FBK's Multilingual Neural Machine Translation System for IWSLT 2017

Surafel M. Lakew[1,2], *Quintino F. Lotito*[2], *Marco Turchi*[1], *Matteo Negri*[1], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Trento, Italy
federico@fbk.eu

Abstract

Neural Machine Translation has been shown to enable inference and cross-lingual knowledge transfer across multiple language directions using a single multilingual model. Focusing on this multilingual translation scenario, this work summarizes FBK's participation in the IWSLT 2017 shared task. Our submissions rely on two multilingual systems trained on five languages (*English, Dutch, German, Italian, and Romanian*). The first one is a 20 language direction model, which handles all possible combinations of the five languages. The second multilingual system is trained only on 16 directions, leaving the others as zero-shot translation directions (*i.e* representing a more complex inference task on language pairs not seen at training time). More specifically, our zero-shot directions are Dutch↔German and Italian↔Romanian (resulting in four language combinations). Despite the small amount of parallel data used for training these systems, the resulting multilingual models are effective, even in comparison with models trained separately for every language pair (*i.e.* in more favorable conditions). We compare and show the results of the two multilingual models against a baseline single language pair systems. Particularly, we focus on the four zero-shot directions and show how a multilingual model trained with small data can provide reasonable results. Furthermore, we investigate how pivoting (*i.e* using a bridge/pivot language for inference in a source→pivot→target translations) using a multilingual model can be an alternative to enable zero-shot translation in a low resource setting.

1. Introduction

Recently, multilingual translation across different languages using a single model showed to perform in a comparable way with single language pair systems. In [1, 2], a multilingual model has been successfully trained using a standard Neural Machine Translation (NMT) architecture by applying a simple preprocessing step on the source side of the training data. It consists in prepending an artificial language token indicating the target language id at the beginning of each sentence. This information guides the system towards a specific target language both at training and inference time. This mechanism of guiding the multilingual model is referred to as *target-forcing* [2].

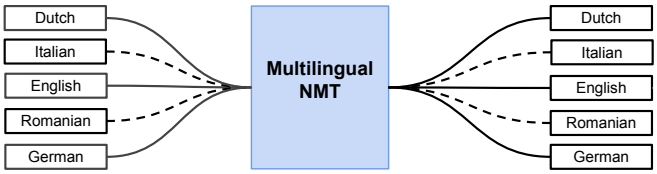

Figure 1: The multilingual system source→target association. A parallel data exists for all the 20 directions in the first multilingual model, where as zero-shot model the Dutch↔German and Italian↔Romanian pairs (dashed line) are excluded

In this work, we present our participation in two IWSLT2017[1] [3] shared tasks: *i)* a multilingual translation task in a small data condition for twenty language directions, and *ii)* a multilingual zero-shot task in a similar small data condition. For convenience, throughout the paper, we refer to the models trained for the two tasks respectively as *Multilingual* and *Zero-shot* models. We trained the two models separately, by sharing a common configuration. The only difference, at training time, is that we removed the four language directions involved in the zero-shot task. Figure 1, shows the twenty possible associations between the source and target pairs, avoiding (*source* = *target*) condition. We trained the models following the same preprocessing and training procedures described in [1]. Note that, due to its small size ($\approx 200K$ for each language pair), the training data set becomes even more sparse after preprocessing and dropping sentences above a certain length (which becomes necessary in order to facilitate and speed-up the training process).

For comparing the performance of the multilingual and zero-shot models, we trained 20 single language pair models. For a fair comparison, the preprocessing and training procedures are similar to the multilingual models. The same models are also used for the comparison against the pivoting method, in which *English* is fixed as the bridging language. In terms of evaluation results, the overall performance of the zero-shot model is satisfactory even if, unsurprisingly, lower than the multilingual model. The largest distance is observed in Romanian→Italian (-3.02 BLEU points), while the smallest difference is observed in the Dutch→German

[1]https://sites.google.com/site/iwsltevaluation2017/

direction (-1.63 BLEU points).

