Towards better translation performance on spoken language

Chao Bei, Hao Zong

Global Tone Communication Technology Co.,Ltd.

{beichao,zonghao}@gtcom.com.cn

Abstract

In this paper, we describe GTCOM's neural machine translation(NMT) systems for the International Workshop on Spoken Language Translation(IWSLT) 2017. We participated in the English-to-Chinese and Chinese-to-English tracks in the small data condition of the bilingual task and the zero-shot condition of the multilingual task. Our systems are based on the encoder-decoder architecture with attention mechanism. We build byte pair encoding (BPE) models in parallel data and back-translated monolingual training data provided in the small data condition. Other techniques we explored in our system include two deep architectures, layer nomalization, weight normalization and training models with annealing Adam, etc. The official scores of English-to-Chinese, Chinese-to-English are 28.13 and 21.35 on test set 2016 and 28.30 and 22.16 on test set 2017. The official scores on German-to-Dutch, Dutch-to-German, Italian-to-Romanian and Romanian-to-Italian are 19.59, 17.95, 18.62 and 20.39 respectively.

1. Introduction

This paper describes the submission of the Global Tone Communication Technology Co., Ltd. (GTCOM) for the first participation in IWSLT evaluation. We participated in the zero-shot condition in the multilingual task and the English-to-Chinese and Chinese-to-English tracks in the small data condition of the bilingual task. Our neural machine translation systems are developed as encoder-decoder architecture [1] with attention mechanism [2] and the experiment toolkit we used in the evaluation is Nematus [3].

The intuition of this participation is to verify whether the model architechture and techniques we applied in our generic system [1] with large training data is also effictive in spoken language domian with small training data. In bilingual task, since the training data is very small in both Chinese-to-English and English-to-Chinese directions, Chinese word segmentation, tokenization, binary pair encoder(BPE), different size of hidden layer, deep transition model and back-translation are involved in our experiments. In multilingual task, we uesd different pre-processing strategy and annealing Adam to enhance the translation performance.

This paper is arranged as follows. We firstly describe the task, including the data size and evaluation method. Then we introduce the techniques used in our system. After that, we present the experiments for the two task, including data pre-processing and model architecture. Finally, we analysis the experiment results and draw the conclusions.

2. Task Description

The task focuses on bilingual and multilingual text translation in spoken language domain; the provided data is mainly collected form TED talks. We participated in Chinese-to-English and English-to-Chinese directions of the bilingual task, as well as zero-shot translation of the multilingual task.

2.1. Bilingual task

For the bilingual task, we focused on Chinese-to-English and English-to-Chinese directions of the small data condition, which only the in-domain training and development data is allowed to use. The detail information about the data is shown in Table 1. In addition, Chinese texts were evaluated at character level. Before evaluation, texts are splitted into Chinese characters, but sequences of non-Chinese characters are kept as they are.

2.2. Multilingual task

For multilingual task, we focused on zero-shot translation which using one model to translate any pair between English, Dutch, German, Italian and Romanian trained with the in-domain training and development data. In addition, training data synthesis from other pair and pivoting are allowed as contrastive conditions. But the directions, which included Dutch-to-German, German-to-Dutch, Italian-to-Romanian and Romanian-to-Italian, must be excluded from the training and development sets. The statistic of the parallel data is shown in Table 2.

3. Methology

This section introduces the techniques we used in our systems.

[1] Our generic translation system covers 10 languages and is available at http://translateport.yeekit.com:4305/index.html

Table 1: *Number of sentences summary for in-domain training and development data for bilingual task.*

NMT direction	training data	development data 2013 2014 2015	monolingual data(target)
en-zh	231K	1,372 1,297 1,205	520K
zh-en	231K	1,372 1,297 1,205	234K

Table 2: *Number of sentences summary for in-domain training and development data for zero-shot multilingual task.*

language	de-en	de-it	de-ro	en-it	en-nl	en-ro	it-nl	nl-ro
training data	204K	203K	200K	230K	236K	219K	232K	205K
development set	1,138	1,133	1,121	1,147	1,181	1,129	1,183	1,123

3.1. Layer normalization and weight normalization

Layer normalization [4] is helpful to accelerate the convergence of model and improve the performance. [5] showed layer normalization is very effective in neural machine translation, especially with deep model. It is known that deep model for neural machine translation is difficult to converge. Weight normalization [6] is another method to accelerate the convergence and improve the performance, especially for recurrent models. Therefore, we used layer normalization in all the models and explore whether weight normalization play a further role on the models with layer normalization in neural machine translation.

3.2. Subword segmentation

To avoid unknow words, we used BPE-based splitting algorithm [7] to segment the word sequence to subword units sequence. This algorithm iteratively merges the most frquent pair of symbols into a single symbol. Therefore, the most frequent words in the corpus remain intact while the rare words are segmented into subunits. Joint BPE were used for the zero-shot condition, while we trained two separate BPE models for bilingual task due to different alphabet shared.

