N Donald Jefferson Thabah[1*]
Sonithoi Ningombam[2]
Saptarishi Paul[3]
Bipul Syam Purkayastha[4]

An Enhanced Neural Machine Translation with Pre-Trained Contextual Encoding Knowledge and Data Augmentation for Low-Resource Khasi Language

Journal of Electrical Systems

Abstract: - This paper explores the potential of neural machine translation (NMT) in developing a translation system for low resource languages. The method utilised in this study builds upon the transformer framework by incorporating substantial enhancements to improve the translation. The modifications include data augmentation on the input sentence, initialization of embedding layer weights with pre-trained embedding and adding a new layer into the encoder block to fuse pre-trained context information with the local context encoding. A Khasi English language pair was considered to carry out translation in order to demonstrate our work. The translation system that is obtained exhibits an encouraging performance in terms of BLEU, METEOR, and ROUGE_L evaluation metrics. Additionally, a comparative analysis is conducted with other existing models, and our proposed model demonstrates superior performance compared to statistical machine translation (SMT) and long short-term memory networks (LSTM). Furthermore, a semantic score was calculated between reference and candidate sentences to facilitate semantic comparisons. The findings indicate that the suggested NMT approach has significant potential as an automated solution for translating low-resource languages.

Keywords: Machine Translation (MT), Low Resource MT, Khasi- English MT, Khasi.

I. INTRODUCTION

The 21st century has witnessed a pivotal communication advancement in machine translation, facilitating global interconnectivity. Neural Machine Translation (NMT), a deep learning paradigm introduced by [31] and further refined by [5], stands at the forefront of automated language translation. Despite its effectiveness, NMT has yet to encompass all languages due to the digital scarcity of certain languages, such as Khasi. While languages like English and French benefit from well-resourced translation systems, this paper addresses the low-resource challenges associated with languages like Khasi by leveraging NMT to establish translation capabilities between English and Khasi language pairs.

The Khasi community, one of Meghalaya's major tribes in northeastern India, presents a linguistic landscape characterized by intricate structures and diverse dialects. Traditionally oral and structurally complex, the Khasi language survived through narratives like myths of the land, legends, and folklore, all of which ensured sustainability in its linguistic evolution. This resilience, rooted in oral tradition, equips society to navigate life's demands and nature's intricacies.

The initial foray into English-to-Khasi translation commenced with the translation of biblical segments. Missionaries from the Serampore Mission attempted to script the Khasi language, with William Carey's 1824 translation of the New Testament into the Shella dialect using the Bengali Script. However, the complexities of the Bengali script rendered it unviable for the community [30]. The subsequent adoption of the Roman script in the 18th century, pioneered by Thomas Jones-I, facilitated the documentation of Khasi cultural facets. This script enabled the codification of customary laws, oral literature, and practices for scholarly pursuits.

Khasi, classified as a low-resource and morphologically complex language, poses translation challenges. It exhibits independent words and bound morphemes, necessitating root words for meaning. Additionally, syntactic differences between Khasi and English, like adjective placement, further make translation challenging. Khasi's socio-cultural context adds another layer of complexity, demanding comprehensive explanations of culturally embedded terms and practices.

[22] emphasizes the fluency requirement in both linguistic and literary realms for effective translation, while [13] underscores the enrichment of Khasi through its linguistic heritage rather than mere English appropriation. Understanding the translation challenges serves as a litmus test for Khasi machine translation, enhancing comprehension of its linguistic and literary legacy.

[1]* Department of Computer Science Assam University Silchar,Email: jefson08@gmail.com
[2] [2]Department of Computer Science, Assam University Silchar,Email: sonithoi.ningombam@aus.ac.in
[3] [3]Department of Computer Science Assam University Silchar, Email: paulsaptarshi@yahoo.co.in
[4] [4]Department of Computer Science, Assam University Silchar, Email: bipul.syam.purkayastha@aus.ac.in

Recent studies attest to NMT's equivalence to human translation [35],[8], particularly for well-resourced languages. However, the lack of resources hinders the accuracy of low resource languages like Khasi [39]. The transformer model, a unique NMT approach introduced by [34], holds promise in natural language processing.

Our paper delves into the investigation and experimental evaluation of an NMT system for English-to-Khasi translation. Section II reviews existing works on NMT and low-resource Indian languages. Section III offers a machine translation overview, followed by the methodology in Section IV. Corpus collection sources and tools are detailed in Section V. Sections VI, VII, and VIII cover implementation, results, and discussion. Finally, Section IX provides a concise summary of our work.

II. RELATED WORK

Advancements in self-attention techniques have notably elevated the quality of machine translation.[34] introduced the transformer model, a pioneering neural network architecture that exclusively employs attention mechanisms, supplanting the need for recurrence or convolutions. In this construct, the encoder maps the input sentence onto a hidden vector, and the decoder then transforms the hidden vector into the desired target sequence. The design's parallelizability expedites training, necessitating substantially less time than alternative methods. Notably, the model achieved a new single-model state-of the-art on the WMT 2014 English-to-French translation task, attaining a BLEU score of 41.8.

Recent research, like [38], introduced the BERT-fused model, which involves training word embedding through BERT and fusing them with each encoder/decoder transformer layer. Their experiments encompassed sentence and document level translations, demonstrating exceptional results across seven benchmark datasets.

Addressing machine translation for North East Indian languages has involved diverse statistical and neural techniques. [10] harnessed NMT models for Manipuri-to-English translation, enhancing quality through fastText-based pre-trained word embeddings. [7] explored Assamese and other Indo-Aryan languages with traditional phrase-based statistical machine translation (SMT) and advanced neural machine translation (NMT). [16] showcased their Assamese-Bengali NMT system, while [12] managed tonal words in NMT models, leveraging BERT-fused approaches for enhanced accuracy.

In the realm of Khasi translation, [33] explored supervised and unsupervised techniques for Khasi-to-English translation. Their work with unsupervised neural machine translation [3], supervised models [14] and statistical machine translation [15] achieved varying BLEU scores. [29] delved into English-Khasi translation, while [32] extended their exploration to crosslingual language models, yielding promising BLEU scores. [11] implemented MT from English to Khasi using transfer learning-based NMT and achieved better results than the baseline NMT models.

The present study aims to devise a low-resource translation system for English and Khasi language pairs, building upon the transformer model of [34] with tailored enhancements to yield improved translations.

