Sainik Kumar Mahata*, Dipankar Das and Sivaji Bandyopadhyay

MTIL2017: Machine Translation Using Recurrent Neural Network on Statistical Machine Translation

https://doi.org/10.1515/jisys-2018-0016
Received January 9, 2018.

Abstract: Machine translation (MT) is the automatic translation of the source language to its target language by a computer system. In the current paper, we propose an approach of using recurrent neural networks (RNNs) over traditional statistical MT (SMT). We compare the performance of the phrase table of SMT to the performance of the proposed RNN and in turn improve the quality of the MT output. This work has been done as a part of the shared task problem provided by the MTIL2017. We have constructed the traditional MT model using Moses toolkit and have additionally enriched the language model using external data sets. Thereafter, we have ranked the phrase tables using an RNN encoder-decoder module created originally as a part of the GroundHog project of LISA lab.

Keywords: Machine translation, recurrent neural network, statistical machine translation, language model.

1 Introduction

Machine translation (MT) is automated translation [19]. MT is the process that uses computer software to translate a text from one natural language (such as English) into another (such as Spanish). Translation itself is a challenging task for humans and is no less challenging for computers because it deals with natural languages. High-quality translation requires a thorough understanding of the source text and its intended function as well as good knowledge of the target language. In an MT system, the translation process must be completely formalized, which is a daunting task because the process is by no means completely understood until date.

Statistical MT (SMT) addresses this challenge by analyzing the output of human translators with statistical methods and extracting implicit information about the translation process from corpora of translated texts [16]. SMT has shown good results for many language pairs and has had its share in the recent surge in terms of MT popularity among the public.

On the contrary, despite being relatively new, neural MT (NMT) has already shown promising results [4, 12, 17], achieving state-of-the-art performances for various language pairs [9, 11, 18, 20, 21, 24]. NMT is a simple new architecture for getting machines to learn how to translate because it is based on the model of neural networks of human brains, with information being sent to different layers to be processed before generating final output. It is said to create much more accurate translations than SMT [6, 7, 15, 26].

In the current work, we have extracted the data provided by MTIL2017 (http://nlp.amrita.edu/mtil_cen/), which had 162k pairs of parallel data for English-Hindi bilingual data. We trained an SMT system using the Moses toolkit (http://www.statmt.org/moses/) [13]. The main idea was to score the phrase table that will be generated by Moses, with a recurrent neural network (RNN) encoder-decoder system that will also be trained using the same data set that was used to train Moses. Moses itself scores phrases as a part of its decoding process. Additionally, we try to score the phrases using the RNN encoder-decoder system to find out if the

***Corresponding author: Sainik Kumar Mahata,** Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, India, e-mail: sainik.mahata@gmail.com
Dipankar Das and Sivaji Bandyopadhyay: Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, India

results improve or not. Sentences with the highest scores are selected as output. Section 2 discusses the related approaches followed by the proposed approach in Section 3. The results of the proposed approach, some additional experiments, and discussion are covered in Sections 4 to 6, respectively.

2 Related Work

Before reporting the proposed model, we have to look into the recent works that have focused on the use of neural networks on SMT to improve the quality of translation. Schwenk [23], in his paper, proposed to use a feed-forward neural network to score phrase pairs. He used a feed-forward neural network with fixed-size inputs consisting of seven words, which has zero padding for shorter phrases. The system also had fixed-size output consisting of seven words for the output. But whenever we talk about real-world translations, the phrase length may often vary. Thereby, it is important that the neural network model that is used should be able to handle variable-length phrases. For this purpose, we decided on using RNNs. Similar to Schwenk's paper, Devlin et al. [5] also used a feed-forward neural network to generate translations, but they predicted one word in a target phrase at a time. The system had impressive performance over the model discussed before. But again, the use of feed-forward neural networks demands the use of fixed-size phrases to work properly. Zou et al. [27] proposed to learn bilingual embedding of words/phrases, which they used to compute distance between phrase pairs and which then was used as an additional annotation to score the phrase pairs of an SMT system. In their paper, Chandar et al. [3] trained a feed-forward neural network to learn the mapping of an input phrase to an output phrase using the bag-of-words approach. This is closely related to the model proposed in Schwenk's paper, except that their input representation of a phrase is a bag-of-words. Gao et al. [8] proposed a similar approach of using bag-of-words as well. A similar encoder-decoder approach that used two RNNs was proposed by Socher et al. [25], but their model was restricted to a monolingual setting. More recently, another encoder-decoder model using an RNN was proposed by Auli et al. [2], where the decoder is conditioned on a representation of either a source sentence or a source context. Kalchbrenner and Blunsom [12], in their paper, proposed a similar model that uses the concept of an encoder and decoder. They used a convolutional n-gram model (CGM) for the encoder part and a combination of inverse CGM and a RNN for the decoder part. The evaluation of their model was based on rescoring the n-best list of the phrases from the phrase table of SMT.

