Multitask Learning For Different Subword Segmentations In Neural Machine Translation

Tejas Srinivasan, Ramon Sanabria, Florian Metze

Language Technologies Institute
Carnegie Mellon University, USA
tsriniva, ramons, fmetze@cs.cmu.edu

Abstract

In Neural Machine Translation (NMT) the usage of subwords and characters as source and target units offers a simple and flexible solution for translation of rare and unseen words. However, selecting the optimal subword segmentation involves a trade-off between expressiveness and flexibility, and is language and dataset-dependent. We present Block Multitask Learning (BMTL), a novel NMT architecture that predicts multiple targets of different granularities simultaneously, removing the need to search for the optimal segmentation strategy. Our multi-task model exhibits improvements of up to 1.7 BLEU points on each decoder over single-task baseline models with the same number of parameters on datasets from two language pairs of IWSLT15 and one from IWSLT19. The multiple hypotheses generated at different granularities can be combined as a post-processing step to give better translations, which improves over hypothesis combination from baseline models while using substantially fewer parameters.

1. Introduction

Neural Machine Translation (NMT) [1, 2, 3] provides a simple, end-to-end framework for translating text from one language to another. NMT approaches have largely outperformed and replaced previous statistical translation methods. Traditionally, NMT systems used words as source and target units, which have three main disadvantages. First, word-based models are unable to translate rare and out of vocabulary (OOV) words in the source language. Second, they can not produce unseen target words, such as morphological variants of observed words (*e.g.*, deriving realistic from real). Third, they have to handle large source and target language vocabularies (*i.e.*, large look-up matrices), which makes them less scalable in term of computation and memory. A large vocabulary also implies data sparsity where the number of tokens is not balanced.

A common solution for the problems mentioned above is to perform word segmentation. The Byte-Pair Encoding (BPE) algorithm [4] groups units together according to their frequency. By presetting the desired vocabulary size, the BPE algorithm generates a segmentation of the data by

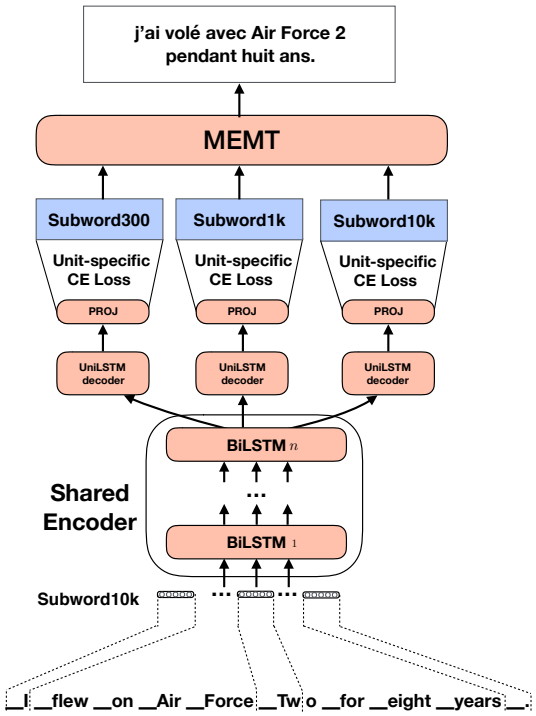


Figure 1: Our Block Multitask Learning (BMTL) Model learns to translate the same sentence in different subword-level units with multiple granularities at the same time. Finally, a Multi-Engine Machine Translation (MEMT) system combines all of them.

representing words as a collection of subword units. More recently, [5] proposed a forward probabilistic subword segmentation algorithm which is based on a unigram language model, in contrast to the deterministic BPE algorithm. Unlike BPE, the unigram language model is capable of providing multiple subwords with probabilities, thereby making the segmentation more flexible. However, unigrams did not exhibit significant improvement over BPE.

