Knowledge Distillation for Machine Translation

Zhen Li[a], Dan Qu[b], Chaojie Xie[c], Xuejuan Wei[d]

National Digital Switching System Engineering & Technological R&D Center P.R. China
[a]li_zhenjojo@sina.com, [b]qudanqudan@sina.com, [c]569889194@qq.com, [d]wei_xuejuan@126.com

Keywords: Neural Machine Translation (NMT), Convolutional Neural Network (CNN), knowledge distillation, encoder-decoder, low-resource

Abstract: Encoder-to-Decoder is a newly architecture for Neural Machine Translation (NMT). Convolutional Neural Network (CNN) based on this framework has gained significant success in NMT task. Challenges remain in the practical use of CNN model, which is in need of bilingual sentence pairs for training and each bilingual data is designed for CNN translation model needing retraining. Although some successful performance has been reported, it is an important research direction to avoid model overfitting caused by the scarcity of parallel corpus. The paper introduces a simple and efficient knowledge distillation method for regularization to solve CNN training overfitting problems by transferring the knowledge of source model to adapted model on low-resource languages in NMT task. The experiment on English-Czech dataset result shows that our model solve the over fitting problem, get better generalization, and improve the performance of a low-resource languages translation task.

1. Introduction

The Encoder-Decoder framework based neural network models can significantly improve the performance of Neural Machine Translation systems in large data scenario. In NMT task, a particular type of neural network model, known as Convolutional Neural Networks (CNN), have demonstrated state-of-the-art performance. An obvious advantage of CNN compared to other models, such as Recurrent Neural Network (RNN) is that CNN models can parallelize each neuron in training and make it easier to optimize. Though CNN models are very powerful, need large amounts of substantially on the availability of sizeable parallel corpora to train them. Usually, if there is not enough data set (training set) to constrain neural networks model (CNN) with vast variables, then it will happen over fitting problems and fail to generalize to new example. In our NMT case, the generalization refers to the ability of a CNN model to be applied to a new bilingual sentence pairs, which is not present in the training set. As a result, CNN will fall short of reaching state-of-the-art performances for these not available on limit language pairs.

In our paper, we focus on Knowledge Distillation method for regularization avoids over fitting of the adapted model trained on limit language pair by transferring the knowledge of source model to

adapted model. It would be nice to generalize to the new limit bilingual sentence pairs without collecting parallel data and avoiding the overfitting.

The experiments on NMT task with the IWSLT16 database verified that Knowledge distillation as the regularization method can improve CNN training and BLEU scores across a range of Low-Resource language pair. As far as we know, we are the first to use Knowledge Distillation method to the NMT task.

The paper contains the following parts: In section 2 we briefly survey the method of CNN Model for Machine Translation. Furthermore, in section 3, we show the Knowledge Distillation method in detail for Machine Translation. Section 4, we introduce our experiment for NMT. Finally, in section 5 we conclude the paper and discuss of future works.

2. CNN Model for Machine Translation

CNN allows parallelization over every element in a sequence, which shows competitive performance in machine translation [24][1]. In the paper [24], Facebook applied an architecture based entirely on convolutional neural networks which is typically implemented with encoder-decoder framework. Latterly, Facebook used muti-hop attention and gated linear units to get better translations performance [1]. Muti-hop attention [23] is the enhanced version of this mechanism, which allows the network to perform several such "reviews" to produce better translations. Gated linear units [1] will control which information is passed to the next unit so that better translation can be produced. The goalkeeper allows it to magnify a particular aspect of the translation depending on what the network considers appropriate in the current context. The experiments on WMT'14 English-French translation show that the convolution model is faster than GNMT [25] in processing speed of one order of magnitude, both on GPU and CPU. Our research based on the setup of Gehring et al.[1] as a baseline.

3. Knowledge Distillation for NMT

3.1 Domain adaptation

Usually, people tend to design more complex networks to collect more data in order to get better performance when solving problems with neural networks. However, the complexity of the model increases rapidly with the larger size of the modules that the hardware resources (memory and GPU) are getting higher. Recent research found that knowledge adaption techniques could be tackle this bottleneck.

Domain adaptation is an important research field of Transfer learning. Transfer learning has been inspired by human learning ways to transfer knowledge between tasks. That is, we apply improvement learning in a new task through the transfer of knowledge from a related task that has already been learned. While, some work in Transfer learning is in the context of natural language processing [12], such as text classification, speech recognition, sentiment analysis and neural machine translation [13], typically in Low-Resource [14].Recent progress in NMT, interesting research have been happened in Low-Resource domain adaption [2,3].Domain adaptation techniques have been successfully applied to (Automatic Speech Recognition) ASR for the adaptation of DNN acoustic models [4-9].