In the following sections of this paper, we begin by introducing the main concepts related to NMT (§2). Then, we review the related work in a multilingual (§3.1) and zero-shot (§3.2) translation domains. In Section 4, we describe the training details (§4.1), the dataset, the preprocessing procedures (§4.2), as well as the results of the single language pair (§4.3) and the multilingual (§4.4) models. For comparing the different approaches, we focus on the zero-shot directions in section (§4.5). Then, we give further analysis in Section 5 and conclude the work in Section 6.

2. Neural Machine Translation

NMT comprises an encoder, a decoder, and an attention-mechanism, which are all trained with maximum likelihood in an end-to-end fashion [4]. The encoder is a recurrent neural network (RNN) that encodes a source sentence into a sequence of hidden state vectors. The decoder is an RNN that uses the representation of the encoder to predict words in the target language [5] [6]. The *attention* mechanism is used to improve the translation by deciding which part of the source sentence can contribute mostly in the prediction process at each time step.

As shown in Figure 2, which simplifies the NMT architecture, first the encoder (green colored section) takes the source words left to right, maps them to vectors and feeds them into the RNN. When the <eos> (*i.e* end of sentence) symbol is seen, the final time step initializes the decoder RNN (blue colored). At each time step, the attention mechanism is applied over the encoder hidden states and combined with the current hidden state of the decoder to predict the next target word. Then, the prediction is fed back to the decoder RNN to predict the next word until the <eos> symbol is generated [7].

In order to build a multilingual model, in this work we used a standard encoder-decoder NMT architecture with a general attention mechanism that combines via dot product the decoder hidden state and a linear transformation of the encoder state [8]. Furthermore, we used four layers of RNN both on the encoder and decoder side.

3. Related Work

3.1. Multilingual NMT

Early works in multilingual NMT are characterized by the use of separate encoder, decoder, and an attention mechanism for every language direction [9] [10]. Firat et al. [11] introduced a way to share the attention mechanism in a many-to-many translation setting still keeping separate encoders and decoders for each source and target language. In a more closely related approach to the one, we utilized in our systems, [1] and [2] introduced a way to share not only the attention mechanism but also a single encoder-decoder. In both works, an artificial language token is prepended at a preprocessing stage to the source sentences in order to enable multilingual translation. In a rather different way, the approach in [2] appended a language-specific code to differentiate words from different languages. The word and subword level language-specific coding mechanism is proved to be expensive, by creating longer sentences that can deteriorate the performance of NMT [5]. In addition, they appended the artificial token as a prefix and postfix on the source side of the training and validation data. In [1], however, only one artificial token is prepended at the beginning of the source sentences. This single token, which specifies the target language proved to work in a comparable performance as specifying two (prefix and postfix) tokens. In this work, we follow the Johnson et al. [1] approach for prepending.

3.2. Zero-Shot Translation

Firat et al. [12], suggested a zero-resource translation by extending their approach in [11] with a shared attention mechanism and a separate encoder-decoder architecture for every language pair. They leverage a pre-trained multi-way multilingual model, and then fine tune it with synthetic parallel data generated by the model itself. Their approach, however, does not allow a zero-shot translation. Instead, they proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus [13] for fine-tuning purposes Moreover, also in this case, the need of separate encoders and decoders for every language pair significantly increases the model complexity. So far, though simple, the most effective approach proposed for zero-shot translation is the one based on *target-forcing* at preprocessing stage [1] [2]. The most attractive benefit of the *target-forcing* comes from the possibility to perform zero-shot translation with the same multilingual setting as in [1, 2].

However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages [14], due to the fact that the target-forcing mechanism requires more examples at training time to effectively handle zero-shot at inference stage. This is particularly visible in case of zero-shot target language which appears only once in comparison with other source $\rightarrow$ target pairs. The promising results in [1] and [2] hence require further investigation to verify if their method can work in various language settings, particularly for low resourced and across distant languages.

