Adaptive Knowledge Sharing in Multi-Task Learning: Improving Low-Resource Neural Machine Translation

Poorya Zaremoodi **Wray Buntine** **Gholamreza Haffari**
Faculty of Information Technology, Monash University, Australia
first.last@monash.edu

Abstract

Neural Machine Translation (NMT) is notorious for its need for large amounts of bilingual data. An effective approach to compensate for this requirement is Multi-Task Learning (MTL) to leverage different linguistic resources as a source of inductive bias. Current MTL architectures are based on the SEQ2SEQ transduction, and (partially) share different components of the models among the tasks. However, this MTL approach often suffers from task interference, and is not able to fully capture commonalities among subsets of tasks. We address this issue by extending the recurrent units with multiple *blocks* along with a trainable *routing network*. The routing network enables adaptive collaboration by dynamic sharing of blocks conditioned on the task at hand, input, and model state. Empirical evaluation of two low-resource translation tasks, English to Vietnamese and Farsi, show +1 BLEU score improvements compared to strong baselines.

1 Introduction

Neural Machine Translation (NMT) has shown remarkable progress in recent years. However, it requires large amounts of bilingual data to learn a translation model with reasonable quality (Koehn and Knowles, 2017). This requirement can be compensated by leveraging curated monolingual linguistic resources in a multi-task learning framework. Essentially, learned knowledge from auxiliary linguistic tasks serves as inductive bias for the translation task to lead to better generalizations.

Multi-Task Learning (MTL) is an effective approach for leveraging commonalities of related tasks to improve performance. Various recent works have attempted to improve NMT by scaffolding translation task on a single auxiliary task (Domhan and Hieber, 2017; Zhang and Zong, 2016; Dalvi et al., 2017). Recently, (Niehues and Cho, 2017) have made use of several linguistic tasks to improve NMT. Their method shares components of the SEQ2SEQ model among the tasks, e.g. encoder, decoder or the attention mechanism. However, this approach has two limitations: (i) it *fully* shares the components, and (ii) the shared component(s) are shared among *all* of the tasks. The first limitation can be addressed using deep stacked layers in encoder/decoder, and sharing the layers partially (Zaremoodi and Haffari, 2018). The second limitation causes this MTL approach to suffer from task interference or inability to leverages commonalities among a *subset* of tasks. Recently, (Ruder et al., 2017) tried to address this issue; however, their method is restrictive for SEQ2SEQ scenarios and does not consider the input at each time step to modulate parameter sharing.

In this paper, we address the task interference problem by learning how to dynamically control the amount of sharing among all tasks. We extended the recurrent units with multiple *blocks* along with a routing network to dynamically control sharing of blocks conditioning on the task at hand, the input, and model state. Empirical results on two low-resource translation scenarios, English to Farsi and Vietnamese, show the effectiveness of the proposed model by achieving +1 BLEU score improvement compared to strong baselines.

2 SEQ2SEQ MTL Using Recurrent Unit with Adaptive Routed Blocks

Our MTL is based on the sequential encoder-decoder architecture with the attention mecha-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 656–661
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

nism (Luong et al., 2015b; Bahdanau et al., 2014). The encoder/decoder consist of recurrent units to read/generate a sentence sequentially. Sharing the parameters of the recurrent units among different tasks is indeed sharing the *knowledge* for controlling the information flow in the hidden states. Sharing these parameters among *all* tasks may, however, lead to task interference or inability to leverages commonalities among *subsets* of tasks. We address this issue by extending the recurrent units with multiple *blocks*, each of which processing its own information flow through the time. The state of the recurrent unit at each time step is composed of the states of these blocks. The recurrent unit is equipped with a *routing* mechanism to softly direct the input at each time step to these blocks (see Fig 1). Each block mimics an expert in handling different kinds of information, coordinated by the router. In MTL, the tasks can use different subsets of these shared experts.

