Modeling Fluency and Faithfulness for Diverse Neural Machine Translation

Yang Feng[1,2] **Wanying Xie**[1,3] **Shuhao Gu**[1,2] **Chenze Shao**[1,2]
Wen Zhang[4] **Zhengxin Yang**[1,2] **Dong Yu**[3*]

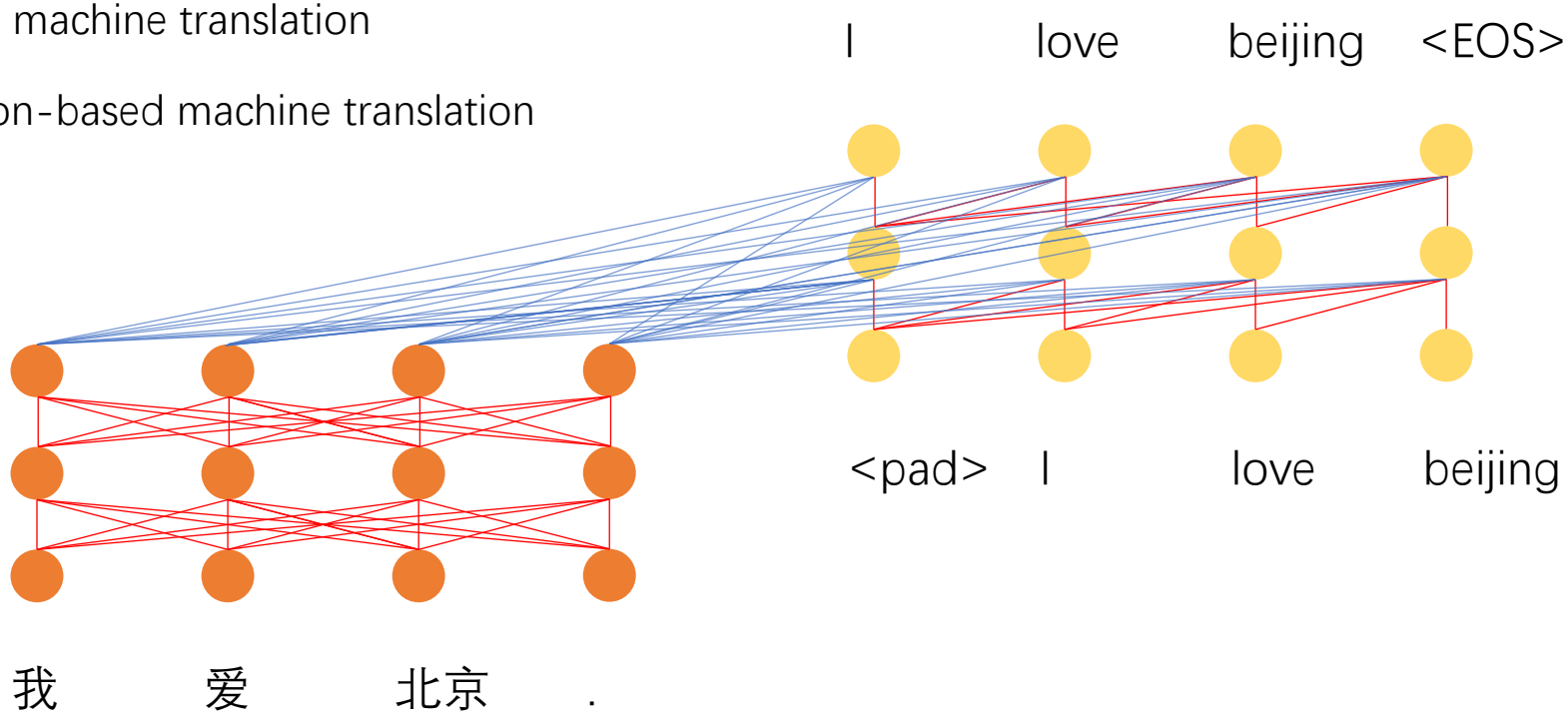
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Beijing Language and Culture University, China
[4] Smart Platform Product Department of Tencent Inc., China

