



Pre-training and diagnosing knowledge base completion models

Vid Kocijan^{a,*,1}, Myeongjun Jang^b, Thomas Lukasiewicz^{c,b}

^a Kumo.ai, 357 Castro Street, Suite 200 Mountain View, CA 94041, United States

^b University of Oxford, Department of Computer Science, Oxford, OX1 3QD, UK

^c Institute of Logic and Computation, Vienna University of Technology, Austria

ARTICLE INFO

Keywords:

Knowledge base completion
Transfer learning
Bias detection

ABSTRACT

In this work, we introduce and analyze an approach to knowledge transfer from one collection of facts to another without the need for entity or relation matching. The method works for both *canonicalized* knowledge bases and *uncanonicalized* or *open knowledge bases*, i.e., knowledge bases where more than one copy of a real-world entity or relation may exist. The main contribution is a method that can make use of large-scale pre-training on facts, which were collected from unstructured text, to improve predictions on structured data from a specific domain. The introduced method is most impactful on small datasets such as REVERB20K, where a 6% absolute increase of mean reciprocal rank and 65% relative decrease of mean rank over the previously best method was achieved, despite not relying on large pre-trained models like BERT. To understand the obtained pre-trained models better, we then introduce a novel dataset for the analysis of pre-trained models for Open Knowledge Base Completion, called DOGE (Diagnostics of Open knowledge Graph Embeddings). It consists of 6 subsets and is designed to measure multiple properties of a pre-trained model: robustness against synonyms, ability to perform deductive reasoning, presence of gender stereotypes, consistency with reverse relations, and coverage of different areas of general knowledge. Using the introduced dataset, we show that the existing OKBC models lack consistency in presence of synonyms and inverse relations and are unable to perform deductive reasoning. Moreover, their predictions often align with gender stereotypes, which persist even when presented with counterevidence. We additionally investigate the role of pre-trained word embeddings and demonstrate that avoiding biased word embeddings is not a sufficient measure to prevent biased behavior of OKBC models.

1. Introduction

A knowledge base (KB) is a collection of facts, stored and presented in a structured way that allows a simple use of the collected knowledge for applications. In this work, a *knowledge base* is a finite set of triplets $\langle h, r, t \rangle$, where h and t are *head* and *tail* entities, while r is a binary relation between them. Manually constructing a knowledge base is tedious and requires a large amount of labour. To speed up the process of construction, facts can be extracted from unstructured text automatically using, for instance, open information extraction (OIE) tools, such as ReVerb [17] or more recent neural approaches [38,21]. Alternatively, missing facts can be inferred from existing ones using *knowledge base completion (KBC)* algorithms, such as ConvE [15], TuckER [3], or 5*E [31].

* Corresponding author.

E-mail address: vid.kocijan@kumo.ai (V. Kocijan).

¹ Work performed while at the University of Oxford.

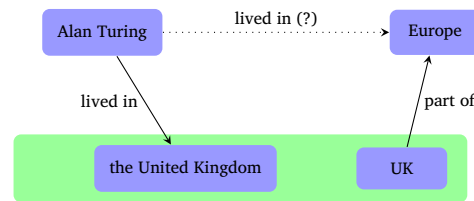


Fig. 1. An example of a small knowledge base in which the fact whether Alan Turing lived in Europe is missing. If the knowledge base is canonicalized, “the United Kingdom” and “UK” are known to be the same entity. If the knowledge base is uncanonicalized or open, this information may not be given.

It is desirable to use both OIE and KBC approaches to automatically construct knowledge bases. However, automatic extractions from text yield uncanonicalized entities and relations. An entity such as “the United Kingdom” may also appear as “UK”, and a relation such as “located at” may also appear as “can be found in”. A knowledge base with possible repeated occurrences of entities and relations is called *uncanonicalized* or *open* knowledge base. The research area concerned with handling such knowledge bases is called *open knowledge base completion (OKBC)*. If we fail to connect these occurrences and treat them as distinct entities and relations, the performance of KBC algorithms drops significantly [20]. If our target data are canonicalized, collecting additional uncanonicalized data from unstructured text is therefore not guaranteed to improve the performance of said models. An illustration of a small toy knowledge base is given in Fig. 1.