Subword-level NMT systems resolve the drawbacks of word-level models by providing open-vocabulary capabilities to the model and reducing the vocabulary size considerably. However, subword-level systems also have some

drawbacks. Most importantly, input and target sequences are longer, which makes them slower to train and decode, and implies long-range dependencies which are difficult to model [6]. Also, the open-vocabulary capabilities of the model might generate undesired variants of correct translations. Finally, subword embeddings do not carry as much semantic information as words and therefore modeling this information becomes much more difficult.

All units discussed above present a trade-off between flexibility and semantic information (*i.e.*, characters are more flexible with less semantic information, whereas words can not translate OOV words but contain more semantic information). This trade-off makes the selection of optimal segmentation a non-trivial problem, for a given dataset and language pair. Generally, the optimal segmentation is treated as a hyper-parameter that needs to be found by brute-force search, and this search is time-consuming and error-prone. This problem is even more emphasized in multilingual settings where the optimal segmentation needs to be found for each language [7].

In this paper, we propose a block multitask learning (BMTL) model that, by using multiple subword segementations in the target domain, translates the same input with different granularities (see Fig. 1). All hypotheses are combined posteriorly with Multi-Engine Machine Translation (MEMT) system [8], which generates the final hypothesis of the system. Our experiments show that, in general, each output segmentation of our BMTL outperforms all single task approaches that use the same number of parameters. By combining the outputs of BMTL with MEMT, our system still outperforms the combination of single-task models, in spite of using lesser parameters in total. We hypothesize that sharing an encoder among different decoder-specific granularities, makes the encoded representation more general and robust, which yields a better translation and therefore an improvement in BLEU score.

The main contributions of this paper are as follows:

- We introduce Block Multitask Learning (BMTL), an NMT framework that, by using multiple subword segmentations in the target domain, translates the same input with different granularities (see Fig. 1) (Section 2.2).

- We present a set of experiments in three different IWSLT language pairs (En-{Fr, Vi, Cs}) that show improvements on each output segmentation of BMTL, outperforming all single task approaches that use the same number of parameters. (Section 3.3)

- We show that by combining the outputs of BMTL with Multi-Engine Machine Translation (MEMT) [8], our system still outperforms the combination of single-task models, in spite of using fewer parameters in total (Section 3.4).

2. Architecture

In this section, we will first introduce our baseline model that consists of a standard attention-based encoder-decoder model [9] (Section 2.1). After that, we present our new BMTL architecture that uses the encoder-decoder model as the main building block (Section 2.2). Finally, we describe MEMT, the mechanism that we use for combining multiple hypotheses.

2.1. Baseline Model

Our baseline model is a standard encoder-decoder model with a multilayer perceptron attention and tanh activation [9]. The encoder is a bidirectional recurrent neural network with Gated Recurrent Units (BiGRU). This block of the system encodes the input subword embeddings. The decoder is also a recurrent neural network, but it uses Conditional GRU decoder [10]. The decoder, conditioned on the previously generated state and each encoded vector, generates an attention matrix that weighs all the hidden states generated by the encoder. The decoder continuously generates symbols until the end-of-sentence symbol is produced. This model can use different subword segmentations in the source and target space. We will refer to this henceforth as the baseline model.

2.2. Block Multitask Learning

In BMTL (see Fig. 1), we extend the baseline model with a multitask learning approach. More specifically, in this case, each task is the generation of the translation in the target language, in different granularities. All tasks share the same encoder as in the baseline model. The encoded matrix is processed by multiple decoders, all of which have the same architecture as the baseline decoder. Each of the decoders has its own attention and set of parameters.

More formally, a BMTL model with decoders outputting units of BPE300, BPE1000 and BPE10000, can be written as

$$\begin{aligned} \boldsymbol{e}_0 &= \text{Shared\_Encoder}(\boldsymbol{X}) \\ \boldsymbol{S}_{\text{bpe300}} &= \text{CGRU\_bpe300}(\boldsymbol{e}_0) \\ \boldsymbol{S}_{\text{bpe1k}} &= \text{CGRU\_bpe1k}(\boldsymbol{e}_0) \\ \boldsymbol{S}_{\text{bpe10k}} &= \text{CGRU\_bpe10k}(\boldsymbol{e}_0) \end{aligned} \tag{1}$$

where $\boldsymbol{S_n}$ is the generated hypothesis and CGRU_n is a decoder for the subword segmentation n.

