NAIST English-to-Japanese Simultaneous Translation System

for IWSLT 2021 Simultaneous Text-to-text Task

Ryo Fukuda, Yui Oka, Yasumasa Kano, Yuki Yano,
Yuka Ko, Hirotaka Tokuyama, Kosuke Doi,
Sakriani Sakti, Katsuhito Sudoh, Satoshi Nakamura

Nara Institute of Science and Technology (NAIST)
Japan

This work is supported by JSPS KAKENHI (Grant number 17H06101).

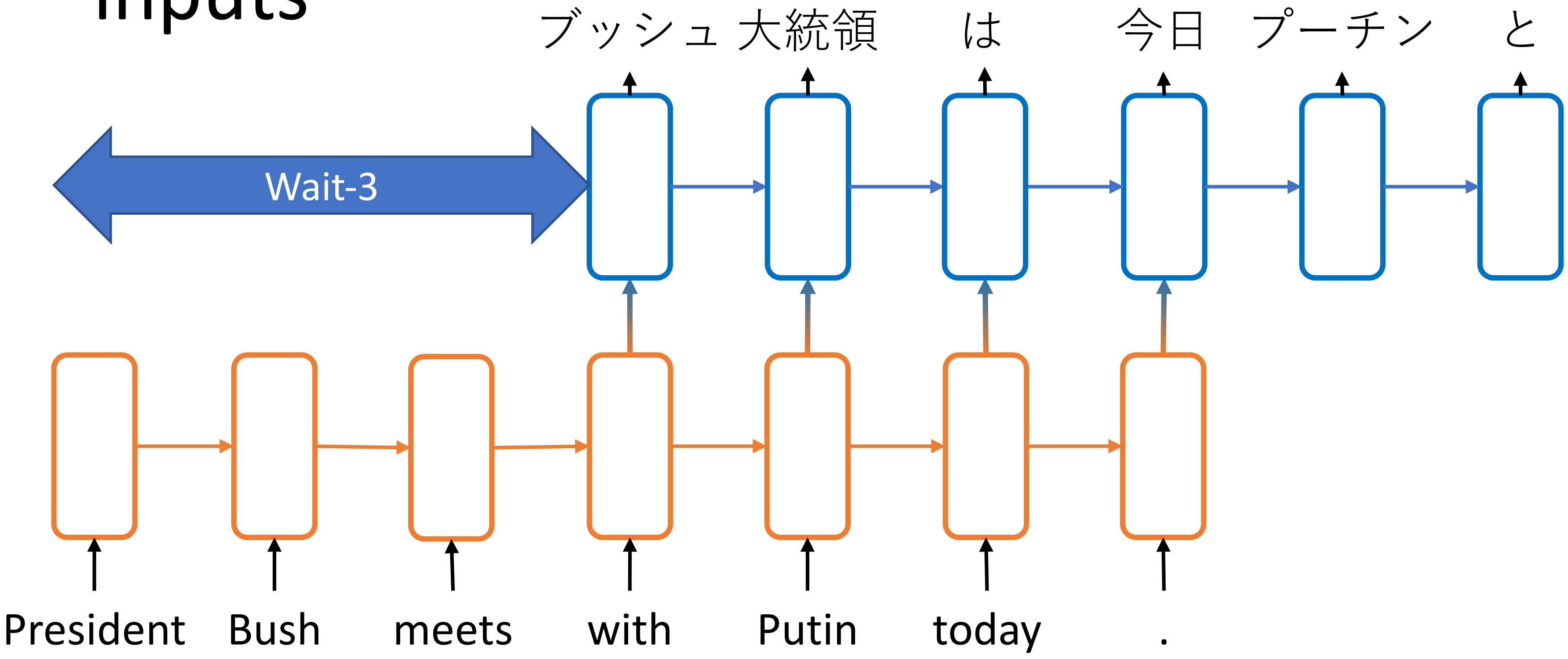
Overview of Our Submissions

Latency regime	**Low (AL ≤ 8)**	**Medium (AL ≤ 12)**	**High (AL ≤ 16)**
Base system	wait-*k* (Ma+ 2019) on Fairseq (Ott+ 2019) & SimulEval (Ma+ 2020) w/ Transformer-base settings		
