

DiverSeg: Leveraging Diverse Subword Segmentations for Neural Machine Translation

2024/9/10
Haiyue Song
@CJNLP 2024

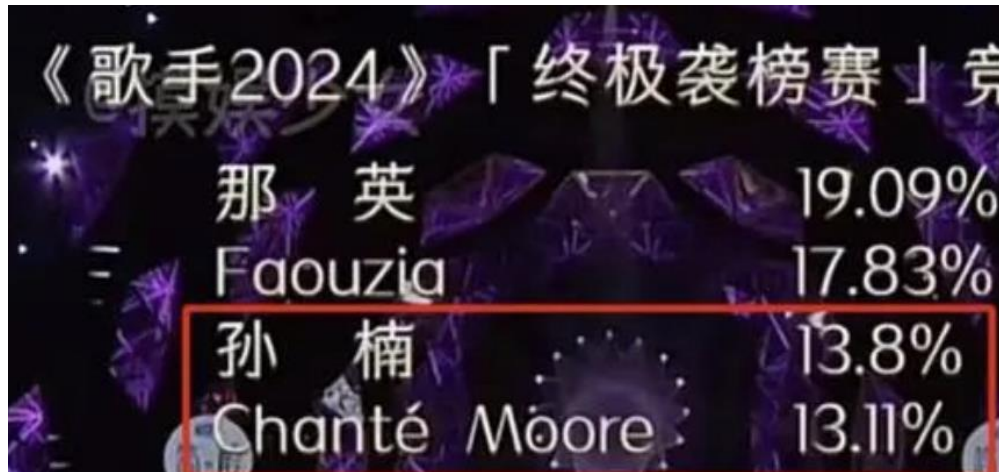


Introduction to subword segmentation

13.11 and 13.8,
which number is larger?

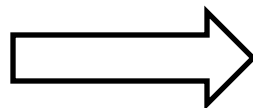
(元ネタ)

In *Singer 2024* Ranking
people in SNS found it amazing that
 $13.8\% > 13.11\%$

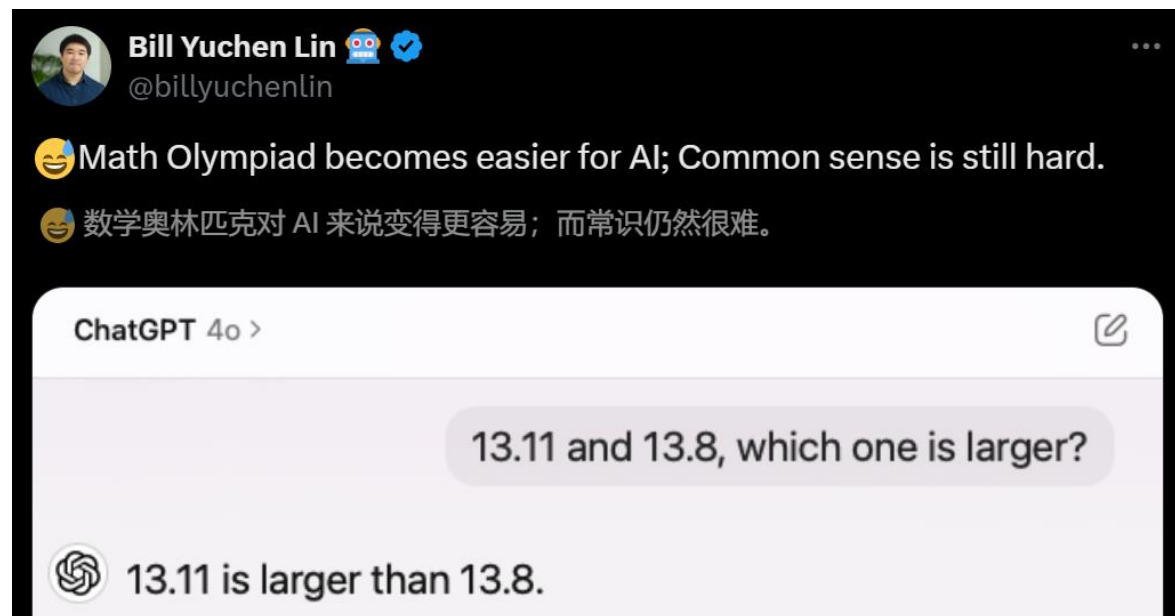


《歌手2024》「终极袭榜赛」竞

那 英	19.09%
Faouzia	17.83%
孙 楠	13.8%
Chanté Moore	13.11%



Discussions in AI community



ChatGPT 4o

13.11 and 13.8, which number is larger?



The number 13.11 is larger than 13.8.

Llama 3.1 405B



The number 13.11 is larger than 13.8.

ChatGPT 4o mini



13.11 is larger than 13.8.

Llama 2 7B chat

To determine which number is larger, we need to compare the two

13.8 > 13.11

Gemma 2B Instruct

13.8 is larger than 13.11.

WHY?

Mistral-7B-Instruct-v0.3

The number 13.8 is larger than 13.11.