(Rosenbaum et al., 2018) uses a routing network for adaptive selection of non-linear functions for MTL. However, it is for fixed-size inputs based on a feed-forward architecture, and is not applicable to SEQ2SEQ scenarios such as MT. (Shazeer et al., 2017) uses Mixture-of-Experts (feed-forward sub-networks) between stacked layers of recurrent units, to adaptively gate state information *vertically*. This is in contrast to our approach where the *horizontal* information flow is adaptively modulated, as we would like to minimise the task interference in MTL.

Assuming there are n blocks in a recurrent unit, we share $n-1$ blocks among the tasks, and let the last one to be task-specific[1]. Task-specific block receives the input of the unit directly while shared blocks are fed with modulated input by the routing network. The state of the unit at each time-step would be the aggregation of blocks' states.

2.1 Routing Mechanism

At each time step, the routing network is responsible to softly forward the input to the shared blocks conditioning on the input $\boldsymbol{x}_t$, and the previous hidden state of the unit $\boldsymbol{h}_{t-1}$ as follows:

$$\begin{aligned} \boldsymbol{s}_t &= \tanh(\boldsymbol{W}_x \cdot \boldsymbol{x}_t + \boldsymbol{W}_h \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_s), \\ \boldsymbol{\tau}_t &= \text{softmax}(\boldsymbol{W}_\tau \cdot \boldsymbol{s}_t + \boldsymbol{b}_\tau), \end{aligned}$$

where $\boldsymbol{W}$'s and $\boldsymbol{b}$'s are the parameters. Then, the i-th shared block is fed with the input of the unit modulated by the corresponding output of the routing network $\tilde{\boldsymbol{x}}_t^{(i)} = \boldsymbol{\tau}_t[i]\boldsymbol{x}_t$ where $\boldsymbol{\tau}_t[i]$ is the scalar output of the routing network for the i-th block.

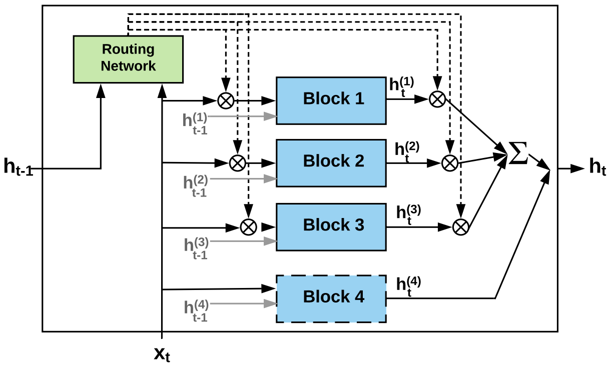

Figure 1: High-level architecture of the proposed recurrent unit with 3 shared blocks and 1 task-specific.

The hidden state of the unit is the concatenation of the hidden state of the shared and task-specific parts $\boldsymbol{h}_t = [\boldsymbol{h}_t^{(shared)}; \boldsymbol{h}_t^{(task)}]$. The state of task-specific part is the state of the corresponding block $\boldsymbol{h}_t^{(task)} = \boldsymbol{h}_t^{(n+1)}$, and the state of the shared part is the sum of states of shared blocks weighted by the outputs of the routing network $\boldsymbol{h}_t^{(shared)} = \sum_{i=1}^{n} \boldsymbol{\tau}_t[i]\boldsymbol{h}_t^{(i)}$.

2.2 Block Architecture

Each block is responsible to control its own flow of information via a standard gating mechanism. Our recurrent units are agnostic to the internal architecture of the blocks; we use the gated-recurrent unit (Cho et al., 2014) in this paper. For the i-th block the corresponding equations are as follows:

