Learning an Interactive Attention Policy for Neural Machine Translation

Samee Ibraheem* sibraheem@berkeley.edu
Nicholas Altieri* naltieri@berkeley.edu
John DeNero denero@berkeley.edu
**Equal contribution.*

Abstract

Interactive machine translation research has focused primarily on predictive typing, which requires a human to type parts of the translation. This paper explores an interactive setting in which humans guide the attention of a neural machine translation system in a manner that requires no text entry at all. The system generates a translation from left to right, but waits periodically for a human to select the word in the source sentence to be translated next. A central technical challenge is that the system must learn when and how often to request guidance from the human. These decisions allow the system to trade off translation speed and accuracy. We cast these decisions as a reinforcement learning task and develop a policy gradient approach to train the system. Critically, the system can be trained on parallel data alone by simulating human guidance at training time. Our experiments demonstrate the viability of this interactive setting to improve translation quality and show that an effective policy for periodically requesting human guidance can be learned automatically.

1 Introduction

Despite rapid advances in neural machine translation, human input is still needed to meet the translation quality requirements of many applications. Interactive machine translation seeks to combine the quality of human translation with the speed and lexical coverage of machine translation. This paper explores an interactive setting in which the human translator does not type at all, but instead guides the attention of a neural machine translation system by selecting relevant source words as the system translates. While we should not expect that the resulting translations will be as accurate as those produced by predictive typing, this interactive approach could provide fast and accurate draft translations that could later be improved by post-editing. Moreover, source word selection enables new user interface options because it can be performed using a wide variety of input devices, including a mouse, a touch screen, or an eye tracker, which may be used in tandem with traditional text entry methods.

We first address the question of whether guiding the attention of a neural machine translation system can provide enough useful information to improve translation quality. Rather than experimenting directly with human subjects, we compute an experimental upper bound on the accuracy gains from guided attention. For each word that the system is meant to generate, we find an oracle attention that maximizes the probability of generating that word. We find that guiding attention toward this oracle provides a great deal of information to the translation system, yielding substantial gains in translation quality.

Second, we define an interactive translation process in which the system generates a translation left-to-right, but pauses on occasion to request guidance from a human collaborator. Ideally, the system would not pause after every word; if the system can generate some portion of the translation accurately without human intervention, then it would be wasteful for it to solicit human input. Therefore, an ideal system must learn to trade off between translating accurately and requiring as little human input as possible.

However, it is difficult to predict the long-term consequences of choosing whether or not to pause at any given position. The value of receiving human guidance is not only that it may improve the prediction of the next word, but that it may improve predictions of all subsequent words. Therefore, pausing early for human input might allow the system to require less guidance in later parts of a sentence. Our primary technical contribution is to cast the sequence of decisions about when to request human guidance as a reinforcement learning problem that properly accounts for the system's uncertainty about all the downstream effects of requesting human intervention. We apply a policy gradient method to this problem and show that the system is able to learn an effective interaction policy. This policy estimates when, during the process of translation, human guidance is likely to provide enough long-term benefit to justify the cost of pausing.

We evaluate our approach using an English-German neural machine translation system trained for the WMT 2016 news translation task. We show that the whole system, including the learned interaction policy, can be trained fully automatically by approximating human input using simulated guidance.

2 Related Work

Interactive machine translation involves human translators working collaboratively with a machine translation system to produce high quality output efficiently (Foster and Lapalme, 2002). Several interactive interfaces to machine translation systems have been designed and evaluated in the research community, such as TransType (Langlais et al., 2000), Thot (Ortiz-Martínez et al., 2010), and Caitra (Koehn, 2009). Green et al. (2014) investigates the trade-off between human effort and translation quality within the paradigms of post-editing and interactive MT.