<https://chatgpt.com/>

<https://www.llama2.ai/>

<https://www.llama2.space/#hg>

<https://huggingface.co/google/gemma-2b-it>

<https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>

Subword Segmentations of

all models that claim $13.11 > 13.8$ are

13.8 and 13.11 which number is larger?

with 11 as one subword, and

all models that judge $13.8 > 13.11$ are

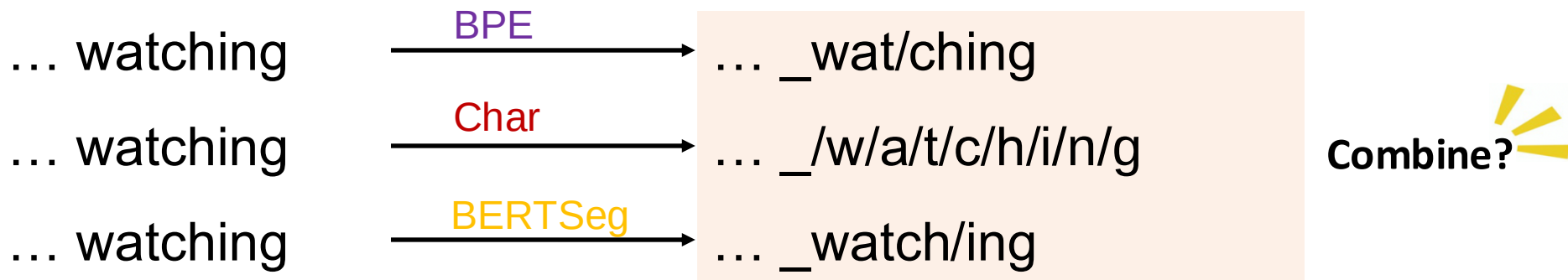
13.8 and 13.11 which number is larger?

with 1 1 as separate subwords.

Segmentation affects how models understand sentences.

Background

- Different segmenters capture different aspects of words:
 - **Byte-pair Encoding (BPE)** (Sennrich+ 16): capture high-frequency pieces
 - **Char**: good at handling rare words due to fine-grained granularity
 - **BERTSeg** (Song+ 22): capture semantic information



Research Question

Can we combine the advantages of diverse subword segmentation methods?

Related Work

- Combine in the data-level (Kambhatla+ 22)

- 1. Tokenize sentences by BPE and SentencePiece.
- 2. Copy the tokenized data to the training data.

However, not effective due to no alignment.

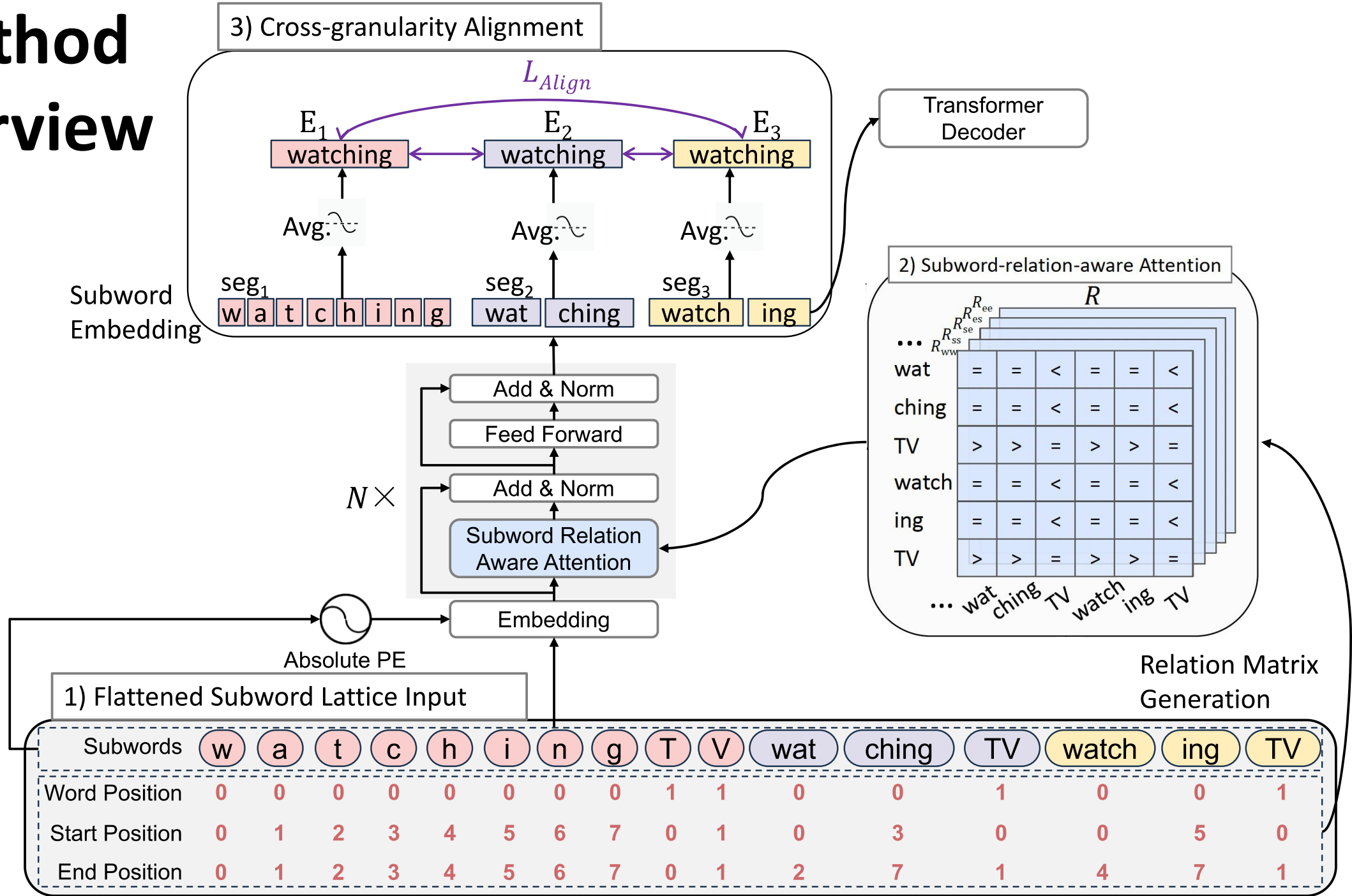
- Combine the outputs of BPE with different vocabulary sizes (different granularity) (Xiao+, 19)

