Bilingual Low-Resource Neural Machine Translation with Round-Tripping: The Case of Persian-Spanish

Benyamin Ahmadnia[1] and **Bonnie J. Dorr**[2]

[1]Department of Computer Science, Tulane University, New Orleans, LA, USA
[2]Institute for Human and Machine Cognition (IHMC), Ocala, FL, USA
ahmadnia@tulane.edu , bdorr@ihmc.us

Abstract

The quality of Neural Machine Translation (NMT), as a data-driven approach, massively depends on quantity, quality and relevance of the training dataset. Such approaches have achieved promising results for bilingually high-resource scenarios but are inadequate for low-resource conditions. This paper describes a round-trip training approach to bilingual low-resource NMT that takes advantage of monolingual datasets to address training data scarcity, thus augmenting translation quality. We conduct detailed experiments on Persian-Spanish as a bilingually low-resource scenario. Experimental results demonstrate that this competitive approach outperforms the baselines.

1 Introduction

Neural Machine Translation (NMT) has made considerable progress in recent years. However, to achieve acceptable translation output, large sets of aligned parallel sentences are required for the training phase. Thus, as a data-driven paradigm, the quality of NMT output strongly depends on the quality as well as quantity of the provided training data (Bahdanau et al., 2015). Unfortunately, in practice, collecting such parallel text by human labeling is very costly and time consuming (Ahmadnia and Serrano, 2017).

Low-resource languages are those that have fewer technologies and datasets relative to some measure of their international importance. The biggest issue with low-resource languages is the extreme difficulty of obtaining sufficient resources. Natural Language Processing (NLP) methods that have been created for analysis of low-resource languages are likely to encounter similar issues to those faced by documentary and descriptive linguists whose primary endeavor is the study of minority languages. Lessons learned from such studies are highly informative to NLP researchers who seek to overcome analogous challenges in the computational processing of these types of languages.

Assuming that large monolingual texts are available, an obvious next step is to leverage these texts to augment the NMT systems' performance. Various approaches have been developed for this purpose. In some approaches, target monolingual texts are employed to train a Language Model (LM) that is then integrated with MT models trained from parallel texts to enhance translation quality (Brants et al., 2007; Gülçehre et al., 2015). Although these approaches utilize monolingual text to train a LM, they do not address the shortage of bilingual training datasets.

In other approaches, bilingual datasets are automatically generated from monolingual texts by utilizing the Translation Model (TM) trained on aligned bilingual text; the resulting sentence pairs are used to enlarge the initial training dataset for subsequent learning iterations (Ueffing et al., 2008; Sennrich et al., 2016). Although these approaches enlarge the bilingual training dataset, there is no quality control and, thus, the accuracy of the generated bilingual dataset cannot be guaranteed (Ahmadnia et al., 2018).

To tackle the issues described above, we apply a new round-tripping approach that incorporates *dual learning* (He et al., 2016) for automatic learning from unlabeled data, but transcends this prior work through effective leveraging of monolingual text. Specifically, the round-tripping approach takes advantage of the bootstrapping methods including self-training and co-training. These methods start their mission from a small set of labelled examples, while also considering one or

Proceedings of Recent Advances in Natural Language Processing, pages 18–24,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_003

two weak translation models, and makes improvement through the incorporation of unlabeled data into the training dataset.

In the round-tripping approach, the two translation tasks (forward and backward) together make a *closed loop*, i.e., one direction produces informative *feedback* for training the TM for the other direction, and vice versa. The feedback signals—which consist of the language model likelihood of the output model and the reconstruction error of the original sentence—drive the process of iterative updates of the forward and backward TMs.

For the purpose of evaluation, we apply this approach to a bilingually low-resource language pair (Persian-Spanish) to leverage monolingual data in a more effective way. By utilizing the round-tripping approach, the monolingual data play a similar role to the bilingual data, effectively reducing the requirement for parallel data. In particular, each model provides guidance to the other throughout the learning process. Our results show that round-tripping for NMT works well in the Persian-Spanish low-resource scenario. By learning from monolingual data, this approach achieves comparable accuracy to a NMT approach trained from the full bilingual data for the two translation tasks (forward and backward).

