Research and Practice of Simultaneous Speech Translation in Huawei Noah's Ark Lab

Prof. Dr. Qun Liu
Chief Scientist of Speech and Language Computing

Huawei Noah's Ark Lab

10 July 2020, AutoSimTrans Workshop

Content

Content

Background

- Along with the success of deep learning in MT, ASR and TTS, Simultaneous Speech Translation (SST) became commercially feasible:

Microsoft Skype Translator

Google Pixel Buds

Tencent Conference Interpretor

Potential SST Scenarios in Huawei

- Huawei has a very long product line.
- There are many potential scenarios where SST can be applied:

Mobile phones

Conference systems

Customer service systems

SST Research in Huawei

- We started our research on SST in 2019.
- We propose a General SST Framework
- We adopt techniques to adapt our existing ASR and NMT systems to a SST system
- We implement and optimize our SST systems in the following scenarios:
  - Conference speech translator on the cloud
  - Travel assistant on mobile phones

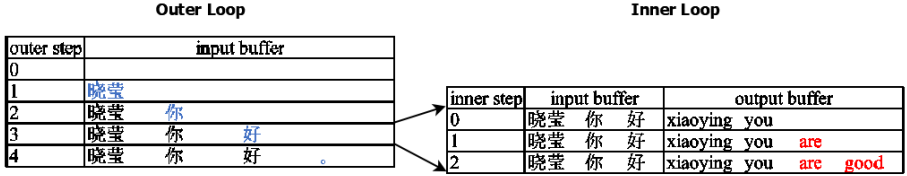
Content

A General Framework for Simultaneous NMT [1]

- Motivation:
  - Adapt an existing NMT system into a simultaneous translation system, without retraining the NMT model.
- Components:
  - An external ASR system
  - A pretrained NMT System
  - An input buffer
  - An output buffer
- Preprint: https://arxiv.org/abs/1911.03154

A General Framework for Simultaneous NMT [1]

- Procedure as two nested loops:
  - Outer loop: update input buffer with ASR streaming output
  - Inner loop: update output buffer with NMT model.
  - The input buffer is updated only when the inner loop stops to update the output buffer.



- Preprint: https://arxiv.org/abs/1911.03154

Key Problems

- How to continue translation
  - Streaming source prefix $\rightarrow$ rebuild encoder hidden states
  - Force decoding with the NMT model
- How to terminate
  - Terminate when predicting the EOS token
  - Terminate when the system is not confident and prefer to wait for more input tokens
    - A Controller is used to determinate when to terminate
    - We define two controllers for this purpose

Length and EOS (LE) Controller

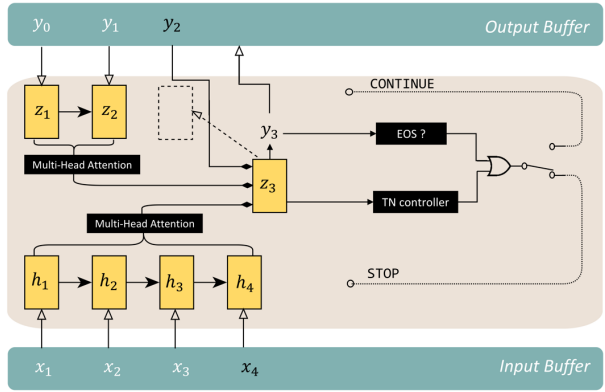
- The inner loop stops to generate output tokens when:
  - A EOS token is generated;
  - The output buffer is full;
  - The delay between the output sequence and the input sequence is smaller than a predefined threshold (e.g. 3 words).

inner step	input buffer	output buffer
0	晓莹 你	
1	晓莹 你	xiaoying
2	晓莹 你	xiaoying you

→ *Stop !*

Trainable (TN) Controller

- The inner loop stops to generate output tokens when:
  - The output buffer is full;
  - The TN Controller output a CONTINUE signal.