However, segmentations are similar that the extra information is not limited.

Our Approach

- Combine advantages of diverse segmenters by
 - **Capture diverse information** through
 - 1) Packing segmentations by diverse segmenters during embedding
 - 2) Injecting relationships of subwords in attention
 - **Align diverse information** through
 - 3) Cross-granularity alignment in training objective

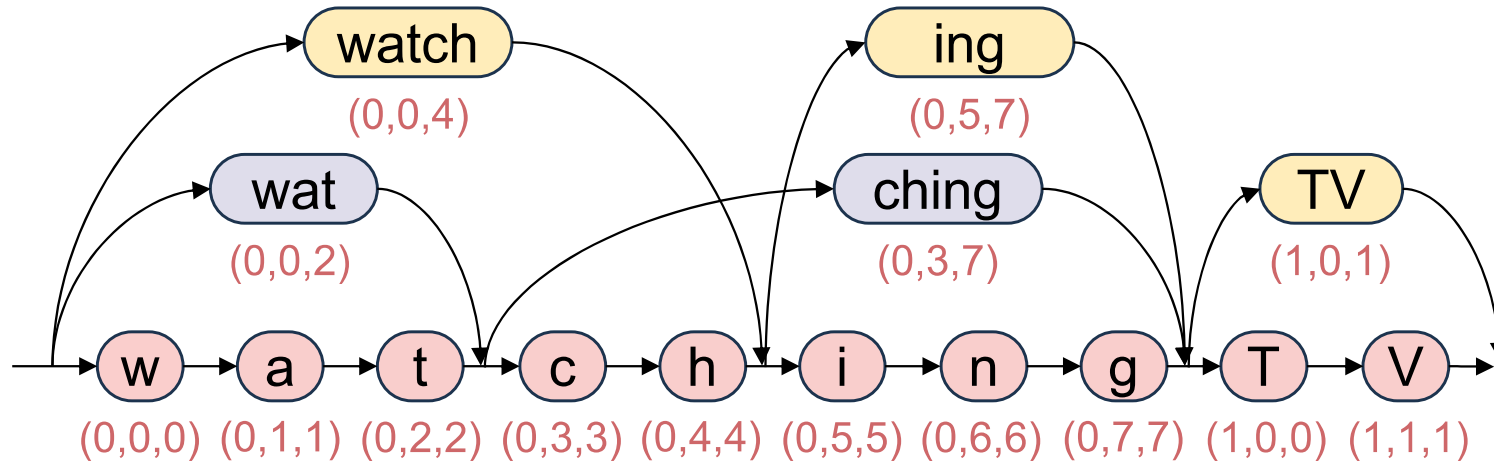
Method Overview



1) Flattened Subword Lattice Input

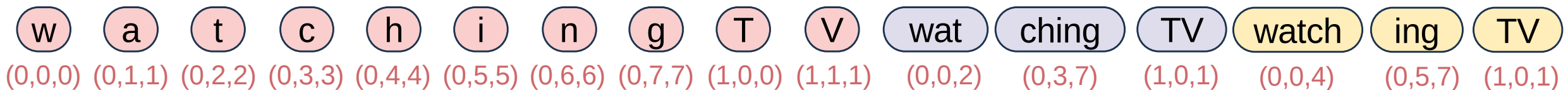
- Lattice structure to represent multiple segmentations of one word.

(Word Pos, Start Pos, End Pos)



- We make it sequential, saving subword with its positional information, which is flattened lattice structure (Li+, 2020).

- Sequential inputs are easier for transformer to process.



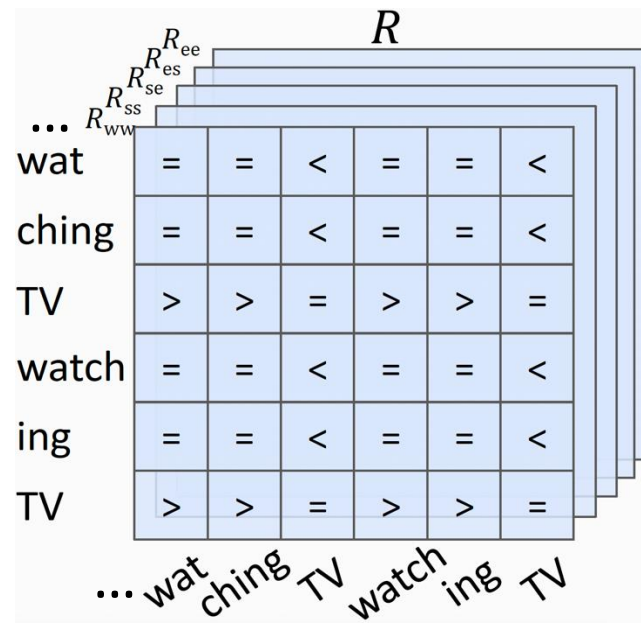
2) Subword-relation-aware Attention

- We represent **positional relations** between each two subwords in the model.
 - **Positions include** word position, start character position, and end character position.
 - **20 relations are defined** according to the combination of relative positions.

