the back-translation of the extracted phrases, we use the pre-trained M-Bart model provided by Facebook and publicly available at Huggingface. Since the model is trained at the character level, we don't need to segment or tokenize the input sentences. We true case all the sentences and also remove those with a number of characters less than four or where the number of words is more than 80. The learning rate is kept at 0.0006. We use the Adam optimizer, and the number of encoder-decoder layers in the transformer model is 6X6, with eight attention heads.

7.2 Experimental Evaluation and Discussion

We carry out the proposed experiment on two different datasets, with a slight variation in approach for both. On the WMT14 Hindi-English [19] dataset, we augmented the training samples using the linguistically motivated phrase-based data augmentation technique. The authentic-to-synthetic training samples ratio is kept at 1:4 as it is the most effective in the literature.

Further, we used the Samanantar dataset [20] without any augmentation and character-level encoding. Table 4 shows the results obtained for both datasets. The BLEU-4, the 4-gram BLEU score, is widely regarded as the gold standard for evaluating machine translation systems. And the weights for the BLEU-4 are 1/4 (25 percent) or 0.25 for each of the 1-gram, 2-gram, 3-gram, and 4-gram scores. However, we also reported the BLEU scores for each 1, 2, and 3-gram, or BLEU-1, BLEU-2, and BLEU-3.

As Table 5 makes evident, the results obtained on the WMT14 Hin-Eng translation task show the model has achieved an improvement in BLEU score(BLEU-4) of +2.52 over the baseline state-of-the-art system using character level encoding. Further, it reports the BLEU Score(BLEU-4) of +2.68 using five phrasal sentences with character-level encoding. We receive similar benchmark results in the Samanantar dataset, on which no augmentation strategy has been applied and is directly trained on the character-level encoder. The model reports a BLEU score of 16.21 on this dataset. This suggests that character embeddings can potentially improve translation accuracy for low-resource and morphologically rich languages like Hindi.

7.3 Analysis of the translation

This section analyses the translations produced by the MT system created using the proposed model. The source Hindi sentences for this analysis came directly from the test sets of the corresponding corpus. Table 6 shows the outputs of the translations produced by the proposed MT system over the Samanantar dataset. We can observe that the trained model is performing reasonably well, and the model can also produce synonyms for the translation when we

Table 4 BLEU scores of the model

Dataset	WMT14	Samanantar
BLEU-1	44.58	47.94
BLEU-2	31.32	34.84
BLEU-3	23.38	26.31
BLEU-4	**17.85**	**19.65**

Bold represents the result of the proposed method in terms of BLEU-4 metric

Table 5 Comparison of proposed work with state-of-the-art method on WMT14 Dataset

Model/Method	BLEU Score
Translation Data Augmentation (TDA) 2017 [28]	13.05
Soft Contextual Data Augmentation NMT 2019 [51]	14.19
Google Transformer NMT 2021(Base) [17]	15.17
Base + CharT(proposed)	**17.69**
Base + Aug(5 Phrases) + CharT(proposed)	**17.85**

Bold represents the name and performance of the proposed method. Non-bold entries represent other works from recent literature

Table 6 Sample output on the Samanantar corpus

Category	Translation
Source	तेलंगाना में सामने आए कोविड-19 के 42 नए पॉजिटिव मामले
Transliteration	telangaana mein saamane aae kovid-19 ke 42 nae pojitiv maamale
Reference Translation	42 fresh positive cases of COVID-19 in Telangana
CharT	42 new COVID19 positive cases reported in Telangana

compare it with the reference translation. It is because character-level models operate more granularly, considering individual characters rather than entire words.

In another example, as shown in Table 7, we observe that the resulting translation has the unaffected quality and slightly deviates from the reference translation. This sentence is taken from the WMT2014 test dataset. The result is obtained after augmenting the model with three syntactic phrases and training on a character basis. The proposed model performs reasonably well using five phrases for augmentation, as shown in Table 8. We also analyze an evaluation done against Google Translate. The model can perform well for a very small sentence and produce an accurate translation over a benchmark Google Translate model. Table 9 shows the results obtained.

