Bag-of-Words as Target for Neural Machine Translation

Shuming Ma[1], Xu Sun[1,2], Yizhong Wang[1], Junyang Lin[3]
[1]MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[2]Deep Learning Lab, Beijing Institute of Big Data Research, Peking University
[3]School of Foreign Languages, Peking University
{shumingma, xusun, yizhong, linjunyang}@pku.edu.cn

Abstract

A sentence can be translated into more than one correct sentences. However, most of the existing neural machine translation models only use one of the correct translations as the targets, and the other correct sentences are punished as the incorrect sentences in the training stage. Since most of the correct translations for one sentence share the similar bag-of-words, it is possible to distinguish the correct translations from the incorrect ones by the bag-of-words. In this paper, we propose an approach that uses both the sentences and the bag-of-words as targets in the training stage, in order to encourage the model to generate the potentially correct sentences that are not appeared in the training set. We evaluate our model on a Chinese-English translation dataset, and experiments show our model outperforms the strong baselines by the BLEU score of 4.55.[1]

1 Introduction

Neural Machine Translation (NMT) has achieve success in generating coherent and reasonable translations. Most of the existing neural machine translation systems are based on the sequence-to-sequence model (Sutskever et al., 2014). The sequence-to-sequence model (Seq2Seq) regards the translation problem as the mapping from the source sequences to the target sequences. The encoder of Seq2Seq compresses the source sentences into the latent representation, and the decoder of Seq2Seq generates the target sentences from the source representations. The cross-entropy loss, which measures the distance of the generated distribution and the target distribution, is minimized in the training stage, so that the generated sentences are as similar as the target sentences.

Source:	今年前两月广东高新技术产品出口37.6亿美元。
Reference:	Export of high - tech products in guangdong in first two months this year reached 3.76 billion us dollars .
Translation 1:	Guangdong 's export of new high technology products amounts to us $3.76 billion in first two months of this year .
Translation 2:	Export of high - tech products has frequently been in the spotlight , making a significant contribution to the growth of foreign trade in guangdong .

Table 1: An example of two generated translations. Although Translation 1 is much more reasonable, it is punished more severely than Translation 2 by Seq2Seq.

Due to the limitation of the training set, most of the existing neural machine translation models only have one reference sentences as the targets. However, a sentence can be translated into more than one correct sentences, which have different syntax structures and expressions but share the same meaning. The correct translations that are not appeared in the training set will be punished as the incorrect translation by Seq2Seq, which is a potential harm to the model. Table 1 shows an example of two generated translations from Chinese to English. Translation 1 is apparently more proper as the translation of the source sentence than Translation 2, but it is punished even more severely than Translation 2 by Seq2Seq.

Because most of the correct translations for one source sentence share the similar bag-of-words, it

[1] The code is available at https://github.com/lancopku/bag-of-words

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 332–338
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

is possible to distinguish the correct translations from the incorrect ones by the bag-of-words in most cases. In this paper, we propose an approach that uses both sentences and bag-of-words as the targets. In this way, the generated sentences which cover more words in the bag-of-words (e.g. Translation 1 in Table 1) are encouraged, while the incorrect sentences (e.g. Translation 2) are punished more severely. We perform experiments on a popular Chinese-English translation dataset. Experiments show our model outperforms the strong baselines by the BLEU score of 4.55.

2 Bag-of-Words as Target

In this section, we describe the proposed approach in detail.

2.1 Notation

Given a translation dataset that consists of N data samples, the i-th data sample (x^i, y^i) contains a source sentence x^i, and a target sentence y^i. The bag-of-words of y^i is denoted as b^i. The source sentence x^i, the target sentence y^i, and the bag-of-words b^i are all sequences of words:

$$x^i = \{x_1^i, x_2^i, ..., x_{L_i}^i\}$$

$$y^i = \{y_1^i, y_2^i, ..., y_{M_i}^i\}$$

$$b^i = \{b_1^i, b_2^i, ..., b_{K_i}^i\}$$

where L_i, M_i, and K_i denote the number of words in x^i, y^i, and b^i, respectively.

