Enhancing Machine Translation with Dependency-Aware Self-Attention

ACL 2020

Emanuele Bugliarello, Naoaki Okazaki
Syntax-Aware MT
Syntax-Aware MT

• Syntax
  • Long-distance dependencies
  • Relations between words
  • Grammatically correct outputs
Syntax-Aware MT

- Syntax
  - Long-distance dependencies
  - Relations between words
  - Grammatically correct outputs

- Syntax-aware SMT and RNNs
Syntax-Aware MT

• Syntax
  • Long-distance dependencies
  • Relations between words
  • Grammatically correct outputs

• Syntax-aware SMT and RNNs

• How to incorporate source syntax in Transformers for NMT?
How to incorporate source syntax in Transformers for NMT?
How to incorporate source syntax in Transformers for NMT?

Recent studies
How to incorporate source syntax in Transformers for NMT?

Recent studies

• Wu et al. (2018): 3 encoders + 2 decoders + target dependencies

How to incorporate source syntax in Transformers for NMT?

Recent studies

• Wu et al. (2018): 3 encoders + 2 decoders + target dependencies

Ours (Pascal)

• Parameter-free

How to incorporate source syntax in Transformers for NMT?

Recent studies

• Wu et al. (2018): 3 encoders + 2 decoders + target dependencies
• Zhang et al. (2019): Closed-vocabulary

Ours (Pascal)

• Parameter-free

How to incorporate source syntax in Transformers for NMT?

Recent studies

- Wu et al. (2018): 3 encoders + 2 decoders + target dependencies
- Zhang et al. (2019): Closed-vocabulary

Ours (Pascal)

- Parameter-free
- Open-vocabulary

How to incorporate source syntax in Transformers for NMT?

Recent studies

• Wu et al. (2018): 3 encoders + 2 decoders + target dependencies
• Zhang et al. (2019): Closed-vocabulary
• Currey & Heafield (2019): Low- vs. high-resource scenarios

Ours (Pascal)

• Parameter-free
• Open-vocabulary

How to incorporate source syntax in Transformers for NMT?

Recent studies

• Wu et al. (2018): 3 encoders + 2 decoders + target dependencies
• Zhang et al. (2019): Closed-vocabulary
• Currey & Heafield (2019): Low- vs. high-resource scenarios

Ours (Pascal)

• Parameter-free
• Open-vocabulary
• For both low- and high-resource scenarios

Transformer (Vaswani et al., 2017)

Transformer with Parent-Scaled Self-Attention
Transformer with Parent-Scaled Self-Attention

Positional Encoding

Add & Normalize

Feed Forward

Add & Normalize

Multi-Head Self-Attention

Add & Normalize

Feed Forward

Add & Normalize

Multi-Head Parent-Scaled Self-Attention

Output Probabilities

Softmax

Linear

Add & Normalize

Feed Forward

Add & Normalize

Multi-Head Self-Attention

Add & Normalize

Multi-Head Self-Attention

Positional Encoding

Input Embedding

Inputs

Positional Dependencies

Output Embedding

Outputs
Self-Attention
Self-Attention

The monkey eats a banana
Self-Attention

The monkey eats a banana

\[ \text{The monkey eats a banana} \]
Self-Attention

The monkey eats a banana

X V Q K

The monkey eats a banana
Self-Attention

The monkey eats a banana

\[
s_{ij}: \text{score of token } i \text{ w.r.t. token } j
\]
Self-Attention

The monkey eats a banana

\[ s_{ij} : \text{score of token } i \text{ w.r.t. token } j \]
Self-Attention

The monkey eats a banana

The monkey eats a banana

$s_{ij}$: score of token $i$ w.r.t. token $j$
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

$X$: input

$Q$: query

$V$: value

$K$: key

$S$: score

$M$: output

$s_{ij}$: score of token $i$ w.r.t. token $j$
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

\[ s_{ij}: \text{score of token } i \text{ w.r.t. token } j \]
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

