Using Target-side Monolingual Data for Neural Machine Translation through Multi-task Learning

Tobias Domhan and **Felix Hieber**
Amazon
Berlin, Germany
{domhant,fhieber}@amazon.com

Abstract

The performance of Neural Machine Translation (NMT) models relies heavily on the availability of sufficient amounts of parallel data, and an efficient and effective way of leveraging the vastly available amounts of monolingual data has yet to be found. We propose to modify the decoder in a neural sequence-to-sequence model to enable multi-task learning for two strongly related tasks: target-side language modeling and translation. The decoder predicts the next target word through two channels, a target-side language model on the lowest layer, and an attentional recurrent model which is conditioned on the source representation. This architecture allows joint training on both large amounts of monolingual and moderate amounts of bilingual data to improve NMT performance. Initial results in the news domain for three language pairs show moderate but consistent improvements over a baseline trained on bilingual data only.

1 Introduction

In recent years, neural encoder-decoder models (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014) have significantly advanced the state of the art in NMT, and now consistently outperform Statistical Machine Translation (SMT) (Bojar et al., 2016). However, their success hinges on the availability of sufficient amounts of parallel data, and contrary to the long line of research in SMT, there has only been a limited amount of work on how to effectively *and efficiently* make use of monolingual data which is typically amply available. We propose a modified neural sequence-to-sequence model with attention (Bahdanau et al., 2014; Luong et al., 2015b) that uses multi-task learning on the decoder side to jointly learn two strongly related tasks: target-side language modeling and translation. Our approach does not require any pre-translation or pre-training to learn from monolingual data and thus provides a principled way to integrate monolingual data resources into NMT training.

2 Related Work

Gülçehre et al. (2015) investigate two ways of integrating a pre-trained neural Language Model (LM) into a pre-trained NMT system: shallow fusion, where the LM is used at test time to rescore beam search hypothesis, requiring no additional fine-tuning and deep fusion, where hidden states of NMT decoder and LM are concatenated before making a prediction for the next word. Both components are pre-trained separately and fine-tuned together.

More recently, Sennrich et al. (2016) have shown significant improvements by *back-translating* target-side monolingual data and using such synthetic data as additional parallel training data. One downside of this approach is the significantly increased training time, due to training of a model in the reverse direction and translation of monolingual data. In contrast, we propose to train NMT models from scratch on both bilingual and target-side monolingual data in a multi-task setting.

Our approach aims to exploit the signals from target-side monolingual data to learn a strong language model that supports the decoder in making translation decisions for the next word. Our approach further relates to Zhang and Zong (2016), who investigate multi-task learning for sequence-to-sequence models by strengthening the encoder using source-side monolingual data. A shared encoder architecture is used to predict both, transla-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1499–1504
Copenhagen, Denmark, September 7–11, 2017.

tions of parallel source sentences and permutations of monolingual source sentences. In this paper we focus on target-side monolingual data and only update encoder parameters based on existing parallel data.

In a broader context, multi-task learning has shown to be effective in the context of sequence-to-sequence models (Luong et al., 2015a), where different parts of the network can be shared across multiple tasks.

3 Neural Machine Translation

We briefly recap the baseline NMT model (Bahdanau et al., 2014; Luong et al., 2015b) and highlight architectural differences of our implementation where necessary.

Given source sentence $\mathbf{x} = x_1, ..., x_n$ and target sentence $\mathbf{y} = y_1, ..., y_m$, NMT models $p(\mathbf{y}|\mathbf{x})$ as a target language sequence model, conditioning the probability of the target word y_t on the target history $\mathbf{y}_{1:t-1}$ and source sentence $\mathbf{x}$. Each x_i and y_t are integer ids given by source and target vocabulary mappings, $\mathbf{V}_{src}, \mathbf{V}_{trg}$, built from the training data tokens. The target sequence is factorized as:

$$p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{t=1}^{m} p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}; \boldsymbol{\theta}). \quad (1)$$

The model, parameterized by $\boldsymbol{\theta}$, consists of an *encoder* and a *decoder* part (Sutskever et al., 2014).

