Attention is Not Only a Weight: Analyzing Transformers with Vector Norms

Goro Kobayashi[1], Tatsuki Kuribayashi[1,2], Sho Yokoi[1,3], Kentaro Inui[1,3]

[1] Tohoku University, [2] Langsmith Inc., [3] RIKEN

EMNLP 2020
November 16-18, 2020

Background

Transformers have been successfully applied to a wide range of NLP tasks.

- Transformer[Vaswani+’17], BERT[Devlin+’19], RoBERTa[Liu+’19], etc.

GLUE (Leaderboard on October 19)

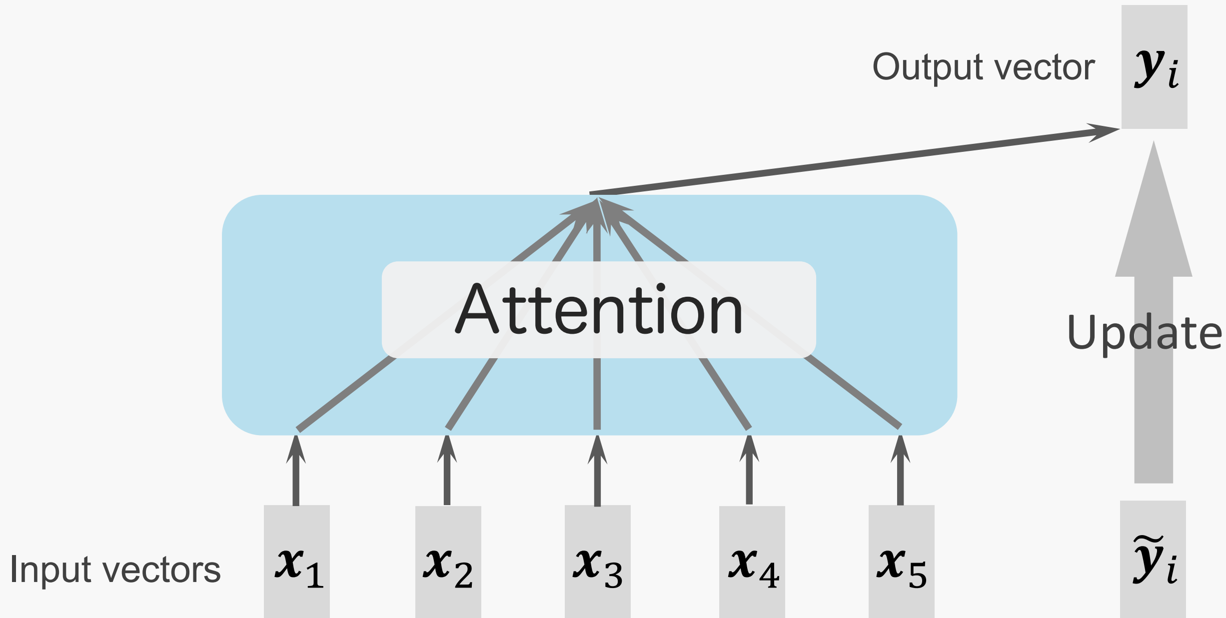
	Rank	Name	Model	URL	Score
	1	HFL iFLYTEK	MacALBERT + DKM		90.7
+	2	Alibaba DAMO NLP	StructBERT + TAPT		90.6
+	3	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
	4	ERNIE Team - Baidu	ERNIE		90.4
	5	T5 Team - Google	T5		90.3

https://gluebenchmark.com/leaderboard

Attention: Key component in Transformers

Attention

- Updates each vector by **mixing** the inputs focusing on relevant information
- **"How attention mixes inputs"** has been investigated from attention weights [Clark+'19;Kovaleva+'19; etc.]

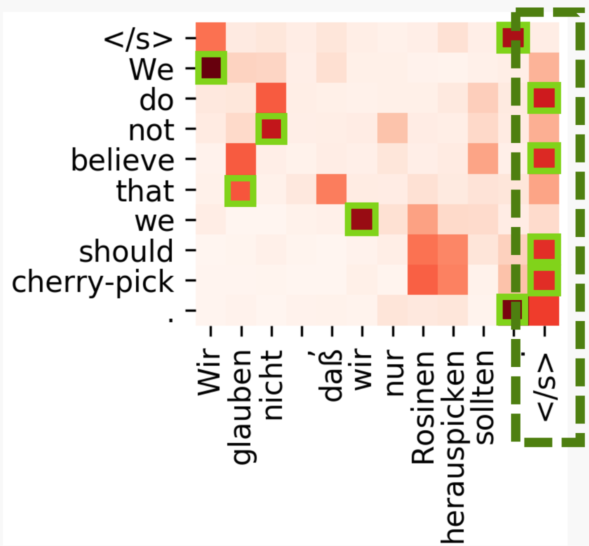


Overview

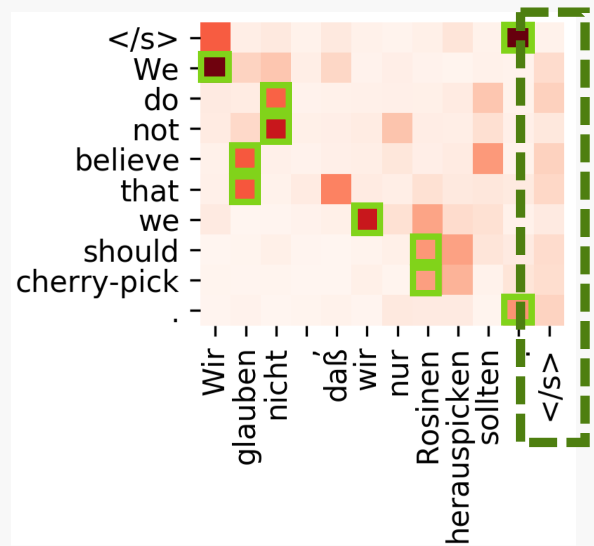
Propose to analyze Transformers using **vector norms** instead of **attention weights**

- Able to consider more from the process within attention
- Intuitive results than those from attention weights

Attention weights

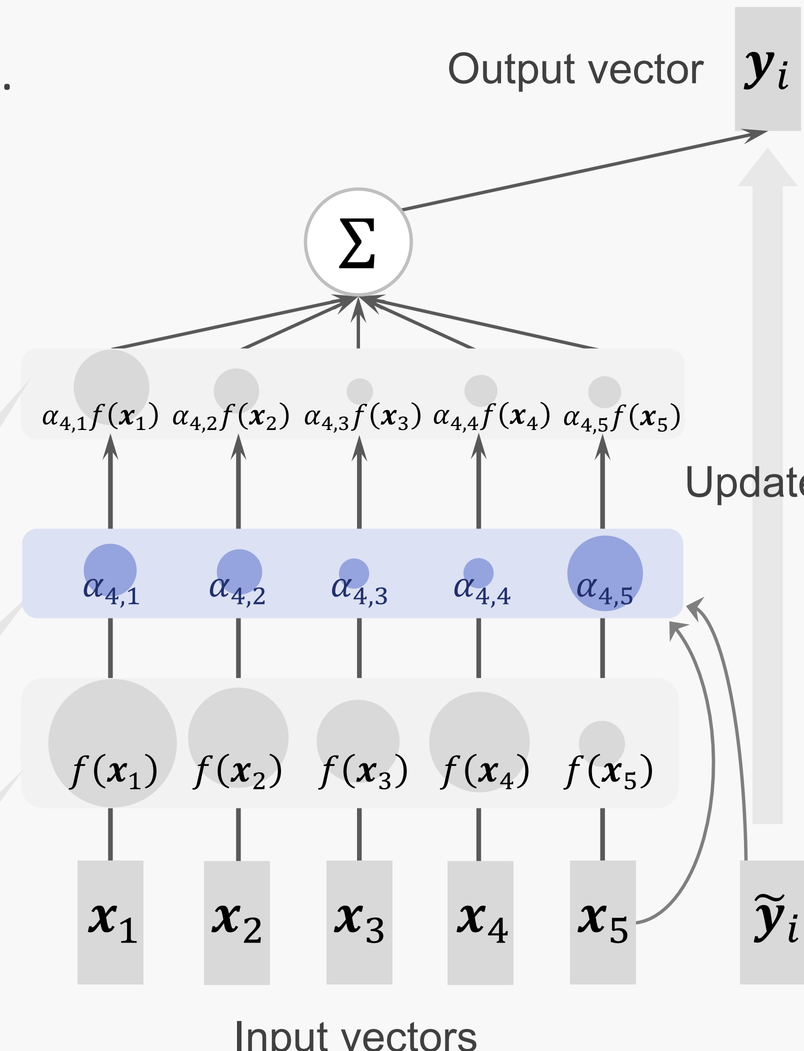


Vector norms (ours)



