Entity Embeddings for Entity Ranking: A Replicability Study

Pooja Oza and Laura Dietz(✉)

University of New Hampshire, Durham, NH 03824, USA
pho1003@wildcats.unh.edu, dietz@cs.unh.edu

Abstract. Knowledge Graph embeddings model semantic and structural knowledge of entities in the context of the Knowledge Graph. A nascent research direction has been to study the utilization of such graph embeddings for the IR-centric task of entity ranking. In this work, we replicate the GEEER study of Gerritse et al. [9] which demonstrated improvements of Wiki2Vec embeddings on entity ranking tasks on the DBpediaV2 dataset. We further extend the study by exploring additional state-of-the-art entity embeddings ERNIE [27] and E-BERT [19], and by including another test collection, TREC CAR, with queries not about person, location, and organization entities. We confirm the finding that entity embeddings are beneficial for the entity ranking task. Interestingly, we find that Wiki2Vec is competitive with ERNIE and E-BERT.

Our code and data to aid reproducibility and further research is available at https://github.com/poojahoza/E3R-Replicability.

Keywords: Entity retrieval · Entity embeddings · Knowledge graphs

1 Introduction

We study the problem of entity ranking since users seek entities in response to their queries [11,20], or such entities can be helpful in improving document rankings [6]. The queries can range from short factoid questions (e.g., "Who is the mayor of Berlin?") that seek a particular entity to the queries that request a list of entities (e.g., "Professional sports teams in Philadelphia"). Knowing relevant entities is also helpful when synthesizing relevant information on popular science topics (e.g., "tell me more about horseshoe crabs"). Given a query, the entity ranking task is to return a list of entities ordered by the relevance of each entity to the query. Such entities are taken from a given Knowledge Graph, such as Wikipedia or DBpedia.

Previous work on entity ranking either uses hand-crafted features within a Learning-To-Rank framework [7,22] or leverages information about entities available in a Knowledge Graph such as types [1,2,10,18] and relations [4,5,23]. However, these entity ranking systems consider only lexical matching between the queries and the entity information and disregard any semantic and structural information of the entities. To overcome this, re-ranking models that use

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13982, pp. 117–131, 2023.
https://doi.org/10.1007/978-3-031-28241-6_8

Knowledge Graph embeddings such as TransE [14], and Wiki2Vec [9] have been proposed. Knowledge Graph embeddings capture the structural and semantic information of the entities in the context of the Knowledge Graph. They project the entities and relations in a continuous vector space that preserves information about the structure of the Knowledge Graph. Such Knowledge Graph embeddings have shown to be successful in IR-centric entity ranking tasks, with explicit queries [14].

Additionally, knowledge-enhanced BERT models such as ERNIE [27] and E-BERT [19] have been proposed in recent years which augments the successful BERT model with the entity information through Knowledge Graph embeddings such as TransE [3] and Wiki2Vec [26]. The entity embeddings generated from these models are a fusion of the entity information from the Knowledge Graph and rich contextual information from BERT embeddings. These knowledge-enhanced BERT models have been demonstrated to improve the performance of entity-centric NLP downstream tasks such as relation classification, entity typing, and entity linking.

In this paper, we reproduce and replicate the work of Gerritse et al. [9] (GEEER) which shows that Knowledge Graph embeddings such as Wiki2Vec are beneficial to improve the performance of the entity-oriented search. We choose to reproduce and replicate this work as it is among the first few papers to study the utilization of Knowledge Graph embeddings in an IR-centric entity ranking task. This is a critical work with a high impact in the field of IR and hence reproducibility with further exploration is important. Within the GEEER framework, we explore the efficacy of pretrained entity embeddings in the entity ranking task with different datasets through replicability experiments. We incorporate new entity embeddings, new datasets, and different learning-to-rank methods in the study. In particular, we study the effect of neural fine-tuning of the embeddings for the ranking task.

In the following, we refer to both Knowledge Graph embeddings and entity embeddings of knowledge-enhanced BERT models as entity embeddings.

Experiments: In this work, we perform several sets of experiments to study whether the original findings still hold.

1. **Reproducibility:** Using the same code, entity embeddings, dataset, and entity re-ranking framework as given in the original paper [9] we confirm the findings of the original work.
2. **Replicability:** For these experiments, we re-implement the method of Gerritse et al. [9], using original and additional pretrained entity embeddings, and explore small changes in the setup.
3. **New Dataset:** While the original dataset was asking about people, organizations, and locations, we are adding another dataset, TREC CAR, which asks about other entity types. We confirm that the original findings still hold.
4. **Effect of Fine-tuning:** We further analyze the effect of fine-tuning the embeddings (as opposed to directly using pretrained embeddings). We study fine-tuning with both, point-wise and pair-wise ranking losses and demonstrate that the gains are even more significant.