III. BACKGROUND

The machine translation problem is a conditional probability modelling $P\left(\frac{x}{z}\right)$ task, where *z* and *x* correspond to the source and target sentences within languages *Z, X*. The objective is to maximise the probability function for identifying the best-matching target sentence, *x*, given a source sentence, *z*. Bayes' Rule articulates this translation task explicitly:

$$argmax_x P(\frac{x}{z}) \rightarrow argmax_z P(\frac{z}{x})P(x) \quad (1)$$

Here, $P\left(\frac{z}{x}\right)$ denotes the translation model, and *P(x)* represents the language model. Equation (1) forms the foundation of statistical machine translation, while NMT directly learned the translation model $P\left(\frac{z}{x}\right)$.

Two prevalent NMT techniques are recurrent neural networks (RNN) [31] and the transformer model [34]. Both are end-to-end translation systems that utilize encoder and decoder blocks to model $P\left(\frac{z}{x}\right)$. RNN follows a sequence-to-sequence architecture, while transformers rely heavily on parallelizable attention mechanisms.

IV. APPROACH

The proposed translation architecture mainly comprises encoder and decoder blocks (Fig. 1). We illustrate the architecture through an algorithm from which we draw influence from Formal Algorithms for Transformers [25], [2]. The encoder generates a contextual representation of the input sequence, which the decoder employs to learn relevant translations for target sentences. It has been observed that incorporating data augmentation techniques into

the input sequence, initialising the embedding layer with word embeddings from a pre-trained model, and appending the learned local context with pretrained contextual knowledge can significantly enhance the efficiency of learning, especially in the case of low-resource languages.

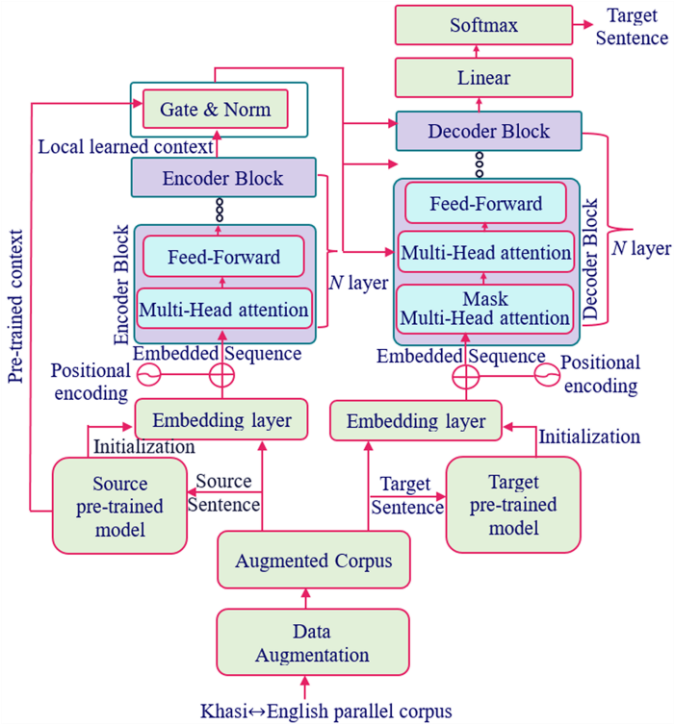

Fig. 1.Proposed translation system

Data augmentation is applied to the complete training and validation corpus [20], effectively doubling the input size. This approach aids in mitigating overfitting as augmentation is applied uniformly across the corpus, ensuring efficient data shuffling [9]. The TextAttack [19] package was used for the purpose of data augmentation. This augmentation process was exclusively applied to English sentences, based on the assumption that the Khasi language may have a smaller vocabulary compared to English. Consequently, while multiple English sentences can be augmented with different synonyms, we can expect to have only one corresponding translation in Khasi.

Local context encoding is computed using an encoder block, as detailed in Section IV-A2. Within the multi-headed attention and feed-forward sub-layer, word comparisons are performed across phrases to gauge word similarity. This fosters the discovery of word relationships, resulting in a robust contextual sentence representation.

Input sentence pre-trained context is obtained from a pretrained [18] language model, described in Section IV-A3. This context is concatenated with local contextual encoding using a gated context fusion technique [37], as detailed in Section IV-A4.

The decoder undertakes translation to the target sentence upon obtaining the combined context, as elaborated in Section IV-B. Gate context fusion is not employed on the decoder side, as the inference process functions sequentially during translation.

A. Encoder

1) Input sentence and Positional Encoding: The source sentence undergoes conversion into lexical and positional representations. Word order is represented using Transformer's word position embedding [34]. Algorithm 1 and 2 illustrates the steps for obtaining word and positional embedding. W_e are the translation model's learnable token embedding weight matrix. The positional embedding matrix W_p can be the individual token sequence in a sentence or use a mixture of *sin()* and *cos()* values, as mentioned in the original paper [34].

Algorithm 1: $e \leftarrow$ TokenEmbedding(v)
/* **Symbol Notation:** Refer Appendix A */
Input: $v \in V \cong [N_V]$, a token ID.
Output: $e \in \mathbb{R}^{d_e}$, the vector representation of the token.
Parameters: $W_e \in \mathbb{R}^{N_V \times d_e}$, the token embedding matrix.
return $e = W_e[v,:]$

Algorithm 2: $e_p \leftarrow$ PositionalEmbedding(ℓ)
/* **Symbol Notation:** Refer Appendix A */

Input: $\ell \in [\ell_{\max}]$, a position of a token in the sequence.

Output: $e_p \in \mathbb{R}^{d_e}$, the vector representation of the position.

