Subwords to Word Back Composition for Morphologically Rich Languages in Neural Machine Translation

Telem Joyson Singh
IIT Guwahati
Assam, India
tjoyson@iitg.ac.in

Sanasam Ranbir Singh
IIT Guwahati
Assam, India
ranbir@iitg.ac.in

Priyankoo Sarmah
IIT Guwahati
Assam, India
priyankoo@iitg.ac.in

Abstract

The proliferation of word forms in morphologically rich languages poses a formidable challenge for neural machine translation (NMT) models, given their highly sparse vocabularies, which render the atomic treatment of surface word forms impractical. To address this issue, the common approach involves preprocessing words into subword units and performing translation at the subword level. However, this method loses the word boundary information and neglects the valuable prior knowledge of the interrelationship between word stems and affixes. In this paper, we explore an approach that segments words into morphemes and composes word representations from these morphemes. Specifically, we represent words by combining the representations of subword units using a bidirectional GRU. Through extensive experiments on Manipuri-English, Tamil-English, and Marathi-English translation tasks, we demonstrate that our model outperforms baseline subword models in terms of translation accuracy.

Our findings hold considerable significance, as they showcase that leveraging the inductive bias derived from word boundary and the interrelationships between word morphemes can significantly improve NMT results compared to approaches that directly translate at the subword level.

1 Introduction

Neural machine translation (NMT) has shown remarkable performance and is widely recognized as the dominant approach to machine translation tasks (Barrault et al., 2020; Akhbardeh et al., 2021; Kocmi et al., 2022). However, NMT models face significant challenges when dealing with morphologically rich languages, as such languages possess highly sparse vocabularies. This sparsity leads to difficulties in handling unknown words and known words appearing in rare. Additionally, treating words as atomic units fails to fully harness the valuable subword information encoded in subword units. To illustrate this morphological complexity, Table 1 provides an example in Manipuri language.

Word	Translation
pu.ba	carry
pu.khi	carried
pu.sin.khi	carried in
pu.sin.ning.khi	wanted to carry in
pu.sin.ning.khi.de	not wanted to carry in

Table 1: A case study of agglutination in manipuri language. The bold part indicates the stem.

Subword-level Neural Machine Translation (NMT) models have emerged as the most preferred option for translating from Morphological Rich Languages (MRLs). These models operate by breaking words into their constituent subwords. To achieve this, purely statistical models like the byte-pair encoding (BPE) model (Sennrich et al., 2016) or linguistically motivated morphological analyzers such as Morfessor (Grönroos et al., 2014) are employed to preprocess the words. This approach holds great promise due to its inherent ability to transform the sparse surface form-based vocabulary set into a much smaller set of fundamental subunits. Especially for MRLs, the vocabulary size of subword units is significantly smaller than that of the word vocabulary set. Consequently, this approach offers a compelling solution to the challenge of out-of-vocabulary words, as encountering an unknown surface form is considerably more likely than encountering an unknown subunit.

Although the previously mentioned subword models show promise in enhancing translation quality, they do present a potential concern when the translation is done directly at the subword level. In models like BPE, which are purely statistical, or linguistically motivated segmentation models like Morfessor, the NMT model receives subwords in-

Table 2: Illustrating an example sentence at different granularity level: word and subword levels. The blue colored words are segmented into morphemes.

Word	amuk hanna eihakna degree phang-basingbu thagatchari
Morfessor	amuk hanna eihakna degree phang-basing bu thagatcha ri
BPE	amuk hanna eihakna degree phang basingbu thagatcha ri

stead of complete words. An example of this is shown in Table 2, where the orange-colored text represents complete words, and the blue-colored text represents segmented subwords. Regrettably, it may not be the optimal way to learn and represent source-side information at the subword level. This method disregards vital word boundary information and overlooks valuable prior knowledge concerning the interrelationship between word stems and affixes. Furthermore, the encoder-decoder architecture (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017) utilized in NMT was originally designed to operate at the word level, but now it is applied to subwords directly. Consequently, the encoder is made to learn word boundaries and the interrelationship between word stems and suffixes on its own.