For many tasks, it has been observed that CNN/RNN-based systems that have been adapted to the limited training scenario are more accurate than unadapted systems, indicating that the adaptation of neural networks models is an important topic that merits investigation.

3.2 Knowledge distillation

In order to solve the over fitting problem, several adaptive approaches of neural network models have been investigated, such as limit the number of parameters [16,17], expanding the input features [18,19] and regularization in optimization [20,21].Regularization contain L2(weight decay)and KLD (Kullback-Leibler).Hinton et al. [6] pointed out that the original cumbersome teacher models can be utilized to improve the performance of small student models by a transfer learning method, knowledge distillation.

Knowledge distillation was originally proposed for model compression [11]. Though "distillation" to transfer the knowledge from the complex model to a small model. We note that our work differs from previous work which has mainly explored compressing deep learning models [22]. In this work, we investigate knowledge distillation as regularization approach in the context of CNN translation model, which is a different kind of training. A large CNN translation model achieves higher accuracy but requires a large number of corpora as the teacher model, and a small model is called the student model, with only limited parallel corpus pairs, and the use of the teacher model's output training to achieve the performance of the teacher's model. This framework can be enhanced generalizability of the student model by using a temperature parameter to control the smoothness.

3.3 Knowledge distillation for CNN model training

In the NMT (CNN) model adaptation task, we have High-resource of bilingual data and a small amount of bilingual data in target domain. Our goal is to obtain an adapted model, which matches the target domain data that solve the over fitting problem and get better generalization.

Knowledge distillation trains the student model to imitate the teacher model. From this viewpoints, knowledge distillation as a regularization that constrains the student model, which can be applied to model adaption. We breakdown the way this model works in these steps:

Get embedding encoder matrix $U \in R^{|V|\times n}$ for trained CNN model on High-resource bilingual data(English-France).Then based on that input matrix $U \in R^{|V|\times n}$ representation for trained CNN model on the Low-resource bilingual data (English-Czech)as new target CNN model called teacher model, which generate the conditional probability $\log q(y/x)$, q_i is the tempered softmax probability of the *i*-th class of the teacher model.

Use domain adaption approach to trained a student model through imitate the teacher model by using the encoder's output of the teacher model. The student model generate the conditional probability $\log p(y/x)$, p_i is output softmax probability of the *i*-th class of the student model. Then use the information theory cross-entropy $H_{\text{soft}}(p,q)$ to measure of the distance between teacher model and student model distributions.

Desire conditional probabilities p generated by student model, to match the true probabilities y. Use the information theory cross-entropy $H_{\text{hard}}(p,y)$ to measure of the distance between true probabilities and student model distributions.

The student model is trained to minimize following loss function

$$L = (1-\lambda) H_{\text{hard}}(p,y) + \lambda H_{\text{soft}}(p,q) \tag{1}$$

parameter λ [0,1] is the weight of the hard and soft cross entry losses.

q is a *K(*the number of output classes*)* dimensional vector, and q_i is computed as follows:

$$q_i = \frac{\exp\left(z_i\left(x\right)/T\right)}{\sum_{j=1}^{K}\exp\left(z_i\left(x\right)/T\right)} \tag{2}$$

Where, x is an input feature, $z_i\left(x\right)$ is pre-softmax output of the i-th class of the teacher model, T is a temperature that is normally set to 1. Using a higher value for T produces a softer probability distribution over classes[26], and q approaches a uniform distribution. when T is changed, $1/T^2$ should be multiplied to the gradient of the second term of the loss function L in backpropagation. This keeps the balance between the contribution of H_{hard} and H_{soft}. Temperature T allows us to control the importance of the class similarity information in training. When T is set larger than 1, small probabilities of non-target classes are emphasized and the class similarity information is more strongly learned by the student model[20].

3.4 Training Details

Our KDCT(Knowledge Distillation convolutional model for translation)model takes monolingual word embeddings as input. We train the CNN with default hyperparameters in word2vec.In the translation model adaptation task, we have a trained teacher model and a small amount of target domain data(the student model).The schematic diagram of KDCT model is shown in Fig.1.

