IMPROVING ADVERSARIAL NEURAL MACHINE TRANSLATION WITH PRIOR KNOWLEDGE

Yating Yang , Xiao Li, Tonghai Jiang, Jinying Kong, Bo Ma, Xi Zhou, Lei Wang
1 Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences
2 Xinjiang Laboratory of Minority Speech and Language Information Processing
Urumqi, China
yangyt@ms.xjb.ac.cn

***Abstract*—Generative adversarial networks (GANs) has achieved great success in the field of image processing, Adversarial Neural Machine Translation(NMT) is the application of GANs to machine translation. Unlike previous work training NMT model through maximizing the likelihood of the human translation, Adversarial NMT minimizes the distinction between human translation and the translation generated by a NMT model. Even though Adversarial NMT has achieved impressive results, while using little in the way of prior knowledge. In this paper, we integrated bilingual dictionaries to Adversarial NMT by leveraging a character model. Extensive experiment shows that our proposed methods can achieve remarkable improvement on the translation quality of Adversarial NMT, and obtain better result than several strong baselines.**

Keywords—Neural Machine Translation;Prior Knowledge; DeepLearning;Uyghur

I. INTRODUCTION

End-to-end translation model become the novel paradigm and achieved the new state-of-the-art translation performance. Unlike most deep neural network simulates real data distribution by maximizing the likelihood of labeled-data, Generative adversarial networks(GANs) minimize the distinction between labeled-data and generated-data. Common end-to-end translation model, such as attention-based NMT model, directly encode the input sentence information and generate translation through deep neural network. Adversarial NMT introduce the strength of GANs to enhance the performance of NMT: we train the Adversarial NMT model through an elaborately designed Convolutional Neural network which also named adversary.

Though novel Adversarial NMT become the new state-of-the-art in some language pairs. It remains a challenging problem how to get a satisfy result by NMT on low-resource language pair such as Uyghur-Chinese. As we all know, NMT is a data-driven method which needs more scale parallel corpus than other statistical machine translation models such as phrase-based model. Due to lack of corpus, the translation of NMT on Uyghur-Chinese is filled with out-of-vocabulary(OOV) words, as Figure1 shows.

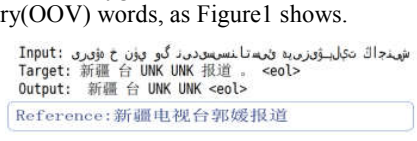

Fig1. An example of OOVs in Uyghur-Chinese machine translation

In this paper, we proposed an improved Adversarial NMT model to cope with the problem of NMT on limited size corpus. In order to introduce extra prior knowledge, we integrated bilingual dictionaries to Adversarial NMT by leveraging a character model.Figure2 shows how our model works. As training Adversarial NMT is to force the translation results be as similar as ground-truth translations generated by human. We aim at making full use of all the bilingual dictionaries to Adversarial NMT. Our basic idea is to transform the low-frequency word pair in bilingual dictionaries into adequate sequence pairs which guarantee the frequent occurrence of the word pair, so that Adversarial NMT can translation mappings between the source word and the target word.

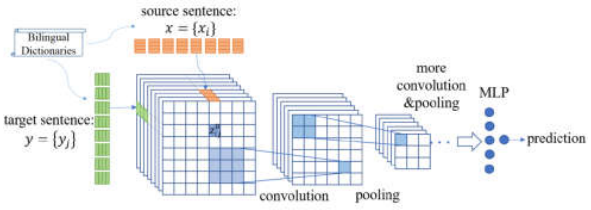

Fig2. Improved Adversarial-NMT by adding extra bilingual dictionaries

This paper is organized as follows. Section 1 is introduction, mainly introduces the background of our research. Section 2 is illustrations of some representative workings on NMT. The details of our improved Adversarial NMT are elaborated in Section 3. The settings and results of experiments on improved Adversarial NMT are given in Section 4. Section 5 is conclusions and future works.

II. RELATED WORK

While there has been substantial work on linguistically motivated SMT, approaches that leverage extra knowledge for NMT start to shed light very recently. Generally speaking, NMT can provide a flexible mechanism for adding extra knowledge, thanks to its strong capability of automatically learning feature representations.

