He et al.: Dual Learning for Machine Translation

Leander Girrbach

January 2020

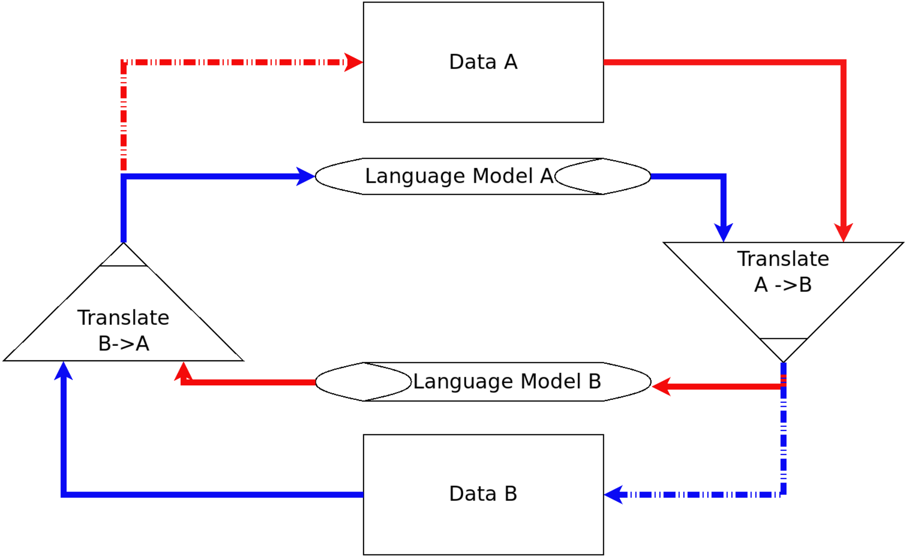
Problem

- MT needs aligned parallel corpora

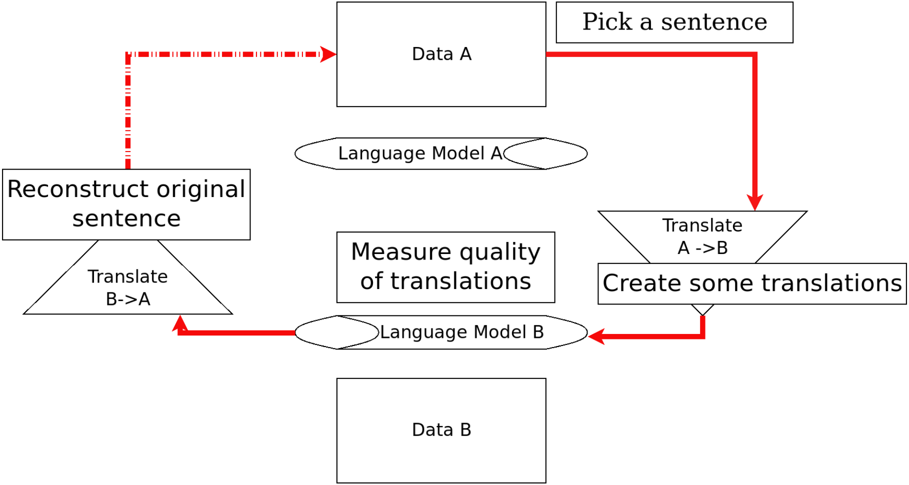
⇒ Difficult to obtain

- Idea: Learn from monolingual data (unsupervisedly)

Proposed unsupervised method: Overview



Proposed unsupervised method: Details



Ingredients

- Sentences in language A, B
- Language Models for A, B
  $LM_A,\ LM_B$
- Translation Models for $A \rightarrow B$ and $B \rightarrow A$
  Parametrised by Θ_{AB} and Θ_{BA}

How to train the translation models

- Translations are sampled

⇒ non-differentiable

⇒ no end-to-end training possible

- Idea: Use Reinforcement Learning instead

Some concepts of RL

General

- Agent in environment performing actions
- Environment specifies **state** s and possible **actions** a
- Try to maximise (expected) **reward** r (provided by environment)
- Probability of action a in state s given by **policy** $\pi(s, a)$

Here (sort of)