$$\begin{aligned} \boldsymbol{z}_t^{(i)} &= \sigma(\boldsymbol{W}_z^{(i)}\tilde{\boldsymbol{x}}_t^{(i)} + \boldsymbol{U}_z^{(i)}\boldsymbol{h}_{t-1}^{(i)} + \boldsymbol{b}_z^{(i)}), \\ \boldsymbol{r}_t^{(i)} &= \sigma(\boldsymbol{W}_r^{(i)}\tilde{\boldsymbol{x}}_t^{(i)} + \boldsymbol{U}_r^{(i)}\boldsymbol{h}_{t-1}^{(i)} + \boldsymbol{b}_r^{(i)}), \\ \tilde{\boldsymbol{h}}_t^{(i)} &= \tanh(\boldsymbol{W}_h^{(i)}\tilde{\boldsymbol{x}}_t^{(i)} + \boldsymbol{U}_h^{(i)}\boldsymbol{h}_{t-1}^{(i)} + \boldsymbol{b}_h^{(i)}), \\ \boldsymbol{h}_t^{(i)} &= \boldsymbol{z}_t^{(i)} \odot \boldsymbol{h}_{t-1}^{(i)} + (1 - \boldsymbol{z}_t^{(i)}) \odot \tilde{\boldsymbol{h}}_t^{(i)}. \end{aligned}$$

2.3 Training Objective and Schedule.

The rest of the model is similar to attentional SEQ2SEQ model (Luong et al., 2015b) which computes the conditional probability of the target sequence given the source $P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_j P_{\boldsymbol{\theta}}(y_j|\boldsymbol{y}_{<j}\boldsymbol{x})$. For the case of training $M+1$ SEQ2SEQ transduction tasks, each of which is associated with a training set $\mathcal{D}_m := \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_m}$, the parameters of MTL architecture $\boldsymbol{\Theta}_{mtl} =$

[1] multiple recurrent units can be stacked on top of each other to consist a multi-layer component

$\{\mathbf{\Theta}_m\}_{m=0}^{M}$ are learned by maximizing the following objective:

$$\mathcal{L}_{mtl}(\mathbf{\Theta}_{mtl}) := \sum_{m=0}^{M} \frac{\gamma_m}{|\mathcal{D}_m|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_m} \log P_{\mathbf{\Theta}_m}(\boldsymbol{y}|\boldsymbol{x})$$

where $|\mathcal{D}_m|$ is the size of the training set for the m-th task, and γ_m is responsible to balance the influence of tasks in the training objective. We explored different values in preliminary experiments, and found that for our training schedule $\gamma = 1$ for all tasks results in the best performance. Generally, γ is useful when the dataset sizes for auxiliary tasks are imbalanced (our training schedule handles the main task).

Variants of stochastic gradient descent (SGD) can be used to optimize the objective function. In our training schedule, we randomly select a mini-batch from the main task (translation) and another mini-batch from a randomly selected auxiliary task to make the next SGD update. Selecting a mini-batch from the main task in each SGD update ensures that its training signals are not washed out by auxiliary tasks.

3 Experiments

3.1 Bilingual Corpora

We use two language-pairs, translating from English to Farsi and Vietnamese. We have chosen them to analyze the effect of multi-task learning on languages with different underlying linguistic structures[2]. We apply BPE (Sennrich et al., 2016) on the union of source and target vocabularies for English-Vietnamese, and separate vocabularies for English-Farsi as the alphabets are disjoined (30K BPE operations). Further details about the corpora and their pre-processing is as follows:

- The English-Farsi corpus has ∼105K sentence pairs. It is assembled from English-Farsi parallel subtitles from the TED corpus (Tiedemann, 2012), accompanied by all the parallel news text in LDC2016E93 *Farsi Representative Language Pack* from the Linguistic Data Consortium. The corpus has been normalized using the Hazm toolkit[3]. We have removed sentences with more than 80 tokens in either side (before applying BPE). 3k and 4k sentence pairs were held out for the purpose of validation and test.
- The English-Vietnamese has ∼133K training pairs. It is the preprocessed version of the IWSLT 2015 translation task provided by (Luong and Manning, 2015). It consists of subtitles and their corresponding translations of a collection of public speeches from TED and TEDX talks. The "tst2012" and "tst2013" parts are used as validation and test sets, respectively. We have removed sentence pairs which had more than 300 tokens after applying BPE on either sides.