Attention performs a weighted sum of vectors

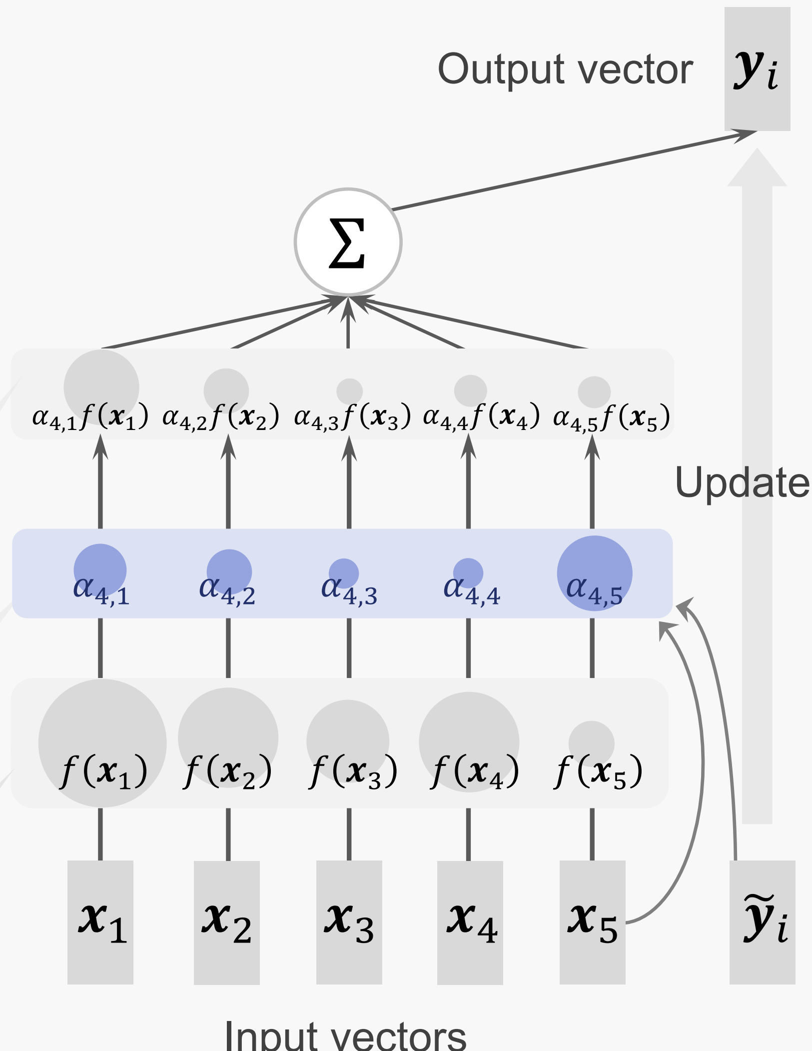
Attention mechanism consists of **3-step process**.



3. Summation
2. Weighting
1. Affine transformation (including transformation to Value vectors)

Attention performs a weighted sum of vectors

Attention mechanism consists of **3-step process**.



3 Summation

2 Weighting

1 Affine transformation (including transformation to Value vectors)

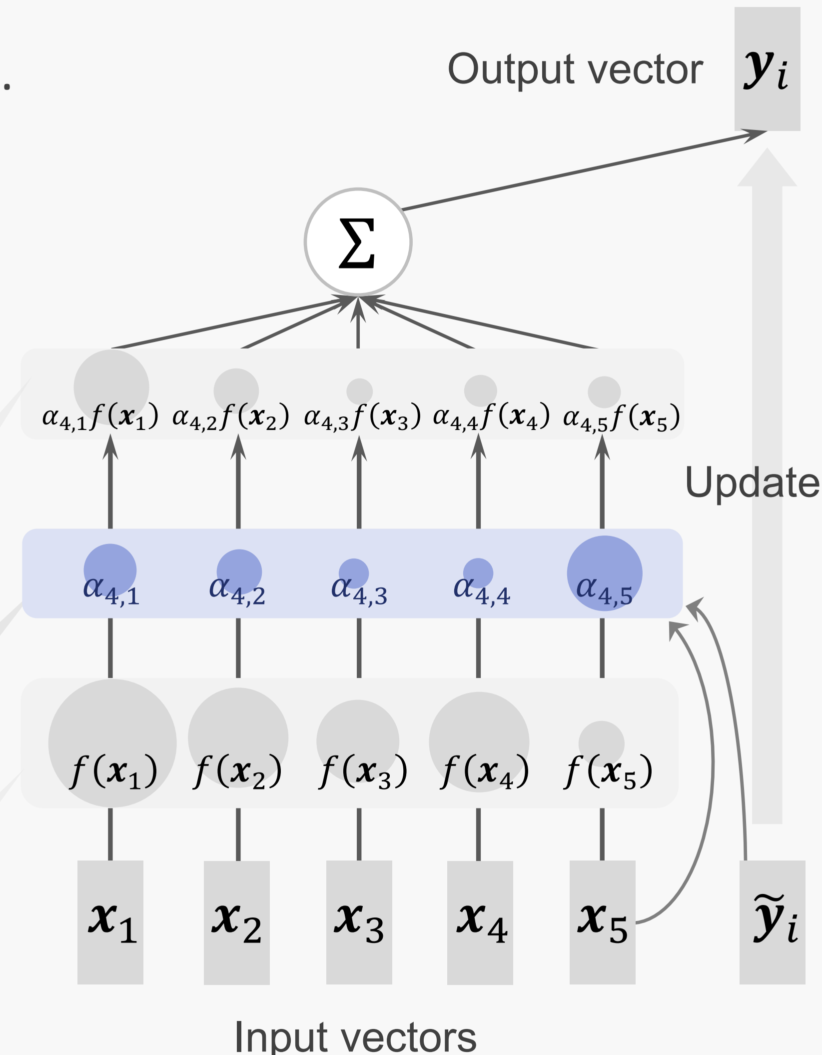
Attention performs a weighted sum of vectors

Attention mechanism consists of **3-step process**.

3 Summation

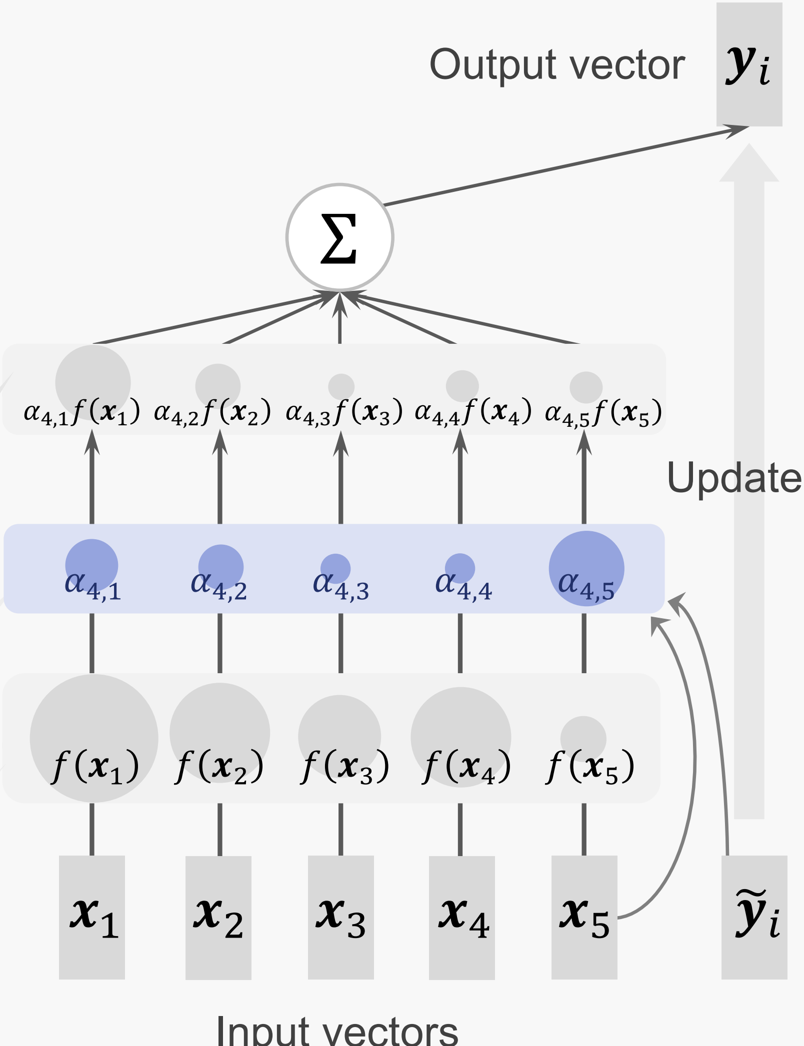
2 Weighting

1 Affine transformation (including transformation to Value vectors)



Attention performs a weighted sum of vectors

Attention mechanism consists of **3-step process**.