Outline

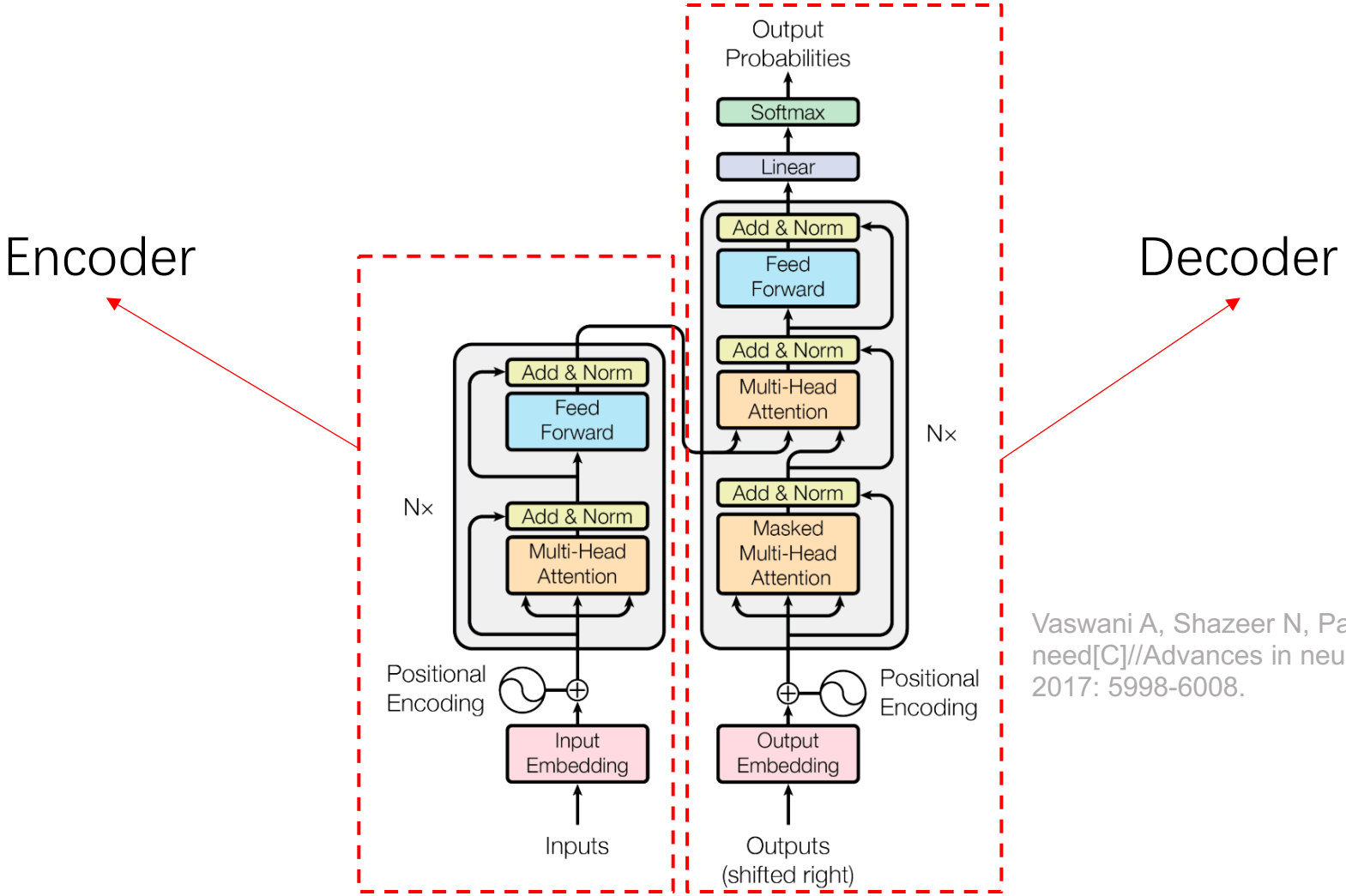
- Introduction of NMT
- Motivation
- Method Description
- Experiments & Conclusion

