On the Importance of Word Boundaries in Character-level Neural Machine Translation

Anonymous ACL submission

Abstract

Neural Machine Translation (NMT) models generally perform translation using a fixed-size lexical vocabulary, which is an important bottleneck on their generalization capability and overall translation quality. The standard approach to overcome this limitation is to segment words into subword units, typically using some external tools with arbitrary heuristics, resulting in vocabulary units not optimized for the translation task. Recent studies have shown that the same approach can be extended to perform NMT directly at the level of characters, which can deliver translation accuracy on-par with subword-based models, on the other hand, this requires relatively deeper networks. In this paper, we propose a more computationally-efficient solution for character-level NMT which implements a hierarchical decoding architecture where translations are subsequently generated at the level of words and characters. We evaluate different methods for open-vocabulary NMT in the machine translation task from English into five languages with distinct morphological typology, and show that the hierarchical decoding model can reach higher translation accuracy than the subword-level NMT model using significantly fewer parameters, while demonstrating better capacity in learning longer-distance context and grammatical dependencies than the standard character-level NMT model.

1 Introduction

Neural Machine Translation (NMT) models are typically trained using a fixed-size lexical vocabulary. In addition to controlling the computational load, this limitation also serves to maintain better distributed representations for the most frequent set of words included in the vocabulary. On the other hand, rare words in the long tail of the lexical distribution are often discarded during translation since they are not found in the vocabulary. The prominent approach to overcome this limitation is to segment words into subword units (Sennrich et al., 2016) and perform translation based on a vocabulary composed of these units. However, subword segmentation methods generally rely on statistical heuristics that lack any linguistic notion. Moreover, they are typically deployed as a pre-processing step before training the NMT model, hence, the predicted set of subword units are essentially not optimized for the translation task. Recently, Cherry et al. (2018) extended the approach of NMT based on subword units to implement the translation model directly at the level of characters, which could reach comparable performance to the subword-based model, although this would require much larger networks which may be more difficult to train. The major reason to this requirement may lie behind the fact that treating the characters as individual tokens at the same level and processing the input sequences in linear time increases the difficulty of the learning task, where translation would then be modeled as a mapping between the characters in two languages. The increased sequence lengths due to processing sentences as sequences of characters also augments the computational cost, and a possible limitation, since sequence models typically have limited capacity in remembering long-distance context.

In many languages, words are the core atomic units of semantic and syntactic structure, and their explicit modeling should be beneficial in learning distributed representations for translation. There have been early studies in NMT which proposed to perform translation at the level of characters while also regarding the word boundaries in the translation model through a hierarchical decoding procedure, although these approaches were generally deployed through hybrid systems, either as a back-off solution to translate unknown words (Luong and Manning, 2016), or as pre-trained components (Ling et al., 2015). In this paper, we explore the benefit of achieving character-level NMT by processing sentences at multi-level dynamic time steps defined by the word boundaries, integrating a notion of explicit hierarchy into the decoder. In our model, all word representations are learned compositionally from character embeddings using bi-directional recurrent neural networks (bi-RNNs) (Schuster and Paliwal, 1997), and decoding is performed by generating each word character by character based on the predicted word representation through a hierarchical beam search algorithm which takes advantage of the hierarchical architecture while generating translations.

We present the results of an extensive evaluation comparing conventional approaches for open-vocabulary NMT in the machine translation task from English into five morphologically-rich languages, where each language belongs to a different language family and has a distinct morphological typology. Our findings show that using the hierarchical decoding approach, the NMT models are able to obtain higher translation accuracy than the subword-based NMT models in many languages while using significantly fewer parameters, where the character-based models implemented with the same computational complexity may still struggle to reach comparable performance. Our analysis also shows that explicit modeling of word boundaries in character-level NMT is advantageous for capturing longer-term contextual dependencies and generalizing to morphological variations in the target language.

2 Neural Machine Translation

In this paper, we use recurrent NMT architectures based on the model developed by Bahdanau et al. (2014). The model essentially estimates the conditional probability of translating a source sequence $x = (x_1, x_2, \ldots x_m)$ into a target sequence $y = (y_1, y_2, \ldots y_n)$, using the decomposition

$$p(y|x) = \prod_{j=1}^{n} p(y_j|y_{<j}, x_m, .., x_1) \qquad (1)$$

where $y_{<j}$ is the target sentence history defined by the sequence $\{y_1...y_{j-1}\}$.

