Stanford Neural Machine Translation Systems for Spoken Language Domains

Minh-Thang Luong, Christopher D. Manning

Computer Science Department, Stanford University, Stanford, CA 94305

{lmthang,manning}@stanford.edu

Abstract

Neural Machine Translation (NMT), though recently developed, has shown promising results for various language pairs. Despite that, NMT has only been applied to mostly formal texts such as those in the WMT shared tasks. This work further explores the effectiveness of NMT in spoken language domains by participating in the MT track of the IWSLT 2015. We consider two scenarios: (a) how to adapt existing NMT systems to a new domain and (b) the generalization of NMT to low-resource language pairs. Our results demonstrate that using an existing NMT framework[1], we can achieve competitive results in the aforementioned scenarios when translating from English to German and Vietnamese. Notably, we have advanced state-of-the-art results in the IWSLT English-German MT track by up to 5.2 BLEU points.

1. Introduction

Neural Machine Translation (NMT) is a radically new way of teaching machines to translate using deep neural networks. Though developed just last year [1, 2], NMT has achieved state-of-the-art results in the WMT translation tasks for various language pairs such as English-French [3], English-German [4, 5], and English-Czech [6]. NMT is appealing since it is conceptually simple. NMT is essentially a big recurrent neural network that can be trained end-to-end and translates as follows. It reads through the given source words one by one until the end, and then, starts emitting one target word at a time until a special end-of-sentence symbol is produced. We illustrate this process in Figure 1.

Such simplicity leads to several advantages. NMT requires minimal domain knowledge: it only assumes access to sequences of source and target words as training data and learns to directly map one into another. NMT beam-search decoders that generate words from left to right can be easily implemented, unlike the highly intricate decoders in standard MT [7]. Lastly, the use of recurrent neural networks allow NMT to generalize well to very long word sequences while not having to explicitly store any gigantic phrase tables or language models as in the case of standard MT.

Despite all the success, NMT has been applied to mostly formal texts as in the case of the WMT translation tasks. As such, it would be interesting to examine the effectiveness of NMT in the spoken language domain through the IWSLT MT track. This work explores two scenarios, namely NMT *adaptation* and NMT for *low-resource translation*. In the first scenario, we ask if it is useful to take an existing model trained on one domain and adapt it to another domain. Our findings show that for the English-German translation task, such adaptation is very crucial which gives us an improvement of +3.8 BLEU points over the model without adaptation. This helps us advance *state-of-the-art* results in the English-German MT track by up to 5.2 BLEU points.

For the latter scenario, we show that even with little English-Vietnamese training data, NMT models trained with an off-the-shelf framework can achieve competitive performance compared to the IWSLT baseline. It is also worthwhile to point out a related work [8] which achieved best results for the low-resource language pair Turkish-English in IWSLT. However, their work makes use of a huge monolingual corpus, the English Gigaword.

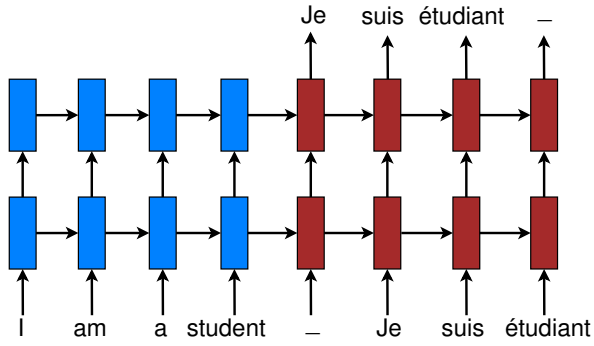

Figure 1: **Neural machine translation** – example of a deep recurrent architecture proposed in [1] for translating a source sentence *"I am a student"* into a target sentence *"Je suis étudiant"*. Here, "_" marks the end of a sentence.

2. Approach

We give background information on NMT and the attention mechanism before discussing our model choices.