(Sennrichet al.2016) define a few linguistically motivated features that are attached to each individual words. Their features include lemmas, subword tags, POS tags, dependency labels, etc. They concatenate feature embeddings with word embeddings and feed the concatenated embeddings into the NMT encoder. On the contrast, we do not specify any feature,

but let the model implicitly learn useful information from the structural label sequence. (Shi et al.2016) design a few experiments to investigate if the NMT system without external linguistic input is capable of learning syntactic information on the source-side as a by-product of training. However, their work is not focusing on improving NMT with linguistic input. Moreover, we analyze what syntax is disrespected in translation from several new perspectives.

Our work is inspired by (Zhang et al.16) and (Wu et.al 17). The former presented two ways to integrating bilingual dictionaries to attention-based NMT model, the latter proposed a novel Adversarial-NMT model. On basis of them, we presented an improved Adversarial-NMT with extra bilingual dictionaries.

III. INTERGERNING BILINGUAL DICTIONARIES

The word translation pairs in bilingual dictionaries are difficult to use in neural machine translation, mainly because they are rarely or never seen in the corpus. This paper attempt to build a bridge between Adversarial-NMT and bilingual dictionaries.

A. NMT model

NMT model consisting of an encoder and a decoder is a novel model for machine translation. Inputting to the layer in Encoder is word vectors in accordance with the source language sentences in the order of the input source language in each of the word through the Embedding. When read to the end of the sentence, the hidden layer in the decoder generates output in accordance with the state of the source language sentence and the encoder's hidden layer, and the hidden layer state is recorded for the next time step usage. The output layer of the decoder generates the probability of each target word according to the output of the hidden layer state and the relevant information of the source language. Finally, based on the probability of the target word generated from each time series, the maximum probability score of the target word is found by using the beam search algorithm.

The study shows that the encoder can get better translation results in reverse order to read the word. So this paper uses (two-way recurrent neural network) BiRNN network as the encoder. The encoder works as follows:

a. Encoder sequentially read word vector to calculate and record the hidden layer state of moment T by using formula 1.

b. Encoder reversely read word vector to calculate and record the hidden layer state of moment T by using formula 1.

$$\overrightarrow{h_t} = f(x^{(t)}, \overrightarrow{h_{(t-1)}}) \tag{1}$$

c. To calculate the state of the BiRNN encoder of the moment T. The calculation method of the hidden layer state of BiRNN at the moment T is shown in formula 2.

$$h_t = \begin{bmatrix} \overrightarrow{h_t} \\ \overleftarrow{h_t} \end{bmatrix} \tag{2}$$

d. Followed by an iterative execution of a, b, and c steps until the input is finished, is saved at each moment.

After reading all the source language sentences, we can get a series of encoder hidden states. General neural network translation system will take the last one of the fixed length vector as the source language for representing all information, and then input to the decoder. The attention based neural network translation will save all the states of hidden layer, and use a soft alignment model to select useful information between the source sentence and output word.

The alignment model in the neural network Machine Translation is different from the traditional statistical alignment model, and it is a kind of soft alignment model. Formula 3 represents the definition of the alignment model.

$$e_{ij} = a(s_{i-1}, h_j) \tag{3}$$

e_{ij} represents the matching digress of a word i in the target language with the word j in the source language. $a()$ represents a nonlinear function (also can be a feed forward neural network). $a()$ can be produced by s_{i-1} (the hidden layer state of RNN of moment $i-1$) together with h_j (the j word vector in the source sentence), and can also trained with other part of Machine Translation system (The problem can be seen as find the probability of h_j given s_{i-1}).

With the alignment model, we can get context dependent vector c_i . The calculation method can be calculated by the formula 4 and the formula 5.

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{4}$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{5}$$

Except the big difference on the calculation of context vector between attention-based and the other neural network, the method for calculating the probability score of the target word vector is similar.

$$s_i = \mathrm{f}(s_{i-1}, y_{i-1}, c_i) \tag{6}$$

According to the formula 6, we can calculate the hidden layer state of time by the hidden layer state of the time, the previous target word vector and the context dependent vector. is a nonlinear function.