3.2 Auxiliary Tasks

We have chosen the following auxiliary tasks to leverage the syntactic and semantic knowledge to improve NMT:

Named-Entity Recognition (NER). It is expected that learning to recognize named-entities help the model to learn translation pattern by masking out named-entites. We have used the NER data comes from the CONLL shared task.[4] Sentences in this dataset come from a collection of newswire articles from the Reuters Corpus. These sentences are annotated with four types of named entities: persons, locations, organizations and names of miscellaneous entities.

Syntactic Parsing. By learning the phrase structure of the input sentence, the model would be able to learn better re-ordering. Specially, in the case of language pairs with high level of syntactic divergence (e.g. English-Farsi). We have used Penn Tree Bank parsing data with the standard split for training, development, and test (Marcus et al., 1993). We cast syntactic parsing to a SEQ2SEQ transduction task by linearizing constituency trees (Vinyals et al., 2015).

Semantic Parsing. Learning semantic parsing helps the model to abstract away the meaning from the surface in order to convey it in the target translation. For this task, we have used the Abstract Meaning Representation (AMR) corpus Release 2.0 (LDC2017T10)[5]. This corpus contains natural language sentences from newswire, weblogs, web discussion forums and broadcast conversations. We cast this task to a SEQ2SEQ transduction task by linearizing the AMR graphs (Konstas et al., 2017).

[2] English and Vietnamese are SVO, and Farsi is SOV.

[3] www.sobhe.ir/hazm

[4] https://www.clips.uantwerpen.be/conll2003/ner

[5] https://catalog.ldc.upenn.edu/LDC2017T10

	English → Farsi						English → Vietnamese					
	Dev			Test			Dev			Test		
Method	PPL	TER	BLEU	PPL	TER	BLEU	PPL	TER	BLEU	PPL	TER	BLEU
NMT (Luong et al., 2015b)	55.36	87.9	8.57	56.21	88.2	8.35	18.21	64.92	18.39	16.3	61.37	20.18
MTL (Full) (Niehues and Cho, 2017)	47.43	85.92	8.97	48.23	87.3	8.73	14.56	61.52	20.55	12.5	57.6	22.6
MTL (Partial) (Zaremoodi and Haffari, 2018)	42.6	80.16	10.58	43.09	81.94	10.54	13.32	59.55	22.2	11.34	55.84	24.65
Our MTL (Routing)	**37.95**	**76.30**	**12.06**	**38.57**	**78.18**	**11.95**	**12.38**	**58.52**	**23.06**	**10.52**	**54.33**	**25.65**

Table 1: The performance measures of the baselines vs our MTL architecture on the bilingual datasets.

3.3 Models and Baselines

We have implemented the proposed MTL architecture along with the baselines in C++ using DyNet (Neubig et al., 2017) on top of Mantis (Cohn et al., 2016) which is an implementation of the attentional SEQ2SEQ NMT model. For our MTL architecture, we used the proposed recurrent unit with 3 blocks in encoder and decoder. For the fair comparison in terms the of number of parameters, we used 3 stacked layers in both encoder and decoder components for the baselines. We compare against the following baselines:

- Baseline 1: The vanilla SEQ2SEQ model (Luong et al., 2015a) without any auxiliary task.
- Baseline 2: The MTL architecture proposed in (Niehues and Cho, 2017) which fully shares parameters in components. We have used their best performing architecture with our training schedule. We have extended their work with *deep* stacked layers for the sake of comparison.
- Baseline 3: The MTL architecture proposed in (Zaremoodi and Haffari, 2018) which uses deep stacked layers in the components and shares the parameters of the top two/one stacked layers among encoders/decoders of all tasks[6].

For the proposed MTL, we use recurrent units with 400 hidden dimensions for each block. The encoders and decoders of the baselines use GRU units with 400 hidden dimensions. The attention component has 400 dimensions. We use Adam optimizer (Kingma and Ba, 2014) with the initial learning rate of 0.003 for all the tasks. Learning rates are halved on the decrease in the performance on the dev set of corresponding task. Mini-batch size is set to 32, and dropout rate is 0.5. All models are trained for 50 epochs and the best models are saved based on the perplexity on the dev set of the translation task.