As an alternative strategy, pivoting is a rather intuitive way to approach zero-shot translation, especially when it involves low-resourced languages. The idea is to translate from/into under-resourced languages (L_{source} and L_{target}) by leveraging data available for a high-resourced one (L_{pivot}) used as "bridge" between the two languages (*i.e.* $L_{source} \rightarrow L_{pivot} \rightarrow L_{target}$) [15]. However, results in the pivoting framework are strictly bounded to the performance of the two combined translation engines, and especially to that of the weaker one. In contrast, multilingual models that leverage knowledge acquired from data for different language combinations (similar to multi-task learn-

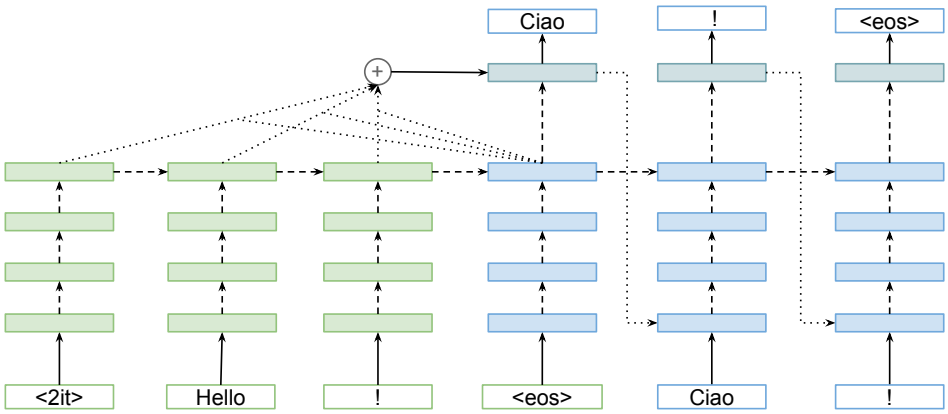

Figure 2: *NMT architecture with encoder-decoder and an attention mechanism, showing an example input "Hello !" translated to Italian "Ciao ! using a <2it> target-forcing mechanism". The first two layers of the encoder (green) are a bidirectional RNN with two additional forward layers. On the decoder side (blue), however, all the layers are forward. The attention mechanism is shown for the first time step of the prediction. Input feed is used to pass the context vector as additional input to the decoder.*

ing) can potentially compete or even outperform the pivoting ones. Taking the different approaches to perform zero-shot in consideration, in Section 4.5, we show the comparison between the zero-shot strategies (i.e direct source→target zero-shot translation and using a pivot language) employing the zero-shot model and the single language pair models.

4. Experiments

4.1. Training Details

For training the multilingual and the single language pair systems we used a standard encoder-decoder NMT architecture with attention mechanism [8][16]. The encoder and decoder sides of the network consist of four layers, where the first two layers of the encoder are bidirectional. As shown on the right side of Figure 2, at each time step an input-feeding mechanism is applied to pass the context vector as an additional input to the decoder by concatenating it with the embedding of the predicted word [8]. Table 1, shows the parameters used for training both the multilingual and single language pair systems.

For optimization, based on preliminary experiments and following best practices from previous work [1], we used Adam [17] with a learning rate of 0.001. Learning rate decay of 0.5 is applied if the perplexity does not decrease on the validation set or the number of epoch passes 8. For reducing perplexity and the network size, we also share the word and softmax embedding of the decoder as suggested by Press and Wolf [18]. To prevent overfitting [19], particularly for the training dataset in this low-resource setting, we applied a dropout of 0.3 on all layers [20]. At time of inference, a beam search of size 10 is utilized to balance decoding time and accuracy of the search. Where each decoding step takes a batch of 128 evaluation set. The experiments are carried out using the

Parameter	Value
RNN type	LSTM
RNN size	1024
embedding	512
encoder	bidirectional
encoder depth	4
decoder depth	4
beam size	10
batch size	128
optimizer	adam
dropout	0.3

Table 1: *Parameters used to train both single language pair and multilingual models.*

open source OpenNMT-py[2] toolkit [7].