3.3. Back-translation

Monolingual in-domain data is also important for small training data condition. Monolingual data was back-translated with a shallow model trained with parallel data from target to source [8]. So we get translated source text and in-domain target text as synthetic parallel data. Then we mixed synthetic data and provided parallel data together to train our model.

3.4. Deep model

Deep model always gets better performance but is harder to converge. We use two architectures, stacked model [9] and deep transition model [10], which has been used in WMT 2017 by [5]. Even though the data size in [5] is larger than this task whose parallel data size is only 231K, deep model was still used to explore the adaptation on small data condition.

3.5. Annealing Adam

A strong baseline [11] gives a training trick, annealing Adam, which is significantly faster than SGD with annealing and obtains better performance. Adam [12] is an optimization algorithm, which applies momentum on a per-parameter basis and automatically adapts step size subject to a user-specified maximum. It speeds up the convergence and is a popular choice for researches. However, the models with Adam are slightly underperform compared to annealing SGD [13]. Thus, we halved learning rate after early stop and trained from the previous best model. We did this operation twice.

4. Experiment setup

4.1. Bilingual task

In this small data condition, we trained our systems using the in-domain data sets. Althrough, Chinese texts are evaluated at character level, we used Jieba [14], a Chinese word segmentation tool, to segment Chinese text in both parallel data and monolingual data. For English text, tokenizer and truecase in Moses [15] toolkit were applid. We applied BPE on both tokenized Chinese and English text. Before that, we calculated the word frequency on the training data and then get the number of words whose frequency is larger 10. Thus, the merge operation is calculate as
$N_{operation} = number\ of\ words(word\ frequency > 10)$
In our experiments , merge operation for English is set to 18000 and to 20000 for Chinese.

We used a 2-layer model trained with in-domain parallel data to translate the monolingual data as synthetic parallel data and mixed it with real parallel data. Translating Enlish monolingual data and Chinese monolingual data took about 4 days.

Our neural machine transition system is an encoder-decoder leverage GRU [16] cell in each layer with attention mechanism. The main model configuration is shown in Table 3. The mini-batches size is set to 64. The models were optimized using Adam with initial learning rate 0.0001 dur-

Table 3: *Model configuration for bilingual task.*

Type	value
English vocabulary size	19623
Chinese vocabulary size	25377
word embedding	512
hidden units	1024
embedding dropout	0.2
hidden dropout	0.2
source dropout	0.1
target dropout	0.1
layer normalization	True
maximum sentence length	100

ing training procedure, we also shuffed the training data after each epoch. For decoding we set the beam size to 10. In general, we trained 4-layer model and deep transition model with transition depth 4 for real parallel data and synthetic parallel data. Beside, the right-to-left model [17] with 4-layer architecture and deep transition architecture respectively were trained to rerank the n-best-list. It[17] showed a complementary target context will be seen at each time step and therefore the expected averaged probabilities will be more robust. In detail, We increase the size of the n-best-list to 50 for the reranking experiments.

4.2. Zero-shot condition in multilingual task

Different from bilingual task, in this zero-shot condition, the training data set consists of in-domain data from any pair between in English, Dutch, German, Italian and Romanian, except German-to-Dutch, Dutch-to-German, Italian-to-Romanian and Romanian-to-Italian data. We applied tokenizer and truecase script in Moses toolkit to preprocess all the corpora.

Zero-shot model aims to translate different langauage directions using the same model. Therefore, BPE segmentation is more useful than bilingual task. It can not only reduce the vocabulary size but also reduce the unknown words drastically. The merge operation of joint BPE model is 39500.

At the end of pre-processing, we add a label which consists of source language label and target language label at the start of each source sentence according [18]. Our processing for the language label is slightly different from [18]. And the model can translate from one specified source language to another specified target language learned from this label, although the model architecture didn't change.

Similar to bilingual task, the main model configuration is shown in Table 4. The mini-batch size is set to 80. And models were trained with Adam with initial learning rate 0.0001, the training data will be shuffed during each epoch. The Beam size in decoding is set to 10. We generally trained shallow model and deep transition model whose transition depth is 4 for all in-domain data. Beside, the right-to-left model with shallow model and deep transition architecture respectively were trained to rerank the n-best-list, which is the same in bilingual task.

