Multi-Source Neural Translation

Barret Zoph and Kevin Knight
Information Sciences Institute
Department of Computer Science
University of Southern California
{zoph,knight}@isi.edu

Abstract

We build a multi-source machine translation model and train it to maximize the probability of a target English string given French and German sources. Using the neural encoder-decoder framework, we explore several combination methods and report up to +4.8 Bleu increases on top of a very strong attention-based neural translation model.

1 Introduction

Kay (2000) points out that if a document is translated once, it is likely to be translated again and again into other languages. This gives rise to an interesting idea: a human does the first translation by hand, then turns the rest over to machine translation (MT). The translation system now has two strings as input, which can reduce ambiguity via "triangulation" (Kay's term). For example, the normally ambiguous English word "bank" may be more easily translated into French in the presence of a second, German input string containing the word "Flussufer" (river bank).

Och and Ney (2001) describe such a *multi-source* MT system. They first train separate bilingual MT systems $F \rightarrow E$, $G \rightarrow E$, etc. At runtime, they separately translate input strings f and g into candidate target strings e_1 and e_2, then select the best one of the two. A typical selection factor is the product of the system scores. Schwartz (2008) revisits such factors in the context of log-linear models and Bleu score, while Max et al. (2010) re-rank $F \rightarrow E$ n-best lists using n-gram precision with respect to $G \rightarrow E$ translations. Callison-Burch (2002) exploits hypothesis selection in multi-source MT to expand available corpora, via co-training.

Others use system combination techniques to merge hypotheses at the word level, creating the ability to synthesize new translations outside those proposed by the single-source translators. These methods include confusion networks (Matusov et al., 2006; Schroeder et al., 2009), source-side string combination (Schroeder et al., 2009), and median strings (González-Rubio and Casacuberta, 2010).

The above work all relies on base MT systems trained on *bilingual data*, using traditional methods. This follows early work in sentence alignment (Gale and Church, 1993) and word alignment (Simard, 1999), which exploited trilingual text, but did not build trilingual models. Previous authors possibly considered a three-dimensional translation table t($e|f, g$) to be prohibitive.

In this paper, by contrast, we train a P($e|f, g$) model directly on *trilingual data*, and we use that model to decode an (f, g) pair simultaneously. We view this as a kind of multi-tape transduction (Elgot and Mezei, 1965; Kaplan and Kay, 1994; Deri and Knight, 2015) with two input tapes and one output tape. Our contributions are as follows:

- We train a P($e|f, g$) model directly on trilingual data, and we use it to decode a new source string pair (f, g) into target string e.
- We show positive Bleu improvements over strong single-source baselines.
- We show that improvements are best when the two source languages are more distant from each other.

We are able to achieve these results using

Proceedings of NAACL-HLT 2016, pages 30–34,
San Diego, California, June 12-17, 2016.

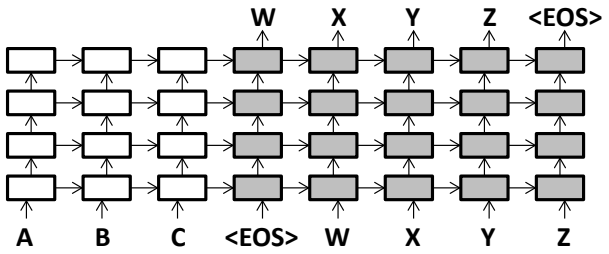

Figure 1: The encoder-decoder framework for neural machine translation (NMT) (Sutskever et al., 2014). Here, a source sentence C B A (presented in reverse order as A B C) is translated into a target sentence W X Y Z. At each step, an evolving real-valued vector summarizes the state of the encoder (white) and decoder (gray).

the framework of neural encoder-decoder models, where multi-target MT (Dong et al., 2015) and multi-source, cross-modal mappings have been explored (Luong et al., 2015a).