3 Proposed Model

3.1 Moses

Moses is an SMT system that allows you to automatically train translation models for any language pair when trained with a large collection of translated texts (parallel corpus). Once the model has been trained, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices.

We trained Moses using the data provided by MTIL along with English-Hindi parallel corpus sourced from ICON2015 (http://ltrc.iiit.ac.in/ICON2015/) shared task, with English as the source language and Hindi as the target language. For building the language model, we used KenLM (https://kheafield.com/code/kenlm/) [10] with 7-g from the target corpus. The Hindi monolingual corpus from MTIL was used to build the language model along with additional Hindi monolingual corpus from the ICON2015 shared task data.

Training the Moses SMT system resulted in the generation of phrase model and translation model, which helps in translating between source-target language pairs. Moses scores the phrase in the phrase table with respect to a given source sentence and produces the best-scored phrases as output.

3.2 RNNs

An RNN is a type of neural network that has a hidden state h and an output y with a variable length sequence $x=(x_1, \ldots, x_T)$, as input. At each time stamp t, the hidden state $h(t)$ is calculated by

$$h_{(t)} = f(h_{(t-1)}, x_t) \tag{1}$$

where f is a nonlinear activation function. A probability distribution over a sequence can be learned by an RNN by being trained to predict the next symbol. In that case, the output at each time stamp t is the conditional distribution:

$$p(x_t \mid x_{t-1}, \ldots, x_1) \tag{2}$$

By combining these probabilities, we can compute the probability of the sequence x using

$$p(x) = \prod_{t=1} T * p(x_t \mid x_{t-1}, \ldots, x_1) \tag{3}$$

From this learned distribution, it is easy to guess a new sequence by sampling a symbol at each time step.

3.3 RNN Encoder-Decoder

In their paper [4], Cho et al. proposed a novel neural network architecture that learns to encode a variable-length sequence into a fixed-length vector representation and to decode a given fixed-length vector representation back into a variable-length sequence as part of LISA lab project. This is shown in Figure 1.

Our proposed system makes use of the same architecture that helps us to compare the scores of the phrases of the phrase table derived from the Moses SMT system. The encoder is an RNN that reads each symbol of an input sequence x sequentially. While it reads each symbol, the hidden state of the RNN changes according to Eq. (1). After reading the end of the sequence, the hidden state of the RNN is a summary c of the whole input sequence. The decoder of the proposed model is another RNN that is trained to generate the output sequence by predicting the next symbol y_t given the hidden state $h(t)$. The hidden state of the decoder at time t is computed by

$$h_{(t)} = f(h_{(t-1)}, y_{t-1}, c) \tag{4}$$

and similarly, the conditional distribution of the next symbol is

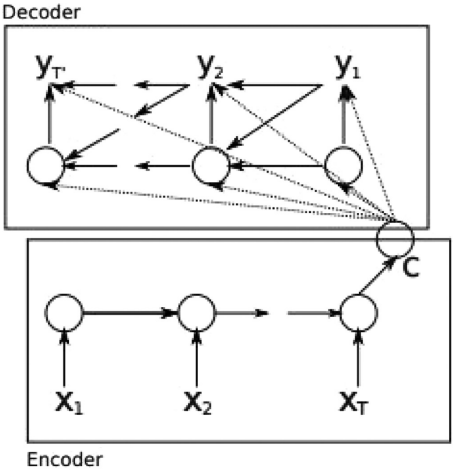

Figure 1: RNN Encoder-Decoder Architecture.

$$P(y_t \mid y_{t-1}, y_{t-2}, \ldots, y_1, c) = g(h_{(t)}, y_{t-1}, c) \tag{5}$$

for given activation functions f and g (the latter must produce valid probabilities, e.g. with a softmax).