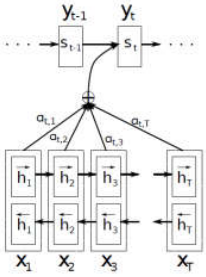

Fig3.The structure of attention-based NMT

Finally, we can get the probability score of the word by the hidden layer state of the $i-1$ time, the previous target word vector and context dependent vector.

$$p(y_i \mid y_1, \ldots, y_{i-1}, x) = g(y_{i-1}, s_i, c_i) \quad (7)$$

$g()$ in formula 7 in the general is a nonlinear function, but also can be a multi-layer neural network. It is important to note that the initial value of the hidden state in the general NMT decoder is the final value of the hidden state in the encoder. Figure3 illustrates the structure of attention-based NMT.

B. Adversarial-NMT

The adversary is used to differentiate translation result $y'=\{y_1,y_2,y_3,\ldots y_i\}$ and the ground-truth translation y, given the source language sentence x. To achieve that, one needs to measure the translative matching degree of source-target sentence pair (x; y). We turn to Convolution Neural Network (CNN) for this task (Yin et al., 2015; Hu et al., 2014), since with its layer-by-layer convolution and pooling strategies, CNN is able to accurately capture the hierarchical correspondence of (x; y) at different abstraction levels.

The general structure is shown in Figure 2. Specifically, given a sentence pair (x; y), we first construct a 2D image like representation by simply concatenating the embedding vectors of words in x and y. That is, for *i-th* word x_i in *x* and *j-th* word y_j in sentence *y*, we have the following feature map:

$$z^{(0)}_{i,j} = [x^T_i, y^T_i]^T \quad (8)$$

Based on such a 2D image-like representation, we perform convolution on every 3*3 window, with the purpose to capture the correspondence between segments in x and segments in y by the following feature map of type f:

After that we perform a max-polling in non-over lapping 2*2 windows:

$$z^{(2,f)}_{i,j} = \max(\{z^{(1,f)}_{2i-1,2j-1}, z^{(1,f)}_{2i-1,2j}, z^{(1,f)}_{2i,2j-1}, z^{(1,f)}_{2i,2j}\}) \quad (9)$$

We could go on for more layer of convolution and max-polling, aiming at capturing the correspondence at different levels of abstraction. The extracted features are then fed into a MLP whose last layer to give the probability that (x,y) is from ground-truth data.

With the notations for NMT model G and adversary model D, the final training objective is:

$$\min_G \max_D V(D,G) = E_{(x,y)\sim p_{data}(x,y)}[\log D(x,y)] + E_{x\sim p_{data}(x), y'\sim G(.\mid x)}[\log(1 - D(x,y'))] \quad (10)$$

Using the language of reinforcement, the NMT model *G(.|x)*is the conditional policy faced with *x*, while the term *log(1-D(x,y'))*, provided by the adversary *D* ,act as an Monte-Carlo estimation of the reward. Empirically, we calculate the gradient of Adversarial-NMT followed the method presented by (*Wu et.al 2017*).

C. Mixed word/character Model

NMT treats translation process a kind of classify problem: NMT model firstly select output words according to their corresponding probabilities, then search the best path in all results by decoding algorithm. It is obvious that NMT need a vocabulary which set before training. Too large vocabulary leads to high computational complexity, too small vocabulary cannot include enough words. All of these reasons cause the problem of OOVs in NMT.

We use a mixed word/character model to cope with the problem of OOVs in NMT. The mixed word/character model introduce data transformation using the character-based method. We all know that character is the basic unit in words and character are frequent even though the word is rare. However, a totally character-based NMT cannot guarantee the generated character sequence would lead to a real target language word. Therefore, we use the framework mixing the words and characters, which is employed to handle the problem of OOVs words.

We perform data transformation on both parallel training corpus and bilingual dictionaries. Here we choose Chinese sentences and words as examples. For each Chinese word w in a parallel sentence pair (X_b,y_b)or in a translation lexicon(Dic_x,Dic_y), if $w \in V$, *w* will be left as it is. Otherwise, w is re-labelled by character sequence.

