Dual Learning for Machine Translation

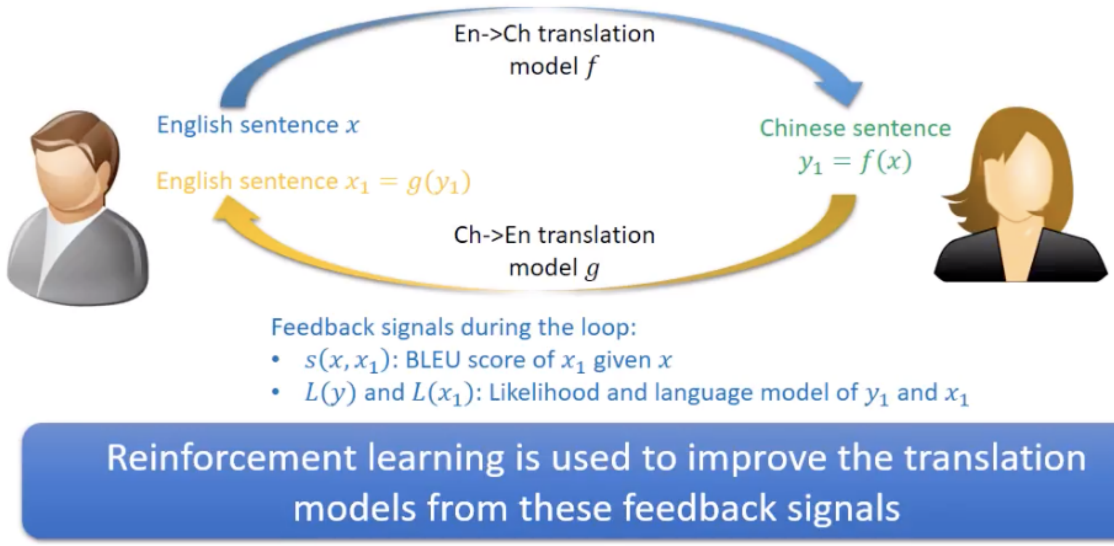
Created	@May 17, 2019 10:15 AM
Tags	
Updated	@Jun 03, 2019 5:10 PM

Reference

[Dual Learning for Machine Translation](https://arxiv.org/abs/1611.00179)
https://arxiv.org/abs/1611.00179

[Improving Neural Machine Translation Models with Monolingual Data](https://arxiv.org/abs/1511.06709)
https://arxiv.org/abs/1511.06709

Introduction



- Dual-learning mechanism
    - use monolingual data (in both the source and target languages) with 10% bilingual data for warm start)
    - monolingual data can play a similar role to the parallel bilingual data
    - significantly reduce the requirement on parallel bilingual data for training
    - described as the following two-agent communication game
- Procedure
    1. The first agent, who only understands language A, sends a message in language A to the second agent through a noisy channel, which converts the message from language A to language B using a translation model.
    2. The second agent, who only understands language B, receives the translated message in language B. She checks the message and notifies the first agent whether it is a natural sentence in language B (note that the second agent may not be able to verify the correctness of the translation since the original message is invisible to her). Then she sends the received message back to the first agent through another noisy channel, which converts the received message from language B back to language A using another translation model.
    3. After receiving the message from the second agent, the first agent checks it and notifies the second agent whether the message she receives is consistent with her original message. Through the feedback, both agents will know whether the two communication channels (and thus the two translation models) perform well and can improve them accordingly.
    4. The game can also be started from the second agent with an original message in language B, and then the two agents will go through a symmetric process and improve the two channels (translation models) according to the feedback.
- use reinforcement procedure (e.g., by means of the policy gradient methods)

Background: Neural Machine Translation

- encoder (RNN)

$$h_i = f(h_{i-1}, x_i) \tag{1}$$

  - xi: source language word
  - hi: encoder hieedn state

- decoder

$$P(y_t|y_{<t}, x) \propto \exp(y_t; r_t, c_t) \tag{2}$$

$$r_t = g(r_{t-1}, y_{t-1}, c_t) \tag{3}$$

$$c_t = q(r_{t-1}, h_1, \cdots, h_{T_x}) \tag{4}$$

  - yt: target language word
  - rt: decoder hidden state
  - ct: contextual information in generating word yt ‘global’ signal summarizing sentence x or ‘local’ signal implemented by an attention mechanism

- learning objective

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \sum_{(x,y)\in D} \sum_{t=1}^{T_y} \log P(y_t|y_{<t}, x; \Theta) \tag{5}$$