Neural Machine Translation

- **Machine Translation**: Use of software to translate text or speech from one language to another.
  - RNN-based machine translation
  - CNN-based machine translation
  - Self-attention-based machine translation



Self-attention-based MT System



Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

Motivation

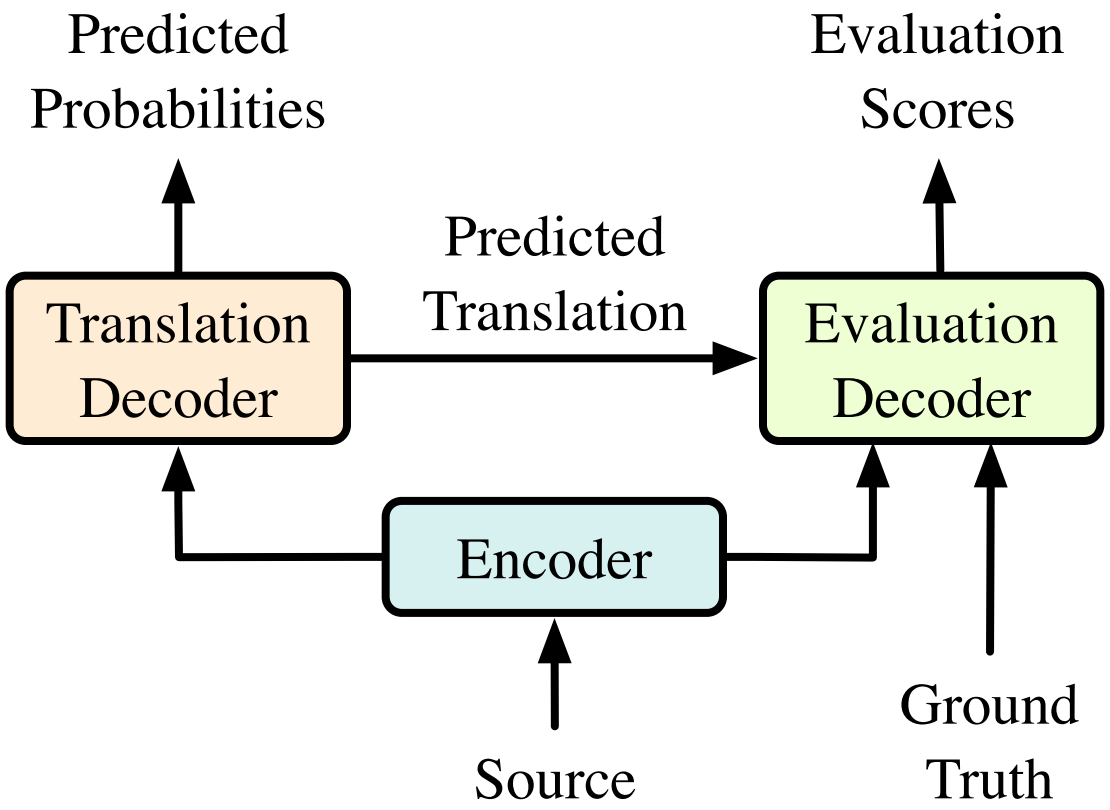
- Teacher forcing strategy forces the probability of each prediction to approach a 0-1 distribution.
- Casts all the portion of the distribution to the ground truth word and ignores other words in the target vocabulary.
- Source: 一场 大雪 袭击 了 北京 . → Target: a heavy snow hit beijing .
- Model: a big snow hit beijing .

Introduction of Our Method

- We propose a method to introduce an evaluation module to guide the distribution of the prediction.

- The evaluation module accesses each prediction from the perspectives of fluency and faithfulness.