The target of our model is to generate both the target sequence y^i and the corresponding bag-of-words b^i. For the purpose of simplicity, $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{b})$ is used to denote each data pair in the rest of this section.

2.2 Bag-of-Words Generation

We regard the bag-of-words generation as the multi-label classification problem. We first perform the encoding and decoding to obtain the scores of words at each position of the generated sentence. Then, we sum the scores of all positions as the sentence-level score. Finally, the sentence-level score is used for multi-label classification, which identifies whether the word appears in the translation.

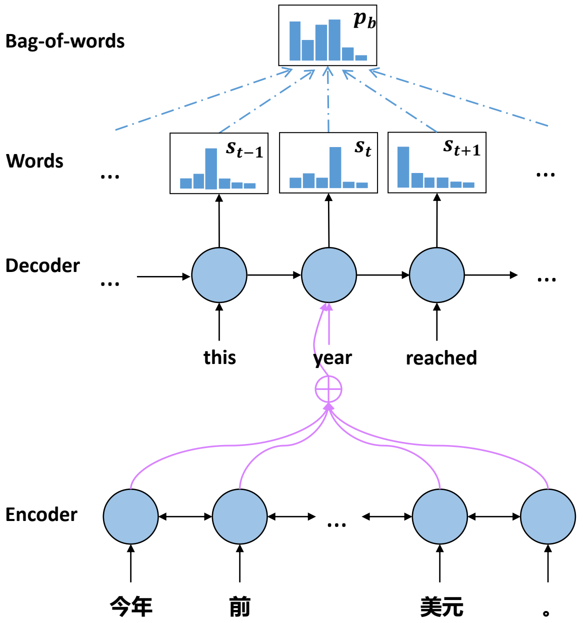

Figure 1: The overview of our model. The encoder inputs the source sentence, and the decoder outputs the word distribution at each position. The distribution of all position is summed up to a sentence-level score, which can be used to generate the bag-of-words.

In our model, the encoder is a bi-directional Long Short-term Memory Network (BiLSTM), which produces the representation $\boldsymbol{h} = \{h_1, h_2, ..., h_L\}$ from the source text $\boldsymbol{x}$:

$$\vec{h}_t = \vec{f}(x_t, \vec{h}_{t-1}) \tag{1}$$

$$\overleftarrow{h}_t = \overleftarrow{f}(x_t, \overleftarrow{h}_{t+1}) \tag{2}$$

$$h_t = \vec{h}_t + \overleftarrow{h}_t \tag{3}$$

where $\vec{f}$ and $\overleftarrow{f}$ are the forward and the backward functions of LSTM for one time step, $\vec{h}_t$ and $\overleftarrow{h}_t$ are the forward and the backward hidden outputs respectively, x_t is the input at the t-th time step, and L is the number of words in sequence $\boldsymbol{x}$.

The decoder consists of a uni-directional LSTM, with an attention, and a word generator. The LSTM generates the hidden output q_t:

$$q_t = f(y_{t-1}, q_{t-1}) \tag{4}$$

where f is the function of LSTM for one time step, and y_{t-1} is the last generated words at t-th time step. The attention mechanism (Luong and Manning, 2015) is used to capture the source information:

$$v_t = \sum_{i=1}^{N} \alpha_{ti} h_i \tag{5}$$

$$\alpha_{ti} = \frac{e^{g(q_t, h_i)}}{\sum_{j=1}^{N} e^{g(q_t, h_j)}} \quad (6)$$

$$g(q_t, h_i) = \tanh(q_t^T W_t h_i) \quad (7)$$

where W_t is a trainable parameter matrix. Then, the word generator is used to compute the probability of each output word at t-th time step:

$$p_{w_t} = softmax(s_t) \quad (8)$$

$$s_t = W_g v_t + b_g \quad (9)$$

where W_g and b_g are parameters of the generator.