7.4 Failure Cases

This section presents some failure cases from the trained model. Even though the proposed technique has accomplished great success in translation tasks, it is not immune to failure situations where the translated texts fall short of expectations. These instances of failure may be subpar translations, unreliable renderings, or grammatical mistakes that make it difficult to understand and transmit the correct meaning. An example of a failure case is depicted in

Table 7 Augmentation using three phrases on WMT14 dataset

Category	Translation
Source	Frontier के डेनवर में अपने घर के शहर में ग्राहकों के एक वफादार बेस है, मगर इसका व्यापार कम होता जा रहा है और यह पैसा खो रहा है.
Transliteration	frontiair ke denavar mein apane ghar ke shahar mein graahakon ke ek vaphaadaar bes hai, magar isaka vyaapaar kam hota ja raha hai aur yah paisa kho raha hai.
Reference	Frontier has a loyal base of customers in its home city of Denver, but its business is shrinking and losing money.
CharT + Aug(3 Phrases)	Frontier's denver has a loyal base for customers in his own city, but its business is losing money.

Table 8 Augmentation using five phrases on WMT14 dataset

Category	Translation
Source	जोधपुर शहर को जन्मदिन की शाही पार्टी के लिए चुना गया है।
Transliteration	jodhapur shahar ko janmadin kee shaahee paartee ke lie chuna gaya hai.
Reference	Jodhpur city was selected for the spectacular birthday party.
CharT + Aug(5 Phrases)	Jodhpur city was selected for the royal party of birthday.

Table 9 A representative example of translation of a very small sentence by the proposed model and Google Translate

Category	Translation
Source	सबसे ऊपर हो तुम
Transliteration	sabase oopar ho tum
Google Translate	you are the top
CharT+Aug(5 Phrases)	Above all you're

Table 10, taken from the Samanantar corpus. The reason behind this incoherent translation output could be under-fitting from the training data.

Moreover, as we can see, the reference translation is also not close to the original sentence. It is also possible because of the use of special characters and quotation marks, which is slightly uncommon. The model did not find sufficient exposure to such training samples, which resulted in its inability to capture the intricacies of the translation problem. Having trouble learning from training data, the model may oversimplify the translation process, producing erroneous or too generalized translations.

Another example of a failure case arising out of the model is depicted in Table 11. The resulting translation is almost accurate regarding closeness as far as the synonyms are concerned. But it needs to be a correct translation. The model tries to blend various words without a proper associated meaning. However, there are a number of possibilities behind this type of model behaviour, like- training data with spelling errors, noise and ambiguity in the input, data quality and pre-processing, etc. The most prominent one is the quality of the training data. The HindEnCorp is full of training samples with code-mixed sentences, some erroneous and mistranslations, or samples in an entirely different language. This poses a considerable problem to the model while testing. A small but quality dataset can improve performance and accuracy over a large but poor-quality dataset.

Table 10 Failure case on Samanantar corpus

Category	Translation
Source	शिमला की मंडियों में प्याज आज 80 रुपए किलो बिका जबकि पिछले हफ्ते इसका दाम 60 रुपए किलो था।
Transliteration	shimala kee mandiyon mein pyaaj aaj 80 rupe kilo bika jabaki pichhale haphte isaka daam 60 rupe kilo tha.
Reference Translation	"""Onion has touched Rs 100 per kg while tomato is being sold at Rs 60,""""" she said.""""
Model output	Onion's markets were sold at Rs 80 while the price of Rs 60 lakh was Rs 60 per kilogram.

Table 11 Failure case on WMT14 corpus

Category	Translation
Source	कैट एक तरह से मजबूती और लाइफ स्टाइल का मिलन है।
Transliteration	kait ek tarah se majabootee aur laiph stail ka milan hai.
Reference Translation	CAT, in a way, is the blend of durability and lifestyle.
Model Output	CAT is a kind of strong and lifestyle.

8 Conclusion and Future Scope

A large training corpus is needed to train an NMT system properly. For many language pairs and domains, such a corpus is not easily accessible. In this paper, we tried to explore the low-resource machine translation scenario for Hindi language and improve the model performance using a straightforward phrase-based augmentation technique extracted from the constituency tree of the target language - English.

The model is trained at the character level to overcome some of the key limitations of a word-level model, as discussed in sections 2 and 3. This results in a highly well-performing neural machine translation model and has worked significantly well, especially for low-resource language pair. The proposed trained model can surpass the cutting-edge baseline and saw an increase in BLEU scores for the WMT14 translation challenge with +2.52 on base paper using three phrase sentences with character level encoding and +2.68 BLEU Score on base paper using five phrase sentences with character level encoding. Further, Character level encoding is evaluated on non augmented samananter dataset; it performs better in the baseline approach for translation purposes. It clearly shows that the proposed model outperforms for Hindi translation. The following points may be considered for future work:

- Character-level models analyze each input token separately, making it challenging to identify long-distance relationships between characters that are separated by a large number of locations in the input sequence. Using a hybrid strategy that blends word-level and character-level representations is one potential solution that may be explored to improve the proposed approach.
- In comparison to word-level models, character-level models require significanly more processing & resources because of the longer input sequences. As a result, training sessions are longer and need more memory. Enhancement of model's training efficiency and reduction in its computing complexity may be a good direction to explore further.