Data	Training: WMT20 News & IWSLT17 train / Dev: IWSLT17 dev Shared BPE vocabulary (16,000)		
Latency hyperparameter	$k = 10$	$k = 20$	$k = 30$
Additional training-time feature	Target-side chunk shuffling	Seq. Knowledge Distillation from an *offline* model	
BLEU/AL (dev)	13.77 / 7.29	15.22 / 11.48	15.57 / 13.70
BLEU/AL (test)	**14.41 / 7.21**	**16.20 / 11.54**	**16.19 / 13.83**

wait-*k* prefix-to-prefix translation

(Ma+ 2019)

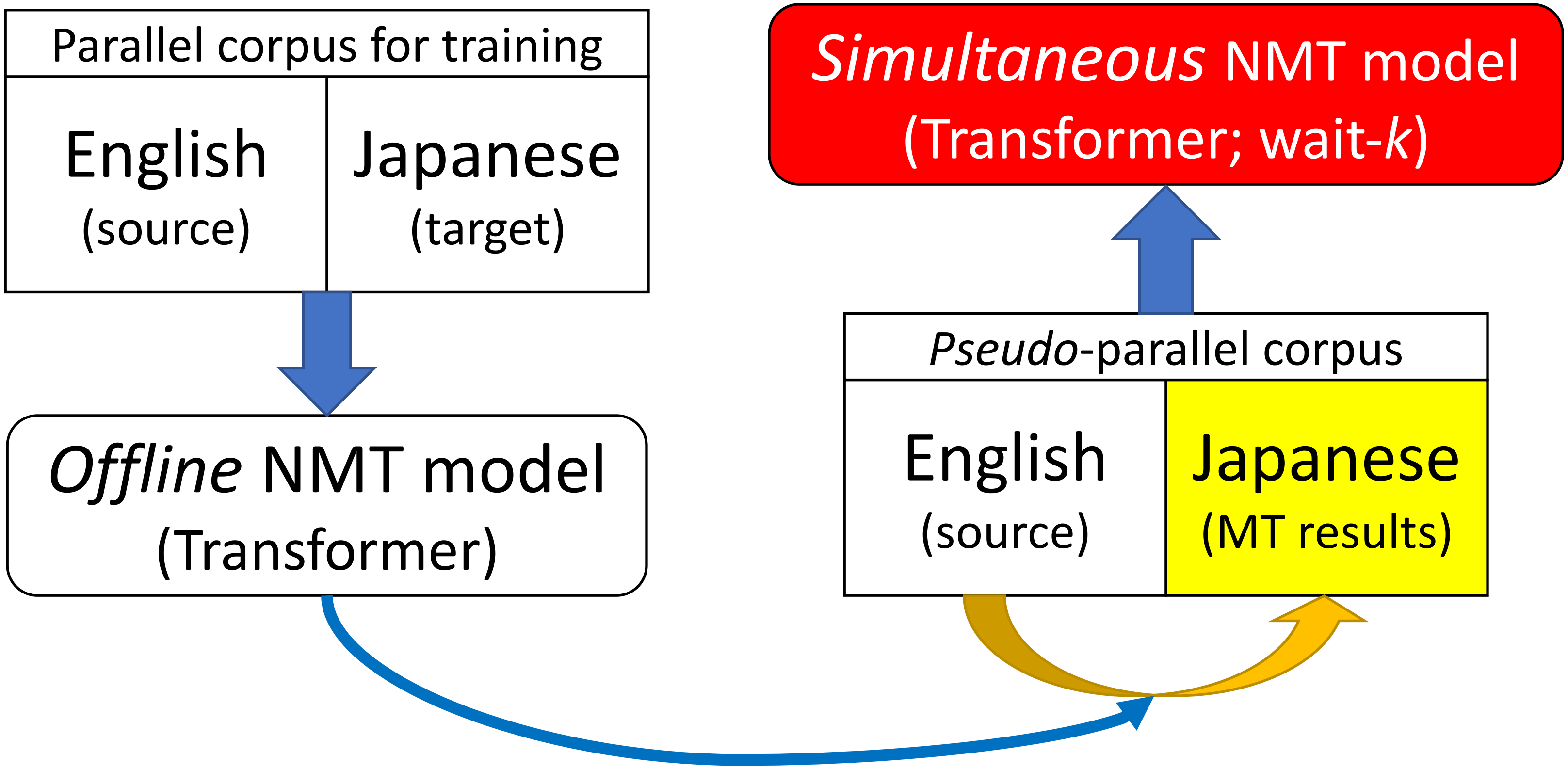
- Wait for *k* tokens first and then generate outputs concurrently with inputs