IV. Experimental Settings

We describe experimental settings in this section including data sets, data preprocessing, the training and evaluation details, and all the translation methods we compare in the experiments.

A. Dataset

We apply our improved Adversarial-NMT on public corpus to verify the effectiveness. The corpus come from the CWMT 2015 public evaluation datasets and we use English-Chinese and Uyghur-Chinese corpus in news domain as our research objects. Since our model is used to machine translation, we divided the corpus into three parts: training set, test set and development set. The details of corpus are showed in Table 2. The parallel English-Chinese training data from CWMT contains 77.8M sentence pair. The parallel Uyghur-Chinese training data from CWMT contains about 0.14M sentence pair. The development set of English-Chinese contains 1K sentences. The development set of Uyghur-Chinese contains 1.1K sentences. Both test set of Uyghur-Chinese and English-Chinese contain 1k sentences.

TABLE I. The corpus of our experiments

Category	Training set	Development set	Test set
Uyghur-Chinese	139,792	1,100	1,000
English-Chinese	7780,000	1,000	1,100

B. Training and Evaluation Details

We compare our approach with three state-of-art Machine Translation system: a phrase-based SMT and an attention-based NMT, an Adversarial-NMT. In experiment, we tried the following combinations: phrase-based MT and phrase-based MT with mixed word/character model, attention-based NMT and attention-based NMT with mixed word/character model, Adversarial-NMT and Adversarial-NMT with mixed word/character model.

The baseline system of phrase-based model was executed on a computer with Moses 2.1 , 4GB memory and Ubuntu 12.04. The word-alignment tool which we selected is open source GIZA++ and then we use the strategy of 'grow-diag-final-and' to implement many-to-many word alignments. The maximal extracted phrase length is 7 and the reordering model selected as the variate in various experiments. In process of tuning parameters, we use MERT method to optimize arguments. In addition, we use SRILM to training two 5-gram language models on each Chinese corpus and estimate parameters according to Kneser-Ney smoothing algorithm. The evaluation metric of machine translation is case-insensitive BLEU-4 scores .

In attention-based NMT model, we use GROUNDHOG on the parallel corpus to train the NMT model. And our approach modified NMT by adding a pre-process of replacing Chinese words and a post-process of replacing or retranslating the OOVs after decoding. We set the vocabulary size to 30K for each language, the beam size for reaching is 10. And the other settings are the default of RNN-RNN model in (Bahdanau et al.,2014).

In Adversarial-NMT, the structure of the NMT model G is the same as the attention-based NMT model (Bahdanau et al.,2014). For the adversary D, the CNN consists of these parts: two pooling layers, one MLP layer and one softmax layer. For the training of NMT model G, similar as previous works (Shen et al., 2016; Tu et al., 2016a), we warm start G from a well-trained attention-based model, and optimize it using vanilla SGD algorithm with mini-batch size 64 for English-Chinese translation and 32 for Uyghur-Chinese translation. The other settings are the same with (Wu et.al 2017).

V. Translation Results and Analysis

Table 2 demonstrated the experiment results of English-Chinese and Uyghur-Chinese Machine Translation system, respectively. 'En-Ch' in table 2 denotes English-Chinese translation, 'Uy-Ch' denotes Uyghur-Chinese translation, the score in table 2 is BLEU4. According to the results of experiments, we can draw following conclusions.

TABLE II. THE RESULT OF EXPERIMENTS ON MACHINE TRANSLATION

system	modifying	En-Ch	Uy-Ch
Phrase-based MT	No	30.12	34.12
	Yes	28.34	33.76
Attention-based NMT	No	31.23	25.23
	Yes	31.57	25.88
Adversarial-NMT	No	32.12	26.48
	Yes	32.87	27.34

The performance of neural machine translation system get improved by applying our mixed word/character model, especially Adversarial-NMT. Compared with the baseline of phrase-based MT, our improved Adversarial-NMT can get a better translation result in English-Chinese machine translation , and a nearly translation result in Uyghur-Chinese machine translation. Compared with the attention-based NMT model, our improved Adversarial-NMT can get a better translation result both in Uyghur-Chinese and English-Chinese machine translation.