2 Multi-Source Neural MT

In the neural encoder-decoder framework for MT (Neco and Forcada, 1997; Castaño and Casacuberta, 1997; Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015b), we use a recurrent neural network (*encoder*) to convert a source sentence into a dense, fixed-length vector. We then use another recurrent network (*decoder*) to convert that vector in a target sentence.[1]

In this paper, we use a four-layer encoder-decoder system (Figure 1) with long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) trained for maximum likelihood (via a softmax layer) with back-propagation through time (Werbos, 1990). For our baseline single-source MT system we use two different models, one of which implements the local attention plus feed-input model from Luong et al. (2015b).

Figure 2 shows our approach to multi-source MT. Each source language has its own encoder. The question is how to combine the hidden states and cell states from each encoder, to pass on to the decoder. Black *combiner* blocks implement a function whose input is two hidden states (h_1 and h_2) and two cell states (c_1 and c_2), and whose output is a single hidden state h and cell state c. We propose two combination methods.

2.1 Basic Combination Method

The Basic method works by concatenating the two hidden states from the source encoders, applying a linear transformation W_c (size 2000 x 1000), then sending its output through a tanh non-linearity. This operation is represented by the equation:

$$h = \tanh\Big(W_c[h_1; h_2]\Big) \tag{1}$$

W_c and all other weights in the network are learned from example string triples drawn from a trilingual training corpus.

The new cell state is simply the sum of the two cell states from the encoders.

$$c = c_1 + c_2 \tag{2}$$

We also attempted to concatenate cell states and apply a linear transformation, but training diverges due to large cell values.

2.2 Child-Sum Method

Our second combination method is inspired by the Child-Sum Tree-LSTMs of Tai et al. (2015). Here, we use an LSTM variant to combine the two hidden states and cells. The standard LSTM input, output, and new cell value are all calculated. Then cell states from each encoder get their own forget gates. The final cell state and hidden state are calculated as in a normal LSTM. More precisely:

$$i = \text{sigmoid}\Big(W_1^i h_1 + W_2^i h_2\Big) \tag{3}$$

$$f = \text{sigmoid}\Big(W_i^f h_i\Big) \tag{4}$$

$$o = \text{sigmoid}\Big(W_1^o h_1 + W_2^o h_2\Big) \tag{5}$$

$$u = \tanh\Big(W_1^u h_1 + W_2^u h_2\Big) \tag{6}$$

$$c = i_f \odot u_f + f_1 \odot c_1 + f_2 \odot c_2 \tag{7}$$

$$h = o_f \odot \tanh(c_f) \tag{8}$$

This method employs eight new matrices (the W's in the above equations), each of size 1000 x 1000. The $\odot$ symbol represents an elementwise multiplication. In equation 3, i represents the input gate of a typical LSTM cell. In equation 4,

[1] We follow previous authors in presenting the source sentence to the encoder in reverse order.

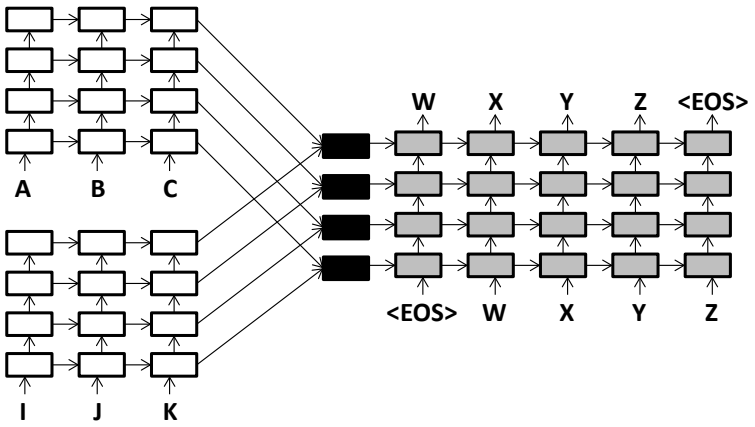

Figure 2: Multi-source encoder-decoder model for MT. We have two source sentences (C B A and K J I) in different languages. Each language has its own encoder; it passes its final hidden and cell state to a set of *combiners* (in black). The output of a combiner is a hidden state and cell state of the same dimension.

there are two forget gates indexed by the subscript i that serve as the forget gates for each of the incoming cells for each of the encoders. In equation 5, o represents the output gate of a normal LSTM. i, f, o, and u are all size-1000 vectors.