The inputs of the network are *one-hot* vectors representing the tokens in the source sentence, which are binary vectors with a single bit set to 1 to identify a specific token in the vocabulary. Each one-hot vector is then mapped to a dense continuous representation, *i.e.* an embedding, of the source tokens via a look-up table. The representation of the source sequence is computed using a multi-layer bi-RNN, also referred as the *encoder*, which maps *x* into *m* dense vectors corresponding to the hidden states of the last bi-RNN layer updated in response to the input token embeddings.

The generation of the translation of the source sentence is called *decoding*, and it is conventionally implemented in an auto-regressive mode, where each token in the target sentence is generated based on an sequential classification procedure defined over the target token vocabulary. In this decoding architecture, a unidirectional recurrent neural network (RNN) predicts the most likely output token y_i in the target sequence using an approximate search algorithm based on the previous target token y_{i-1}, represented with the embedding of the previous token in the target sequence, the previous decoder hidden state, representing the sequence history, and the current attention context in the source sequence, represented by the *context vector* c_t. The latter is a linear combination of the encoder hidden states, whose weights are dynamically computed by a dot product based similarity metric called the *attention model* (Luong et al., 2015).

The probability of generating each target word y_i is estimated via a softmax function

$$p(y_i = z_j|x; \theta) = \frac{e^{z_j^T o_i}}{\sum_{k=1}^{K} e^{z_k^T o_i}} \qquad (2)$$

where z_j is the j^{th} one-hot vector of the target vocabulary of size K, and o_i is the decoder output vector for the i^{th} target word y_i. The model is trained by maximizing the log-likelihood of a parallel training set via stochastic gradient-descent (Bottou, 2010), where the gradients are computed with the back propagation through time (Werbos, 1990) algorithm.

Due to the softmax function in Equation 2, the size of the target vocabulary plays an important role in defining the computational complexity of the model. In the standard architecture, the embedding matrices account for the vast majority of the network parameters, thus, the amount of embeddings that could be learned and stored efficiently needs to be limited. Moreover, for many words corresponding to the long tail of the lexical distribution, the model fails in learning accurate embeddings, as they are rarely observed in varying context, leading the model vocabulary to typically include the most frequent set of words in the target language. This creates an important bottleneck over the vocabulary coverage of the model, which is especially crucial when translating into low-resource and morphologically-rich languages, which often have a high level of sparsity in the lexical distribution.

The standard approach to overcome this limitation has now become applying a statistical segmentation algorithm on the training corpus which splits words into smaller and more frequent *subword* units, and building the model vocabulary composed of these units. The translation problem is then modeled as a mapping between sequences of subword units in the source and target languages (Sennrich et al., 2016; Wu et al., 2016; Ataman et al., 2017). The most popular statistical segmentation method is Byte-Pair Encoding (BPE) (Sennrich et al., 2016), which finds the optimal description of a corpus vocabulary by iteratively merging the most frequent character sequences. One problem related to the subword-based NMT approach is that segmentation methods are typically implemented as pre-processing steps to NMT, thus, they are not optimized simultaneously with the translation task in an end-to-end fashion. This can lead to morphological errors at different levels, and cause loss of semantic or syntactic information (Ataman et al., 2017), due to the ambiguity in subword embeddings. In fact, recent studies have shown that the same approach can be extended to implement the NMT model directly at the level of characters, which could reach comparable performance to the subword-based NMT models. However, this would typically require increasing the computational cost of the model, defined by the network parameters (Kreutzer and Sokolov, 2018; Cherry et al., 2018). Figure 1a illustrates translation by implementing a 3-layer NMT decoder directly at the level of characters, where the attention mechanism and the RNNs modeling the target language process each sentence as a sequence of characters.

3 Hierarchical Decoding

In this paper, we propose to perform character-level decoding in NMT by modeling translation through a hierarchical architecture (Luong and Manning, 2016). In this architecture, the input embedding layer of the decoder is augmented with a character-level bi-RNN, which estimates a composition function over the embeddings of the characters in each word to compute distributed representations of target words in the sentence.

