Supervised Attentions for Neural Machine Translation

Haitao Mi   **Zhiguo Wang**   **Abe Ittycheriah**
T.J. Watson Research Center
IBM
1101 Kitchawan Rd, Yorktown Heights, NY 10598
{hmi, zhigwang, abei}@us.ibm.com

Abstract

In this paper, we improve the attention or alignment accuracy of neural machine translation by utilizing the alignments of training sentence pairs. We simply compute the distance between the machine attentions and the "true" alignments, and minimize this cost in the training procedure. Our experiments on large-scale Chinese-to-English task show that our model improves both translation and alignment qualities significantly over the large-vocabulary neural machine translation system, and even beats a state-of-the-art traditional syntax-based system.

1 Introduction

Neural machine translation (NMT) has gained popularity in recent two years (e.g. (Bahdanau et al., 2014; Jean et al., 2015; Luong et al., 2015; Mi et al., 2016b; Li et al., 2016), especially for the attention-based models of Bahdanau et al. (2014).

The attention model plays a crucial role in NMT, as it shows which source word(s) the model should focus on in order to predict the next target word. However, the attention or alignment quality of NMT is still very low (Mi et al., 2016a; Tu et al., 2016).

In this paper, we alleviate the above issue by utilizing the *alignments* (human annotated data or machine alignments) of the training set. Given the alignments of all the training sentence pairs, we add an alignment distance cost to the objective function. Thus, we not only maximize the log translation probabilities, but also minimize the alignment distance cost. Large-scale experiments over Chinese-to-English on various test sets show that our best method for a single system improves the translation quality significantly over the large vocabulary NMT system (Section 5) and beats the state-of-the-art syntax-based system.

2 Neural Machine Translation

As shown in Figure 1, attention-based NMT (Bahdanau et al., 2014) is an encoder-decoder network. the encoder employs a bi-directional recurrent neural network to encode the source sentence $\mathbf{x} = (x_1, ..., x_l)$, where l is the sentence length (including the end-of-sentence $\langle \text{eos} \rangle$), into a sequence of hidden states $\mathbf{h} = (h_1, ..., h_l)$, each h_i is a concatenation of a left-to-right $\overrightarrow{h_i}$ and a right-to-left $\overleftarrow{h_i}$.

Given $\mathbf{h}$, the decoder predicts the target translation by maximizing the conditional log-probability of the correct translation $\mathbf{y}^* = (y_1^*, ...y_m^*)$, where m is the sentence length (including the end-of-sentence). At each time t, the probability of each word y_t from a target vocabulary V_y is:

$$p(y_t|\mathbf{h}, y_{t-1}^*..y_1^*) = g(s_t, y_{t-1}^*), \tag{1}$$

where g is a two layer feed-forward neural network over the embedding of the previous word y_{t-1}^*, and the hidden state s_t. The s_t is computed as:

$$s_t = q(s_{t-1}, y_{t-1}^*, H_t) \tag{2}$$

$$H_t = \begin{bmatrix} \sum_{i=1}^{l} (\alpha_{t,i} \cdot \overleftarrow{h}_i) \\ \sum_{i=1}^{l} (\alpha_{t,i} \cdot \overrightarrow{h}_i) \end{bmatrix}, \tag{3}$$

where q is a gated recurrent units, H_t is a weighted sum of $\mathbf{h}$; the weights, α, are computed with a two layer feed-forward neural network r:

$$\alpha_{t,i} = \frac{\exp\{r(s_{t-1}, h_i, y_{t-1}^*)\}}{\sum_{k=1}^{l} \exp\{r(s_{t-1}, h_k, y_{t-1}^*)\}} \tag{4}$$

We put all $\alpha_{t,i}$ ($t = 1...m$, $i = 1...l$) into a matrix $\mathcal{A}'$, we have a matrix (alignment) like (c) in Figure 2, where each row (for each target word) is a probability distribution over the source sentence $\mathbf{x}$.