5. **Study on Missing Entities:** Pretrained entity embeddings often don't contain embeddings for all candidate entities obtained via initial rankings. We quantify performance losses due to missing entity embeddings separately from those due to quality issues with available embeddings.

Findings in the original paper: In the original paper [9], the authors find two important results, of which we focus on the first: (1) Entity embeddings are advantageous to improve the performance of the entity ranking task and (2) Entity embeddings that contain both context and structural information of the Knowledge Graph perform better than the entity embeddings that contain only contextual information.

Findings in our paper: In our paper, we concentrate only on the first finding of the original paper, i.e., entity embeddings are advantageous to improve entity ranking task performance. We are able to reproduce the experiment, and additionally can replicate it under several changes to the setup.

We make the following additional observations: (1) Pretrained and fine-tuned entity embeddings help to improve the performance of entity ranking. While pretrained entity embeddings provide only a slight gain over the baselines, fine-tuned embeddings improve the performance by a significantly large margin. (2) Pretrained and fine-tuned Wiki2Vec embeddings outperform or perform similarly to knowledge-enhanced BERT models ERNIE and E-BERT. (3) In line with prior work [13], we find that in most cases fine-tuning with a pair-wise loss performs better than a point-wise loss for both Wiki2Vec and ERNIE. (4) We find that the pretrained entity embeddings help to improve performance losses of ranking baselines.

2 Related Work

2.1 Knowledge Graph Embeddings

Knowledge Graph embeddings are vector representations of the entities present in the Knowledge Graph. Such embeddings capture the semantic and structural information of the entities. Bordes et al. [3] proposed TransE, a translational-based model, that learns the embeddings of both entities and relations on the modeling assumption that the relation $\boldsymbol{r}$ is a translation between two entities $\boldsymbol{h}$ and $\boldsymbol{t}$. TransE projects both entities and relations in the same vector space. However, since TransE considers only 1-to-1 relations, it does not work well with 1-to-N, N-to-N, and N-to-1 relations. To overcome this issue, TransH [25] model was proposed that projects each relation $\boldsymbol{r}$ with two vectors. TransR [12] projects each relation $\boldsymbol{r}$ in its own space and projects the entities $\boldsymbol{h}$ and $\boldsymbol{t}$ with respect to the relation $\boldsymbol{r}$. Recently, Yamada et al. [26] proposed Wiki2Vec that learns entity (and word) embeddings using text and structural information from Wikipedia. We further detail entity embeddings used in our work in Section 3.

2.2 Knowledge-enhanced BERT Models

Recently, knowledge-enhanced BERT models are proposed that infuse knowledge into the BERT model through knowledge graph embeddings such as TransE [3] and Wiki2Vec [26]. ERNIE [27] incorporates entity information in the BERT model through TransE entity embeddings in pretraining, while E-BERT [19] adapts entity embeddings of Wiki2Vec to BERT without any additional pretraining. KEPLER [24] utilizes entity descriptions corresponding to the entities in relation triples and jointly optimizes Knowledge Graph and Language Model representations. KELM [15] injects knowledge in the BERT model via multi-relational subgraphs from the Knowledge Graph and text. ERNIE and E-BERT models are further explained in Sect. 3.

2.3 Entity Retrieval

Retrieval through Pseudo-Relevance Feedback Documents. Prior work of Entity Retrieval uses the unstructured text of pseudo-relevance feedback documents. Dalton et al. [6] uses the entities linked in the feedback documents and the fields of the Knowledge Graph such as entity links and the candidate set of entities for query expansion to retrieve a ranking of documents. Entities and text features such as co-occurrence, and mention features can be combined through a Learning-To-Rank approach [7]. Furthermore, Knowledge Graph links and entity co-occurrence from the feedback runs can be integrated [17].

Retrieval through Knowledge Graph Embeddings. Gerritse et al. [9] use Knowledge Graph embeddings of Wiki2Vec to determine the embedding score between the candidate set of entities and entities linked in the queries. For the final ranking, the embedding score is interpolated with the initial candidate relevance score through a Learning-To-Rank approach. Liu et al. [14] use TransE entity embeddings in the entity retrieval framework. The authors utilize the TransE embeddings to calculate the similarity between the entities in the query and candidate set of entities and further interpolate it through the Learning-To-Rank methods RankSVM and Coordinate Ascent.