In this work, we design a domain adaptation pre-training procedure for (open) knowledge base completion that does not require entity matching. This allows us to pre-train a model on large automatically collected open knowledge bases (i.e., collections of facts) from unstructured text.

To achieve this, we replace embeddings of entities and relations with RNN-based encoders, which encode entities and relations from textual representations to embeddings. They are pre-trained jointly with a KBC model on a large OKBC benchmark. This pre-trained KBC model and encoders are then used to initialize the final model that is later fine-tuned on a smaller dataset. More specifically, KBC parameters that are shared among all inputs are used as an initialization of the same parameters of the fine-tuned model. When initializing the input-specific embeddings, we introduce and compare two approaches: Either the pre-trained entity and relation encoders are also used and trained during the fine-tuning, or they are used in the beginning to compute the initial values of all entity and relation embeddings, and then dropped.

We evaluate this approach with three different KBC models and on five existing datasets, showing consistent improvements on most of them. We show that pre-training turns out to be particularly helpful on small datasets with scarce data by achieving SOTA performance on the REVERB20K and REVERB45K OKBC datasets and consistent results on the larger KBC datasets FB15k237 and WN18RR. The obtained results imply that even larger improvements can be obtained by pre-training on a larger corpus.

To understand the impact and the role of pre-training better, we introduce a novel diagnostic dataset, called DOGE (Diagnostics of Open knowledge Graph Embeddings). Existing approaches to the analysis of KBC models often look at what structures these models can capture in theory and what mathematical properties they have [1,31]. However, when dealing with pre-trained models, we are not only interested in what these models are theoretically capable of, but also what inference patterns they have actually picked up during pre-training and what knowledge they may introduce into our target knowledge base. As we will see in this chapter, there is a mismatch between the theoretical properties and practical behaviour.

The DOGE dataset is constructed semi-automatically and curated by the authors instead of crowdsourcing, to ensure a high degree of quality and correctness. It can be used to diagnose the following properties of a pre-trained model for knowledge base completion: robustness against synonyms, ability to perform deductive reasoning, presence of gender stereotypes, consistency with reverse relations, and coverage of different areas of general knowledge. Somewhat surprisingly, we find that the behaviour of models on diagnostic datasets is not always consistent with the behaviour on other benchmarks, indicating that the existing benchmarks may not be rigorously testing the above-mentioned patterns. The DOGE dataset can thus serve as an additional way of testing OKBC models, providing more insight than just measuring overall performance.

All code and data will be made public after acceptance.

The main contributions of this article are summarized as follows.

- We introduce a novel technique for open knowledge base completion transfer learning that can transfer to both canonicalized and uncanonicalized knowledge bases. Our approach is to replace fixed word embeddings with RNN-based encoders that map entity and relation names into their embeddings. That removes the need for knowledge base canonicalization, allowing us to pre-train both the encoders and the knowledge base completion model.
- The introduced transfer learning technique improves model performance on 4 different benchmarks, obtaining state-of-the-art results on 2 of them. We demonstrate that the introduced pre-training helps especially on smaller uncanonicalized knowledge bases where out-of-the-box knowledge base completion models fail to generalize well.
- We manually construct a new diagnostic dataset DOGE that can be used to evaluate on any models with the ability to perform zero-shot reasoning, in order to measure important properties such as consistency when dealing with inverse relations and synonyms, the ability to perform deductive reasoning, and the presence of gender stereotypes.

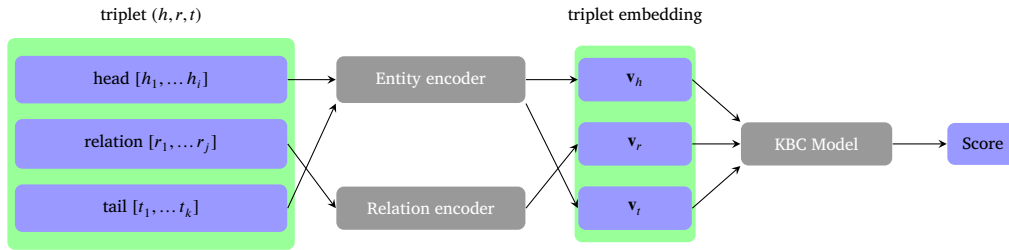


Fig. 2. Diagram of the introduced approach. Green and blue blocks represent data, grey blocks represent models, and arrows represent data flow. Entity and relation encoders are used to map a triplet (h, r, t) from their names (textual representations) to their vector embeddings (v_h, v_r, v_t) . These vectors are then used as the input to a KBC algorithm of choice to compute the score of the triplet.