**Positional
information**

Subwords	w	a	t	c	h	i	n	g	T	V	wat	ching	TV	watch	ing	TV
Word Position	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1
Start Position	0	1	2	3	4	5	6	7	0	1	0	3	0	0	5	0
End Position	0	1	2	3	4	5	6	7	0	1	2	7	1	4	7	1

**Subword relations
calculated by
relative positions.**

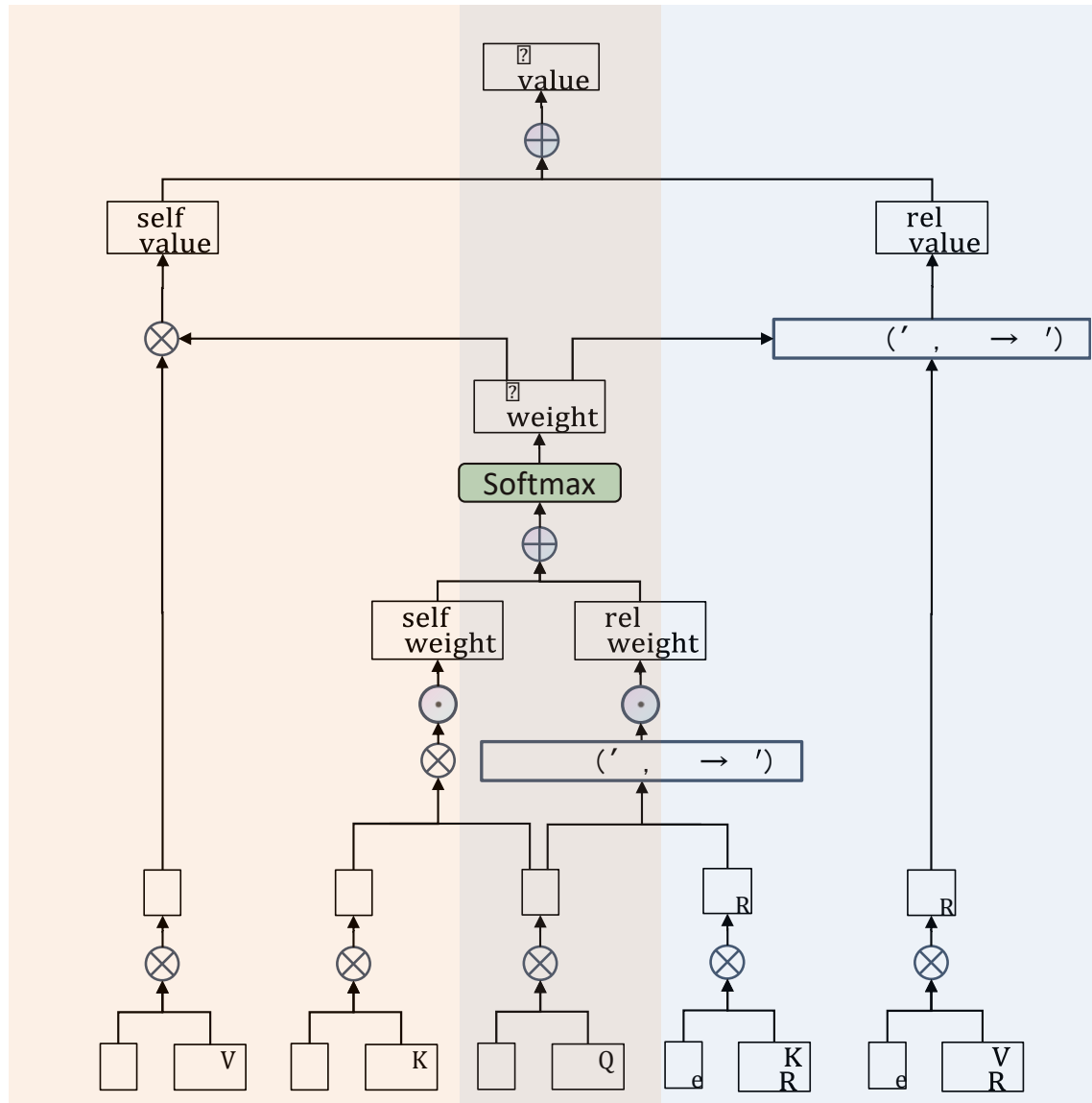


2) Subword-relation-aware Attention

- Fusing traditional self-attention and subword-relation attention.