To get a sentence-level score for the generated sentence, we generate a sequence of word-level score vectors s_t at all positions with the output layer of decoder, and then we sum up the word-level score vectors to obtain a sentence-level score vector. Each value in the vector represents the sentence-level score of the corresponding word, and the index of the value is the index of the word in the dictionary. After normalizing the sentence-level score with sigmoid function, we get the probability for each word, which represents how possible the word appears in the generated sentence regardless of the position in the sentence. Compared with the word-level probability p_{w_t}, the sentence-level probability p_b of each word is independent of the position in the sentence.

More specifically, the sentence-level probability of the generated bag-of-words p_b can be written as:

$$p_b = sigmoid(\sum_{t=1}^{M} s_t) \quad (10)$$

where M is the number of words in the target sentence.

2.3 Targets and Loss Function

We have two targets at the training stage: the reference translation (appears in the training set) and the bag-of-words. The bag-of-words is used as the approximate representation of the correct translations that do not appear in the training set. For the targets, we have two parts of loss functions:

$$l_1 = -\sum_{t=1}^{M} y_t \log p_{w_t}(y_t) \quad (11)$$

$$l_2 = -\sum_{i=1}^{K} b_i \log p_b(b_i) \quad (12)$$

The total loss function can be written as:

$$l = l_1 + \lambda_i l_2 \quad (13)$$

where λ_i is the coefficient to balance two loss functions at i-th epoch. Since the bag-of-words generation module is built on the top of the word generation, we assign a small weight for the bag-of-words training at the initial time, and gradually increase the weight until a certain value λ:

$$\lambda_i = min(\lambda, k + \alpha i) \quad (14)$$

In our experiments, we set the $\lambda = 1.0$, $k = 0.1$, and $\alpha = 0.1$, based on the performance on the validation set.

3 Experiments

This section introduces the details of our experiments, including datasets, setups, baseline models as well as results.

3.1 Datasets

We evaluated our proposed model on the NIST translation task for Chinese-English translation and provided the analysis on the same task. We trained our model on 1.25M sentence pairs extracted from LDC corpora [2], with 27.9M Chinese words and 34.5M English words. We validated our model on the dataset for the NIST 2002 translation task and tested our model on that for the NIST 2003, 2004, 2005, 2006, 2008 translation tasks. We used the most frequent 50,000 words for both the Chinese vocabulary and the English vocabulary. The evaluation metric is BLEU (Papineni et al., 2002).

3.2 Setting

We implement the models using *PyTorch*, and the experiments are conducted on an *NVIDIA 1080Ti* GPU. Both the size of word embedding and hidden size are 512, and the batch size is 64. We use Adam optimizer (Kingma and Ba, 2014) to train the model with the default setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$, and we initialize the learning rate to 0.0003.

Based on the performance on the development sets, we use a 3-layer LSTM as the encoder and a 2-layer LSTM as the decoder. We clip the gradients (Pascanu et al., 2013) to the maximum norm

[2] The corpora include LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

Model	MT-02	MT-03	MT-04	MT-05	MT-06	MT-08	All
Moses (Su et al., 2016)	33.19	32.43	34.14	31.47	30.81	23.85	31.04
RNNSearch (Su et al., 2016)	34.68	33.08	35.32	31.42	31.61	23.58	31.76
Lattice (Su et al., 2016)	35.94	34.32	36.50	32.40	32.77	24.84	32.95
CPR (Zhang et al., 2017)	33.84	31.18	33.26	30.67	29.63	22.38	29.72
POSTREG (Zhang et al., 2017)	34.37	31.42	34.18	30.99	29.90	22.87	30.20
PKI (Zhang et al., 2017)	36.10	33.64	36.48	33.08	32.90	24.63	32.51
Bi-Tree-LSTM (Chen et al., 2017)	36.57	35.64	36.63	34.35	30.57	-	-
Mixed RNN (Li et al., 2017)	37.70	34.90	38.60	35.50	35.60	-	-
Seq2Seq+Attn (our implementation)	34.71	33.15	35.26	32.36	32.45	23.96	31.96
+Bag-of-Words (this paper)	**39.77**	**38.91**	**40.02**	**36.82**	**35.93**	**27.61**	**36.51**

Table 2: Results of our model and the baselines (directly reported in the referred articles) on the Chinese-English translation. "-" means that the studies did not test the models on the corresponding datasets.

of 10.0. Dropout is used with the dropout rate set to 0.2. Following Xiong et al. (2017), we use beam search with a beam width of 10 to generate translation for the evaluation and test, and we normalize the log-likelihood scores by sentence length.