Retrieval through Fielded Retrieval Models. A variation of the well-known retrieval method Sequential Dependence Model (SDM) [21,28] uses Knowledge Graph fields such as entity types, names and also documents to determine the relevance of the entities.

3 Approach

We follow the framework of Gerritse et al. [9] to rank entities. To obtain the final ranking of entities, embedding scores are determined using the entity embeddings of Wiki2Vec, ERNIE, and E-BERT which we describe below.

3.1 Entity Embeddings

Wiki2Vec. Wiki2Vec [26] learns a shared embedding space for both, word and entity embeddings, using data from Wikipedia. In particular, the model learns word embeddings using the Word2Vec Skipgram model [16], which uses a fixed-size context to learn the embeddings for each word. The similarity between the embeddings of the two entities is trained to coincide with Wikipedia's link graph. The final element of the model relates both words and entities through anchor text. These three elements are combined linearly to form the final loss function for training.

ERNIE. ERNIE [27] injects TransE entity embeddings in BERT word embeddings to enhance BERT with knowledge. It aligns the entity embeddings of TransE with the BERT word embedding of the first wordpiece token of the corresponding entity mention to generate encoded embeddings in a common embedding space. TransE [3] is a translational model that projects the entities and relations of the Knowledge Graph relation triples in a shared embedding space. The pretraining objective of the model for the knowledge fusion predicts masked entities through aligned tokens. ERNIE is further fine-tuned on NLP tasks of Relation Classification and Entity Typing.

E-BERT. E-BERT [19] infuses Wikipedia knowledge to contextualized BERT wordpiece embeddings by aligning BERT word embeddings with entity embeddings of Wiki2Vec. As word and entity embeddings share the same embedding space in Wiki2Vec, E-BERT uses the word embeddings of Wiki2Vec to learn the weight matrix through the linear transformation of Wiki2Vec word embeddings to BERT-like embeddings. Using the learned weight matrix, it constructs a function to align the entity embeddings of Wiki2Vec with the BERT word embeddings. E-BERT is fine-tuned on the downstream NLP tasks of Relation Classification and Entity Linking.

3.2 Entity Re-Ranking Framework of Gerritse et al.

In this section, we describe the entity re-ranking framework used in the original paper [9]. The authors use a two-stage entity re-ranking framework to identify the relevant entities for the query.

For our reproducibility and replicability experiments, we follow Gerritse et al. [9] and use an existing entity retrieval method to produce a candidate set of entities at the first stage of the framework. In the second stage of the framework, we first get the entity-embedding-based similarity score of each candidate entity with the entities present in the query. Then the candidate set of entities is re-ranked using interpolation.

For a query Q, query entities $E(Q)$ are identified with an entity linker and the link confidence scores $s(e)$ are retained. The entity-embedding-based similarity score for every candidate entity E and query Q is obtained as follows:

$$F(E, Q) = \sum_{e \in E(Q)} s(e) \cdot \cos(\overrightarrow{E}, \overrightarrow{e}) \tag{1}$$

To determine the final score of the entities, we combine the entity-embedding-based similarity score with the relevance score of the first stage entity retrieval method via interpolation as follows (Eq. 6 in the original paper):

$$score_{total}(E, Q) = (1 - \lambda) \cdot score_{other}(E, Q) + \lambda \cdot F(E, Q) \quad \lambda \in [0, 1] \tag{2}$$

Learning-to-rank frameworks can readily learn unnormalized weighted aggregations, through coefficients λ_1 and λ_2 on two features, which is a rank-equivalent reparametrization of the original model.

$$score_{total}(E, Q) = \lambda_1 \cdot score_{other}(E, Q) + \lambda_2 \cdot F(E, Q) \quad \lambda_1, \lambda_2 \in R \tag{3}$$

3.3 Fine-Tuning with Neural Networks

While the original paper determines the re-ranking of the candidate set through interpolation of embedding scores and candidate relevance score retrieved in the first stage, here we describe our fine-tuning approach within end-to-end entity re-ranking.

For a query Q, we identify the entities $E(Q)$ for the query and average their embeddings to obtain a single entity embedding E_Q of the query. We train a similarity metric between query embeddings E_Q and candidate entity embedding E_c as follows: We train a bilinear projection with $E_Q^T W E_c$ to capture correlations across different entries. This is followed by a linear layer to predict the rank score. The model is trained with a point-wise loss (binary cross-entropy loss) and a pair-wise loss (margin ranking loss with `tanh` activation) using the test collection.