Given a bi-RNN with a forward (f) and backward (b) layer, the word representation $\mathbf{w}$ of a token of t characters is computed from the hidden states $\mathbf{h}_f^t$ and $\mathbf{h}_b^0$, *i.e.* the final outputs of the forward and backward RNNs, as follows:

$$\mathbf{w} = \mathbf{W}_f \mathbf{h}_f^t + \mathbf{W}_b \mathbf{h}_b^0 + \mathbf{b} \tag{3}$$

where $\mathbf{W}_f$ and $\mathbf{W}_b$ are weight matrices associated to each RNN and $\mathbf{b}$ is a bias vector. The embeddings of characters and the parameters of the word composition layer are jointly learned while training the NMT model. Since all target word representations are computed compositionally, the hierarchical decoding approach eliminates the necessity of storing word embeddings, significantly reducing the number of parameters.

Each word in the target sentence is predicted as in the standard architecture using the compositional target word representations, target sentence history and the context vector computed by the attention mechanism by a word-level RNN. However, instead of classifying the predicted target word in the vocabulary, its distributed representation is fed to a character-level RNN to generate the surface form of the word one character at a time by modeling the probability of observing the k_{th} character of the j_{th} word with length l, $p(y_{j,k}|y_{<j}, y_{j,<k})$, given the previous words in the sequence and the previous characters in the word.

The translation probability is then decomposed as:

$$p(y|x) = \prod_{j=1}^{n} \prod_{k=1}^{l} p(y_{j,k}|y_{j,<k}, y_{<j}, x_{<m}) \tag{4}$$

Similar to Luong and Manning (2016), the information necessary to generate the surface form is encoded into the attentional vector $\hat{h}_t$:

$$\hat{h}_t = \tanh(W[c_t; h_t]) \tag{5}$$

where h_t is the hidden state of the word-level RNN representing the current target context. The attentional vector is used to initialize the character RNN, and after the generation of the first character in the word, character decoding continues in an