State = "Original" sentence s

Action = Translated sentence s_{mid}

Policy = Translation models

Reward = Feedback from language model and reconstructed sentence

Policy gradients

- Given state s, use differentiable function (approximator) to calculate probabilities for possible actions
- Function (parametrised by Θ) represents policy π_Θ
- Example: NN calculates action probabilities from state representation
- Change parameters so that actions with high reward get high probability

How to train the translation models

Scenario: Pick sentence of language A

1. s_{mid} := Translated sentence from A to B
2. Calculate rewards

$$r_1 = LM_B(s_{mid}) \quad r_2 = \log P(s \mid s_{mid}; \Theta_{BA}) \quad r = \alpha r_1 + (1-\alpha) r_2$$

3. Update parameters (gradient ascent)

$$\nabla_{\Theta_{AB}} E[r] = E[r \nabla_{\Theta_{AB}} \log P(s_{mid} \mid s;\ \Theta_{AB})]$$
$$\nabla_{\Theta_{BA}} E[r] = E[(1-\alpha) \nabla_{\Theta_{BA}} \log P(s \mid s_{mid};\ \Theta_{BA})]$$

Deriving the gradients

Policy Gradient Theorem

$$\nabla_\Theta J(\Theta) \propto \sum_s d^\pi(s) \sum_a \nabla_\theta \pi_\Theta(a \mid s) Q^\pi(s, a)$$

$$\nabla_\Theta E[r] = \sum_{s_{mid}} \nabla_\Theta \pi_\Theta(s_{mid} \mid s) r$$

Gradients for Θ_{AB}

$$\begin{aligned}\nabla_{\Theta_{AB}} E[r] &= \sum_{s_{mid}} P(s_{mid} \mid s; \Theta_{AB}) r \frac{\nabla_{AB} P(s_{mid} \mid s; \Theta_{AB})}{P(s_{mid} \mid s; \Theta_{AB})} \\ &= \mathbb{E}\left[r \nabla_{AB} \log P(s_{mid} \mid s; \Theta_{AB})\right]\end{aligned}$$

Deriving the gradients

Policy Gradient Theorem

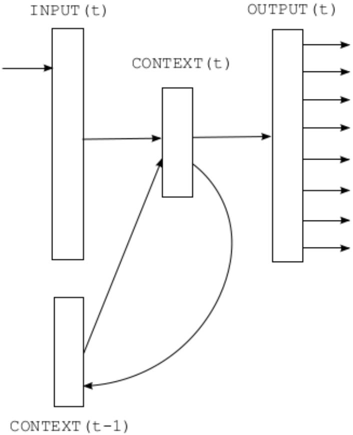
$$\nabla_\Theta E[r] = \sum_{s_{mid}} \nabla_\Theta \pi_\Theta(s_{mid} \mid s) r$$

Gradients for Θ_{BA}

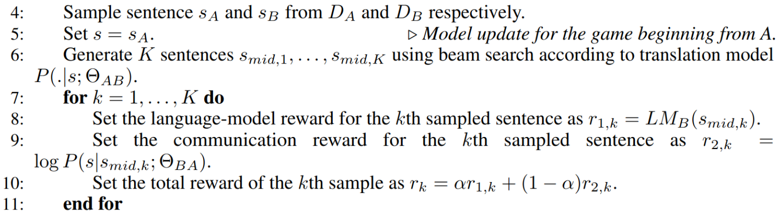
$$\begin{aligned}
\nabla_{\Theta_{BA}} E[r] &= \sum_{s_{mid}} P(s_{mid} \mid s; \Theta_{AB}) \nabla_{BA} r \\
&= \sum_{s_{mid}} P(s_{mid} \mid s; \Theta_{AB}) \cdot \\
&\qquad (\nabla_{BA} \alpha LM_B(s_{mid}) + \nabla_{BA}(1-\alpha) \log P(s \mid s_{mid}; \Theta_{BA}) \\
&= \mathbb{E}\left[(1-\alpha)\nabla_{BA} \log P(s \mid s_{mid}; \Theta_{BA})\right]
\end{aligned}$$

Language Models

- Trained on monolingual data
- Training: Predict next word conditioned on all words on the left
- RNN for prediction (Mikolov et al. (2010))