Seq.-level Knowledge Distillation

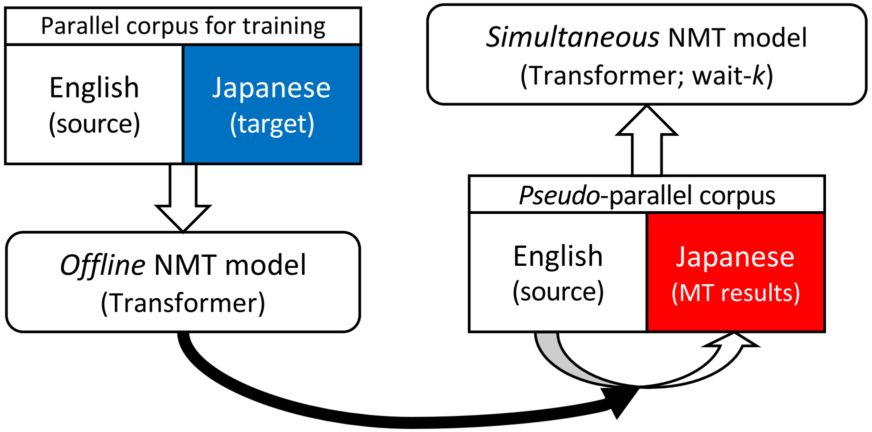
(Kim and Rush, 2016)

- Train simultaneous (student) NMT using outputs from offline (teacher) NMT



Why Seq.-level KD?

- Colloquial expressions and free (non-literal) translation in Japanese subtitles
    - Difficult to generate for NMT…
- Literal translation by NMT
    - Would be easy to generate for NMT


Parallel corpus for training
English (source)
Japanese (target)
Offline NMT model (Transformer)
Pseudo-parallel corpus
English (source)
Japanese (MT results)
Simultaneous NMT model (Transformer; wait-*k*)

Motivated by recent non-autoregressive NMT

Mitigate *non-parallelism* in the training data

Target-side Chunk Shuffling (Cshuf)

- Reorder Japanese text chunks randomly in the training data
  - With a small probability p_r (1 to 3%)
  - Chunk size *k* is fixed and set as the same as latency hyperparameter for wait-*k*

Japanese sentence in the training data (tokenized)

$t_1, t_2, \cdots, t_k$	$t_{k+1}, t_{k+2}, \cdots, t_{2k}$	$t_{2k+1}, t_{2k+2}, \cdots, t_{3k}$	$t_{3k+1}, t_{3k+2}, \cdots, t_{4k}$

Chunk shuffling

$t_{k+1}, t_{k+2}, \cdots, t_{2k}$	$t_{3k+1}, t_{3k+2}, \cdots, t_{4k}$	$t_1, t_2, \cdots, t_k$	$t_{2k+1}, t_{2k+2}, \cdots, t_{3k}$

Why Chunk Shuffling?

