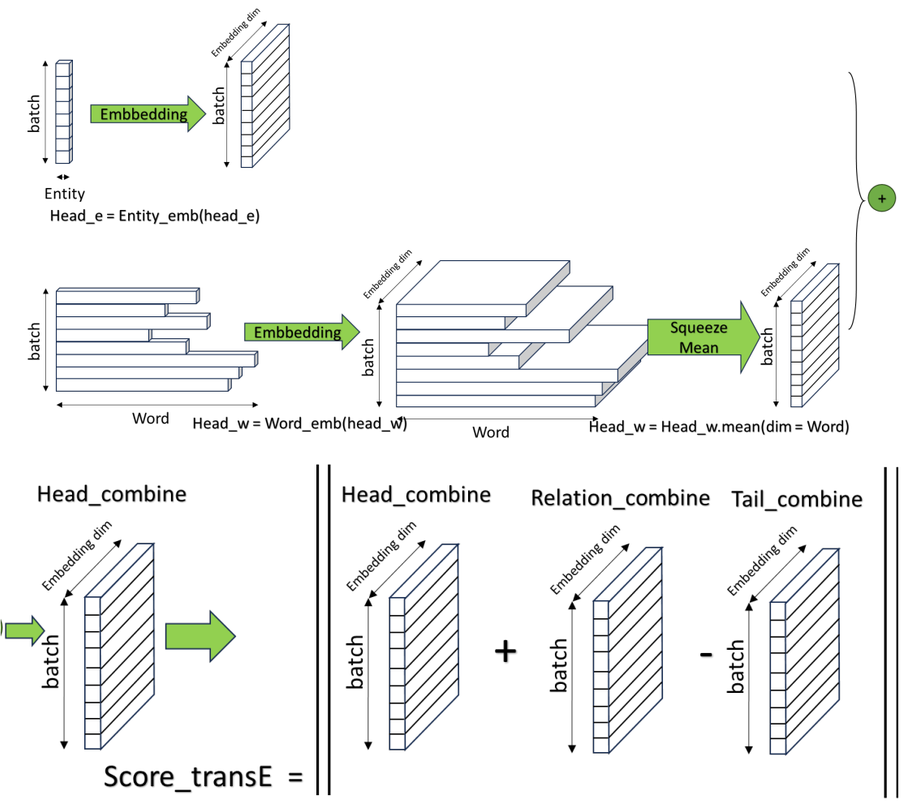
GRAPH NEURAL NETWORKS FOR TEXT EMBEDDINGS

Team 02: Jiwei Pan, Ziming Fang

Universität Stuttgart

Institute for Parallel and Distributed Systems, University of Stuttgart, Germany

1. Entity + Word Embedding



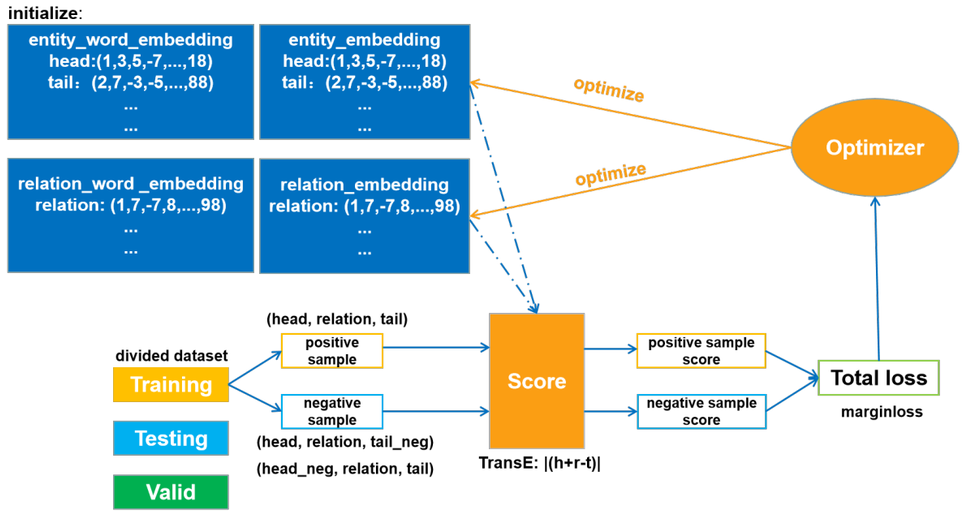
This embedding contains two parts of embeddings:

