CascadER: Cross-Modal Cascading for Knowledge Graph Link Prediction

Tara Safavi[1]*, **Doug Downey**[2,3], **Tom Hope**[2,4]
tsafavi@umich.edu, dougd@allenai.org, tomh@allenai.org
[1]**University of Michigan,** [2]**Allen Institute for Artificial Intelligence (AI2),**
[3]**Northwestern University,** [4]**The Hebrew University of Jerusalem**

Abstract

Knowledge graph (KG) link prediction is a fundamental task in artificial intelligence, with applications in natural language processing, information retrieval, and biomedicine. Recently, promising results have been achieved by leveraging cross-modal information in KGs, using ensembles that combine knowledge graph embeddings (KGEs) and contextual language models (LMs). However, existing ensembles are either **(1)** not consistently effective in terms of ranking accuracy gains or **(2)** impractically inefficient on larger datasets due to the combinatorial explosion problem of pairwise ranking with deep language models. In this paper, we propose a novel tiered ranking architecture CascadER to maintain the ranking *accuracy* of full ensembling while improving *efficiency* considerably. CascadER uses LMs to rerank the outputs of more efficient base KGEs, relying on an adaptive subset selection scheme aimed at invoking the LMs minimally while maximizing accuracy gain over the KGE. Extensive experiments demonstrate that CascadER improves MRR by up to 9 points over KGE baselines, setting new state-of-the-art performance on four benchmarks while improving efficiency by one or more orders of magnitude over competitive cross-modal baselines. Our empirical analyses reveal that diversity of models across modalities and preservation of individual models' confidence signals help explain the effectiveness of CascadER, and suggest promising directions for cross-modal cascaded architectures.

1. Introduction

Knowledge graphs (KGs) are critical ingredients for applications across natural language processing, information retrieval, and biomedicine [Weikum et al., 2021]. Motivated by the observation that most KGs have high precision but low coverage, the goal of **knowledge graph link prediction**, also known as KG completion, is to automatically augment KGs with new factual information by predicting missing links between entities [Nickel et al., 2015].

Link prediction is typically framed as a ranking problem in a multi-relational graph [Bordes et al., 2013]: Given a query consisting of a *head* entity (e.g., *aspirin*) and *relation* type (e.g., *treats*), the task is to rank candidate *tail* entities by the likelihood that they answer the query and form a factual link the graph. Currently, the prevailing approach is to learn vector representations of entities and relations, or **knowledge graph embeddings** (KGEs), and use vector composition functions to score candidate links [Ruffinelli et al., 2020]. While often effective at modeling structural KG patterns [Sun et al., 2019], KGEs typically do not leverage textual information like entity descriptions in KGs, even though such texts help ameliorate KG sparsity and improve ranking accuracy [Xie et al., 2016, Chandak et al., 2022].

*. Work performed during an internship at the Allen Institute for Artificial Intelligence.

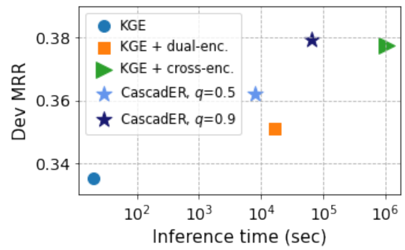


Figure 1: CASCADER maintains accuracy (y-axis) while improving efficiency (x-axis) by 1+ orders of magnitude over our most competitive ensemble baseline on CODEX-M. We use a three-tier CASCADER with dynamic pruning at quantiles $q = 0.5$ and $q = 0.9$ (§ 3.3).

To address this gap, recent studies have proposed to *ensemble* KGEs with advanced **language models** (LMs) like BERT [Devlin et al., 2019] in order to integrate structure and text for link prediction [Nadkarni et al., 2021, Wang et al., 2021]. These studies suggest the promise of cross-modal ensembles, but results are inconclusive due to two challenges. First, the cross-modal ensembles achieving the largest gains rely on impractically expensive models that require jointly encoding pairs of texts with deep LMs [Nadkarni et al., 2021], greatly increasing the computational cost of inference—e.g., from a few minutes with a KGE to *one month* with an LM on the same dataset [Kocijan and Lukasiewicz, 2021]). Second, the cross-modal ensembles that rely on more efficient "Siamese" dual-encoder LM architectures do not necessarily improve performance over KGEs alone [Wang et al., 2021].

In this paper, we introduce a novel cross-modal ensemble that simultaneously maximizes ranking *accuracy*, i.e., mean reciprocal rank (MRR) and hits@k, while maintaining *efficiency*. Our approach, **CASCADER**, is a tiered ranking architecture that uses a sequence of LMs of increasing complexity to adaptively reweight and rerank the outputs of more efficient base KGEs. CASCADER relies on a novel subset selection scheme: At each tier t of the architecture, we predict the minimal set of candidates that should progress to be reweighted at tier $t + 1$. By passing progressively smaller sets of outputs from one tier to the next, CASCADER balances accuracy gain and efficiency, as it minimizes invocation of the more computationally complex LMs further down the cascade.

Evaluated on five link prediction datasets, CASCADER achieves state-of-the-art performance, improving MRR over the strongest KGE baseline by up to 9 points, while also improving efficiency by orders of magnitude over our most accurate but computationally intensive ensemble baseline (Figure 1). Our qualitative analyses illuminate how cross-modal ensembling uniquely exploits complementary signals among graph and text models: We observe that promoting diversity and preserving confidence signals among the models in the ensemble help explain CASCADER's excellent performance, suggesting promising directions for research in cross-modal cascaded architectures.

2. Preliminaries

In this section, we provide preliminaries on single-modality and cross-modal approaches for link prediction. We discuss additional related work in Appendix A.1.