TN Controller Training: Policy Gradient

- Fix the NMT model and train the TN controller with policy gradient:

Objective: $$\mathcal{J}_{rl} = \mathbb{E}_{\pi_\phi}(\sum_{t=1}^{T} r_t)$$

Gradient: $$\nabla_\phi \mathcal{J}_{rl} = \mathbb{E}_{\pi_\phi}\left[\sum_{t=1}^{T} R_t \nabla_\phi \log \pi(\sigma_t|\cdot;\phi)\right]$$

where $$R_t = \sum_{k=t}^{T} r_k$$

TN Controller Training: Reward

- The reward function trades off quality and delay:

Reward $r_t = r_t^Q + \alpha \cdot r_t^D$

Quality

$$r_t^Q = \begin{cases} \Delta\text{BLEU}^0(\mathbf{y}, \hat{\mathbf{y}}, t) & t < T \\ \text{BLEU}(\mathbf{y}, \hat{\mathbf{y}}) & t = T \end{cases}$$

where

$$\Delta\text{BLEU}^0(\mathbf{y}, \hat{\mathbf{y}}, t) = \text{BLEU}^0(\mathbf{y}^t, \hat{\mathbf{y}}) - \text{BLEU}^0(\mathbf{y}^{t-1}, \hat{\mathbf{y}})$$

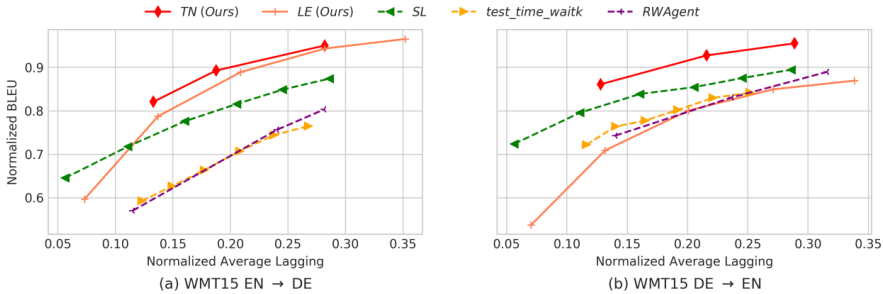
Delay

$$0 < d^{AL}(\mathbf{x}, \mathbf{y}) = \frac{1}{\tau_e} \sum_{\tau=1}^{\tau_e} l(\tau) - \frac{\tau - 1}{\lambda},$$

$$r_t^D = \left\{ \begin{array}{ll} 0 & t < T \\ -\lfloor d^{AL}(\mathbf{x}, \mathbf{y}) - d^* \rfloor_+ & t = T \end{array} \right\}$$

HUAWEI

Experiments



- baselines:
  - test-time-waitk: Ma et al. [2018]
  - SL: Zheng et al. [2019]
  - RWAgent: Gu et al. [2017]

HUAWEI

Content

Content

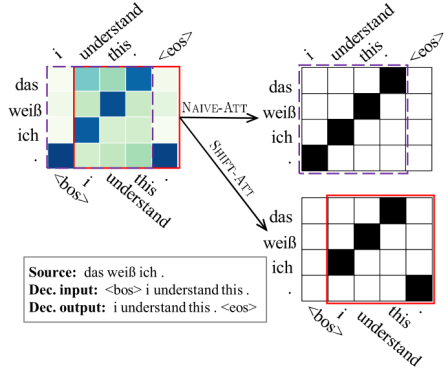
Terminology Translation in NMT

- Terminology (including user-defined terms, entities, acronyms, etc.) translation is crutial in many scenarios.
- It is not a trivial task in NMT because there is no explicit phrase table used in NMT process.
- Solutions:
  - Without word alignment: Constrained Decoding
    - Time consuming or inaccurate [6; 9].
  - With word alignment: terminology replacement in automatic post-editing
    - Word alignment induced from NMT is not accurate enough [5; 15; 8; 3].

Accurate Alignment for Terminology Translation [2]

- Our observation:

  The hidden states at decoding step $i + 1$ are better representing target word y_i than the hidden states at step i for inducing word alignment



- Preprint: https://arxiv.org/abs/2004.14837

Accurate Alignment for Terminology Translation [2]

- Our work:
  - We introduce SHIFT-ATT, a pure interpretation method to induce alignments from attention weights of vanilla Transformer.
    - To select the best layer to induce alignments, we propose a surrogate layer selection criterion, to ensure the induced word alignments agree best in both translation directions, without manually labelled word alignments.
  - We further propose SHIFT-AET, which extracts alignments from an additional alignment module.
- Preprint: https://arxiv.org/abs/2004.14837