4 Experimental Setup

We address the following research questions in our experiments:[1]

- **RQ1:** Can we reproduce the findings of Gerritse et al. [9]?
- **RQ2:** To what extent do the findings of the original paper generalize to other entity embeddings and to another dataset collection that does not focus on frequently used entities such as persons, organizations, or locations?
- **RQ3:** How much improvement can we achieve when we fine-tune the entity embeddings?
- **RQ4:** Missing entities aside, what is the quality of the entity embeddings?

4.1 RQ1: Reproducibility

To reproduce the results, we use the dataset DBpediaV2 which is used in the original paper [9]. The dataset consists of four different types of queries:

[1] Our code and data are available at https://github.com/poojahoza/E3R-Replicability.

(1) INEX-LD contains IR-styled keywords. e.g., "electronic music genre"; (2) SemSearchES contains short one entity search type of queries, e.g., "brooklyn bridge" (3); QALD2 consists of natural questions which are answerable by entities, e.g., "who is the mayor of Berlin?"; (4) ListSearch which consists of queries searching for a list of entities, e.g., "Professional sports team in Philadelphia". The dataset consists of 467 queries and has 49280 assessed query-entity pairs.

The existing entity retrieval method used to retrieve the top 1000 candidate set of entities is BM25F-CA, which is the best-performing method for DBpediaV2 and provided by the creators. We use the Wiki2Vec embeddings trained on the 2019-07 dump by the authors of the original paper [9] to calculate the embedding reranking score. We use Wiki2Vec embeddings with 100 dimensions for all reproducibility experiments.

Interpolation: To perform the interpolation, we use the Learning-To-Rank (L2R) approach by utilizing the RankLib library, version 1.12, as used in the original paper. We train the L2R with Co-ordinate Ascent, optimized for NDCG. We perform all experiments on 5-fold cross-validation, on the folds given in the DBpediaV2 collection.

To reproduce the results, we use the code, Wiki2Vec embeddings, and first stage run files provided by the authors of the original paper.[2]

4.2 RQ2 and RQ3: Replicability and Fine-tuning

In addition to DBpediaV2, which focuses on people, organization, and location entities, we include an additional dataset from TREC CAR, that emphasizes other entity types. The TREC Complex Answer Retrieval (CAR) [8] provides test collections for the entity ranking task in *Y2Test*. We use BenchmarkY1-train-automatic for fine-tuning and use *Y2 Test*-automatic for training the interpolation and evaluation. Y2-test consists of 65 topical queries such as "air pollution".

For both datasets, we use binary relevance judgments: 0 (non-relevant) and 1 (relevant) and evaluate with mean-average precision (MAP) and R-precision, i.e., precision at the cutoff of the number of relevant entities. We entity-link the queries using the TAGME entity linker.

Baseline: For a first-stage entity retrieval method and baseline, we use *BM25F-CA* for DBpediaV2 experiments and a high-performing input ranking for ENT-Rank called ExpEcm[3] for the TREC CAR dataset.

Embeddings: We use the Wiki2Vec 100-dimensional embeddings trained by Gerritse et al. on the Wikipedia 2019-07 dump. Additionally, we use the pretrained 100-dimensional ERNIE [27] and 768-dimensional E-BERT [19] embeddings.

[2] Source code for GEEER is available at https://github.com/informagi/GEEER.
[3] ExpEcm available at https://www.cs.unh.edu/~dietz/appendix/ent-rank/.

Table 1. (Relevant) candidate entities for which embeddings are not available.

Dataset	Missing Candidate Entities			Missing Relevant Entities		
	Wiki2Vec	ERNIE	E-BERT	Wiki2Vec	ERNIE	E-BERT
TREC CAR	4.06%	16.12%	3.04%	0.24%	0.82%	0.11%
DBpediaV2	16.47%	22.44%	21.89%	5.31%	6.66%	6.74%

Interpolation: We change the learning-to-rank framework to learn linear interpolation. We use the Rank-Lips[4] library optimizing for Mean Average Precision (MAP) with Coordinate Ascent, using five random restarts. Additionally, for DBpediaV2, we use different cross-validation folds. For all the replicability experiments, we re-implement the code of the GEEER entity ranking framework.

Fine-tuning: As an optional step, embeddings are fine-tuned for the entity ranking task with a neural network, we use the same datasets, evaluation, and baselines as we do for the replicability experiments.

We use a batch size of 1000, 10 epochs, 1000 warmup steps, and a 2e-05 learning rate.

We apply two different loss functions, the Margin Ranking loss function for pairwise experiments and `BCELogitLoss` for pointwise experiments. Since the high dimensionality (768) of E-BERT exceeds the memory of our available hardware, we can not include these experiments.