The two components of the proposed RNN encoder-decoder are jointly trained to maximize the conditional log-likelihood.

$$\max \frac{1}{N} \sum_{n-1}^{N} \log p\Theta(y_n \mid x_n) \tag{6}$$

where Θ is the set of the model parameters and each (x_n, y_n) is an (input sequence, output sequence) pair from the training set. Once the RNN encoder-decoder from the authors was trained, it can be used to score a given pair of input and output sequences, where the score is simply a probability $p\Theta(y \mid x)$ from Eqs. (2) and (3). Once the RNN encoder-decoder is trained, we add a new score for each phrase pair to the existing phrase table. This allows the new scores to enter into the existing tuning algorithm with minimal additional overhead in computation. The top scoring phrases for the translation model and RNN encoder-decoder for the same source sentence is shown in Table 2.

4 Experimental Setup

4.1 Data

The MTIL2017 shared task was organized for participants to develop MT systems under a constrained environment. For this, parallel corpus for four languages with respect to English were developed [1, 22]. The details of the data developed were as follows:

1. English-Tamil: 138k pairs [46k from Amrita + 92k from TDIL (http://tdil.mit.gov.in/)].
2. English-Malayalam: 103k pairs (40k from Amrita + 63k from TDIL).
3. English-Hindi: 162k pairs (60k from Amrita + 102k from TDIL).
4. English-Punjabi: 130k pairs (53k from Amrita + 77k from TDIL).

Our paper made use of the English-Hindi parallel corpus and developed an MT system by incorporating SMT as well NMT.

4.2 Moses Setup

To prepare the data for training the SMT system, we performed the following steps:

1. Tokenization: Spaces are inserted between (e.g.) words and punctuation.
2. Truecasing: The initial words in each sentence are converted to their most probable casing. This helps reduce data sparsity.
3. Cleaning: Long sentences and empty sentences are removed as they can cause problems with the training pipeline and misaligned sentences are removed.

The language model was built with the target language, Hindi in our case, to ensure fluent output. KenLM, which comes bundled with the Moses toolkit, was used for building this model. As the organizers gave us a free hand in using our own language model, we added some more Hindi sentences (from ICON shared task) to the Hindi corpus provided by MTIL2017. To train the translation model, GIZA++ (http://www.statmt.org/moses/giza/GIZA++.html) was used for word alignment. Finally, the phrases were extracted and scored. This phrase table was used later to translate test sentences that were compared to the results of the RNN encoder-decoder as discussed in Section 5.

Table 1: Optimization of Hyperparameters.

Optimizer	Batch size	Error rate	No. of epochs	No. of translations
Adam	256	0.001	500	72
Adam	512	0.001	500	67
RMS Prop	256	0.001	500	77
RMS Prop	512	0.001	500	68

4.3 RNN Encoder-Decoder Setup

The following steps were followed to build the NMT system with the proposed RNN encoder-decoder architecture:

1. Building Vocabulary: Vocabulary for both the source and target corpus was prepared.
2. Corpus Shuffle: The vocabulary corpus of both the source and target languages were shuffled.
3. Dictionary Creation: Finally, a dictionary was created by giving in an alignment file to replace the unknown (UNK) words. The Anymalign (https://anymalign.limsi.fr/) [14] tool was used to generate the alignment file.

The above files were fed to the RNN encoder for training of the NMT system. Epoch was kept as 500. RNN cells were used with the RMS Prop as optimizer and 0.001 and 256 as error rate and batch size, respectively. The validation split of 0.2 was used for the training. The validation accuracy was recorded as 25.0258%. The hyperparameters were optimized as shown in Table 1. After the training, the RNN decoder was used to translate given sentences.

5 Experimental Results

Cho et al. [4] tried to analyze the improvement in the performance of the RNN encoder-decoder over the SMT system by analyzing the scores of the phrase pairs. We did the same by selecting those phrase pairs that were scored higher by the RNN encoder-decoder compared to the SMT system with respect to a given source sentence. In most cases, the scoring of the phrases by the RNN encoder-decoder was similar to the scoring of the phrases by the phrase table of the SMT system as the quality of the translation is approximately the same. This is shown in Table 2.