For training set $\mathbb{P}$ consisting of parallel sentence pairs $(\mathbf{x}, \mathbf{y})$, we minimize the cross-entropy loss w.r.t $\boldsymbol{\theta}$:

$$\mathcal{L}_{\boldsymbol{\theta}} = \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{P}} -\log p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}). \quad (2)$$

Encoder Given source sentence $\mathbf{x} = x_1, ..., x_n$, the encoder produces a sequence of hidden states $\mathbf{h}_1 \dots \mathbf{h}_n$ through an Recurrent Neural Network (RNN), such that:

$$\overrightarrow{\mathbf{h}}_i = f_{enc}(\mathbf{E}_S \mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i-1}), \quad (3)$$

where $\mathbf{h}_0 = \mathbf{0}$, $\mathbf{x}_i \in \{0,1\}^{|\mathbf{V}_{src}|}$ is the one-hot encoding of x_i, $\mathbf{E}_S \in \mathbb{R}^{e\times|\mathbf{V}_{src}|}$ is a source embedding matrix with embedding size e, and f_{enc} some non-linear function, such as the Gated Rectified Unit (GRU) (Cho et al., 2014) or a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) network.

Attentional Decoder The decoder also consists of an RNN to predict one target word at a time through a *state vector* $\mathbf{s}$:

$$\mathbf{s}_t = f_{dec}([\mathbf{E}_T \mathbf{y}_{t-1}; \bar{\mathbf{s}}_{t-1}], \mathbf{s}_{t-1}), \quad (4)$$

where $\mathbf{y}_{t-1} \in \{0,1\}^{|\mathbf{V}_{trg}|}$ is the one-hot encoding of the previous target word, $\mathbf{E}_T \in \mathbb{R}^{e\times|\mathbf{V}_{trg}|}$ the target word embedding matrix, f_{dec} an RNN, $\mathbf{s}_{t-1}$ the previous state vector, and $\bar{\mathbf{s}}_{t-1}$ the source-dependent *attentional vector*. The initial decoder hidden state is a non-linear transformation of the last encoder hidden state: $\mathbf{s}_0 = \tanh(\mathbf{W}_{init}\mathbf{h}_n + \mathbf{b}_{init})$. The attentional vector $\bar{\mathbf{s}}_t$ combines the decoder state with a *context vector* $\mathbf{c}_t$:

$$\bar{\mathbf{s}}_t = \tanh(\mathbf{W}_{\bar{s}}[\mathbf{s}_t; \mathbf{c}_t]), \quad (5)$$

where $\mathbf{c}_t$ is a weighted sum of encoder hidden states: $\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{ti}\mathbf{h}_i$ and brackets denote vector concatenation.

The attention vector $\boldsymbol{\alpha}_t$ is computed by an attention network (Bahdanau et al., 2014; Luong et al., 2015b):

$$\alpha_{ti} = softmax(score(\mathbf{s}_t, \mathbf{h}_i))$$
$$score(\mathbf{s}, \mathbf{h}) = \mathbf{v}_a^\top \tanh(\mathbf{W}_u\mathbf{s} + \mathbf{W}_v\mathbf{h}). \quad (6)$$

The next target word y_t is predicted through a softmax layer over the attentional vector $\bar{\mathbf{s}}_t$:

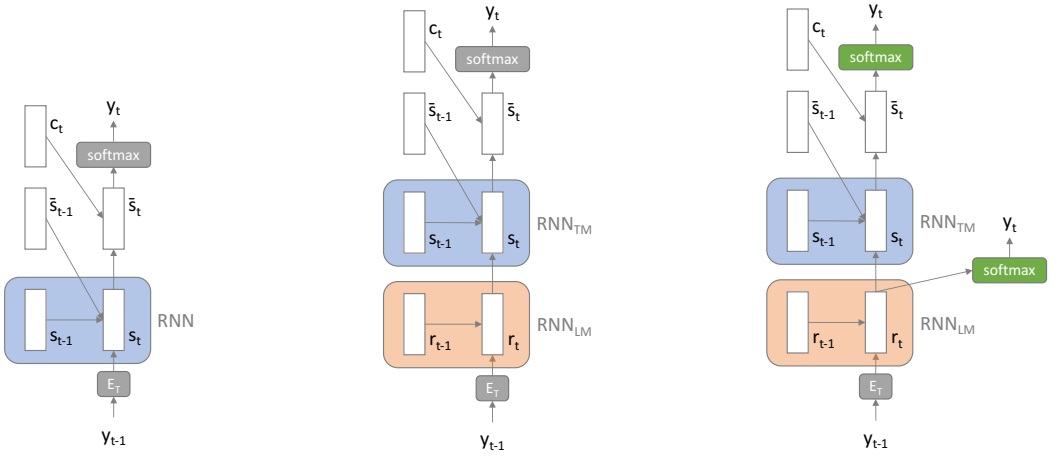
$$p(y_t|\mathbf{y}_{1:t-1}, \mathbf{x}; \boldsymbol{\theta}) = softmax(\mathbf{W}_o\bar{\mathbf{s}}_t + \mathbf{b}_o) \quad (7)$$

where $\mathbf{W}_o$ maps $\bar{\mathbf{s}}_t$ to the dimension of the target vocabulary. Figure 1a depicts this decoder architecture. Note that source information from $\mathbf{c}$ indirectly influences the states $\mathbf{s}$ of the decoder RNN as it takes $\bar{\mathbf{s}}$ as one of its inputs.