2.1. Neural Machine Translation

Neural machine translation aims to directly model the conditional probability $p(y|x)$ of translating a source sentence,

[1] http://nlp.stanford.edu/projects/nmt/

$x_1, \ldots, x_n$, to a target sentence, $y_1, \ldots, y_m$. It accomplishes such goal through the *encoder-decoder* framework [1, 2]. The *encoder* computes a representation $\boldsymbol{s}$ for each source sentence. Based on that source representation, the *decoder* generates a translation, one target word at a time, and hence, decomposes the conditional probability as:

$$\log p(y|x) = \sum_{j=1}^{m} \log p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) \tag{1}$$

A natural choice to model such a decomposition in the decoder is to use a recurrent neural network (RNN) architecture, which most of the recent NMT work have in common. They, however, differ in terms of the RNN architectures used and how the encoder computes the source representation $\boldsymbol{s}$.

Kalchbrenner and Blunsom [9] used an RNN with the vanilla RNN unit for the decoder and a convolutional neural network for encoding the source. On the other hand, Sutskever et al. [1] and Luong et al. [3, 5] built deep RNNs with the Long Short-Term Memory (LSTM) unit [10] for both the encoder and the decoder. Cho et al., [2], Bahdanau et al., [11], and Jean et al. [4, 8] all adopted an LSTM-inspired hidden unit, the gated recurrent unit (GRU), and used bidirectional RNNs for the encoder.

In more details, considering the top recurrent layer in a deep RNN architecture, one can compute the probability of decoding each target word y_j as:

$$p\left(y_j|y_{<j}, x, \boldsymbol{s}\right) = \text{softmax}\left(\boldsymbol{h}_j\right) \tag{2}$$

with $\boldsymbol{h}_j$ being the current target hidden state computed as:

$$\boldsymbol{h}_j = f(\boldsymbol{h}_{j-1}, y_{j-1}, \boldsymbol{s}) \tag{3}$$

Here, f derives the current state given the previous state $\boldsymbol{h}_{j-1}$, the current input (often the previous word y_{t-1}), and optionally, the source representation $\boldsymbol{s}$. f can be a vanilla RNN unit, a GRU, or an LSTM. The early NMT approach [9, 1, 2, 3] uses the last source hidden state $\boldsymbol{s} = \bar{\boldsymbol{h}}_n$ once to initialize the decoder hidden state and sets $\boldsymbol{s} = [\,]$ in Eq. (3).

The training objective is formulated as follows:

$$J = \sum_{(x,y)\in\mathbb{D}} -\log p(y|x) \tag{4}$$

with $\mathbb{D}$ being our parallel training corpus.

2.2. Attention Mechanism

Here, we present a simplified version of the attention mechanism proposed in [11] on top of a deep RNN architecture, which is close to our actual models.

Regarding the aforementioned NMT approach, Bahdanau et al. [11] observed that the translation quality degrades as sentences become longer. This is mostly due to the fact that the model has to encode the entire source information into a single fixed-dimensional vector $\bar{\boldsymbol{h}}_n$, which is problemmatic for long variable-length sentences. While Sutskever et al. [1] addressed that problem by proposing the *source reversing* trick to improve learning, a more elegant approach would be to keep track of a memory of source hidden states and only refer to relevant ones when needed, which is basically the essence of the *attention mechanism* proposed in [11].

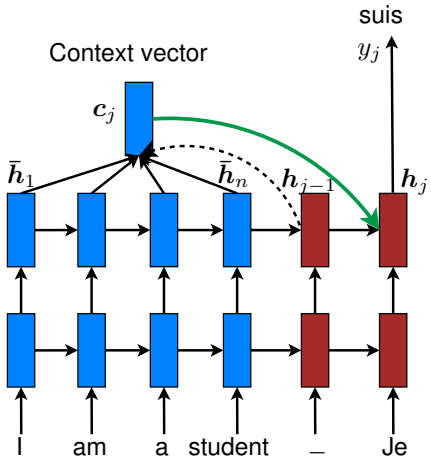

Figure 2: **Attention mechanism** – a simplified view of the attention mechanism proposed in [11]. The attention mechanism involves two steps: first, compute a *context vector* based on the previous hidden state and all the source hidden states; second, use the context vector as an additional information to derive the next hidden state.