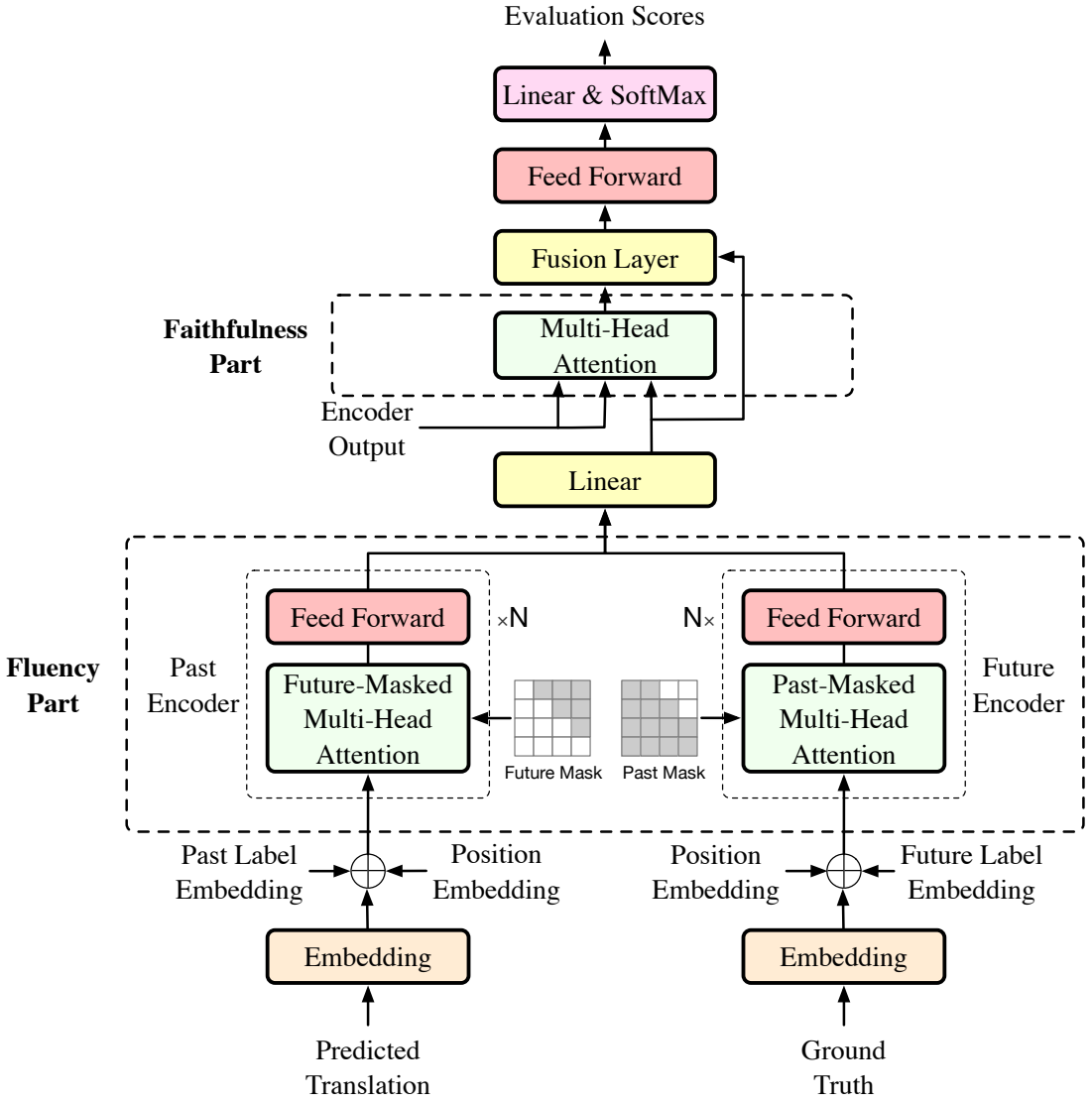
Our Model



$$\mathcal{L}_t = -\sum_{k=1}^{K}\sum_{i=1}^{I} \log p(y_i^*|\mathbf{y}_{<i}, \mathbf{x})$$

$$\mathcal{L}_e = -\sum_{k=1}^{K}\sum_{i=1}^{I} \log p_e(y_i^*|\mathbf{y}_{>i}^*, \mathbf{y}_{<i}, \mathbf{x})$$