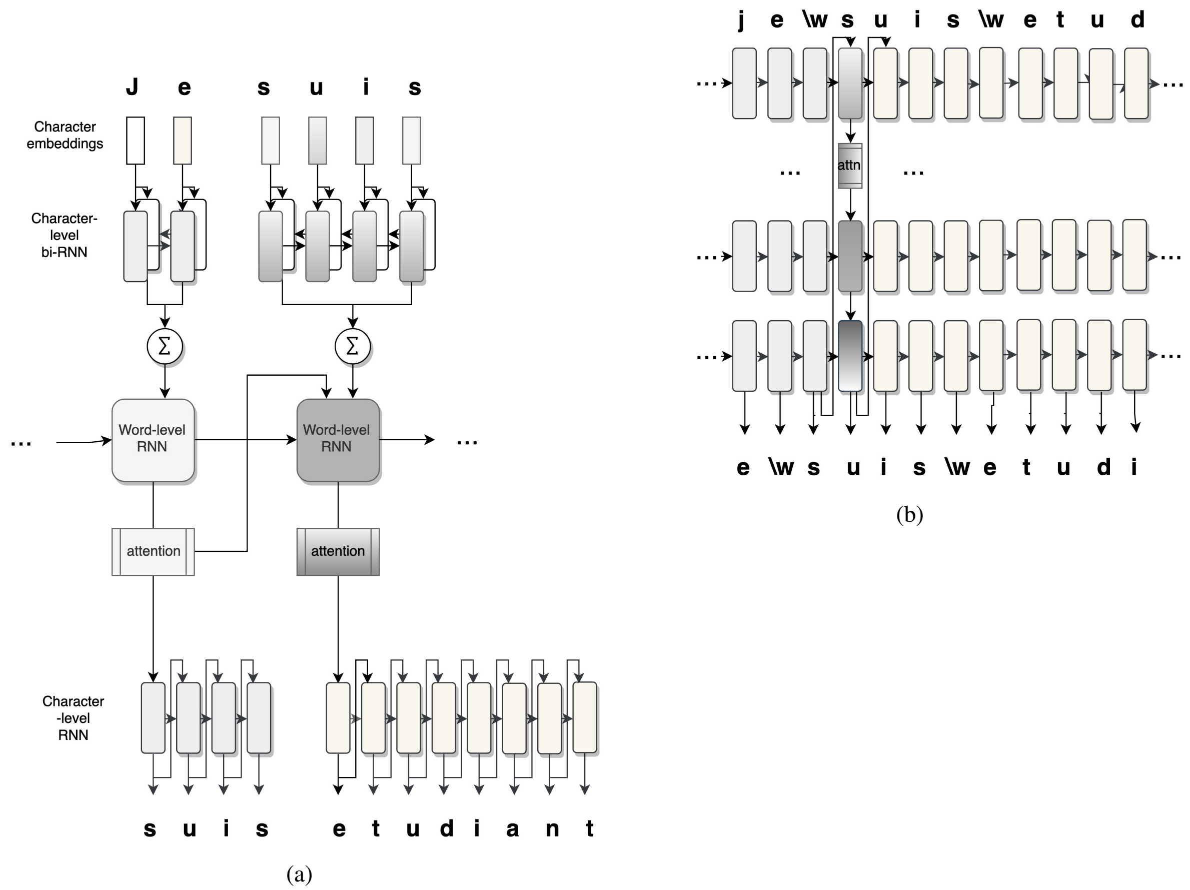

Figure 1: (a) Hierarchical NMT decoder: input words are encoded as character sequences and the translation is predicted at the level of words. The output words are generated as character sequences. (b) Character-level NMT decoder: the next token in the sentence is predicted by computing the attention weights and the target context repetitively for each character in the sentence.

auto-regressive mode, where the embedding of the each character is fed to the RNN to predict the next character in the word. The decoder consecutively iterates over the words and characters in the target sentence, where each RNN is updated at dynamic time steps based on the word boundaries.

4 Hierarchical Beam Search

In order to achieve efficient decoding with the hierarchical NMT decoder, we implement a **hierarchical beam search** algorithm, described in Algorithm 1. The algorithm starts decoding by predicting the B most likely characters and storing them in a character beam along with their probabilities. Different than the standard algorithm, the beams are reset each time the generation of a word is complete and the B most likely words computed after beam search are stored in an intermediate word-level beam. The word beam is used to compute the B distributed representations corresponding to the most likely B next target words, which are fed to the character RNN to continue the beam search. When the beam search is complete, the most likely character sequence is generated as the best hypothesis.

function HierarchicalBeamSearch(X)
WordBeam $\leftarrow \{\}$
for i = 0 ... B do:
  CharBeam $\leftarrow \{\}$
  $\hat{Y}_i \leftarrow$ WordRNN$_{Fwd}$(X, WordBeam[i])
  for k = 0 ... B do:
    $\hat{y_k} \leftarrow$ CharRNN$_{Fwd}(\hat{Y}_i)$
    CharBeam[k] $\leftarrow$ CharBeam[k-1] $\cup$ $TopB$(softmax($\hat{y_k}$))
  WordBeam[i] $\leftarrow$ WordBeam[i-1] $\cup$ $TopB$(CharBeam)
return Top(WordBeam)

Algorithm 1: The hierarchical beam search algorithm. $\hat{Y}_i$ and $\hat{y_k}$ are the current word and character predictions, and B is the beam size.

5 Experiments

We evaluate decoding architectures using different levels of granularity in the vocabulary units and the attention mechanism, including the standard decoding architecture implemented either with subword (Sennrich et al., 2016) or fully character-level (Cherry et al., 2018) units, which constitute the baseline approaches, and the hierarchical decoding architecture, by implementing all in Pytorch (Paszke et al., 2017) within the OpenNMT-py framework (Klein et al., 2017). In order to evaluate how each generative method performs in languages with different morphological typology, we model the machine translation task from English into five languages from different language families and exhibiting distinct morphological typology: Arabic (*templatic*), Czech (*mostly fusional, partially agglutinative*), German (*fusional*), Italian (*fusional*) and Turkish (*agglutinative*). We use the TED Talks corpora (Cettolo et al., 2012) for training the NMT models, which range from 110K to 240K sentences, and the official development and test sets from IWSLT[1] (Cettolo et al., 2017). The low-resource settings for the training data allows us to examine the quality of the internal representations learned by each decoder under high data sparseness. The details of the statistical characteristics of training data are given in Table 1. In order to evaluate how the performance of each method scales with increasing data size, we evaluate the models also by training with a multi-domain training data using the public data sets from WMT[2] (Bojar et al., 2016) in the English-to-German direction, followed by an analysis on each model's capability in generalizing to morphological variations in the target language, using the Morpheval (Burlot et al., 2018) evaluation sets. The details of the resulting multi-domain training corpus can be seen in Table 2.