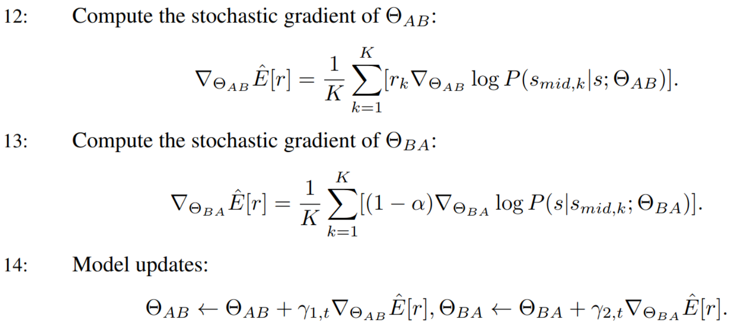


A look at the pseudo-code



```
    Sample sentence $s_A$ and $s_B$ from $D_A$ and $D_B$ respectively.
    Set $s = s_A$.                ▷ *Model update for the game beginning from A.*
    Generate $K$ sentences $s_{mid,1}, \dots, s_{mid,K}$ using beam search according to translation model $P(.|s;\Theta_{AB})$.
    for $k = 1, \dots, K$ do
        Set the language-model reward for the $k$th sampled sentence as $r_{1,k} = LM_B(s_{mid,k})$.
        Set the communication reward for the $k$th sampled sentence as $r_{2,k} = \log P(s|s_{mid,k};\Theta_{BA})$.
        Set the total reward of the $k$th sample as $r_k = \alpha r_{1,k} + (1-\alpha) r_{2,k}$.
    end for
```

A look at the pseudo-code



```
Compute the stochastic gradient of $\Theta_{AB}$:
    $\nabla_{\Theta_{AB}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^{K} [r_k \nabla_{\Theta_{AB}} \log P(s_{mid,k} | s; \Theta_{AB})].$
Compute the stochastic gradient of $\Theta_{BA}$:
    $\nabla_{\Theta_{BA}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^{K} [(1 - \alpha) \nabla_{\Theta_{BA}} \log P(s | s_{mid,k}; \Theta_{BA})].$
Model updates:
    $\Theta_{AB} \leftarrow \Theta_{AB} + \gamma_{1,t} \nabla_{\Theta_{AB}} \hat{E}[r], \Theta_{BA} \leftarrow \Theta_{BA} + \gamma_{2,t} \nabla_{\Theta_{BA}} \hat{E}[r].$
```

Learning from parallel and monolingual data

- Sample sentences from parallel and monolingual corpora
- Update parameters according to RL and supervised learning loss
- Decrease ratio of parallel data over time ("warm start")

Pseudo-NMT Baseline & Joint Training

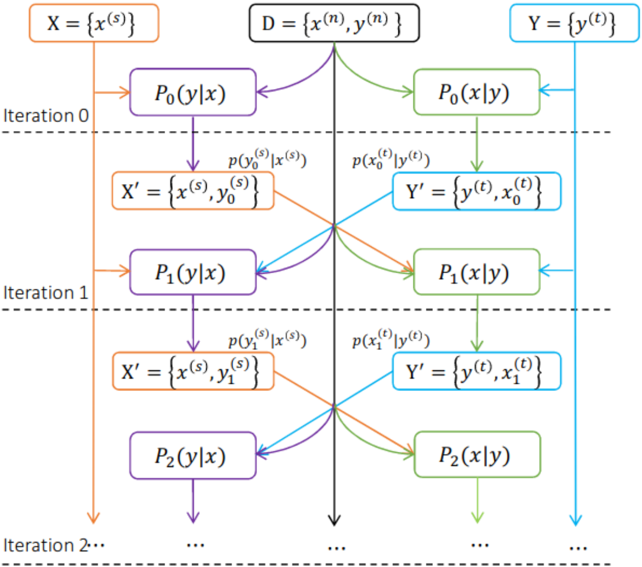
Backtranslation

1. Train TMs on parallel data
2. Create "synthetic" translations from monolingual data
3. Retrain with augmented dataset

Joint Training

- Repeat creation of "synthetic" translations
  ⇒ improved automatically generated translations from improved TMs
- Weight "synthetic" translations by translation probability