During training, the losses obtained by all the decoders are normalized according to length, summed and averaged. We found that this approach works better than backpropagating each loss independently through its own decoder as well as the shared encoder. This allows the encoder to learn more generalized representations which are independent of the output subword granularity.

It is important to note that BMTL is a model agnostic technique and it can be easily ported to other architectures such as Transformer [11].

Corpus	Model	BMTL1			BMTL2		
		BPE300	BPE1K	BPE10K	BPE10K	BPE16K	BPE32K
En-Fr	Baseline	35.6	35.1	**36**	35	36	34.8
	BMTL	**36**	**35.6**	35.7	**36.5**	**36.1**	**36.5**
En-Vi	Baseline	26.4	27.1	26.3	27.6	27	27.3
	BMTL	**27**	**27.7**	**27.6**	**27.8**	**27.5**	**27.6**
En-Cs	Baseline	17	16.5	16.7	**16.7**	16.4	**16.4**
	BMTL	**17.6**	**17.7**	**17.4**	16.6	**16.8**	16.3

Table 1: BLEU scores of our BMTL1 (*i.e.*, BMTL combining BPE 300, BPE1K and BPE10K) and BMTL2 (*i.e.*, BMTL combining BPE 10K, BPE16K and BPE32K) models, as well as the baseline models, on each of our three IWSLT language pairs (*i.e.*, English to {French, Czech, Vietnamese}).

2.3. Multi-Engine Machine Translation

As a post-processing step, our system uses MEMT to combine all generated hypotheses [8]. MEMT uses a variant of the METEOR aligner to align all the results of each decoder. It applies four constraints to generate the combined hypothesis. First, the sentences must start with start-of-sentence symbol and end-of-sentence symbol. Second, a token is used only once. Third, it forces weak monotonicity between the alignments, preventing too many jumps from the search algorithm. Fourth, it forces the completeness of the combined hypothesis by not skipping tokens unless the sentence ends. It is important to note that MEMT uses tokenized-word hypotheses as input.

3. Experiments

3.1. Datasets and Preprocessing

We report results on two language pairs from the IWSLT 2015 TED Talks corpus - English to {French, Vietnamese}. We use the `tst2012` and `tst2013` sets as our development and testing sets. Furthermore, we also report results on the IWSLT 2019 text translation task from English to Czech. We use the provided training and development sets, and the `tst-COMMON` set for testing.

For preprocessing, each corpus is normalized, tokenized and truecased using Moses [12]. Each corpus is then segmented to different BPE vocabulary sizes using the sentencepiece implementation[1].

3.2. Implementation Details

All models are trained using Adam optimizer [13], with a learning rate of 0.0001, decay of 0.9 and batch size of 32. All models have 2 layers of bidirectional encoders of size 512 in each direction, decoder of size 1024, and input and output embeddings of size 512. We also apply dropout with probability 0.1 in the encoder and decoder. The norm of the gradient is clipped with a threshold of 1 [14]. All models are implemented using the nmtpytorch framework [15]. The output hypotheses are detokenized and detruecased using Moses, before using sacreBLEU [16] for scoring the translations. Finally, all hypothesis are combined with the MEMT implementation[2] with the default configuration provided.

3.3. Results

We experiment with two variants of the BMTL model. BMTL1 has inputs of BPE10K and decoders of BPE300, BPE1K and BPE10K (as seen in Figure 1). BMTL2 has inputs of BPE32K and decoders of BPE10K, BPE16K and BPE32K. We experiment with different input segmentations to show that our architecture shows improvements irrespective of the input unit. For each BMTL model, we also train three baseline encoder-decoder models - each with the same input units as BMTL and an output corresponding to one of the BMTL decoders. For instance, we compare the output of BMTL1's BPE300 decoder with an encoder-decoder model that has input units of BPE10K and output units of BPE300.