A growing line of research has explored the use of neural machine translation with attention (Bahdanau et al., 2014) in an interactive setting. Wuebker et al. (2016) compares the performance of neural and statistical machine translation models for interactive prediction, and shows that neural models are substantially more accurate. Knowles and Koehn (2016) also demonstrates that neural models provide more accurate interactive predictions than statistical models and addresses efficiency challenges. Hokamp and Liu (2017) describes a search algorithm for neural models that specifically targets a typical interactive workflow in which the terms in a bilingual lexicon must be prioritized over alternatives.

Werling et al. (2015) investigates the trade-off between the cost of human intervention and accuracy for three other tasks: named-entity recognition, sentiment classification, and image classification. That work also proposes an approach to decision making that considers the uncertain long-term consequences of actions.

Mi et al. (2016) demonstrates the usefulness of providing additional attention information to a fully automated neural machine translation system. In this work, the authors add an additional loss to the translation model which encourages the attention computed by the NMT system to resemble alignments predicted by an IBM word alignment model.

3 Guided Attention

Neural machine translation with attention (Bahdanau et al., 2014) is a variant of the seq2seq model (Sutskever et al., 2014) that incorporates attention over the source encodings into the decoder. The attention is a distribution over source positions that can be interpreted as a soft indicator of what part of the source sentence will be translated next. We propose to replace the attention predicted by the model with a *guided* attention distribution that is provided directly by a human selecting a source word. In this paper, we simulate the human selection using the source word that is most helpful in the translation decision, described in detail below.

3.1 Neural Machine Translation with Attention

Given a source sentence $x = x_1, \ldots, x_n$ and a target sentence $y = y_1, \ldots, y_m$, the model first encodes x to form input representations $z_1, \ldots, z_n$. To predict the target labels y, the model conditions on a concatenation of two vectors, one being a hidden representation of the output generated so far, and the other being the input representations weighted by the attention: $\sum_i \alpha_i^{(t)} z_i$, where $\alpha_i^{(t)}$ is the attention computed at time t for the ith word in the source sentence. The input representations and hidden decoder states can be defined using an LSTM (Bahdanau et al., 2014) or convolution (Gehring et al., 2017) over word embeddings.

The attention vector is a distribution over source positions: $\sum_i \alpha_i^{(t)} = 1$ and $\alpha_i^{(t)} \geq 0$. To compute $\alpha_i^{(t)}$, a feed-forward neural network is used that takes in as inputs (z_i, h_t) where h_t is the hidden decoder state at time t. Finally, given the attention, hidden decoder state, and input representations, the label y_t is predicted using a learned distribution $p(y_t | h_t, \sum_i \alpha_i^{(t)} z_i)$.

3.2 Simulated Attention

Instead of using human input to train the model, we attempt to simulate the behavior of an accurate human, allowing for faster and cheaper training. We do this by, at each time step, calculating the distribution over the target vocabulary $p(y_t | h_t, z_i)$ for each i, which is equivalent to evaluating a one-hot attention vector for each source sentence word. We then provide the one-hot attention for the source word that had the highest predictive probability for the correct next target word to be translated. That is, if $i^* = \arg\max_i p(y^* | h_j, z_i)$, where y^* is the correct target word, then

$$\alpha^{(t)} = e_{i^*} \implies \sum_i \alpha_i^{(t)} z_i = z_{i^*}.$$

4 Learning When to Ask for Guidance

Given that we have a method for simulating the guidance that a human would provide, we turn to the problem of deciding when to request guidance at all. Each request for guidance affects the input representation used for predicting a single word. Over the course of a sentence, the system can request guidance multiple times.

4.1 Interaction Policy

To implement our interactive method, we use a greedy decoder. For each predicted word, the model decides whether to translate using guided attention or to translate using the attention predicted by the model. At the end of each iteration, there will be a loss penalty corresponding to the amount of guidance requested as well as the likelihood of the sentence under the model. Guidance improves likelihood by providing more information to each decision, but incurs a penalty for requesting guidance.