Taking into account the aforementioned shortcomings, we have undertaken an investigation into an approach encompassing two key aspects: (i) the segmentation of words into subword units, and (ii) the subsequent composition of these subword units back to the word level. In particular, we represent words by combining the representations of their subword units using a bidirectional GRU. By breaking words down into morphemes, this method effectively addresses the issue of data sparsity and effectively leverages subword information. Moreover, the approach of composing word representations from subword representations helps overcome the word boundary problem.

Our extensive experimentation with Manipuri-English, Tamil-English, and Marathi-English translation tasks demonstrates that our model surpasses baseline subword models in terms of translation accuracy. These findings hold importance as they highlight the profound impact of utilizing the inductive bias derived from the word boundary and the interrelationships between word morphemes. It is evident that this approach leads to remarkable enhancements in Neural Machine Translation (NMT) results compared to methods that translate directly at the subword level.

The main contributions of our work can be summarized as follows:

- We explore a method that segments words into subword units to address data sparsity, and then recompose these subword units back to the word level to address word boundary issue.
- We demonstrated that our method outperforms baseline subword models through experiments on multiple language pairs.

2 Background

2.1 Morphological Typology and MT

Morphologically, languages are often characterized on the basis of two dimensions of morphological variation. The first dimension concerns the number of morphemes per word, with **isolating** languages like Vietnamese and Cantonese that typically have one morpheme per word, while **polysynthetic** languages like Siberian Yupik can contain numerous morphemes within a single word, equivalent to an entire English sentence. The second dimension revolves around the degree to which morphemes are segmentable, classified into **agglutinative** languages like Manipuri and Tamil, where morphemes have clear boundaries, and **fusion** languages like Hindi and Punjabi, where a single affix can combine multiple morphemes.

Previous studies (Sennrich et al., 2016) have found that the translation of languages with rich morphology requires handling structures beyond the word level. For this reason, modern systems typically employ subword models such as purely statistical models like byte-pair encoding (BPE) model (Sennrich et al., 2016) or linguistically motivated morphological analyzers such as Morfessor (Grönroos et al., 2014).

2.2 Related Work

The impact of morphology on the architecture and performance of machine translation models is a crucial concern, especially when dealing with languages that possess rich morphology. In this context, two approaches for adjusting the granularity of translation, subword-based and character-based Neural Machine Translation (NMT) models, have shown promise in addressing specific challenges.

Huck et al. (2017) proposed the idea of first splitting words at morpheme boundaries, such as prefixes and suffixes, and then performing unsupervised segmentation using Byte-Pair Encoding (BPE).

Luong and Manning (2016) introduced a hierarchical model called "back-off revisited." This model utilizes a word-level model to generate UNK (unknown token) replacements, which leverages a character-level model to predict words from UNKs based on hidden states. The approach offers more flexibility than dictionary look-up while maintaining efficiency over pure character-level translation. However, it comes with the drawback of independence assumptions between the main and back-off models.

Character-level output, without word segmentation on the target side and employing a BPE-level encoder, has shown excellent results for languages like EN, DE, CS, RU, and FI. Nonetheless, the training time is approximately twice as long as the BPE-level model (Vylomova et al., 2016).

Lee et al. (2016) proposed Fully Character-Level Neural Machine Translation (NMT) to eliminate word boundaries using Character-Level Recurrent Neural Networks (RNNs) on the target side, combined with convolution and max-pooling layers on the source side. Later, Libovický et al. (2021) adapted Lee et al. (2016)'s model to the Transformer architecture.

Clark et al. (2021) introduced a model in 2021 that compresses character sequences into fewer hidden states by employing local self-attention and strided convolutions.

Tay et al. (2021) proposed Charformer, which utilizes convolutional operations and merges character blocks to acquire latent subword representations.