Table 4: *Model configuration for multilingual task.*

Type	value
Source vocabulary size	40000
target vocabulary size	40000
word embedding	512
hidden units	1024
embedding dropout	0.2
hidden dropout	0.2
source dropout	0.1
target dropout	0.1
layer normalization	True
maximum sentence length	80

Table 5: *Results on Official Test Sets for binglingual task.*

direction	tst2016	tst2017
en-zh	28.13	28.30
zh-en	21.35	22.16

5. Result and analysis

5.1. Results of bilingual task

Table 6 shows the case-insensitive BLEU score in development set of Chinese-to-English and Table 7 is for English-to-Chinese. We observed the improvement of 0-0.81 BLEU score from annealing Adam training trick and 0 to 0.88 BLEU score from training with a mix of parallel and synthetic data. But we find a fluctuation of -0.57 to 0.81 BLEU score from weight normalization especially in deep transition model. Weight normalization is not robust based on layer normalization in this condition. Ensembling of the independent models gives further imporvement by 0.97-1.28 BLEU score. Finally, our submitted system was reranked by right-to-left models with 50 n-best-list output of ensembling decoding of left-to-right models. This improved 0.3 to 0.55 BLEU score. Table 5 shows the official test results.

5.2. Results of multilingual task

Table 8 shows the case-insensitive BLEU score for development set of the zero-shot condition. It can be observed that adopting annealing Adam training algorithm also gets improvement of 0.28 to 0.36 BELU points, while weight normalization gets the worse performance. Ensemble decoding improves 1.93 BLEU points, compared shallow model. Then, we found in this condition, right-to-left reranking didn't improve the performance of model. We think that the zero-shot condition is a complex problem, which can translate from multilingual source language to multilingual target language. The model of right-to-left reranking may be hard

Table 6: *Case-insensitive BLEU score in development set of Chinese-to-English in small data condition. WN means weight normalization and SD means synthetic data.*

	tst2013	tst2014	tst2015	average
2 layers	20.32	18.07	21.48	20.03
+ annealing Adam	20.85	18.39	22.04	20.47
4 layers	20.89	17.91	21.87	20.33
+ annealing Adam	20.81	17.91	22.24	20.33
4 layers with WN	20.95	17.99	21.98	20.43
+ annealing Adam	21.24	18.1	21.81	20.48
4 layers with SD	21.05	18.4	21.94	20.49
+ annealing Adam	20.94	18.57	22.41	20.65
4 layers with SD and WN	21.34	18.72	22.5	20.91
+ annealing Adam	21.53	18.72	22.46	20.98
Deep transition	20.68	17.56	21.49	19.97
+ annealing Adam	21.11	17.66	21.64	20.28
Deep transition with WN	20.71	17.98	21.96	20.78
+ annealing Adam	21.40	18.33	22.30	20.80
Deep transition with SD	21.49	18.1	22.40	20.73
+ annealing Adam	21.75	18.83	22.77	21.16
Q Deep transition with SD and WN	21.31	18.78	22.07	20.78
+ annealing Adam	21.86	18.64	22.23	20.97
ensemble	22.83	19.72	23.73	22.13
+ r2l reranking	23.02	19.94	24.26	22.43

Table 7: *Case-insensitive BLEU score in development set of English-to-Chinese in small data condition. WN means weight normalization and SD means synthetic data.*

	tst2013	tst2014	tst2015	average
2 layers	23.71	21.03	26.80	23.83
+ annealing Adam	24.3	21.45	26.69	24.14
4 layers	23.94	21.63	27.34	24.30
+ annealing Adam	24.05	21.90	27.26	24.37
4 layers with WN	24.27	21.61	27.64	24.54
+ annealing Adam	24.46	21.8	27.42	24.54
4 layers with SD	24.43	21.89	28.00	24.74
+ annealing Adam	24.73	21.73	28.14	24.85
4 layers with SD and WN	24.39	21.47	27.61	24.47
+ annealing Adam	24.69	21.69	28.04	24.79
Deep transition	23.83	21.51	27.15	24.13
+ annealing Adam	23.75	21.37	27.06	24.03
Deep transition with WN	23.85	21.77	27.66	23.74
+ annealing Adam	24.21	21.92	27.43	24.49
Deep transition with SD	24.04	21.53	27.43	24.31
+ annealing Adam	24.47	22.1	27.98	24.82
Deep transition with SD and WN	23.7	21.7	26.5	23.74
+ annealing Adam	24.41	21.64	27.65	24.55
ensemble	25.86	23.21	29.41	26.13
+ r2l reranking	26.21	23.61	30.35	26.68