Parameters: $W_p \in \mathbb{R}^{\ell_{\max} \times d_e}$, the positional embedding matrix.

return $e_p = W_p[\ell, :]$

2) Local context Encoding: The context embedding for each input sentence is generated through a stack of transformer layers (Fig. 1). As shown in algorithm 4, given a source sentence *z*, apply word and positional embedding to generate *Z*. *Z* represents query, key, and value, respectively, in the encoder block. The local context is computed using multi-head attention and a feed-forward layer as described in equation (2a), (2b), and (2c):

$$Z = \text{MultiHead}(\text{SelfAttention}(Q \equiv Z, K \equiv Z, V \equiv Z)) \tag{2a}$$

$$Z = \text{MultiHead}(Q + \text{LayerNorm}(\sum_n \text{Similarity}(Q, K_n) \times V_n)) \tag{2b}$$

$$Z = Z + \text{LayerNorm}(\text{FeedForward}(Z)) \tag{2c}$$

Where SelfAttention($Q \equiv Z, K \equiv Z, V \equiv Z$) in equation 2a denotes self-attention obtained by finding the similarity using a scale-dot product between a query vector Q and each key $K_0 < K_i < K_n$, where $0 < i < N$ represents the order of key vectors for words in a sentence. The resultant similarity value is then multiplied with $V_0 < V_i < V_n$ to retain the originality of the word in a sentence, where $0 < i < N$ represents the order of value vectors for words in a sentence. According to [36], [21], applying layer normalization before a sub-layer makes the model more stable for training [4], as shown in equation 2b. Equation 2c shows how to obtain the local context of a sentence by combining feed-forward with multi-head attention. The algorithmic procedures outlined in Algorithms 3 and 4 provide a comprehensive breakdown of the steps involved. Equation (3) shows how to compute the softmax of a matrix *A*. One's matrix is the *Mask* for bi-directional attention. The unidirectional attention value is $[[t_x \leq t_z]]$, which is one's lower triangular matrix used to prevent the network from seeing future values during training.

$$\text{softmax}(A)[t_x, t_z] := \frac{\exp A[t_x, t_z]}{\sum_t \exp A[t, t_z]} \tag{3}$$

$$\text{Mask}[t_x, t_z] = \begin{cases} 1 \text{ for bidirectional attention} \\ [[t_x \leq t_z]] \text{ for unidirectional attention} \end{cases} \tag{4}$$

3) Pre-trained Contextual Encoding: The pre-trained context for Khasi sentences is obtained by training a RoBERTa [18] model with monolingual corpus data. However, a pretrained masked language model from Hugging Face is employed for English. Thus, an extra layer, built on top of local context encoding layers (Fig. 1), computes the final context embedding by concatenating local and pre-trained contexts through a gated context fusion technique [37].

Algorithm 3: $\tilde{V} \leftarrow$ MHAttention(X, $Z|W$, Mask) /* Compute Multi-Head (mask) self-attention */
/* **Symbol Notation:** Refer Appendix A */

Input: $X \in \mathbb{R}^{\ell_x \times d_e}, Z \in \mathbb{R}^{\ell_z \times d_e}$, vector representation of primary and context sequence.

Output: $\tilde{V} \in \mathbb{R}^{\ell_x \times d_e}$, updated representation of tokens in *X*, folding in information from tokens in *Z*.
Hyperparameters: *H*, number of attention heads

Hyperparameters: Mask $\in \{0,1\}^{\ell_x \times \ell_z}$, Refer equation (4)
Parameters: *W* consisting of

For $h \in [H]$, W_{qkv}^h consisting of:

$W_q^h \in \mathbb{R}^{d_e \times d_{\text{attn}}}, b_q^h \in \mathbb{R}^{d_{\text{attn}}}$, $W_k^h \in \mathbb{R}^{d_e \times d_{\text{attn}}}, b_k^h \in \mathbb{R}^{d_{\text{attn}}}$, $W_v^h \in \mathbb{R}^{d_e \times d_{\text{attn}}}, b_v^h \in \mathbb{R}^{d_{\text{attn}}}$.

$W_o \in \mathbb{R}^{H d_{\text{attn}} \times d_e}, b_o \in \mathbb{R}^{d_e}$

1. For h $\in [H]$:
2. $Q^h \leftarrow XW_q^h + 1^{\mathrm{T}} b_q^h \quad Query \in \mathbb{R}^{\ell_x \times d_{\text{attn}}}$
3. $K^h \leftarrow ZW_k^h + 1^{\mathrm{T}} b_k^h \quad Key \in \mathbb{R}^{\ell_z \times d_{\text{attn}}}$
4. $V^h \leftarrow ZW_v^h + 1^{\mathrm{T}} b_v^h \quad Value \in \mathbb{R}^{\ell_z \times d_e}$

5. $S^h \leftarrow Q(K^{\mathrm{T}})^h \square Score \in \square^{\ell_x \times \ell_z} \square$
6. $\forall t_x, t_z, if \neg Mask[t_x, t_z], then\, S^h[t_x, t_z] \leftarrow -\infty$
7. $Y^h \leftarrow \mathrm{softmax}(S^h \sqrt{d_{\mathrm{attn}}}).V^h$ /* Note: $Y \leftarrow [Y^1; Y^2; ....; Y^H]$ is a concatenated list*/
8. **return** $\tilde{V} = YW_O + 1^{\mathrm{T}} b_o$

Algorithm 4: $Z \leftarrow$ Encoder(z, $P_e|\theta$) /* Encoder transformer forward pass */
/* **Symbol Notation:** Refer Appendix A */

Input: $z \in V^*$, is source token ID, $P_e \in \square^{\ell_z \times d_e}$, is a source token pre-training embedding.

Output: $Z \in \square^{\ell_z \times d_e}$.

Hyperparameters: $\ell_{\max}, L_{\mathrm{enc}}, H, d_e, d_{\mathrm{mlp}} \in \square$

Parameters: θ includes all the following parameters:

$W_e \in \square^{N_v \times d_e}$, $W_p \in \square^{\ell_{\max} \times d_e}$, the token and positional embedding matrices.

For $l \in [L_{\mathrm{enc}}]$:

| W_l^{enc}, multi-head encoder attention parameters for layer l.

| $W_{\mathrm{mlp1}}^l \in \square^{d_e \times d_{\mathrm{mlp}}}, b_{\mathrm{mlp1}}^l \in \square^{d_{\mathrm{mlp}}}$, $W_{mlp2}^l \in \square^{d_{mlp} \times d_e}, b_{mlp2}^l \in \square^{d_e}$, MLP parameters.