3 Summation

2 Weighting

1 Affine transformation (including transformation to Value vectors)

Attention performs a weighted sum of vectors

Attention mechanism consists of **3-step process**.

Output:

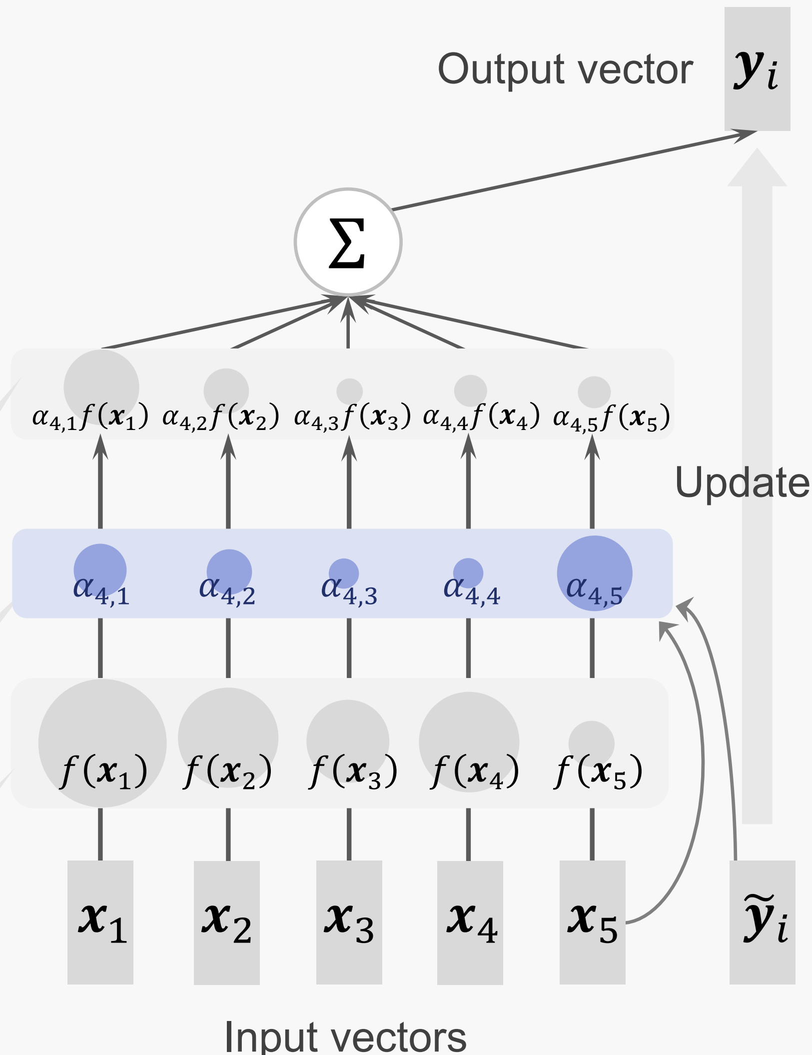
Weighted sum of transformed vectors

$$y_i = \sum_j \alpha_{i,j} f(\boldsymbol{x}_j)$$

3 Summation

2 Weighting

1 Affine transformation (including transformation to Value vectors)

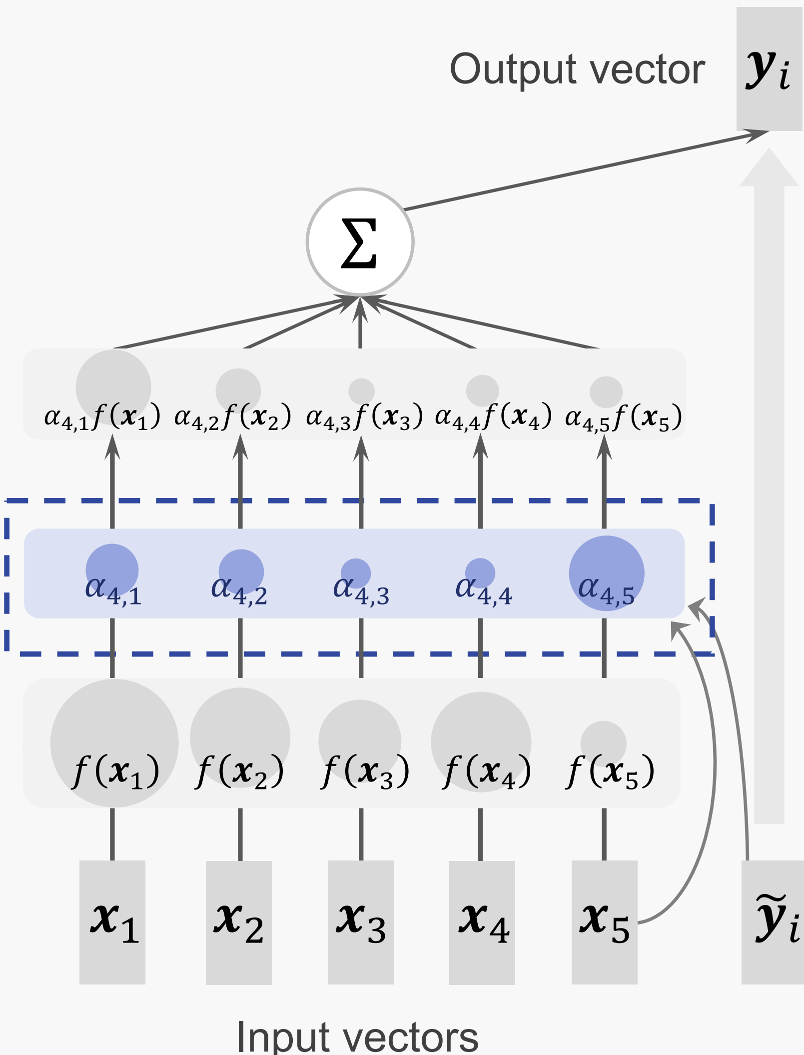


Mixed amount ≠ Attention weight

Attention weight analysis
[Clark+’19;Kovaleva+’19;Reif+’19;etc.]

$$y_i = \sum_j \alpha_{i,j} f(x_j)$$

☹ Ignore the effect of transformed vector $f(x)$



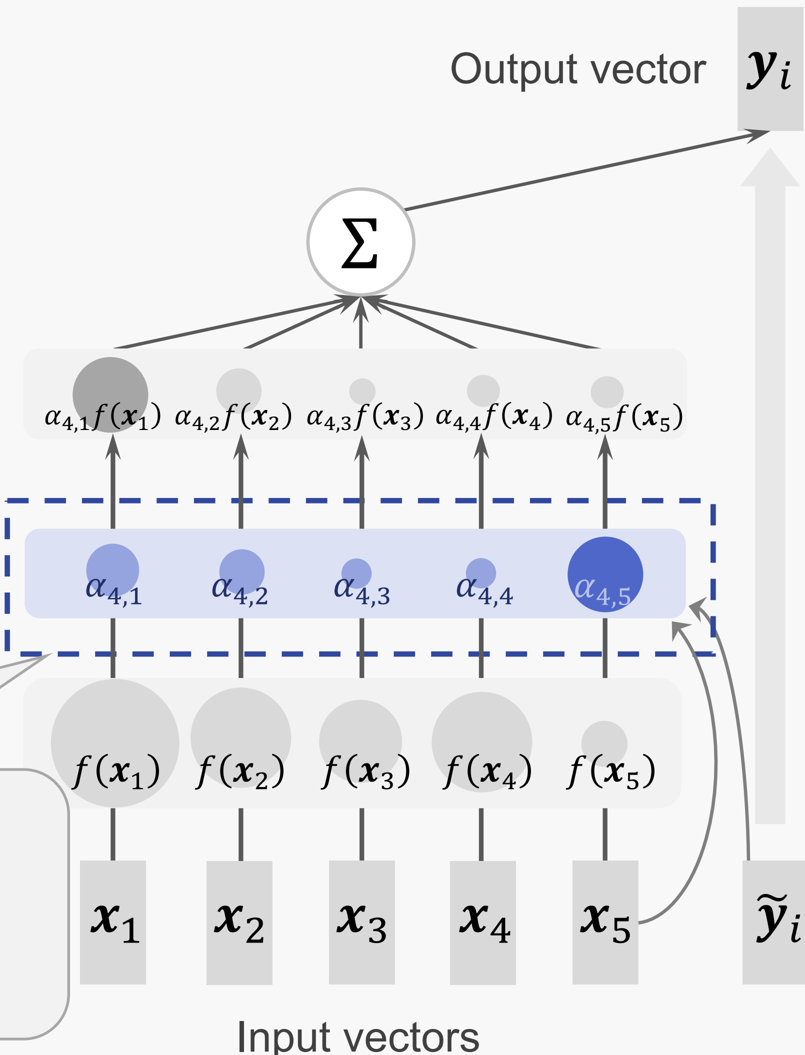
Mixed amount ≠ Attention weight

Attention weight analysis
[Clark+'19;Kovaleva+'19;Reif+'19;etc.]