The remainder of this paper is organized as follows; Section 2 presents the previous related work. In Section 3, we briefly review the relevant mathematical background of NMT paradigm. Section 4 describes the round-trip training approach. The experiments and results are presented in Section 5. Conclusions and future work are discussed in Section 6.

2 Related Work

The integration of monolingual data for NMT models was first proposed by (Gülçehre et al., 2015), who train monolingual LMs independently, and then integrate them during decoding through rescoring of the beam (adding the recurrent hidden state of the LM to the decoder state of the encoder-decoder network). In this approach, an additional controller mechanism controls the magnitude of the LM signal. The controller parameters and output parameters are tuned on further parallel training data, but the LM parameters are fixed during the fine tuning stage.

Jean et al. (2015) also report on experiments with reranking of NMT output with a 5-gram LM, but improvements are small. The production of synthetic parallel texts bears resemblance to data augmentation techniques, where datasets are often augmented with rotated, scaled, or otherwise distorted variants of the (limited) training set (Rowley et al., 1998).

A similar avenue of research is self-training (McClosky et al., 2006). The self-training approach as a bootstrapping method typically refers to the scenario where the training dataset is enhanced with training instances with artificially produced output labels (whereas we start with neural network based output, i.e., the translation, and artificially produce an input). We expect that this is more robust towards noise in MT.

Hoang et al. (2018) showed that the quality of back translation matters and proposed an iterative back translation, in which back translated data are used to build better translation systems in forward and backward directions. These, in turn, are used to reback translate monolingual data. This process is iterated several times.

Improving NMT with monolingual source data, following similar work on phrase-based SMT (Schwenk, 2008), remains possible future work. Domain adaptation of neural networks via continued training has been shown to be effective for neural language models by (Ter-Sarkisov et al., 2015).

Round-tripping has already been utilized in SMT by (Ahmadnia et al., 2019). In this work, forward and backward models produce informative feedback to iteratively update the TMs during the training of the system.

3 Neural Machine Translation

NMT consists of an encoder and a decoder. Following (Bahdanau et al., 2015), we adopt an *attention-based* encoder-decoder model, i.e., one that selectively focuses on sub-parts of the sentence during translation. Consider a source sentence $X = \{x_1, x_2, ..., x_J\}$ and a target sentence $Y = \{y_1, y_2, ..., y_I\}$. The problem of translation from the source language to the target is solved by finding the best target language sentence $\hat{y}$ that maximizes the conditional probability:

$$\hat{y} = \arg\max_{y} P(y|x) \tag{1}$$

The conditional word probabilities given the source language sentence and preceding target language words compose the conditional probability as follows:

$$P(y|x) = \prod_{i=1}^{I} P(y_i|y_{<i}, x) \tag{2}$$

where y_i is the target word emitted by the decoder at step i and $y_{<_i} = (y_1, y_2, ..., y_{i-1})$.

To compose the model, both the encoder and decoder are implemented employing Recurrent neural Networks (RNNs) (Rumelhart et al., 1986), i.e., the encoder converts source words into a sequence of vectors, and the decoder generates target words one-by-one based on the conditional probability shown in the Equation (2). More specifically, the encoder takes a sequence of source words as inputs and returns forward hidden vectors $\overrightarrow{h_j}(1 \leq j \leq J)$ of the forward-RNN:

$$\overrightarrow{h_j} = f(\overrightarrow{h_{j-1}}, x) \tag{3}$$

Similarly, we obtain backward hidden vectors $\overleftarrow{h_j}(1 \leq j \leq J)$ of the backward-RNN, in the reverse order.

$$\overleftarrow{h_j} = f(\overleftarrow{h_{j-1}}, x) \tag{4}$$

These forward and backward vectors are concatenated to make source vectors $h_j(1 \leq j \leq J)$ based on Equation (5):

$$h_j = \left[\overrightarrow{h_j}; \overleftarrow{h_j}\right] \tag{5}$$

The decoder takes source vectors as input and returns target words. It starts with the initial hidden vector h_J (concatenated source vector at the end), and generates target words in a recurrent manner using its hidden state and an output context.