Additionally, previous work has considered incorporating morphological information in NMT. Sennrich and Haddow (2016) applied morphology on the source side and utilized word+lemma as input, enabling the merging of multiple features conveniently.

Tamchyna et al. (2017) proposed a 2-step translation approach in NMT, involving predicting interleaved lemmas and morphological categories, enabling inflection generation using finite state transducers.

Passban et al. (2018a) enhanced the character-based decoder by incorporating a morphology table to provide guidance on the morphological structures of the target language.

Furthermore, Passban et al. (2018b) improved the existing NMT architecture with a double-channel encoder and a double-attentive decoder.

Ataman et al. (2019) generated words character by character through a combination of two latent representations: a continuous one to capture lexical semantics and a set of (approximately) discrete features to capture morphosyntactic function.

2.3 The Transformer

In this study, we apply our approach within the framework of the Transformer (Vaswani et al., 2017), which shall be briefly introduced herein. It is essential to acknowledge that our method can also be integrated with various other Neural Machine Translation (NMT) architectures.

Let $(\mathbf{x_1}, \ldots, \mathbf{x_J})$ represent the input sequence of symbol representations, $(y_1^*, \ldots, y_{K^*}^*)$ denote the ground truth sequence, and $(y_1, \ldots, y_K)$ indicate the translation.

Encoder & Decoder The encoder is composed of N identical layers, each comprising two sublayers: a *multi-head self-attention* sublayer and a fully connected *feedforward* network. Both sublayers are succeeded by a residual connection and a layer normalization operation. To prepare the input sequence, it is initially transformed into a sequence of vectors $E_x = [E_x[\mathbf{x_1}]; \ldots; E_x[\mathbf{x_J}]]$, where $E_x[\mathbf{x_j}]$ denotes the sum of the word embedding and the position embedding of the source word $\mathbf{x_j}$. Subsequently, this sequence of vectors is fed into the encoder, and the output of the N-th layer is denoted as H, representing the hidden states of the source.

The decoder also comprises identical N layers. Besides the same sublayers present in the encoder, each decoder layer includes a *cross-attention* sublayer inserted in between them. This sublayer performs multihead attention over the output of the encoder. The final output of the N-th layer provides the target hidden states $[\mathbf{s_1}; \ldots; \mathbf{s_{K^*}}]$, where $\mathbf{s_k}$ represents the hidden state of the target word represenation $\mathbf{y_k}$.

Objective The primary goal of the model is to optimize by minimizing the cross-entropy loss concerning the ground-truth:

$$L = -\frac{1}{K} \sum_{k=1}^{K} \log p(y_k^* | \mathbf{y_{<k}}, \mathbf{x_{1:J}})$$