The training objective is to maximize the conditional log-probability of the correct translation y^*

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2283–2288,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

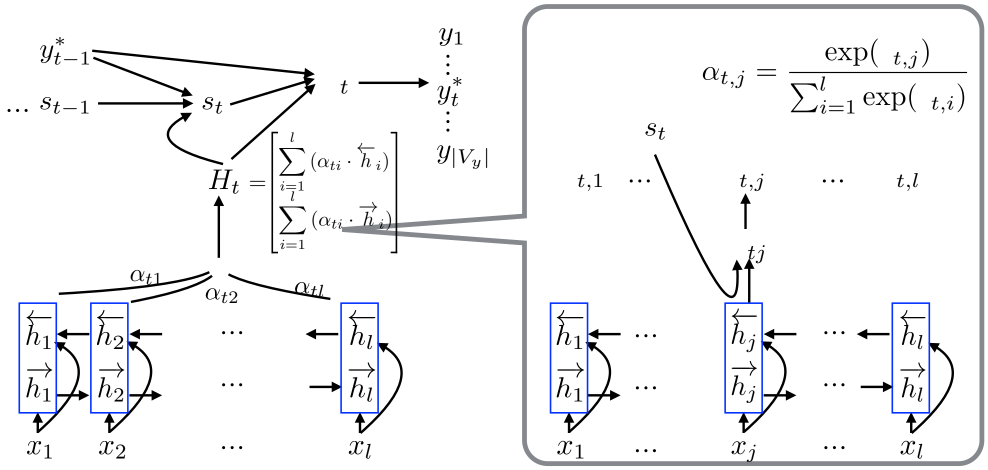

y^*_{t-1}
$\dots\ s_{t-1}$
s_t
t
y_1
y^*_t
$y_{|V_y|}$
$H_t = \begin{bmatrix} \sum_{i=1}^{l} (\alpha_{ti} \cdot \overleftarrow{h}_i) \\ \sum_{i=1}^{l} (\alpha_{ti} \cdot \overrightarrow{h}_i) \end{bmatrix}$
α_{t1}
α_{t2}
α_{tl}
$\overleftarrow{h}_1$ $\overrightarrow{h}_1$ $\overleftarrow{h}_2$ $\overrightarrow{h}_2$ $\overleftarrow{h}_l$ $\overrightarrow{h}_l$
x_1 x_2 $\dots$ x_l
$\alpha_{t,j} = \frac{\exp(\ _{t,j})}{\sum_{i=1}^{l} \exp(\ _{t,i})}$
s_t
$_{t,1}$ $\dots$ $_{t,j}$ $\dots$ $_{t,l}$
$_{tj}$
$\overleftarrow{h}_j$ $\overrightarrow{h}_j$
x_1 $\dots$ x_j $\dots$ x_l

Figure 1: The architecture of attention-based NMT (Bahdanau et al., 2014). The source sentence $\mathbf{x} = (x_1, ..., x_l)$ with length l, x_l is an end-of-sentence token $\langle \text{eos} \rangle$ on the source side. The reference translation is $\mathbf{y}^* = (y^*_1, ..., y^*_m)$ with length m, similarly, y^*_m is the target side $\langle \text{eos} \rangle$. $\overleftarrow{h_i}$ and $\overrightarrow{h_i}$ are bi-directional encoder states. $\alpha_{t,j}$ is the attention probability at time t, position j. H_t is the weighted sum of encoding states. s_t is a hidden state. o_t is an output state. Another one layer neural network projects o_t to the target output vocabulary, and conducts softmax to predict the probability distribution over the output vocabulary. The attention model (the right box) is a two layer feedforward neural network, $A_{t,j}$ is an intermediate state, then another layer converts it into a real number $e_{t,j}$, the final attention probability at position j is $\alpha_{t,j}$.

given x with respect to the parameters θ

$$\theta^* = \arg\max_{\theta} \sum_{n=1}^{N} \sum_{t=1}^{m} \log p(y^{*n}_t | \mathbf{x}^n, y^{*n}_{t-1}..y^{*n}_1), \tag{5}$$

where n is the n-th sentence pair $(\mathbf{x}^n, \mathbf{y}^{*n})$ in the training set, N is the total number of pairs.

3 Alignment Component

The attentions, $\alpha_{t,1}...\alpha_{t,l}$, in each step t play an important role in NMT. However, the accuracy is still far behind the traditional MaxEnt alignment model in terms of alignment F1 score (Mi et al., 2016b; Tu et al., 2016). Thus, in this section, we explicitly add an alignment distance to the objective function in Eq. 5. The "truth" alignments for each sentence pair can be from human annotated data, unsupervised or supervised alignments (e.g. GIZA++ (Och and Ney, 2000) or MaxEnt (Ittycheriah and Roukos, 2005)).