1. $\ell_z \leftarrow \mathrm{length}(z)$
2. for $t \in [\ell_z]$: $e_t \leftarrow$ TokenEmbedding($z[t]$) + PositionalEmbedding(t)
3. $Z \leftarrow [e_1, e_2, ..., e_{l_z}]^{\mathrm{T}}$
4. for l =1, 2 , …. , L_{enc} do
5. $Z \leftarrow Z + \mathrm{Layer\_norm}(\mathrm{MHAttention}(Z, Z \,|\, W_l^{\mathrm{enc}}, \mathrm{Mask} \equiv 1))$
6. $Z \leftarrow Z + \mathrm{Layer\_norm}((\mathrm{ReLU}(ZW_{\mathrm{mlp1}}^l + 1^{\mathrm{T}} b_{\mathrm{mlp1}}^l))W_{\mathrm{mlp2}}^l + 1^{\mathrm{T}} b_{\mathrm{mlp2}}^l)$
7. $Z \leftarrow \mathrm{ContextFusion}(Z, P_e)$, Refer equation 5.
8. **return** Z

4) Gated Context Fusion: A gating mechanism integrates context for a source sentence. The final encoder representation h is calculated according to equations (5a) and (5b), following Zheng et al. (2020)'s approach [37]:

$$g = \sigma(W_g[Z; P_e]) \tag{5a}$$

$$h = LayerNorm((1-g) \square Z + g \square P_e) \tag{5b}$$

Algorithm 5: $\tilde{Z} \leftarrow$ ContextFusion(Z, P_e)
/* Computes Context Fusion between Local and Global Embedding. */
/* **Symbol Notation:** Refer Appendix A */

Input: $Z \in \square^{\ell_z \times d_e}, P_e \in \square^{\ell_z \times d_e}$, is source token pre-training embedding.

Output: $\tilde{Z} \in \square^{\ell_z \times d_e}$, updated context embedding.
Hyperparameters: H, number of attention heads.
Parameters: $W_g \in \square^{\ell_{\max} \times 2d_e}, b_g \in \square^{2d_e}$

1. $g \leftarrow \mathrm{ReLU}(([Z; P_{em}])W_g + 1^{\mathrm{T}} b_g$, ; is concatenation
2. $\tilde{Z} \leftarrow \mathrm{Layer\_norm}((1-g) \square Z + g \square P_{em}), \square$ is element-wise multiplication
3. **return** $\tilde{Z}$

B. Decoder

The decoder's purpose is to translate from source to target language utilizing the encoder's hidden vector representation as a reference point; thus, the hidden representation Z (Algorithm 4) is fetched as input to the decoder block. Likewise, during training, the target sentence x is one of the primary inputs, which is transformed into word and positional encoding as shown in algorithm 6. The embedded representation X is fed into the masked multi-head attention layer, which computes the self-attention for the target sentence as shown in 6a and 6b.

$$D = \mathrm{MaskedMultiHead}(\mathrm{SelfAttention}(Q \equiv X,\ K \equiv X, V \equiv X, Mask)) \tag{6a}$$

$$D = \text{MaskedMultiHead}(Q + \text{LayerNorm}(\sum_n \text{Similarity}(Q, K_n) \times V_n)) \quad (6b)$$

Here, D is the self-attended embedding, and *Mask* is the target sentence masked input. *Mask* input will attend only to previously generated words and masked future words; see equation (4). Further, the hidden representation from the encoder block will be the value and key for the multi-head attention of the decoder's block shown in equations (7a), (7b), and (7c):

$$T = \text{MultiHead}(\text{SelfAttention}(Q \equiv D, K \equiv Z,\ V \equiv Z)) \quad (7a)$$

$$T = \text{MultiHead}(Q + \text{LayerNorm}(\sum_n \text{Similarity}(Q, K_n) \times V_n)) \quad (7b)$$

$$\hat{T} = T + \text{LayerNorm}(\text{FeedForward}(T)) \quad (7c)$$

$\hat{T}$ is linearly projected to the dimension of the target vocabulary size. The probability distribution throughout the vocabulary size is then returned using a softmax function. The most likely word of the target sentence is picked. Algorithm 6 provides a comprehensive breakdown of the operational procedures involved in the functioning of the decoder block.

Algorithm 6: $P \leftarrow$ Decoder(x, $Z|\theta$) /* Decoder transformer forward pass */
/* **Symbol Notation:** Refer Appendix A */
Input: $x \in V^*$, is target token ID, $Z \in \mathbb{R}^{\ell_x \times d_e}$, is encoder context embedding.
Output: $P \in (0,1)^{\ell_x \times N_v}$, is the output probability distribution across the target vocabulary.
Hyperparameters: $\ell_{\max}, L_{\text{dec}}, H, d_e, d_{\text{mlp}} \in \mathbb{R}$
Parameters: θ includes all the following parameters:
$W_e \in \mathbb{R}^{N_v \times d_e}$, $W_p \in \mathbb{R}^{\ell_{\max} \times d_e}$, the token and positional embedding matrices.
For $l \in [L_{\text{dec}}]$:
| W_l^{dec}, multi-head decoder attention parameters for layer l.
| $W_l^{e/d}$, multi-head cross-attention parameters for later l.
| $W_{\text{mlp3}}^l \in \mathbb{R}^{d_e \times d_{mlp}}, b_{\text{mlp3}}^l \in \mathbb{R}^{d_{\text{mlp}}}$, $W_{\text{mlp4}}^l \in \mathbb{R}^{d_{\text{mlp}} \times d_e}, b_{\text{mlp4}}^l \in \mathbb{R}^{d_e}$, MLP parameters.
$W_u \in \mathbb{R}^{d_e \times N_v}$, the unembedded matrix.

1. $\ell_x \leftarrow \text{length}(x)$
2. for $t \in [\ell_x]: e_t \leftarrow$ TokenEmbedding($x[t]$) + PositionalEmbedding(t)
3. $X \leftarrow [e_1, e_2, ..., e_{l_x}]^{\text{T}}$
4. for i =1, 2 , …. ,L_{dec} do
5. $X \leftarrow X + \text{Layer\_norm}(\text{MHAttention}(X, X \mid W_l^{\text{dec}}, \text{Mask}[t,t'] \equiv\ t \le t'\ ))$
6. $X \leftarrow X + \text{Layer\_norm}(\text{MHAttention}(X, Z \mid W_l^{e/d}))$
7. $X \leftarrow X + \text{Layer\_norm}((\text{ReLU}(XW_{\text{mlp3}}^l + 1^{\text{T}} b_{\text{mlp3}}^l))W_{\text{mlp4}}^l + 1^{\text{T}} b_{\text{mlp4}}^l)$
8. **return** $P = \text{Softmax}(XW_u)$

C. Inference

Inference/translation steps are very similar to training except that the input to the decoder begins from the start of a sentence <*sos*> symbol. Then the model continuously predicts the next word of a sentence in a sequence-to-sequence fashion till it encounters an end of a sentence <*eos*> or maximum sequence size.