4.2 Interactive Machine Translation as Reinforcement Learning

We believe that reinforcement learning is an appropriate framework for our set up, since deciding when to ask for assistance can have long term ramifications on final accuracy that are hard to anticipate before training. We therefore model our framework by a Markov decision process (MDP). In this MDP, our agent is the machine translation system, whose actions are whether or not to request guided attention, and our reward function is the cross-entropy between our prediction of the next word and a distribution that predicts the reference with probability 1.

4.3 Reinforcement Learning

An MDP is a tuple (S, A, T, R). S is the set of all possible states that an agent can be in. A is the set of all possible actions the agent can take. T is the transition function $p(s_{t+1}|s_t, a_t) = T(s_{t+1}|s_t, a_t)$ that is the distribution over the next state given the current state and the action to be taken. Finally, R is the reward function $R(s_{t+1}, a_t, s_t)$ that determines the reward for transitioning into s_{t+1} from s_t with action a_t.

An agent acting in a MDP can be described by a policy function $\pi : S \rightarrow A$, that takes in states and returns actions. It is the goal of reinforcement learning to learn a policy that maximizes the expected sum of (discounted) rewards: $\mathbf{E}[\sum_t \gamma^t R(s_{t+1}, \pi(s_t), s_t)]$, where $\gamma \in (0, 1]$ is the discount factor.

In the case of interactive attention in machine translation, a state s_t captures the activation of the translation network just before it would generate the next target word w_t. There are only two possible actions: whether to go ahead and generate w_t or to request guidance. If guidance is requested, then a new activation of the translation network is computed by replacing the model's attention weights with the guide's attention weights, and then a new word w'_t is generated using these new activations. If guidance is not requested, then w_t is generated. In either case, the reward function is the cross entropy sequence loss of the correct translation.

4.3.1 Policy Gradient

Policy gradient is a common reinforcement learning method to learn a policy π_θ parameterized by θ. The policy gradient method aims to perform stochastic gradient ascent on the objective

$$J(\theta) = \mathbf{E}\left[\sum_{t=1}^{T-1} \gamma^t R(s_{t+1}, \pi_\theta(s_t), s_t)\right].$$

Let $\pi_\theta(a_t|s_t)$ be the probability of choosing an action a_t in state s_t according to the policy π_θ. The policy gradient theorem states that if a_t are sampled according to $\pi_\theta(s_t)$, and s_{t+1} are sampled according to $T(\cdot|s_t, a_t)$, then an unbiased estimator of $\nabla_\theta J(\theta)$ is

$$\sum_{t=1}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau).$$

Although using the above expression is an unbiased estimator, it can have high variance, prompting the use of variance reduction methods. For any function $b(s)$, the following is also an unbiased estimator:

$$\sum_{t=1}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} (R(s_{\tau+1}, a_\tau, s_\tau) - b(s_\tau)),$$

And the choice that minimizes variance is

$$b(s_\tau) = \mathbf{E}_{\pi_\theta}\left[\sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right].$$

This optimal $b(s_\tau)$ can be approximated by a parameterized function V_ϕ, where we learn V_ϕ by approximately minimizing

$$\mathbf{E}_{\pi_\theta}\left(V_\phi(s_t) - \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right)^2 .$$

Finally, a policy gradient algorithm alternates between taking a step of stochastic gradient ascent on $J(\theta)$ and taking multiple gradient steps on V_ϕ.

When using the approximate value function V_ϕ to reduce variance, the inner expression of the gradient is typically called the *advantage function* and denoted $A(s_t)$:

$$A(s_t) = \sum_{\tau=t}^{T-1} \left[\gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right] - V_\phi(s_t).$$

For our value function, we use a feed-forward neural network with two hidden layers of 32 units each, and for our policy function we use a neural network with one 32-unit hidden layer. The input to the former is the standard decoder inputs, which consist of the previously output token and the weighted sum of the hidden representations $\sum_i \alpha_i^{(t)} z_i$. The input to the latter additionally includes the original softmax layer input.

