A review on Deliberation Networks: Sequence Generation Beyond One-Pass Decoding*

[1]Yingce Xia, [2]Fei Tian, [3]Lijun Wu, [1]Jianxin Lin, [2]Tao Qin, [1]Nenghai Yu, [1]Tie-Yan Liu
[1]University of Science and Technology of China, Hefei, China
[2]Microsoft Research, Beijing, China
[3]Sun Yat-sen University, Guangzhou, China
[1]yingce.xia@gmail.com, linjx@mail.ustc.edu.cn, ynh@ustc.edu.cn
[2]{fetia,taoqin,tie-yan.liu}@microsoft.com, [3]wulijun3@mail2.sysu.edu.cn

1 Introduction

The paper is about building a refinement decoder on top of the existing sequence-to-sequence model, which is aimed to polish a sequence generated by the base model. As long as the base model's decoding is autoregressive, while decoding a current token there's no information about next tokens in the decoding sequence. So adding a refinement model, which «knows» about next tokens can potentially improve quality. Authors also pay attention to the fact that people usually firstly generate a draft and then do some polishing of the generated text. Authors test their model on machine translation and text summarization tasks.

2 Novelty and related work

It's a relatively new approach, there are few works on refinement of the generated text. Although there are some approaches to perform post-editing (Chatterjee et al., 2016, Niehues et al., 2016). Deliberation process is slightly different, and the difference is a deliberation network is trained end-to-end, but in post-editing approaches refinement was performed by a separate seq2seq algorithm. End-to-end training makes the whole system consistent, gradients flow through entire network, affect all parameters of the net and inputs of several parts of the whole system can be changed (unlike a case if several parts are trained one after another). This can potentially lead to better quality, and that's what authors find in their experiments.

3 Model architecture

Authors use two architectures as base models: standard encoder-decoder model with attention of Bahdanau et al., 2015 also known as RNNSearch and Google's GNMT (Y. Wu et al., 2016). After publication of these papers a lot of other architectures have been proposed (Gehring et al., 2017, Vaswani et al., 2017, Chen et al., 2018, F. Wu et al., 2019) that tend to work better then the standard model, so obviously there should be an improving after replacement of the base model's architecture with a transformer (Vaswani et al., 2017).

In order to build a consistent model, authors model conditional probability $P(y|y', x, \theta)$ with the deliberation network (which is a very straightforward way in this case), which is identical to a standard decoder from Bahdanau et al., 2015, the only difference is it takes other variables as input, because it's build on top of the base model. The base model takes source sequence x as input and produces conditional distribution $P(y'|x, \theta)$ (encoder and decoder function will be omitted in

*This work was done when Yingce Xia, Lijun Wu and Jianxin Lin were interns at Microsoft Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

notation for simplicity). The goal is to maximize $\log P(Y|X,\theta)$ using maximum likelihood method, where $(X,Y) = \{(x_i, y_i)\}_{i=1}^n$ is the training set. Using the law of total probability we have

$$P(y|x,\theta) = \sum_{y' \in W} P(y|y', x, \theta) P(y'|x, \theta)$$

This probability is hard to compute even for one pair (x, y), because there's a sum over $|W|$ summands, which is usually about 10^5 or more (depending on vocabulary size). The gradients w.r.t. model parameters are even harder to compute. Let's take one group of parameters to show it: θ_1 are the parameters of decoder in the base model.

$$\mathcal{L}(x, y, \theta_1) = \log \sum_{y' \in W} P(y|y', x) P(y'|x, \theta_1)$$

$$\nabla_{\theta_1} \mathcal{L}(x, y, \theta_1) = \frac{\sum\limits_{y' \in W} P(y|y', x) \nabla_{\theta_1} P(y'|x, \theta_1)}{\sum\limits_{y' \in W} P(y|y', x) P(y'|x, \theta_1)}$$

Authors don't try to calculate these gradients (nor accurately, nor approximately). Instead they replace the loss function with it's lower bound using the Jensen's inequality:

$$\log \sum_{y' \in W} P(y|y', x) P(y'|x, \theta_1) \geq \sum_{y' \in W} \Big[\log P(y|y', x) \Big] P(y'|x, \theta_1)$$

then differentiate the lower bound and make Monte-Carlo estimate of the gradients, going to doubly stochastic optimization. Here's a problem: a gap between the true loss function and it's lower bound can be too high. So optimizing a variational lower bound instead of that authors proposed can make the training slower (because it's harder to optimize a variational lower bound), but more accurate. Even if our translation model is strong enough and almost always produces probability vectors over tokens close to one-hot vectors (and then Jensen's inequality gives a good approximation) we can start training a weeker model, achieve quality close to quality of the strong model, but save inference time, because inference in weaker models is faster.