V. CORPUS

A Python library called Beautiful Soup was used to scrape data from local newspapers (*U Rupang*, *Kynjatshai*, *SP News Agency*, *Syllad*), NGO portals, other organization websites and district-level government websites to create a monolingual corpus. The parallel corpus contains web scraps of the Bible, books, articles, newspapers, and magazines, among other publications. We perform manual and automated pre-processing on the obtained corpora using regular expressions in Python scripts to further ensure a high-quality corpus. A summary of the corpus size for training, validation, and testing data can be found in Table I. The size of the monolingual corpus for the Khasi language is 296.1 megabytes.

Type	Augmentation	No. of sentences	Size(MB)	
			Khasi	English
Test	No	11476	1.1	0.89
Validation	No	8730	0.9	0.73
	Yes	17242	1.79	1.48
Train	No	78578	8.1	6.57
	Yes	155182	16.11	13.32

Table 1: Parallel Corpus

VI. IMPLEMENTATION

The translation system is implemented in the PyTorch framework using Google collaboratory[5] [24]. For the Khasi language, the hugging face transformer Byte-Pair Encoding(BPE) [28] model was trained using a monolingual corpus, and a pre-trained BPE model for English was used for tokenization and vocabulary construction. The Khasi language model is trained using RoBERTa language model, and an English pre-trained language model from the Hugging Face community is used. Further, TextAttack library [20] is used for automatic data augmentation. It is restricted to English sentences, and the Khasi phrases are left as they are, with the assumption that Khasi has a smaller vocabulary compared to English; hence, variations of an English sentence will have the same Khasi translation.

The best hyperparameter for training the neural networks should be found before the translation model is trained. Hence, the network is initially trained for two epochs. Table II shows the parameters considered when training the system. We discovered that shuffled data with a learning rate 0.0001 and a batch size of 128 is the best potential setup. The actual translation system is trained for around 34 hours.

Parameter	Values
Batch Size	8, 64, 128, 512
Learning rate	0.1, 0.001, 0.0001, 0.00001
Shuffle	Yes, No

Table 2: Possible Hyperparameters Settings

After determining the best hyperparameter, examining the network's weight distribution is critical. An illustration of the bias and weight distribution in the 4th layer of the encoder block's layer can be seen in figure 2. X-axis shows the actual weight or bias value, y-axis shows the epoch number, and z-axis shows the number of weights. The model does not learn if the histogram remains constant across all epochs. Figure 2 shows that the weights and biases change continuously, indicating that the network is learning appropriately. It is also critical to look at training accuracy and loss. From Figure 3, training accuracy should increase with increasing training steps, while training and validation loss should decrease. As a result, the network is learning effectively.

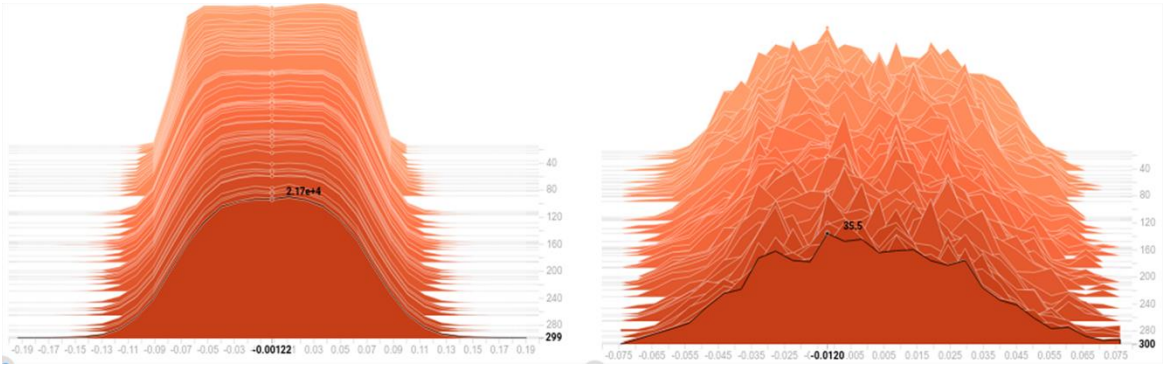

Fig. 2: Layer 4 self-attention weights and bias distributions in encoder's block.

[5] https://github.com/jefson08/KhasiNMT.git

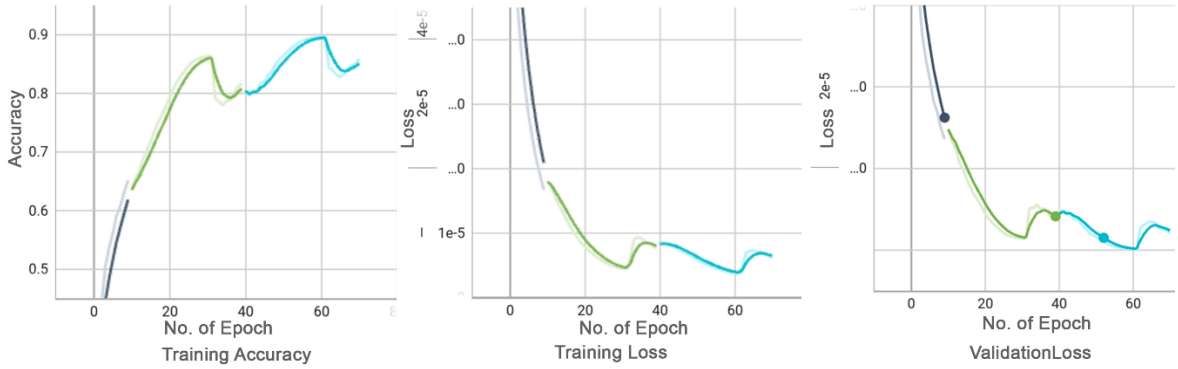

Fig. 3: Train and loss accuracy

VII. RESULTS

The translation quality is assessed using a variety of metrics. The results in Table III indicated several metrics tested on English-to-Khasi and Khasi-to-English translation systems. The candidates' sentences generated by the system are compared to a pre-translated reference sentence. Metric values for BLEU, METEOR, and ROUGE_L vary between 0 and 1, with values closer to 1 suggesting a better or more acceptable translation. In this paper, all Metrics values are converted to equivalent percentages. BLEU reflects the similarity of the candidate text to the reference text, with higher values indicating greater similarity [23]. It gives a general assessment of the model's quality. ROUGE_L refers to how much of the reference summary is recovered or captured by the candidate summary [17]. METEOR claims to have a higher correlation to human judgement [6].