$$\boldsymbol{y}_i = \sum_j \alpha_{i,j} f(\boldsymbol{x}_j)$$

☹ Ignore the effect of transformed vector $f(x)$

misunderstand that attention gathers a lot from $\boldsymbol{x}_5$ to generate $\boldsymbol{y}_i$ even if $\alpha f(\boldsymbol{x}_1)$ is predominant in $\boldsymbol{y}_i$



Proposal: Norm analysis
Measure the norm of the vector actually summed

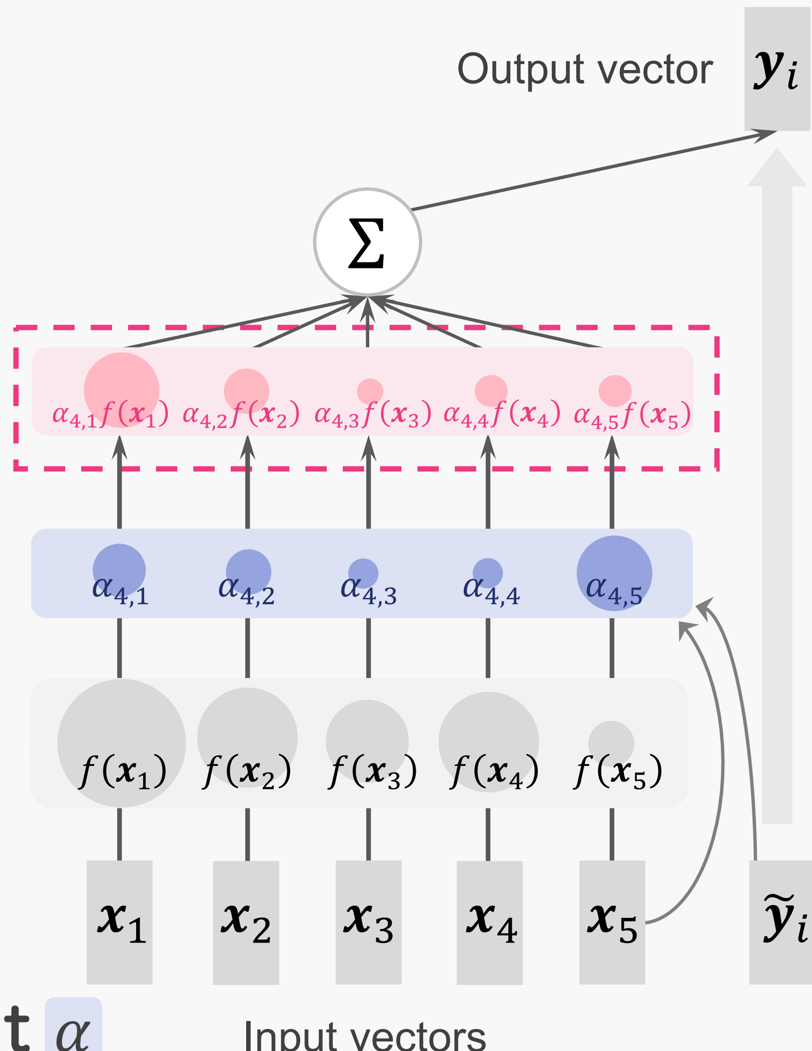
Propose a new analysis

- Focus on **the vector to be actually summed**

$$y_i = \sum_j \alpha_{i,j} f(x_j)$$

- Measure the mixed amount of each input by **norm** $\|\alpha_{i,j} f(x_j)\|$

Consider the vector $f(x)$ in addition to attention weight α



Proposal: Norm analysis
Measure the norm of the vector actually summed

Propose a new analysis

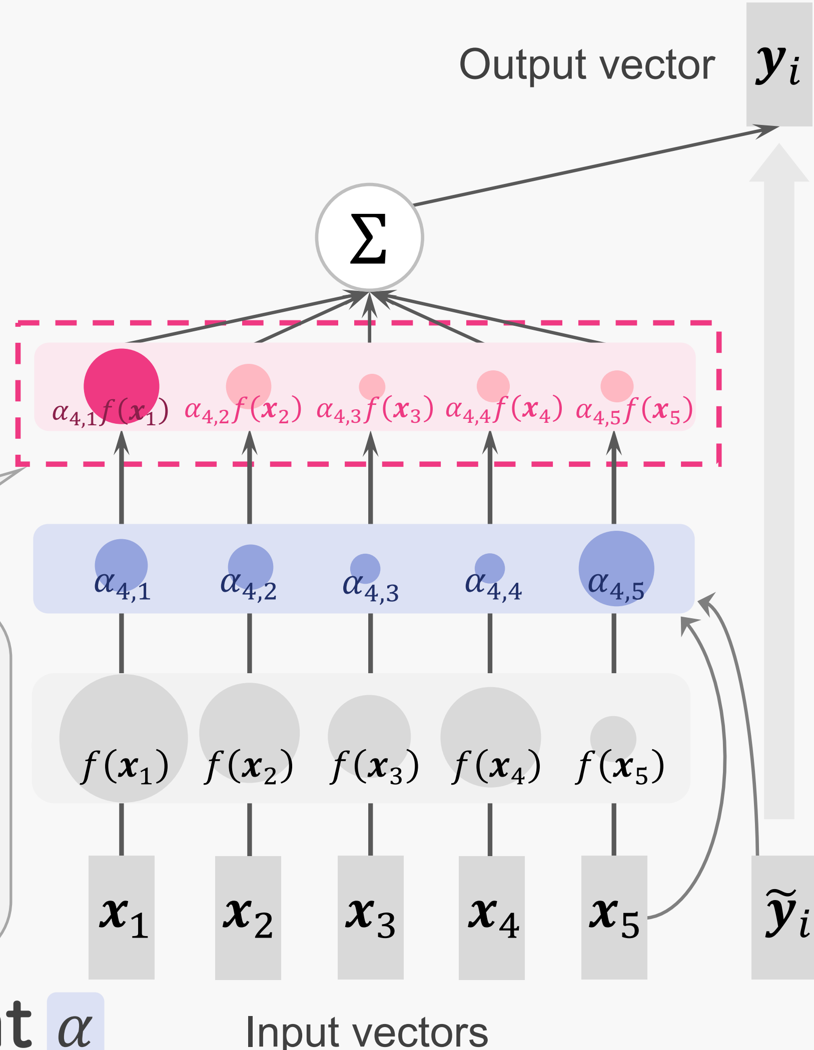
- Focus on **the vector to be actually summed**

$$\boldsymbol{y}_i = \sum_j \alpha_{i,j} f(\boldsymbol{x}_j)$$

- Measure the mixed … of

correctly interpret that attention gathers the most from $\boldsymbol{x}_1$ to generate $\boldsymbol{y}_i$ (a little from $\boldsymbol{x}_5$)