Experiments and Results

Method	Inter.	Fullc	de-en			fr-en			ro-en		
			de→en	en→de	bidir	fr→en	en→fr	bidir	ro→en	en→ro	bidir
Statistical Methods											
FAST-ALIGN (Dyer et al., 2013)	-	Y	28.5	30.4	25.7	16.3	17.1	12.1	33.6	36.8	31.8
GIZA++ (Brown et al., 1993)	-	Y	18.8	19.6	17.8	7.1	7.2	6.1	27.4	28.7	26.0
Neural Methods											
NAIVE-ATT (Garg et al., 2019)	Y	N	33.3	36.5	28.1	27.5	23.6	16.0	33.6	35.1	30.9
SMOOTHGRAD (Li et al., 2016)	Y	N	36.4	45.8	30.3	25.5	27.0	15.6	41.3	39.9	33.7
SD-SMOOTHGRAD (Ding et al., 2019)	Y	N	36.4	43.0	29.0	25.9	29.7	15.3	41.2	41.4	32.7
PD (Li et al., 2019)	Y	N	38.1	44.8	34.4	32.4	31.1	23.1	40.2	40.8	35.6
ADDSGD (Zenkel et al., 2019)	N	N	26.6	30.4	21.2	20.5	23.8	10.0	32.3	34.8	27.6
MTL-FULLC (Garg et al., 2019)	N	Y	-	-	20.2	-	-	7.7	-	-	26.0
Our Neural Methods											
SHIFT-ATT	Y	N	20.9	25.7	17.9	17.1	16.1	6.6	27.4	26.0	23.9
SHIFT-AET	N	N	16.1	19.3	**15.5**	10.3	10.5	**5.0**	22.4	23.7	**21.2**

Table 1: AER on the test set with different alignment methods. *bidir* are symmetrized alignment results. The column Inter. represents whether the method is an interpretation method that can extract alignments from a pretrained vanilla Transformer model. The column Fullc denotes whether full target sentence is used to extract alignments at test time. The lower AER, the better. We mark best bidir interpretation results of vanilla Transformer with underlines, and best bidir results among all with boldface.

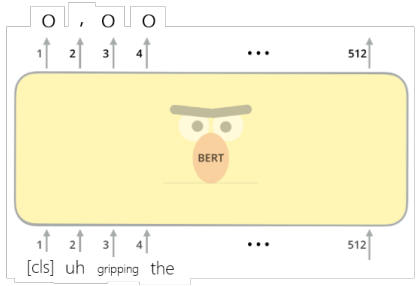
Content

Punctuation Prediction

- Models
  - Baseline: BiLSTM Yi et al. [13]
  - BERT with Fine-tune
  - TinyBERT [7] (Distilled from BERT)
- Data: IWSLT2011 TED Corpus
- Labels:

4 labels	3 labels
Comma(,)	
Period(.)	Period(.)
Question(?)	Question(?)
None(O)	None(O)

uh, gripping the onion like a you know, like a tennis ball, holding it together in place.



uh gripping the onion like a you know like a tennis ball holding it together in place

Punctuation Prediction

	Model	Comma			Period			Question			Overall		
		$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
Ref.	DNN	58.1	35.8	44.3	62.1	64.8	63.4	60.5	48.9	54.1	60.2	49.8	53.9
	T-BRNN-pre [17]	65.5	47.1	54.8	73.3	72.5	72.9	70.7	63.0	66.7	70.0	59.7	64.4
	BLSTM-CRF	58.9	59.1	59.0	68.9	72.1	70.5	71.8	60.6	65.7	66.5	63.9	65.1
	Teacher-Ensemble	**66.2**	**59.9**	**62.9**	**75.1**	**73.7**	**74.4**	**72.3**	**63.8**	**67.8**	**71.2**	**65.8**	**68.4**
ASR	DNN	47.5	32.3	38.5	58.3	60.5	59.4	57.1	46.8	51.4	54.3	46.5	49.8
	T-BRNN-pre [17]	59.6	42.9	49.9	70.7	72.0	71.4	60.7	48.6	54.0	66.0	57.3	61.4
	BLSTM-CRF	55.7	56.8	56.2	68.7	71.5	70.1	63.8	53.4	58.1	62.7	60.6	61.5
	Teacher-Ensemble	**60.6**	**58.3**	**59.4**	**71.7**	**72.9**	**72.3**	**66.2**	**55.8**	**60.6**	**66.2**	**62.3**	**64.1**