4.3 RQ4: Entities with Missing Embeddings

Many pretrained entity embeddings are derived from Wikipedia and DBpedia snapshots that differ slightly. As a result, some entities in the candidate set do not have available entity embeddings. This is a practical problem that will be encountered whenever pretrained embeddings are used. In particular, embeddings with many missing entities will obviously obtain lower performance in evaluation results. As it is unclear whether lower performance is due to the missing entities or quality issues of the embeddings, we analyze this in a controlled experiment.

In Table 1, we show the percentage of the candidate entities and relevant entities with unavailable embeddings under each of the three embeddings. We find that up to 7% of relevant entities do not have available embeddings. Furthermore, up to 22% of candidate entities from the baseline retrieval method, are missing in the embedding resource.

We perform an additional experiment where entities whose embeddings are unavailable, are removed from the candidate entities set (and baseline ranking) as well as the qrels. This way we avoid penalizing an embedding for missing entities. As each embedding is missing a different set of entities, we obtain different baseline rankings for each embedding. We only display results that were most affected by this experimental change in Table 4.

[4] Rank-Lips is available at https://github.com/TREMA-UNH/rank-lips.

Table 2. Overall Reproduction. The reproduced results using BM25F-CA baseline with Entity Ranking Framework of the original paper on DBpediaV2 dataset. $^{\triangle}$ indicates significant performance improvement compared to * (baseline) using paired t-test with $p < 0.05$. We show the equivalent original results from Gerritse et al. as taken from the paper in the lower half of the table. As seen in the table, we can reproduce the same results as the original paper.

DBpediaV2	INEX_LD		QALD_2		SemSearch		ListSearch		All	
Model	@10	@100	@10	@100	@10	@100	@10	@100	@10	@100
Wiki2Vec	0.217	0.286	0.212	0.282	0.417	0.478	0.211	0.302	0.262	0.335
BM25F-CA	0.439 *	0.530 *	0.369 *	0.461 *	0.628 *	0.720 *	0.425 *	0.511 *	0.461 *	0.551 *
+ Wiki2Vec	0.466	$0.552^{\triangle}$	$0.390^{\triangle}$	$0.483^{\triangle}$	$0.660^{\triangle}$	0.736	$0.452^{\triangle}$	$0.536^{\triangle}$	$0.487^{\triangle}$	$0.572^{\triangle}$
ESim_{cg}	0.217	0.286	0.212	0.282	0.417	0.478	0.211	0.302	0.262	0.335
BM25F-CA	0.439	0.530	0.369	0.461	0.628	0.720	0.425	0.511	0.461	0.551
+ ESim_{cg}	0.466	0.552	0.390	0.483	0.660	0.736	0.452	0.535	0.487	0.572

5 Results and Analysis

5.1 RQ1: Reproduction

We reproduce the results of the original paper [9] as shown in Table 2. We are able to generate the same results as given in the original paper. Wiki2Vec method represents the reranking of the candidate entities set based on the embedding score. BM25F-CA+Wiki2Vec model is the linear combination of the candidate entities set retrieved using the BM25F-CA baseline and the entity-embedding-based similarity score method i.e., Wiki2Vec. We observe the same findings: (1) Entity embeddings are beneficial to improve the performance of the entity ranking task. As shown in Table 2, combining entity embeddings with the baseline significantly improves the performance for evaluation metrics of NDCG@10 and NDCG@100, in particular for QALD_2 and ListSearch queries. (2) Entity embeddings do not perform well on their own.

5.2 RQ2: Replicability

We test whether the finding that entity embeddings are beneficial for entity ranking generalizes when re-implemented with slight technical differences as described earlier. We evaluate the performance of Wiki2Vec, ERNIE, and E-BERT through the evaluation metrics of MAP and P@R.

Table 3 shows the results. Methods Baseline+Wiki2Vec, Baseline+ERNIE, and Baseline+E-BERT represent interpolations of the baseline (first stage ranking) with embedding-based similarities. We observe that while untrained embeddings on their own are not performing well, we find several small improvements when interpolated with the baseline. For DBpediaV2, the ERNIE embeddings provide the most consistent gains. For TREC CAR, Wiki2Vec obtains the strongest improvement. Both are significant according to a paired-t-test with 5%.

Table 3. Results on TREC CAR Y2 Test and DBpediaV2 datasets. The best results are marked in bold. Significance results in text. The standard error for fine-tuned embeddings of Wiki2Vec and ERNIE is 1% for both datasets. Fine-tuning E-Bert exceeded the memory available on our GPU.