Table 8: *Case-insensitive BLEU score in development set of the zero-shot condition. WN means weight normalization.*

	en-de	en-nl	en-it	en-ro	de-en	de-it	de-ro	nl-en	nl-it	nl-ro	it-de	it-en	it-nl	ro-de	ro-en	ro-nl	average
shallow model	28.29	32.22	29.67	27.56	34.43	20.60	19.47	38.01	22.42	20.79	20.75	34.22	22.1	22.05	35.81	23.15	27.28
+ annealing Adam	28.79	32.70	30.13	28.03	34.46	20.9	19.76	38.27	22.43	21.31	20.85	34.61	22.22	22.26	36.06	23.34	27.56
shallow model with WN	27.68	32.63	29.82	27.32	34.15	20.50	19.36	37.78	21.90	21.15	20.64	34.25	21.87	22.09	35.62	22.58	27.3
+ annealing Adam	27.79	32.56	30.15	27.72	34.42	20.82	19.81	38.03	22.05	20.78	20.29	33.71	22.04	21.63	35.31	22.48	27.05
deep transition	29.43	32.79	30.86	28.96	35.33	21.93	20.54	39.45	23.48	22.13	21.51	35.25	22.99	22.84	37.06	23.29	28.3
+ annealing Adam	29.9	32.85	31.56	28.78	35.72	22.18	20.91	39.79	23.67	22.16	22.20	35.99	23.29	23.16	37.71	23.53	28.66
deep transition with WN	28.85	33.19	30.98	28.37	34.83	22.07	20.28	38.96	23.06	21.83	21.55	35.13	22.86	22.73	37.09	23.63	28.17
ensemble	29.82	34.22	31.98	29.39	36.50	22.8	21.32	40.31	23.84	22.93	22.56	36.15	23.93	23.35	38.05	24.49	29.21
+ r2l reranking	29.60	32.70	31.58	28.77	35.76	22.48	21.45	39.50	24.22	22.74	24.41	35.74	23.76	23.68	37.47	24.61	28.99

Table 9: *Results on Official Test Sets for multilingual task.*

direction	en-de	en-nl	en-it	en-ro	de-en	de-it	de-ro	de-nl	nl-en	nl-it	nl-ro	nl-de	it-de	it-en	it-nl	it-ro	ro-de	ro-en	ro-nl	ro-it
BLEU	23.08	29.08	32.84	23.89	28.04	18.56	16.23	19.59	32.78	21.21	18.11	17.95	18.09	37.84	21.80	18.62	17.95	31.79	20.02	20.39
Nist	5.86	6.81	7.22	5.91	6.85	5.36	4.69	5.57	7.42	5.72	4.97	5.06	5.09	8.10	5.78	5.03	5.06	5.59	5.59	5.57
Ter	60.63	51.46	47.63	58.81	51.41	63.43	69.04	61.26	47.34	60.83	66.55	67.02	67.28	41.05	60.09	65.53	67.02	41.22	67.81	61.11

to converge. In other words, we didn't get a good enough model of right-to-left reranking. Therefore, our submission was the results of ensemble decoding. And the result of the official test set is show in Table 9.

6. Summary

We presented our neural machine transition system for both bilingual task and multilingual task. The intution is mostly coming from the training of our generic translation system and the experiments shows the approaches we applied in our generic model is also effective in spoken langauge domain. Overall, the annealing Adam training algorithm and deep model always get a better performance, while weight normalization is not robust in this experiment. And right-to-left reranking for zero-shot model didn't help.

7. Acknowledgement

This work is supported by 2020 Cognitive Intelligence Research Institute[2] of Global Tone Communication Technology Co., Ltd.[3] We also want to thank Yiming Wang and Xinjie Li for their kindness help.

8. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[3] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. M. Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," *CoRR*, vol. abs/1703.04357, 2017. [Online]. Available: http://arxiv.org/abs/1703.04357

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[5] R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. M. Barone, and P. Williams, "The university of edinburgh's neural mt systems for wmt17," *arXiv preprint arXiv:1708.00726*, 2017.

[6] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *CoRR*, vol. abs/1602.07868, 2016. [Online]. Available: http://arxiv.org/abs/1602.07868

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

[8] ——, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[9] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *CoRR*, vol. abs/1606.04199, 2016. [Online]. Available: http://arxiv.org/abs/1606.04199

[10] A. V. M. Barone, J. Helcl, R. Sennrich, B. Haddow, and A. Birch, "Deep architectures for neural machine translation," *CoRR*, vol. abs/1707.07631, 2017. [Online]. Available: http://arxiv.org/abs/1707.07631

[11] M. J. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *CoRR*, vol. abs/1706.09733, 2017. [Online]. Available: http://arxiv.org/abs/1706.09733

[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[14] J. Sun, "jiebachinese word segmentation tool," 2012.

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[16] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[17] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," *CoRR*, vol. abs/1606.02891, 2016. [Online]. Available: http://arxiv.org/abs/1606.02891

[2] http://www.2020nlp.com/

[3] http://www.gtcom.com.cn/

[18] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *CoRR*, vol. abs/1611.04558, 2016. [Online]. Available: http://arxiv.org/abs/1611.04558