Khasi - English translation					
Pretrained Context	Pre-trained Embedding	Data Augmentation	Bleu	METEOR	ROUGE_L
No	No	No	56.98	43.12	65.14
No	Yes	Yes	57.20	44.40	63.76
Yes	Yes	Yes	57.43	45.80	65.47
Hybrid			59.06	45.95	66.86
English - Khasi translation					
No	No	No	46.87	39.95	62.44
No	Yes	Yes	49.05	42.09	65.45
Yes	Yes	Yes	49.29	42.54	65.54
Hybrid			49.82	41.99	64.04

Table 3: Metrics Score

Table III shows two different translation systems from Khasi to English and from English to Khasi. Many different combinations of hyperparameters were considered during training. A model that included pre-trained context encoding, pre-trained word embedding initialization, and the augmentation of input sentences is the best translation system, as the third row of Table III illustrates. It was found that a model without pre-trained context encoding, pre-trained word embedding initialization, and input sentence augmentation gives better translation results for short sentences. Conversely, longer sentences are best translated using a model equipped with pre-trained context encoding, pre-trained word embedding initialization, and input sentence augmentation. Thus, a hybrid system determines which translation model to use during translation based on the sentence input's length. This is shown in the fourth row of Table III.

Figures 4 show a graphic representation of the various metrics scores of our trained models, which will be discussed further in Section VIII. Table IV also compares the BLEU and semantics scores. Semantic scores are a sentence embedding method that demonstrates the similarity between two sentences [27]. Hugging Face's SentenceTransformer library is used to accomplish this. A more detailed analysis is discussed in Section VIII.

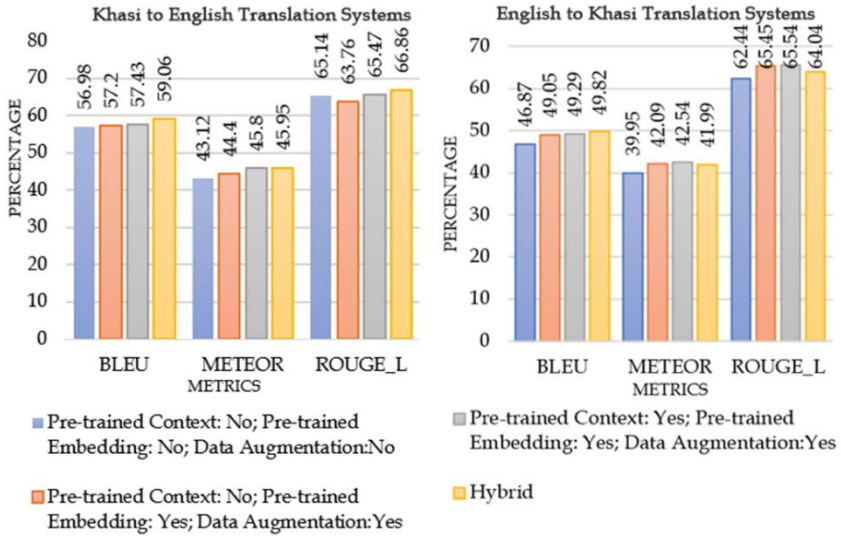

Fig 4: Translation Metrics

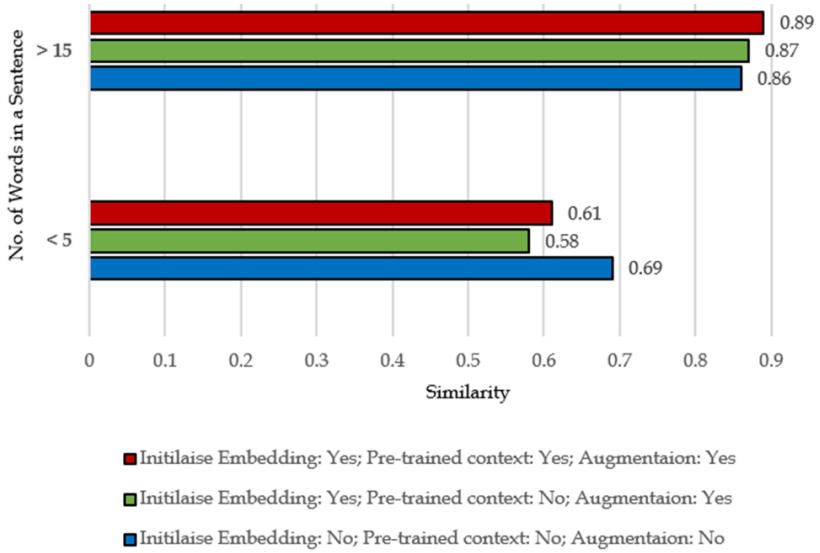

Fig 5. Average semantic similarity scores between reference and candidate sentences.

Khasi (Source)	kumta u la kyntait hit ïa ki briew jong u
Reference	so he rejected his people completely
Predicted	so he rejected his people.
Bleu	50.81
Semantic score	96.33
Khasi (Source)	Balei U Blei um ai ha nga ia kaba nga pan?
Reference	Why won't God give me what I ask?
Predicted	Why should God give me what I requesting for?
Bleu	46.71
Semantic score	81.52
Khasi (Source)	U la iarap ïa nga ban lait na ka jingma
Reference	He helped me out of danger
Predicted	He helped me to escape from danger.
Bleu	22.08
Semantic score	91.57

Table 4: BLEU and Semantic Similarity

Additionally, the proposed model is compared to the existing SMT model using the Moses Toolkit [15] and the LSTM model using the PyTorch version of openNMT [14]. From Table V, it was observed that our proposed model outperformed both SMT and LSTM models. Furthermore, Figure 5 depicts the average semantic scores for the different models. The semantic score ranges from 0 to 1, with a higher score indicating greater semantic similarity between sentences. When translating sentences with more than 15 words, a model with pre-trained context knowledge, initializing embedding layer weights with pre-trained embeddings, and incorporating data augmentation contribute to better semantic scores. In the case of sentences with less than 15 words, translation gives better semantic scores without incorporating the above parameters, as shown in Figure 5. Finally, the results of the comparison between the experimental model and ChatGPT4o [1] are shown in Table VI. It shows the BLUE score, the chrF score [26], and the semantic similarity. Section VIII further examines these comparisons.