Given an alignment matrix $\mathcal{A}$ for a sentence pair $(\mathbf{x}, \mathbf{y})$ in Figure 2 (a), where we have an end-of-source-sentence token $\langle \text{eos} \rangle = x_l$, and we align all the unaligned target words (y^*_3 in this example) to $\langle \text{eos} \rangle$, also we force y^*_m (end-of-target-sentence) to be aligned to x_l with probability one. Then we conduct two transformations to get the probability distribution matrices ((b) and (c) in Figure 2).

3.1 Simple Transformation

The first transformation simply normalizes each row. Figure 2 (b) shows the result matrix $\mathcal{A}^*$. The last column in red dashed lines shows the alignments of the special end-of-sentence token $\langle \text{eos} \rangle$.

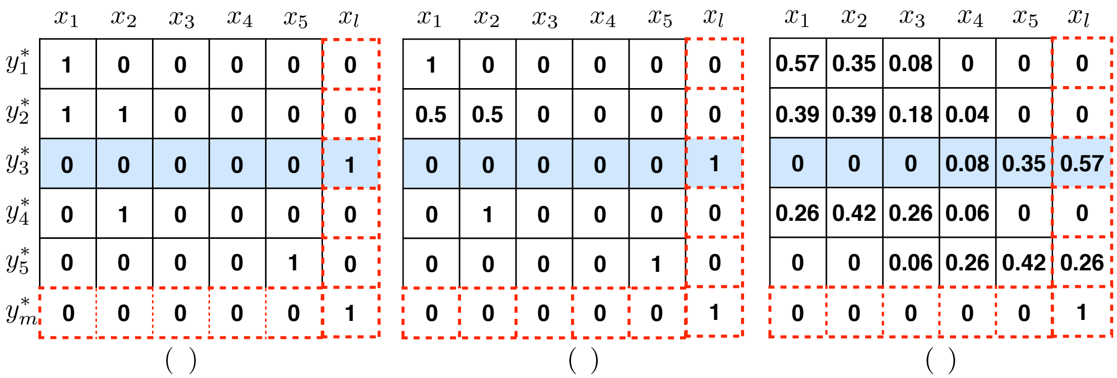
3.2 Smoothed Transformation

Given the original alignment matrix $\mathcal{A}$, we create a matrix $\mathcal{A}^*$ with all points initialized with zero. Then, for each alignment point $\mathcal{A}_{t,i} = 1$, we update $\mathcal{A}^*$ by adding a Gaussian distribution, $g(\mu, \sigma)$, with a window size w (t-w, ... t ... t+w). Take the $\mathcal{A}_{1,1} = 1$ for example, we have $\mathcal{A}^*{}_{1,1}$ += 1, $\mathcal{A}^*{}_{1,2}$ += 0.61, and $\mathcal{A}^*{}_{1,3}$ += 0.14 with w=2, $g(\mu, \sigma)$=$g(0, 1)$. Then we normalize each row and get (c). In our experiments, we use a shape distribution, where σ = 0.5.

3.3 Objectives

Alignment Objective: Given the "true" alignment $\mathcal{A}^*$, and the machine attentions $\mathcal{A}'$ produced by NMT model, we compute the Euclidean distance bewteen $\mathcal{A}^*$ and $\mathcal{A}'$.

$$d(\mathcal{A}', \mathcal{A}^*) = \sqrt{\sum_{t=1}^{m} \sum_{i=1}^{l} (\mathcal{A}'_{t,i} - \mathcal{A}^*{}_{t,i})^2}. \tag{6}$$



	x_1	x_2	x_3	x_4	x_5	x_l
y_1^*	1	0	0	0	0	0
y_2^*	1	1	0	0	0	0
y_3^*	0	0	0	0	0	1
y_4^*	0	1	0	0	0	0
y_5^*	0	0	0	0	1	0
y_m^*	0	0	0	0	0	1

( )

	x_1	x_2	x_3	x_4	x_5	x_l
y_1^*	1	0	0	0	0	0
y_2^*	0.5	0.5	0	0	0	0
y_3^*	0	0	0	0	0	1
y_4^*	0	1	0	0	0	0
y_5^*	0	0	0	0	1	0
y_m^*	0	0	0	0	0	1

( )

	x_1	x_2	x_3	x_4	x_5	x_l
y_1^*	0.57	0.35	0.08	0	0	0
y_2^*	0.39	0.39	0.18	0.04	0	0
y_3^*	0	0	0	0.08	0.35	0.57
y_4^*	0.26	0.42	0.26	0.06	0	0
y_5^*	0	0	0.06	0.26	0.42	0.26
y_m^*	0	0	0	0	0	1