2.1 Problem definition

We consider the task of **ranking-based link prediction** in a knowledge graph $\mathcal{G}$ consisting of entities $\mathcal{E}$, relations $\mathcal{R}$, and factual (*head*, *relation*, *tail*) triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. The link prediction task consists of two directionalities: Score all tail entities $\hat{t} \in \mathcal{E}$ to "answer queries"—that is, complete known KG links—$(h, r, ?)$, and score all head entities $\hat{h} \in \mathcal{E}$ to answer queries $(?, r, t)$. The evaluation metrics are mean reciprocal rank (MRR), or the average reciprocal of each gold answer entity's rank over all queries, and hits@k, or the proportion of queries for which the gold answer entity is ranked in the top-k candidates.

A link prediction model is a scoring function $f : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$ that outputs real values indicating the plausibility of factual links in $\mathcal{G}$. At inference time, assume that we have N_{test} link prediction queries. For each query we have $|\mathcal{E}|$ potential answers, which are the entities in the KG. We define a query-answer **score matrix** $\mathbf{S} \in \mathbb{R}^{N_{\text{test}} \times |\mathcal{E}|}$, in which S_{ij} denotes the predicted probability that entity j answers link prediction query i to form a factual link in $\mathcal{G}$. Then, for each query i, all candidate entities j are ranked by their score descending, and the model's ranking accuracy is evaluated using these rankings.

2.2 Single-modality link prediction

Structure-based The most competitive structure-only approaches to link prediction are shallow knowledge graph embeddings (**KGE**s), which are decoder models that train entity and relation embeddings by optimizing with ranking losses. The main architectural distinction among different KGEs is how embeddings are combined to produce ranking scores for links, as both additive [Bordes et al., 2013, Sun et al., 2019] and multiplicative [Yang et al., 2015, Trouillon et al., 2016, Balazevic et al., 2019] scoring functions have been proposed. For details, we refer the reader to relevant surveys [Wang et al., 2017, Ji et al., 2020].

Text-based Recent text-based approaches to link prediction rely on advanced encoder language models (LMs) like BERT [Devlin et al., 2019] based on the Transformer architecture [Vaswani et al., 2017]. Let $X_h = [w_1, \ldots, w_h]$ denote the description of head entity h and $X_t = [w_1, \ldots, w_t]$ the description of tail entity t; for example, given the entity *aspirin*, a corresponding description could be "aspirin is known as a salicylate and a nonsteroidal anti-inflammatory drug." Siamese or **dual-encoder** link prediction approaches assume two LMs, potentially with shared weights [Wang et al., 2021]. As input, one LM takes the head entity description [`[CLS]`, X_h, `[SEP]`] and the other takes the tail entity description [`[CLS]`, X_t, `[SEP]`], where `[CLS]` and `[SEP]` refer to the LM's special classification and delimiter tokens. Both encoders output embeddings of their inputs, which are optimized such that linked entity pairs in the KG are scored highly compared to negative samples.[1]

In contrast to dual-encoders, **cross-encoder** LM approaches pack entity description pairs into a single sequence [`[CLS]`, X_h, `[SEP]`, X_t, `[SEP]`], and perform full cross-attention over all tokens in the sequence [Yao et al., 2019, Nadkarni et al., 2021]. At the output of the encoder, these approaches stack a scoring layer and train with ranking loss [Kim et al., 2020]. Cross-encoder LMs are typically more powerful for pairwise text ranking than dual-encoders [Luan et al., 2021]. However, they are less efficient. Whereas dual-encoders

1. Previous studies have found that the relation text may be omitted, and that including a relation disambiguation loss term in training is sufficient [Kim et al., 2020, Nadkarni et al., 2021].

can precompute all text embeddings and score text pairs at test time with fast vector dot products [Karpukhin et al., 2020], cross-encoders must jointly encode and score each text pair, which is impractically slow for large-scale text ranking [Reimers and Gurevych, 2019].

2.3 Cross-modal link prediction

Structure and text have recently been integrated for link prediction by ensembling KGEs and LMs with additive reweighting [Wang et al., 2021, Nadkarni et al., 2021]. Given a query i and candidate answer j, additive reweighting outputs a new link prediction score as the convex combination of the base models' scores:

$$S_{ij}^{\text{ens}} = \alpha \cdot S_{ij}^{\text{KGE}} + (1-\alpha) \cdot S_{ij}^{\text{LM}}, \tag{1}$$

where the weight $\alpha \in [0, 1]$ is a hyperparameter tuned on a held-out set.

As shown in Figure 1, additive reweighting can significantly improve link prediction ranking accuracy, up to 4 points MRR. However, Figure 1 also shows that ensembling increases inference complexity. Whereas the base KGE requires under one minute to score all query-answer pairs on a single NVIDIA Quadro RTX 8000 GPU, the KGE + dual-encoder ensemble takes around 3 hours, and the KGE + cross-encoder ensemble takes over *11 days* on the same hardware. This added expense is due to the fact that deep Transformer LMs typically consist of 12+ layers, and the complexity of Transformer encoding scales quadratically with the input length. Moreover, as discussed previously, cross-encoders must jointly encode all query/answer pairs, which further increases their computational cost.

3. Methodology

Assuming we are willing to pay some computational cost to improve link prediction performance, how can we achieve the *accuracy* of the cross-encoder ensemble while maintaining the *efficiency* of the dual-encoder ensemble, as shown in Figure 1? Our answer is **CascadER**, a cross-modal cascade architecture that achieves a delicate balance between these goals.

3.1 CascadER overview

As illustrated in Figure 2, CascadER is a progressive reranking architecture. We first obtain a base set of link prediction scores with an efficient KGE, then use increasingly complex LMs to rerank the base scores on progressively smaller sets of outputs—specifically, only the highly-ranked, most promising candidates, while leaving the rankings of the less promising candidates static. CascadER thus benefits from the performance gains of cross-modal ensembles without the full computational overhead (c.f. Figure 1).