Dual Learning for Neural Machine Translation

Algorithm 1 The dual-learning algorithm

1: **Input**: Monolingual corpora D_A and D_B, initial translation models Θ_{AB} and Θ_{BA}, language models LM_A and LM_B, hyper-parameter α, beam search size K, learning rates $\gamma_{1,t}, \gamma_{2,t}$.
2: **repeat**
3: $t = t + 1$.
4: Sample sentence s_A and s_B from D_A and D_B respectively.
5: Set $s = s_A$. ▷ *Model update for the game beginning from A.*
6: Generate K sentences $s_{mid,1}, \dots, s_{mid,K}$ using beam search according to translation model $P(.|s; \Theta_{AB})$.
7: **for** $k = 1, \dots, K$ **do**
8: Set the language-model reward for the kth sampled sentence as $r_{1,k} = LM_B(s_{mid,k})$.
9: Set the communication reward for the kth sampled sentence as $r_{2,k} = \log P(s|s_{mid,k}; \Theta_{BA})$.
10: Set the total reward of the kth sample as $r_k = \alpha r_{1,k} + (1 - \alpha) r_{2,k}$.
11: **end for**
12: Compute the stochastic gradient of Θ_{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 Θ_{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:

$$\Theta_{AB} \leftarrow \Theta_{AB} + \gamma_{1,t} \nabla_{\Theta_{AB}} \hat{E}[r], \Theta_{BA} \leftarrow \Theta_{BA} + \gamma_{2,t} \nabla_{\Theta_{BA}} \hat{E}[r].$$

15: Set $s = s_B$. ▷ *Model update for the game beginning from B.*
16: Go through line 6 to line 14 symmetrically.
17: **until** convergence

Supposed

- 2 corpora (Da, Db)
  - not necessarily aligned with each other
  - have no topical relationship with each other at all
- 2 (weak) translation models
  - translate sentences from A to B and verse visa
- corpus Da contains Na sentences, and Da contains Nb sentences
- P (.|s; ΘAB) and P(.|s;ΘBA) as two neural translation models
- two well-trained language models LMA(.) and LMB(.): how confident the sentence is a natural sentence
- reward r1 = LMB(Smid): how natural the output sentence is in language B

- reward r2 = log P (s|Smid; ΘBA): log probability of S recovered from Smid
- total reward; r = αr1 + (1 − α)r2 (α is a hyper-parameter)
- policy gradient

$$\nabla_{\Theta_{BA}} E[r] = E[(1-\alpha)\nabla_{\Theta_{BA}} \log P(s|s_{mid};\Theta_{BA})] \tag{6}$$

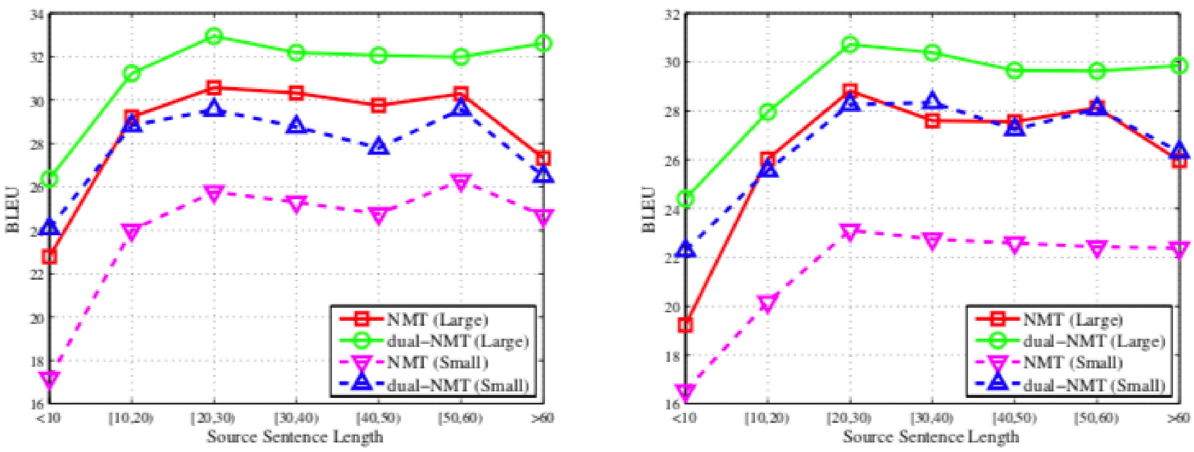
$$\nabla_{\Theta_{AB}} E[r] = E[r\nabla_{\Theta_{AB}} \log P(s_{mid}|s;\Theta_{AB})] \tag{7}$$

- use beam search to obtain more meaningful results (more reasonable middle translation outputs) for gradient computation

Experiments

blue socre

	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**



(a) En→Fr (b) Fr→En

Reconstruction performance blue score

	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**

Code Review

NonameAuPlatal/Dual_Learning
https://github.com/NonameAuPlatal/Dual_Learning

yistLin/pytorch-dual-learning
https://github.com/yistLin/pytorch-dual-learning

pytorch/examples
https://github.com/pytorch/examples/tree/master/word_language_model