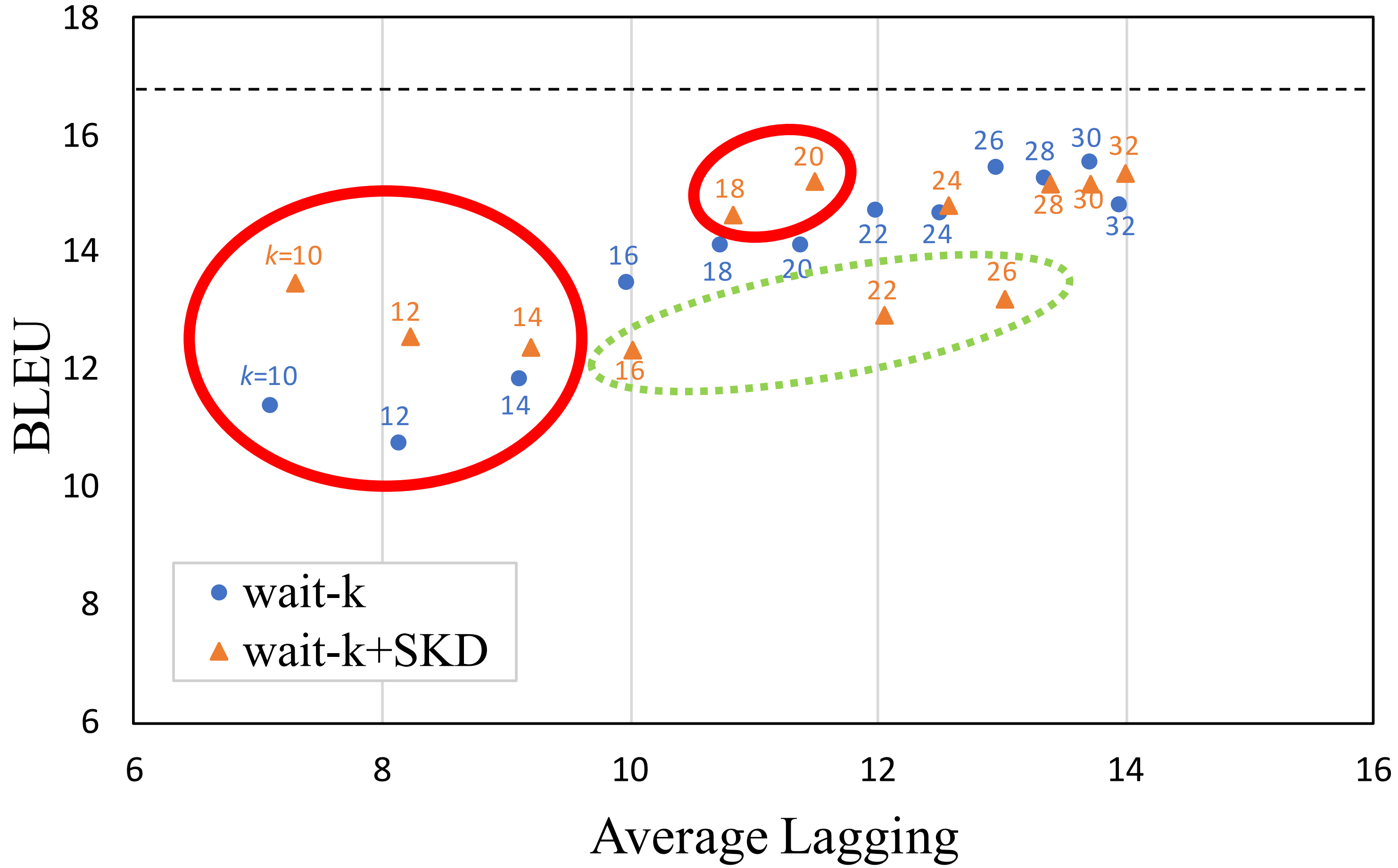
- Rough simulation of chunk reordering allowed in Japanese
  - The order of Japanese chunks (*bunsetsu*; 文節) is not so strict
- CShuff encourages order-free outputs in the training, which would help monotonic translation using wait-*k*
- More linguistically-motivated reordering would be worth trying in future work…