Self-attention

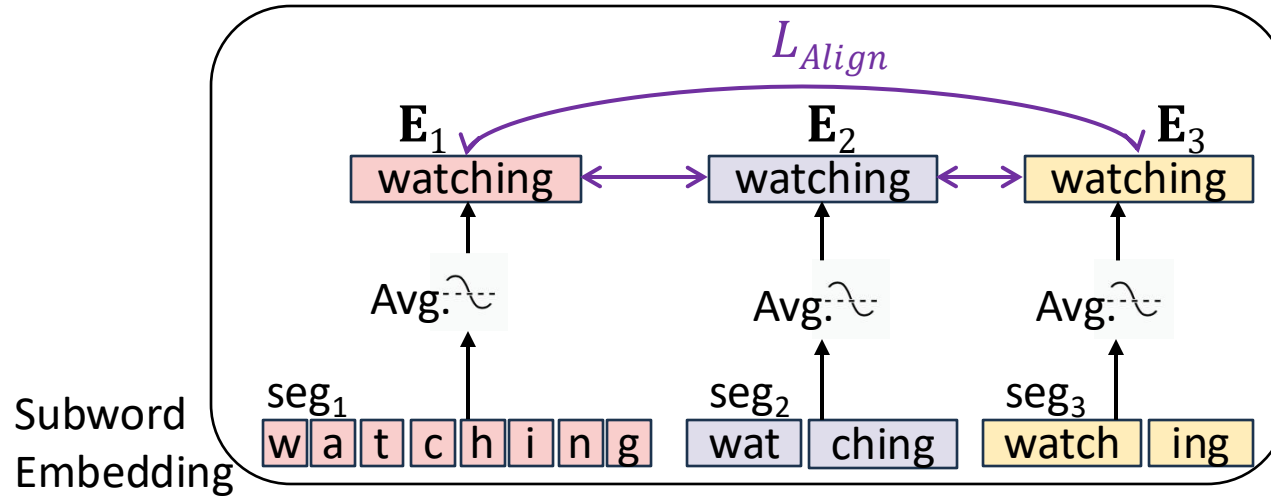
Relation-aware attention*



*Similar with "Self-Attention with Relative Position Representations"

3) Cross-granularity Alignment

- Align embeddings of subword segmentations of the same word.



- Add loss objective to maximize the **cosine similarity** ($\cos<.,.>$) among embeddings E_i of all segmentations.

$$Loss_{Align} = -\frac{1}{s(s-1)} \sum_{i=1}^s \sum_{j=1}^s \cos < E_i, E_j >$$

Experiment Settings

- Combination of segmenters
 - Character-level, BPE and BERTSeg
 - Stochastic setting: Character-level segmentation and BPE-dropout
- Baselines
 - Single segmenter
 - BPE (Sennrich+ 16), SentencePiece (Kudo+ 18), ...
 - Multiple segmenter
 - Mixing the data in the dataset (Kambhatla+ 22)
 - BPE with different vocab sizes (Xiao+ 19)
 - Stochastic segmenter
 - BPE-dropout (Provilkov+ 20)
- Model
 - Transformer base model (with conventional MT training settings)

Results: Low-Resource Scenarios

- Higher translation quality (BLEU scores) compared with baselines on the Asian Language Treebank (18k/1k/1k) dataset.

	En-Fil		En-Id		En-Ja		En-Ms		En-Vi		En-Zh		Avg.	
	→	←	→	←	→	←	→	←	→	←	→	←	→	←
<i>Single-Segmenter</i>														
BPE	18.85	16.42	22.34	19.02	10.40	5.89	26.49	21.37	20.95	15.18	7.80	6.97	17.81	14.14
SentencePiece	19.64	16.64	21.92	18.74	10.97	5.74	25.90	21.01	19.69	13.93	7.13	7.36	17.54	13.90
VOLT	18.37	16.83	22.34	19.02	10.40	5.89	26.90	21.12	21.15	14.94	8.04	7.00	17.87	14.13
DPE	19.23	15.77	22.81	19.71	10.45	5.97	27.46	22.70	20.02	15.16	8.36	7.44	18.06	14.46
BERTSeg	20.08	17.75	21.87	20.04	10.93	6.56	25.46	21.92	20.95	15.51	8.07	7.91	17.89	14.95
<i>Multiple-Segmenter</i>														
Kambhatla et al. (2022)	13.54	11.06	14.99	13.16	7.30	5.58	18.79	15.24	14.29	10.79	3.84	6.96	12.13	10.47
Xiao et al. (2019)	20.86	18.79	23.70	19.81	10.46	5.87	27.10	24.05	22.32	16.91	8.01	7.64	18.74	15.51
DiverSeg	♣ 22.89	18.28	♣ 25.22	♣ 20.80	10.72	♣ 7.73	♣ 30.51	♣ 26.28	♣ 24.02	16.49	8.60	♣ 8.52	♣ 20.33	♣ 16.35
<i>Stochastic Method</i>														
BPE-dropout	29.14	20.40	31.47	28.31	13.48	11.12	35.93	32.03	27.86	21.96	9.98	12.42	24.64	21.04
DiverSeg w/ BPE-dropout	♣ 30.61	♣ 30.29	♣ 34.27	♣ 30.45	♣ 17.38	♣ 13.17	♣ 39.52	32.28	♣ 30.29	♣ 24.80	♣ 12.88	♣ 13.90	♣ 27.49	♣ 24.15

♣ denotes $p < 0.01$

Results: Higher-Resource Scenarios

- All methods show similar performance.
- Subword segmentation is no more bottleneck given enough data.