The conditional output probability of a target language word y_i is defined as follows:

$$P(y_i|y_{<i}, x) = \mathit{softmax}\left(f(d_i, y_{i-1}, c_i)\right) \tag{6}$$

where f is a non-linear function and d_i is the hidden state of the decoder at step i:

$$d_i = g(d_{i-1}, y_{i-1}, c_i) \tag{7}$$

where g is a non-linear function taking its previous state vector with the previous output word as inputs to update its state vector. c_i is a context vector to retrieve source inputs in the form of a weighted sum of the source vectors h_j, first taking as input the hidden state d_i at the top layer of a stacking LSTM (Hochreiter and Schmidhuber, 1997). The goal is to derive a context vector c_i that captures relevant source information to help predict the current target word y_i.

While these models differ in how the context vector c_i is derived, they share the same subsequent steps. c_i is calculated as follows:

$$c_i = \sum_{j=1}^{J} \alpha_{t,j} h_j \tag{8}$$

where h_j is the annotation of source word x_j and $\alpha_{t,j}$ is a weight for the j^{th} source vector at time step t to generate y_i:

$$\alpha_{t,j} = \frac{\mathit{exp}\left(\mathit{score}\left(d_i, h_j\right)\right)}{\sum_{j'=1}^{J} \mathit{exp}\left(\mathit{score}\left(d_i, h_{j'}\right)\right)} \tag{9}$$

The score function above may be defined in a variety of ways as discussed by Luong et al. (2015).

In this paper, we denote all the parameters to be optimized in the neural network as Θ and denote C as the dataset that contains source-target sentence pairs for the training phase. Hence, the learning objective is to seek the optimal parameters Θ^*:

$$\Theta^* = \arg\max_{\Theta} \sum_{(x,y)\in C} \sum_{(i=1)}^{I} \log P(y_t|y_{<t}, x; \Theta) \tag{10}$$

4 Method Description

Round-tripping involves two related translation tasks: the outbound-trip (source-target direction) and the inbound-trip (target-source direction). The defining traits of these forward and backward tasks are that they form a closed loop and both produce informative feedback that enables simultaneous training of the TMs.

We assume availability of: (1) monolingual datasets (C_X and C_Y) for the source and target languages; and (2) two weak TMs (emergent from training on initial small bilingual corpora) that bidirectionally translate sentences from source and target languages. The goal of the round-tripping approach is to augment the accuracy of the two TMs by employing the two monolingual datasets instead of a bilingual text.

We start by translating a sample sentence in one of the monolingual datasets, as the outbound-trip (forward) translation to the target language. This step generates more bilingual sentence pairs between the source and target languages. We then translate the resulting sentence pairs backward through the inbound-trip translation to the original language. This step finds high-quality sentences throughout the entirety of the generated sentence pairs. Evaluating the results of this round-tripping approach will provide an indication of the quality of the two TMs, and will enable their enhancement, accordingly. This process is iterated for several rounds until both TMs converge.

We define K_X as the number of sentences in C_X and K_Y as the number of sentences in C_Y. We take $P(.|S;\Theta_{XY})$ and $P(.|S;\Theta_{YX})$ to be two neural TMs in which Θ_{XY} and Θ_{YX} are supposed as their parameters. We also assume the existence of two LMs for languages X and Y, trained in advance either by using other resources or using the monolingual data (C_X and C_Y). Each LM takes a sentence as input and produces a real number, based on target-language fluency (LM correctness) together translation accuracy (TM correctness). This number represents the confidence of the translation quality of the sentence in its own language.

We start with a sentence in C_X and denote S_{sample} as a translation output sample. This step has a score as follows:

$$R_1 = LM_Y(S_{sample}) \tag{11}$$

The R_1 score indicates the well-formedness of the output sentence in language Y.