System Setup

- wait-*k* on fairseq & SimulEval (based on the official baseline)
  - Transformer-base
  - BPE-based subwords with voc. size of 16,000, shared btw. En-Ja
- Training: 17.9M from WMT20, 223K from IWSLT17 train
- Fine-tuning: IWSLT17 dev (fine-tune)
- Dev-test: IWSLT21 dev (dev-test)

IWSLT21 dev results: Seq.-level KD

- BLEU improvements with small *k*



IWSLT21 dev results: CShuf

- Worked the best with $p_r = 0.02$
  - Very sensitive with p_r, in output length

	p_r	BLEU	Length ratio (hyp / ref)
Baseline (wait-10)	0	11.80	1.233
Target-side Chunk Shuffling (Cshuf)	0.01	10.57	1.372
	0.02	**13.77**	**1.053**
	0.03	9.87	1.516

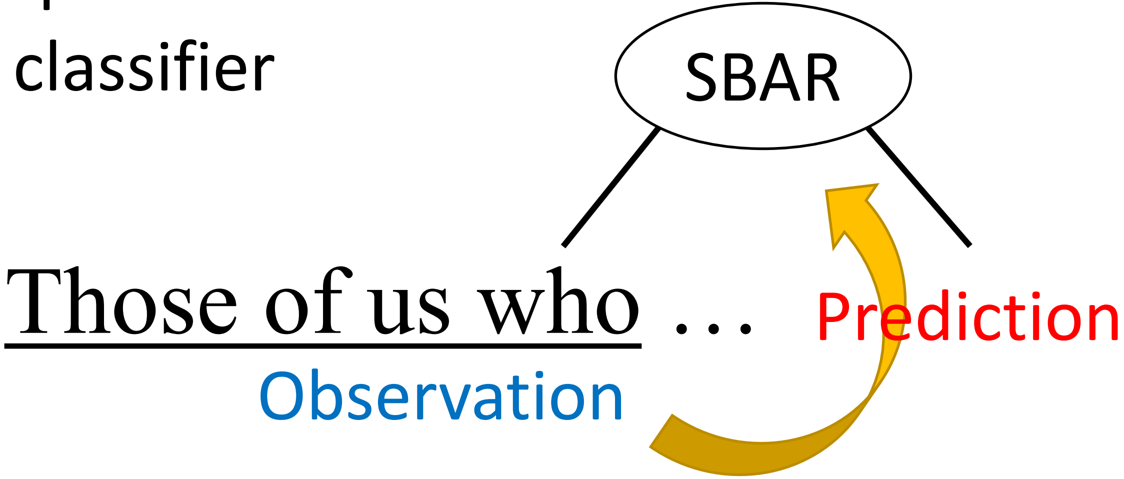
Results Summary

- Mid-latency system worked (relatively) well, but still 2 pts. behind other teams

System	IWSLT21 dev		IWSLT21 test (Official)	
	BLEU	AL	BLEU	AL
Offline	16.80	-	-	-
wait-10 +Cshuf (low latency)	13.77	7.29	14.41	7.21
wait-20 +Seq.KD (medium latency)	15.22	11.48	16.20	11.54
wait-30 (high latency)	15.57	13.70	16.19	13.83

Another attempt

- The use of *future* syntax information
  - Motivated by Oda et al. (2015) that used it to determine when to start translation in simultaneous SMT
  - *Next Constituent Label Prediction* (NCLP)
    - 1-lookahead prediction
    - BERT-based classifier