Our Model



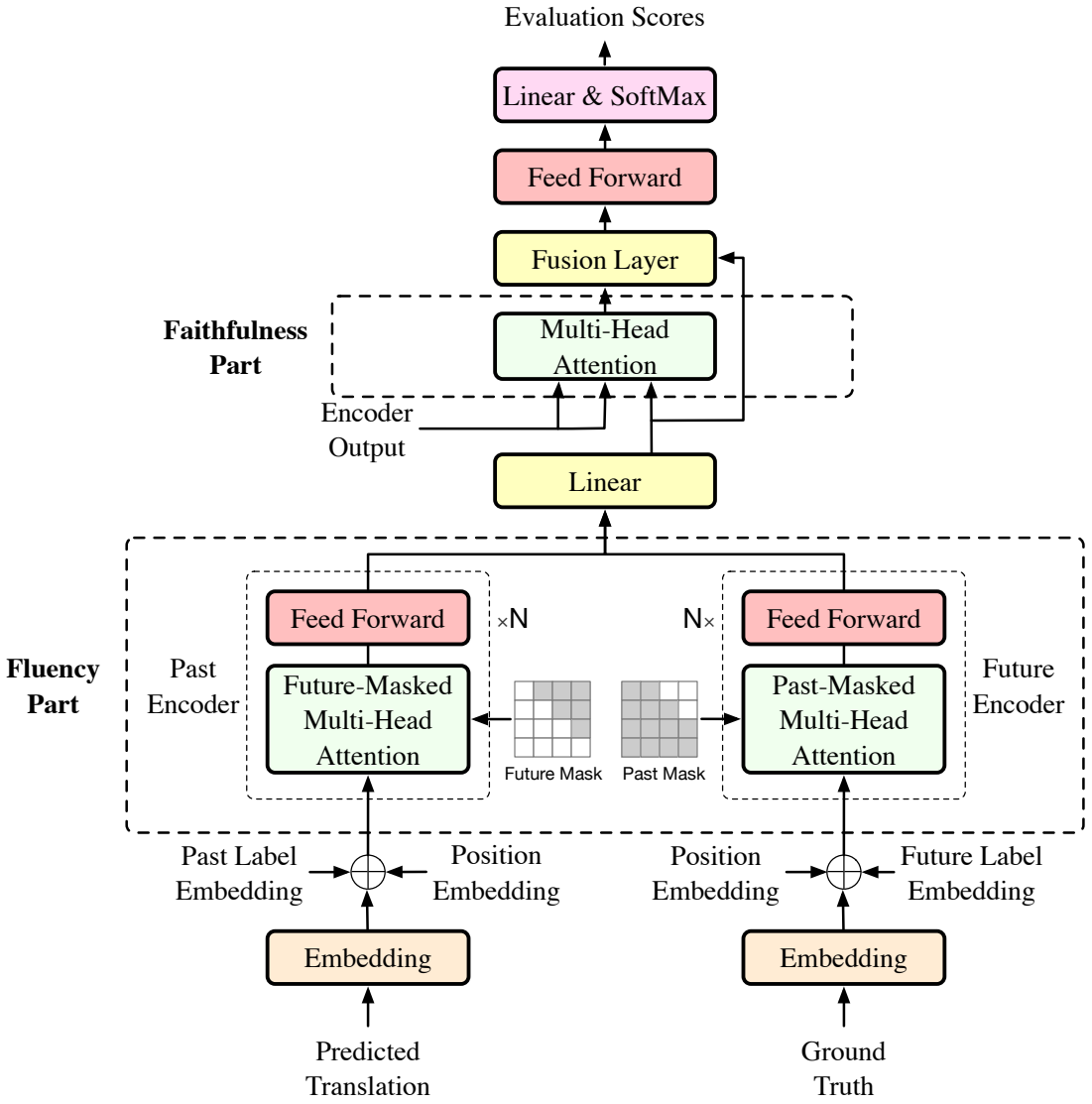
$$\mathcal{L}_{\mathrm{c}} = \sum_{k=1}^{K} \sum_{i=1}^{I} p_e(y_i|\mathbf{y}^*_{>i}, \mathbf{y}_{<i}, \mathbf{x}) \log p(y_i|\mathbf{y}_{<i}, \mathbf{x})$$