Concretely, the attention mechanism will set $\boldsymbol{s} = [\bar{\boldsymbol{h}}_1, \ldots, \bar{\boldsymbol{h}}_n]$ in Eq. (3). The f function now consists of two stages: (a) *attention context* – the previous hidden state $\boldsymbol{h}_{j-1}$ is used to compare with individual source hidden states in $\boldsymbol{s}$ to learn an alignment vector $\boldsymbol{a}_j$; then a context vector $\boldsymbol{c}_j$ is derived as a weighted average of the source hidden states according to $\boldsymbol{a}_j$; and (b) *extended RNN* – the RNN unit is extended to take into account not just the previous hidden state $\boldsymbol{h}_{j-1}$, the current input y_{j-1}, but also the context vector $\boldsymbol{c}_j$ when computing the next hidden state $\boldsymbol{h}_j$. These stages are illustrated in Figure 2.

2.3. Our Models

We follow the attention-based NMT models proposed by Luong et al. [5], which includes two types of attention, *global* and *local*. The *global* model is similar to the one proposed in [11] with some simplifications. The *local* model is, on the other hand, a new model that has a more "focused" attention, i.e., it only puts attention on a subset of source hidden states each time, which results in better performance compared to the global attention approach. We train both types of models so that the ensembling approach as proposed in [1] can benefit from having a variety of models to make decisions.

3. NMT Adaptation

In this section, we explore the possibility of adapting existing models previously trained on one domain to a new domain.

3.1. Training Details

First, we take the existing state-of-the-art English-German system [5], which consists of 8 individual models trained on WMT data with mostly formal texts (4.5M sentence pairs). We then further train on the English-German spoken language data provided by IWSLT 2015 (200K sentence pairs). We use the default Moses tokenizer. The vocabularies are limited to the top 50K frequent words in the WMT data for each language. All other words not in the vocabularies are represented by the special token <unk>. We use the TED tst2012 as a validation dataset for early stopping and report results in BLEU [12] for TED tst2013 (during development) and tst2014, tst2015 (during evaluation).

Our models are deep LSTM networks of 4 layers with 1000-dimensional embeddings and LSTM cells. We further train existing models for 12 epochs in which after the first epoch, learning rates (initially set to 1.0) are halved every two epochs. Effective techniques are applied such as dropout [13], source reversing [1], attention mechanism [11, 5], and rare word handling [3, 4]. More details of these techniques and other hyperparameters can be found in [5]. It takes about 3-5 hours to train a model on a Tesla K40.

3.2. Results

As highlighted in Table 1, adaptation turns out to be very useful for NMT which gives an absolute gain of +3.8 BLEU points compared to using an original model without further training. Additionally, by ensembling multiple models as done in [1], we can achieve another significant gain of +2.0 BLEU points on top of the single adapted model. Compared to the best entry in IWSLT'14 [14], we have advanced the *state-of-the-art* result by +5.2 BLEU points.

System	BLEU
IWSLT'14 best entry [14]	26.2
Our systems	
Single NMT (non-adapted)	25.6
Single NMT (adapted)	29.4 (+3.8)
Ensemble NMT (adapted)	**31.4** (+2.0)

Table 1: *English-German results on TED tst2013* – BLEU scores of various systems. Progressive gains between our systems are given in parentheses.

Furthermore, according to the evaluation results provided by the organizer (Table 2), we are up to +10.0 BLEU points better than the IWSLT'15 baseline system and +4.3 BLEU point better than the best IWSLT'14 entry [14].

System	BLEU	
	tst2014	*tst2015*
IWSLT'14 best entry [14]	23.3	-
IWSLT'15 baseline	18.5	20.1
Our system	**27.6** (+9.1)	**30.1** (+10.0)

Table 2: *English-German evaluation results* – BLEU scores of various systems on the two evaluation sets. We show the differences between our submission and the IWSLT'15 baseline in parentheses.

4. NMT for Low-resource Translation

Until now, state-of-the-art NMT systems rely on large amounts of parallel corpora to sucessfully train translation models such as English-French with 12M-36M sentence pairs [3, 4] and English-German with 4.5M sentence pairs [6, 5]. There is few work examining low-resource translation direction. In [8], the authors examined translation from Turkish to English with 160K sentence pairs, but utilized large monolingual data, the English Gigaword corpus. In this work, we consider applying NMT to the low-resource translation task from English to Vietnamese in IWSLT 2015.