Dataset	TREC CAR		DBpedia-All		INEX_LD	QALD-2	SemSearch	ListSearch
Model	MAP	P@R	MAP	P@R	MAP	MAP	MAP	MAP
Wiki2Vec	0.084	0.129	0.360	0.382	0.325	0.301	0.428	0.397
ERNIE	0.061	0.101	0.287	0.325	0.243	0.242	0.339	0.328
E-BERT	0.075	0.107	0.346	0.371	0.307	0.289	0.416	0.381
Baseline	0.157	0.223	0.454	0.433	0.420	0.366	**0.606**	0.441
+Wiki2Vec	0.164	0.228	0.450	0.431	0.413	0.371	0.595	0.453
+ERNIE	0.161	0.227	0.459	0.436	0.426	0.371	0.601	0.454
+E-BERT	0.159	0.219	0.455	0.433	0.423	0.367	0.601	0.447
Wiki2Vec-Pair	0.472	0.440	**0.540**	**0.551**	**0.524**	**0.560**	0.521	**0.550**
Wiki2Vec-Point	0.451	0.427	0.504	0.520	0.485	0.528	0.486	0.511
ERNIE-Pair	**0.474**	**0.458**	0.491	0.519	0.454	0.519	0.465	0.512
ERNIE-Point	0.429	0.434	0.485	0.520	0.460	0.528	0.423	0.514

Table 4. Impact on evaluation results when not penalizing for entities for which embeddings are not available (missing removed). The starkest difference for ERNIE and Wiki2Vec is on the weakest method.

Dataset		ERNIE		ERNIE-Pair		Wiki2Vec		Wiki2Vec-Pair	
		MAP	P@R	MAP	P@R	MAP	P@R	MAP	P@R
TREC CAR	Original	0.061	0.101	0.474	0.458	0.084	0.129	0.472	0.440
	Missing removed	0.081	0.129	0.601	0.549	0.104	0.157	0.560	0.508
	% difference	+33%	+28%	+28%	+19%	+24%	+22%	+19%	+15%
DBpediaV2	Original	0.287	0.325	0.491	0.519	0.360	0.382	0.540	0.551
	Missing removed	0.360	0.361	0.597	0.534	0.424	0.405	0.627	0.574
	% difference	+25%	+11%	+21%	+2%	+18%	+6%	+16%	+4%

While results show significant improvements and hence support the replicability of the original findings, without fine-tuning only small gains are obtained over the baseline.

SemSearch. We observe that across all the experiments for SemSearch, the baseline (first stage ranking) performs best—in particular, it is better than or similar to all the three pretrained entity embeddings, including fine-tuned results.

We notice that for several queries in SemSearch, the relevant entities have lexical overlap with query terms, hence being easy to retrieve with keyword search, which might be one of the potential reasons for the baseline to perform the best.

For example, the query "brooklyn bridge" has a total of 14 relevant entities out of which 12 entities contain either one or both query terms. Other such examples are "harry potter" and "nokia e73".

5.3 RQ3: Fine-tuned Embeddings

We examine the performance of the fine-tuned entity embeddings with the baseline and pretrained entity embeddings to study the effect of task-specific fine-tuning. In Table 3 these are listed as Wiki2Vec-Point and ERNIE-Point for results with embeddings that are trained with point-wise loss functions, and equivalently"-Pair" for the pairwise ranking loss. E-BERT exceeded the memory available on our GPU hardware, hence we cannot provide results.

We observe that fine-tuning the existing pretrained entity embeddings significantly improves the performance for both datasets (except for SemSearch, as discussed above). We observe that fine-tuning specifically increases the performance of the TREC CAR dataset, which focuses on entities other than people, organizations, and locations.

Our findings show that the pair-wise ranking loss obtains better results in most cases than the point-wise ranking loss, thus agreeing with the common wisdom.

5.4 RQ 4: Model Performance When Correcting for Missing Entities

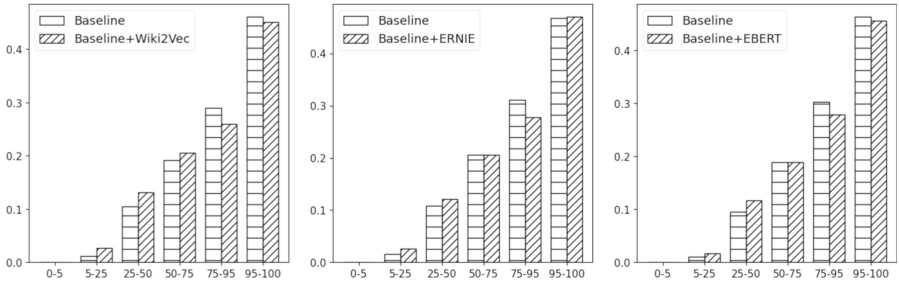
We discussed previously that missing entity embeddings of entities from the candidate set can result in lower performance for those embeddings, without providing an insight into the quality of embeddings. To observe the quality of embeddings without the missing entities, we change the experimental setup as described in Sect. 4.3.

