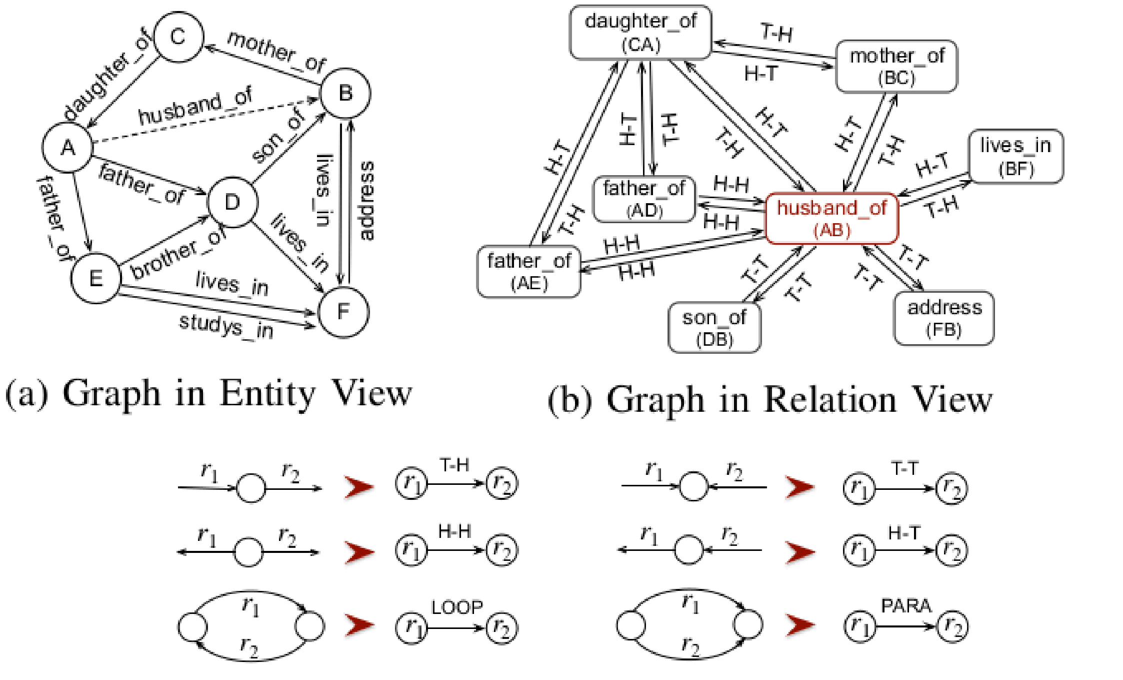
Entity-to-Relation Transformation:

The paper uses a graph transformation process that converts the entity-centric graph into a relation-centric view.

Each relation in the original graph becomes a node in the transformed graph.

Edges in this transformed graph represent relational adjacency based on shared entities in the original graph.

For example, if relation r1r_1r1 connects entity e1e_1e1 to e2e_2e2 and r2r_2r2 also connects e1e_1e1 to e3e_3e3, r1r_1r1 and r2r_2r2 are connected in the transformed graph.



(a) Graph in Entity View

(b) Graph in Relation View

(c) Connection Patterns of Relations

Fig. 3: (a) and (b): A running example of transforming an entity-view subgraph to a relation-view subgraph (a fragment for the relation nodes *husband_of* and *daughter_of*). (c): connection patterns (relationships) of relations according to their commonly associated entities.

Initial Embedding from Ontology (h0h_0h0):

- The ontology provides semantic information like domain, range, subproperties, and hierarchical relationships among relations.
- These semantic features are encoded into an initial embedding h0h_0h0 for each relation. This step involves mapping ontology-derived features to a trainable embedding space, often using a function fschemaf_{\text{schema}}fschema such as a linear transformation:

$$\boldsymbol{h}_{r_i}^0 = \boldsymbol{W}_1(\boldsymbol{W}_2\boldsymbol{h}_{r_i}^{onto})$$

Relational Message Passing:

- The relational graph is constructed, where each node represents a relation, and edges capture co-occurrence patterns or structural adjacencies in the KG.
- Starting from the initial embeddings h0h_0h0, message passing layers iteratively update the embeddings hkh_khk by aggregating information from neighboring relations:

also apply **a target relation-aware neighborhood attention mechanism** to highlight the neighbors that are highly related to the target relation. Formally, the AGGREGATE function at the k-th GNN layer is defined as follows:

$$\boldsymbol{h}_{\mathcal{N}_{r_i}}^k = \sigma_1\Big(\sum_{e=1}^{6} \sum_{r_j \in \mathcal{N}_{r_i}^e} \alpha_{r_t r_j}^k \boldsymbol{W}_e^k \boldsymbol{h}_{r_j}^{k-1}\Big) \tag{6}$$

$$\alpha_{r_t r_j}^k = \frac{exp(\sigma_2((\boldsymbol{h}_{r_t}^{k-1})^T \cdot \boldsymbol{h}_{r_j}^{k-1}))}{\sum_{r_{j'} \in \mathcal{N}_{r_i}^e} exp(\sigma_2((\boldsymbol{h}_{r_t}^{k-1})^T \cdot \boldsymbol{h}_{r_{j'}}^{k-1}))} \tag{7}$$

$$\boldsymbol{h}_{r_i}^k = \boldsymbol{h}_{\mathcal{N}_{r_i}}^{k-1} + \boldsymbol{h}_{r_i}^{k-1}$$

Final Relation Embeddings:

After

KKK

layers of message passing, the embeddings

hr(K)h_r^{(K)}hr(K)

for all relations are obtained. These embeddings integrate both:

- Semantic knowledge from the ontology (via h0h_0h0).
- Structural patterns from the KG (via iterative message passing and attention).

Prediction:

- For a given triple (h,r,t)(h, r, t)(h,r,t), the score is computed by combining the learned relation embedding hr(K)h_r^{(K)}hr(K) with the embeddings of entities hhh_hhh and hth_tht (if available):

$$score(u, r_t, v) = \boldsymbol{W}(\boldsymbol{h}_{r_t}^{K} + \boldsymbol{h}_{\mathcal{N}_{r_t}^{d}}^{d})$$