4.1. Training Details

We use the provided English-Vietnamese parallel data (133K sentence pairs). Apart from tokenizing the corpus with the default Moses tokenizer, no other preprocessing step, e.g., lowercasing or running word segmenter for Vietnamese, was done. We preserve casing for words and replace those whose frequencies are less than 5 by <unk>. As a result, our vocabulary sizes are 17K and 7.7K for English and Vietnamese respectively. We use the TED tst2012 as a valid set for early stopping and report BLEU scores on TED tst2013 (during development) and TED tst2015 (during evaluation).

At such a small scale of data, we could not train deep LSTMs with 4 layers as in the English-German case. Instead, we opt for 2-layer LSTM models with 500-dimensional embeddings and LSTM cells. Our other hyperparameters are: (a) we train for 12 epochs using plain SGD; (b) our learning rate is set to 1.0 initially and after 8 epochs, we start to halve the learning rate every epoch; (c) parameters are uniformly initialized in range $[0.1, 0.1]$; (d) gradients are scaled whenever their norms exceed 5; (e) source sentences are reversed which is known to help learning [1], and (f) we use dropout with probability 0.2. We train models with various attention mechanisms, global and local, as detailed in [5]. It takes about 4-7 hours to train a model on a Tesla K40.

4.2. Results

Our results during development are presented in Table 3. Similar to the trend observed in the English-German case, ensembling 9 models significantly boosts the performance by +3.6 BLEU points. Since this is the first time Vietnamese is included in IWSLT, there has not been any published number for us to compare with.

For the final evaluation, our system is, unfortunately, behind the IWSLT baseline as detailed in Table 4. Still, the gap is small and it remains interesting to see how other teams perform. Examining the translation outputs, the first author, as a

System	BLEU
Single NMT	23.3
Ensemble NMT	**26.9**

Table 3: English-Vietnamese results on TED tst2013.

native Vietnamese speaker, was quite amazed at how well the translations can be from an off-the-shelf NMT framework.

System	BLEU
IWSLT'15 baseline	**27.0**
Our system	26.4

Table 4: English-Vietnamese results on TED tst2015 provided by the organizer.

We also notice that the rare word handling technique as often done in NMT [3, 4] yields little gain for our case. We expect that this can be improved by utilizing a Vietnamese word segmenter or simple heuristics to combine collocated words such as the formula used in [15]. The rationale is that many words in English correspond to multiple-character words in Vietnamese such as "success" – "thành công" and "city" – "thành phố". The rare word handling technique requires a word dictionary built from the unsupervised alignments, and in our case, without a segmenter, we are using a word-to-char English-Vietnamese dictionary. As a result, the model will fail when trying to translate English words whose Vietname counterparts are multi-character words.

5. Conclusion

In this work, we have explored the use of Neural Machine Translation (NMT) in the spoken language domain under two interesting scenarios, namely NMT *adaptation* and NMT for *low-resource translation*. We show that NMT adaptation is very effective: models trained on a large amount of data in one domain can be finetuned on a small amount of data in another domain. This boosts the performance of an English-German NMT system by 3.8 BLEU points. This helps advance *state-of-the-art* results in the IWSLT English-German MT track by up to +5.2 BLEU points. For the latter scenario, we demonstrate that an off-the-shelf NMT framework can achieve competitive performance with very little data as in the case of the English to Vietnamese translation direction. For future work, we hope to incorporate phrase-based units in NMT to compensate for the fact that languages like Vietnamese and Chinese often need a word segmenter.

6. Acknowledgment

We gratefully acknowledge support from a gift from Bloomberg L.P. and the support of NVIDIA Corporation with the donation of Tesla K40 GPUs. We thank Thanh-Le Ha for useful discussions and the annonymous reviewers for valuable feedback.

7. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014.

[2] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[3] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *ACL*, 2015.

[4] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *ACL*, 2015.

[5] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.

[6] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, "Montreal neural machine translation systems for WMT'15," in *WMT*, 2015.

[7] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL*, 2003.

[8] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015.

[9] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *EMNLP*, 2013.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[12] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[13] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[14] M. Freitag, J. Wuebker, S. Peitz, H. Ney, M. Huck, A. Birch, N. Durrani, P. Koehn, M. Mediani, I. Slawik, J. Niehues, E. Cho, A. Waibel, N. Bertoldi, M. Cettolo, and M. Federico, "Combined spoken language translation," in *IWSLT*, 2014.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.