LowFER: Low-rank Bilinear Pooling for Link Prediction — Appendix

A. Proofs

A.1. Proposition 1

Proof. First, we will prove the case for $k = d_e$, with the proof for the case $k = d_r$ following a similar argument. For both cases, we represent entity embedding vector as $\mathbf{e}_i \in \{0,1\}^{|\mathcal{E}|}$, such that only i-th element is 1, and similarly, relation embedding vector as $\mathbf{r}_j \in \{0,1\}^{|\mathcal{R}|}$, such that only j-th element is 1. We represent with $\mathbf{U} \in \mathbb{R}^{d_e \times k d_e}$ and $\mathbf{V} \in \mathbb{R}^{d_r \times k d_e}$ the model parameters, then, given any triple $(e_i, r_j, e_l) \in \mathcal{T}$ with indices (i, j, l), such that $1 \leq i, l \leq |\mathcal{E}|$ and $1 \leq j \leq |\mathcal{R}|$:

For $k = d_e$: We let $\mathbf{U}_{mn} = 1$ for $n = m + (o-1)d_e$, for all m in $\{1, ..., d_e\}$ and for all o in $\{1, ..., k\}$ and 0 otherwise. Further, let $\mathbf{V}_{pq} = 1$ for $p = j$ and $q = (l-1)d_e + i$ and 0 otherwise. Applying $\mathbf{g}(e_i, r_j)$ and taking dot product of the resultant vector with $\mathbf{e}_l$ (Eq. 5) perfectly represents the ground truth as 1. Also, for any triple in $\mathcal{T}'$, a score of 0 is assigned.

For $k = d_r$: We let $\mathbf{U}_{mn} = 1$ for $m = i$ and $n = (l-1)d_e + j$ and 0 otherwise. Further, let $\mathbf{V}_{pq} = 1$ for $q = p + (o-1)d_e$, for all p in $\{1, ..., d_r\}$ and for all o in $\{1, .., k\}$ and 0 otherwise. Rest of the argument follows the same as for $k = d_e$. □

Note that these bounds are not tight, in the sense that they are derived for checking the full expressibility of a model in handling *all-types* of relations with zero error, i.e., perfect reconstruction of the binary tensor $\mathbf{T}$ for a given $\mathcal{KG}$. More useful bounds can be derived with some ϵ-error allowed on the reconstruction.

A.2. Proposition 2

Proof. From Eq. 7 and 8, observe that the m^{th} slice of the core tensor $\mathcal{W}$ on object dimension is approximated by adding k rank-1 matrices, each of which is a cross product between m^{th} column in $\mathbf{W}_U^{(l)}$ and m^{th} column in $\mathbf{W}_V^{(l)}$, for all l in $\{1, ..., d_e\}$. Each slice of the core tensor $\mathcal{W}$ on object dimension has a maximum rank $\min(d_e, d_r)$ and from Singular Value Decomposition (SVD), there exists n ($\leq \min(d_e, d_r)$) scaled left singular and scaled right singular vectors whose sum of the cross products is equal to the slice. By choosing these scaled left singular vectors, scaled right singular vectors and zero vectors (in case the rank of the corresponding slice is less than the maximum rank of any such slice) as columns for matrices $\mathbf{W}_U^{(l)}$, $\mathbf{W}_V^{(l)}$, for all l in $\{1, ..., d_e\}$, the core tensor $\mathcal{W}$ is obtained from Eq. 7 with $k \leq \min(d_e, d_r)$. □

B. Experiments

In this section we will present the details of the datasets, evaluation metrics, model implementation, the choice of hyperparameters and report additional experiments.

B.1. Data

We conducted the experiments on four benchmark datasets: WN18 (Bordes et al., 2013) - a subset of Wordnet, WN18RR (Dettmers et al., 2018) - a subset of WN18 created through the removal of inverse relations from validation and test sets, FB15k (Bordes et al., 2013) - a subset of Freebase, and FB15k-237 (Toutanova et al., 2015) - a subset of FB15k created through the removal of inverse relations from validation and test sets. Table A.1 shows the statistics of all datasets.

Table A.1. Datasets used for link prediction experiments, where n_e=number of entities, n_r=number of relations and the entities-to-relations ratio n_e/n_r is approximated to the nearest integer.