Table 1 shows the results of our experiments on the BMTL1 and BMTL2 models, as well as the baseline models. We observe that almost all of our BMTL decoders (in both BMTL1 and BMTL2) outperform the corresponding baseline models across all three languages, with an improvement of upto 2 BLEU points. This exhibits our architecture's ability to learn more robust encoded representations, irrespective of language, input units, and combination of output segmentations. [3]

These improvements are on models that have the same size as the baselines. Although at training time, the model includes multiple decoders and a shared encoder, while testing, we need to utilize only a single decoder and encoder, thus making it comparable to the baseline models. Each of our BMTL decoders also converges faster than the corresponding individual baseline models (Figure 2).

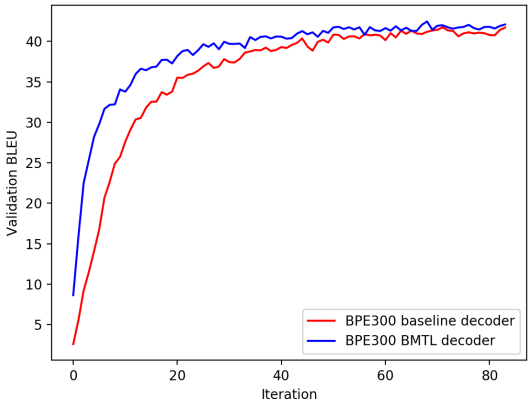
3.4. Hypothesis Combination

We explore the possibility of combining hypotheses from each of the decoders in BMTL (see Fig. 1). We use Multi-

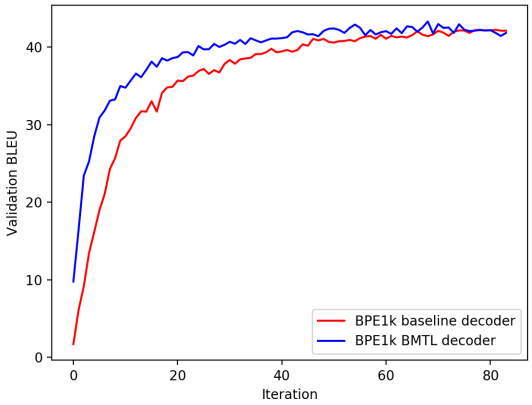
[1] `https://github.com/google/sentencepiece`

[2] `https://github.com/kpu/MEMT`

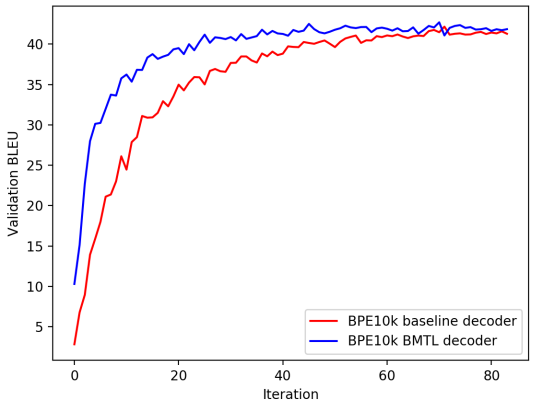
[3] All of our baseline numbers are comparable to numbers on the same datasets in [17].



(a) BLEU scores from BPE300 decoder



(b) BLEU scores from BPE1K decoder



(c) BLEU scores from BPE10K decoder

Figure 2: Number of iterations versus validation BLEU scores for decoders in our BMTL1 and Baseline1 models, on the IWSLT15 En-Fr dataset. We compare the number of iterations to converge between each of the decoders in our BMTL1 model and the corresponding Baseline1 model

Engine Machine Translation (MEMT) [8], to get a single hypothesis by combining the hypotheses of each BMTL decoder, the results of which can be seen in Table 2.