EM-algorithm can't be directly used here (despite the fact discrete distributions are conjugate), because we try to get rid of large computations in denominator of the Bayes formula. Introduction of a variational lower bound can be done exactly as it's done in VAE (Kingma et al., 2014), assuming that $\{y'_i\}_{i=1}^n$ are latent variables, but it's not a VAE-like model and it can't be optimized the same way, because reparametrization trick doesn't work with latent variables, so Gumbel-softmax trick (Jang et al., 2017) can be used here.

Also adding a deliberation part and variational inference can easily lead to overfitting especially using a strong base model such as transformer big with variational inference, so it's necessary to carefully observe the metrics on the validation set.

4 Data augmentation

As long as VAE learns a variational distribution $q(z|x,\phi)$, which tries to be close to posterior distribution $\big(P(y'|y)$ in this case$\big)$, we get a deterioration model, which can help us create a lot of dirty translations given some clean (which can be generated in large amount from a strong monolingual language model of target language). This can be used for training set augmentation, and training loss can be rewritten with the additional summand, which is loss on synthetic data. In some later papers (Hassan et al., 2018, Freitag et al., 2019) back-translation was used for this purpose, but it seems like back-translation doesn't give ideal training data despite that it models mistakes of a translation system. The problem is that a sentence in the target language is translated twice, and translations in the source language may not be representative samples from unconditional distribution of the source language (due to mistakes, which the translation model do). The non-representative sentences are translated back in the target language, and these samples are used to train the post-editing system. It seems the usage of GANs or VAEs can help train a better polishing system, because these generative models can produce examples, which back-translation doesn't produce due to some basic mistakes. This training data can for additional training objective, which can be added to the initial loss with some coefficient λ, which is a hyperparameter.

5 Optimization and inference

Authors use beam search of size 2 for intermediate sequence generation instead of teacher forcing, which forces the whole model behave the same way during training and inference. On inference step authors use beam search of size 12 to generate refined sequences. Comparing a 2-layer decoder of the standart model without refinement with a deliberation network consisting of 2 one-layer decoders (one from the base model and one from the deliberation part), deliberation network performs beam search twice, which is slightly more expensive as long as beam search is the most expensive part of decoding. Another problem is that these 2 beam search steps are not parallelizable, because the whole sequence y' is used in attention module for generating tokens of y. Assuming wall time of a decoder is proportional to beam size, adding one more beam search step is still appropriate and still can be implemented in a production systems. Training time is comparable with the other models too (5 days with a standard RNNSearch (1-layer encoder, 1-layer first-step decoder and 1-layer second-step or deliberation decoder, all networks are GRUs with 1000-neuron hidden layers), where all parts are pre-trained within 2 weeks on a single NVIDIA K40 GPU).

6 Metrics and quantitative results

Authors achieve 0.97 BLEU gain compared to the previous SOTA in machine translation task on WMT'14 dataset and 2.64, 0.96 and 1.81 gain of ROUGE-1, ROUGE-2, ROUGE-L respectively on Gigaword Corpus. Authors measure only BLEU for MT task, but their feature could affect fluency, number of grammatical errors and other linguistic metrics. It would be interesting to find the these metrics increased too, though it takes time to do assesory marking.

7 Future work

I'll recap key ideas of improvement of the model introduced in the article:

1. Replacement of the base architecture from RNN (or LSTM) with transformer;
2. Using stochastic variational inference like that used to train VAE with Gumbel-softmax in order to achieve better translation quality or in order to decrease number of parameters in a network leaving the same quality;
3. Using a GAN or a VAE to generate additional training samples for the deliberation network;
4. To measure other metrics addition to BLEU.

Before implementing all of these methods, it needs to be verified (if possible) whether these modifications lead to improvement. For example we can explicitly measure the gap in Jensen's inequality and measure distance between the empirical distribution of coarse sequences and empirical distribution of those generated by back-translation.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

[2] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Association for Computational Linguistics*.

[3] Rajen Chatterjee, Jos e G. C. de Souza, Matteo Negri, Marco Turchi. The fbk participation in the wmt 2016. Automatic post-editing shared task. 2016. In *Proceedings of the First Conference on Machine Translation*: Volume 2, Shared Task Papers.

[4] Markus Freitag, Isaac Caswell, Scott Roy. 2019. APE at Scale and its Implications on MT Evaluation Biases. In *WMT*.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning* - Volume 70, pages 1243–1252. JMLR.org.

[6] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Feder-mann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving Human Parity on Automatic Chinese to English News Translation. In *arXiv*.

[7] Eric Jang, Shixiang Gu, Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

[8] Diederik P Kingma, Max Welling. 2014. Auto-encoding variational Bayes. In *ICLR*.

[9] Jan Niehues, Eunah Cho, Thanh-Le Ha, Alex Waibel. 2016. Pre-translation for neural machine translation. In *COLING*.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[11] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convlutions. In *Proc. of ICLR*

[12] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.