As input, we are given $n \geq 2$ trained link prediction models $\{f^{(i)}, i = 1 \ldots n\}$ consisting of one KGE and one or more LMs. We sort the models by computational complexity, leading to an ordered sequence $(f^{(1)}, \ldots, f^{(n)})$ in which $f^{(1)}$ is the KGE and the subsequent models are LMs in ascending order of complexity (i.e., dual-encoders before cross-encoders). We first use the KGE to score all query/answer pairs in the inference set, leading to a score matrix $\mathbf{S}^{(1)} \in \mathbb{R}^{N_{\text{test}} \times |\mathcal{E}|}$ in which $S_{ij}^{(1)}$ denotes the KGE's plausibility score between query i and entity j. Then, at each tier $t = 1, \ldots, n-1$, we apply a pruning function that, for each

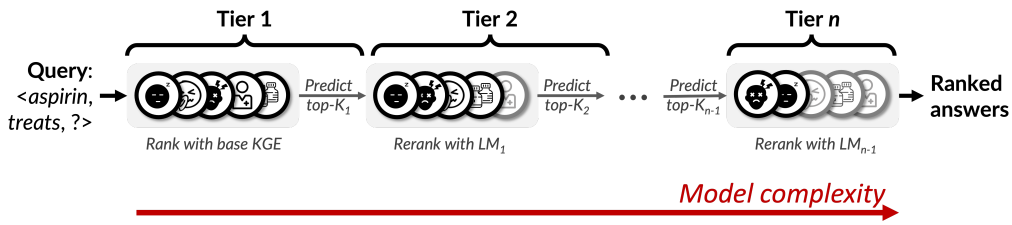


Figure 2: CascadER sequential reranking architecture.

query i, selects a subset of candidate answer entities $\mathcal{E}_i^{(t)} \subseteq \mathcal{E}$ to be reranked by the next-tier LM $f^{(t+1)}$; we postpone the discussion of pruning strategies to § 3.2 and 3.3.

For link prediction query i (that is, a query entity and relation type) and candidate answer entity j, we define the additive reranking function between tiers t and $t+1$ as follows:

$$\mathbb{I}_{j \in \mathcal{E}_i^{(t)}} \left[\alpha^{(t)} \cdot S_{ij}^{(t)} + \left(1 - \alpha^{(t)}\right) \cdot S_{ij}^{(t+1)} \right] + \mathbb{I}_{j \notin \mathcal{E}_i^{(t)}} \left[S_{ij}^{(t)} \right], \tag{2}$$

in which $\mathbb{I}$ denotes the set indicator function, $S_{ij}^{(t)}$ denotes the query-answer score output at tier t, and $\alpha^{(t)} \in [0, 1]$ is a hyperparameter that controls the additive influence of model $f^{(t)}$ in reranking the candidates in $\mathcal{E}_i^{(t)}$. Intuitively, (2) states that if the candidate answer entity j is within the subset $\mathcal{E}_i^{(t)}$ of candidates progressed to tier $t+1$, then we reweight its score with that of model $f^{(t+1)}$ at tier $t+1$, else we do not reweight its score.

3.2 Static candidate pruning

Candidate pruning with CascadER should balance two aims: **(1)** *Coverage*: For each link prediction query $(h, r, ?)$ or $(?, r, t)$, progress a sufficiently large set of candidate entities to the following tier, in order to increase coverage of the correct candidates; and **(2)** *Efficiency*: Progress as few candidates as possible to avoid unnecessarily invoking the next-tier reranking model on candidates that would not benefit from it.

A straightforward pruning approach used in information retrieval cascades is to progress only the top-k candidates from tier to tier using a predefined, global value of k [Wang et al., 2011, Matsubara et al., 2020]. This value helps control the accuracy-efficiency tradeoff, as a smaller k decreases coverage of promising candidates but improves efficiency, whereas a larger k increases coverage of promising candidates but incurs more computational cost. Formally, given query i and a selected value of k, we define **static pruning** as selecting the subset of candidates $\mathcal{E}_i^{(t)}$ such that $\mathcal{E}_i^{(t)} = \arg\max_{\mathcal{E}_i^{(t)} \subset \mathcal{E} \text{ and } |\mathcal{E}_i^{(t)}| = k} \sum_{j=1}^{|\mathcal{E}|} S_{ij}^{(t)}$.

The key challenge with static pruning is selecting the "right" value of k. One solution is to set k ad-hoc, e.g., $k = 100$ [Matsubara et al., 2020]. However, this approach may result in suboptimal performance from the accuracy or efficiency perspectives. To address this challenge, we propose a more principled strategy that searches for the best value of k per dataset and per tier t of the cascade. Given tier t and held-out query i, we obtain the cascade's rank $R_i^{(t)}$ of the gold answer entity. We construct a distribution of ranks $R_i^{(t)}$ over all hold-out queries, and use quantiles of this distribution to choose the grid of $k^{(t)}$ over which to search. For example, quantiles of 0.5, 0.75, and 0.9 means that we search over $k^{(t)}$

	Structure					Avg. desc. length
	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	# train	# dev	# test	
CoDEx-S	2,034	42	32,888	1827	1828	259.24
RepoDB	2,748	1	5,342	667	668	55.46
FB15K-237	14,541	237	272,115	17,535	20,466	138.95
CoDEx-M	17,050	51	185,584	10,310	10,311	159.48
WN18RR	40,943	11	86,835	3,034	3,134	13.91

Table 1: Statistics of the existing KG link prediction datasets considered in our experiments. Avg. desc. length refers to the average description of an entity in the KG.

equal to the median, 75th percentile, and 90th percentile of ranks $R_i^{(t)}$, which ensures that our selected $k^{(t)}$ is tailored to each dataset and each tier of reranking.