We see that the MEMT combination from the BMTL models almost always outperforms the baselines. Although these gains are slightly lower than the ones achieved by the individual BMTL decoders, they are achieved using models with significantly fewer parameters (greater than 20% reduction from the combined baseline models). This reduction is because the combined baseline models have multiple encoders, whereas BMTL has a shared encoder for each of the decoders.

4. Related Work

Multiple previous works have analyzed the differences between target units of different subword resolution in NMT systems. These works make the observation that the optimal segmentation depends on three elements: number of OOV words [4], language [5], and size of the model [18]. Each of these dependencies makes the task of finding the optimal subword segmentation computationally infeasible and prone to error.

In an attempt to solve this problem, [19] propose a dynamic end-to-end, data-driven segmentation. [19] uses the Adaptive Computation Time paradigm [20] to let the network learn an optimal segmentation. This approach, however, does not match nor overcome the BLEU score of any of the manual segmentations proposed. Our work, instead, benefits from having mutliple representations for the same input outperforming almost all single-segmentation baselines proposed.

To take advantage of the different available segmentations and solve the problem of OOV words, [21] proposes

Corpus	Model	BLEU	# param. (M)
En-Fr	Baseline1	37.4	92.03
	BMTL1	**37.5**	71.82
	Baseline2	37.3	149.78
	BMTL2	**37.8**	114.85
En-Vi	Baseline1	28.8	92.03
	BMTL1	**29**	71.82
	Baseline2	28.5	149.78
	BMTL2	**29**	114.85
En-Cs	Baseline1	18.1	92.03
	BMTL1	**18.7**	71.82
	Baseline2	**18.1**	149.78
	BMTL2	18	114.85

Table 2: BLEU scores using MEMT. We compare two systems. First, BMTL1 with Baseline1, that combines 300, 1K and 10K systems in BMTL and independently-trained baselines respectively. Second, BMTL2 with Baseline2 that combines 10k, 16k and 32K systems in BMTL and baselines respectively. # param. lists the number of trainable parameters in (M)illions. For baseline MEMT, we report the sum of the parameters of the independently-trained baselines.

a hybrid system that combines words and character-based models. Translation occurs primarily at the word level, and the system uses the character-level model when an unknown symbol is predicted. [21] is similar to our approach in its usage of multiple target unit segmentations. Even though [21] achieves similar improvements to ours, the incorporation of a second character-based architecture makes their approach more memory intensive and slow than the baseline model (*i.e.*, word-based NMT) at test time.

Perhaps more related to our work, there are two recent NMT approaches that combine multiple BPE segmentations. First, [22] sequentially increase the number of units during training each time the architecture converges. This method achieves comparable results to grid search, without the need of training the model a number of times. Second, [23] proposes summing the multiple subword embeddings from different segmentations to the same embedding layer. Although [22, 23] propose to use multiple target segmentations in the same system, there are multiple differences from our work. Even though [22] finds the optimal segmentation, they do not use concurrently different target units. [23] uses multiple representations in parallel in the input while we use ours in the output.

Related to the architecture of our model, [24] proposed the first work on multitask learning. Similarly, many approaches integrated other tasks in NMT models such as the translation of more languages [25], Part-Of-Speech tagging [26] or general syntax [27]. However, none of them combined different granularities of the same sentence. More recently, in Automatic Speech Recognition, [28] proposed also the use of multiple levels of segmentation in a hierarchical multitask learning structure. The improvements showed in [28] inspired this paper.

5. Conclusions And Future Work

We propose Block Multitask Learning (BMTL) model that translates the same input to multiple subword granularities in the target language and is trained in a multi-task learning fashion. Our BMTL decoders outperform the single-task baseline models across all languages, for different input units, and different combinations of output segmentations, while having comparable model size. We also use Multi-Engine Machine Translation to combine multiple decoders' hypotheses as a post-processing step; this also achieves improvements over combining single-task hypotheses, despite having significant fewer parameters.