\( s_{ij} \): score of token \( i \) w.r.t. token \( j \)
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

$X$: Input sequence

$V$, $Q$, $K$: Query, key, value matrices

$S$: Attention scores

$D$: Distance matrix

$M$: Softmax layer

Dist: Distance function

$s_{ij}$: Score of token $i$ w.r.t. token $j$

$d_{ij}$: Proximity of token $j$ to the parent token of $i$
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

\[
V = \text{softmax}(\text{dist})
\]

\[
K = Q^T X
\]

\[
P = \text{score of token } i \text{ w.r.t. token } j
\]

\[
d_{ij} = \text{proximity of token } j \text{ to the parent token of } i
\]

\[
d_{ij} = f_X(j \mid p[i], \sigma^2)
\]
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

\[
s_{ij}: \text{score of token } i \text{ w.r.t. token } j
\]
\[
d_{ij}: \text{proximity of token } j \text{ to the parent token of } i
\]
\[
d_{ij} = f_X(j | p[i], \sigma^2)
\]
\[
n_{ij} = s_{ij} d_{ij}
\]
Robustness to noisy annotations
Robustness to noisy annotations

• No gold parses
Robustness to noisy annotations

- No gold parses
- Parent ignoring
  - Randomly disregard dependencies at training time
Experiments

Data

low-resource
- NC11 en-de, de-en
- WMT18 en-tr

high-resource
- WMT16 en-de
- WMT17 en-de
- WAT en-ja
Experiments

Data

low-resource
• NC11 en-de, de-en
• WMT18 en-tr

high-resource
• WMT16 en-de
• WMT17 en-de
• WAT en-ja

Models

• Transformer
Experiments

Data

low-resource

• NC11 en-de, de-en
• WMT18 en-tr

high-resource

• WMT16 en-de
• WMT17 en-de
• WAT en-ja

Models

• Transformer
• + Pascal
Experiments

Data

low-resource
• NC11 en-de, de-en
• WMT18 en-tr

high-resource
• WMT16 en-de
• WMT17 en-de
• WAT en-ja

Models

• Transformer
• + Pascal
• + LISA (Strubell et al., 2018)

Experiments

Data

*low-resource*
- NC11 en-de, de-en
- WMT18 en-tr

*high-resource*
- WMT16 en-de
- WMT17 en-de
- WAT en-ja

Models

- Transformer
- + Pascal
- + LISA (Strubell et al., 2018)
- + Multi-Task (Currey & Heafield, 2019)

Experiments

Data

low-resource

- NC11 en-de, de-en
- WMT18 en-tr

high-resource

- WMT16 en-de
- WMT17 en-de
- WAT en-ja

Models

- Transformer
- + Pascal
- + LISA (Strubell et al., 2018)
- + Multi-Task (Currey & Heafield, 2019)
- + S&H (Sennrich & Haddow, 2016)

Results
Results

Test performance
Results

Test performance

Analysis by sentence length
Conclusion
Conclusion

• Pascal
  • A parameter-free, syntax-aware self-attention
Conclusion

• Pascal
  • A parameter-free, syntax-aware self-attention
  • Parent ignoring regularisation for noisy annotations
Conclusion

• Pascal
  • A parameter-free, syntax-aware self-attention
  • Parent ignoring regularisation for noisy annotations

• Approaches for RNNs don’t always transfer to Transformers
Conclusion

- Pascal
  - A parameter-free, syntax-aware self-attention
  - Parent ignoring regularisation for noisy annotations
- Approaches for RNNs don’t always transfer to Transformers
- Core components of the Transformer can best embed syntax
Conclusion

• Pascal
  • A parameter-free, syntax-aware self-attention
  • Parent ignoring regularisation for noisy annotations

• Approaches for RNNs don’t always transfer to Transformers

• Core components of the Transformer can best embed syntax

• Code available online at https://github.com/e-bug/pascal