4.4 Action Frequency Regularization

Since our goals are to maximize translation accuracy while minimizing the number of times a human would have to intervene, we introduce an action weight parameter w_a, in order to manage the trade-off between accuracy and human effort. To promote accuracy during training, we have part of the reward at time step t be the negative cross entropy of the predictions at time t. To incorporate the number of times that the system requests guidance, we include not only the probability of requesting guidance, but also whether or not guidance was requested. In addition to these, we incorporate a threshold parameter ρ_a, to ensure that the action probabilities do not exceed the designated value. We thus use the following policy gradient objective:

$$\hat{A}(s_t) \cdot \log p_\theta(a_t) + w_a \cdot \max(0, p(a_t) - \rho_a) \cdot a_t,$$

where a_i is a binary scalar that takes value 1 if guidance was requested, and 0 otherwise, and $\hat{A}$ is the standardized advantage function.

That is,

$$\hat{A}(s_t) = \frac{A(s_t) - \mu(A(s_t))}{\sigma(A(s_t))}.$$

5 Experiments

We evaluate our model on the task of translating from English to German. Specifically, we first train a sequence-to-sequence model with attention, and then continue training using our reinforcement learning model. The baseline neural machine translation model was trained for 508,387 iterations.

5.1 Datasets

We use the English-German WMT 2016 news task dataset, which contains 4.2 million training sentence pairs. We apply BPE with 32,000 merge operations.

5.2 Architecture Details

For our base NMT system, we used Google's large seq2seq system implementation (Britz et al., 2017). For the encoder, we had 512 hidden units. For the decoder, both the GRU and the attention have 512 units. [1]

5.3 Results

We evaluate our approach on all 3000 sentences of the WMT 2016 news-test2013 development set. We first evaluate the baseline fully automatic NMT model, which yields a BLEU (Papineni et al., 2002) score of 19.37. In comparison, our model which asks for guidance with a 100% probability has a BLEU score of 32.51. Thus, requesting guidance indeed improves translation quality for this model. However, requesting guidance for every word would require maximal human effort, as the human translator would be required to click at each time step.

We also evaluate a variety of learned policies on the same data and using the same baseline model. During policy learning, the parameters of the translation model are frozen, and only the parameters of the policy and value functions are learned. Varying the action weight and threshold values yields various guidance frequencies and corresponding BLEU scores. To determine whether the learned policy is requesting guidance efficiently, for each trained policy we also evaluate a random policy that asks for guidance with the same frequency as the reinforcement learning policy (Figure 5.3). The learned policy was able to achieve a BLEU score of 27.25 with observed guidance of about 54%, which improved upon the random policy by almost 2 BLEU and upon the baseline model by about 8 BLEU.

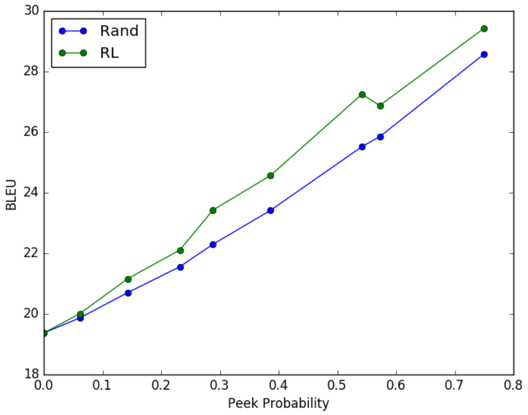

Figure 1: Translation accuracy for a random policy (blue) and a learned policy (green), for different guidance frequencies. More guidance provides higher accuracy. Across a range of guidance frequencies, the learned policy outperforms a policy that makes the same number of guidance requests, but at randomly chosen times.