With regards to how we can expand this work in the future, we are investigating several potential future directions for BMTL. First, we are investigating how we can effectively encode source sentences using different input segmentations (similar to [29]). Second, we are exploring if weighing each loss function provided by each decoder can help the model to learn a better representation. Third, we are exploring an online beam-search strategy that uses the hypothesis of all decoders. This technique will constitute a more elegant solution for joint decoding than MEMT. On a similar note, we want to explore an online decoding strategy during training that can selectively switch between decoders. Finally, while our models have so far been tested on small-scale IWSLT datasets, we plan to investigate the performance of our model on larger WMT datasets, and using the state-of-the-art Transformer network.

6. References

[1] M. L. Forcada and R. P. Ñeco, "Recursive hetero-associative memories for translation," in *Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Biological and Artificial Computation: From Neuroscience to Technology*, ser. IWANN '97. Springer-Verlag, 1997.

[2] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, 2014.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, 2014.

[4] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2016.

[5] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the Annual Meeting of*

the Association for Computational Linguistics (ACL). ACL, 2018.

[6] S. Hochreiter, Y. Bengio, and P. Frasconi, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *Field Guide to Dynamical Recurrent Networks*. IEEE Press, 2001.

[7] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, 2017.

[8] K. Heafield and A. Lavie, "CMU system combination in WMT 2011," in *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP*, 2011.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Computing Research Repository (CoRR)*, 2014.

[10] R. Sennrich, O. Firat, K. Cho, A. Birch-Mayne, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*. ACL, 2017.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*. NIPS, 2017.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, 2014.

[14] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR*, 2012.

[15] O. Caglayan, M. García-Martínez, A. Bardet, W. Aransa, F. Bougares, and L. Barrault, "Nmtpy: A flexible toolkit for advanced neural machine translation systems," *Prague Bull. Math. Linguistics*, 2017.

[16] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT*, 2018.

[17] K. Wolk and K. Marasek, "PJAIT systems for the IWSLT 2015 evaluation campaign enhanced by comparable corpora," *CoRR*, 2015.

[18] C. Cherry, G. Foster, A. Bapna, O. Firat, and W. Macherey, "Revisiting character-based neural machine translation with capacity and compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2018.

[19] J. Kreutzer and A. Sokolov, "Learning to segment inputs for nmt favors character-level processing," in *Proceedings of Workshop on Spoken Language Technologies (SLT)*. IEEE, 2018.

[20] A. Graves, "Adaptive computation time for recurrent neural networks," *CoRR*, 2016.

[21] M.-T. Luong and C. D. Manning, "Achieving open vocabulary neural machine translation with hybrid word-character models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2016.

[22] E. Salesky, A. Runge, A. Coda, J. Niehues, and G. Neubig, "Optimizing segmentation granularity for neural machine translation," *CoRR*, vol. abs/1810.08641, 2018.

[23] M. Morishita, J. Suzuki, and M. Nagata, "Improving neural machine translation by incorporating hierarchical subword features," in *Proceedings of the International Conference on Computational Linguistics (CICLing)*. ACL, 2018.

[24] R. Caruana, "Multitask learning," *Machine learning*, 1997.

[25] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) and the International Joint Conference on Natural Language Processing (IJCNLP)*. ACL, 2015.

[26] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," in *Proceedings of the Second Conference on Machine Translation*. ACL, 2017.

[27] E. Kiperwasser and M. Ballesteros, "Scheduled multi-task learning: From syntax to translation," *Transactions of the Association for Computational Linguistics (ACL)*, 2018.

[28] R. Sanabria and F. Metze, "Hierarchical multi task learning with ctc," in *Proceedings of Workshop on Spoken Language Technologies (SLT)*. IEEE, 2018.

[29] P. Passban, Q. Liu, and A. Way, "Improving character-based decoding using target-side morphological information for neural machine translation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 2018, pp. 58–68.