Our Method: $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_e + \mathcal{L}_{\mathrm{c}}$

$$\mathcal{L}_{\mathrm{KL}} = \sum_{k=1}^{K} \sum_{\mathbf{y}_i \neq \mathbf{y}^*_i} \mathrm{D}_{\mathrm{KL}}(p_e(y_i|\mathbf{y}^*_{>i}, \mathbf{y}_{<i}, \mathbf{x}) || p(y_i|\mathbf{y}_{<i}, \mathbf{x}))$$

Our Method-KL: $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_e + \mathcal{L}_{\mathrm{KL}}$

Our Model



Source: 一场 大雪 袭击 了 北京 .
Target: a heavy snow hit beijing .
Model: a big snowfall hit beijing .

L_c : e.g. $P(\text{big}|Evaluation) \log P(\text{big}|Translation)$

L_{KL} : e.g. $D_{KL}P(Evaluation) \,||\, P(Translation)$

Experiment

- Data preparation:
  - CN→EN
    - NIST, 1.25M; valid: MT02; test: MT03, MT04, MT05, MT06, MT08
    - BPE: 30k merge operations
  - EN→DE
    - WMT2014, 4.5M; valid: test-2013; test: test-2014
    - BPE: 32k merge operations
  - EN→RO
    - WMT16, 0.6M; valid: news-dev 2016; test: news-test 2016
    - BPE: 40k merge operations
- Model:
  - Transformer_Base

Results

	CN→EN							EN→DE		EN→RO	
	MT03	**MT04**	**MT05**	**MT06**	**MT08**	**AVE**	**Δ**	**WMT14**	**Δ**	**WMT16**	**Δ**
TRANSFORMER	44.74	46.27	44.16	43.29	34.72	42.63		27.21		32.85	
+RL	44.50	45.96	44.26	43.92	35.55	42.83	*+0.20*	27.25	*+0.04*	33.00	*+0.15*
+BOW	44.59	46.40	45.03	43.91	35.31	43.04	*+0.41*	27.35	*+0.14*	32.95	*+0.10*
Our Method-KL	45.17	46.86**	45.01**	44.51*	36.03*	43.51	*+0.88*	**27.55**	***+0.34***	33.44	*+0.59*
Our Method	**46.20***	**47.39***	**46.22***	**45.63***	**36.78***	**44.44**	***+1.81***	27.35	*+0.14*	**34.00***	***+1.15***

Table 1: BLEU scores on three translation tasks. * and ** mean the improvements over TRANSFORMER is statistically significant (Collins, Koehn, and Kucerova 2005) ($\rho < 0.01$ and $\rho < 0.05$, respectively).

- Our method with $\mathbf{L_c}$ as an additional loss can outperform all the baselines significantly on the CN⟶EN and EN⟶RO translation tasks.
- The evaluation module can help improve translation performance and the loss $\mathbf{L_c}$ is more reasonable.

Analysis

	MT03	MT04	MT05	MT06	MT08	AVE
Full	46.20	47.39	46.22	45.63	36.78	44.44
-Faithfulness	44.95	46.85	45.15	44.45	36.18	43.51
-$\mathcal{L}_c$	44.93	45.77	44.61	44.76	35.87	43.18
-Evaluation	44.74	46.27	44.16	43.29	34.72	42.63