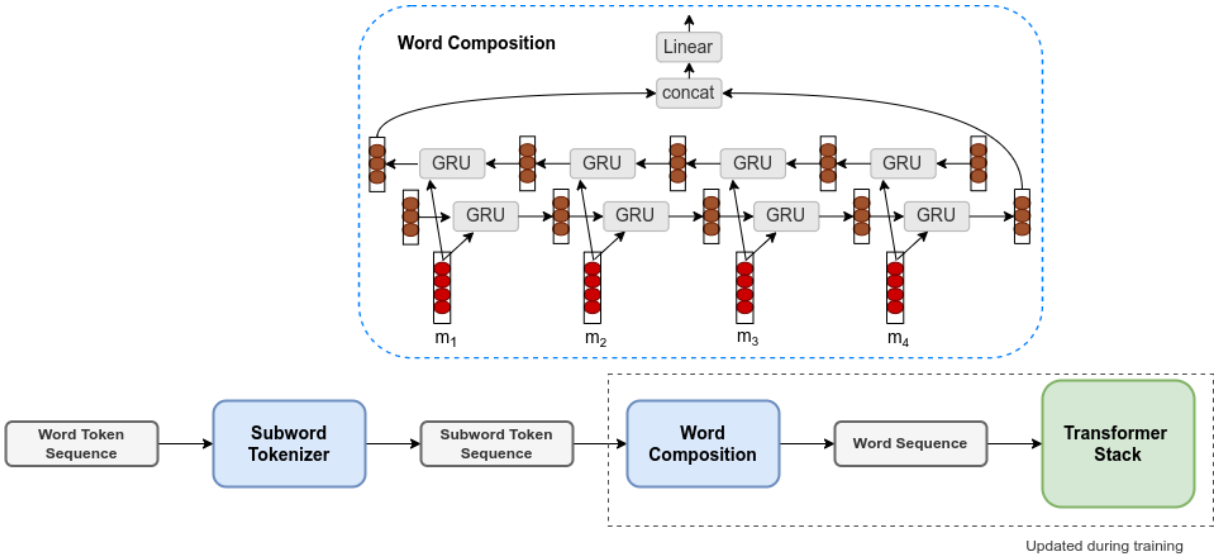

Figure 1: **Model diagram.** We show the procedure of segmentation of words into morphemes and subsequent composition of these subword units back to word level. A bidirectional GRU reads the morpheme sequence in our word composition model shown on the top.

where K denotes the length of the target sentence.

3 The Proposed Method

In this section, we first discuss the rationale for breaking words into morphemes and composing the word representation back from the morphemes. Then, we show the detailed workflow of our method. An overview of how the word composition module is applied can be found in Figure 1.

3.1 Motivation

The benefits of decomposing words into morphemes and subsequently reassembling them can be summed up in two points. **(1)** A word is composed of morphemes, which are recognized as the smallest units that convey meaning or carry grammatical significance in a language. By utilizing these meaningful morphemes to explicitly generate word embeddings, we incorporate prior semantic and grammatical knowledge into the learning process. Morphologically similar words often exhibit semantic connections, and their association can be facilitated by sharing the same morpheme vector. **(2)** Directly translating at the subword level disregards crucial information regarding word boundaries and overlooks valuable prior knowledge concerning the interrelationship between word stems and affixes. Reconstructing word representations from subword representations addresses the issue of the word boundary problem and enables a more comprehensive understanding of word structure.

3.2 Workflow

Our model is an adaptation within the framework of the Transformer (Vaswani et al., 2017). In Transformer, the encoder maps an input sequence of symbol representations $(\mathbf{x_1}, \ldots, \mathbf{x_J})$ to a sequence of continuous representations $(\mathbf{z_1}, \ldots, \mathbf{z_J})$. Given $(\mathbf{z_1}, \ldots, \mathbf{z_J})$, the decoder then generates an output sequence $(y_1, \ldots, y_K)$ of symbols one element at a time. In our method, we compute the representation $\mathbf{x}$ of word x as follows:

$$\mathbf{x} = f(\mathbf{W}_s, \sigma(x)) \quad (1)$$

where σ is a deterministic function that returns a sequence of subword units, $\mathbf{W}_s$ is a parameter matrix of representations for the vocabulary of subword units, and f is a composition function that takes $\sigma(x)$ and $\mathbf{W}_s$ as input and returns $\mathbf{x}$.

3.2.1 Generating Subword Units

To generate subword units, we define a mapping $\sigma : \mathcal{V} \mapsto \mathcal{M}^+$ of a word into a sequence of morphemes, that is, $\sigma(x) = (m_1, m_2, ...m_l)$, where $x \epsilon \mathcal{V}$ and $m_i \epsilon \mathcal{M}$. Therefore, we factor a surface word into its morphemes.

