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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
Zhang et al. Syntax-enhanced neural machine translation with syntax-aware word representations. NAACL 2019.
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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
Zhang et al. Syntax-enhanced neural machine translation with syntax-aware word representations. NAACL 2019.
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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
Zhang et al. Syntax-enhanced neural machine translation with syntax-aware word representations. NAACL 2019.
Currey & Heafield. Incorporating source syntax into transformer-based neural machine translation. WMT 2019.
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

Wu et al. Dependency-to-dependency neural machine translation. TASLP 2018.
Zhang et al. Syntax-enhanced neural machine translation with syntax-aware word representations. NAACL 2019.
Currey & Heafield. Incorporating source syntax into transformer-based neural machine translation. WMT 2019.
Transformer (Vaswani et al., 2017)

Vaswani et al. Attention is all you need. NeurIPS 2017.
Transformer with Parent-Scaled Self-Attention
Transformer with Parent-Scaled Self-Attention
Self-Attention
Self-Attention

The monkey eats a banana
The monkey eats a banana
Self-Attention

The monkey eats a banana

The monkey eats a banana

\[ X \rightarrow V \rightarrow Q \rightarrow K \]
Self-Attention

The monkey eats a banana

$X$: The monkey eats a banana
$V$: Score of token $i$ w.r.t. token $j$
Self-Attention

The monkey eats a banana

$X$: input sequence

$V$: value vector

$Q$: query vector

$K$: key vector

$S$: attention scores

$softmax$: output

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

The monkey eats a banana

$X$: score of token $i$ w.r.t. token $j$
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

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

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

The monkey eats a banana

\[ X = QV \]

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

The monkey eats a banana

\[ \text{softmax} \]

\[ \text{dist} \]

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

\[ d_{ij}: \text{proximity of token } j \text{ to the parent token of } i \]
Pascal: Parent-Scaled Self-Attention

The monkey eats a banana

X

softmax

V

Q

K

M

The monkey eats a banana

s_{ij}: score of token \( i \) w.r.t. token \( j \)

d_{ij}: proximity of token \( j \) to the parent token of \( i \)

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

The monkey eats a banana

The monkey eats a banana

$P$: score of token $i$ w.r.t. token $j$

$D$: proximity of token $j$ to the parent token of $i$

$s_{ij} = f_{XY}(j | p[i], \sigma^2)$

$n_{ij} = s_{ij} d_{ij}$
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)

Strubell et al. Linguistically-Informed Self-Attention for Semantic Role Labeling. EMNLP 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)

Strubell et al. Linguistically-Informed Self-Attention for Semantic Role Labeling. EMNLP 2018.
Currey & Heafield. Incorporating source syntax into transformer-based neural machine translation. WMT 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)

Sennrich & Haddow. Linguistic Input Features Improve Neural Machine Translation. WMT 2016.
Strubell et al. Linguistically-Informed Self-Attention for Semantic Role Labeling. EMNLP 2018.
Currey & Heafield. Incorporating source syntax into transformer-based neural machine translation. WMT 2019.
Results
Results

Test performance

ACL 2020
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