(Yi, Jiangyan, et al. Interspeech 2017, baseline)

	comma	period	question	other	Overall		
	acc	acc	acc	acc	pre	rec	f1(macro)
4 labels	0.716	0.84	0.664	0.983	0.755	0.745	0.75
3 labels	—	0.92	0.71	0.99	0.893	0.887	0.894
3 labels vs. 4 labels					↑18%	↑19%	↑19%
3 labels vs. baseline-ref					↑35%	↑42%	↑31%

(3 labels VS 4 labels VS baseline, ↑indicate relative improvements)

	precision	recall	f1(macro)	train speed	infer speed
teach model (bert base)	0.893	0.887	0.894		
student model (tiny-bert)	0.857	0.837	0.867	X12	X5
			-3%		

(Knowledge Distillation for Inference acceleration)

Content

Disfluency Detection and Correction

- English Switchboard Corpus:

I want a flight [$\underbrace{\textit{to Boston}}_{\text{RM}} + \underbrace{\{\textit{um}\}}_{\text{IM}}$ $\underbrace{\text{to Denver}}_{\text{RP}}$]

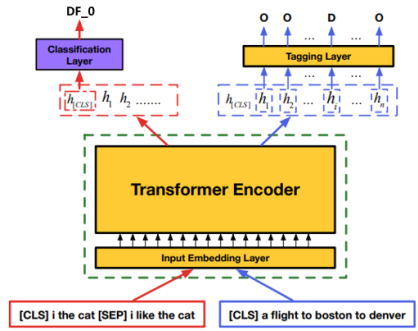
Figure 1: A sentence from the English Switchboard corpus with disfluencies annotated. RM=Reparandum, IM=Interregnum, RP=Repair. The preceding RM is corrected by the following RP.

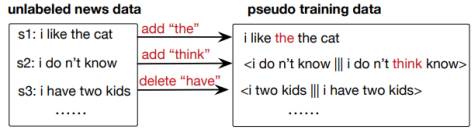
(from Wang et al. [11])

- IWSLT2020 TED Corpus (Chinese):
  - ASR: 黑胡桃木，黑胡桃木是我们店内最好的木材。
  - REF: 黑胡桃木是我们店内最好的木材。
  - ASR: 啊它的性质很好不容易开裂，
  - REF: 它的性质很好不容易开裂，

Disfluency Detection and Correction

- BERT with Multitask Fine-tuning
  - follow Wang et al. [11]
- Task1: Sequence labeling
  - RM and IM $\rightarrow$ D (DELETE)
  - RP and Others $\rightarrow$ O (OTHER)
- Task2: Sentence pair classification
  - Smooth/Disfluent $\rightarrow$ DF_0
  - Disfluent/Smooth $\rightarrow$ DF_1
- Training Data: English Switchboard
- Data augmentation:
  - Randomly insert IM
  - Swap RM & RP





Disfluency Detection and Correction

- Experiment (English Switchboard Corpus):

	P	R	F1
UBT [12]	90.3	80.5	85.1
Semi-CRF [4]	90	81.2	85.4
Bi-LSTM [14]	91.8	80.6	85.9
Transition-based [10]	91.1	84.1	87.5
MTL Self-supervised [11]	**93.4**	87.3	**90.2**
Our MTL-DA	91.3	**87.7**	89.4

- Experiment (IWSLT2020 TED Corpus, Chinese):

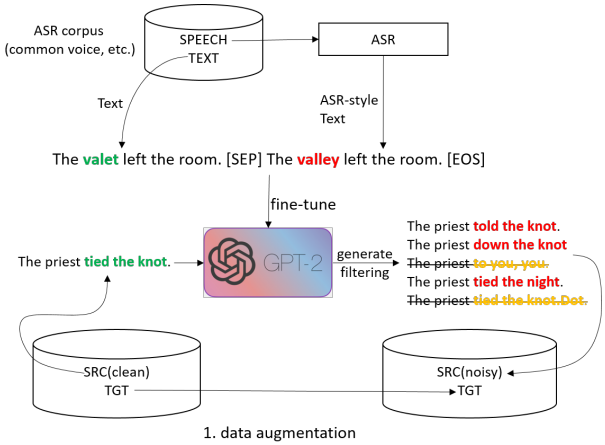
TER (Translation Error Rate)	Train	Dev	Test
Baseline	35.2	36.3	33.1
+ Expert Rules (pre-processing)	31.8(-3.4)	31.9(-4.4)	30.1(-3.0)
+ BERT Fine-tuning	28.3(-3.5)	28.3(-3.6)	56.7(-3.4)
+ Dictionary	**26.4(-1.9)**	**25.9(-2.4)**	**24.9(-1.8)**