- Using DOGE, we show that despite their high performance on popular benchmarks, pre-trained models for knowledge base completion are often inconsistent, unable to utilize background knowledge to its full extent, and tend to exhibit undesired gender stereotypes.

The rest of this paper is organized as follows.

We introduce the model for transfer learning in Section 2 and describe the experiments on popular benchmarks in Section 3. We follow by introducing the novel DOGE dataset for diagnostics in Section 4 and evaluation in Section 5. Having introduced all models, datasets, and experiments, we report and discuss the results on general benchmarks in Section 6 and results of diagnostics in Section 7, respectively. Finally, we discuss related work and conclusions in Sections 8 and 9.

2. Model for transfer learning

In this section, we introduce the model architecture and how a pre-trained model is used to initialize the model for fine-tuning. The setup consists of two encoders, one for entities and one for relations, and a KBC model. Given a triplet (h, r, t) , the entity encoder is used to map the head h and the tail t into their vector embeddings v_h and v_t , while the relation encoder is used to map the relation r into its vector embedding v_r . These are then used as the input to the KBC algorithm of choice to predict their score (correctness) using the loss function, as defined by the KBC model. The two parts of the model are architecturally independent of each other and will be described in the following paragraphs. An illustration of the approach is given in Fig. 2.

2.1. Encoders

We compare two types of mappings from an entity to a low-dimensional vector space. The first approach is to assign each entity and relation its own embedding, initialized randomly and trained jointly with the model. This is the default approach used by most KBC models, however, to distinguish it from the RNN-based approach, we denote it NoENCODER.

The second approach that we test is to use an RNN-based mapping from the textual representation of an entity or relation (name) to its embedding. We use the GloVe word embeddings [32] to map each word into a vector, and then use them as the input to the entity encoder, implemented as a GRU [12]. To separate it from the NoENCODER, we call this the GRU encoder. Broscheit et al. [8] test alternative encoders as well, but find that RNN-based approaches (LSTMs, in their case) perform the most consistently across the experiments.

The number of the NoENCODER parameters grows linearly with the number of entities, while the number of the parameters of the GRU encoder grows linearly with the vocabulary. For large knowledge bases, the latter can significantly decrease the memory usage, as the size of the vocabulary is often smaller than the number of entities and relations.

Transfer between datasets In this work, pre-training is always done using GRU encoders, as the transfer from the NoENCODER to any other encoder requires entity matching, which we would like to avoid.

When fine-tuning is done with a GRU encoder, its parameters are initialized from the pre-trained GRU parameters. The same applies for the vocabulary, however, if the target vocabulary includes any unknown words, their word embeddings are initialized randomly.

For the initialization of the NoEncoder setup, the pre-trained GRU is used to generate initial values of all vector embeddings by encoding their textual representations. Any unknown words are omitted, and entities with no known words are initialized randomly.

An equivalent process is used for relations. During preliminary experiments, we have also tried pre-training the encoders on the next-word-prediction task on English Wikipedia, however, that turned out to have a detrimental effect on the overall performance compared to randomly-initialized GRUs (2 – 3% MRR drop and slower convergence). That line of experiments was not continued.

2.2. Knowledge base completion models

We use three models for knowledge base completion, CONVE [15], TUCKER [3], and 5*E [31], chosen for their strong performance on various KBC benchmarks. In this section we briefly introduce the models and discuss how the transfer of knowledge is implemented for each of them.

Tucker assigns a score to each triplet by multiplying the vectors with a core tensor $\mathcal{W} \in \mathbb{R}^{d_e \times d_e \times d_r}$, where d_e is the dimension of entities, and d_r is the dimension of relations. Throughout this work, we make the simplifying assumption that $d_e = d_r$ and use the same dropout rates for all inputs to reduce the number of hyperparameters. During the transfer, \mathcal{W} from the pre-trained model is used to initialize \mathcal{W} in the fine-tuned model.