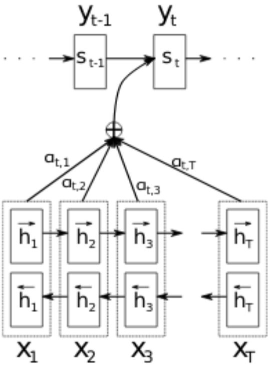
Joint training: Illustration



Experiments

Settings

Data	from WMT'14
Translation Models	Encoder-Decoder (Bahdanau Attention)
Batch Size	80
Maximum Sentence length	50
Beam size	2
Evaluation Metric	BLEU



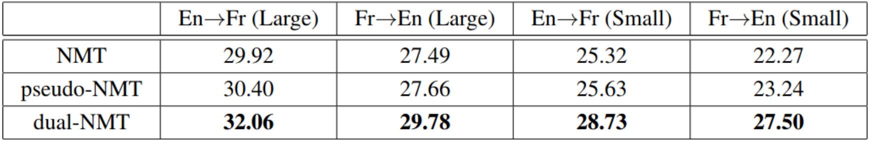
Results

NMT Bahdanau Attention Encoder-Decoder

pseudo-NMT with dataset augmentation (automatic translation)

Large 100% of parallel data

Small 10% of parallel data



	En→Fr (Large)	Fr→En (Large)	En→Fr (Small)	Fr→En (Small)
NMT	29.92	27.49	25.32	22.27
pseudo-NMT	30.40	27.66	25.63	23.24
dual-NMT	**32.06**	**29.78**	**28.73**	**27.50**

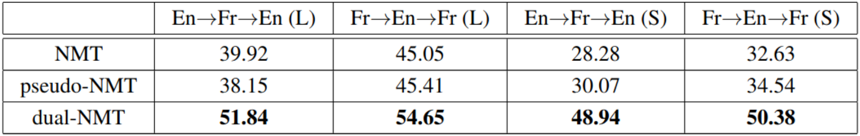
- Improvement greater with less parallel bilingual data

Comparison to other models on En→Fr(large)

ConvS2S 41.44

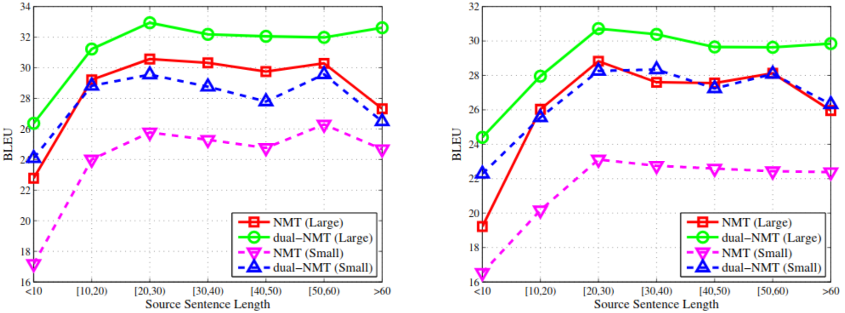
Transformer (Big) 41.8

Reconstruction Results



	En→Fr→En (L)	Fr→En→Fr (L)	En→Fr→En (S)	Fr→En→Fr (S)
NMT	39.92	45.05	28.28	32.63
pseudo-NMT	38.15	45.41	30.07	34.54
dual-NMT	**51.84**	**54.65**	**48.94**	**50.38**

BLEU Scores/sentence length



(a) En→Fr

(b) Fr→En

Literature I

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

Lilian Weng. Policy gradient algorithms. *lilianweng.github.io/lil-log*, 2018. URL `https://lilianweng.github.io/lil-log/2018/04/08/policy-gradient-algorithms.html`.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

Literature II

Stefan Riezler and Sariya Karimova. Policy gradient methods, 2019. URL `https://www.cl.uni-heidelberg.de/courses/ss19/HRL/material/rl-intro.28-40.pdf`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*, 2017.

Literature III

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*, 2018.

Jiajun Zhang and Chengqing Zong. Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv:1610.07272*, 2016.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.

Literature IV

Philip Arthur, Graham Neubig, and Satoshi Nakamura. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*, 2016.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*, 2015.

Wikipedia contributors. Noisy channel model — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Noisy_channel_model&oldid=897908960`, 2019. [Online; accessed 15-January-2020].