2.3 Multi-Source Attention

Our single-source attention model is modeled off the local-p attention model with feed input from Luong et al. (2015b), where hidden states from the top decoder layer can look back at the top hidden states from the encoder. The top decoder hidden state is combined with a weighted sum of the encoder hidden states, to make a better hidden state vector ($\tilde{h_t}$), which is passed to the softmax output layer. With input-feeding, the hidden state from the attention model is sent down to the bottom decoder layer at the next time step.

The local-p attention model from Luong et al. (2015b) works as follows. First, a position to look at in the source encoder is predicted by equation 9:

$$p_t = S \cdot \text{sigmoid}(v_p^T \tanh(W_p h_t)) \tag{9}$$

S is the source sentence length, and v_p and W_p are learned parameters, with v_p being a vector of dimension 1000, and W_p being a matrix of dimension 1000 x 1000.

After p_t is computed, a window of size $2D + 1$ is looked at in the top layer of the source encoder centered around p_t ($D = 10$). For each hidden state in this window, we compute an alignment score $a_t(s)$, between 0 and 1. This alignment score is computed by equations 10, 11 and 12:

$$a_t(s) = \text{align}(h_t, h_s)\text{exp}\Big(\frac{-(s - p_t)^2}{2\sigma^2}\Big) \tag{10}$$

$$\text{align}(h_t, h_s) = \frac{\exp(\text{score}(h_t, h_s))}{\sum_{s'} \exp(\text{score}(h_t, h_{s'}))} \tag{11}$$

$$\text{score}(h_t, h_s) = h_t^T W_a h_s \tag{12}$$

In equation 10, σ is set to be $D/2$ and s is the source index for that hidden state. W_a is a learnable parameter of dimension 1000 x 1000.

Once all of the alignments are calculated, c_t is created by taking a weighted sum of all source hidden states multiplied by their alignment weight.

The final hidden state sent to the softmax layer is given by:

$$\tilde{h_t} = tanh\Big(W_c[h_t; c_t]\Big) \tag{13}$$

We modify this attention model to look at both source encoders simultaneously. We create a context vector from each source encoder named c_t^1 and c_t^2 instead of the just c_t in the single-source attention model:

$$\tilde{h_t} = tanh\Big(W_c[h_t; c_t^1; c_t^2]\Big) \tag{14}$$

In our multi-source attention model we now have two p_t variables, one for each source encoder. We

	French	English	German
Word tokens	66.2m	59.4m	57.0m
Word types	424,832	381,062	865,806
Segment pairs	2,378,112		
Ave. segment length (tokens)	27.8	25.0	24.0

Figure 3: Trilingual corpus statistics.

also have two separate sets of alignments and therefore now have two c_t values denoted by c_t^1 and c_t^2 as mentioned above. We also have distinct W_a, v_p, and W_p parameters for each encoder.

3 Experiments

We use English, French, and German data from a subset of the WMT 2014 dataset (Bojar et al., 2014). Figure 3 shows statistics for our training set. For development, we use the 3000 sentences supplied by WMT. For testing, we use a 1503-line trilingual subset of the WMT test set.

For the single-source models, we follow the training procedure used in Luong et al. (2015b), but with 15 epochs and halving the learning rate every full epoch after the 10th epoch. We also re-scale the normalized gradient when norm > 5. For training, we use a minibatch size of 128, a hidden state size of 1000, and dropout as in Zaremba et al. (2014). The dropout rate is 0.2, the initial parameter range is [-0.1, +0.1], and the learning rate is 1.0. For the normal and multi-source attention models, we adjust these parameters to 0.3, [-0.08, +0.08], and 0.7, respectively, to adjust for overfitting.

Figure 4 shows our results for target English, with source languages French and German. We see that the Basic combination method yields a +4.8 Bleu improvement over the strongest single-source, attention-based system. It also improves Bleu by +2.2 over the non-attention baseline. The Child-Sum method gives improvements of +4.4 and +1.4. We confirm that two copies of the same French input yields no BLEU improvement. Figure 5 shows the action of the multi-attention model during decoding.

When our source languages are English and French (Figure 6), we observe smaller BLEU gains (up to +1.1). This is evidence that the more distinct the source languages, the better they disambiguate each other.