3.3 Dynamic candidate pruning

We propose to extend static pruning to an adaptive **dynamic pruning** approach. At any tier t of CascadER, we improve the quality of the current ranking by *predicting* how many top-ranked candidates *for each query* we should progress to the next tier. That is, given a query i and tier t, we predict an integer $\hat{k}_i^{(t)}$ that represents the number of candidates to progress to tier $t+1$. To achieve this, we train a lightweight model to predict the rank of the gold answer entity using quantile regression [Koenker and Hallock, 2001]. Formally, given a quantile q and the rank of the gold answer entity $R_i^{(t)}$ at tier t, we train a regressor to predict $\hat{k}_i^{(t)}$ by minimizing $\mathcal{L}_q(\hat{k}_i^{(t)}, R_i^{(t)}) = \max\left[q(R_i^{(t)} - \hat{k}_i^{(t)}), (q-1)(R_i^{(t)} - \hat{k}_i^{(t)})\right]$. As input features, we represent the i-th query by its sorted $|\mathcal{E}|$-dimensional score distribution $S_{i1}^{(t)}, \ldots S_{i|\mathcal{E}|}^{(t)}$ from tier t of the cascade, hypothesizing that these score distributions encode uncertainty information correlated to the relative "difficulty" of queries. In practice, we implement our regressor as a single-layer MLP trained on a random half of the dev set, and validated on the remaining dev examples.[2] We will subsequently show that this approach boosts CascadER's ability to balance accuracy and efficiency compared to static pruning.

4. Evaluation

As introduced in § 2.1, we evaluate CascadER for the link prediction task. Our evaluation metrics are MRR and hits@k for $k \in \{1, 3, 10\}$. Following the literature standard [Ruffinelli et al., 2020], we report metrics in the filtered setting to avoid false negatives.

4.1 Datasets

We consider the following link prediction benchmarks, as shown in Table 1: **FB15K-237** [Toutanova and Chen, 2015], **WN18RR** [Dettmers et al., 2018], **CoDEx-S** and **CoDEx-M** [Safavi and Koutra, 2020], and **RepoDB** [Brown and Patel, 2017, Nadkarni et al., 2021]. In terms of content, FB15K-237, CoDEx-S, and CoDEx-M comprise encyclopedic knowledge drawn from Freebase and Wikidata, WN18RR is a subset of the WordNet semantic network, and RepoDB is a subset of the RepoDB drug repurposing biomedical database [Brown and Patel, 2017]. For all datasets, we use the standard splits

2. Note that if the dev set is unavailable, we can alternatively hold out a small subset of train examples.

provided by the authors. We use the entity descriptions provided by Wang et al. [2021] for FB15K-237 and WN18RR, Nadkarni et al. [2021] for RepoDB, and Safavi and Koutra [2020] for CoDEx-S and CoDEx-M.

4.2 Baselines

KGE baselines We consider the competitive **RESCAL** [Nickel et al., 2011], **TransE** [Bordes et al., 2013], **ComplEx** [Trouillon et al., 2016], and **RotatE** [Sun et al., 2019] KGEs.

LM baselines We consider the **StAR dual-encoder** architecture [Wang et al., 2021] and the **KG-BERT cross-encoder** architecture [Yao et al., 2019], both trained in a multi-task setting with triple classification, margin ranking, and relation classification losses following the literature [Kim et al., 2020]. Due to the inference cost of KG-BERT on the larger datasets FB15K-237 and WN18RR (e.g., around one month for FB15K-237 [Kocijan and Lukasiewicz, 2021]), we report performance for these datasets from [Kim et al., 2020].

Ensemble baselines We consider the following additive ensembling baselines as defined in § 2.3: **KGE + KGE** ensembles the two strongest KGE baselines in terms of dev MRR; **KGE + StAR** [Wang et al., 2021] ensembles the best KGE with StAR; and **KGE + KG-BERT** [Nadkarni et al., 2021] ensembles the best KGE with KG-BERT.

SOTA We report the best published performance of which we are aware as of April 2022: NBFNet [Zhu et al., 2021] on FB15K-237, self-adaptive KGE + LM ensembling [Wang et al., 2021] on WN18RR, and ComplEx for the CoDEx datasets. Note that RepoDB has not been considered under the full entity ranking setting before, as Nadkarni et al. [2021] used a partial sample of negatives for each test query, so we do not report SOTA for RepoDB.

4.3 CascadER

Our first-tier KGE in CascadER is the best-performing baseline KGE in terms of dev MRR. We search for the optimal cascade in terms of dev MRR among the following hyperparameters: The choice of LMs (StAR dual-encoder, KG-BERT cross-encoder, or both); candidate pruning strategy (static versus dynamic); quantile $q \in \{0.5, 0.75, 0.9, 0.95\}$; and weighting hyperparameter $\alpha^{(t)} \in [0.05, 0.95]$ at each tier. Appendix A.2 provides additional details on hardware, software, and model selection.

4.4 Inference budget

We imposed an inference time limit on CascadER and all baselines. First, we set a global maximum inference time of 24 hours on a single NVIDIA Quadro RTX 8000 GPU with 48 GB of RAM as a hard limit. Then, we set dataset-specific budgets $\leq$ 24H roughly as a function of the test set size and KG size. For example, WN18RR has fewer test queries and shorter text lengths (Table 1) than FB15K-237, but 3$\times$ as many answer entities. Given a budget of 24H for FB15K-237, our largest dataset, we decided to allow 25% of that time to WN18RR to account for these differences. Concretely, we set inference time limits to 2 hours for our two smallest datasets CoDEx-S and RepoDB, 24 hours for our two largest datasets CoDEx-M and FB15K-237, and 6 hours for WN18RR. To understand the impact of our chosen time limits, we vary these budgets in an ablation in § 4.5.