With the aim to compare the performance of the multilingual models, we trained twenty single language pair models with the same amount of training data used by each direction of the multilingual models (see Table 2 for details). For every direction of the multilingual models and every single pair model we report case sensitive detokenized (*i.e* using the internal tokenization of the scorer) BLEU scores [21] computed using `mteval-v13a.pl`.

4.2. Dataset and Preprocessing

In the source-target pair of the five languages considered in this work, there are $\approx 200k$ parallel training sentences in each pair. As shown in Table 2, test2010 is used for evaluating the models, whereas test2017 is used for comparison purposes and as the official submission test set. For training both the multilingual and single language pair models, the same number of sentences are used.

[2]https://github.com/OpenNMT/OpenNMT-py

Direction	Training	test2010	test2017
English ↔ German	197,489	1,497	1,138
English ↔ Italian	221,688	1,501	1,147
English ↔ Dutch	231,669	1,726	1,181
English ↔ Romanian	211,508	1,633	1,129
German ↔ Italian	197,461	1,502	1,133
German ↔ Romanian	194,257	1,626	1,121
Dutch ↔ Italian	228,534	1,623	1,183
Dutch ↔ Romanian	199,762	1,637	1,123
German ↔ Dutch	209,169	1,729	1,174
Italian ↔ Romanian	209,668	1,605	1,127

Table 2: *Number of sentences used for training and evaluation in a source↔target combination. The German↔Dutch and Italian↔Romanian four language directions shown in the third row is removed from the training data of the zero-shot multilingual model.*

To prepare the data for training, we first prepare a tokenized version. Then, using a shared byte pair encoding (BPE) model, we segment the tokens into sub-word units [22]. The BPE model is trained on a joint source and target dataset covering all the language directions. For this operation we used $8,000$ BPE merging rules. A frequency threshold of 30 is used to apply the segmentation. For choosing the BPE segmentation rules, we follow the suggestion of Denkowski and Neubig [23] in such small data condition. When training the multilingual models, we add the *target-forcing* language token at the source side of each parallel data, both for training and validation sets [1]. Apart from the data set provided by the IWSLT17 shared task [24], for the multilingual small data condition no additional data are utilized, neither for the preprocessing stage nor for the experiments.

4.3. Single Language Pair Models

To compare the two evaluation tasks (multilingual and zero-shot model), we trained twenty single language pair models. As discussed in training details (4.1), these models are trained in a similar setting with the multilingual models. Table 3, summarizes the performance each of the twenty models on *test2017*. Except for the slight gain in the Romanian→Italian direction over the results of the multilingual model (see Table 4), the performance of the single language pair models (see Table 3), are poorer in the rest of the other 19 directions.

4.4. Multilingual Models

In this experiment, we present the multilingual 20 direction and zero-shot 16 direction models. Note: in case of the zero-shot model the training data for the German↔Dutch and Italian↔Romanian directions are dropped. As in the single language pair models, the rest of the training follows the steps described in Section 4.1. The results shown in Table 4, are the primary runs of the official submission for the multilingual and zero-shot small data condition tasks. The term of comparison between these two multilingual models is focused on the four zero-shot directions. As expected, the zero-shot model performed poorly than the multilingual model in all of the four directions.

Particularly, we see a larger gap of 3.02 for the Romanian → Italian, whereas the Italian → Romanian direction has a difference of 2.48 BLEU score. In case of German → Dutch and Dutch → German the gap closes to 1.99 and 1.63 respectively. For the other 16 non-zero-shot directions, the multilingual model performed slightly better than the zero-shot model. However, in case of Dutch → English and Italian→Dutch there exists a pattern where the zero-shot model performed better. In Table 5, we separately reported additional results for the multilingual small-data condition task evaluated using a model from an on-time submission. Except for the reporting purpose the results from Table 5, are not included in any of the comparisons made in this work.