The reasons of the big difference of Adversarial-NMT between English-Chinese and Uyghur-Chinese should include these parts. First, the corpus on English-Chinese we selected is open-domain, while the corpus on Uyghur-Chinese we selected is limited-domain of news. Experiments in WMT2016 shows that NMT can get the better results in open-domain translation. Second, performance of NMT is quite different in different language-pair. Even though NMT should have a better performance in Uyghur-Chinese than phrase-based model in theory, as Uyghur-Chinese have a bigger gap than English-Chinese and NMT is good at translating language pair with big difference. Third, the scale of English-Chinese corpus is much bigger than the Uyghur-Chinese corpus. This should be the vital reason as the NMT is heavily rely on big corpus.

Our improved Adversarial-NMT can get the state-of-the-art result in English-Chinese translation as the OOVs problem has been relieved. The mixed model not only can works on attention-based NMT, but also works on Adversarial-NMT. As the better learning ability, Adversarial-NMT get a better improvement than attention-based NMT when applying mixed model. With Integrating extra knowledge, improved Adversarial-NMT generates better result than original model.

VI. Conclusions and Future Work

In this paper, we have proposed an improved Adversarial-NMT integrating extra knowledge. We introduce bilingual dictionaries to Adversarial-NMT by a mixed word/character model. Experiments on public corpus show that our proposed methods can achieve remarkable improvement on the translation quality of Adversarial NMT.

Nowadays, many research works treat neural machine translation model as a feature function on phrase-based machine translation. And many of them got great results. While in this paper, we simply use the extra bilingual dictionaries to Adversarial-NMT. Our next step is treating phrase-based MT model as an input in the last softmax layer of NMT.

Acknowledgments. This work is supported by the Xinjiang Fun under Grant (No.2015KL031), the Natural Science Foundation of Xinjiang (No.2015211B034) and the Xinjiang Science and Technology Major Project (No.2016A03007-3) .

References

[1] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units[J]. arXiv preprint arXiv:1508.07909, 2015.

[2] Shi X, Padhi I, Knight K. Does String-Based Neural MT Learn Source Syntax?[C]//Proc. of EMNLP. 2016.

[3] Zhang J, Zong C. Bridging Neural Machine Translation and Bilingual Dictionaries[J]. arXiv preprint arXiv:1610.07272, 2016.

[4] Wu L, Xia Y, Zhao L, et al. Adversarial Neural Machine Translation[J]. arXiv preprint arXiv:1704.06933, 2017.

[5] Wang, X, Lu, Z, Tu, Z, et al. Neural Machine Translation Advised by Statistical Machine Translation[J]. arxiv, 2016.

[6] Sankaran B, Mi H, Al-Onaizan Y, et al. Temporal Attention Model for Neural Machine Translation[J]. arxiv, 2016.

[7] Yang Z, Hu Z, Deng Y, et al. Neural Machine Translation with Recurrent Attention Modeling[J]. arxiv, 2016.

[8] Tu Z, Lu Z, Liu Y, et al. Modeling coverage for neural machine translation[J]. arXivpreprint arXiv:1601.04811, 2016.

[9] Zhang B, Xiong D, Su J. Recurrent Neural Machine Translation[J]. arxiv, 2016.

[10] Shen S, Cheng Y, He Z, et al. Minimum risk training for neural machine translation[J]. arXiv preprint arXiv:1512.02433, 2015.

[11] Ling W, Trancoso I, Dyer C, et al. Character-based Neural Machine Translation[J] . arxiv, 2015.

[12] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with SubwordUnits[C]. ACL, 2016.

[13] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[14] Yin W, Schütze H, Xiang B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs[J]. arXiv preprint arXiv:1512.05193, 2015.

[15] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences[C]//Advances in neural information processing systems. 2014: 2042-2050.

[16] Papineni K, Roukos S, Ward T, Zhu W.BLEU: a method for automatic evaluation of machine translation[C]. Proceedings of the 40th annual meeting on association for computational linguistics. 2002: 311-318..