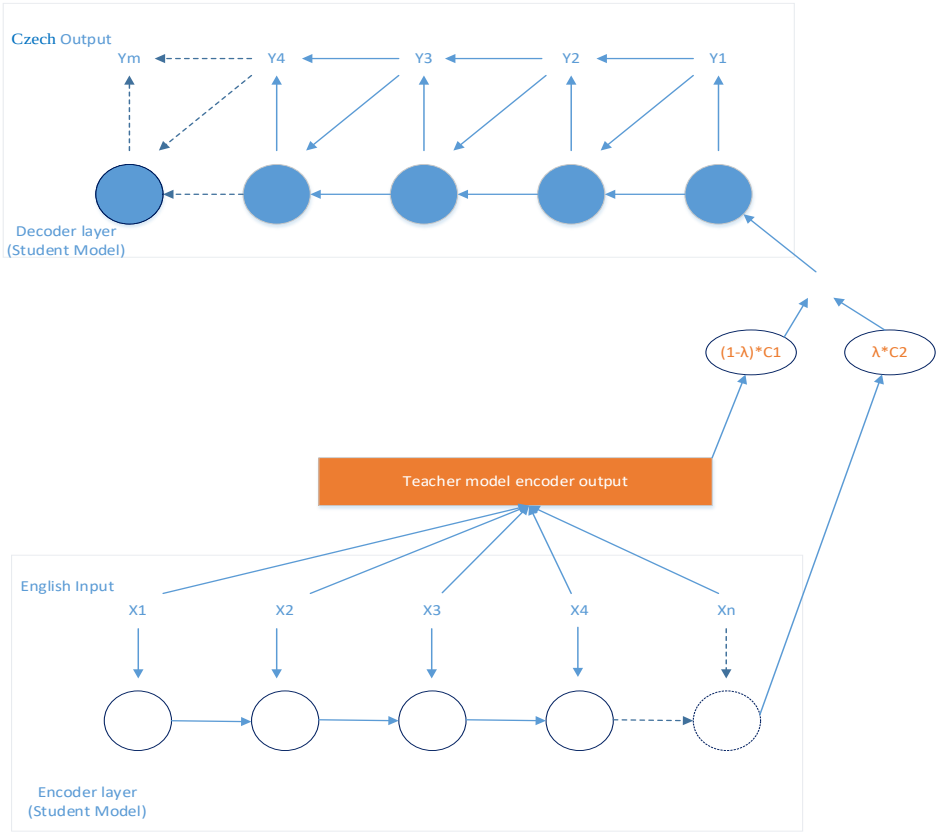

Fig.1 The schematic diagram of KDCT model

The KDCT step is conducted as follows:

The English source sentence is the input block. The C1 block as a part of encoder representations. It is dot products between the teacher model's encoder weight matrix and input.

The C2 is dot products between the student model and input. $(1-\lambda)C_1+\lambda C_2$, as the input of the decoder layer.

Modify the λ and the softmax layer's T value to achieve the best performance in training.

Define the encoder convolution layer as follows:

$$h_0 = f_0\left(conv\left(W_0; X_0\right)+b_0\right) \tag{3}$$

$$h_i = f_i\left(conv\left(W_i; h_{i-1}\right)+b_i\right) \tag{4}$$

$$C1 = f_t\left(conv\left(W_t; X\right)+b_t\right) \tag{5}$$

$$C2 = f_s\left(conv\left(W_s; X\right)+b_s\right) \tag{6}$$

Where $X \in \mathbb{R}^{d\times\bar{k}}$. f_i is a non-linear function, i=1,2,...,*n*. $conv\left(W; X\right)$ is the convolution of the input layer *X*. We input to the teacher model weight matrix W_t and get the encoder layer output $C1$.

Plus the initialized encoder layer output of the student model and achieve the final output: $h_m = \left(1-\lambda\right)C1+\lambda C2$ and input to the decoder layer: $Y_m = W_m h_m + b_m$.

4. Experiments

4.1 Data

Our experiments are centered around translation task. See the summary in Table 1. We experiment with two types of corpora:

(1) a large corpus – In the teacher model, we use the WMT'14 English-French dataset. Following Gehring et al.[1], we use the full training set of 36M sentence pairs, and remove sentences longer than 175words as well as pairs with a source/target length ratio exceeding1.5. This results in 35.5M sentence-pairs for training, 27K sentence-pairs for validation. Results are reported on news test 2014.We use a source and target vocabulary with 40K BPE types.

(2) a small corpus – In the student model, we use the IWSLT16 English-Czech dataset. We use the same setup as Luong et al.[27]which comprises 122k sentence pairs for training, 1% of the training dataset for validation and we test on newstest2014. As vocabulary we use 40K sub-word tokens based on byte-pair encoding.