All models are implemented using gated recurrent units (GRU) (Cho et al., 2014) with the same number of parameters. The hierarchical decoding model implements a 3-layer GRU architecture, which is compared with a fully character-level decoder which also uses a 3-layer stacked GRU architecture. The subword-level decoder has a 2-layer stacked GRU architecture, to account also for the larger number of embedding parameters. The models using the standard architecture have the attention mechanism after the first GRU layer, and have residual connections after the second layer (Barone et al., 2017). The hierarchical decoder implements the attention mechanism after the second layer in order to compute the context vector at the level of words, and has a residual connection between the first and second layers.

Lang. Pair	# sents (K)	# tokens (M)		# types (K)	
		Src	Tgt	Src	Tgt
EN-AR	238	5	4	120	220
EN-CS	118	2	2	50	118
EN-DE	212	4	4	69	144
EN-IT	185	4	3	63	95
EN-TR	136	2	3	53	171

Table 1: Training sets in the TED Talks benchmark. Development and test sets are on average 50K to 100K tokens. (*M*: Million, *K*: Thousand.)

Lang. Pair	# sents (M)	# tokens (M)		# types (K)	
		Src	Tgt	Src	Tgt
EN-DE	5	119	114	106	152

Table 2: Multi-domain training set (*M*: Million, *K*: Thousand.)

The source sides of the data used for training all NMT models, and the target sides of the data used in training the subword-level NMT models are segmented using BPE with 16,000 merge rules on the IWSLT data, and 32,000 on WMT. The models use an embedding and hidden unit size of 512 under low-resource and 1024 under high-resource settings, and are trained using the Adam optimizer with a learning rate of 0.0003 and decay of 0.9, a batch size of 100 and a dropout rate of 0.2. Decoding in all models is performed with a beam size of 5. The accuracy of each output is measured in terms of the BLEU metric (Papineni et al., 2002) and the significance of the improvements are computed using bootstrap hypothesis testing (Wasserman and Bockenholt, 1989).

6 Results

The results of the experiments given in Table 3 show that the hierarchical decoder can reach performance comparable to or better than the NMT model based on subword units in all languages while using almost three times less number of pa-

[1] The International Workshop on Spoken Language Translation.

[2] The Conference on Machine Translation, with shared task organized for news translation.

Model	BLEU					Avg. Num. Params
	AR	CS	DE	IT	TR	
Subwords	14.67	16.60	**24.29**	26.23	8.85	22M
Characters	12.72	**16.94**	22.23	24.33	**10.65**	7.3M
Hierarchical	**15.55**	16.79	23.91	**26.64**	9.74	7.3M

Table 3: Results of the evaluation of models in translating languages with different morphological typology using the IWSLT data sets. The average number of parameters are calculated only for the decoders of the NMT models at a resolution of millions (M). The best scores for each translation direction are in bold font. All improvements over the baselines are statistically significant (p-value < 0.01).

rameters. The improvements are especially evident in Arabic and Turkish, languages with the most complex morphology, where the accuracy with the hierarchical decoder is **0.88** and **0.89** BLEU points higher, respectively, and comparable in Czech, Italian and German, which represent the fusional languages. In Czech, the hierarchical model outperforms the subword-based model by **0.19** BLEU and in Italian by **0.41** BLEU points. The subword-based NMT model achieves the best performance in German, a language that is rich in compounding, where explicit subword segmentation might allow learning better representations for translation units.

The fully character-level NMT model, on the other hand, obtains higher translation accuracy than the hierarchical model in Turkish, with an improvement of **0.91** BLEU, and in Czech with **0.15** BLEU points. As can be seen in the statistical characteristics of the training sets in Table 1, or a better illustration of it by plotting the token-to-type ratios in each language (Figure 2), these two directions constitute the most sparse settings, where Turkish has the highest amount of sparsity in the benchmark, followed by Czech, and the improvements seem to be proportional to the amount of sparsity in the language. This suggests that in case of high lexical sparsity, learning to translate based on representations of characters might aid in reducing contextual sparsity, allowing to learn better distributed representations. As the training data size increases, one would expect the likelihood of observing rare words to decrease, especially in languages with low morphological complexity, along with the significance of representing rare and unseen words (Cherry et al., 2018). Our results support this hypothesis, where decreasing lexical sparsity, either in the form of the training data size, or the morphological complexity of the target language, eliminates the advantage of character-level translation. In Arabic and Italian, where the training data is almost twice as large as the other languages, using the hierarchical model provides improvements of **2.83** and **2.31** BLEU points over the character-level NMT model. In German, the fully character-level NMT model still achieves the lowest accuracy, with **2.06** BLEU points below the subword-based model. This might be due to the increased level of contextual ambiguity leading to difficulty in learning reliable character embeddings when the model is trained over larger corpora. Another factor which might affect the lower performance of character-level models is the average sentence