We present results for ERNIE and Wiki2Vec in Table 4, and we obtain analogous results for the remaining experiment. We find that while the results change between the two experimental setups, the overarching story is still consistent: Embeddings by themselves are not effective, and interpolation with the baseline yields small gains.

We notice that the difference between the two experimental setups is more pronounced for the weakest and the strongest methods for both ERNIE and Wiki2Vec as shown in Table 4. Compared to Wiki2Vec, we observe a higher increase in the performance of ERNIE which is expected as ERNIE has a higher number of missing entities. This shows that the ERNIE embeddings, when available, are beneficial for the task.

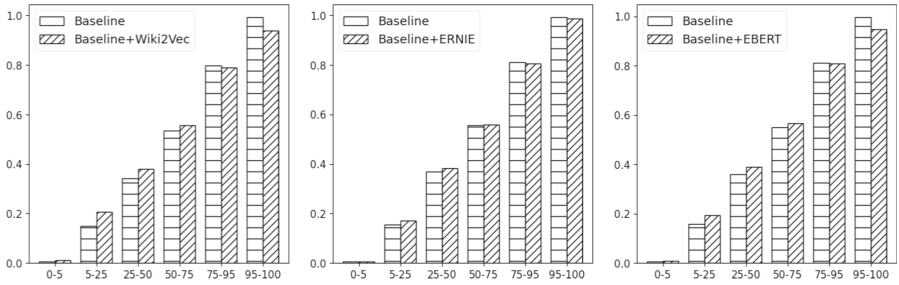
Query-level analysis. We further investigate the performance of the baseline and interpolations with pretrained embeddings at the query-level: We divide the queries into bins based on their difficulty for the baseline measured in MAP. Queries with lower MAP performance are considered to be more difficult queries.

In Fig. 1 and Fig. 2, we observe that interpolating retrieval and the embeddings yield improvements for the difficult queries of (5–75%) for both the TREC CAR dataset and DBpediaV2. This indicates that the embedding scores are a complementary source to the baseline for difficult queries, though they provide



(a) Wiki2Vec / CAR (b) ERNIE / CAR (c) E-BERT / CAR

Fig. 1. CAR-Query-level Difficulty Test for MAP Performance, corrected for missing entities. The above figure shows the difficulty test performance of MAP for the TREC CAR dataset, where y-axis is MAP performance and x-axis is the difficulty percentile according to MAP. Here, (a), (b) and (c) compare the entity rankings between the baselines and the linear combination of baselines with embedding scores. Most difficult 5% queries for the baseline are on the left side and the easiest 5% queries are on the right side.



(a) Wiki2Vec / DBpedia (b) ERNIE / DBpedia (c) E-BERT / DBpedia

Fig. 2. DBpedia-Query-level Difficulty Test for MAP Performance for all queries. Here, (a), (b) and (c) compare the entity rankings as in Fig 1.

only small gains. For the easy queries, in the 75–100 percentile the retrieval baseline often performs better than the combined methods.

Even after correcting for missing entities, we find that for pretrained entity embeddings Wiki2Vec and E-BERT obtain better performance. Closer inspection shows that they are placing relevant entities above non-relevant entities more often than ERNIE. For fine-tuned embeddings, Wiki2Vec and ERNIE are at par with each other.

6 Conclusion

In this work, we reproduce and replicate the work of Gerritse et al. [9]. Through reproducibility and replication experiments, on the two datasets of TREC CAR

and DBpediaV2, we can confirm the findings that the entity embeddings are beneficial for entity ranking. We find that consistent yet small gains are obtained with available pretrained embeddings and, confirming common wisdom, fine-tuning these pretrained embeddings achieves significantly large improvements.

One of the most interesting findings in the reproducibility paper is to use the GEEER framework to evaluate different pretrained entity embeddings. For example, the fact that matrix-factorization based Wiki2Vec embeddings are competitive to transformer-based BERT embedding models, is a sign that none of the currently available pretrained entity embedding models are particularly suitable for an IR task. We speculate that part of the problem is that ERNIE and E-Bert are over-trained on syntactic entity understanding tasks like entity linking, entity typing, and relation extraction for which the entity name fields are informative. In contrast, the entity ranking tasks of DBpediaV2 and (even more so) the TREC CAR datasets require to understand the abstract semantics of entities and their topically related entities. Wiki2Vec was pretrained on lead text, anchor text context, and the general link structure, which is likely to yield entity representations that are more amenable to entity retrieval tasks. A major takeaway from this study is that the IR community needs to train their own entity embedding models that are better suited for topical information retrieval tasks (as opposed to syntactic tasks).