As a result, we got 77 translations (with scores higher) out of 562 source language sentences in the test data, as most of the phrases for both SMT and RNN encoder-decoder system have more or less the same score.

6 Additional Experiments

Additionally, some other experiments were done using the NPLM (https://github.com/moses-smt/nplm?files=1) [26] language model coupled with Moses to analyze the improvement in performance if any.

Table 2: Top Scoring Phrases for the Translation Model and Encoder-Decoder.

Source	The belly can be reduced by regular exercise
Phrase table	(niyamit rup se kam sakta hai)[Hindi]
RNN encoder-decoder	(niyamit vyayam se kam sakta hai)[Hindi]
Source	Such incidents are happening in civilized societies
Phrase table	(aisi stithi me hoti hai)[Hindi]
RNN encoder-decoder	(sabhya me aisi ghatnaye rahi hai)[Hindi]

The Hindi text has been denoted in Roman script.

Table 3: Result of Evaluation Indicating Adequacy, Fluency, Rating, and BLEU Score when Evaluated with script provided by the Organizers.

Adequacy				
Eval 1	**Eval 2**	**Eval 3**	**Avg**	**Adequacy and fluency in percentage**
1.966	1.503	1.967	1.81	35.28
Fluency				
Eval 1	**Eval 2**	**Eval 3**	**Avg**	**Rating in percentage**
1.802	1.537	1.805	1.812	31.5
Rating				
Eval 1	**Eval 2**	**Eval 3**	**Avg**	**BLEU score**
1.611	1.508	1.604	1.574	3.57

NPLM is a module that generates language models for target language with the help of neural networks in contrast to KenLM that uses statistical methods to generate the same.

The results of the above experiments are documented below.

1. Using KenLM with Moses and vanilla RNN encoder-decoder resulted in generating 77 translations that had phrase scores higher than SMT.
2. Using NPLM with Moses and vanilla RNN encoder-decoder resulted in generating 64 translations that had phrase scores higher than SMT.
3. The use of NPLM results in the better performance of the SMT system; hence, we could extract less phrase pairs that were scored higher compared to the SMT system.

7 Conclusion

In the current work, we tried to implement the RNN encoder-decoder architecture and SMT for comparing the quality of translation by scoring the phrase table generated by SMT, with an RNN encoder-decoder system. Sentences with the highest scores are selected as output. The proposed system, when evaluated with script provided by organizers, yields an percentage score of 35.28% in adequacy and fluency and 31.5% in rating [1]. The BLEU score came in as 3.57. The detailed evaluation is detailed in Table 3. We noticed that SMT works well for long sentences and NMT works well for short sentences. As a future scope, we will try to improve this limitation of RNN cells using LSTM cells with the Attention model. Also, we will have to deal with the unknown words problem that is not dealt by the recurrent encoder-decoder by training it with more bilingual corpus.

Acknowledgments: Sainik Kumar Mahata and Dipankar Das are supported by Media Lab Asia, MeitY, Government of India, under the Visvesvaraya Ph.D. Scheme for Electronics & IT.

Bibliography

[1] M. Anand Kumar, B. Premjith, S. Shivkaran, B. Kavirajan, S. Rajendran and K. P. Soman, Overview of the shared task on machine translation in Indian languages (MTIL-2017), *J. Intell. Syst.* (2017).

[2] M. Auli, M. Galley, C. Quirk and G. Zweig, Joint language and translation modeling with recurrent neural networks, in: *EMNLP*, vol. 3, Association of Computational Linguistics, Seattle, WA, USA, 2013.

[3] S. Chandar AP, S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar and A. Saha, An autoencoder approach to learning bilingual word representations, in: *Advances in Neural Information Processing Systems*, pp. 1853–1861, MIT Press, Cambridge, MA, USA, 2014.

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

[5] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. M. Schwartz and J. Makhoul, Fast and robust neural network joint models for statistical machine translation, in: *ACL (1)*, pp. 1370–1380, Association of Computational Linguistics, Baltimore, MD, USA, 2014.

