An Effective Optimization Method for Neural Machine Translation: The Case of English-Persian Bilingually Low-Resource Scenario

Benyamin Ahmadnia
Department of Linguistics
University of California, Davis, USA
ahmadnia@ucdavis.edu

Raul Aranovich
Department of Linguistics
University of California, Davis, USA
raranovich@ucdavis.edu

Abstract

In this paper, we propose a useful optimization method for low-resource Neural Machine Translation (NMT) by investigating the effectiveness of multiple neural network optimization algorithms. Our results confirm that applying the proposed optimization method on English-Persian translation can exceed translation quality compared to the English-Persian Statistical Machine Translation (SMT) paradigm.

1 Introduction

Employing neural networks in Machine Translation (MT) significantly reduces the time-consuming and laborious operation steps such as word alignment, phrase extraction, feature selection, etc. Although the quality of Neural MT (NMT) models heavily rely on the quantity as well as the quality of the training dataset, considering the low-resource condition, the impact of NMT is still not as much as the Statistical Machine Translation (SMT). NMT has recently achieved great success, which surpasses SMT in many high-resource language pairs, and it has become the MT approach.

In this paper, we compare the impact of multiple neural network optimization algorithms under with respect to the low-resource condition, and then, we proposes an effective optimization method for our case-study language pair. The motivation for choosing English and Persian as the case-study is the linguistic differences between these languages, which are from different language families and have significant differences in their properties, may pose a challenge for MT.

Following Ahmadnia and Dorr (2019), low-resource languages are those that have fewer technologies and datasets relative to some measure of their international importance. In simple words, the languages for which bilingual training data is extremely sparse, requiring recourse to techniques that are complementary to standard MT approaches. 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 (Ahmadnia et al., 2017). 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.

Our results show that the proposed optimization algorithm for English-Persian NMT works well and improves translation results compared to the English-Persian SMT paradigm.

This paper is organized as follows; Section 2 describes the methodology. The experimental results and analysis are covered by Section 3. Section 4 investigates the previous related work. Conclusions and future work are provided in Section 5.

2 Methodology

NMT originates from sequence-to-sequence learning. So, in this paper, we take the attention-based (Attentional) NMT model.

2.1 Attention-based NMT

Attentional NMT (Bahdanau et al., 2015) models are divided into three parts;

- **Encoder** that encodes the source sentences into vector sequences as source language representations.
- **Decoder** that acquires the source context information through attention mechanism and generates target word sequences in turn.

Proceedings of the 7th Workshop on Asian Translation, pages 45–49
December 4, 2020. ©2020 Association for Computational Linguistics

- **Attention mechanism** that connects encoders and decoders to make the whole model interrelated.

In NMT module, a source sentence $x = x_1, x_2, ..., x_J$ is encoded into an internal representation $h = h_1, h_2, ..., h_J$, and then h is decoded into a target sentence $y = y_1, y_2, ..., y_I$. For example, to translate an English sentence *the dog likes to eat an apple* into Persian, each word is transformed into a *1-hot* encoding vector (with a single 1 associated with the index of that word, and all other indexed values 0). Each word in the dataset has a distinct 1-hot encoding vector that serves as a numerical representation that serves as input to the model. The first step toward creating these vectors is to assign an index to each unique word in English (as the input language). This process is then repeated for Persian (as the output language). The assignment of an index to each unique word creates a vocabulary for each language.

The encoder portion of the NMT model takes a sentence in English and creates a representational vector from this sentence. This vector represents the meaning of the sentence and is subsequently passed to a decoder which outputs the translation of the sentence in Persian. NMT models the conditional probability of the target sentence as:

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

where y_i is the target word emitted by the decoder at step i and $y_{<_i} = (y_1, y_2, ..., y_{i-1})$. The conditional output probability of a target word y_i defined as follows:

$$P(y_i|y_{<i}, x) = softmax\ (f(d_i, y_{i-1}, c_i)) \tag{2}$$

where f is a non-linear function and $d_i = g(d_{i-1}, y_{i-1}, c_i)$, g is a non-linear function. c_i is a context vector computed as the weighted sum of the hidden vectors h_j,

$$c_i = \sum_{j=1}^{J} \alpha_{t,j} h_j, \tag{3}$$

where h_j is the annotation of source word x_j, $\alpha_{t,j}$ is computed by what is known as the *attentional model*, which focuses on sub-parts of the sentence during translation:

$$\alpha_{ij} = \frac{exp\ (score\ (d_i, h_j))}{\sum_{j'=1}^{J} exp\ (score\ (d_i, h_{j'}))} \tag{4}$$

The *score* function above can be defined in some different ways as discussed by Luong et al. (2015).