Content

Motivation

- ASR outputs always contain errors which severely harms the quality of the downstream MT system.
- Ideally, fine-tuning the MT system with the pairs of noisy ASR outputs and their translations will benefit.
- However, there is very little parallel data with ASR-output in the source side.
- Our solution:
  - Using ASR training data (SPEECH, TEXT) and an existing ASR system to train a GPT-2 system to generate ASR-like texts from normal texts.
  - Using the obtained system to transfer the source text of a bilingual courpus to ASR-like text.

Robust MT Training



Noisy Data Filtering

- The noisy data generated by the GPT-2 system may be TOO noisy.
- We use an edit rate threshold on pronunciations to filter the GPT-2 outputs.
  - Sentence $\rightarrow$ pronunciation using cmudict[1].
  - Edit Rate: Edit distance normalized by the original length.

clean: The priest **tied the knot**.

['DH', 'AH0', 'P', 'R', 'IY1', 'S', 'T', 'T', 'AY1', 'D', 'DH', 'AH0', 'N', 'AA1', 'T']

noisy1: The priest **told the knot**.

['DH', 'AH0', 'P', 'R', 'IY1', 'S', 'T', 'T', 'OW1', 'L', 'D', 'DH', 'AH0', 'N', 'AA1', 'T']

D=2 ER = 0.4

noisy2: The priest **to you, you**

['DH', 'AH0', 'P', 'R', 'IY1', 'S', 'T', 'T', 'UW1', 'Y', 'UW1', 'Y', 'UW1']

D=7 ER = 1.4

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Experiments

- Data: EN→ZH, Huawei STW meeting dataset and MSLT v1.1 [2]
- Models:
  - transformer-small, emb size:256, intermediate size:1024, attention heads:4, training data: 2M pairs
  - transformer-big, emb size:1024, intermediate size:4096, attention heads:16, training data: 1.4B pairs

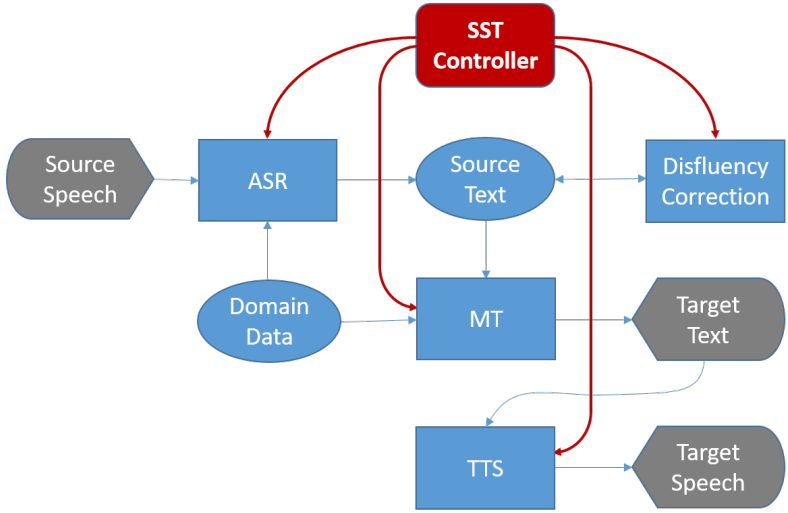
		STW		MSLT	
		dev	test	dev	test
small	clean	36.29	36.84	29.61	31.21
	robust	**36.93(+0.64)**	**37.69(+0.85)**	**30.69(+1.08)**	**32.56(+1.35)**
big	clean	44.18	44.03	33.75	34.29
	robust	**45.85(+1.67)**	**45.46(+1.43)**	**34.65(+0.9)**	**35.23(+0.96)**