– Homographs are words with the exact same spelling but different meanings. The precise definition of homographs may be complex for character-level models to determine based only on the input characters. Contextual cues like the words around them or the overall structure of the sentence, which might help clarify the intended meaning, may be a possible direction to further improve the proposed model.
– Word-level models are frequently considered more interpretable than character-level models. Future research could look into ways to visualize learned representations or include attention processes to make character-level models more interpretable.
– Other low-resource languages from various language families may also be incorporated for use along with the proposed model. To see how well the proposed approach performs in these situations, some of the highly specialized & restricted domains may also be considered.

Funding The authors do not have any funding information to declare.

Declarations

Conflict of interest The authors do not have any conflict of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473
2. Chen MX, Firat O, Bapna A, Johnson M, Macherey W, Foster G, Jones L, Parmar N, Schuster M, Chen Z et al. (2018) The best of both worlds: Combining recent advances in neural machine translation. arXiv preprint arXiv:1804.09849
3. Mistry JG, Verma A, Bhattacharyya P (2017) Literature survey: study of neural machine translation. Resource Centre for Indian Language Technology Solutions (CFILT)
4. Wu Y, Schuster M, Chen Z, Le QV, Mohammad N, Wolfgang M, Maxim K, Yuan C, Qin G, Klaus M et al. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144
5. Su J, Tan Z, Xiong D, Ji R, Shi X, Liu Y (2017) Lattice-based recurrent neural network encoders for neural machine translation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31
6. Greenstein E (2015) Japanese-to-english machine translation using recurrent neural networks
7. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
8. Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. Technical report, University of Southern California Marina Del Rey Information Sciences Inst
9. Salam MS, Thoudam DS (2022) Low resource machine translation of english-manipuri: a semi-supervised approach. Expert Syst Appl 209:118187

10. Chauhan S, Saxena S, Daniel P (2022) Improved unsupervised neural machine translation with semantically weighted back translation for morphologically rich and low resource languages. Neural Process Lett 54(3):1707–1726
11. Läubli S, Sennrich R, Volk M (2018) Has machine translation achieved human parity? a case for document-level evaluation. Science 11:852
12. Zhang L, Komachi M (2021) Using sub-character level information for neural machine translation of logographic languages. Trans Asian Low-Resource Lang Inf Process 20(2):1–15
13. Escribe M (2019) Human evaluation of neural machine translation: The case of deep learning. In: Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019), pp 36–46
14. Bawa-Mason S, Bywood L, Gittins C, Kaye P, McNab R, Olohan M, Wells M (2018) Translators in the digital era: What kind of jobs will we have ten years from now. Language Show, Olympia, London, UK, vol 11
15. Koehn P, Knowles R (2017) Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872
16. Sennrich R, Haddow B, Birch A (2015) Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709
17. Gupta KK, Sen S, Haque R, Ekbal A, Bhattacharyya P, Way A (2021) Augmenting training data with syntactic phrasal-segments in low-resource neural machine translation. Mach Transl 35(4):661–685
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:52
19. Bojar O, Diatka V, Rychlỳ P, Straňák P, Suchomel, V, Tamchyna A, Zeman D (2014) Hindencorp-hindi-english and hindi-only corpus for machine translation. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pp 3550–3555
20. Ramesh G, Doddapaneni S, Bheemaraj A, Jobanputra M, Raghavan AK, Sharma A, Sahoo S, Diddee H, Mahalakshmi J, Kakwani D, Kumar N, Pradeep A, Nagaraj S, Deepak K, Raghavan V, Kunchukuttan A, Kumar P, Khapra MS (2021) Samanantar: The largest publicly available parallel corpora collection for 11 indic languages
21. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp 311–318
22. Gao F, Zhu J, Wu L, Xia Y, Qin T, Cheng X, Zhou W, Liu T-Y (2019) Soft contextual data augmentation for neural machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5539–5544, Florence, Italy. Association for Computational Linguistics
23. Currey A, Miceli-Barone AV, Heafield K (2017) Copied monolingual data improves low-resource neural machine translation. In: Proceedings of the second conference on machine translation, pp 148–156
24. Fadaee M, Monz C (2018) Back-translation sampling by targeting difficult words in neural machine translation. arXiv preprint arXiv:1808.09006
25. Gulcehre C, Firat O, Xu K, Cho K, Barrault L, Lin H-C, Bougares F, Schwenk H, Bengio Y (2015) On using monolingual corpora in neural machine translation. arXiv preprint arXiv:1503.03535
26. Zhang J, Zong C (2016) Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 1535–1545
27. Nguyen TQ, Chiang D (2017) Transfer learning across low-resource, related languages for neural machine translation. arXiv preprint arXiv:1708.09803
28. Fadaee M, Bisazza A, Monz C (2017) Data augmentation for low-resource neural machine translation. arXiv preprint arXiv:1705.00440
29. Wang X, Pham H, Dai Z, Neubig G (2018) Switchout: an efficient data augmentation algorithm for neural machine translation. arXiv preprint arXiv:1808.07512
30. Yanbin Zhao L, Chen ZC, Kai Yu (2020) Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. Proc AAAI Conf Artif Intell 34:9668–9675
31. Shah H, Barber D (2018) Generative neural machine translation. Adv Neural Inf Process Syst 31:524
32. Kudo T, Richardson J (2018) Sentencepiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. CoRR, abs/1808.06226
33. Sennrich R, Haddow B, Birch A (2015) Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909
34. Luong M-T, Manning CD (2016) Achieving open vocabulary neural machine translation with hybrid word-character models. arXiv preprint arXiv:1604.00788