The attention mechanism supports memorization of long source sentences in NMT. Rather than building a single context vector out of the encoder's last hidden state, an attention model creates shortcuts between the context vector and the entire source input. The weights of these shortcut connections are customizable for each output element.

The context vector has access to the entire input sequence—for retention of the full context of the sentence—and controls the alignment between the source and target. Stated simply: the attention mechanism converts two sentences into a matrix where the words of one sentence form the columns, and the words of another sentence form the rows. From this, matches are obtained, thus identifying the relevant and yielding a positive impact on MT. Apart from improving the performance on MT, attention-based networks allow models to learn alignments between different modalities (different data types) for e.g., between speech frames and text or between visual features of a picture and its text description.

2.2 Optimization Method

Following Ahmadnia and Dorr (2020), given a training dataset with N bilingual sentences, an attentional NMT training loss function is defined as the conditional log-likelihood:

$$Loss = \sum_{n=1}^{N} \sum_{i=1}^{I} -logP(y_i^n|y_{<i}^n, x^n) \tag{5}$$

The performance of NMT systems is determined by the method of model optimization. Three typical model optimization methods are as follows:

- **Adam** that combines the best properties of the "AdaGrad" and "RMSProp" algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Its main advantage is that after offset correction, the learning rate of each iteration has a certain range, which makes the parameters more stable. Also, different parameters have different adaptive learning rates, which are suitable for large-scale data sets and high-dimensional parameter space (Kingma and Ba, 2015).

- **Adadelta** which is an extension of "Adagrad" that reduces its aggressive, monotonically decreasing learning rate. Instead of accumulating all past squared gradients, Adadelta restricts the window of accumulated past gradients to some fixed size. Its advantage is that the learning rate is adaptive, and the experimental results are reasonable. However, the disadvantage is that the convergence speed is slower (Zeiler, 2012).

- **Stochastic Gradient Descent (SGD)** as an iterative method for optimizing an objective function with suitable smoothness properties can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient by an estimate thereof. Its advantage is that it is simple to implement and the experimental results are more stable and reliable under the appropriate learning rate scheduling scheme. While its disadvantage is that it is difficult to select the appropriate learning rate (Robbins and Monro, 1951).

The SGD method calculates the gradient in each iteration of training corpus, and then updates the model parameters which is the most basic optimization method of neural network model. In this paper, this method refers to the mini-batch gradient descent.

In the following section, we compare the different model optimization methods and make a comparative analysis of their application impact on English-Persian attentional NMT.

3 Experimental Results

For the experiments, we utilized TEP[1] (Pilevar et al., 2011) English-Persian parallel corpus that contains about 594K sentences. We allocated ≈550K sentences to training step, ≈10K sentences to validation step, and ≈30K sentences to testing step. We employed Byte Pair Encoding (BPE) (Sennrich et al., 2016b) as an effective way to overcome the unknown word problem in standard NMT. In the experiments, we limited the vocabulary size to the most frequent 10K tokens and replacing the rest with a special token <UNK>. We accelerate training by discarding all sentences with more than 30 elements (either BPE units or actual tokens). The vector dimension of bilingual words is 512, the size of hidden layer is 1024, the beam size is 10, the size of mini-batch is 80, and the dropout of output layer is set to 0.1. In order to reduce the problem of unlisted words, the size of Persian and English dictionaries is set to 20K to cover about 95% words. In order to reduce fitting, we set epoch, as the maximum number of training rounds, to 60. BLEU (Papineni et al., 2001) is our standard evaluation metric.

We employed the following experimental systems:

- **Moses:** We adapted the baseline system on top of Moses (Koehn et al., 2007) as a standard phrase-based SMT.
- **RNNSearch:** Which is compared with the method using by the paper under the same experimental setting.
- **Mantis:** We employed Mantis (Cohn et al., 2016) on top of DyNet as the attentional NMT open source system. Its cycle unit is Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) that in which, the default parameter configuration is used.
- **Mantis+Adadelta/SGD/Adam:** which is used as the optimization method of model parameters for English-Persian NMT system.

Mantis+SGD+DC represents learning rate Decay when the iteration exceeds 40 rounds, and the decay rate is 0.1.

As seen in Table 1, the translation impacts of Model 3, Model 4 and Model 8 are lower than SMT (Model 1). It shows that English-Persian attention-based NMT is ineffective considering the low-resource conditions. Also, the results conform to the characteristics of the general low-resource NMT systems.

The results of Model 4, Model 5, Model 6, Model 8, and Model 10 demonstrate that increasing learning rate can extremely improve the English-Persian translation quality where when the optimization method of SGD and Adam were employed. However, when the learning rate is too high, the performance of translation system will be reduced. Therefore, it can be seen in the case of low-resource conditions, the actual NMT system is sensitive in various model optimization methods and corresponding learning rates. So, selecting the appropriate model optimization method and learning rate has a great influence on the final translation results.