( )

Figure 2: Alignment transformation. A special token, ⟨eos⟩, is introduced to the source sentence, we align all the unaligned target words (y_3^* in this case) to ⟨eos⟩. (a): the original alignment matrix $\mathcal{A}$ from GIZA++ or MaxEnt aligner. (b): simple normalization by rows (probability distribution over the source sentence $\mathbf{x}$). (c): smoothed transformation followed by normalization by rows, and typically, we always align end-of-source-sentence x_l to end-of-target-sentence y_m by probability one.

NMT Objective: We plug Eq. 6 to Eq. 5, we have

$$\theta^* = \arg\max_{\theta} \sum_{n=1}^{N} \Bigg\{ \sum_{t=1}^{m} \log p(y_t^{*n} | \mathbf{x}^n, y_{t-1}^{*n}..y_1^{*n}) - d(\mathcal{A}'^n, \mathcal{A}^{*n}) \Bigg\}. \tag{7}$$

There are two parts: translation and alignment, so we can optimize them jointly, or separately (e.g. we first optimize alignment only, then optimize translation). Thus, we divide the network in Figure 1 into alignment **A** and translation **T** parts:

- **A**: all networks before the hidden state s_t,
- **T**: the network $g(s_t, y_{t-1}^*)$.

If we only optimize **A**, we keep the parameters in **T** unchanged. We can also optimize them jointly **J**. In our experiments, we test different optimization strategies.

4 Related Work

In order to improve the attention or alignment accuracy, Cheng et al. (2016) adapted the agreement-based learning (Liang et al., 2006; Liang et al., 2008), and introduced a combined objective that takes into account both translation directions (source-to-target and target-to-source) and an agreement term between the two alignment directions. By contrast, our approach directly uses and optimizes NMT parameters using the "supervised" alignments.

5 Experiments

5.1 Data Preparation

We run our experiments on Chinese to English task. The training corpus consists of approximately 5 million sentences available within the DARPA BOLT Chinese-English task. The corpus includes a mix of newswire, broadcast news, and webblog. We do not include HK Law, HK Hansard and UN data. The Chinese text is segmented with a segmenter trained on CTB data using conditional random fields (CRF). Our development set is the concatenation of several tuning sets (GALE Dev, P1R6 Dev, and Dev 12) initially released under the DARPA GALE program. The development set is 4491 sentences in total. Our test sets are NIST MT06 (1664 sentences) , MT08 news (691 sentences), and MT08 web (666 sentences).

For all NMT systems, the full vocabulary size of the training set is $300k$. In the training procedure, we use AdaDelta (Zeiler, 2012) to update model parameters with a mini-batch size 80. Following Mi et al. (2016a), the output vocabulary for each mini-batch or sentence is a sub-set of the full vocabulary. For each source sentence, the sentence-level target vocabularies are union of top $2k$ most frequent target words and the top 10 candidates of the word-to-word/phrase translation tables learned

single system			MT06			MT08						avg.
						News			Web			
			BP	BLEU	T-B	BP	BLEU	T-B	BP	BLEU	T-B	T-B
Tree-to-string			0.95	34.93	9.45	0.94	31.12	12.90	0.90	23.45	17.72	13.36
Cov. LVNMT (Mi et al., 2016b)			0.92	35.59	10.71	0.89	30.18	15.33	0.97	27.48	16.67	14.24
+Alignment	**Zh → En**	**A → J**	0.95	35.71	10.38	0.93	30.73	14.98	0.96	27.38	16.24	13.87
		A → T	0.95	28.59	16.99	0.92	24.09	20.89	0.97	20.48	23.31	20.40
		A → T → J	0.95	35.95	10.24	0.92	30.95	14.62	0.97	26.76	17.04	13.97
		J	0.96	36.76	9.67	0.94	31.24	14.80	0.96	28.35	**15.61**	13.36
	GDFA	**J**	0.96	36.44	10.16	0.94	30.66	15.01	0.96	26.67	16.72	13.96
	MaxEnt	**J**	0.95	36.80	**9.49**	0.93	31.74	14.02	0.96	27.53	16.21	13.24
		J + Gau.	0.96	**36.95**	9.71	0.94	**32.43**	**13.61**	0.97	**28.63**	15.80	**13.04**