Khasi - English translation	
Models	BLEU Score
LSTM	43.75
SMT	47
Transformer with context embedding	57.43
English - Khasi translation	
LSTM	48.11
SMT	48.07
Transformer with context embedding	49.29

Table 5: Statistical Vs Neural Machine Translation Comparisons

Model	BLEU	chrF	Semantic Similarity
ChatGPT4o	36	57	70
Experimented model	63	76	81

Table 6: Khasi to English Translation: A Comparison with ChatGPT4o

VIII. DISCUSSION

The models with pre-trained contextual encoding, pretrained word embedding, and adopting data augmentation generated the highest BLEU score of 57.43, as seen in Table III and Figure 4. This is primarily due to two factors. Firstly, the pre-trained language model could capture more contextual information about the input sentence because it was trained using a more readily available monolingual corpus. Secondly, data augmentation can increase the training or validation corpus's size, thereby increasing the data's diversity and thus contributing to effective learning. METEOR score on the same model is 45.80, which shows how similar the sentences manually translated are to the predicted sentences. The degree to which the candidate captures the reference or predicted translation is indicated by the ROUGE_L score of 65.47.

According to our findings, the model incorporating pretrained context encoding, initializing pre-trained word embeddings, and adopting data augmentation to the input sentences performs better for longer sentences. The pre-trained context encoding of a source sentence is extracted from the language model trained from a more readily available monolingual corpus. Therefore, the translation model will be able to capture a greater variety of meanings for a given input sentence. In contrast, the model performs well for shorter sentences without the above parameters. Thus, a hybrid system can determine which translation model to deploy depending on the length of the input sentence, as shown in Table III.

Adopting one of the best-determined parameters, the translation systems from English to Khasi attain a BLEU, METEOR, and ROUGE_L score of 49.82, 41.99, and 64.04, respectively. The limited monolingual data available for Khasi to train a language model for extracting pre-trained contextual information may have contributed to a lower metric score when translating English to Khasi.

Furthermore, since the above metrics (BLEU, METEOR, ROUHE_L) does not check the sentence's semantic meaning, using them alone to judge the translation quality may be insufficient. Therefore, as shown in Table 6, we used a semantic score. It is observed that the above metrics perform poorly when the models make predictions using different synonyms, so semantic similarity aids in a deeper understanding of the translation system. For instance, the BLEU score between the reference sentence "so he rejected his people completely" and the predicted sentence "so he rejected his people." is 50.81, while for the same sentences, the semantic score is 96.33. Figure 5 shows the average semantic score between reference and predicted sentences.

Upon comparing the optimal model from the current experiment with the most recent ChatGPT4o, Table VI reveals that ChatGPT lags behind in all metrics, potentially because the test sentences align more closely with the experimental model. Generally, we observed that the ChatGPT4o model excelled in translating commonly used texts, yet it frequently failed to capture the subtle meaning, cultural context, and unique Khasi expressions, a task that the experimental model typically accomplished. Moreover, ChatGPT4o was very accurate in translating biblical texts as compared to the other model.

IX. CONCLUSION & FUTURE WORKPYRIGHT

This work is based on the transformer model, a neural machine translation technique. We developed an approach that extends the functionality of the transformer by incorporating some modifications to the basic model. The modifications include data augmentations to the parallel corpus, initializing the embedding layers with pre-trained word embedding, and appending the input sentence learned encoding with a contextual encoding from a pre-trained RoBERTa language model. It is found that incorporating these three functional components resulted in a significantly improved translation system concerning low-resource languages like Khasi. The proposed model for Khasi to English translation achieves a BLEU, METEOR and ROUGE_L score of 59.06, 49.95 and 66.86, respectively. English to Khasi obtained a BLEU, METEOR, and ROUGE_L score of 49.82, 41.99, and 64.04, respectively. There is a possibility that the limited monolingual corpus size used to train the language model may have influenced the low scores for English to Khasi translation. In comparison, the suggested model outperforms the SMT and LSTM translation models. Our findings suggest that the model with no pre-trained context encoding, no pre-trained word embedding initialization, and no data augmentation of input sentences perform better for shorter sentences. In contrast, the model performs well for longer sentences using the above parameters, enhancing sentence predictability and resulting in higher accuracy. Thus, a hybrid system can determine which translation model to adopt depending on the length of the input sentence, as shown in Table III. Finally, the semantic similarity between the reference and predicted sentences is measured by computing a semantic score.

Lastly, we compare the optimal model from the current experiment with the latest ChatGPT4o version as shown in Table VI. The test data's greater alignment with our model could explain why ChatGPT4o's metrics scores were lower than those of the considered model. The paper does not claim that the model under consideration is better than ChatGPT, but it does indicate that even with comparatively few resources, the model can achieve a relatively quality translation.

In the future, expanding both the monolingual and parallel corpora will be necessary. Additionally, it is worth considering the implementation of automatic text post-editing as a potential solution to address the occasional translation of duplicate words or groups of words within the proposed system.

ACKNOWLEDGMENT

The author places on record sincerest thanks to the faculty, Department of Computer Science, Assam University, for their assistance and support, the District level Government of Meghalaya Portal, the Bible Society Shillong, Ri Khasi Press and different Press Managers of Local Newspapers in Shillong, for facilitating my access to their portals and websites and reports, the North Eastern Hill University Central Library, Shillong with its excellent collection of books and journals - to each one of sources I availed, my abundant gratitude for enriching my corpus of knowledge, enabling me to shape up my paper and strengthening my research work.

DECLATIONS

- Funding : There is no funding available.
- Conflict of interest/Competing interests : The authors declare that there is no conflict of Interest
- Ethics approval : Not applicable
- Consent to participate: Not Applicable
- Consent for publication : Not Applicable
- Availability of data and materials: https://github.com/jefson08/KhasiNMT.git
- Code availability: https://github.com/jefson08/KhasiNMT.git
- Authors' contributions: All authors contributed equally to this work

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Altay. Implementing Formal Algorithms for Transformers. https://gabriel-altay.medium.com/implementing-formal-algorithmsfor-transformers-c36d8a5fc03d, may 19 2023.