Consider the vector $f(\boldsymbol{x})$ in addition to attention weight α



Experiment 1: BERT

Experiment 1: BERT --- Setup

Investigate the behavior of attention with previous and proposed methods

- Models
  - **pre-trained BERT-base** (uncased)
    - 12 layers, 12 head (total of 144 self-attentions in the model)
- Data
  - 992 segments extracted from Wikipedia [Clark+'19]

https://github.com/clarkkev/attention-analysis

token used for classification tasks

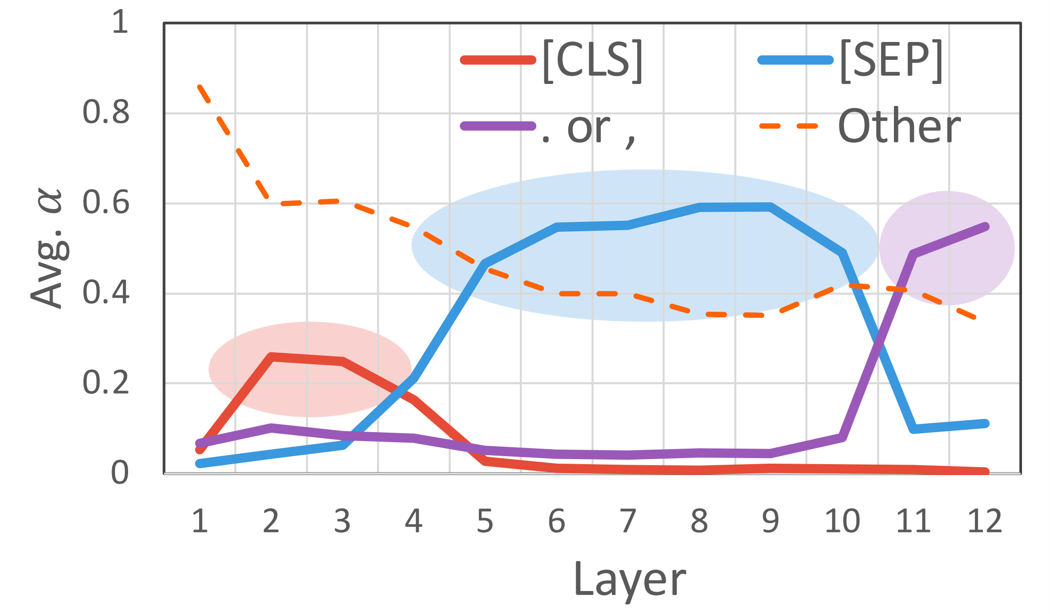
separator tokens

Input segment: [CLS] paragraph1 [SEP] paragraph2 [SEP]

Previous result of attention weight analysis

[Clark+'19]

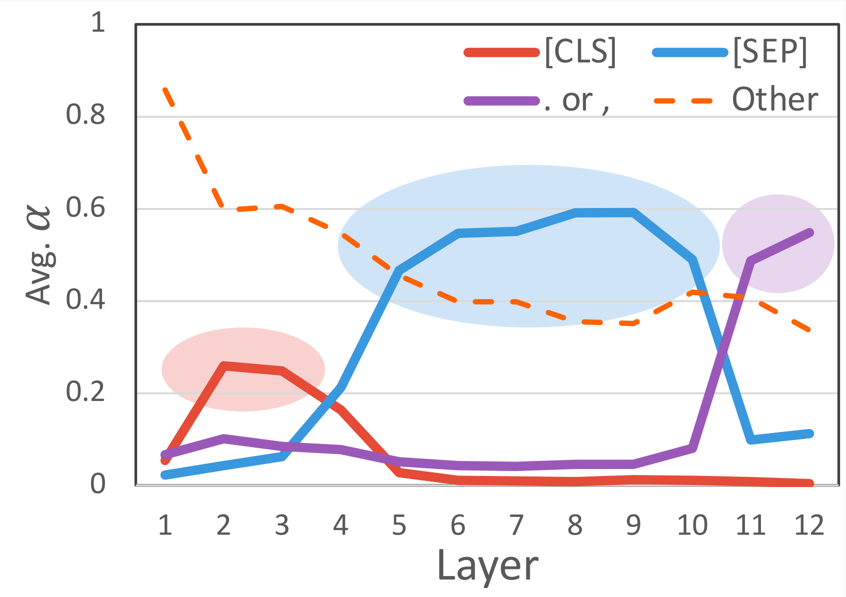
Average attention weight in each layer



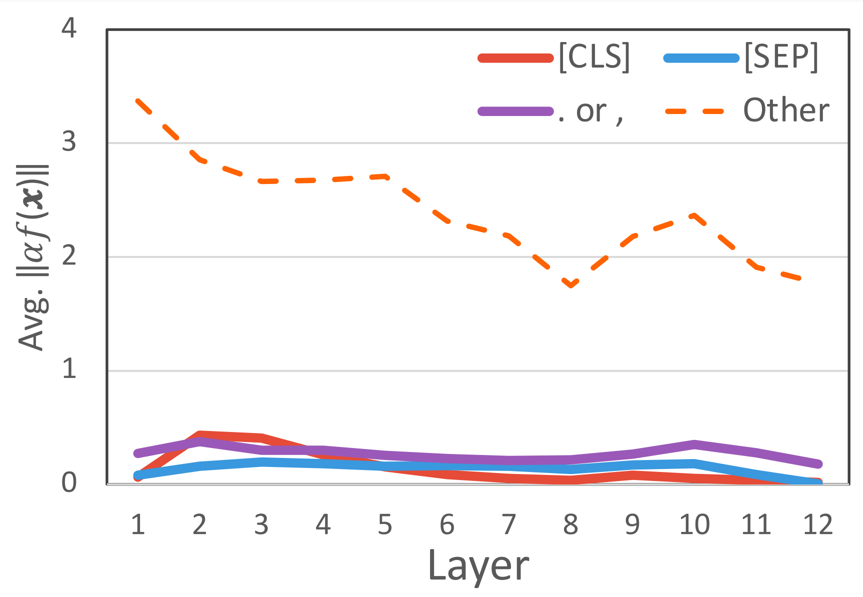
- Attention weights are biased towards specific token categories
  - Early layers --> [CLS]
  - Middle layers --> [SEP]
  - Deep layers --> periods or commas

Different results between the methods

Attention weight analysis [Clark+'19]



Proposed norm-based analysis



Largely different results

- Self-attention gathers only a little from special tokens, periods, and commas, and most from the other tokens.

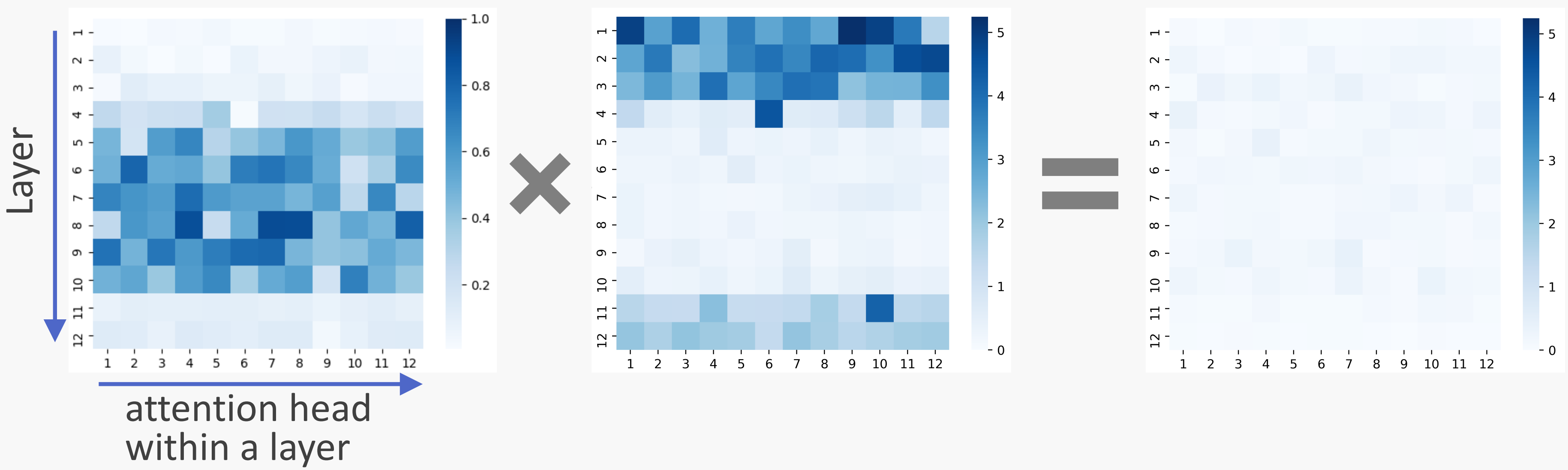
Detailed analysis ([SEP])

Why $\|\alpha f(\boldsymbol{x})\|$ is small despite its large weight α?