Suppose that there are $|V|$ individual words. We first segment each word as a sequence of morphemes using Morfessor tool (Grönroos et al., 2014). For Morfessor, we use the default parameters. To facilitate processing in neural networks,

we truncate morpheme sequences to a fixed length l (e.g., we set $l = 10$). For words containing fewer than l morphemes, we add additional tokens for padding. For instance, a single-morpheme word is padded with $l - 1$ padding morphemes. For words containing more than l morphemes, we concatenate the remaining morphemes into a single entity as the final one. After performing morpheme segmentation, we obtain an l-length morpheme sequence for each word. The processing could be explained as follows:

$$\sigma(x) = \begin{cases} [m_1, \ldots, m_i, \ldots, \text{pad}_n], & < l \\ [m_1, \ldots, \text{concat}(m_l, \ldots)], & > l \end{cases} \quad (2)$$

3.2.2 Composing Word Representation

Our composition function is a bidirectional recurrent neural network, particularly the gated recurrent unit (GRU) variant (Cho et al., 2014), which is inspired by its successful use in the character-level model and its widespread adoption in natural language processing (NLP). Given a morpheme representation $\mathbf{m_i}$ and the previous hidden state $\mathbf{h_{i-1}}$, a GRU computes the following outputs for the subword at position i:

$$\mathbf{h_i} = \text{GRU}(\mathbf{m_i}, \mathbf{h_{i-1}}) \quad (3)$$

In a bi-rnn (Graves and Schmidhuber, 2005), the final state of an RNN over the input sequence is combined with the final state of the RNN over the reversed input sequence. Given the hidden state produced by the last input of the forward GRU, $\mathbf{h_l^{(f)}}$, and the hidden state produced by the last input of the backward GRU, $\mathbf{h_0^{(b)}}$, we compute the word representation $\mathbf{x}$ as:

$$\mathbf{x} = \mathbf{W_f} \cdot \mathbf{h_l^{(f)}} + \mathbf{W_b} \cdot \mathbf{h_0^{(b)}} + b \quad (4)$$

where $\mathbf{W_f}$, $\mathbf{W_b}$, and b are parameter matrices, and $\mathbf{h_l^{(f)}}$ and $\mathbf{h_0^{(b)}}$ are the final forward and backward GRU states, respectively.

4 Experiments

4.1 Dataset

We evaluate our method on the following three datasets.

WAT2021[1] **Tamil→English** (140K pairs): We evaluate using WAT2021 validation set and test set using PMIndia (Haddow and Kirefu, 2020) and PIB Dataset (Siripragrada et al., 2020).

[1] http://lotus.kuee.kyoto-u.ac.jp/WAT

Algorithm 1: Training Algorithm

Input : x
Output : $\mathbf{x} \in R^d$

Procedure $WordComposition(x)$
  Generate the morphemes from x: $\sigma(x) = \begin{cases} [m_1, \ldots, m_i, \ldots, \text{pad}_n], & < l \\ [m_1, \ldots, \text{concat}(m_l, \ldots)], & > l \end{cases}$
  Take the final hidden states $\mathbf{h_l^{(f)}}$ and $\mathbf{h_0^{(b)}}$ from passing morpheme sequence through forward GRU and backward GRU.
  Compute the final word representation $\mathbf{x}$ as: $\mathbf{x} = \mathbf{W_f} \cdot \mathbf{h_l^{(f)}} + \mathbf{W_b} \cdot \mathbf{h_0^{(b)}} + b$
End
Procedure $Training(D)$
  Get word representation sequence from token sequence using $WordComposition(.)$
  Transformer stack operates on the word representation instead of subword sequence.
End

Manipuri→English (120K pairs): We use a training set of 120K parallel sentences from PMIndia (Haddow and Kirefu, 2020; Singh et al., 2021) and PIB dataset. The validation and test sets consist of approximately 1k sentence pairs each, sampled from the corpus.

WAT2021[1] **Marathi→English** (132K pairs): We evaluate using WAT2021 validation set and test set using PMIndia and PIB Dataset.