Given the translation output S_{sample}, we employ the log probability value of s recovered from the S_{sample} as the score of the construction:

$$R_2 = \log P(S|S_{sample};\Theta_{YX}) \tag{12}$$

We then adopt the LM score and construction score as the total reward score:

$$R_{total} = \alpha R_1 + (1-\alpha)R_2 \tag{13}$$

where α is an input hyper-parameter.

The total reward score is considered a function of S, S_{sample}, Θ_{XY} and Θ_{YX}. To maximize this score, we optimize the parameters in the TMs utilizing Stochastic Gradient Descent (SGD) (Sutton et al., 2000). According to the forward TM, we sample the s_{sample} and then compute the gradient of the expected score ($E[R_{total}]$), where E is taken from S_{sample}:

$$\nabla_{\Theta_{XY}} E[R_{total}] = E[R_{total}\nabla_{\Theta_{XY}} \log P(S_{sample}|S;\Theta_{XY})] \tag{14}$$

$$\nabla_{\Theta_{YX}} E[R_{total}] = E[(1-\alpha)\nabla_{\Theta_{YX}} \log P(S|S_{sample};\Theta_{YX})] \tag{15}$$

Algorithm 1 shows the round-tripping procedure.

```
Algorithm 1 Round-trip training for NMT
Input: Monolingual dataset in source and target languages ($C_X$ and $C_Y$), initial translation models in outbound and inbound trips ($\Theta_{XY}$ and $\Theta_{YX}$), language models in source and target languages ($LM_X$ and $LM_Y$), trade-off parameter between 0 and 1 ($\alpha$), beam search size ($N$), learning rates ($\gamma_{1,t}$ and $\gamma_{2,t}$).
1:  repeat:
2:      $t = t + 1$.
3:      Sample sentences $S_X$ and $S_Y$ from $C_X$ and $C_Y$ respectively.
4:      // Update model starting from language $X$.
        Set $S = S_X$.
5:      // Generate top-$N$ translations using $\Theta_{XY}$.
        Generate sentences $S_{sample,1}, ..., S_{sample,N}$.
6:      for $n = 1, ..., N$ do
7:          // Set LM score for $n^{th}$ sampled sentence.
            $R_{1,n} = LM_Y(S_{sample,n})$.
8:          // Set TM score for $n^{th}$ sampled sentence.
            $R_{2,n} = logP(S|S_{sample,n};\Theta_{YX})$.
9:          // Set total score of $n^{th}$ sampled sentence.
            $R_n = \alpha R_{1,n} + (1-\alpha)R_{2,n}$.
10:     end for
11:     // SDG computing for $\Theta_{XY}$.
        $\nabla_{\Theta_{XY}}\hat{E}[R_{total}] = \frac{1}{N}\sum_{n=1}^{N} [R_n\nabla_{\Theta_{XY}} \log P(S_{sample,n}|S;\Theta_{XY})]$.
12:     // SDG computing for $\Theta_{YX}$.
        $\nabla_{\Theta_{YX}}\hat{E}[R_{total}] = \frac{1}{N}\sum_{n=1}^{N} [(1-\alpha)\nabla_{\Theta_{YX}} \log P(S|S_{sample,n};\Theta_{YX})]$.
13:     // Model update.
        $\Theta_{XY} \leftarrow \Theta_{XY} + \gamma_{1,t}\nabla_{\Theta_{XY}}\hat{E}[R_{total}]$.
14:     // Model update.
        $\Theta_{YX} \leftarrow \Theta_{YX} + \gamma_{2,t}\nabla_{\Theta_{YX}}\hat{E}[R_{total}]$.
15:     // Update model starting from language $Y$.
        Set $S = S_Y$.
16:     Go through lines $5 - 14$ symmetrically.
17: until convergence.
```

To achieve reasonable translations we use beam search. We generate N-best sample translations and use the averaged value on the beam search results to estimate the true gradient value.[1]

[1] We used beam sizes 500 and 1000.