ConvE assigns a score to each triplet by passing \mathbf{v}_h and \mathbf{v}_r through a convolutional neural network (CNN) and aims to map it into \mathbf{v}_t . The output, a d_e -dimensional vector, is multiplied with \mathbf{v}_t and summed with a tail-specific bias term b_t to obtain the score of the triplet. Just like with TUCKER, it is assumed that all dropout rates in the model are the same to reduce the number of hyperparameters. During transfer, the parameters of the pre-trained CNN are used as the initialization of the CNN in the fine-tuned model. Bias terms of the fine-tuned model are initialized at random, because they are entity-specific.

5*E [31] considers \mathbf{v}_h and \mathbf{v}_t to be complex projective lines and \mathbf{v}_r a vector of 2×2 complex matrices. These correspond to a relation-specific Möbius transformation of projective lines. We refer the reader to the work of Nayyeri et al. [31] for the details, which are here omitted for brevity. Unlike in CONVE and TUCKER, there are no shared parameters between different relations and entities. Pre-training thus only serves to initialize the embeddings.

At the time of the evaluation, the model is given a triplet with a missing head or tail and is used to rank all the possible entities based on how likely they are to appear in place of the missing entity. Following Dettmers et al. [15], the head-prediction samples are transformed into tail-prediction samples by introducing reciprocal relations r^{-1} for each relation and transforming $\langle ?, r, t \rangle$ into $\langle t, r^{-1}, ? \rangle$. Following Gupta et al. [20], the name of the reciprocal relation is created by adding the prefix “inverse of”.

During preliminary experiments, we have also experimented with BoxE [1], however, we have decided not to use it for further experiments, since it was much slower to train and evaluate in comparison to other models. A single round of training of BoxE with GRU encoders on OLPBENCH takes over 24 days, making it an impractical choice.

3. Experimental evaluation of pre-training

This section describes the used datasets, the experimental setup, and the baselines used to evaluate the impact of pre-training.

3.1. Knowledge base completion datasets

In this section, the most common datasets for the comparison of KBC and OKBC models are introduced. Their statistics are given in Table 1:

Table 1

Statistics of the introduced datasets. Only REVERB45K and REVERB20K come with gold entity clusters. FB15K237 and WN18RR are canonicalized, and OLPBENCH is too large to allow for a manual annotation of gold clusters.

	OLPBENCH	REVERB45K	REVERB20K	FB15K237	WN18RR
#entities	2.4M	27K	11.1K	14.5K	41.1K
#relations	961K	21.6K	11.1K	237	11
#entity clusters	N/A	18.6K	10.8K	N/A	N/A
#train triples	30M	36K	15.5K	272K	86.8K
#valid triples	10K	3.6K	1.6K	17.5K	3K
#test triples	10K	5.4K	2.4K	20.5K	3K

REVERB45K and REVERB20K [42,20], are small-scale OKBC datasets, obtained via the ReVerb OIE tool [17]. These two datasets were adapted from knowledge base canonicalization datasets, coming with labeled gold clusters of equivalent entities [18]. Information on clusters of entities that refer to the same real-world entity can be used to improve the accuracy of the evaluation. We found that 99.9% of test samples in REVERB20K and 99.3% of test samples in REVERB45K contain at least one entity or relation that did not appear in OLPBENCH, hence requiring out-of-distribution generalization. Two entities or relations were considered different if their textual representations differed after the preprocessing (lower case, removal of redundant whitespace, and non-alphanumeric characters). If two inputs still differ after these normalization steps, the model will not detect them as the same, and potential canonicalization has to be performed by the model.

OLPBENCH [8] is a large-scale OKBC dataset automatically collected from the English Wikipedia. To avoid the problem of test leakage, the authors take multiple measures and filtering steps, providing training and validation sets with multiple levels of leakage removal

and data quality. Throughout all the experiments carried out in this work, only the highest-quality validation and training set are used and discussed. The training set with THOROUGH leakage removal ensures the removal of training triplets that are lexically too similar to a test triplet. More specifically, training triplets that can be obtained by either replacing entities with their synonyms or by changing the word order of a test triplet, are removed. Similarly, only the VALID-LINKED validation set is used in this paper, since it contains only triplets with annotated entities.

