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

Goro Kobayashi\textsuperscript{1}, Tatsuki Kuribayashi\textsuperscript{1,2}, Sho Yokoi\textsuperscript{1,3}, Kentaro Inui\textsuperscript{1,3}

\textsuperscript{1}Tohoku University, \textsuperscript{2}Langsmith Inc., \textsuperscript{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)

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

[https://gluebenchmark.com/leaderboard](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**

\[ y_i = \text{Attention}(x_1, x_2, x_3, x_4, x_5) \]

[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.

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

\[
\text{Output vector } y_i = \sum \alpha_i f(x_i)
\]

Input vectors: \(x_1, x_2, x_3, x_4, x_5\)

Value vectors: \(f(x_1), f(x_2), f(x_3), f(x_4), f(x_5)\)

Weights: \(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\)
Attention performs a weighted sum of vectors

Attention mechanism consists of 3-step process.

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

Output vector \( y_i \)

Input vectors \( x_1, x_2, x_3, x_4, x_5 \)

Affine transformation (including transformation to Value vectors)

Weighting

Summation

Update

\[ \sum \alpha_i f(x_i) \]
Attention performs a weighted sum of vectors

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

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

**Output vector** $\mathbf{y}_i$

**Input vectors** $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$

**Update** $\tilde{\mathbf{y}}_i$
Attention performs a weighted sum of vectors.

Attention mechanism consists of 3-step process.

1. Affine transformation (including transformation to Value vectors)
2. Weighting
3. Summation
Attention performs a weighted sum of vectors

Attention mechanism consists of 3-step process.

Output:
Weighted sum of transformed vectors

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

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

Output vector: \( y_i \)

Input vectors: \( x_1, x_2, x_3, x_4, x_5 \)

Output vector: \( \tilde{y}_i \)
**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.]

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

😢 Ignore the effect of transformed vector \( f(x) \)

misunderstand that attention gathers a lot from \( x_5 \) to generate \( y_i \) even if \( \alpha f(x_1) \) is predominant in \( y_i \)
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 \( \text{norm} \| \alpha_{i,j} f(x_j) \| \)

😊 Consider the vector \( f(x) \) in addition to attention weight \( \alpha \)
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

\[ \alpha \]

\[ f(x) \]

\[ \gamma \]

Input vectors

Output vector

Correctly interpret that attention gathers the most from \( x_1 \) to generate \( y_i \) (a little from \( x_5 \))

Consider the vector \( y_i (\alpha) \) in addition to attention weight \( \alpha \)
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](https://github.com/clarkkev/attention-analysis)

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

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 $\|\alpha f(x)\|$ is small despite its large weight $\alpha$?

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

• Attention weight $\alpha$ and norm of transformed vector $\|f(x)\|$ 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** $\alpha$ have empirically been shown **noisy** [Li+'19; Zenkel+'19; Ding+'19]
- Hypothesis: much cleaner alignments can be extracted from **norm** $\|\alpha f(x)\|$

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

- Alignment extraction
  - Extract the source word with the highest weight $\alpha$ or norm $\|\alpha f(x)\|$ 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)

<table>
<thead>
<tr>
<th></th>
<th>Attention weight</th>
<th>Norm (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alignment error rate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alignment with output</td>
<td>Alignment with input</td>
</tr>
<tr>
<td>layer mean</td>
<td>68.4</td>
<td>62.9</td>
</tr>
<tr>
<td>best layer</td>
<td>47.7 (layer 4 or 5)</td>
<td>41.4 (layer 2)</td>
</tr>
<tr>
<td></td>
<td>29.8 (layer 5)</td>
<td>25.0 (layer 2)</td>
</tr>
</tbody>
</table>

- Possible to extract cleaner word alignments from **norms** than **weights**
## Results:
Alignment Error Rate (lower is better)

<table>
<thead>
<tr>
<th></th>
<th>Attention weight</th>
<th>Norm (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alignment error rate</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>layer mean</td>
<td>layer mean</td>
</tr>
<tr>
<td>fast_align</td>
<td>68.4</td>
<td>62.9</td>
</tr>
<tr>
<td>GIZA++</td>
<td>28.4</td>
<td>21.0</td>
</tr>
<tr>
<td></td>
<td>best layer</td>
<td>best layer</td>
</tr>
<tr>
<td>fast_align</td>
<td>47.7 (layer 4 or 5)</td>
<td>41.4 (layer 2)</td>
</tr>
<tr>
<td>GIZA++</td>
<td>29.8 (layer 5)</td>
<td>25.0 (layer 2)</td>
</tr>
</tbody>
</table>

- Alignments from **norms** in the alignment with input setting are as good as those from **fast_align**.
One Reason: Large weights for EOS

- 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
3 min Overview for Zoom Q&A Session 11A
Attention is Not Only a Weight: Analyzing Transformers with Vector Norms

Goro Kobayashi\textsuperscript{1}, Tatsuki Kuribayashi\textsuperscript{1,2}, Sho Yokoi\textsuperscript{1,3}, Kentaro Inui\textsuperscript{1,3}

\textsuperscript{1}Tohoku University, \textsuperscript{2}Langsmith Inc., \textsuperscript{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 $\| \alpha f(x) \|$

Previous: Attention weights $\alpha$

Able to additionally consider input vector $x$ 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.
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.
I’m not good at English...
Please speak slowly and simply🙏

Thank you!!