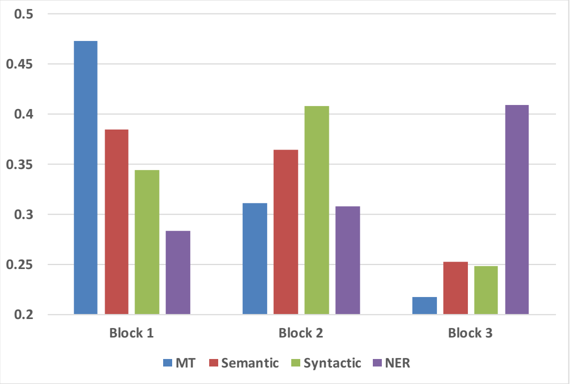

Figure 2: Average percentage of block usage for each task.

For each task, we add special tokens to the beginning of source sequence (similar to (Johnson et al., 2017)) to indicate which task the sequence pair comes from.

We used greedy decoding to generate translation. In order to measure translation quality, we use BLEU[7] (Papineni et al., 2002) and TER (Snover et al., 2006) scores.

3.4 Results and analysis

Table 1 reports the results for the baselines and our proposed method on the two aforementioned translation tasks. As expected, the performance of MTL models are better than the baseline 1 (only MT task). As seen, partial parameter sharing is more effective than fully parameter sharing. Furthermore, our proposed architecture with adaptive

[6] In preliminary experiments, we have tried different sharing scenarios and this one led to the best results.

[7] Using "multi-bleu.perl" script from Moses (Koehn et al., 2007).

sharing performs better than the other MTL methods on all tasks, and achieve +1 BLEU score improvements on the test sets. The improvements in the translation quality of NMT models trained by our MTL method may be attributed to less interference with multiple auxiliary tasks.

Figure 2 shows the average percentage of block usage for each task in an MTL model with 3 shared blocks, on the English-Farsi test set. We have aggregated the output of the routing network for the blocks in the encoder recurrent units over all the input tokens. Then, it is normalized by dividing on the total number of input tokens. Based on Figure 2, the first and third blocks are more specialized (based on their usage) for the translation and NER tasks, respectively. The second block is mostly used by the semantic and syntactic parsing tasks, so specialized for them. This confirms our model leverages commonalities among subsets of tasks by dedicating common blocks to them to reduce task interference.

4 Conclusions

We have presented an effective MTL approach to improve NMT for low-resource languages, by leveraging curated linguistic resources on the source side. We address the task interference issue in previous MTL models by extending the recurrent units with multiple *blocks* along with a trainable *routing network*. Our experimental results on low-resource English to Farsi and Vietnamese datasets, show +1 BLEU score improvements compared to strong baselines.

Acknowledgments

The research reported here was initiated at the 2017 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, hosted at Carnegie Mellon University and sponsored by Johns Hopkins University with unrestricted gifts from Amazon, Apple, Facebook, Google, and Microsoft. We are very grateful to the workshop members for the insightful discussions and data pre-processing. This work was supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au), and by the Australian Research Council through DP160102686. The first author was partly supported by CSIRO's Data61. We would like to thank the anonymous reviewers for their constructive feedback.

References

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

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 142–151.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1501–1506.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 146–157.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Jan Niehues and Eunah Cho. 2017. Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proceedings of the Second Conference on Machine Translation*, pages 80–89.

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.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. 2018. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *International Conference on Learning Representations*.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. *CoRR*, abs/1705.08142.

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*, pages 1715–1725.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 2214–2218.

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28*, pages 2773–2781.

Poorya Zaremoodi and Gholamreza Haffari. 2018. Neural machine translation for bilingually scarce scenarios: A deep multi-task learning approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.