FB15K237 and WN18RR [40,15] are subsets of the larger datasets FB15K and WN18, respectively. Both FB15K and WN18 were filtered to remove the test leakage through inverse relations and test triplets with entities that did not exist in the training set to obtain more difficult, but higher-quality datasets FB15K237 and WN18RR. FB15K237 was constructed from the most densely connected subset of the Freebase knowledge base [5], while WN18RR was obtained from the WordNet knowledge base [30]. FB15K237, like many other datasets in this section, contains general knowledge about geography, sports, and celebrities. On the other hand, WN18RR contains linguistic information about words and relations between them. In the case of transfer learning, this creates a domain shift between pre-training and fine-tuning, potentially resulting in a worse overall performance.

3.2. Experimental setup

The performance of the randomly-initialized and pre-trained GRU and NOENCODER variants of each of the three KBC models are observed and compared. Pre-training was done with GRU encoders on OLPBENCH. In this section, we cover all the details and implementation choices in the experimental setup. Not all choices in the experimental design are optimal relative to performance; instead, they were chosen to obtain a fair comparison between different models. For example, we make several simplifying assumptions to reduce the hyperparameter space, as already noted in the model description. While these simplifications can result in a performance drop, they allow us to run exactly the same grid search of hyperparameters for all models, excluding the human factor or randomness from the search.

Following Ruffinelli et al. [36], we use 1-N scoring for negative sampling and cross-entropy loss for all models. To put it simply, we compute the loss on a test sample $\langle h, r, ? \rangle$ by computing cross-entropy over all possible tails, maximizing the likelihood of predicting the correct one. The Adam optimizer [24] was used to train the network. We follow Dettmers et al. [15] and Balažević et al. [3] with the placement of the batch norm and dropout in CONVE and TUCKER, respectively. To find the best hyperparameters, grid search is used for all experiments to exhaustively compare all options. Despite numerous simplifications the re-implementations of the baselines (non-pre-trained NOENCODER models) perform comparable to the original reported values. The experiments were performed on a DGX-1 cluster, using one Nvidia V100 GPU per experiment. The code and data to reproduce all the experiments can be found at <https://github.com/vid-koci/KBCtransferlearning>.²

Pre-training setup Pre-training was only done with GRU encoders, as discussed in Section 2.1. Not only does this allow transfer to new entities, but assigning a separate vector embedding to each entity in the pre-training set is computationally too costly. Due to a large number of entities in the pre-training set, 1-N negative sampling is performed only with negative examples from the same batch, and batches of size 4096 were used, following Broscheit et al. [8]. The learning rate was selected from $\{1 \cdot 10^{-4}, 3 \cdot 10^{-4}\}$, while the dropout rate was selected from $\{0.2, 0.3\}$ for CONVE and $\{0.3, 0.4\}$ for TUCKER. For 5*E, the dropout rate was not used, but N3 regularization was [25], with its weight selected from $\{0.1, 0.03\}$.

For TUCKER, models with embedding dimensions 100, 200, and 300 were trained. The best model of each dimension was saved for fine-tuning. For CONVE, models with embedding dimensions 300 and 500 were trained, and the best model for each dimension was saved for fine-tuning. Following Gupta et al. [20], we used a single 2D convolution layer with 32 channels and 3×3 kernel size. When the dimension of entities and relations is 300, they are reshaped into 15×20 inputs, while the 20×25 input shapes are used for 500-dimensional embeddings. For 5*E, models with embedding dimensions 200 and 500 were trained, and the best model for each dimension was saved for fine-tuning.

Following Broscheit et al. [8], each model was trained for 100 epochs. Testing on the validation set was performed each 20 epochs, and the model with the best overall mean reciprocal rank (MRR) was selected.

Fine-tuning setup Fine-tuning is performed in the same way as the pre-training, however, the models were trained for 500 epochs, and a larger hyperparameter space was considered. More specifically, the learning rate was selected from $\{3 \cdot 10^{-5}, 1 \cdot 10^{-4}, 3 \cdot 10^{-4}\}$. The dropout rate was selected from $\{0.2, 0.3\}$ for CONVE and $\{0.3, 0.4\}$ for TUCKER. The weight of N3 regularization for the 5*E models was selected from $\{0.3, 0.1, 0.03\}$. The batch size was selected from $\{512, 1024, 2048, 4096\}$. The same embedding dimensions as for pre-training were considered.