	FB15K-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RESCAL	0.3559	0.2629	0.3926	0.5406	0.4666	0.4387	0.4797	0.5172
TransE	0.3128	0.2206	0.3473	0.4973	0.2278	0.0531	0.3682	0.5201
ComplEx	0.3477	0.2533	0.3836	0.5359	0.4749	0.4381	0.4898	0.5474
RotatE	0.3333	0.2396	0.3676	0.5218	0.4781	0.4395	0.4941	0.5527
StAR	0.296	0.205	0.322	0.482	0.401	0.243	0.491	0.709
KG-BERT	0.267	0.172	0.298	0.458	0.331	0.203	0.383	0.597
KGE + KGE	0.3630	0.2672	0.4016	0.5535	0.4900	0.4521	0.5016	0.5617
KGE + StAR	0.3643	0.2709	0.3989	0.5522	0.5385	0.4716	0.5645	0.6651
KGE + KG-BERT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
SOTA	**0.415**	**0.321**	**0.454**	**0.599**	0.551	0.459	0.594	0.732
CASCADER	0.3860	0.2903	0.4231	0.5782	**0.5651**	**0.4756**	**0.6126**	**0.7379**

Table 2: CASCADER outperforms or is competitive with the state of the art on FB15K-237 and WN18RR. **Bold + underline**: Best performance. Underline: Second-best performance. The performance of StAR, KG-BERT, and SOTA are reported from papers referenced in 4.2. OOT refers to out-of-time using our inference cost budget (§ 4.4).

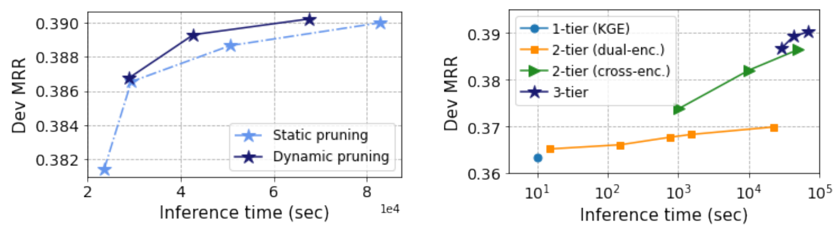
	CoDEx-M				RepoDB			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RESCAL	0.3173	0.2444	0.3477	0.4557	0.4351	0.3144	0.4903	0.6767
TransE	0.3026	0.2232	0.3363	0.4535	0.3472	0.2043	0.3728	0.6400
ComplEx	0.3365	0.2624	0.3701	0.4758	0.4620	0.3406	0.5225	0.7043
RotatE	OOM	OOM	OOM	OOM	0.2971	0.1811	0.4903	0.5314
StAR	0.2726	0.1888	0.3042	0.4342	0.3472	0.2043	0.4102	0.6400
KG-BERT	OOT	OOT	OOT	OOT	0.2991	0.1602	0.3428	0.5996
KGE + KGE	0.3466	0.2695	0.3808	0.4925	0.4637	0.3398	0.5262	0.7081
KGE + StAR	0.3554	0.2767	0.3901	0.5064	0.4774	0.3496	0.5434	0.7208
KGE + KG-BERT	OOT	OOT	OOT	OOT	0.5101	0.3713	0.5771	0.7799
SOTA	0.3365	0.2624	0.3701	0.4758	-	-	-	-
CASCADER	**0.3830**	**0.2998**	**0.4221**	**0.5423**	**0.5156**	**0.3817**	**0.5831**	**0.7814**

Table 3: CASCADER achieves state-of-the-art test performance on CODEX-M and REPODB.

4.5 Results and discussion

Table 2 and 3 provide link prediction performance results for FB15K-237 and WN18RR, and CODEX-M and REPODB, respectively; Table 7 in Appendix A.3 provides results on CODEX-S, omitted here for brevity. We observe that CASCADER achieves robust and appreciable gains over baselines across datasets, setting a new state of the art on WN18RR, CODEX-S, CODEX-M, and REPODB, and performing second to the reported SOTA on FB15K-237. It outperforms the best KGE by up to 8.70 points MRR (WN18RR) and the best LM by up to 16.84 points MRR (REPODB), demonstrating that cross-modal ensembling can significantly improve upon single-modality approaches.

We also remark that full additive ensembling is *not* necessary to maximize accuracy. Our KGE + KG-BERT additive ensemble baseline is competitive on CODEX-S and REPODB, but it encounters out-of-time errors on the other datasets. By contrast, CASCADER achieves state-of-the-art or competitive performance on all five datasets while staying within our time limits. This suggests that full additive ensembling is not necessary to achieve the majority of gains in link prediction, and that cascaded reranking is sufficient.



(a) Dynamic pruning balances accuracy and efficiency better than static pruning.

(b) CascadER is effective at any level of cost constraint compared to the base KGE.

Figure 3: Top-left corner is best: Pareto curve analysis on the dev set of FB15K-237. We use quantiles $q \in \{0.5, 0.75, 0.9, 0.95, 1\}$ in our analyses and exclude any quantiles that lead to CascadER exceeding our inference time limit of 24 hours.

Pareto curve analysis We provide a Pareto curve analysis to characterize the accuracy-efficiency tradeoff of CascadER. In Figure 3, we plot the accuracy (dev MRR) and efficiency (inference cost in wall-clock time) against CascadER's key hyperparameters, the candidate pruning strategy and the number of tiers. We observe that dynamic pruning balances accuracy and efficiency better than static pruning. Figure 3a shows that dynamic pruning leads to steeper MRR improvements than static pruning, with comparable inference times.

Consistent with the information retrieval literature [Matsubara et al., 2020, Luan et al., 2021], cross-encoders improve ranking performance more than their dual-encoder counterparts. Figure 3b confirms that two-tiered CascadER with a cross-encoder is much more accurate than two-tiered CascadER with a dual-encoder. At an inference time of around 1000 seconds, our two-tiered dual-encoder and cross-encoder architectures achieve 0.3683 and 0.3738 MRR respectively, suggesting that reranking *very few* candidates with a cross-encoder is often more beneficial than reranking *many* candidates with a dual-encoder.