Dataset	n_e	n_r	n_e/n_r	Training	Validation	Testing
WN18	40,943	18	2275	141,442	5,000	5,000
WN18RR	40,943	11	3722	86,835	3,034	3,134
FB15k	14,951	1,345	11	483,142	50,000	59,071
FB15k-237	14,541	237	61	272,115	17,535	20,466

B.2. Evaluation Metrics

We report the standard metrics of Mean Reciprocal Rank (MRR) and Hits@k for $k \in \{1, 3, 10\}$. For each test triple (e_s, r, e_o), we score all the triples (e_s, r, e) for all $e \in \mathcal{E}$. We then compute the inverse rank of true triple and average them over all examples. However, Bordes et al. (2013) identified an issue with this evaluation and introduced *filtered* MRR, where we only consider triples of the form $\{(e_s, r, e) \mid \forall e \in \mathcal{E}$ s.t. $(e_s, r, e) \notin$ train $\cup$ valid $\cup$ test$\}$ during evaluation. We therefore reported *filtered* MRR for all the experiments. The Hits@k metric computes the percentage of test triples whose ranking is less than or equal to k.

Table A.2. Best performing hyper-parameter values for LowFER, where lr=learning rate, dr=decay rate, d_e=entity embedding dimension, d_r=relation embedding dimension, k=LowFER factorization rank, dE=entity embedding dropout, dMFB=MFB dropout, dOut=output dropout and ls=label smoothing

Dataset	lr	dr	d_e	d_r	k	dE	dMFB	dOut	ls
WN18	0.005	0.995	200	30	10	0.2	0.1	0.2	0.1
WN18RR	0.01	1.0	200	30	30	0.2	0.2	0.3	0.1
FB15k	0.003	0.99	300	30	50	0.2	0.2	0.3	0.0
FB15k-237	0.0005	1.0	200	200	100	0.3	0.4	0.5	0.1

B.3. Implementation and Hyperparameters

We implemented LowFER[1] using the open-source code released by TuckER (Balažević et al., 2019a)[2]. We did random search over the embedding dimensions in $\{30, 50, 100, 200, 300\}$ for d_e and d_r. Further, we varied the factorization rank k in $\{1, 5, 10, 30, 50, 100, 150, 200\}$, with $k = 1$ (LowFER-1) and $k = 10$ (LowFER-10) as baselines. For WN18RR and WN18, we found best $d_e = 200$ and $d_r = 30$ with k value of 30 and 10 respectively. For FB15k-237, we found best $d_e = d_r = 200$ at $k = 100$. All of these embedding dimensions match the best reported in TuckER (Balažević et al., 2019a). However, for FB15k, we found using the configuration of $d_e = 300$ and $d_r = 30$ to be consistently better than $d_e = d_r = 200$. For fair comparison, we also reported the results for $d_e = d_r = 200$ and the best configuration when $d_e = 200$ and $(d_r, k) \leq 200$ (Table 5).

Similar to (Balažević et al., 2019a), we used Batch Normalization (Ioffe & Szegedy, 2015) but additionally power normalization and l_2-normalization to stabilize training from large outputs following the Hadamard product in main scoring function (Yu et al., 2017). We tested the best reported hyperparmeters of Balažević et al. (2019a) with random search and observed good performance in initial testing. With d_e, d_r and k selected, we used fixed set of values for rest of the hyperparameters reported in Balažević et al. (2019a), including learning rate, decay rate, entity embedding dropout, MFB dropout, output dropout and label smoothing (Szegedy et al., 2016; Pereyra et al., 2017) (see Table A.2 for the best hyperparameters). We used Adam (Kingma & Ba, 2015) for optimization. In all the experiments, we trained the models for 500 epochs with batch size 128 and reported the final results on test set.

B.4. Models Comparison

We compared LowFER with non-linear models including ConvE (Dettmers et al., 2018), R-GCN (Schlichtkrull et al., 2018), Neural LP (Yang et al., 2017), RotatE (Sun et al., 2019) (where we reported their results without the self-adversarial negative sampling, see Appendix H in their paper for details), TransE (Bordes et al., 2013), HolE (Nickel et al., 2016), TorusE (Ebisu & Ichise, 2018) and HypER (Balažević et al., 2019b). In linear models, we compared against DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), ANALOGY (Liu et al., 2017), SimplE (Kazemi & Poole, 2018) and state-of-the-art TuckER (Balažević et al., 2019a) model.

Additional models that were not reported in the main results (Table 2) but were outperformed by LowFER include M-Walk (Shen et al., 2018) (MRR=0.437, Hits@1=0.414, Hits@3=0.445 on WN18RR) and MINERVA (Das et al., 2017) (Hits@10=0.456 on FB15k-237). The results for all the models other than LowFER are reported from Balažević et al. (2019b) and Balažević et al. (2019a). Lastly, in section 5.4 for per relations comparison, we trained the TuckER models with the best configurations reported in Balažević et al. (2019a) for WN18 and WN18RR.

[1] https://github.com/suamin/LowFER

[2] https://github.com/ibalazevic/TuckER