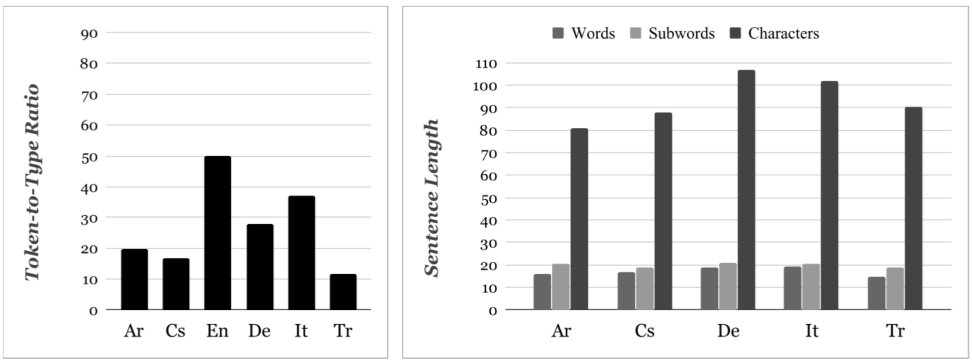

Figure 2: Lexical sparsity and average sentence lengths in different languages.

Variation	Chars	Subwords	Hier.
Paradigm contrast features			
Positive vs. comparative adjective	**71.4**	68.4	70.1
Present vs. future tense	85.7	**92.0**	90.6
Negation	**97.8**	97.0	94.8
Singular vs. plural noun	88.2	**88.8**	88.6
Present vs. past tense	92.0	93.3	**95.4**
Compound generation	60.2	**65.4**	57.8
Indicative vs. conditional mode	86.4	88.2	**92.3**
Average	83.1	**84.7**	84.2
Agreement features			
Pronoun vs. Nouns (gender)	96.5	97.4	**98.8**
Pronoun vs. Nouns (number)	95.4	**96.0**	93.4
Pronoun (plural)	88.6	**94.3**	92.2
Pronoun (relative-gender)	74.2	76.4	**78.9**
Pronoun (relative-number)	84.2	**90.2**	87.0
Positive vs. superlative adjective	76.2	68.2	**80.4**
Simple vs. coordinated verbs (number)	**96.4**	93.4	97.2
Simple vs. coordinated verbs (person)	92.3	92.8	**93.5**
Simple vs. coordinated verbs (tense)	82.4	86.0	**90.2**
Average	87.4	88.3	**90.17**

Table 4: Results of the evaluation of models in capturing morphological variations in the output using the Morpheval English-German test set. The accuracy is measured with the percentage of correctly captured morphological contrasts. The best scores for each translation direction are in bold font.

lengths, which are much longer compared to the sentence lengths resulting from with subword segmentation (Figure 2).

In the experiments conducted in the English-to-German translation direction, the results of which are given in Table 5, accuracy obtained with the hierarchical and subword-based NMT decoders significantly increase with the extension of the training data, where the subword-based model obtains the best accuracy, followed by the hierarchical model, and the character-level NMT model obtains significantly lower accuracy compared to both approaches. Studies have shown that character-level NMT models could potentially reach the same performance with the subword-based NMT models (Cherry et al., 2018), although this might require increasing the capacity of the network. On the other hand, the consistency in the accuracy obtained using the hierarchical decoding model from low to mid resource settings suggests that explicit modeling of word boundaries aids in achieving a more computationally efficient solution to character-level translation.