Parameter comparison A natural question is whether CascadER's improvement is simply a result of increased representational capacity, i.e., more parameters. Depending on the dataset, the number of KGE parameters is on the order of 100K-10M, scaling linearly (for most architectures) with the number of entities and relations in the KG. By contrast, each LM, when implemented with BERT-Base, has roughly 100M parameters, regardless of KG size; therefore, CascadER has more parameters than all KGE baselines. However, note that LMs *alone* for the link prediction task are not competitive with KGEs, as shown in Tables 2 and 3, which suggests that increased parameter count does not fully explain performance improvement. Moreover, the Pareto improvements we observe are in comparison to ensembles of the same number of parameters (e.g., 2-tier dual-encoding CascadER vs 2-tier cross-encoding CascadER, Figure 3b), where we achieve up to an order of magnitude efficiency improvement at the same or better accuracy.

Inference budget ablation We next analyze the impact of changing the inference budget (§ 4.4) on CascadER's performance. As shown in Table 4, CascadER's performance does not change drastically when tightening the budget. Halving the inference budget for CoDEx-M from 24H to 12H, we get a test accuracy of 0.3787 MRR for CascadER, a

	CoDEx-M	WN18RR
Best baseline	0.3554	0.5510
CascadER, half budget	0.3787	0.5590
CascadER, full budget	0.3830	0.5651

Table 4: CascadER still maintains SOTA performance (MRR) under tightened inference budgets: Inference time limit ablation. "Full budget" refers to the time limits imposed in § 4.4, and "half budget" refers to a time limit of half that.

	(KGE, KGE)	(KGE, dual-enc.)	(KGE, cross-enc.)
RepoDB	0.7371	0.3494	0.2025
CoDEx-S	0.8081	0.7265	0.6163
CoDEx-M	0.6471	0.5406	0.4865

Table 5: The ranks of gold answers are least correlated (Pearson's correlation coefficient, all p-values 0) between the KGE and the cross-encoder on the dev set, suggesting that these two model types provide the most diverse or complementary link prediction performance.

drop of $<$ 1 point MRR from our best CascadER under 24H, and still an improvement of 2 points MRR over the next-best baseline. Similarly, halving the inference budget for WN18RR from 6H to 3H, we again observe a drop of $<$ 1 point MRR.

Qualitative analysis Finally, we elucidate the benefits of cross-modal ensembling from the perspective of model *diversity*, a key characteristic of effective ensembles [Kuncheva and Whitaker, 2003]. We explore two simple facets of diversity between pairs of models in an ensemble. The first is **rank correlation**, or the Pearson correlation coefficient of the gold answer ranks between two models. As shown in Table 5, rank correlation is highest between pairs of KGEs and lowest between the KGE and cross-encoder. This observation corresponds well with ensemble performance, as we find that the most "diverse" ensembles in terms of rank correlation (i.e., those that combine a KGE + cross-encoder) are the most competitive.

Next, we consider each model's empirical **score distributions** to two queries in Figure 4. The figure demonstrates that the cross-encoder's score distributions are skewed left compared to those of the KGE and the dual-encoder. Moreover, we find that this trend holds across queries and datasets, suggesting a fundamental difference in scoring behavior. We hypothesize that cross-encoders may filter out irrelevant candidate answers to queries more aggressively, perhaps because they model term overlap between text pairs with relatively high precision compared to dual-encoder models that do not use cross-attention [Luan et al., 2021]. Again, this finding corresponds well with ranking performance, as the model pairs with more diverse score distributions (i.e., KGEs and cross-encoders) make stronger cascades.

Finally, having established the importance of diversity, we answer the question: When ensembling highly diverse models, how crucial is additive reweighting as defined in (1), versus simply summing the models' individual scores without reweighting? Across all datasets, we observe CascadER's dev MRR drops 6-8 points without reweighting (see Appendix A.3 for exact comparisons). To explain this phenomenon, we investigate the **average margin** between the gold answer and all negative candidates for a query, i.e., for query i with gold tail entity j^+ and N^- negative candidates j^-, the average margin is defined as $\frac{1}{N^-}\sum_{j^-} S_{ij}^+ - S_{ij}^-$.

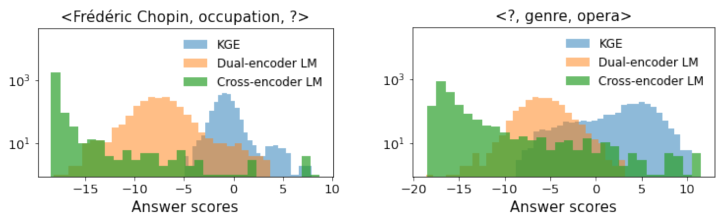
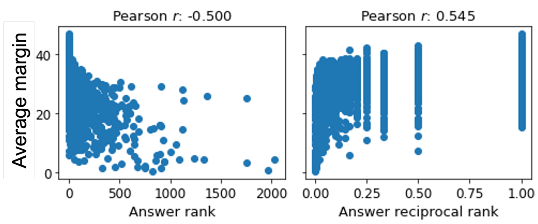
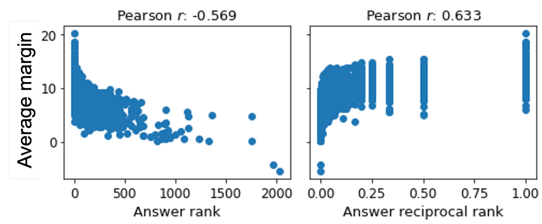


Figure 4: The cross-encoder's score distributions are skewed left compared to the KGE's and dual-encoder's score distributions. Shown are two randomly-selected link prediction queries from CoDEx-M. The trends in the plots hold across queries and datasets.