3.3 Baselines

We compare our model with several NMT systems, and the results are directly reported in their articles.

- **Moses** is an open source phrase-based translation system with default configurations and a 4-gram language model trained on the training data for the target language.

- **RNNSearch** (Bahdanau et al., 2014) is a bidirectional GRU based model with the attention mechanism. The results of Moses, and RNNSearch come from Su et al. (2016).

- **Lattice** (Su et al., 2016) is a Seq2Seq model which encodes the sentences with multiple tokenizations.

- **Bi-Tree-LSTM** (Chen et al., 2017) is a tree-structured model which models source-side syntax.

- **Mixed RNN** (Li et al., 2017) extends RNNSearch with a mixed RNN as the encoder.

- **CPR** (Wu et al., 2016) extends RNNSearch with a coverage penalty.

- **POSTREG** (Ganchev et al., 2010) extends RNNSearch with posterior regularization with a constrained posterior set. The results of CPR, and POSTREG come from Zhang et al. (2017).

- **PKI** (Zhang et al., 2017) extends RNNSearch with posterior regularization to integrate prior knowledge.

3.4 Results

Table 2 shows the overall results of the systems on the Chinese-English translation task. We compare our model with our implementation of Seq2Seq+Attention model. For fair comparison, the experimental setting of Seq2Seq+Attention is the same as BAT, so that we can regard it as our proposed model removing the bag-of-words target. The results show that our model achieves the BLEU score of 36.51 on the total test sets, which outperforms the Seq2Seq baseline by the BLEU of 4.55.

In order to further evaluate the performance of our model, we compare our model with the recent NMT systems which have been evaluated on the same training set and the test sets as ours. Their results are directly reported in the referred articles. As shown in Table 2, our model achieves high BLEU scores on all of the NIST Machine Translation test sets, which demonstrates the efficiency of our model.

We also give two translation examples of our model. As shown in Table 3, The translations of Seq2Seq+Attn omit some words, such as "*of*", "*committee*", and "*protection*", and contain some redundant words, like "*human chromosome*" and "*<unk>*". Compared with Seq2Seq, the translations of our model is more informative and ade-

Source: 人类共有二十三对染色体。
Reference: Humans have a total of 23 pairs of chromosomes .
Seq2Seq+Attn: Humans have 23 pairs chromosomes in human chromosome .
+Bag-of-Words: There are 23 pairs of chromosomes in mankind .
Source: 一名奥林匹克筹备委员会官员说:「这项倡议代表筹委会对环保的敏感性。」
Reference: An official from the olympics organization committee said : “ this proposal represents the committee ’s sensitivity to environmental protection . ”
Seq2Seq+Attn: An official of the olympic preparatory committee said : “ this proposal represents the <unk> of environmental sensitivity . ”
+Bag-of-Words: An official of the olympic preparatory committee said : “ this proposal represents the sensitivity of the preparatory committee on environmental protection . ”

Table 3: Two translation examples of our model, compared with the Seq2Seq+Attn baseline.

quate, with a better coverage of the bag-of-words of the references.

4 Related Work

The studies of encoder-decoder framework (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014) for this task launched the Neural Machine Translation. To improve the focus on the information in the encoder, Bahdanau et al. (2014) proposed the attention mechanism, which greatly improved the performance of the Seq2Seq model on NMT. Most of the existing NMT systems are based on the Seq2Seq model and the attention mechanism. Some of them have variant architectures to capture more information from the inputs (Su et al., 2016; Xiong et al., 2017; Tu et al., 2016), and some improve the attention mechanism (Luong et al., 2015; Meng et al., 2016; Mi et al., 2016; Jean et al., 2015; Feng et al., 2016; Calixto et al., 2017), which also enhanced the performance of the NMT model.