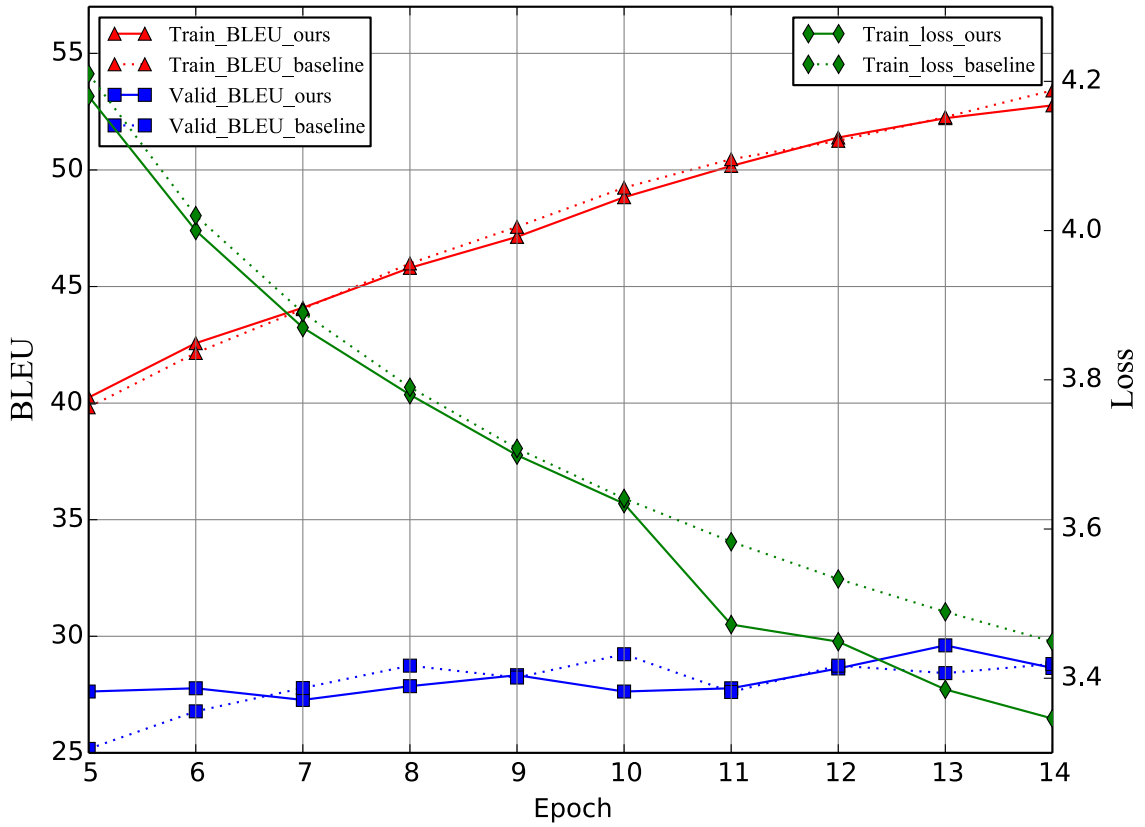
- Our method introduces three new factors, including the **cross attention for faithfulness**, **past and future encoders for fluency** and the **additional loss L_c**.
- Ablation study shows that each factor plays an important role in our method.

Analysis

	MT03	MT04	MT05	MT06	MT08	AVE	EN→DE
Tran	31.31	23.87	29.02	30.61	22.26	27.41	31.76
Eval	45.86	39.34	44.18	45.21	36.02	42.12	34.56

- Ground truth is fed to both the modules in order to show the performance between the translation and evaluation modules.
- The evaluation module indeed has **a great margin** in performance over the translation module.

Analysis



- When converged, our method has higher BLEU scores on both the training and valid sets, then We can think our method reaches better optimization than Transformer.

Analysis

	1-gram	2-gram	3-gram	4-gram	Cosine
Transformer	79.10	52.72	35.34	23.82	0.873
Our method	79.82	54.18	36.90	25.28	0.877

- Generate translation with **higher n-gram accuracy** for order 1 to order 4 and the difference becomes wider as the order increases.
- Have a **bigger cosine similarity** to the reference and this means the generate translation is more faithful in meaning to the source sentence.

Conclusion

- We introduce an evaluation module to draw a new distribution over all the words and further use the new distribution to guide the training.
- We estimate the translation from the perspectives of fluency and faithfulness to appreciate the translation which is fluent in the target and faithful in meaning to the source.
- Our method can get better performance on multiple data sets with better optimization and meanwhile the generated translation is more fluent in the target and more faithfulness to the source.

Thanks for Listening!