To preprocess the data, we segment English words into subword units using Byte-Pair Encoding(BPE) (Sennrich et al., 2016) with 16,000 merge operations. Morphemes are split from Manipuri, Tamil, and Marathi words using the Morfessor Flatcat tool (Grönroos et al., 2014). Table 3 summarizes some statistics about our training corpora.

4.2 Settings

We conduct experiments on the following systems.

Byte Pair Encoding (BPE) : This system utilizes a standard transformer model trained on data segmented with the BPE method (Sennrich et al., 2016).

	Manipuri-English		Tamil-English		Marathi-English	
	MNI	EN	TA	EN	MR	EN
Sentence	119412	119412	148180	148180	142654	142654
Token	1991343	2387089	2560332	3399785	2739815	3354532
Type	161064	90368	232640	124352	187944	120248
TTR	12.36	26.41	11.00	27.34	14.57	27.89
Stem	30800	×	29360	×	32313	×
Affix	967	×	2032	×	1103	×
Character	311	246	190	195	191	170

Table 3: Statistics of the corpus including the number of parallel sentences, number of words, number of unique words, token-to-type ratio(TTR), number of unique stems and affixes and the number of unique characters.

Morfessor : This system is trained on data segmented with the Morfessor approach (Grönroos et al., 2014).

Vanilla Character : In this configuration, the model is trained on character sequences.

Fully Character-based : This model is based on the approach used by (Libovický et al., 2021).

In this study, we conducted experiments using the Transformer(Vaswani et al., 2017) model, while implementing our models through the adaptation of the Fairseq-py open source toolkit (Ott et al., 2019). There are three input and output layers with embedding dimension of 256, the inner feed-forward layer dimension of 512, and the number of heads in the multi-head modules in both the encoder and decoder layers is 4. For character, embedding dimension is 64. The training batches consists of sets of 64 source and target sentences. The models are trained and evaluated on two Tesla P100 GPUs. The test set is evaluated using a single model obtained by taking the best checkpoint, which is validated on the development set at each epoch. The BLEU metric (Papineni et al., 2002) is used to evaluate the translation performance using the SacreBLEU (Post, 2018).

4.3 Main Results

Table 4 shows the BLEU scores achieved by various models on the translation tasks from Mni→En, Ta→En, and Mr→En test sets. We compare our proposed method with several existing approaches, including BPE (Sennrich et al., 2016), Morfessor (Grönroos et al., 2014), Vanilla Character and Fully Character-based (Libovický et al., 2021) models.

Morfessor and Vanilla Character models outperform BPE model. Among the baselines, BPE achieves a BLEU score of 23.00, 19.93, and 21.26 for Mni→En, Ta→En, and Mr→En translation tasks, respectively. Morfessor performs slightly better with BLEU scores of 23.74, 21.79, and 22.52 for the respective translation tasks. The vanilla character-based model exhibits similar performance with BLEU scores of 23.77, 21.70, and 21.81.

Our method outperform BPE, Morfessor and Vanilla Character models. In contrast, our proposed method, denoted as "Our Method" in the table, outperforms all the baselines significantly. It achieves a remarkable BLEU score of 25.61 for Mni→En, 22.56 for Ta→En, and 24.04 for Mr→En. The average BLEU score across all translation tasks for our method is 24.07.

The results indicate that our proposed method surpasses the existing models in translation accuracy and demonstrates its effectiveness in handling multilingual translation tasks. These findings confirm the superiority of our approach in capturing and preserving semantic meaning during translation.

Our method can handle morphologically rich languages of different families. We validated our method on translation datasets of Manipuri→English, Tamil→English, and Marathi→English. Our method has been shown to handle different morphologicaly rich languages from different families effectively. Marathi belongs to Indo-Aryan, Tamil belongs to Dravidian, and Manipuri belongs to Tibeto-Burman language families.