Target = English			
Source	**Method**	**Ppl**	**BLEU**
French	—	10.3	21.0
German	—	15.9	17.3
French+German	Basic	8.7	23.2
French+German	Child-Sum	9.0	22.5
French+French	Child-Sum	10.9	20.7
French	Attention	8.1	25.2
French+German	B-Attent.	5.7	30.0
French+German	CS-Attent.	6.0	29.6

Figure 4: Multi-source MT for target English, with source languages French and German. Ppl reports test-set perplexity as the system predicts English tokens. BLEU is scored using the multi-bleu.perl script from Moses. For our evaluation we use a single reference and they are case sensitive.

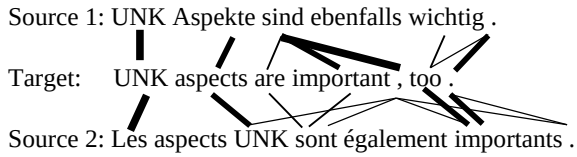

Figure 5: Action of the multi-attention model as the neural decoder generates target English from French/German sources (test set). Lines show strengths of $a_t(s)$.

4 Conclusion

We describe a multi-source neural MT system that gets up to +4.8 Bleu gains over a very strong attention-based, single-source baseline. We obtain this result through a novel encoder-vector combination method and a novel multi-attention system. We release the code for these experiments at www.github.com/isi-nlp/Zoph_RNN.

Target = German			
Source	**Method**	**Ppl**	**BLEU**
French	—	12.3	10.6
English	—	9.6	13.4
French+English	Basic	9.1	14.5
French+English	Child-Sum	9.5	14.4
English	Attention	7.3	17.6
French+English	B-Attent.	6.9	18.6
French+English	CS-Attent.	7.1	18.2

Figure 6: Multi-source MT results for target German, with source languages French and English.

5 Acknowledgments

This work was carried out with funding from DARPA (HR0011-15-C-0115) and ARL/ARO (W911NF-10-1-0533).

References

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*.

O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, and L. Specia, editors. 2014. *Proc. of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics.

C. Callison-Burch. 2002. Co-training for statistical machine translation. Master's thesis, School of Informatics, University of Edinburgh.

M. A. Castaño and F. Casacuberta. 1997. A connectionist approach to machine translation. In *EUROSPEECH*.

A. Deri and K. Knight. 2015. How to make a Frenemy: Multitape FSTs for portmanteau generation. In *Proc. NAACL*.

D. Dong, H. Wu, W. he, D. Yu, and H. Wang. 2015. Multi-task learning for multiple language translation. In *Proc. ACL*.

C. Elgot and J. Mezei. 1965. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68.

W. A Gale and K. W Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

J. González-Rubio and F. Casacuberta. 2010. On the use of median string for multi-source translation. In *Proc. ICPR*.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8).

R. Kaplan and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

M. Kay. 2000. Triangulation in translation. Keynote at MT 2000 Conference, University of Exeter.

M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. 2015a. Multi-task sequence to sequence learning. In *arXiv*. http://arxiv.org/abs/1511.06114.

M. Luong, H. Pham, and C. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*.

E. Matusov, N. Ueffing, and H. Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. EACL*.

A. Max, J. Crego, and F. Yvon. 2010. Contrastive lexical evaluation of machine translation. In *Proc. LREC*.

R. Neco and M. Forcada. 1997. Asynchronous translations with recurrent neural nets. In *International Conf. on Neural Networks*, volume 4, pages 2535–2540.

F. J. Och and H. Ney. 2001. Statistical multi-source translation. In *Proc. MT Summit*.

J. Schroeder, T. Cohn, and P. Koehn. 2009. Word lattices for multi-source translation. In *Proc. EACL*.

L. Schwartz. 2008. Multi-source translation methods. *Proc. AMTA*.

M. Simard. 1999. Text-translation alignment: Three languages are better than two. In *Proc. EMNLP/VLC*.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*.

K. S. Tai, R. Socher, and C. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. ACL*.

P. J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE*, 78(10):1550–1560.

W. Zaremba, I. Sutskever, and O. Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.