There are also some effective neural networks other RNN. Gehring et al. (2017) turned the RNN-based model into CNN-based model, which greatly improves the computation speed. Vaswani et al. (2017) only used attention mechanism to build the model and showed outstanding performance. Also, some researches incorporated external knowledge and also achieved obvious improvement (Li et al., 2017; Chen et al., 2017).

There is also a study (Zhao et al., 2017) shares a similar name with this work, i.e. bag-of-word loss, our work has significant difference with this study. First, the methods are very different. The previous work uses the bag-of-word to constraint the latent variable, and the latent variable is the output of the encoder. However, we use the bag-of-word to supervise the distribution of the generated words, which is the output of the decoder. Compared with the previous work, our method directly supervises the predicted distribution to improve the whole model, including the encoder, the decoder and the output layer. On the contrary, the previous work only supervises the output of the encoder, and only the encoder is trained. Second, the motivations are quite different. The bag-of-word loss in the previous work is an assistant component, while the bag of word in this paper is a direct target. For example, in the paper you mentioned, the bag-of-word loss is a component of variational autoencoder to tackle the vanishing latent variable problem. In our paper, the bag of word is the representation of the unseen correct translations to tackle the data sparseness problem.

5 Conclusions and Future Work

We propose a method that regard both the reference translation (appears in the training set) and the bag-of-words as the targets of Seq2Seq at the training stage. Experimental results show that our model obtains better performance than the strong baseline models on a popular Chinese-English translation dataset. In the future, we will explore how to apply our method to other language pairs, especially the morphologically richer languages than English, and the low-resources languages.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61673028), National High Technology Research and Development Program of China (863 Program, No. 2015AA015404), and the National Thousand Young Talents Program. Xu Sun is the corresponding author of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1913–1924.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *ACL 2017*, pages 1936–1945.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014*, pages 1724–1734.

Shi Feng, Shujie Liu, Nan Yang, Mu Li, Ming Zhou, and Kenny Q. Zhu. 2016. Improving attention modeling with implicit distortion and fertility for machine translation. In *COLING 2016*, pages 3082–3092.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML 2017*, pages 1243–1252.

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

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP 2013*, pages 1700–1709.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *ACL 2017*, pages 688–697.

Junyang Lin, Shuming Ma, Qi Su, and Xu Sun. 2018. Decoding-history-based adaptive control of attention for neural machine translation. *CoRR*, abs/1802.01812.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*, pages 1412–1421.

Shuming Ma, Xu Sun, Wei Li, Sujian Li, Wenjie Li, and Xuancheng Ren. 2018. Query and output: Generating words by querying distributed word representations for paraphrase generation. In *NAACL 2018*.

Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. In *COLING 2016*, pages 2174–2185.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *EMNLP 2016*, pages 2283–2288.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL, 2002*, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.

Jinsong Su, Zhixing Tan, Deyi Xiong, and Yang Liu. 2016. Lattice-based recurrent neural network encoders for neural machine translation. *CoRR*, abs/1609.07730.

Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. 2017a. meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *ICML 2017*, pages 3299–3308.

Xu Sun, Bingzhen Wei, Xuancheng Ren, and Shuming Ma. 2017b. Label embedding network: Learning label representation for soft training of deep networks. *CoRR*, abs/1710.10393.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS, 2014*, pages 3104–3112.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1054–1059.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *ACL 2016*.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Hao Xiong, Zhongjun He, Xiaoguang Hu, and Hua Wu. 2017. Multi-channel encoder for neural machine translation. *CoRR*, abs/1712.02109.

Jingjing Xu, Xu Sun, Xuancheng Ren, Junyang Lin, Binzhen Wei, and Wei Li. 2018a. Dp-gan: Diversity-promoting generative adversarial network for generating informative and diversified text. *CoRR*, abs/1802.01345.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018b. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL 2018*.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017. Prior knowledge integration for neural machine translation using posterior regularization. In *ACL 2017*, pages 1514–1523.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664.