	IWSLT'15 En-Vi 133k → ←		WMT'16 En-Ro 0.6M → ←		WMT'15 En-Fi 1.8M → ←		WMT'14 En-De 4.5M → ←	
<i>Single-Segmenter</i>								
BPE	26.23	22.36	32.48	31.93	14.94	17.10	25.13	30.27
SentencePiece	26.60	22.68	32.10	32.24	14.97	17.45	25.30	30.00
VOLT	25.87	22.28	31.82	31.54	14.27	17.06	25.35	30.56
DPE	26.45	23.78	32.07	31.89	14.67	17.44	25.30	30.55
BERTSeg	26.83	23.10	32.36	32.24	15.07	17.83	24.74	29.71
<i>Multiple-Segmenter</i>								
Kambhatla et al. (2022)	27.14	23.89	32.52	32.34	15.04	17.79	24.96	30.33
Xiao et al. (2019)	28.67	24.91	32.08	31.29	13.73	16.44	25.26	30.07
DiverSeg	♠ 29.19	♠ 25.78	32.68	32.47	14.89	17.85	25.58	30.36
<i>Stochastic Method</i>								
BPE-dropout	30.29	28.06	34.91	34.76	16.32	19.05	24.93	30.39
DiverSeg w/ BPE-dropout	♠ 30.84	28.03	34.62	34.07	15.97	18.94	-	-

♣ denotes p<0.05

Ablation Study

■ Improvement (BLEU scores) brought by each component

- Flattened Subword Lattice Input: **0.93**
- Subword-relation-aware attention: **0.88**
- Cross-granularity alignment: **0.38**

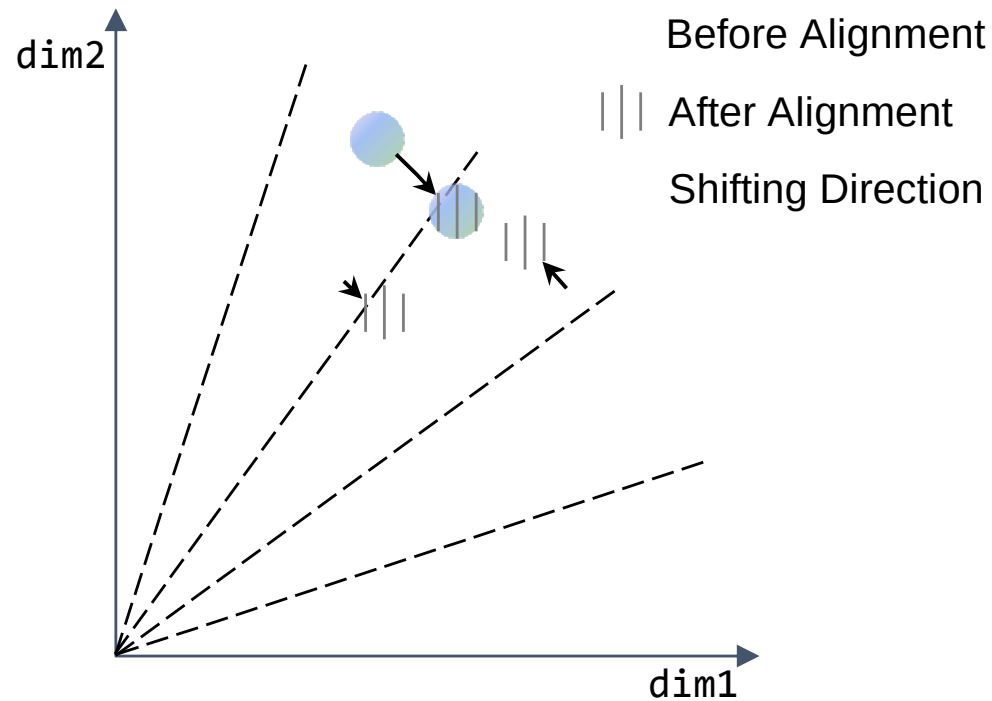
	Lattice PE	Subword Attn	$\mathcal{L}_{\text{Align}}$	BLEU	BLEURT	METEOR
3)	✓	✓	✓	29.19	56.20	33.78
	✓	✓	✗	28.81	55.44	33.51
2)	✓	✗	✓	28.31	55.11	33.23
	✓	✗	✗	28.45	54.40	33.49
1)	✗	✓	✓	28.26	55.61	32.97
	✗	✓	✗	27.52	54.11	32.77
	✗	✗	✓	12.48	35.84	24.79
	✗	✗	✗	13.85	37.64	26.04