(a) KGE + cross-enc. without reweighting



(b) KGE + cross-enc. with reweighting

Figure 5: Ensembling with additive reweighting preserves the correlation between gold answer ranks and average margins.

In Figure 5, we plot the gold ranks and reciprocal ranks of answer entities on CoDEx-M against their average margins with and without reweighting a KGE + cross-encoder ensemble. We observe that the correlation between margins and gold ranks is higher under additive reweighting, which suggests that reweighting helps preserve the confidence signals in the base models' margins, whereas ensembling without reweighting dilutes these signals.

5. Conclusion

In this paper, we considered the task of KG link prediction with cross-modal ensembles. Motivated by the inherent accuracy-efficiency tradeoff, we proposed CascadER, a novel cross-modal reranking architecture that uses deep language models to rerank the outputs of knowledge graph embeddings. We showed that CascadER achieves state-of-the-art performance on multiple link prediction benchmarks by effectively combining structure and text, while improving efficiency over our strongest ensemble baseline by orders of magnitude.

Our work opens up several avenues for future research. For one, more advanced candidate pruning strategies may further increase the efficiency of CascadER while maintaining accuracy. For another, "hybrid" LMs that attempt to interpolate between the efficiency of dual-encoders and the effectiveness of cross-encoders [Khattab and Zaharia, 2020, Luan et al., 2021] may improve CascadER's ability to balance these two desiderata. Finally, extensions of CascadER may help solve the inductive link prediction, in which novel entities/relations are presented at test time [Galkin et al., 2022], as contextual LMs are naturally inductive and can correct the shortcomings of transductive KGEs in this setting.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1522. URL https://aclanthology.org/D19-1522.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems*, 2013.

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - a knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 165–174, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.22. URL https://aclanthology.org/2020.emnlp-demos.22.

Adam S Brown and Chirag J Patel. A standard database for drug repositioning. *Scientific data*, 4(1):1–7, 2017. URL https://www.nature.com/articles/sdata201729.

Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *bioRxiv*, 2022. URL https://www.biorxiv.org/content/10.1101/2022.05.01.489928v2.

Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J Shane Culpepper. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 445–454, 2017. URL https://dl.acm.org/doi/pdf/10.1145/3077136.3080819.

Daniel Daza, Michael Cochez, and Paul Groth. Inductive entity representations from text via link prediction. In *Proceedings of the Web Conference 2021*, pages 798–808, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3442381.3450141.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. An open challenge for inductive link prediction on knowledge graphs. *arXiv preprint arXiv:2203.01520*, 2022. URL https://arxiv.org/pdf/2203.01520.

Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J Shane Culpepper. Joint optimization of cascade ranking models. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 15–23, 2019. URL https://dl.acm.org/doi/pdf/10.1145/3289600.3290986.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3458754.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/pdf/1905.01969.pdf.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020. URL https://arxiv.org/pdf/2002.00388.pdf.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3397271.3401075.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.153. URL https://aclanthology.org/2020.coling-main.153.

Vid Kocijan and Thomas Lukasiewicz. Knowledge base completion meets transfer learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6521–6533, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.524. URL https://aclanthology.org/2021.emnlp-main.524.

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001. URL https://www.aeaweb.org/articles?id=10.1257/jep.15.4.143.

Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003. URL https://link.springer.com/content/pdf/10.1023/A:1022859003006.pdf.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021. URL https://www.morganclaypool.com/doi/pdfplus/10.2200/S01123ED1V01Y202108HLT053.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021. doi: 10.1162/tacl_a_00369. URL https://aclanthology.org/2021.tacl-1.20.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1577–1580, 2020. URL https://dl.acm.org/doi/pdf/10.1145/3397271.3401266.

Rahul Nadkarni, David Wadden, Iz Beltagy, Noah Smith, Hannaneh Hajishirzi, and Tom Hope. Scientific language models for biomedical knowledge base completion: An empirical study. In *3rd Conference on Automated Knowledge Base Construction*, 2021. URL https://arxiv.org/pdf/2106.09700.pdf.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, 2011. URL https://icml.cc/2011/papers/438_icmlpaper.pdf.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015. URL https://ieeexplore.ieee.org/document/7358050.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*, 2019.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=BkxSmlBFvr.

Tara Safavi and Danai Koutra. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 8328–8350, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.669.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2019.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *ACL CVSC Workshop*, 2015.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL https://aclanthology.org/D15-1174.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1434–1444, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1136. URL https://aclanthology.org/P16-1136.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2016. URL http://proceedings.mlr.press/v48/trouillon16.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001. URL https://ieeexplore.ieee.org/iel5/7768/21353/00990517.pdf.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748, 2021. URL https://dl.acm.org/doi/pdf/10.1145/3442381.3450043.

Lidan Wang, Jimmy Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 105–114, 2011. URL https://dl.acm.org/doi/pdf/10.1145/2009916.2009934.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017. URL https://ieeexplore.ieee.org/iel7/69/4358933/08047276.pdf.

Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M Kitani, Yair Alon, and Elad Eban. Wisdom of committees: An overlooked approach to faster and more accurate models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/pdf?id=MvO2t0vbs4-.

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Found. Trends Databases*, 10(2-4):108–490, 2021. doi: 10.1561/1900000064. URL https://www.nowpublishers.com/article/DownloadSummary/DBS-064.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. URL https://ojs.aaai.org/index.php/AAAI/article/view/10329/10188.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=zeFrfgyZln.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2015.

Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019. URL https://arxiv.org/pdf/1909.03193.pdf?ref=https://githubhelp.com.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f6a673f09493afcd8b129a0bcf1cd5bc-Abstract.html.

Appendix A.