4.5. Zero-shot Vs. Pivoting

In this analysis, we compare zero-shot translation mechanisms using the Zero-shot multilingual model and models trained on single language pair. Specifically, we compared three different results of a zero-shot translation on the IWSLT *tst2017*. The first is a direct zero-shot from a source → target language using the Zero-shot multilingual model. The other two results are acquired through a pivoting translation mechanism in a *two-step* translation. Hence, pivoting using single language pair models requires a source→pivot and a pivot→target model. However, this is not the case for the Zero-shot model which assumes to already have the pivot paired with the source and target languages. In both cases, we use English as a pivot language. Thus, for the Italian ↔ Romanian zero-shot directions we follow Italian ↔ English ↔ Romanian, whereas the German ↔ Dutch translation is done as German ↔ English ↔ Dutch *two-step* translations.

Approaches	De→Nl	Nl→De	It→Ro	Ro→It
Zero-shot	17.17	**16.96**	16.58	18.32
Zero-shot Pivot	**17.67**	16.84	**17.3**	**19.57**
Single Pair Pivot	15.3	14.9	15.22	17.2

Figure 3: A BLEU score comparison of German ↔ Dutch and Italian ↔ Romanian four language directions using three different zero-shot translation mechanisms. The first row is a direct zero-shot translation using the Zero-shot model, while the last two rows show the results of the pivoting mechanism.

The results in Table 3, shows better performance of the Zero-shot model using a pivoting mechanism (except the Nl→De direction). In a surprising way, the pivoting using two separate single language pair models for each translation direction perform worse than the direct zero-shot and the pivoting zero-shot using the multilingual model in row 1 and 2.

Single Pair	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	19.84	26.41	29.90	21.41	18.93	15.52	12.52	18.47	14.71	18.67
←	24.69	30	34.03	28.03	17.93	15.47	13.81	20.13	16.78	**21.71**

Table 3: BLEU score on IWSLT *tst2017* from twenty single language pair models that are trained separately. The bold highlighted Romanian→Italian direction is the only gain over the multilingual system.

Multilingual	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	20.88	26.72	29.6	21.95	19.16	16.84	14.62	19.33	16.54	19.06
←	25.62	29.79	34.24	28.93	18.59	16.88	15.87	20.27	18.92	21.34
Zero-shot										
→	20.67	26.11	28.86	21.54	17.17	16.28	13.93	**19.76**	15.88	16.58
←	25.22	**30.04**	34.16	28.52	16.96	16.13	15.47	20.00	17.72	18.32

Table 4: BLEU scores on the IWSLT *tst2017* using the multilingual model trained on 20 directions and the zero-shot model trained using the dataset of the 16 directions. Bold highlighted Nl→En and Nl→It are the only cases where the zero-shot model performed better than the multilinugal.

5. Discussion

The experiments in this work showed that a single multilingual system can perform better than independently trained single language pair systems. Hence, training a single system on the concatenation of all the language directions helps to maximize the parameter sharing in the common representation space. The consistent gain of the multilingual model in 19 directions except for the slight loss for the Romanian→Italian shows the potential behind multilingual approaches. Unlike the scenario in previous work [1], we showed the improvements in a low resource setting, without any additional data to tune the system. In case of the zero-shot model, we considered the non-zero-shot 16 directions for comparison with the bilingual models. In an equivalent way with the multilingual model, the zero-shot model has shown gains over the single language pair models.

Even though the zero-shot model showed a comparable performance with the multilingual model in the 16 non-zero-shot directions, there is a slight performance degradation in all but the Dutch→English direction. For instance, a 29.6 BLEU score for English→Italian of the multilingual model decreases to 28.86 BLEU with the zero-shot model. However, for the translation directions Source→English the maximum loss for the zero-shot model is 0.41 BLEU in the Romanian→English direction. As we expected initially, these results reflect a condition where the number of language pairs with English (on the encoder and decoder side) stayed the same in both multilingual models. Whereas the absence of the four zero-shot (source↔target) combinations influenced the translation performance of the Zero-shot model even for the language pairs seen at training time.