Models	BLEU			
	MNI→EN	TA→EN	MR→EN	AVG
BPE (Sennrich et al., 2016)	23.00	19.93	21.26	21.73
Morfessor (Grönroos et al., 2014)	23.74	21.79	22.52	22.68
Vanilla Character	23.77	21.70	21.81	22.09
Fully Character-based (Libovický et al., 2021)	21.96	21.76	19.61	21.44
Our Method	**25.61**	**22.56**	**24.04**	**24.07**

Table 4: BLEU scores on Mni→En, Ta→En and Mr→En test sets. "BPE" refers to (Sennrich et al., 2016); Morfessor refers to (Grönroos et al., 2014); Fully Character-based refers to (Libovický et al., 2021)

4.4 Ablation on Compositionally Derived Word Representation from Subwords

As our method preserves the word boundary information, we hope our model can better translate. In order to gain insights on whether the empirical usefulness comes from using word representation derived from morphemes, we perform an ablation test. For "Fixed," we composed the higher level representation in fixed manner, rather than word-delimited . We take fixed size of 3. Results in Table 5 shows that the word boundary information is indeed useful for obtaining a significant improvement.

Model	BLEU	
	MNI→EN	TA→EN
Fixed	21.56	19.31
Word-delimited	25.61	22.56

Table 5: We compare " Word-delimited," our word-delimited composition; and "Fixed," fixed size delimited composition.

4.5 Impact of Different Composition Functions

Table 6 shows an investigation of various composition functions for deriving word representation from the morphemes. We evaluate our word representation model using a convolutional neural network (CNN) with max pooling, in addition to our composition function based on gated recurrent units (GRU).

4.6 Analysis of Time Consumption and Parameter Size

Table 7 presents a detailed examination of training and decoding times, as well as parameter sizes, in the Mni→En translation models.

Composition Functions	BLEU	
	MNI→EN	TA→EN
CNN	24.95	21.86
GRU	25.61	22.56

Table 6: We compare word representation derived using GRU and CNN("GRU", "CNN").

Models	Mni→En		
	TIME1	TIME2	PARAM
BPE	1.0	1.0	16.4M
Morfessor	0.9	0.9	18.8M
Vanilla Char	3.6	3.6	12.3M
Fully Char-based	2.5	2.5	13.8M
Our Method	1.4	1.4	18.9M

Table 7: Training, test time, and size of the model parameters in Mni→En translation models. "Time1" denotes the training time (in ratio), "Time2" denotes the decoding time (in ratio), and "Param" denotes the size of model parameters (M for million).

4.7 Effect of Target Sentence Length

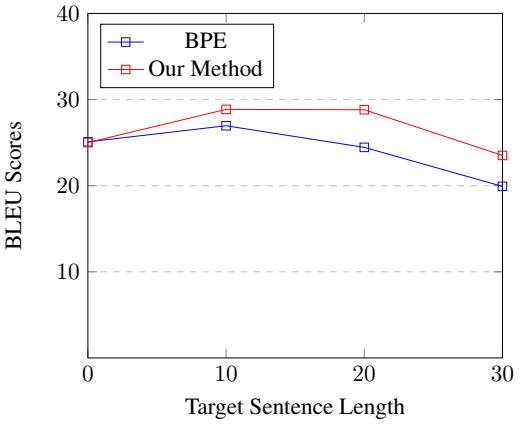

Figure 2: BLEU Scores over different sentence lengths

[Src] *Ishinggi Mamaldi* ***Ishinggi awaba*** *Myoknduna Hingliba Mioiduna Khangi.*
[Ref] *the value of water is understood by those who face* ***water scarcity****.*
[BPE] Undestanding the value of water in this water rule is very important
[Our Method] the value of water is understood by people who face water scarcity.

Table 8: Case study of Manipuri-to-English(transliterated) translation generated by BPE model and Our method

We also examined the impact of our method on Mni→En dataset by dividing test set into four groups based on the length. The result, shown in Figure 2, indicated that for longer sentences, our method had a significant improvement in the BLEU score compared to the BPE model. The improvement in BLEU score became more pronounced as the length of target translations increase, indicating that our method is effective for longer sentences.