4 Incorporating Monolingual Data

4.1 Separate Decoder LM layer

The decoder RNN (Figure 1a) is essentially a target-side language model, additionally conditioned on source-side sequences. Such sequences are not available for monolingual corpora and previous work has tried to overcome this problem by either using synthetically generated source sequences or using a `NULL` token as the source sequence (Sennrich et al., 2016). As previously shown empirically, the model tends to "forget" source-side information if trained on much more monolingual than parallel data.



(a) baseline (b) +LML (c) +LML +MTL

Figure 1: Illustration of the proposed decoder architecture. (a) Baseline model with a single-layer decoder RNN and attention (b) Addition of a source-independent LM layer that feeds into the source-dependent decoder (c) Multi-task setting next-word prediction from both layers; green softmax layers are shared.

In our approach we explicitly define a source-independent network that only learns from target-side sequences (a language model), and a source-dependent network on top, that takes information from the source sequence into account (a translation model) through the attentional vector $\bar{\mathbf{s}}$. Formally, we modify the decoder RNN of Equation 4 to operate on the outputs an LM layer, which is independent of any source-side information:

$$\mathbf{s}_t = f_{dec}([\mathbf{r}_t; \bar{\mathbf{s}}_{t-1}], \mathbf{s}_{t-1}) \quad (8)$$

$$\mathbf{r}_t = f_{lm}(\mathbf{E}_T \mathbf{y}_{t-1}, \mathbf{r}_{t-1}) \quad (9)$$

Figure 1b illustrates this separation graphically.

4.2 Multi-task Learning

The separation from above allows us to train the target embeddings $\mathbf{E}_T$ and f_{lm} parameters from monolingual data, concurrent to training the rest of the network on bilingual data. Let us denote the source-independent parameters by $\boldsymbol{\sigma}$. We connect a second loss to f_{lm} to predict the next target word also conditioned only on target history information (Figure 1c). Parameters for softmax layers are shared such that predictions of the LM layer are given by:

$$p(y_t|\mathbf{y}_{1:t-1}, \boldsymbol{\sigma}) = softmax(\mathbf{W}_o \mathbf{r}_t + \mathbf{b}_o). \quad (10)$$

Formally, for a heterogeneous data set $\mathbb{Z} = \{\mathbb{P}, \mathbb{M}\}$, consisting of parallel and monolingual sentences $(\mathbf{x}, \mathbf{y}), (\mathbf{y})$, we optimize the following joint loss:

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\sigma}} = \frac{1}{|\mathbb{P}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{P}} -\log p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) + \gamma \frac{1}{|\mathbb{M}|} \sum_{\mathbf{y}\in\mathbb{M}} -\log p(\mathbf{y};\boldsymbol{\sigma}), \quad (11)$$

where the source-independent parameters $\boldsymbol{\sigma} \subset \boldsymbol{\theta}$ are updated by gradients from both mono- and parallel data examples, and source-dependent parameters $\boldsymbol{\theta}$ are updated only through gradients from parallel data examples. $\gamma \geq 0$ is a scalar to influence the importance of the monolingual loss. In practice, we construct mini-batches of training examples, where 50% of the data is parallel, and 50% of the data is monolingual and set $\gamma = 1$.

Since parts of the decoder are shared among both tasks and we optimize both loss terms concurrently, we view this approach as an instance of multi-task learning rather than transfer learning, where optimization is typically carried out sequentially.

5 Experiments

We conduct experiments for three different language pairs in the news domain: FR→EN, EN→DE, and CS→EN.

5.1 Data

For EN→DE and CS→EN we use *news-commentary-v11* as bilingual training data, *NewsCrawl 2015* as monolingual data, and news development and test sets from

System	Data	EN→DE	FR→EN	CS→EN
baseline		20.3 39.9 63.0	21.7 27.5 59.1	17.0 24.4 65.2
+ LML		20.4 39.8 63.1	21.3 27.2 59.8	16.9 24.4 65.4
+ LML + MTL	+ mono	21.4 40.8 61.4	22.3 27.7 58.3	17.2 24.7 64.3
Sennrich et al. (2016)	+ synthetic	24.4 43.4 56.4	27.4 31.5 52.1	21.2 27.5 59.4
ensemble baseline		22.2 41.6 60.6	23.9 29.1 56.4	18.3 25.5 63.0
+ LML		22.4 41.8 60.9	23.5 28.7 57.2	18.3 25.6 63.4
+ LML + MTL	+ mono	23.6 42.8 58.9	24.2 29.2 55.9	18.8 25.9 62.2
ensemble Sennrich et al. (2016)	+ synthetic	25.7 44.6 55.0	29.1 32.6 50.3	22.5 28.4 57.8

Table 1: BLEU/METEOR/TER scores on test sets for different language pairs. For BLEU and METEOR higher is better. For TER lower is better.