Acknowledgements. This material is based upon work supported by the National Science Foundation under Grant No. 1846017. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Balog, K., Bron, M., De Rijke, M.: Query modeling for entity search based on terms, categories, and examples. ACM Trans. Inf. Syst. 29(4) (Dec 2011). https://doi.org/10.1145/2037661.2037667
2. Balog, K., Bron, M., de Rijke, M.: Category-based query modeling for entity search. In: Gurrin, C., et al. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 319–331. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12275-0_29
3. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, pp. 2787–2795. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
4. Bron, M., Balog, K., de Rijke, M.: Example based entity search in the web of data. In: Serdyukov, P., et al. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 392–403. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36973-5_33
5. Ciglan, M., Nørvåg, K., Hluchý, L.: The semsets model for ad-hoc semantic list search. In: Proceedings of the 21st International Conference on World Wide Web, pp. 131–140. WWW 2012, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2187836.2187855
6. Dalton, J., Dietz, L., Allan, J.: Entity query feature expansion using knowledge base links. In: Proceedings of the 37th International ACM SIGIR Conference on

Research and Development in Information Retrieval, pp. 365–374. SIGIR 2014, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2600428.2609628
7. Dietz, L.: Ent rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 215–224. SIGIR 2019, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331257
8. Dietz, L., Gamari, B.: Trec complex answer retrieval. In: Proceedings of Text REtrieval Conference (TREC) (2018)
9. Gerritse, E.J., Hasibi, F., de Vries, A.P.: Graph-embedding empowered entity retrieval. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 97–110. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_7
10. Kaptein, R., Kamps, J.: Exploiting the category structure of wikipedia for entity ranking. Artif. Intell. **194**, 111–129 (2013). https://doi.org/10.1016/j.artint.2012.06.003
11. Lin, T., Pantel, P., Gamon, M., Kannan, A., Fuxman, A.: Active objects: actions for entity-centric search. In: Proceedings of the 21st International Conference on World Wide Web, pp. 589–598. WWW '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2187836.2187916
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI (2015)
13. Liu, T.Y., et al.: Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval 3(3), 225–331 (2009)
14. Liu, Z., Xiong, C., Sun, M., Liu, Z.: Explore entity embedding effectiveness in entity retrieval. In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 105–116. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_9
15. Lu, Y., Lu, H., Fu, G., Liu, Q.: KELM: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. arXiv preprint arXiv:2109.04223 (2021)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Oza, P., Dietz, L.: Which entities are relevant for the story? In: CEUR workshop proceedings, vol. 2860 (2021)
18. Pehcevski, J., Thom, J.A., Vercoustre, A.M., Naumovski, V.: Entity ranking in wikipedia: Utilising categories, links and topic difficulty prediction. Inf. Retr. **13**(5), 568–600 (2010). https://doi.org/10.1007/s10791-009-9125-9
19. Poerner, N., Waltinger, U., Schütze, H.: E-BERT: efficient-yet-effective entity embeddings for BERT. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 803–818. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.71
20. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 771–780. WWW 2010, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1772690.1772769
21. Raviv, H., Carmel, D., Kurland, O.: A ranking framework for entity oriented search using markov random fields. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search, pp. 1–6 (2012)

22. Schuhmacher, M., Dietz, L., Paolo Ponzetto, S.: Ranking entities for web queries through text and knowledge. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1461–1470. CIKM 2015, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2806416.2806480
23. Tonon, A., Demartini, G., Cudré-Mauroux, P.: Combining inverted indices and structured search for ad-hoc object retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 125–134. SIGIR '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2348283.2348304
24. Wang, X., et al.: Kepler: a unified model for knowledge embedding and pre-trained language representation. Trans. Assoc. Comput. Linguist. **9**, 176–194 (2021)
25. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence 28(1) (Jun 2014). https://doi.org/10.1609/aaai.v28i1.8870, https://ojs.aaai.org/index.php/AAAI/article/view/8870
26. Yamada, I., et al.: Wikipedia2Vec: an efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 23–30. Association for Computational Linguistics (2020)
27. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1441–1451. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1139, https://aclanthology.org/P19-1139
28. Zhiltsov, N., Kotov, A., Nikolaev, F.: Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 253–262 (2015)