The pivoting experiments discussed in Section 3, is another way of showing the reasonable performance of the zero-shot model. The two-step inference (i.e source → pivot, and then pivot→target) for zero-shot translation provided a better performance in three directions out of four (see Table 3), in comparison with a direct zero-shot translation. We observed that using English (the only language that has a pair and better performance with all the zero-shot directions) as the bridge language played the major role for the gain. However, as discussed in Section 3, pivoting using two separate bilingual systems is found to be weaker (see the third row of Table 3) in leveraging the pivot language. This can be observed from the weaker bilingual systems in comparison with the zero-shot model. Particularly, both in the source→English, and as well in the English→target the bilingual model performance is poor in comparison with the zero-shot model, see Table 3 and 4.

Overall the reasonable performance of the zero-shot model shows the potential of a multilingual approach. In the subsequent comparisons using a pivoting method, it becomes clearer that in a multilingual setting it is possible to train a more robust model that can handle the noise from the output of the first step translation.

6. Conclusion

In this work, we showed how a multilingual system can deliver better performance over bilingual systems in twenty different directions. In addition, we explored the performance of a multilingual model for a zero-shot translation task in a direct source-to-target translation and using a pivot language in a two-step translation. The Zero-shot model proved to be an effective way of achieving a zero-shot translation for German ↔ Dutch and Italian ↔ Romanian directions, while showing a comparable performance in the non-zero-shot directions with the Multilingual model trained on the full training dataset. In addition to avoiding training several independent systems, multilingual model showed to be beneficial in such low-resource setting.

In future works, we plan to thoroughly investigate the

Multilingual	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	20.28	25.68	29.32	21.12	18.67	15.85	13.43	19.25	15.48	17.89
←	24.27	30.16	33.86	28	17.65	15.98	14.99	18.77	17.5	21.28

Table 5: BLEU scores for the twenty language directions evaluated using a multilingual model on *tst2017* (results are using a model from an on-time submission of the multilingual small data condition task).

behavior of the multilingual systems, seeing that the target-forcing mechanism plays the main role in redirecting the translation to the right target language, and susceptible to ambiguities in a low-resource setting. In addition, we plan to explore a better way to balance the training dataset for the different language directions. Particularly, for achieving a zero-shot translation we expect that finding the right language combinations, amount of dataset, and the number of languages require further investigation. Furthermore, a human evaluation on the outputs of the bilingual and the multilingual models would be interesting to assess the translation quality, in addition to confirming the evaluation scores, reported in this work.

7. Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). The Titan Xp used for this research was donated by the NVIDIA Corporation. This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft.

8. References

[1] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *arXiv preprint arXiv:1611.04558*, 2016.

[2] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[3] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[7] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[9] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation." in *ACL (1)*, 2015, pp. 1723–1732.

[10] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[11] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[12] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multilingual neural machine translation," *arXiv preprint arXiv:1606.04164*, 2016.

[13] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[14] S. M. Lakew, A. D. G. Mattia, and F. Marcello, "Multilingual neural machine translation for low resource languages," in *CLiC-it 2017 – 4th Italian Conference on Computational Linguistics, to appear*, 2017.

[15] H. Wu and H. Wang, "Pivot language approach for phrase-based statistical machine translation," *Machine Translation*, vol. 21, no. 3, pp. 165–181, 2007.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] O. Press and L. Wolf, "Using the output embedding to improve language models," *arXiv preprint arXiv:1608.05859*, 2016.

[19] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[23] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *arXiv preprint arXiv:1706.09733*, 2017.

[24] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, vol. 261, 2012, p. 268.