IWSLT'15 En->Vi translation direction

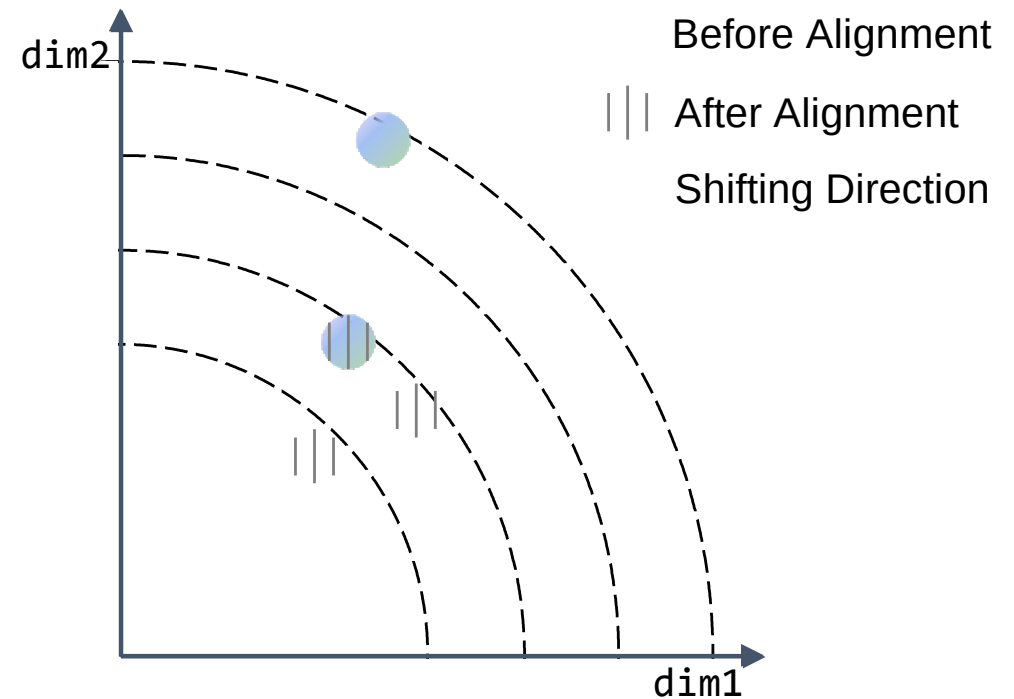
How to Align Embeddings?

- Q: Why maximizing cosine similarity instead of minimizing distance?
- A: Embedding vectors shifted towards origin when we minimize L1/L2.

Maximizing Cosine Similarity



Minimizing L1/L2 Distance



Effect of Align Embeddings Methods

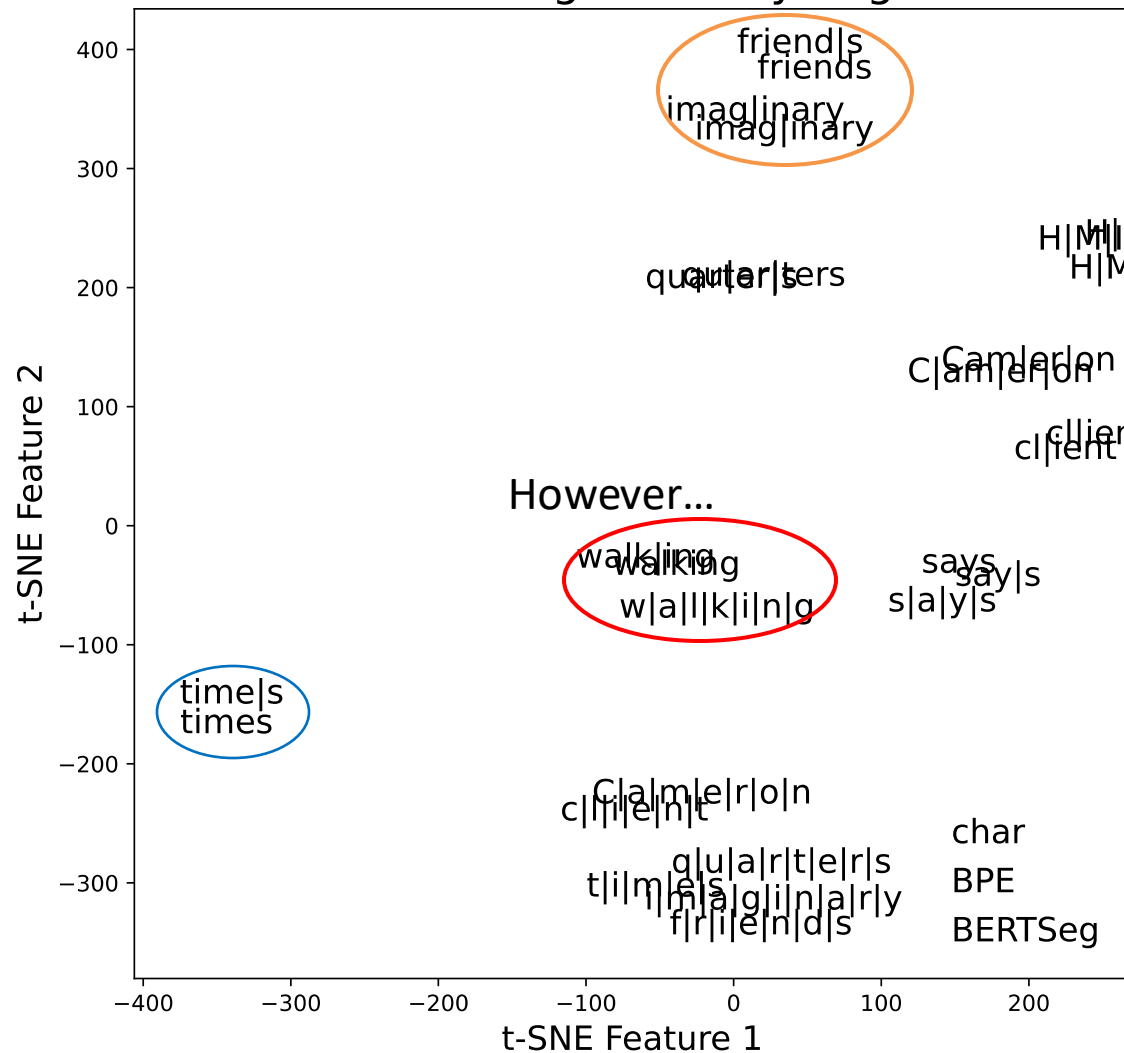
- While normalizing L1/L2 distance alleviates the problem,
- Cosine similarity is the most effective/robust method.