Since solely relying on BLEU scores may not be sufficient in understanding the generative properties of different NMT models, we perform an additional evaluation in order to assess the capacity of models in learning syntactic or morphological dependencies using the Morpheval test suites, which consist of sentence pairs that differ by one morphological contrast, and each output accuracy is measured in terms of the percentage of translations that could convey the morphological contrast in the target language. Table 4 lists the performance of different NMT models implementing decoding at the level of subwords, characters, or hierarchical word-character units in capturing variances in each individual morphological paradigm and preserving the agreement between inflected words and their dependent lexical items. The results of our analysis support the benefit of

Model	newstest15
Subwords	**22.71**
Characters	20.34
Hierarchical	22.19

Table 5: Experiment results in the English-to-German direction with WMT data sets. Translation accuracy is measured with BLEU. Best scores are in bold font.

Input	when a friend of mine told me that I needed to see this great video about a guy protesting bicycle fines in New York City, I admit I wasn't very interested.
Output *Subword-based Decoder*	**bir arkadaşım New York'ta bisiklet protestosunu** protesto etmek **için bu filmi izlemeye ihtiyacım olduğunu söylemişti.**
Output *Character-based Decoder*	**bana bir arkadaşım** bana **New York'ta bir adam ile ilgili** bir adam hakkında **görmem gereken bir** adam hakkında görmem gerektiğini **söyledi.**
Output *Hierarchical Decoder*	**bir arkadaşım New York'ta bisiklet yapmaya ihtiyacım olduğunu söylediği zaman, kabul ettim**.
Reference	bir arkadaşım New York şehrindeki bisiklet cezalarını protesto eden bir adamın bu harika videosunu izlemem gerektiğini söylediğinde, kabul etmeliyim ki çok da ilgilenmemiştim.

Table 6: Example translations with different approaches in Turkish

using BPE in German as a subword segmentation algorithm, which obtains the highest accuracy in most of the morphological paradigm generation tasks, although the character-level model shows to be promising in capturing some morphological features better than the former, such as negation or comparative adjectives. In capturing syntactic agreement features, the hierarchical decoding model performs much better than the subword and character-level models, which is likely due to processing the sentence context at the word level, inducing a better notion of syntactic ordering during generation.

In order to better illustrate the differences in the outputs of each NMT model, we also present some sample translations in Table 6, obtained by translating English into Turkish using the NMT models trained on the TED Talks corpus. The input sentences are selected such that they are sufficiently long so that one can see the ability of each model in capturing long-distance dependencies in context. The input sentence is from a typical conversation, which requires remembering a long context with many references. We highlight the words in each output that is generated for the first time. Most of the models fail to generate a complete translation, starting to forget the sentence history after the generation of a few words, indicated by the start of generation of repetitions of the previously generated words. The character-level decoder seems to have the shortest memory span, followed by the subword-based decoder, which completely omits the second half of the sentence. Despite omitting the translations of the last four words in the input and some lexical errors, the hierarchical decoder is the only model which can generate a meaningful and grammatically-correct sentence, suggesting that modeling translation based on a context defined at the lexical level might help to learn better grammatical and contextual dependencies, and remembering longer history.

7 Conclusion

In this paper, we explored the idea of performing the decoding procedure in NMT in a multi-dimensional search space defined by word and character level units via a hierarchical decoding structure and beam search algorithm. Our model obtained comparable to better performance than the conventional open-vocabulary NMT solutions, such as character and subword-level models, in many languages while using a significantly smaller number of parameters, showing promising application under high-resource settings. Our software will be available for public usage after publication.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

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

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep architectures for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 99–107.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Léon Bottou. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proceedings of 19th International Conference on Computational Statistics (COMPSTAT)*, pages 177–186. Springer.

Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen, and François Yvon. 2018. The wmt'18 morpheval test suites for english-czech, english-german, english-finnish and turkish-english.

Mauro Cettolo, Girardi Christian, and Federico Marcello. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of European Association for Machine Translation*, pages 261–268.

Mauro Cettolo, Federico Marcello, Bentivogli Luisa, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, pages 103–111.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Julia Kreutzer and Artem Sokolov. 2018. Learning to segment inputs for nmt favors character-level processing.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

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)*, volume 1, pages 1715–1725.

Stanley Wasserman and Ulf Bockenholt. 1989. Bootstrapping: Applications to psychophysiology. *Psychophysiology*, 26(2):208–221.

Paul J Werbos. 1990. Backpropagation Through Time: What it does and How to do it. In *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, volume 78, pages 1550–1560.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Googles neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*.