6 Analysis

We compare the simulated clicks to the attention generated by the neural machine translator. In order to compare them, we compare the optimal word attention location computed by our simulator against the word with the largest weight according to the NMT system. This does provide a problem if the NMT was attending primarily to more than a single object, but nevertheless we believe this method of comparison may still provide useful intuition. In the figure below we

[1] For full specification see: `https://github.com/google/seq2seq/blob/master/example_configs/nmt_large.yml`

only include arrows for which the attended words differ. We note that using the simulated attention seems mostly intuitive with respect to where a human translator would click and corrects some of the NMT system errors. In particular, it makes *von* point to *of* and *zu* point to *counter*. However, there are also a few quirks. For example, it makes *kanishe* point to *Republic* and EOS point to *to*.

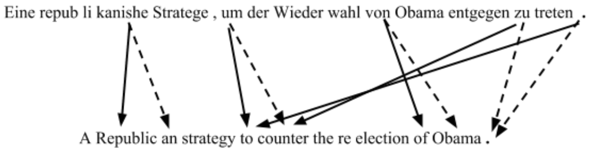
Eine repub li kanishe Stratege , um der Wieder wahl von Obama entgegen zu treten .
A Republic an strategy to counter the re election of Obama .

Figure 2: Guided attention (solid) vs NMT attention (dashed)

7 Future work

Our experiments demonstrate that reinforcement learning is an effective framework for requesting human guidance in interactive machine translation. However, we can identify several open questions that merit further investigation. First, we have focused on greedy decoding in this paper, because it is not trivial to apply a more sophisticated search procedure on top of our method. Developing an extension that incorporates beam search could improve performance. Second, during baseline training, the attention mechanism sees soft attention over the entire sentence as opposed to one hot attention over a single word, and the discrepancy between training and testing may limit the performance of the system. In addition, this method assumes that the word that gives the best predictive probability of the next target word is the same word that a human would choose. Another related limitation with our system is that it assumes that the previous system output is the same as the correct translation, and so the best next word to be translated by the system is the same as that of the reference translation.

As our approach is intended to reduce human effort, we look forward to conducting human subject experiments in future work, to see whether the gains we witnessed in simulation carry over to real-world conditions. One interesting direction that our method could provide is investigating whether the behaviors of humans interacting with such a system may be the same as those when interacting with other humans, and if not, to test in which ways human actions might be similar and how they may diverge from expected behavior. Another extension to this work would be incorporating the attention supervision into the main model. Currently, if asked to translate the same sentence twice, the current framework would ask for the same attention help twice, which seems inherently wasteful. Ideally, after getting the supervision, it would be able to incorporate it into the model to reduce redundant queries.

8 Conclusion

We have demonstrated an approach to interactive machine translation that aims to limit the amount of effort required by human translators while maintaining translation quality. We hope that our method inspires further research into this area.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate.

Britz, D., Goldie, A., Luong, M.-T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Foster, G. and Lapalme, G. (2002). *Text prediction for translators*. Université de Montréal.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Green, S., Wang, S. I., Chuang, J., Heer, J., Schuster, S., and Manning, C. D. (2014). Human effort and machine learnability in computer aided translation. In *EMNLP*, pages 1225–1236.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Knowles, R. and Koehn, P. (2016). Neural interactive translation prediction. *AMTA 2016, Vol.*, page 107.

Koehn, P. (2009). A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Langlais, P., Foster, G., and Lapalme, G. (2000). Transtype: a computer-aided translation typing system. In *Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems-Volume 5*, pages 46–51. Association for Computational Linguistics.

Mi, H., Wang, Z., and Ittycheriah, A. (2016). Supervised Attentions for Neural Machine Translation.

Ortiz-Martínez, D., García-Varea, I., and Casacuberta, F. (2010). Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, Montral, Canada.

Werling, K., Chaganty, A. T., Liang, P. S., and Manning, C. D. (2015). On-the-Job Learning with Bayesian Decision Theory. In *Advances in Neural Information Processing Systems*, pages 3465–3473, Montral, Canada.

Wuebker, J., Green, S., DeNero, J., Hasan, S., and Luong, M.-T. (2016). Models and inference for prefix-constrained machine translation. *54th ACL*, 1:66–75.