35. Cherry C, Foster GF, Bapna A, Firat O, Macherey W (2018) Revisiting character-based neural machine translation with capacity and compression. In: Conference on Empirical Methods in Natural Language Processing
36. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. Adv Neural Inf Process Syst 28:96
37. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Thirtieth AAAI conference on artificial intelligence
38. Costa-Jussa MR, Fonollosa JAR (2016) Character-based neural machine translation. arXiv preprint arXiv:1603.00810
39. Chung J, Cho K, Bengio Y (2016) A character-level decoder without explicit segmentation for neural machine translation. arXiv preprint arXiv:1603.06147
40. Lee J, Cho K, Hofmann T (2017) Fully character-level neural machine translation without explicit segmentation. Trans Assoc Comput Ling 5:365–378
41. Gupta R, Besacier L, Dymetman M, Gallé M (2019) Character-based nmt with transformer. arXiv preprint arXiv:1911.04997
42. Banar N, Daelemans W, Kestemont M (2020) Character-level transformer-based neural machine translation. In: Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, pp 149–156
43. Almansor EH, Al-Ani A (2018) A hybrid neural machine translation technique for translating low resource languages. In: Machine Learning and Data Mining in Pattern Recognition: 14th International Conference, MLDM 2018, New York, NY, USA, July 15-19, 2018, Proceedings, Part II 14, pp 347–356. Springer
44. Karpukhin V, Levy O, Eisenstein J, Ghazvininejad M (2019) Training on synthetic noise improves robustness to natural noise in machine translation. arXiv preprint arXiv:1902.01509
45. Zhang L, Komachi M (2018) Neural machine translation of logographic language using sub-character level information. In: Proceedings of the Third Conference on Machine Translation: Research Papers
46. Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN (2017) Convolutional sequence to sequence learning. In: International conference on machine learning, pp 1243–1252. PMLR
47. Srivastava RK, Greff K, Schmidhuber J (2015) Training very deep networks. Adv Neural Inf Process Syst 28:241
48. He D, Xia Y, Qin T, Wang L, Yu N, Liu T-Y, Ma W-Y (2016) Dual learning for machine translation. Adv Neural Inf Process Syst 29:758
49. Firat O, Cho K, Bengio Y (2016) Multi-way, multilingual neural machine translation with a shared attention mechanism. arXiv preprint arXiv:1601.01073
50. Tan Z, Wang S, Yang Z, Chen G, Huang X, Sun M, Liu Y (2020) Neural machine translation: a review of methods, resources, and tools. AI Open 1:5–21
51. Gao F, Zhu J, Wu L, Xia Y, Qin T, Cheng X, Zhou W, Liu T-Y (2019) Soft contextual data augmentation for neural machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp 5539–5544
52. Lin Z, Feng M, Santos CN dos, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.