Questions

Question

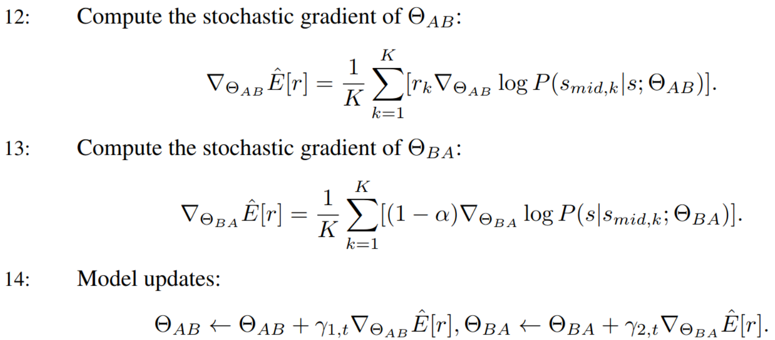
- The reward is a real value computed from a sentence on the $LM_B(.)$ funtion, how is computed?

Some information

- "log likelihood of a received message was used to reward [. . .] the translation model"
- My interpretation ($s_{\text{mid}} = w_1 \dots w_T$):

$$r_{LM} = \log \prod_{t=1}^{T} P_{LM}\left(w_t \mid w_1; \dots; w_{t-1}\right)$$

Questions



```
12:    Compute the stochastic gradient of $\Theta_{AB}$:
           $\nabla_{\Theta_{AB}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^{K} [r_k \nabla_{\Theta_{AB}} \log P(s_{mid,k} | s; \Theta_{AB})].$
13:    Compute the stochastic gradient of $\Theta_{BA}$:
           $\nabla_{\Theta_{BA}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^{K} [(1-\alpha) \nabla_{\Theta_{BA}} \log P(s | s_{mid,k}; \Theta_{BA})].$
14:    Model updates:
           $\Theta_{AB} \leftarrow \Theta_{AB} + \gamma_{1,t} \nabla_{\Theta_{AB}} \hat{E}[r], \Theta_{BA} \leftarrow \Theta_{BA} + \gamma_{2,t} \nabla_{\Theta_{BA}} \hat{E}[r].$
```

Figure: can you explain the model update in algorithm 1 in the paper?

Questions

Question

- Could you briefly explain how the pseudo-NMT model work? Are there some details provided?
- How does the pseudo-NMT work?

Some information

- basically: Train translation models, then augment dataset by automatically generated translations and train again
- ”For the baseline pseudo-NMT [11], we used the trained NMT model to generate pseudo bilingual sentence pairs from monolingual data, removed the sentences with more than 50 words, merged the generated data with the original parallel training data, and then trained the model for testing.”

Questions

Question

- What is a 'noisy channel'?
- What is the general idea of a noisy channel? (Where does the term "noisy" originate from?)

Wikipedia definition

- "The noisy channel model is a framework used in spell checkers, question answering, speech recognition, and machine translation. In this model, the goal is to find the intended word given a word where the letters have been scrambled in some manner."

Noisy channel: Details (Wikipedia)

Given an alphabet Σ , let Σ^* be the set of all finite strings over Σ . Let the dictionary D of valid words be some subset of Σ^*, i.e., $D \subseteq \Sigma^*$.
The **noisy channel** is the matrix

$$\Gamma_{ws} = \Pr(s|w)$$

where $w \in D$ is the intended word and $s \in \Sigma^*$ is the scrambled word that was actually received.

Questions

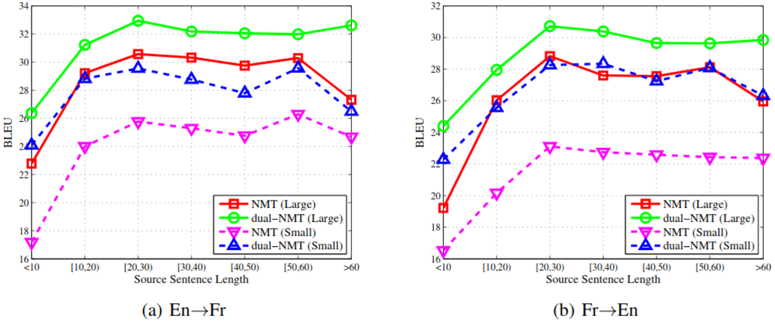


Figure: Why does the performance of NMT large and small in the En→Fr direction of NMT go in opposite directions for long source sentences as opposed to Fr→En, where the bleu score does not fluctuate as much?