Table 1: Single system results in terms of (TER-BLEU)/2 (T-B, the lower the better) on 5 million Chinese to English training set. BP denotes the brevity penalty. NMT results are on a large vocabulary ($300k$) and with UNK replaced. The second column shows different alignments (**Zh** → **En** (one direction), **GDFA** ("grow-diag-final-and"), and **MaxEnt** (Ittycheriah and Roukos, 2005). **A**, **T**, and **J** mean optimize alignment only, translation only, and jointly. **Gau.** denotes the smoothed transformation.

from 'fast_align' (Dyer et al., 2013). The maximum length of a source phrase is 4. In the training time, we add the reference in order to make the translation reachable.

The Cov. LVNMT system is a re-implementation of the enhanced NMT system of Mi et al. (2016a), which employs a coverage embedding model and achieves better performance over large vocabulary NMT Jean et al. (2015). The coverage embedding dimension of each source word is 100.

Following Jean et al. (2015), we dump the alignments, attentions, for each sentence, and replace UNKs with the word-to-word translation model or the aligned source word.

Our SMT system is a hybrid syntax-based tree-to-string model (Zhao and Al-onaizan, 2008), a simplified version of the joint decoding (Liu et al., 2009; Cmejrek et al., 2013). We parse the Chinese side with Berkeley parser, and align the bilingual sentences with GIZA++ and MaxEnt. and extract Hiero and tree-to-string rules on the training set. Our two 5-gram language models are trained on the English side of the parallel corpus, and on monolingual corpora (around 10 billion words from Gigaword (LDC2011T07), respectively.As suggested by Zhang (2016), NMT systems can achieve better results with the help of those monolingual corpora. In this paper, our NMT systems only use the bilingual data. We tune our system with PRO (Hopkins and May, 2011) to minimize (TER- BLEU)/2 [1] on the development set.

5.2 Translation Results

Table 1 shows the translation results of all systems. The syntax-based statistical machine translation model achieves an average (TER-BLEU)/2 of 13.36 on three test sets. The Cov. LVNMT system achieves an average (TER-BLEU)/2 of 14.24, which is about 0.9 points worse than Tree-to-string SMT system. Please note that all systems are single systems. It is highly possible that ensemble of NMT systems with different random seeds can lead to better results over SMT.

We test three different alignments:

- **Zh** → **En** (one direction of GIZA++),
- **GDFA** (the "grow-diag-final-and" heuristic merge of both directions of GIZA++),
- **MaxEnt** (trained on $67k$ hand-aligned sentences).

[1] The metric used for optimization in this work is (TER-BLEU)/2 to prevent the system from using sentence length alone to impact BLEU or TER. Typical SMT systems use target word count as a feature and it has been observed that BLEU can be optimized by tweaking the weighting of the target word count with no improvement in human assessments of translation quality. Conversely, in order to optimize TER shorter sentences can be produced. Optimizing the combination of metrics alleviates this effect (Arne Mauser and Ney, 2008).

The alignment quality improves from **Zh** → **En** to **MaxEnt**. We also test different optimization strategies: **J** (jointly), **A** (alignment only), and **T** (translation model only). A combination, **A** → **T**, shows that we optimize **A** only first, then we fix **A** and only update **T** part. **Gau.** denotes the smoothed transformation (Section 3.2). Only the last row uses the smoothed transformation, all others use the simple transformation.

Experimental results in Table 1 show some interesting results. First, with the same alignment, **J** joint optimization works best than other optimization strategies (lines 3 to 6). Unfortunately, breaking down the network into two separate parts (**A** and **T**) and optimizing them separately do not help (lines 3 to 5). We have to conduct joint optimization **J** in order to get a comparable or better result (lines 3, 5 and 6) over the baseline system.

Second, when we change the training alignment seeds (**Zh** → **En**, **GDFA**, and **MaxEnt**) NMT model does not yield significant different results (lines 6 to 8).

Third, the smoothed transformation (**J + Gau.**) gives some improvements over the simple transformation (the last two lines), and achieves the best result (1.2 better than LVNMT, and 0.3 better than Tree-to-string). In terms of BLEU scores, we conduct the statistical significance tests with the sign-test of Collins et al. (2005), the results show that the improvements of our **J + Gau.** over LVNMT are significant on three test sets ($p < 0.01$).