[2] https://github.com/MicrosoftTranslator/MSLT-Corpus

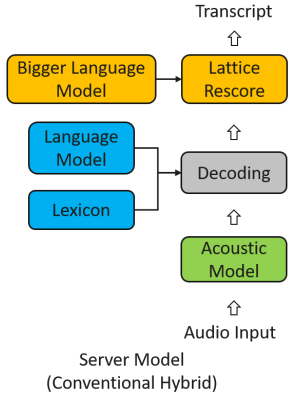
Content

HUAWEI

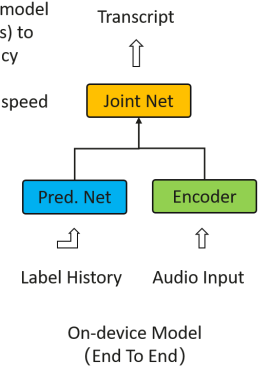
System Architecture



Adapting ASR from Cloud to Mobile Devices



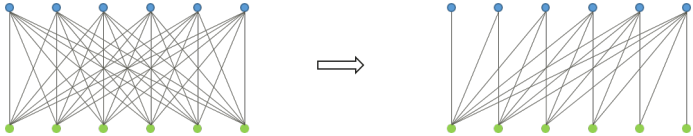
1. Propose the E2E Conv-Transformer Transducer model
2. The model is compact enough (60m parameters) to run on a mobile phone while has similar accuracy with server-side conventional hybrid model
3. Using truncated attention, states are reused to speed up Transformer
4. Low framerate, 80ms/frame.
5. Small look-ahead window, 140ms.
6. Audio encoder and Prediction net running on different threads.
7. Powered by Noah's Bolt framework.