5 Experiments and Results

We apply the round-trip training approach to bilingual Persian-Spanish translation, and evaluate the results. We used the Persian-Spanish small bilingual corpora from the *Tanzil* corpus (Tiedemann, 2012),[2] which contains about 50K parallel sentence pairs. We also used 5K and 10K parallel sentences extracted from the *OpenSubtitles2018* collection (Tiedemann, 2012),[3] as the validation and test datasets, respectively. Finally, we utilized 70K parallel sentences from the *KDE4* corpus (Tiedemann, 2012),[4] as the monolingual data.

We implemented the DyNet-based model architecture (Mi et al., 2016) on top of *Mantis* (Cohn et al., 2016) which is an implementation of the attention sequence-to-sequence (Seq-to-Seq) NMT. For each language, we constructed a vocabulary with the most common 50K words in the parallel corpora, and OOV words were replace with a special token <*UNK*>. For monolingual corpora, sentences containing at least one OOV word were removed. Additionally, sentences with more than 80 words were removed from the training set.[5] The encoders and decoders make use of Long Short-Term Memory (LSTM) with 500 embedding dimensions, 500 hidden dimensions. For training, we used the SGD algorithm as the optimizer. The batch size was set as 64 with 20 batches pre-fetched and sorted by sentence lengths.

Below we compare the system based on the optimized round-trip training (round-tripping) through two translation systems; the first one is the standard NMT system (baseline), and the second one is the system that generates pseudo bilingual sentence pairs from monolingual corpora to assist the training step (self-training). For the pseudo-NMT we used the trained NMT model to generate pseudo bilingual sentence pairs from monolingual text, removed sentences with more than 80 words (as above), merged the generated data with the original parallel training data, and then trained the model for testing. Each of the translation systems was trained on a single GPU until their performances stopped improving on the validation set. This approach required an LM for each language. We trained an RNN-based LM (Mikolov et al., 2010) for each language using its corresponding monolingual corpus. The LM was then fixed and the log-likelihood of a received message was utilized for scoring the TM.

To start the round-trip training approach, the systems are initialized using warm-start TMs trained from initial small bilingual data. The goal is to see whether the round-tripping augments the baseline accuracy. We retrain the baseline systems by enlarging the initial small bilingual corpus: we add the optimized generated bilingual sentences to the initial parallel text. The new enlarged translation system contains both the initial and optimized generated bilingual text. For each translation task, we train the round-trip training approach.

We employ Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2001) (using *multi-bleu.perl* script from Moses) as the evaluation metric. BLEU is calculated for individual translated segments by comparing them with a data set of reference translations. The scores of each segment, ranging between 0 and 100, are averaged over the entire evaluation dataset to yield an estimate of the overall translation quality (higher is better).

The baseline systems for Persian-Spanish are first trained, while our round-trip method conducts joint training. We summarize the overall performances in Table 1:

NMT systems	Pe-Es	Es-Pe
baseline	31.12	29.56
self-train	29.29	27.36
round-trip	34.91	33.43

Table 1: BLEU scores for Persian-Spanish translation task and vice-versa.

As seen in Table 1, the round-tripping systems outperform the others in both translation directions. In Persian to Spanish translation, the round-tripping system outperforms the baseline by about 3.87 BLEU points and also outperforms the self-training system by about 6.07 BLEU points. In the back translation from Spanish to Persian, the improvement of the round-tripping outperforms both the baseline and self-training by about 3.79 and 5.62 BLEU points, respectively.

These results demonstrate the effectiveness of the round-trip training approach. The baseline systems outperform the self-training ones in all cases

[2] http://opus.nlpl.eu/Tanzil.php

[3] http://opus.nlpl.eu/OpenSubtitles-v2018.php

[4] http://opus.nlpl.eu/KDE4-v2.php

[5] The average sentence length is 47; an upper bound of 80 ensured exclusion of non-sentential and other spurious material.

because of the noise in the generated bilingual sentences used by self-training. Upon further examination, this result might have been expected given that the aim of round-trip training is to optimize the generated bilingual sentences by selecting the high-quality sentences to get better performance over self-training systems. When the size of bilingual corpus is small, the round-tripping makes a larger improvement. This outcome is an indication that round-trip training approach makes effective use of monolingual data.