Table 1 Data– Information about the different datasets used in this work. Foreach task, we display the following statistics: (a) Train Size: the number of training sentences, (b) Valid Size: the sizes of the validation, (c)Vocab Size : the sizes of the vocabulary, (d) Test Set: the test set.

Model	Task	Train Size	Valid Size	Vocab Size	Test Set
Teacher	WMT'14 English-French	35.5M	27k	40K BPE	newstest2014
Student	IWSLT16 en-cs	122k	1K	4K BEP	newstest2014

4.2 Model Parameters and Optimization

When training the teacher model, both encoder and decoder are initialized with 15convolutional layers. All embeddings, including the output produced by the decoder before the final linear layer, have dimensionality 768; we use the same dimensionalities for linear layers mapping between the hidden and embedding sizes. We train our convolutional models with Nesterov's accelerated gradient method using a momentum value of 0.99 and renormalize gradients if their norm exceeds 0.1. We use a learning rate of 0.25 and once the validation perplexity(cross-entropy) stops improving, we reduce the learning rate by an order of magnitude after each epoch until it falls below 10^{-4}.The student model uses the same network structure as the teacher model with 15 convolutional layers. The source language of encoder is English. Therefore, the teacher model can be used to obtain the corresponding output sequence and the output sequence can be regularized and brought into the student model for training.

Unless otherwise stated, we use a mini-batch size of 2000.The max words of a sentence is 1024. We apply dropout to encoder and decoder. In order to keep the timing information of sentence, we apply position-embedding to the input of the encoder and the attention mechanism is used for supervised learning in the decoder. All models are implemented by Py-Torch. We use eight Nvidia K80 GPU setup on a single machine for training.

4.3 Evaluation

The beam-search is used for the results of translation and the size of beam is 5. Unrecognized words are replaced. Finally, we evaluate the performance of model on BLEU scores.

4.4 Results

We first evaluate convolutional model on two translation tasks.

On IWSLT16 English-Czech translation, our encoder and decoder use the same network structure and both have 15 layers, the first nine layers use 512 hidden units and subsequent four layers use 1024 units, all using kernel width 3. The last two layers have 2048 units which are just linear mappings with a single input. We trained this model on a multiple GPUs over a period of 30 days with a max tokens of 4000 a batch. 16.2BLEU is our best run.

Another test is on the larger WMT'14 English-French datasets. The conv_seq2seq model for this experiment uses 15 layers both in the encoder and decoder, the first 6 layers use 512 hidden units, the subsequent 4 layers use 768 hidden units, the next 3 layers use 1024 units, all using kernel width 3; the last 2 layers have 2048 units and 4096 units each but the they are linear mappings with kernel width 1. This model has an effective context size of only 50 words, beyond which it cannot access any information on the target size. We trained the model with batch size 32 on each worker and use 8 GPUs for about 50 days. The best result is 40.5 BLEU.

Finally, we use our KDCT model to train on the IWSLT16 English- Czech again and use the same network structure. The difference is that we add the distillation parameter on the encoder representations. The results are shown in table 2. It can be observed that when λ=0.1 and T=3, the distillation-based model can achieve the best performance. KDCT model appear beneficial even in this challenging（Low-Resource of bilingual data） set-up: we obtain best performance +1.2 BLEU point improvements from using Knowledge Distillation method of CNNs for English-Czech.

Table 2 Results of translation. λ and T are the weight and the temperature parameters of knowledge distillation. Note that λ=0.0 means use of baseline in different conditions. The best results in each column are highlighted in bold.

WMT'16 English-Czech distillation				
layers	Kernel width	λ	T	BLEU
8	3	0.0	-	10.3
		0.1	3	***10.4***
		0.1	5	8.9
		0.3	3	10.0
		0.3	5	8.2
8	5	0.0	-	10.9
		0.1	3	***11.1***
		0.1	5	9.4
		0.3	3	10.3
		0.3	5	8.9
15	3	0.0	-	16.2
		0.1	3	***16.4***
		0.1	5	16.37
		0.3	3	16.5
		0.3	5	14.9
15	5	0.0	-	16.1
		0.1	3	***17.4***
		0.1	5	16.7
		0.3	3	15.9
		0.3	5	17.0

5. Conclusion

In this paper, we introduces a simple and efficient Knowledge Distillation method to solve CNN training model overfitting problem on Low-Resource languages NMT task. Our KDCT model improve the performance of a low-Resource languages translation task. The experiment show that there is still room for improvement in the future.