[6] S. Doherty, S. O'Brien and M. Carl, Eye tracking as an MT evaluation technique, *Mach. Transl.* **24** (2010), 1–13.
[7] B. J. Dorr, P. W. Jordan and J. W. Benoit, A survey of current paradigms in machine translation, *Adv. Comput.* **49** (1999), 1–68.
[8] J. Gao, X. He, W.-T. Yih and L. Deng, Learning semantic representations for the phrase translation model, arXiv preprint arXiv:1312.0482 (2013).
[9] W. He, Z. He, H. Wu and H. Wang, Improved neural machine translation with SMT features, in: *AAAI*, pp. 151–157, AAAI Press, Phoenix, 2016.
[10] K. Heafield, KenLM: faster and smaller language model queries, in: *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pp. 187–197, Edinburgh, Scotland, United Kingdom, July 2011.
[11] A. Joshi, A. R. Balamurali and P. Bhattacharyya, A fall-back strategy for sentiment analysis in Hindi: a case study, in: *Proceedings of the 8th ICON*, Macmillan Publishers, IIT Kharagpur, India, 2010.
[12] N. Kalchbrenner and P. Blunsom, Recurrent continuous translation models, in: *EMNLP*, vol. 3, p. 413, Association of Computational Linguistics, Seattle, WA, USA, 2013.
[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, E. Herbst, Moses: open source toolkit for statistical machine translation, in: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 177–180, Association of Computational Linguistics, Prague, 2007.
[14] A. Lardilleux and Y. Lepage, Sampling-based multilingual alignment, in: *Recent Advances in Natural Language Processing*, pp. 214–218, Borovets, Bulgaria, 2009.
[15] S. Liu, N. Yang, M. Li and M. Zhou, A recursive recurrent neural network for statistical machine translation, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 1491–1500, Association for Computational Linguistics, Baltimore, MD, USA, 2014.
[16] M.-T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025 (2015).
[17] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals and W. Zaremba, Addressing the rare word problem in neural machine translation, arXiv preprint arXiv:1410.8206 (2014).
[18] S. Mahata, D. Das and S. Pal, WMT2016: a hybrid approach to bilingual document alignment, in: *WMT*, pp. 724–727, Association of Computational Linguistics, Berlin, Germany, 2016.
[19] S. K. Mahata, D. Das and S. Bandyopadhyay, BUCC2017: a hybrid approach for identifying parallel sentences in comparable corpora, in: *ACL 2017*, p. 56, Association of Computational Linguistics, Vancouver, Canada, 2017.
[20] F. J. Och and H. Ney, The alignment template approach to statistical machine translation, *Comput. Linguist.* **30** (2004), 417–449.
[21] S. Pal, B. G. Patra, D. Das, S. K. Naskar, S. Bandyopadhyay and J. van Genabith, How sentiment analysis can help machine translation, in: *11th International Conference on Natural Language Processing*, p. 89, Association of Computational Linguistics, Goa, India, 2014.
[22] B. Premjith, S. Sachin Kumar, R. Shyam, M. Anand Kumar and K. P. Soman, A fast and efficient framework for creating parallel corpus, *Indian J. Sci. Technol.* **9** (2016).
[23] H. Schwenk, Continuous space translation models for phrase-based statistical machine translation, in: *COLING (Posters)*, pp. 1071–1080, Association of Computational Linguistics, Montreal, Canada, 2012.
[24] R. Sennrich, B. Haddow and A. Birch, Improving neural machine translation models with monolingual data, arXiv preprint arXiv:1511.06709 (2015).
[25] R. Socher, E. H. Huang, J. Pennin, C. D. Manning and A. Y. Ng, Dynamic pooling and unfolding recursive autoencoders for paraphrase detection, in: *Advances in Neural Information Processing Systems*, pp. 801–809, Curran Associates Inc., Granada, Spain, 2011.
[26] A. Vaswani, Y. Zhao, V. Fossum and D. Chiang, Decoding with large-scale neural language models improves translation, in: *EMNLP*, pp. 1387–1392, Association of Computational Linguistics, Seattle, WA, USA, 2013.
[27] W. Y. Zou, R. Socher, D. M. Cer and C. D. Manning, Bilingual word embeddings for phrase-based machine translation, in: *EMNLP*, pp. 1393–1398, Association of Computational Linguistics, Seattle, WA, USA, 2013.