Novel positional encodings to enable tree-based transformers

Vighnesh Leonardo Shiv, Chris Quirk

Transformers treat input as bag of tokens, annotated with *positional encodings*

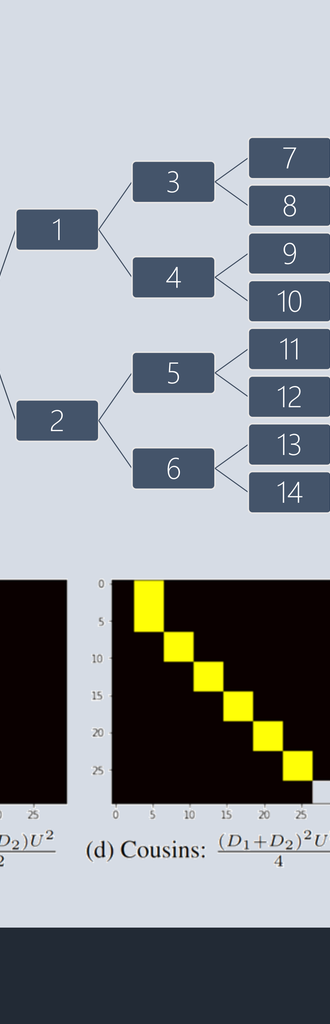
- Sinusoidal encodings allow indexing in sequence models
  - For any position x, we can represent $p_{x+k} = M_k p_x$ using transform M_k
  - Query/key transforms can represent both absolute and relative positions
- In trees, *paths* represent relative positions

Goal: tree positional encodings that capture paths as affine transforms

Tree positional encodings

- Tree positional encodings are represented as a stack.

$$
\begin{aligned}
r &= & \mathbf{0} &= [0,0,0,0,0,0,0,0,0,0] \\
x &= & D_0\mathbf{0} &= [1,0,0,0,0,0,0,0,0,0] \\
y &= & D_2D_0\mathbf{0} &= [0,0,1,1,0,0,0,0,0,0] \\
z &= D_1D_2D_0\mathbf{0} &= & [0,1,0,0,0,1,1,0,0]
\end{aligned}
$$



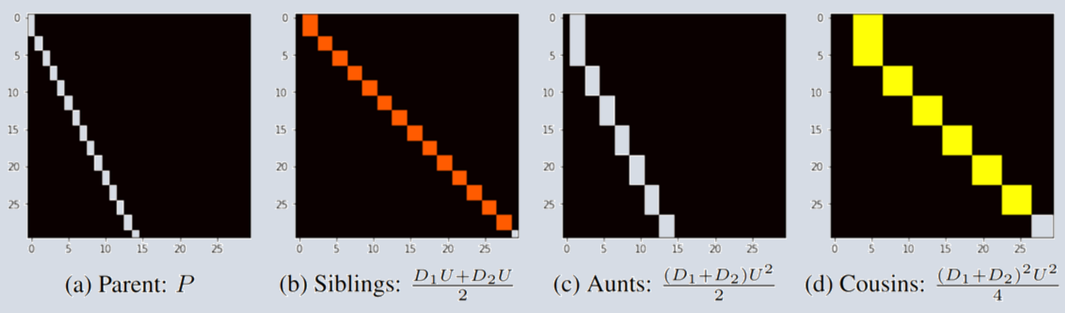
- Traveling down a branch corresponds to pushing onto the stack.
- Traveling up a branch corresponds to popping off the stack.
- Each path is represented by an affine transform.
- Improve inductive bias by concatenating multiple copies with different scaling factors.

Decoder

- Select between depth-first search, breadth-first search, etc.
- Decoder deterministically selects next position to decode to.
- Separates tree traversal from machine learning.

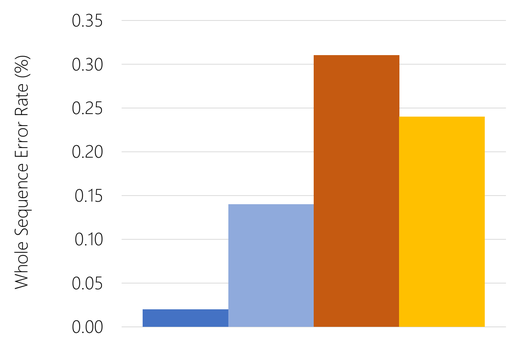
Example: attention heatmaps

- Figures demonstrate how these encodings successfully capture common tree relationships.
- Axes indexed by breadth-first traversal through a balanced binary tree.

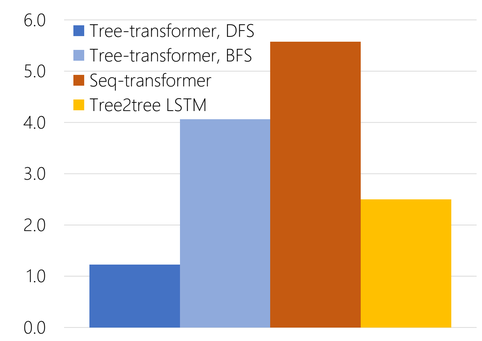


(a) Parent: P (b) Siblings: $\frac{D_1U+D_2U}{2}$ (c) Aunts: $\frac{(D_1+D_2)U^2}{2}$ (d) Cousins: $\frac{(D_1+D_2)^2U^2}{4}$

Tree-to-tree evaluation: program translation

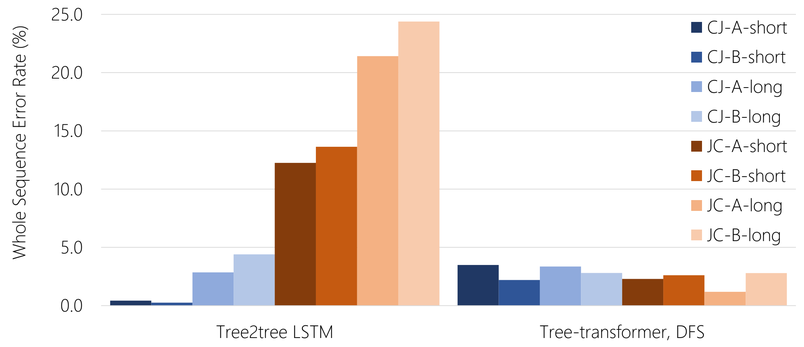


(a) Short synthetic programs



(a) Long synthetic programs

On *synthetic iterative-functional language translation*, the tree-transformer **outperformed all baselines**, including both the sequence-transformer and the tree-LSTM.



On *CoffeeScript-JavaScript translation*, tree-transformers achieved state-of-the-art results on the majority of datasets, including a **22% performance increase on the hardest task.**

Seq-to-tree evaluation: semantic parsing

When mapping natural language requests to tree-structured database queries, tree-transformers **outperform** sequence-based models on larger datasets. Tree-transformers **achieve 86.4%** whole-program accuracy on ATIS.

Dataset	Training instances	Seq2Tree Tform	Seq2Seq Tform	Literature
JOBS	500	84.3	85.0	**90.7** (Liang et al., 2011)
GEO	680	84.6	81.1	**89.0** (Kwiatkowski et al., 2013)
ATIS	4,480	**86.4**	84.4	84.6 (Dong & Lapata, 2016)

Published at NeurIPS 2019.