α
Attention weight
(previous analysis)

$\|f(\boldsymbol{x})\|$
Norm of transformed vector
(previously ignored element)

$\|\alpha f(\boldsymbol{x})\|$
Norm of weighted vector
(proposed analysis)



- Attention weight α and norm of transformed vector $\|f(\boldsymbol{x})\|$ cancel each other out
  - Same tendency for [CLS], periods, and commas

Experiment 2: Transformer NMT model

Experiment 2: Transformer NMT model --- Setup

Compare the quality of word alignments extracted from attention by two approaches: **weight** and **norm**

- Alignments induced from **attention weight** $\boldsymbol{\alpha}$ have empirically been shown **noisy** [Li+’19; Zenkel+’19; Ding+’19]
- Hypothesis: much cleaner alignments can be extracted from **norm** $\|\boldsymbol{\alpha f}(\boldsymbol{x})\|$

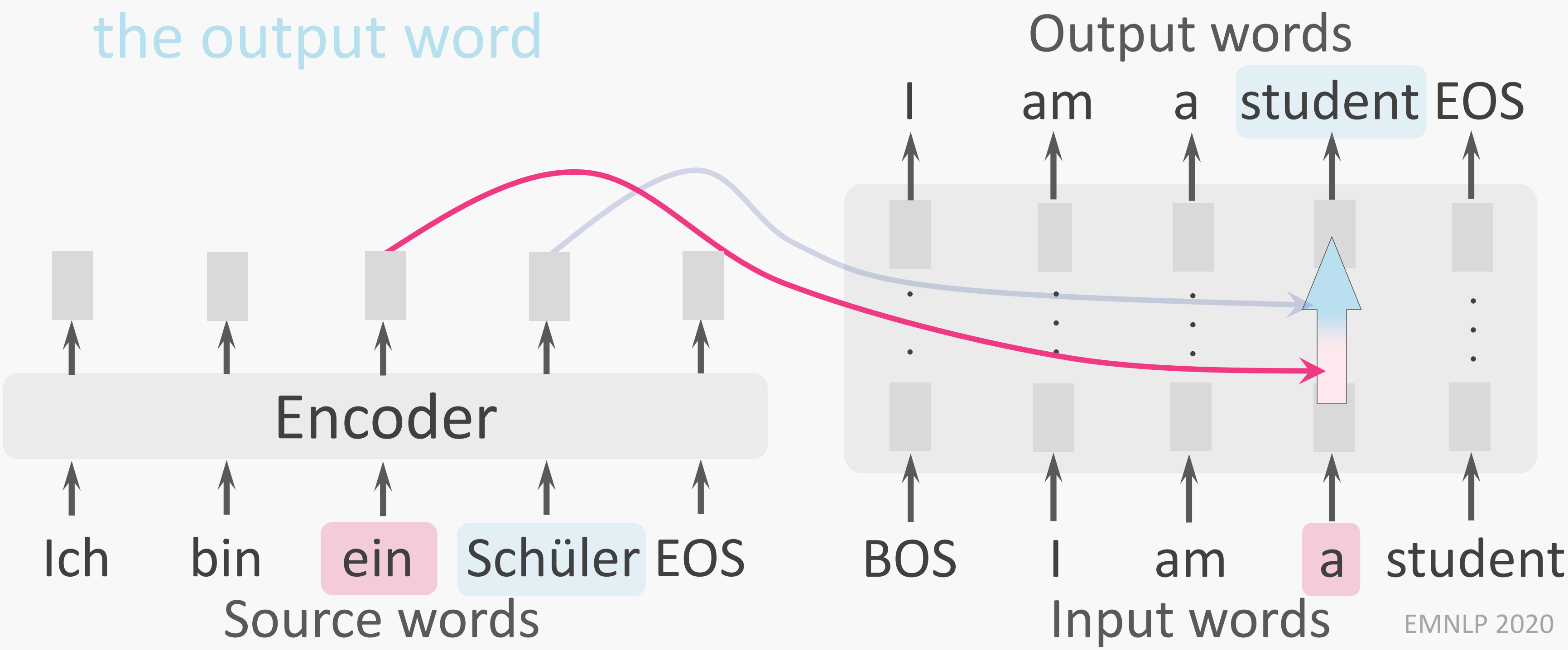
- Model (see the paper for detailed settings)
  - **Transformer** (German-English, 6 layers, 4 heads)

- Alignment extraction
  - Extract the source word with the highest weight α or norm $\|\alpha f(\boldsymbol{x})\|$ as the alignment target

Preliminary observation: Behavior of attention differs in layers

From preliminary observation,

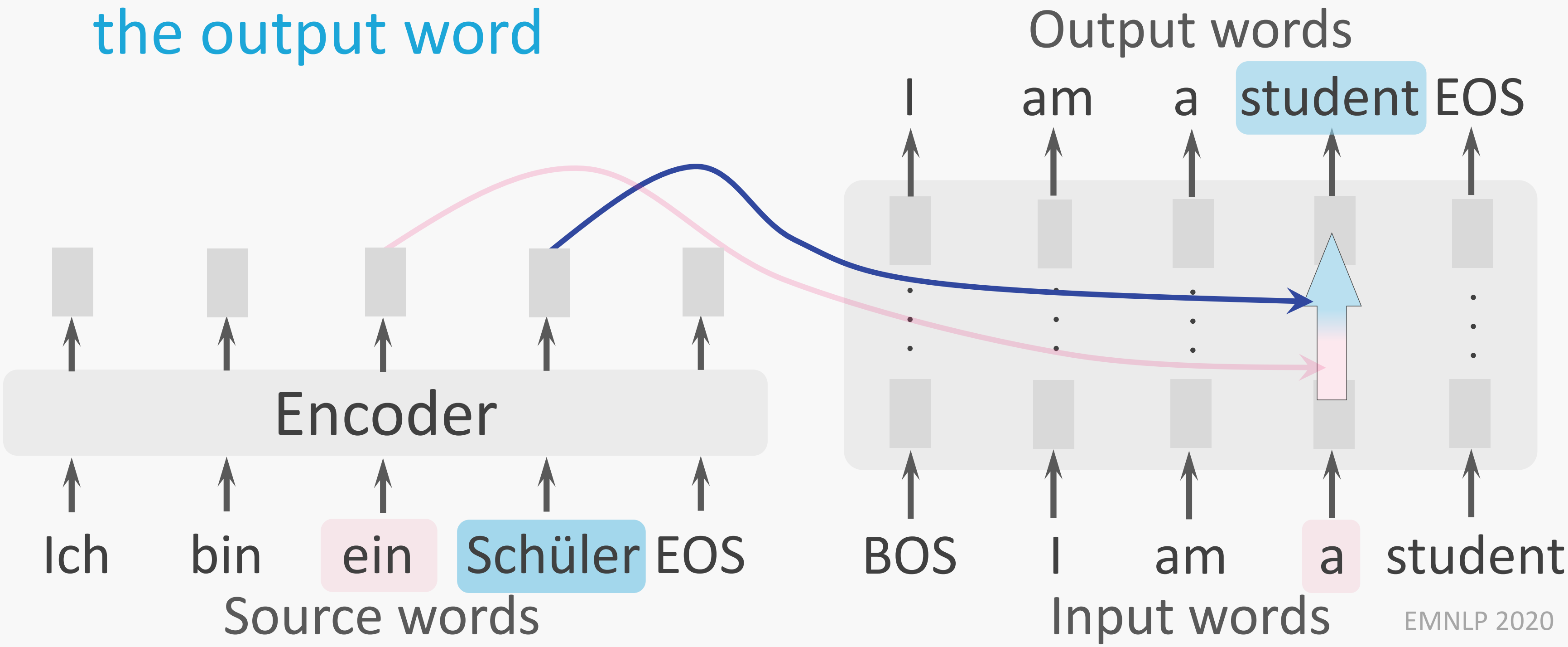
- Earlier layers focus on a source word corresponding to the input word
- Latter layers focus on a source word corresponding to the output word