wait-k with NCLP

- Enhance inputs with predicted labels
  - The length of an input is *doubled*

Those of us who are underrepresented …

Those

Those PP of

Those PP of NP us

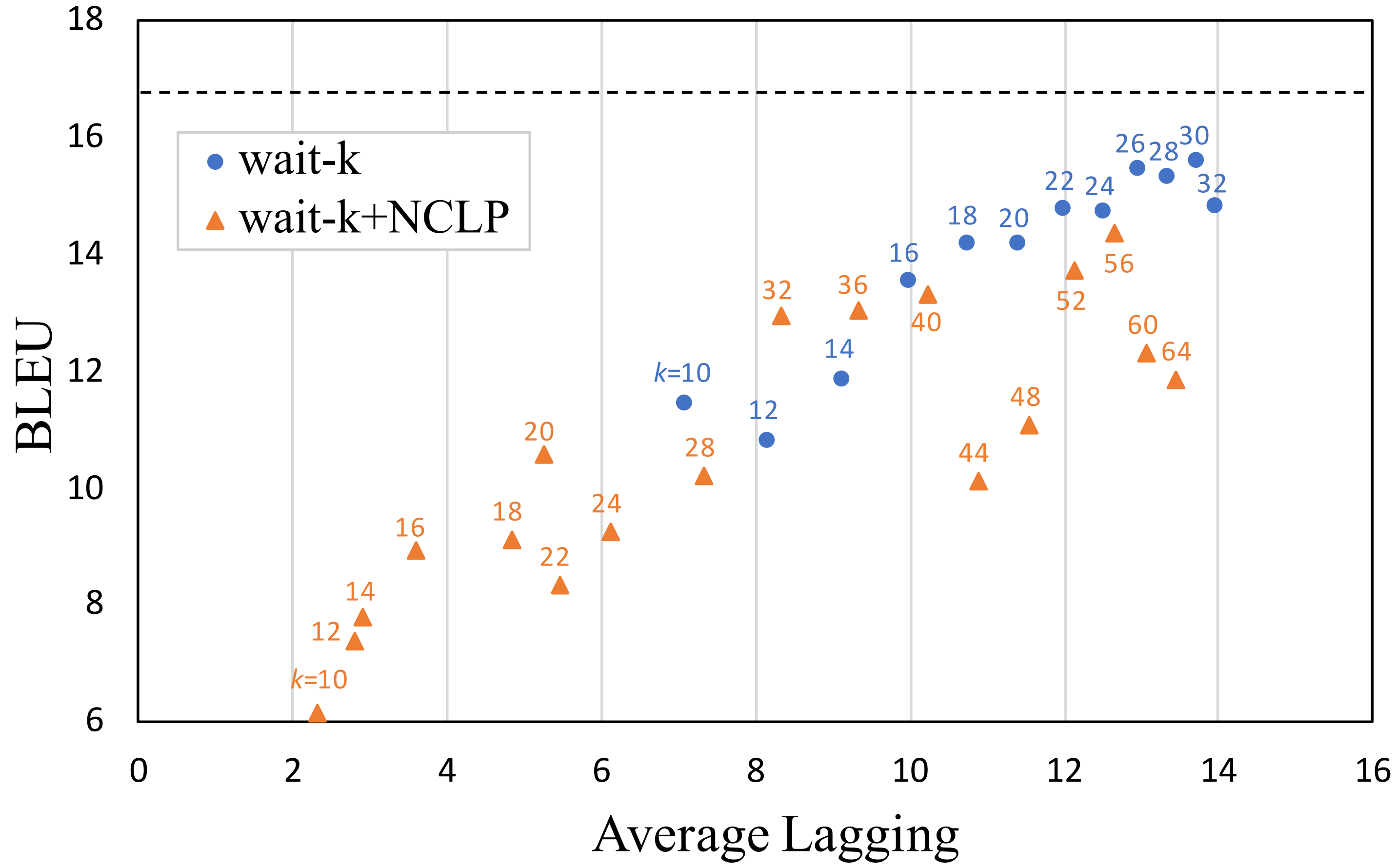
Those PP of NP us SBAR who

Those PP of NP us SBAR who SQ are

Those PP of NP us SBAR who SQ are VP under…

Results of wait-k with NCLP

- No advantages were observed…



Conclusions

- Some improvements in En-Ja simultaneous translation by:
  - Sequence-level Knowledge Distillation
    - Encouraging literal translation
  - Target-side Chunk Shuffling
    - Encouraging order-free translation
- NCLP did not work by the current way
  - Further investigation needed