

# CONVOLUTIONAL COMPLEX KNOWLEDGE GRAPH EMBEDDINGS

## SUPPLEMENTAL MATERIAL.

**Anonymous authors**

Paper under double-blind review

### 1 EVALUATION SETTING

Following Bordes et al. (2013), we applied filtered Mean Reciprocal Rank (MRR) as our main evaluation metric for link prediction. During testing, we generated a set of candidate triples  $\mathcal{C}_i$  for each  $(s, p, o)_i \in \mathcal{G}^{\text{Test}}$ :  $\mathcal{C}_i = \{x | x \in \mathcal{E} \wedge (s, p, x)\}$ . As  $\mathcal{C}_i$  can contain several true triples, we removed all known triples from  $\mathcal{C}_i$ . Finally, we rank each triple in  $\mathcal{C}_i$  in descending order of scores. The rank of a missing entity in  $i$ -th test triple is then calculated as

$$\text{Rank}_i = 1 + \sum_{x \in \mathcal{C}_i \setminus \mathcal{G}} I[\phi((s, p, o)_i) < \phi((s, p, x))], \quad (1)$$

where  $I[P] = 1$  iff the  $P$  is true and 0 otherwise. Following the definition of the rank of a triple, the MRR and H@N are defined in Equation (2) and Equation (3):

$$\text{MRR} = \frac{1}{|\mathcal{G}^{\text{Test}}|} \sum_i \frac{1}{\text{Rank}_i}, \quad (2)$$

$$\text{H@N} = \frac{1}{|\mathcal{G}^{\text{Test}}|} \sum_i I[\text{Rank}_i < N]. \quad (3)$$

### 2 LINK PREDICTION RESULTS PER RELATIONS

Table 1: Link prediction results of the most frequent 10 relations in the test data for CONEX tested on UMLS. # denotes the number of occurrences of a relation.

Relations	#Train	#Test	MRR	H@10	H@3	H@1
affects	803	110	.987	1.00	.999	.981
result_of	455	71	.992	1.00	1.0	.985
interacts_with	363	49	.920	1.00	.979	.857
isa	399	47	.820	.957	.829	.765
causes	283	37	1.00	1.00	1.00	1.00
location_of	244	36	1.00	.944	1.00	.972
produces	221	28	1.00	1.00	1.00	1.00
process_of	369	27	.925	1.00	.962	.888
complicates	219	26	.980	1.00	1.00	.961
issue_in	223	24	1.00	1.00	1.00	1.00

### 3 HYPER-PARAMETERS

Throughout experiments, we set learning rate to 0.0005 and decay rate 1.0 and omit them from grid-search. ls, dout, c and r denote, label smoothing, dropout rate, the number of output channels

Table 2: Link prediction results of most frequent 10 relations in the test data for CONEX tested on FB15k-237. fre denotes the number of occurrences of a relation.

Relations	freq in Train-Test	MRR	H@1	H@3	H@10
../film_regional_release_date/film_release_region	12,893-1447	.508	.338	.618	.844
/people/person/profession	10,945-1311	.608	.467	.707	.874
../award_nomination/award	12,157-1067	.307	.168	.344	.627
../nominees/award_nomination/nominated_for	9,465-858	.089	.026	.090	.222
../film/film/performance/film	9,494-836	.027	.009	.025	.057
/film/film/genre	7,268-722	.496	.349	.580	.793
/music/genre/artists	5,880-664	.114	.069	.104	.199
../film/film_crew_gig/film_crew_role	5,305-606	.719	.582	.822	.967
/people/person/nationality	4,197-494	.717	.619	.783	.909
/people/person/gender	3,717-436	.939	.878	1.0	1.0

Table 3: Link prediction results on KINSHIP with different embedding sizes.

Model	Param. count	Emb. size	MRR	H@10	H@3	H@1
DistMult	2,660	20	.602	.920	.695	.453
DistMult	6,550	50	.673	.936	.762	.540
DistMult	13,300	100	.687	.949	.775	.555
DistMult	26,600	200	.669	.934	.751	.540
ComplEx	5,320	20	.742	.972	.837	.615
ComplEx	13,300	50	.828	.980	.908	.736
ComplEx	26,600	100	.850	.977	.922	.770
ComplEx	53,200	200	.845	.974	.925	.760
ConvE	14,650	20	.729	.959	.814	.608
ConvE	93,490	50	.761	.962	.855	.643
ConvE	380,990	100	.814	.977	.895	.717
ConvE	1,529,490	200	.794	.966	.885	.688
TuckER	11,160	20	.808	.976	.891	.711
TuckER	132,900	50	.873	.979	.940	.803
TuckER	1,015,800	100	.842	.985	.907	.758
TuckER	8,031,600	200	.868	.983	.939	.792
CONEX	5,560	20	.779	.972	.874	.666
CONEX	11,329	50	.836	.979	.912	.754
CONEX	32,504	100	.872	<b>.988</b>	<b>.945</b>	.793
CONEX	46,540	200	<b>.886</b>	.985	.944	<b>.814</b>

in 2D convolution, data augmentation technique by include reciprocal triples, respectively. Table 4 shows best performing hyper-parameter settings.

#### 4 IMPLEMENTATION, SOFTWARE VERSIONS AND HARDWARE

We implemented CONEX in Pytorch Paszke et al. (2017). Our experiments are conducted on the framework provided in Balažević et al. (2019). All experiments were carried out on Ubuntu 18.04 with 16 GB RAM with 4 Intel(R) Core(TM) i5-7300U CPU @ 2.60GHz processors.

Table 4: Best performing hyperparameter values

Dataset	$d$	$dout_{input}$	$dout_{hidden}$	$dout_{feature\ map}$	channel-out	r
CONEX-FB15K-237	20	.1	.1	.1	64	False
CONEX-WN18RR	100	.1	.2	.1	64	False
CONEX-KINSHIP	100	.3	.5	.3	2	False
CONEX-UMLS	100	.4	.4	.4	8	True
ComplEx-KINSHIP	200	.3	-	-	-	False
ConvE-KINSHIP	100	.1	.1	.1	32	False
DistMult-KINSHIP	200	.3	-	-	-	False
TuckER-KINSHIP	100	.3	.4 and .5	-	-	True
HypER-KINSHIP	50	.2	.3	.2	16	False
ComplEx-UMLS	100	.3	-	-	-	False
DistMult-UMLS	100	.3	-	-	-	False
TuckER-UMLS	200	.3	.4 and .5	-	-	True
HypER-UMLS	20	.2	.3	.2	16	False

## REFERENCES

- Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*, 2019.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.