[3] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*, April 2018.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[7] Rupjyoti Baruah, Rajesh Kumar Mundotiya, and Anil Kumar Singh. Low resource neural machine translation: Assamese to/from other indoaryan (indic) languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(1), nov 2021.

[8] Jacob Devlin. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu, 2017.

[9] Alex Hernández-García and Peter König. Further advantages of data augmentation on convolutional neural networks. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 95–103, Cham, 2018. Springer International Publishing.

[10] Rudali Huidrom and Yves Lepage. Introducing EM-FT for ManipuriEnglish neural machine translation. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 1–6, Marseille, France, June 2022. European Language Resources Association.

[11] Aiusha V Hujon, Thoudam Doren Singh, and Khwairakpam Amitab. Transfer learning based neural machine translation of english-khasi on low-resource settings. *Procedia Computer Science*, 218:1–8, 2023.

[12] Vanlalmuansangi Khenglawt, Sahinur Rahman Laskar, Partha Pakray, Riyanka Manna, and Ajoy Kumar Khan. Machine translation for lowresource english-mizo pair encountering tonal words. *Computación y Sistemas*, 26(3):1377–1398, 2022.

[13] Marbhador Manik Khymdeit. Ka ktien khasi bad ki kyntien shim kylliang. *Ka Thwet Jingstad: A Journal of The Society for Khasi Studies*, III(2 August):12–18, 2015.

[14] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017.

[15] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[16] Sahinur Rahman Laskar, Partha Pakray, and Sivaji Bandyopadhyay. Neural machine translation for low resource assamese–english. In *Proceedings of the International Conference on Computing and Communication Systems*, pages 35–44, Singapore, 2021. Springer Singapore.

[17] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[19] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020.

[20] John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.

[21] Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. *CoRR*, abs/1910.05895, 2019.

[22] Kynpham Sing Nongkynrih. Problems of translation: The khasi perspective. *Ka Thwet Jingstad: A Journal of The Society for Khasi Studies*, III(2 August):37–48, 2015.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[25] Mary Phuong and Marcus Hutter. Formal algorithms for transformers. *arXiv preprint arXiv:2207.09238*, 2022.

[26] Maja Popovic. chrf: character n-gram f-score for automatic mt eval-´ uation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395, 2015.

[27] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[28] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.

[29] Thoudam Doren Singh and Aiusha Vellintihun Hujon. Low resource and domain specific english to khasi smt and nmt systems. In *2020 International Conference on Computational Performance Evaluation (ComPE)*, pages 733–737, 2020.

[30] Overland Snaitang. *Christianity and social change in northeast India: a study of the role of Christianity in social change among the KhasiJaintia Hill tribes of Meghalaya*. Vendrame Institute ; Firma KLM Private Ltd., Shillong, 1993.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[32] N Donald Jefferson Thabah and Bipul Shyam Purkayastha. Low resource neural machine translation from english to khasi: A transformerbased approach. In *Proceedings of the International Conference on Computing and Communication Systems: I3CS 2020, NEHU, Shillong, India*, pages 3–13. Springer, 2021.

[33] N Donald Jefferson Thabah and Bipul Syam Purkayastha. Khasi to english neural machine translation: an implementation perspective. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(2):4330–4336, December 2019.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[35] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao,Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[36] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10524–10533. PMLR, 13–18 Jul 2020.

[37] Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. Towards making the most of context in neural machine translation, 2020.

[38] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating BERT into neural machine translation. *CoRR*, abs/2002.06823, 2020.

[39] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation, 2016.

APPENDIX A: LIST OF NOTATION

Symbol	Type	Explanation
$[N]$	$:= \{1,\ldots,N\}$	Set of integers 1, 2, … , N-1, N
i, j		Indices
N_V	$\in \mathbb{N}$	Vocabulary size
V	$\cong [N_V]$	Vocabulary
V^*	$= \bigcup_{\ell=0}^{\infty} V^\ell$	Set of token sequence
P_e	$\in \mathbb{R}^{\ell \times d_e}$	Token sequence pre-trained embedding
$\ell_{\max}$	$\in \mathbb{N}$	Maximum sequence length
ℓ	$\in [\ell_{\max}]$	Length of token sequence
t	$\in [\ell]$	Index of a token in a sequence
x	$\equiv x[1:\ell]$	$\equiv x[1]x[2]...x[\ell] \in V^\ell$, Primary token sequence
z	$\equiv z[1:\ell]$	$\equiv z[1]z[2]...z[\ell] \in V^\ell$, Context token sequence
d	$\in \mathbb{N}$	Dimension of various vectors
$M[i,:] \equiv M[i]$	$\in \mathbb{R}^{d'}$	i-th row of matrix $M \in \mathbb{R}^{d \times d'}$
$M[:, j]$	$\in \mathbb{R}^{d}$	j-th column of matrix $M \in \mathbb{R}^{d \times d'}$
e	$\in \mathbb{R}^{d_e}$	Vector representation/embedding of a token
$L, L_{\text{enc}}, L_{\text{dec}}$	$\in \mathbb{N}$	Number of network (encoder, decoder) layers

l	$\in [L]$	Index of network layer
H	$\in$	Number of attention heads
h	$\in [H]$	Index of attention head
W_e	$\in$ $^{N_V \times d_e}$	Token embedding matrix
W_p	$\in$ $^{\ell_{\max} \times d_e}$	Positional embedding matrix
W_u	$^{d_e \times N_v}$	Unembedding matrix
W_q	$\in$ $^{d_x \times d_{\text{attn}}}$	Query weight matrix
b_q	$\in$ $^{d_{\text{attn}}}$	Query bias
W_k	$\in$ $^{d_e \times d_{\text{attn}}}$	Key weight matrix
b_k	$\in$ $^{d_{\text{attn}}}$	Query bias
W_v	$\in$ $^{d_e \times d_{\text{out}}}$	Value weight matrix
b_v	$\in$ $^{d_{\text{out}}}$	Value bias
W_o	$\in$ $^{Hd_{\text{out}} \times d_{\text{out}}}$	Output weight matrix
b_o	$\in$ $^{d_{\text{out}}}$	Output bias
W_{mlp}	$\in$ $^{d_1 \times d_2}$	Weight matrix corresponding to an MLP layer in a Transformer
b_{mlp}	$\in$ ^{d_1}	Bias corresponding to an MLP layer in a Transformer
$\theta, \tilde{\theta}$	$\in$ ^{d}	Collection of all learnable/learned Transformer parameters