Preliminary observation: Different layers focus on different words

From preliminary observation,

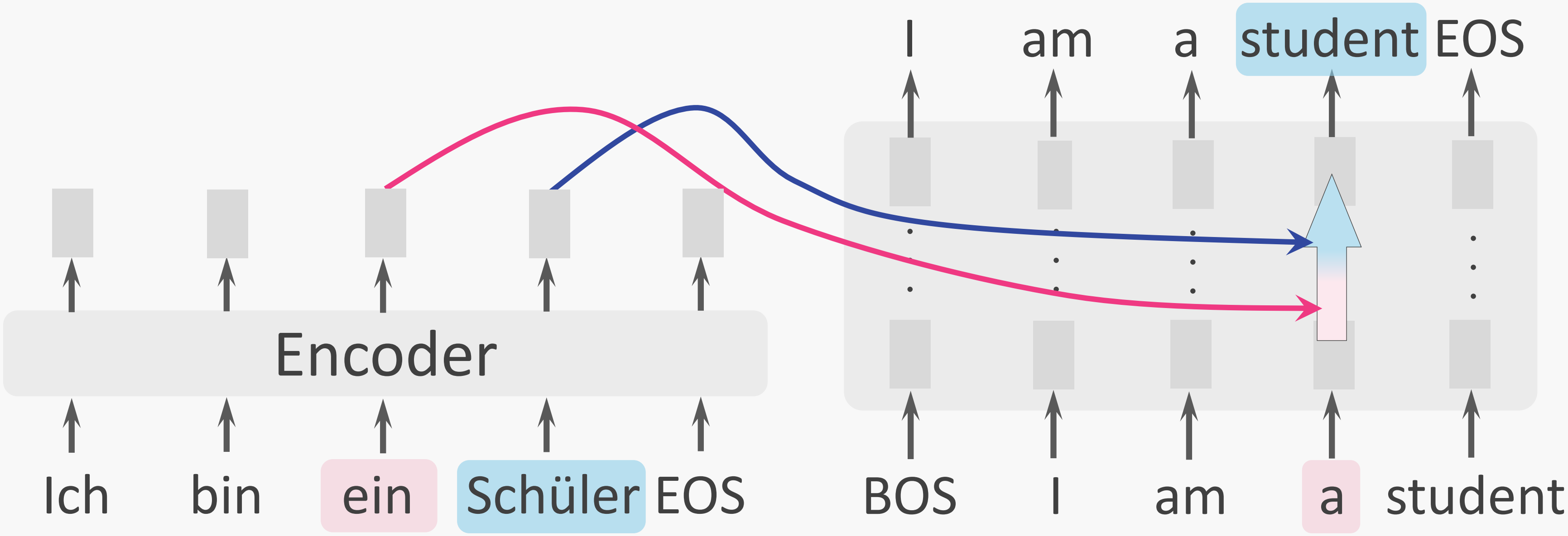
- Earlier layers focus on a source word corresponding to the input word
- Latter layers focus on a source word corresponding to the output word



2 settings: Alignment with input/output

Explored two settings for alignment extraction:

- Alignment with output setting
  - Extract the source word as alignment target of the output word
- Alignment with input setting
  - Extract the source word as alignment target of the input word



Results: Alignment Error Rate (lower is better)

		Alignment error rate	
		Alignement with output	Alignment with input
Attention weight	layer mean	68.4	68.6
	best layer	47.7 (layer 4 or 5)	29.8 (layer 5)
Norm (Ours)	layer mean	62.9	60.5
	best layer	41.4 (layer 2)	25.0 (layer 2)

- Possible to extract cleaner word alignments from **norms** than **weights**

Results:
Alignment Error Rate (lower is better)

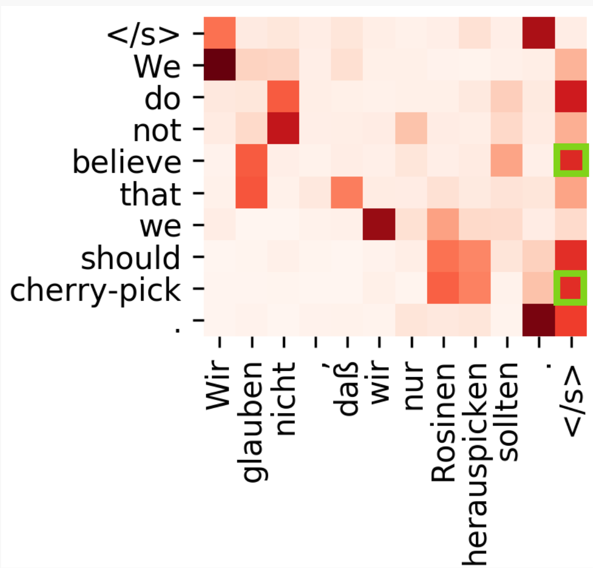
		Alignment error rate	
		Alignement with output	Alignment with input
Attention weight	layer mean	68.4	68.6
	best layer	47.7 (layer 4 or 5)	29.8 (layer 5)
Norm (Ours)	layer mean	62.9	60.5
	best layer	41.4 (layer 2)	**25.0** (layer 2)

		Alignment error rate
Word aligner	fast_align	28.4
	GIZA++	21.0

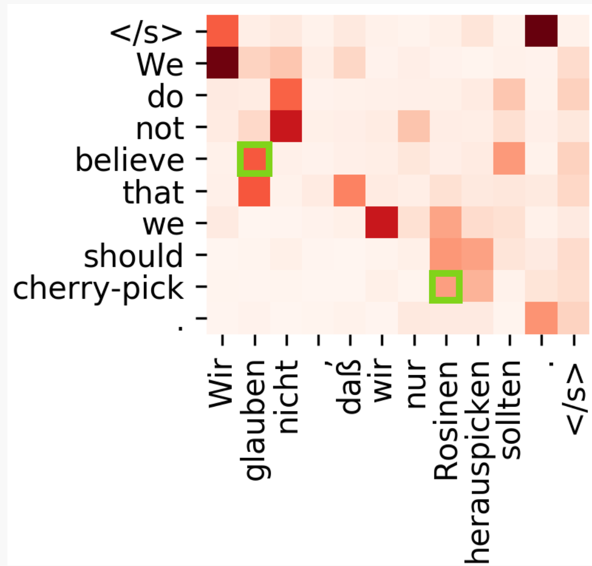
- Alignments from **norms** in the alignment with input setting are as good as those from **fast_align**

One Reason: Large weights for EOS

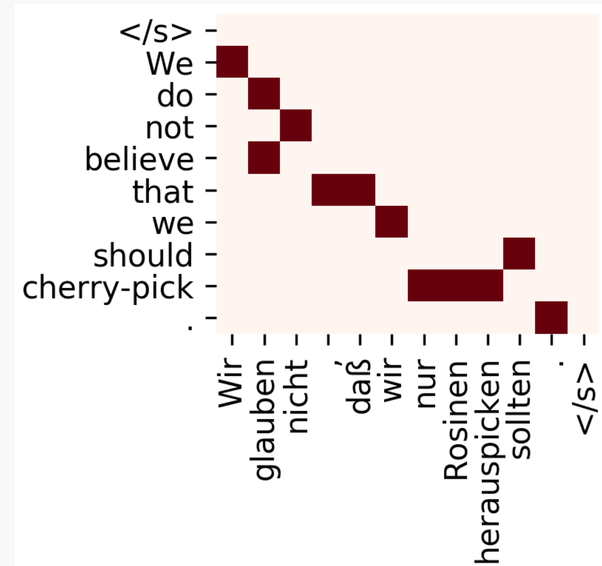
Attention weight α



Norm $\|\alpha f(\boldsymbol{x})\|$ (ours)



Reference



In layer 2
(alignment with input setting)

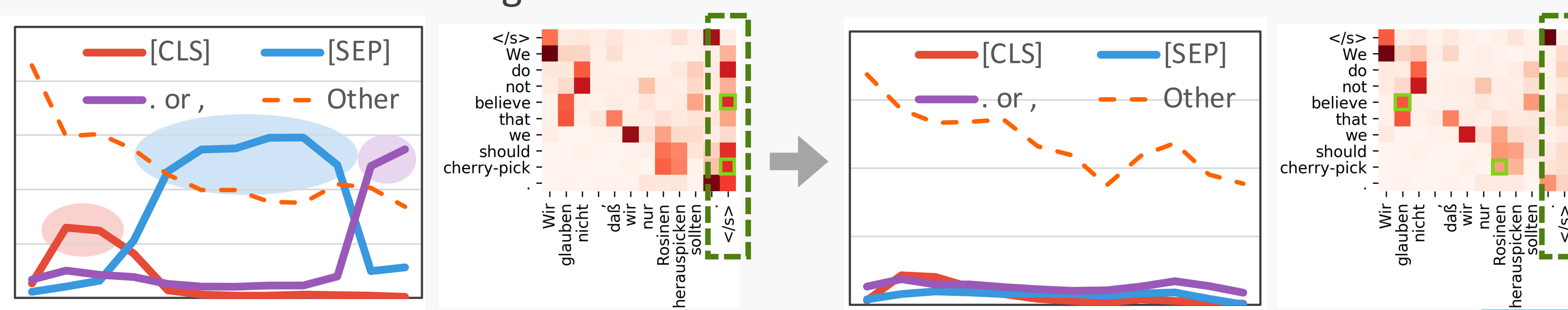
- In the weight-based extraction, EOS is often misaligned with some target words
  - Norm is small despite its large weights