4.8 Qualitative Analysis

In the translation example in Table 8, we illustrate the effectiveness of our method in comparison to the BPE baseline. Our method is able to accurately translate "Ishinggi awaba" as "water scarcity," while the base model fails to capture this and translates it to "water rule." This is likely due to the ability of capturing subword morphological information in our method.

5 Discussion

Our novel method for NMT in morphologically rich languages differs from previous approaches. It combines subword segmentation with word boundary info, using bidirectional GRU to retain word-level context and capture morphological features. This approach effectively addresses data sparsity and outperforms BPE and Morfessor-based models, as shown by improved translation accuracy on multiple language pairs. Its flexibility and capability to handle diverse languages highlight its potential for enhancing NMT in various linguistic settings.

6 Conclusion

In conclusion, we have presented a novel approach to address the challenges posed by morphologically rich languages in Neural Machine Translation (NMT). By combining word segmentation into subword units with the subsequent composition of these subword representations back to the word level, we effectively tackle data sparsity and word boundary issues. Our extensive experiments on various language pairs, including Manipuri-English, Tamil-English, and Marathi-English, have demonstrated the superiority of our proposed model over baseline subword-level NMT approaches in terms of translation accuracy.

The key contribution of our work lies in leveraging the inductive bias derived from word boundaries and interrelationships between word morphemes. This leads to remarkable enhancements in translation quality and significantly improves the performance of NMT for morphologically rich languages. Our findings emphasize the importance of harnessing subword information while preserving word-level knowledge. Moving forward, this approach opens up exciting avenues for further research and could potentially be extended to other language processing tasks, driving advancements in the field of natural language translation and understanding.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondrej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta Ruiz Costa-jussà, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Conference on Machine Translation*.

Duygu Ataman, Wilker Aziz, and Alexandra Birch. 2019. A latent morphology model for open-vocabulary neural machine translation. *ArXiv*, abs/1910.13890.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Loïc Barrault, Magdalena Biesialska, Ondrej Bojar, Marta Ruiz Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow,

Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi kiu Lo, Nikola Ljubesic, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Conference on Machine Translation*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

J. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society*, 18 5-6:602–10.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *International Conference on Computational Linguistics*.

Barry Haddow and Faheem Kirefu. 2020. Pmindia - a collection of parallel corpora of languages of india. *ArXiv*, abs/2001.09907.

Matthias Huck, Simon Riess, and Alexander M. Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Conference on Machine Translation*.

Tom Kocmi, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popovic. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Conference on Machine Translation*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Jindřich Libovický, Helmut Schmid, and Alexander M. Fraser. 2021. Why don't people use character-level machine translation? In *Findings*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *ArXiv*, abs/1604.00788.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *North American Chapter of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Peyman Passban, Qun Liu, and Andy Way. 2018a. Improving character-based decoding using target-side morphological information for neural machine translation. *ArXiv*, abs/1804.06506.

Peyman Passban, Andy Way, and Qun Liu. 2018b. Tailoring neural architectures for translating from morphologically rich languages. In *International Conference on Computational Linguistics*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Conference on Machine Translation*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. *ArXiv*, abs/1606.02892.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Telem Joyson Singh, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2021. English-manipuri machine translation: An empirical study of different supervised and unsupervised methods. *2021 International Conference on Asian Language Processing (IALP)*, pages 142–147.

Shashank Siripragrada, Jerin Philip, Vinay P. Namboodiri, and C. V. Jawahar. 2020. A multilingual parallel corpora collection effort for indian languages. *ArXiv*, abs/2007.07691.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ales Tamchyna, Marion Weller-Di Marco, and Alexander M. Fraser. 2017. Modeling target-side inflection in neural machine translation. In *Conference on Machine Translation*.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *ArXiv*, abs/2106.12672.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2016. Word representation models for morphologically rich languages in neural machine translation. In *SWCN@EMNLP*.