- **Entity embeddings**
  - **Entity embeddings from dimension (batch x 1) to (batch x embedding_dim)**
- **Word embeddings**
  - **Word embeddings from dimension (word_dim x 1) to (word_dim[i] x embedding_dim)**
  - **Combine words**

$$\begin{bmatrix} \text{word\_dim[1]} & \text{x embedding\_dim} \\ \text{word\_dim[2]} & \text{x embedding\_dim} \\ \text{word\_dim[3]} & \text{x embedding\_dim} \\ \dots & \\ \text{word\_dim[batch]} & \text{x embedding\_dim} \end{bmatrix}$$

  - **Squeeze dimension and take mean value on word_dim, get final embedding (batch x embedding_dim)**
- **Combine: Embedding_combine = entity_embedding + word_embedding**

2. Traning: TransE - Score



- **During the training process, we use the TransE model to perform L1 or L2 calculations on the embedding matrices of positive and negative samples and score them accordingly (the score is the negative distance).**
- **Marginloss:**

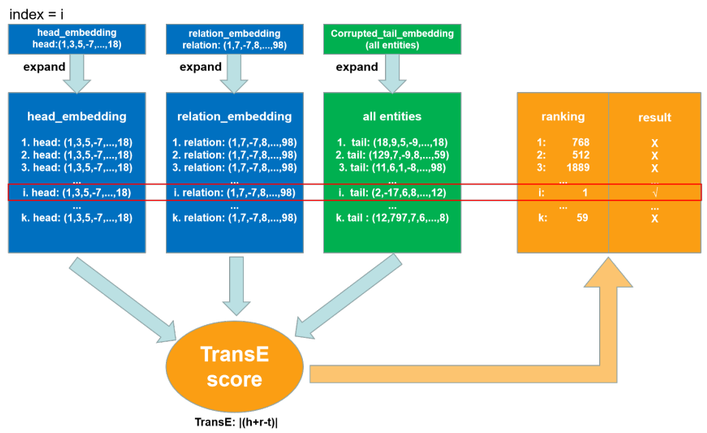
$$L_1 = \sum_{(h,l,t)\in S} \sum_{(h',l,t)\in S'} [\gamma + d(h+l,t) - d(h'+l,t)]_+$$

$$L_2 = \sum_{(h,l,t)\in S} \sum_{(h,l,t')\in S'} [\gamma + d(h+l,t) - d(h+l,t')]_+$$

$$Loss = L_1 + L_2$$

- **Afterwards, an optimizer is used to train the neural network by continuously updating the loss function until the loss converges or becomes stable.**

3. Testing: Ranking



The i-th row of the head and relationship matrices as an example:

- **We expand head and tail matrices to k x d**
- **The corrupted_tail_embedding is replaced with a k x d matrix representing all the entities.**
- **After scoring with TransE, we rank the scores and obtain the best match in the i-th row, which is ranked as 1st.**

4. Result and Conclusion

Model	TransE + Entity/Relation_emb	TransE + Entity/Relation_emb + Word_emb
learning rate	0,001	0,001
dimension	60	60
optimizer	Adam	Adam
gamma	1	1
activationfunction	ReLu	ReLu
margin	1	1
loss	marginrnakingloss	marginrankingloss
negative sample size	1	1
L2 or L1	L2	L2
best epoch	60	19
hit@10	0,6221	0,9354
MRR	0,3828	0,9024

- **Experimental results show that TransE is effective in predicting tail entities.**
- **Word embedding significantly improves the accuracy of the experiments!**

Deep Learning Lab 2023, July 17, Stuttgart, Germany