A.1 Additional related work

Joint modeling on KGs The task of inferring novel links in KGs has been widely studied. The most prevalent approaches are structure-only KG embeddings [Nickel et al., 2011, Bordes et al., 2013, Trouillon et al., 2016, Sun et al., 2019, Balazevic et al., 2019, Ji et al., 2020]. That said, prior to the advent of pretrained contextual LMs, a few cross-modal structure and text approaches for link prediction were proposed [Toutanova et al., 2015, 2016, Xie et al., 2016]. Such approaches rely on convolutional text representation architectures to obtain embeddings of entities or relations using, e.g., entity descriptions [Xie et al., 2016] or textual relation mentions [Toutanova et al., 2015]. These text-based embeddings are then composed using structural KG embedding scoring functions to score novel links in the KG.

More recently, motivated by the successes of Transformer language models, pretrained Transformer LMs like BERT [Devlin et al., 2019] have begun to gain traction for variants of the link prediction task [Yao et al., 2019, Kim et al., 2020, Daza et al., 2021, Wang et al., 2021, Nadkarni et al., 2021]. The approaches most related to CascadER are the ensembles considered by Wang et al. [2021] and Nadkarni et al. [2021], both of which construct additive ensembles of structural KG embeddings and contextual LMs. Compared to these baseline ensembles, CascadER improves accuracy and/or efficiency over both.

Cascade models Multi-stage cascade ensembles have been successful in computer vision [Viola and Jones, 2001, Wang et al., 2022] and text retrieval [Wang et al., 2011, Chen et al., 2017, Gallagher et al., 2019, Lin et al., 2021]. Recently, several studies have proposed to use BERT as a late-stage ranker in multi-stage document retrieval [Nogueira et al., 2019] and passage retrieval [Matsubara et al., 2020] pipelines. Similar to our work, these studies are motivated by the observation that using BERT in a multi-stage cascaded setting can significantly boost retrieval accuracy while maintaining efficiency [Lin et al., 2021]. Yet other studies have attempted to balance the effectiveness-efficiency tradeoff by proposing dual-encoding architectures that are more efficient but usually less effective than cross-encoder BERT models for information retrieval [Reimers and Gurevych, 2019, Humeau et al., 2020, Karpukhin et al., 2020, Xiong et al., 2020, Khattab and Zaharia, 2020]. Our work builds upon all of these important insights, which have been instrumental in scaling contextual LMs to large-scale text ranking. As far as we are aware, we are the first to bridge these ideas with the traditional graph learning task of link prediction.

A.2 Model selection

We implement all KG embeddings using the open-source LibKGE PyTorch library [Broscheit et al., 2020]. We use the pretrained KGE checkpoints provided by LibKGE for FB15K-237 and WN18RR. For the other datasets, we follow a similar hyperparameter tuning strategy to that proposed by Ruffinelli et al. [2020] for tuning our KGE baselines.

We implement all LMs with the Huggingface transformers PyTorch library [Wolf et al., 2020] using the same base language model, which is BERT-Base [Devlin et al., 2019] for all benchmarks except RepoDB, and PubMedBERT [Gu et al., 2021] for RepoDB. We use the following hyperparameters: Batch size of 16, learning rate of 10^{-5}, and 10 epochs. We use a maximum sequence length of 32, 64, and 256 respectively for WN18RR, RepoDB,

	Inference time total
FB15K-237	20 hours
WN18RR	3.5 hours
CoDEx-S	26 min
CoDEx-M	17 hours
RepoDB	5 min

Table 6: Wall-clock inference time of CascadER using a single RTX 8000 GPU.

	MRR	H@1	H@3	H@10
RESCAL	0.4040	0.2935	0.4494	0.6225
TransE	0.3540	0.2185	0.4218	0.6335
ComplEx	0.4646	0.3714	0.5038	0.6455
RotatE	0.2587	0.1586	0.2916	0.4609
StAR	0.3540	0.2306	0.4051	0.6007
KG-BERT	0.2849	0.1472	0.3310	0.5848
KGE + KGE	0.4665	0.3712	0.5082	0.6518
KGE + StAR	0.4751	<u>0.3717</u>	0.5249	0.6712
KGE + KG-BERT	<u>0.4812</u>	<u>**0.3764**</u>	<u>0.5290</u>	<u>**0.6898**</u>
SOTA	0.4646	0.3714	0.5038	0.6455
CascadER	<u>**0.4839**</u>	<u>**0.3764**</u>	<u>**0.5383**</u>	<u>0.6871</u>

Table 7: CascadER achieves state-of-the-art test performance on CoDEx-S. <u>**Bold + underline**</u>: Best performance. <u>Underline</u>: Second-best performance.

and all other datasets. For the dual-encoder LM we use 16 negative samples per positive. For the cross-encoder LM we use 2 negative samples per positive.

For all of the ensemble baselines and CascadER, we tune the weighting hyperparameter $\alpha \in [0.05, 0.95]$. All experiments are conducted on a single NVIDIA Quadro RTX 8000 GPU with 48 GB of RAM. All main results reported in the paper use a three-tiered structure with no pruning between tiers one and two and dynamic pruning at $q = 0.9$ between tiers two and three. We provide exact wall-clock inference time for CascadER in Table 6.

A.3 Additional results

Results on CoDEx-S Table 7 provides link prediction performance on the CoDEx-S dataset [Safavi and Koutra, 2020]. CascadER again achieves state-of-the-art performance on this dataset, suggesting its wide applicability across KGs.

Importance of additive reweighting Table 8 compares performance of a KGE + cross-encoder ensemble with and without additive reweighting, as defined in Eq. (1). We observe that MRR drops 6-8 points without the reweighting term, confirming that additive reweighting is key to preserving the base models' confidence signals in the ensemble.

	No reweighting	Reweighting
RepoDB	0.4498	0.5156
CoDEx-S	0.4204	0.4839
CoDEx-M	0.3114	0.3830

Table 8: Additive reweighting is crucial to ensembling KGEs and LMs. We compare the MRR of the best CascadER architecture *with* additive reweighting as defined in Eq (1) to the best CascadER architecture without the reweighting term.