	BLEU	BLEURT	METEOR
Cosine Similarity	29.19	56.20	33.78
L1 distance	0.81	20.30	10.97
L2 distance	21.67	45.28	30.23
Norm L1 distance	24.35	48.86	31.40
Norm L2 distance	28.57	55.61	33.69

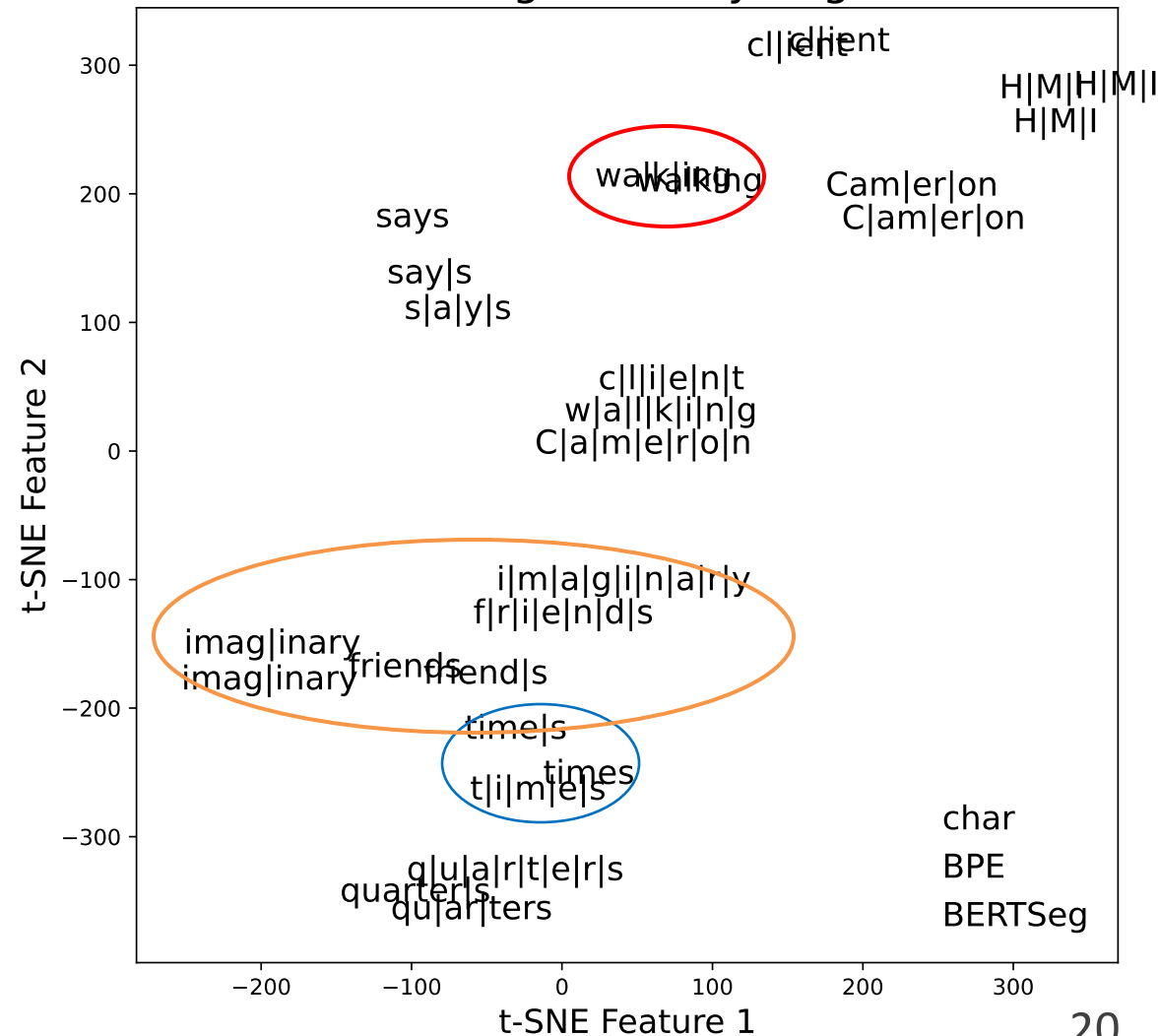
BLEU results for the IWSLT'15 En->Vi direction.

Visualization of Embeddings Alignment

Before Without Cross-granularity Alignment



After With Cross-granularity Alignment



Conclusion & Discussion

- We utilize diverse segmentations for NMT through
 - Flattened lattice input
 - Subword-relation-aware attention
 - Cross-granularity alignment
- Experiments show
 - Higher translation performance for low-resource scenarios
 - The improvement brought by each component
- Let's improve subword segmentations in LLMs :)
 - Balance between efficiency and performance
 - Multilingualism, etc