Summary

- Proposed the norm-based analysis considering input vectors and vector transformations as well
- Self-attentions in BERT gather only a little from specific tokens despite assigning high attention weights to them
- Cleaner word alignments can be extracted from attentions in a Transformer NMT model

Attention weight α

proposed analysis $\|\alpha f(\boldsymbol{x})\|$



3 min Overview
for Zoom Q&A Session 11A

Attention is Not Only a Weight: Analyzing Transformers with Vector Norms

Goro Kobayashi[1], Tatsuki Kuribayashi[1,2], Sho Yokoi[1,3], Kentaro Inui[1,3]

[1] Tohoku University, [2] Langsmith Inc., [3]RIKEN

EMNLP 2020, Zoom Q&A Session 11A
November 18, 2020

Proposal: analyzing attentions through vector norms

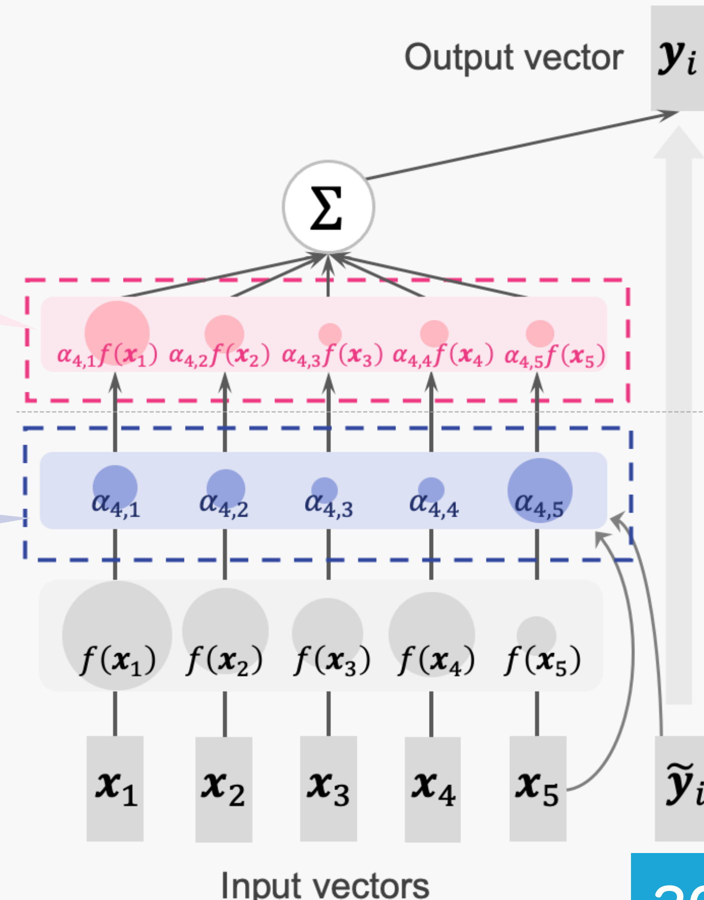
Propose to analyze Transformers (attentions) using **vector norms** instead of **attention weights**

Ours: Norms of weighted vectors

$\|\alpha f(\boldsymbol{x})\|$

Previous: Attention weights

α

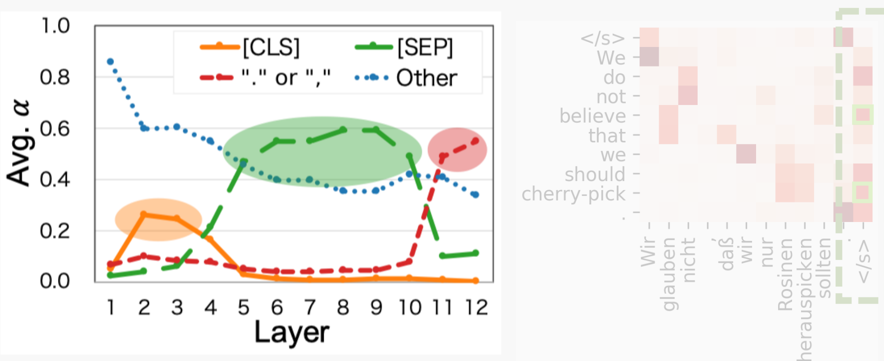


Able to addtionally consider input vector $\boldsymbol{x}$ and transformation f

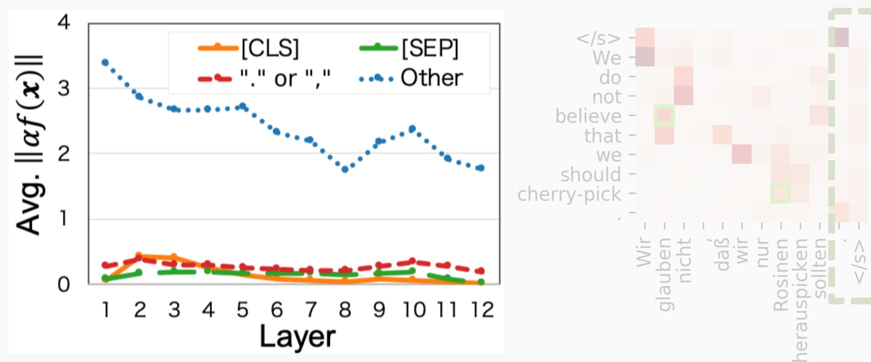
Summary of experiment results

- Self-attentions in BERT gather **only a little** from specific tokens despite assigning high attention weights to them
- Cleaner word alignments can be extracted from attentions in a Transformer NMT model by norms

Attention weight α



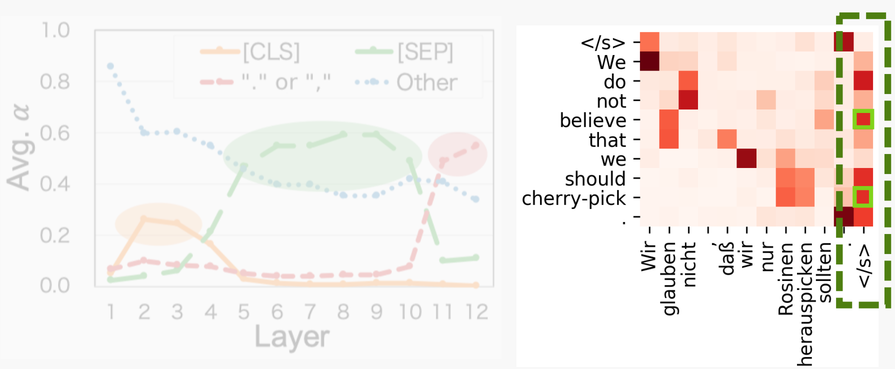
Vector norm $||\alpha f(\boldsymbol{x})||$



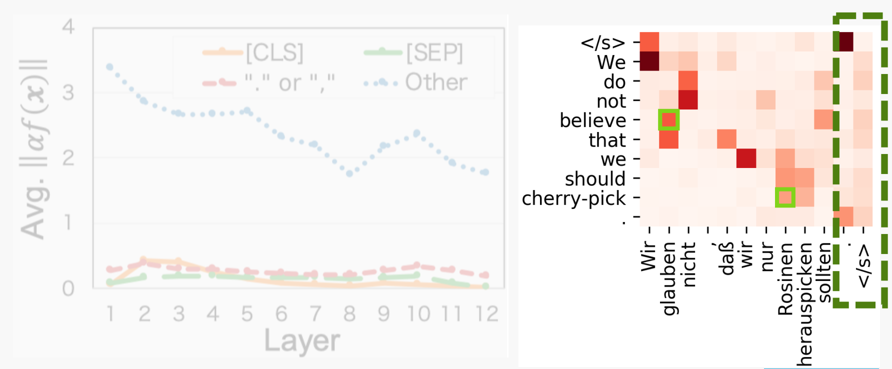
Summary of experiment results

- Self-attentions in BERT gather only a little from specific tokens despite assigning high attention weights to them
- **Cleaner word alignments** can be extracted from attentions in a Transformer NMT model by vector norms

Attention weight α



Vector norm $\|\alpha f(\boldsymbol{x})\|$



I’m not good at English…
Please speak slowly and simply🙏

Thank you!!