HUAWEI

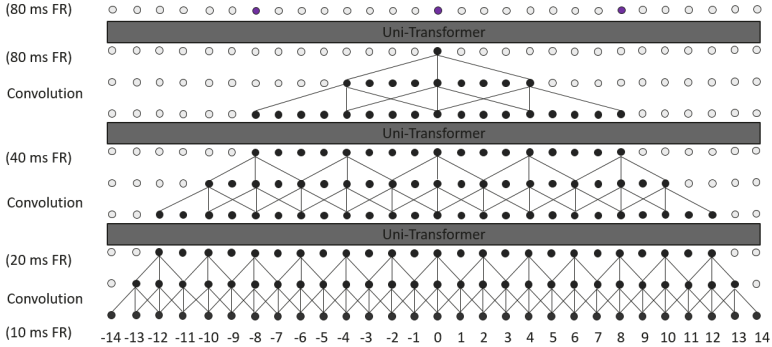
Conv-Transformer Transducer

- Streamble: unidirectional Transformer for audio encoding



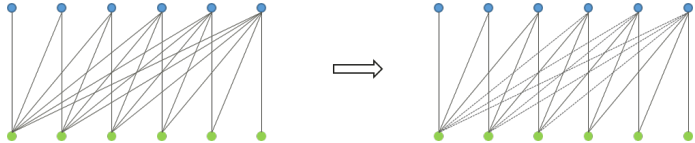
Conv-Transformer Transducer

- Low Frame Rate: interleaved convolutions for gradually downsampling



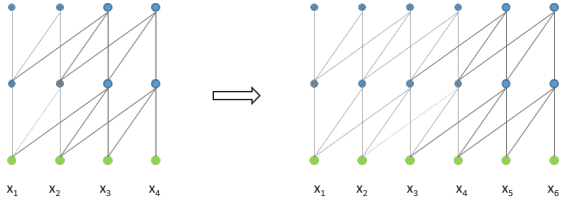
Conv-Transformer Transducer

- Reduced Computation Cost: self-attention with fixed context window

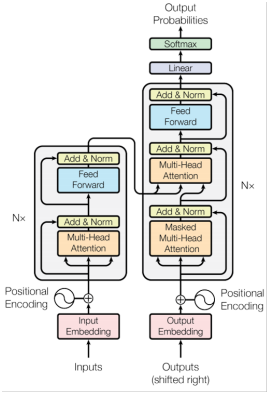


Conv-Transformer Transducer

- Reduced Computation Cost: hidden state reuse with relative positional encoding

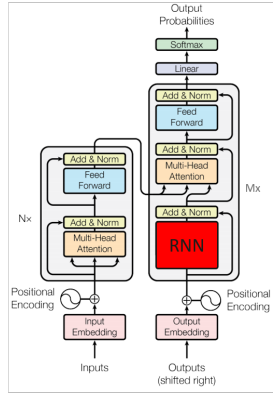


Adapting NMT from Cloud to Mobile Devices



Cloud Model

1. Use RNN decoder instead of Transformer decoder.
2. Decrease layer number to compress model size.
3. Share parameter for different language.
4. Use smaller vocab size.
5. Model quantization for model storage to lower hard disk usage.
6. Model quantization during training to limit parameter value range which enable more stable quantized inference.
7. Model quantization during inference to enhance using experience.
8. Use cloud model to generate more data for training on-device model.
9. Powered by Bolt framework.



On-device Model

HUAWEI

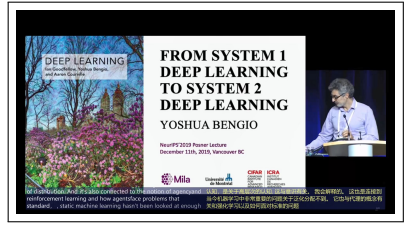
Model Simplication on Model Devices

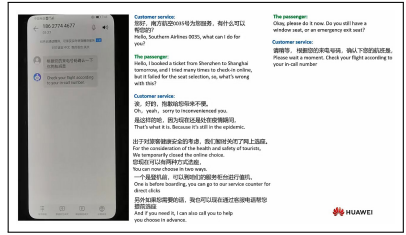
- Use RNN layers instead of masked multi-head self-attention layers;
- Decrease number of decoder model stacks, from N layers on cloud, to M layers on devides;
- Share model parameters for different languages;
- Use smaller vocab size, from x to y.

Content

HUAWEI

Video Demonstrations





References I

[1] Yun Chen, Liangyou Li, Xin Jiang, Xiao Chen, and Qun Liu. How to do simultaneous translation better with consecutive neural machine translation?, 2019.

[2] Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. Accurate word alignment induction from neural machine translation, 2020.

[3] Shuoyang Ding, Hainan Xu, and Philipp Koehn. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of WMT 2019)*, pages 1–12, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5201. URL https://www.aclweb.org/anthology/W19-5201.

[4] James Ferguson, Greg Durrett, and Dan Klein. Disfluency detection with a semi-markov model and prosodic features. In *Proceedings of the 2015 Conference of NAACL-HLT 2015*, pages 257–262, 2015.

[5] Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. In *Proceedings of EMNLP-IJCNLP 2019*, pages 4453–4462, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1453. URL https://www.aclweb.org/anthology/D19-1453.

[6] Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*, 2017.

[7] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *proceedings of ICLR 2020*, 2020.

[8] Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. On the word alignment from neural machine translation. In *Proceedings of ACL 2019*, pages 1293–1303, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1124. URL https://www.aclweb.org/anthology/P19-1124.

HUAWEI

References II

[9] Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*, 2018.

[10] Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. Transition-based disfluency detection using lstms. In *Proceedings of EMNLP 2017*, pages 2785–2794, 2017.

[11] Shaolei Wang, Wanxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. Multi-task self-supervised learning for disfluency detection. In *Proceedings of AAAI 2020*, 2020.

[12] Shuangzhi Wu, Dongdong Zhang, Ming Zhou, and Tiejun Zhao. Efficient disfluency detection with transition-based parsing. In *Proceedings of ACL-IJCNLP 2015*, pages 495–503, 2015.

[13] Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Ya Li, et al. Distilling knowledge from an ensemble of models for punctuation prediction. In *Proceedings of Interspeech 2017*, 2017.

[14] Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. Disfluency detection using a bidirectional lstm. *arXiv preprint arXiv:1604.03209*, 2016.

[15] Thomas Zenkel, Joern Wuebker, and John DeNero. End-to-end neural word alignment outperforms giza++, 2020.

Thank for listening!

Team members:

CHEN Xiao, CHEN Yun, CUI Tong, DENG Yao, FU Xu, ZHANG Guchun, GUO Weisheng, HU Wenchao, HUANG Wenyong, JIANG Xin, LI Liangyou, LIN Fuguo, LIN Xia, LIU Jianfeng, LIU Kai, LIU Qun, LIU Xin, PENG Wei, WANG Minghan, XIAO Jinghui, XIA Hairong, YANG Hao, ZHOU Huan.