Table 2 shows the performance of the baseline alongside of the enlarged translation systems, where the latter leverages the training text of the baseline and the round-tripping systems as well.

NMT systems	**Pe-Es**	**Es-Pe**
baseline	31.12	29.56
enlarged	34.21	33.03

Table 2: BLEU scores comparing the baseline and enlarged NMT systems for Persian-Spanish translation task and vice-versa.

As seen in Table 2, the BLEU scores of the enlarged NMT systems are better than the baseline ones in both translation directions. The enlarged system in the Persian-Spanish direction outperforms the baseline by about 3.47 BLEU points, and outperforms the baseline by about 3.09 BLEU points in the back translation. The improvements show that the optimized round-trip training system is promising for tackling the training data scarcity and it also helps to enhance translation quality.

6 Conclusions and Future Work

In this paper, we applied a round-tripping approach based on a retraining scenario to tackle training data scarcity in NMT systems. An exciting finding of this work is that it is possible to learn translations directly from monolingual data of the two languages. We employed a low-resource language pair and verified the hypothesis that, regardless of the amount of training resources, this approach outperforms the baseline. The results demonstrate that round-trip training is promising and better utilizes the monolingual data.

Many Artificial Intelligence (AI) tasks are naturally in dual form. Some examples are: (1) speech recognition paired with text-to-speech; (2) image captioning paired with image generation; and (3) question answering paired with question generation. Thus, a possible future direction would be to design and test the round-tripping approach for more tasks beyond MT. We note that round-tripping is not restricted to two tasks only. Indeed, the key idea is to form a closed loop so feedback signals are extracted by comparing the original input data with the final output data. Therefore, if more than two associated tasks form a closed loop, this approach can applied in each task for improvement of the overall model, even in the face of unlabeled data.

Acknowledgments

We would like to express our sincere gratitude to Dr. Michael W. Mislove (Tulane University, USA), Dr. Javier Serrano (Universitat Autònoma de Barcelona, Spain) and Dr. Gholamreza Haffari (Monash University, Australia) for all their support. We also would like to acknowledge the financial support received from the School of Science and Engineering, Tulane University (USA).

References

Benyamin Ahmadnia, Gholamreza Haffari, and Javier Serrano. 2018. Statistical machine translation for bilingually low-resource scenarios: A round-tripping approach. In *Proceedings of the 3rd IEEE International Conference on Machine Learning and Natural Language Processing*. pages 261–265.

Benyamin Ahmadnia, Gholamreza Haffari, and Javier Serrano. 2019. Round-trip training approach for bilingually low-resource statistical machine translation systems. *International Journal of Artificial Intelligence* 17(1):167–185.

Benyamin Ahmadnia and Javier Serrano. 2017. Employing pivot language technique through statistical and neural machine translation frameworks: The case of under-resourced persian-spanish language pair. *International Journal on Natural Language Computing* 6(5):37–47.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pages 858–867.

Trevor Cohn, Cong Duy Vu Huang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza

Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*. pages 876–885.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *CoRR* abs/1503.03535.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Proceedings of the 30th Conference on Neural Information Processing Systems*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. pages 18–24.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007* .

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. pages 11–19.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. pages 152–159.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*. pages 2283–2288.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*. pages 1045–1048.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. pages 311–318.

Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. 1998. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(1):23–38.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323:533–536.

Holger Schwenk. 2008. Investigations on large-scale lightly-supervised training for statistical machine translation. In *Proceedings of IWSLT*. pages 182–189.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of Association for Computational Linguistics*.

Richard S. Sutton, David A. Mcallester, Satinder P. Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*. volume 12, pages 1057–1063.

Aram Ter-Sarkisov, Holger Schwenk, Loïc Barrault, and Fethi Bougares. 2015. Incremental adaptation strategies for neural network language models. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. pages 48–56.

Jørg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2008. On using monolingual corpora in statistical machine translation. *Journal of Machine Translation* .