At last, the brevity penalty (BP) consistently gets better after we add the alignment cost to NMT objective. Our alignment objective adjusts the translation length to be more in line with the human references accordingly.

5.3 Alignment Results

Table 2 shows the alignment F1 scores on the alignment test set (447 hand aligned sentences). The MaxEnt model is trained on $67k$ hand-aligned sentences, and achieves an F1 score of 75.96. For NMT systems, we dump the alignment matrixes and convert them into alignments with following steps. For each target word, we sort the alphas and add the max probability link if it is higher than 0.2. If we only tune the alignment component (**A** in line 3), we improve the alignment F1 score from 45.76 to 47.87. And we further boost the score to 50.97 by tuning alignment and translation jointly (**J** in line 7). Interestingly, the system using **MaxEnt** produces more alignments in the output, and results in a higher recall. This suggests that using **MaxEnt** can lead to a sharper attention distribution, as we pick the alignment links based on the probabilities of attentions, the sharper the distribution is, more links we can pick. We believe that a sharp attention distribution is a great property of NMT.

system			pre.	rec.	F1
MaxEnt			74.86	77.10	75.96
Cov LVNMT (Mi et al., 2016b)			51.11	41.42	45.76
+Alignment	**Zh** → **En**	**A**	50.88	45.19	47.87
		A → **J**	53.18	49.37	51.21
		A → **T**	50.29	44.90	47.44
		A → **T** → **J**	53.71	49.33	51.43
		J	**54.29**	48.02	50.97
	GDFA	**J**	53.88	48.25	50.91
	MaxEnt	**J**	44.42	55.25	49.25
		J + Gau.	48.90	**55.38**	**51.94**

Table 2: Alignment F1 scores of different models.

Again, the best result is **J + Gau.** in the last row, which significantly improves the F1 by 5 points over the baseline Cov. LVNMT system. When we use **MaxEnt** alignments, **J + Gau.** smoothing gives us about 1.7 points gain over **J** system. So it looks interesting to run another **J + Gau.** over **GDFA** alignment.

Together with the results in Table 1, we conclude that adding the alignment cost to the training objective helps both translation and alignment significantly.

6 Conclusion

In this paper, we utilize the "supervised" alignments, and put the alignment cost to the NMT objective function. In this way, we directly optimize the attention model in a supervised way. Experiments show significant improvements in both translation and alignment tasks over a very strong LVNMT system.

Acknowledgment

We thank the anonymous reviewers for their useful comments.

References

Sasa Hasan Arne Mauser and Hermann Ney. 2008. Automatic evaluation measures for statistical machine translation system optimization. In *Proceedings of LREC 2008*, Marrakech, Morocco, may.

D. Bahdanau, K. Cho, and Y. Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv e-prints*, September.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based joint training for bidirectional attention-based neural machine translation. In *Proceedings of IJCAI*, New York, USA, July.

Martin Cmejrek, Haitao Mi, and Bowen Zhou. 2013. Flexible and efficient hypergraph interactions for joint hierarchical and forest-to-string decoding. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 545–555, Seattle, Washington, USA, October. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*, pages 531–540, Ann Arbor, Michigan, June.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of EMNLP*.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *HLT '05: Proceedings of the HLT and EMNLP*, pages 89–96.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of ACL*, pages 1–10, Beijing, China, July.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. Towards zero unknown word in neural machine translation. In *Proceedings of IJCAI 2016*, pages 2852–2858, New York, NY, USA, July.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *North American Association for Computational Linguistics (NAACL)*, pages 104–111.

P. Liang, D. Klein, and M. I. Jordan. 2008. Agreement-based learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Yang Liu, Haitao Mi, Yang Feng, and Qun Liu. 2009. Joint decoding with multiple translation models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 576–584, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016a. A coverage embedding model for neural machine translation. *ArXiv e-prints*.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016b. Vocabulary manipulation for neural machine translation. In *Proceedings of ACL*, Berlin, Germany, August.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 440–447, Stroudsburg, PA, USA. Association for Computational Linguistics.

Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li. 2016. Coverage-based Neural Machine Translation. *ArXiv e-prints*, January.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*.

Jiajun Zhang. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of EMNLP 2016*, Austin, Texas, USA, November.

Bing Zhao and Yaser Al-onaizan. 2008. Generalizing local and non-local word-reordering patterns for syntax-based machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 572–581, Stroudsburg, PA, USA. Association for Computational Linguistics.