Questions

Question

- Can be considered like the porposed approximation as an special case of finetuning?

Questions

Question

- Am I correct in my interpretation that the algorithm first goes through the code for LM_{AB} and then LM_{BA} (line 16 "[...] symmetrically"? If so, why not use a more thought out scheduling system, e.g. depended on the update-size/the improvement of each model?

My thoughts

- LMs not updated
- Improvement difficult to measure
- Prevent adaption to other TM's errors
  ⇒ difficult to balance
- "Keep both TMs moving"

Questions

Question

- Why was the beam search size set to 2 instead of a bigger number? Any reasoning/intuition for this?

Some information

- Tunable hyperparameter
- Rewards: high variance → "noisy gradients"
- Maybe using larger beam worsens effects of high variance

Questions

Question

- If the dual learning approach significantly outperforms NMT models which require parallel data, why is it not more widely used?(maybe it is and I'm just not aware of it) Is current research focusing on this kind of approach?
- Was dual learning used on some of the other tasks proposed in the paper and was it as successful as in this paper?

Some information

- Google Scholar: 303 citations (14.01.2020)
- For example: Deliberation networks
- For example: Image domain translation, Image captioning
- "Human parity" in English-Chinese translation (Hassan et al. (2018))

Questions

Question

- Even for the Small model, the authors still warm-start on 1.2M bilingual sentence pairs, which is not really that few. Nonetheless, performance of the Small model already decreases quite a bit. As far as I can see, the authors don't report a "cold start" at all - thus, how much of a performance drop could we expect in that case? Would this approach still be useful when there is no parallel data whatsoever for the language pair, or significantly less than 1.2M sentence pairs?

My thoughts

- If it could learn without any parallel data, this would have been mentioned in the paper

Questions

Question

- The idea in this work is similar [to] "On Using Monolingual Corpora in Neural Machine Translation". Has there been any comparison between the performance of the approaches made? Especially since this paper also claims to significantly improve model performance in low resources cases (second setting "small")

My thoughts

- No direct comparison (to my knowledge)
- Indirect comparison: He et al. (2016) (this) > Sennrich et al. (2016) (pseudo-nmt) > Gulcehre et al. (2015) (above)

Questions

Question

- What are the drawbacks of dual/looped learning? Is the training time more costly than for an average NMT model?

My thoughts

- Training time: 1 week
  Convolutional Seq2Seq: 8 GPUs for about 37 days
  Transformer: 3.5 days on eight GPUs
  "Bahdanau attention": 5 days
- Drawback: Indirect optimisation through RL
- Maybe hard to train
- Depends on good language model (?)
- Speculation: Still needs *some* parallel data for warm start

Questions

Question

- If i understand it correctly, the rewards receive a different weighting through alpha. Do the authors provide an explanation for this choice?

My thoughts

- No ($\Rightarrow$ hyperparameter)
- Intuition: balance rewards
- Example: Prevent bad language model from destroying the gradients

Questions

Question

- Do you think that completely out-of-domain monolingual data would strongly decrease the quality of the system?

My thoughts

- Depends on the ratio of parallel and monolingual data
- Otherwise, I don't see a problem
  (given enough training time, as distinguishing domains complicates the learning process)

Questions

Question

- Could it happen that the two models develop some sort of "code" (such as a "degenerated" version of French in the case of the experimental setup) that makes internal communication between the agents efficient evaluation-wise, yet works around actually producing natural translations?

My thoughts

- I agree with the objection
  (Evidence: Reconstruction performance much better than translation performance)

Questions

Question

- Consider the case where there are two words in D_A that only appear once. Can we expect the model to learn the correctly translate them into language B, if we do not pretrain the translation system?

My thoughts

- Speculation: TMs develop "slang"
  (Evidence: Reconstruction performance much better than translation performance)
  $\Rightarrow$ TMs shift meaning (of rare words)
- Speculation: Language model will assign low probability to rare words $\Rightarrow$ TMs try to avoid rare words

Questions

Question

- If not, can we incorporate a dictionary to alleviate this problem?

Some information

Zhang and Zong (2016) Generate backtranslations especially for rare words (apparently phrase-based SMT can guarantee the dictionary translation)

Luong et al. (2014) Use special output tokens indicating aligned source words

Arthur et al. (2016) Use dictionary translation probabilities to bias/interpolate prediction probabilities