Transformers as Graph-to-Graph Models

Andrei Catalin Coman

Graphs in Transformers

- Transformers are essentially latent graph processing models, with sequences just being a special case.
- **Attention weights are functionally equivalent to graph edges,** since they determine how information propagates.
- Attention weights are computed from a set of vectors.
- Graphs are encoded in Transformers' set-of-vector embeddings.

"Transformers are latent graph models"

Graph2Graph Transformers (G2GT)

- Our G2GT makes this graph processing ability explicit.
- **Observed graphs** are input to the attention weight computations as relation embeddings.
- **Predicted graphs** are output with attention-like functions.
- **Latent graphs** are computed by pretrained Transformer weights.

\[
e_{ij} = \frac{1}{\sqrt{d}} \left[ x_i W^Q (x_j W^K)^T + x_i W^Q (r_{ij} W^R_1)^T + r_{ij} W^R_2 (x_j W^K)^T \right]
\]

\[
z_i = \sum_j \alpha_{ij} (x_j W^V + r_{ij} W^R_3)
\]
"Transformers are latent graph models"

**Efficient Global Graph Prediction**

- G2GT encodes the **observed graph**, **predicted graph**, and **latent graph** in a single joint Transformer embedding.
- G2GT enables **iterative refinement** of the predicted graph to capture global patterns over the graph and text, without any bespoke pipeline or decoding strategy.