Acknowledgements

This work was supported by the Group of Artificial Intelligence. We thank each member of our team for assistance with training models and helpful discussions.

This work was supported by the National Natural Science Foundation of China (61673395,61403415); The Natural Science Foundation of Henan(162300410331).

References

[1] Jonas Gehring, Michael Auli, et al. Convolutional Sequence to Sequence Learning. arXiv:1705.03122, 2017

[2] Zoph Barret, Yuret Deniz, et al. Transfer Learning for Low-Resource Neural Machine Translation. arXiv:1604.02201, 2016

[3] M,Fadaee, A,Biszazza, et al. Data Augmentation for Low-Resource Neural Machine Translation. arXiv:1705.00440, 2017

[4] P. Swietojanski, S. Renals. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. in Proc. of SLT, 2014,pp. 171–176.
[5] T. Ochiai, S. Matsuda, et al. Speaker adaptive training for deep neural networks embedding linear transformation net-works. in Proc. of ICASSP, 2015, pp. 4605–4609.
[6] G. Saon, H. Soltau, et al. Speaker adaptation of neural network acoustic models using i-vectors. in Proc. of ASRU, 2013, pp. 55–59.
[7] M. Delcroix, K. Kinoshita, et al. Context adaptive neural net-work for rapid adaptation of deep CNN based acoustic models. in Proc. of INTERSPEECH, 2016, pp. 1573–1577.
[8] O. Abdel-Hamid, H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code. in Proc. Of ICASSP, 2013, pp. 7942–7946.
[9] H. Liao. Speaker adaptation of context dependent deep neural networks. in Proc. of ICASSP, 2013, pp. 7947–7951.
[10] G. Hinton, O. Vinyals, et al. Distilling the knowledge in a neural network. in Proc. of NIPS DeepLearning and Representation Learning Workshop, 2014.
[11] C.Bucilu, R.Caruana, et al. Model compression. in Proceedings of the 12thACM SIGKDD international conference on Knowledge discovery and data mining.ACM,2006,PP.535-541.
[12] D.Wang and T. Fang. Transfer learning for speech and language processing. arXiv:1511.06066.
[13] B. Zoph, et al. Transfer Learning for Low-Resource Neural Machine Translation. 2016,arXiv:1604.02201.
[14] S.J.Pan, Q.Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering,22(10):1345-1359.
[15] L.Torry, J. Shavlik. Transfer Learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Global.
[16] P. Swietojanski, S. Renals. Learning hidden unit contributions for unsupervised speaker adaptation ofneural network acoustic models. in Proc. of SLT, 2014,pp. 171–176.
[17] T. Ochiai, S. Matsuda, et al. Speaker adaptive training for deep neural networks embedding linear transformation net-works. in Proc. of ICASSP, 2015, pp. 4605–4609.
[18] G. Saon, H. Soltau, et al. Speaker adaptation of neural network acoustic models using i-vectors," in Proc. of ASRU, 2013, pp. 55–59.
[19] M. Delcroix, K. Kinoshita, et al. Context adaptive neural net-work for rapid adaptation of deep CNN based acoustic models. in Proc. of INTERSPEECH, 2016, pp. 1573–1577.
[20] Taichi Asami, Ryo Masumura, et al. Domain Adaptation of DNN Acoustic Models Using Knowledge Distillation. in ICASSP 2017, 978-1-5090-4117-6/17
[21] Dong Wang, Chao Liu, et al. Recurrent Neural Network Training with Dark Knowledge Transfer. in Journal of Latex Class Files, 2014 ,VOL. 13, NO. 9.
[22] Y. Kim , AM. Rush.Sequence-Level Knowledge Distillation 2015, arXiv:1505.04630.
[23] Sukhbaatar, Sainbayar, et al. End-to-end Memory Networks. 2015, In Proc.of NIPS, pp. 2440–2448.
[24] Gehring J., Auli M., Grangier D., et al. A Convolutional Encoder Model for Neural Machine Translation. arXiv:1611.02344, 2016
[25] Wu Y., Schuster M., et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv:1609.08144, 2016
[26] G. Hinton, O. Vinyals, et al. Distilling the Knowledge in a Neural Network. arXiv:1503.02531, 2015
[27] Luong, Minh-Thang, Pham, Hieu, et al. Effective Approaches to Attention-based Neural Machine Translation. In Proc. of EMNLP, 2015.