Furthermore, Model 7 has achieved the highest translation impact, which surpasses the SMT

[1] http://opus.nlpl.eu/TEP.php

Model	Translation system	Learning rate	BLEU
1	Moses	–	34.80
2	RNNSerach	–	38.51
3	Mantis+Adadelta	–	32.06
4	Mantis+SGD	1	18.66
5	Mantis+SGD	2	38.42
6	Mantis+SGD	3	35.35
7	Mantis+SGD+DC	2	**39.46**
8	Mantis+Adam	0.001	34.90
9	Mantis+Adam	0.002	38.73
10	Mantis+Adam	0.003	37.64

Table 1: English-Persian translation results.

system using Moses and the NMT system using RNNSearch. It can be found that English-Persian NMT still achieves better translation results by adopting higher learning rates and learning rate scheduling strategies with fewer corpus when choosing appropriate model optimization methods.

Figure 1 shows the convergence curves of different system models.

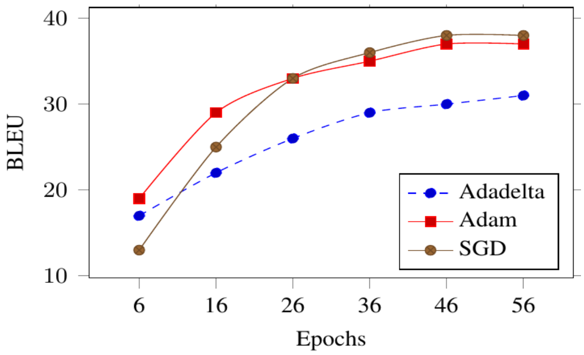

Figure 1: Contrast of the convergence rates of various translation systems.

Also, from the learning curve, it can be found that the system using Adadelta optimization method converges slowly, and the corresponding translation effect is the worst. Adam optimization method converges quickly. The SGD optimization method uses a large learning rate and achieves the effect of Adam optimization method in about 26 rounds. When the execution learning rate decreases, the translation performance can further be improved, and ultimately the best translation impact can be achieved.

4 Related Work

To construct pseudo bilingual corpus, various useful methods have already been proposed; 1) Back-translation (Sennrich et al., 2016a), 2) Dual learning (He et al., 2016), and 3) Round-tripping (Ahmadnia et al., 2018; Ahmadnia and Dorr, 2019). Also, integrating additional language models to use monolingual corpus (Zoph et al., 2016), using transfer learning to transfer the model of high-resource language pairs to low-resource ones, etc. The core idea of the mentioned approaches is to integrate more external resources so that the NMT model can sufficiently acquire translation knowledge and augment translation quality. Although the above methods have practically achieved remarkable results, the disadvantage is that the application effect is limited by the quality of external (generated) sentences.

In contrast the existing work, this paper compares various optimization methods of NMT models, and proposes a translation model optimization method which is useful in low-resource condition to enhance the effectiveness of English-Persian NMT. Our method does not employ any additional (generated) resources and has certain generality.

5 Conclusions and Future Work

In this paper, an effective optimization method for bilingually low-resource NMT models was applied to English-Persian translation. The investigated optimization method significantly enhances the impact of the English-Persian NMT system, and surpasses the SMT system and the previous similar work, which achieves the best translation results. Noting worth that our optimization method not only does not depend on external resources but also it has language independence. As a future work, we want to investigate other methods of NMT to enhance effects in low-resource conditions, and the application of this method to other languages.

Acknowledgments

The authors would like to acknowledge the financial support received from the Department of Linguistics at UC Davis (USA).

References

Benyamin Ahmadnia and Bonnie J. Dorr. 2019. Augmenting neural machine translation through round-trip training approach. *Open Computer Science*, 9(1):268–278.

Benyamin Ahmadnia and Bonnie J. Dorr. 2020. Impact of a new word embedding cost function on farsi-spanish low-resource neural machine translation. In *Proceedings of the Thirty-Third International FLAIRS Conference*, pages 222–227.

Benyamin Ahmadnia, Gholamreza Haffari, and Javier Serrano. 2018. Statistical machine translation for bilingually low-resource scenarios: A round-tripping approach. In *Proceedings of the IEEE 5th International Congress on Information Science and Technology*, pages 261–265.

Benyamin Ahmadnia, Javier Serrano, and Gholamreza Haffari. 2017. Persian-Spanish low-resource statistical machine translation through English as pivot language. In *Proceedings of Recent Advances in Natural Language Processing*, pages 24–30.

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*.

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.

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*.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/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 Association for Computational Linguistics Companion*, pages 177–180.

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.

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.

Mohammad Taher Pilevar, Heshaam Faili, and Abdol Hamid Pilevar. 2011. Tep: Tehran english-persian parallel corpus. In *Proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 68–79.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.