WMT2016 (Bojar et al., 2016). For FR→EN we use *newscommentary-v9* as bilingual data, *NewsCrawl 2009-13* as monolingual data, and news development and test sets from WMT 2014 (Bojar et al., 2014). The number of sentences for these corpora is shown below:

Data Set	bilingual	monolingual
EN→DE	242,770	51,315,088
FR→EN	183,251	51,995,709
CS→EN	191,432	27,236,445

5.2 Experimental Setup

We tokenize all data and apply Byte Pair Encoding (BPE) (Sennrich et al., 2015) with 30k merge operations learned on the joined bilingual data. Models are evaluated in terms of BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2009) and TER (Snover et al., 2006) on tokenized, cased test data. Decoding is performed using beam search with a beam of size 5. We implement all models using MXNet (Chen et al., 2015)[1].

Baselines Our baseline model consists of a 1-layer bi-directional LSTM encoder with an embedding size of 512 and a hidden size of 1024. The 1-layer LSTM decoder with 1024 hidden units uses an attention network with 256 hidden units. The model is optimized using Adam (Kingma and Ba, 2014) with a learning rate of 0.0003, no weight decay and gradient clipping if the norm exceeds 1.0. The batch size is set to 64 and the maximum sequence length to 100. Dropout (Srivastava et al., 2014) of 0.3 is applied to source word embeddings and outputs of RNN cells. We initialize all RNN parameters with orthogonal matrices (Saxe et al., 2013) and the remaining parameters with the Xavier (Glorot and Bengio, 2010) method. We use early stopping with respect to perplexity on the development set. We train each model configuration three times with different seeds and report average metrics across the three runs.

Further, we train models with synthetic parallel data generated through *back-translation* (Sennrich et al., 2016). For this, we first train a baseline model in the reverse direction and then translate a random sample of 200k sentences from the monolingual target data. On the combined parallel and synthetic training data we train a new model with the same training hyper-parameters as the baseline.

Language Model Layer The architecture with an additional source-independent LM layer (+LML) is trained with the same hyper-parameters and data as the baseline model. The LM RNN uses a hidden size of 1024. The multi-task system (+LML + MTL) is trained on both parallel and monolingual data. In practice, all +LML +MTL models converge before seeing the entire monolingual corpus and at about the same number of updates as the baseline.

6 Results

Table 1 shows results on the held-out test sets. We observe that a separate LM layer does not significantly impact performance across all metrics. Adding monolingual data in the described multi-task setting improves translation performance by a small but consistent margin across all metrics. Interestingly, the improvements from monolingual data are additive to the gains from ensembling of

[1]Baseline systems are equivalent to an earlier version of Sockeye: https://github.com/awslabs/sockeye

3 models with different random seeds. However, the use of synthetic parallel data still outperforms our approach both in single and ensemble systems.

While separating out a language model allowed us to carry out multi-task training on mixed data types, it constrains gradients from monolingual data examples to a subset of source-independent network parameters (σ). In contrast, synthetic data always affects all network parameters (θ) and has a positive effect despite source sequences being noisy. We speculate that training from synthetic source data may also act as a model regularizer.

7 Conclusion

We proposed a way to directly integrate target-side monolingual data into NMT through multi-task learning. Our approach avoids costly pre-training processes and jointly trains on bilingual and monolingual data from scratch. While initial results show only moderate improvements over the baseline and fall short against using synthetic parallel data, we believe there is value in pursuing this line of research further to simplify training procedures.

References

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

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT'14)*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT'16)*, pages 131–198.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*.

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'14)*, pages 1724–1734, Doha, Qatar.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AIStats*, volume 9, pages 249–256.

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.

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

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, Seattle.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Alon Lavie and Michael J. Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114.

Minh-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 (EMNLP'15)*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

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 (ACL'02)*, Philadelphia, Pennsylvania.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the*

54th Annual Meeting of the Association for Computational Linguistics, ACL'16, Berlin, Germany.

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,*.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of EMNLP*.