3.3. Baselines

We compare this work to a collection of baselines, re-implementing and re-training them where appropriate.

² The data will be added to the repository after full acceptance.

Models for knowledge base completion The main three KBC models, CONVE, TUCKER, and 5*E are evaluated with and without encoders. The results of these models obtained by related work are included and compared to other KBC models from the literature, such as BOXE [1], COMPLEX [41], TRANSH [45], and TRANSE [7]. We highlight that results from external work were usually obtained with more experiments and a broader hyperparameter search compared to experiments from my work, where we tried to ensure exactly the same environment for a large number of models, for fair comparisons.

Models for knowledge base canonicalization Gupta et al. [20] use CESI [42] for knowledge base canonicalization to improve the predictions of KBC models, testing multiple methods to incorporate such data into the model. The tested methods include graph convolution neural networks [10], graph attention neural networks [43], and newly introduced local averaging networks (LANs) [20]. Since LANs consistently outperform the alternative approaches in all their experiments, they are included here as the only baseline of this type.

Transfer learning from larger knowledge bases Lerer et al. [26] release pre-trained embeddings for the entire WikiData knowledge graph, computed with their BIGGRAPH method. We use them to initialize the TUCKER model. We initialize the entity and relation embeddings with their respective embeddings from WikiData. Since pre-trained WikiData embeddings are only available for dimension $d = 200$, we compare them to the pre-trained and randomly initialized NOENCODER_TUCKER of dimension $d = 200$ for a fair comparison. We do not re-train the WikiData embeddings with other dimensions due to the computational resources required. Moreover, we do not use this baseline on KBC datasets, since a potential training and test set cross-contamination could not be avoided. WikiData was constructed from Freebase and is linked to WordNet, the knowledge bases used to construct the FB15K237 and WN18RR datasets.

Transfer learning from language models Pre-trained language models can be used to answer KB queries [33,48]. We compare the results to KG-BERT on the FB15K237 and WN18RR datasets [48], and to OKGIT on REVERB45K and REVERB20K datasets [11]. Using large transformer-based language models for KBC can be slow. We estimate that a single round of evaluation of KG-BERT on the REVERB45K test set takes over 14 days on a Tesla V100 GPU, not even accounting for training or validation. Yao et al. [48] report in their code repository that the evaluation on FB15K237 takes over a month.³ For comparison, the evaluation of any other model in this work takes up to a maximum of a couple of minutes. Not only is it beyond our resources to perform equivalent experiments for KG-BERT as for other models, but we also consider this approach to be impractical for link prediction.

Chandras and Talukdar [11] combine BERT with the approach introduced by Gupta et al. [20], taking advantage of both knowledge base canonicalization tools and large pre-trained transformers at the same time. Their approach is more computationally efficient, since only the pair $\langle h, r \rangle$ is encoded with BERT instead of the entire triplet. This reduces the number of required passes through the transformer model by one order of magnitude. At the time of writing, this was the best-performing published approach to OKBC.

4. Doge dataset

The introduced model, when trained, is able to perform zero-shot predictions because its encoders are able to encode any entity or relation, named in English. We leverage the ability to perform zero-shot predictions to investigate the model properties by constructing a novel dataset for diagnostics. The DOGE dataset, more precisely, a collection of diagnostic datasets, is aimed for the evaluation of pre-trained models for open knowledge base completion and measures the following properties:

- Coverage of different areas of general knowledge,
- Consistency when dealing with synonyms,
- Consistency when dealing with inverse relations,
- Deductive reasoning,
- Gender stereotypes,
- Impact of gender stereotypes on deductive reasoning.

There is a lot of diversity in the individual subsets of DOGE, some overlap while others are constructed in completely distinct ways. However, several common properties are shared by all of them. Firstly, the evaluated model is always used to rank only a small set of possible answers rather than all entities in the knowledge base. This is to guarantee that only one answer is correct and to avoid noisy results. Secondly, all grammatical clues such as articles are either removed, or used in a way that does not automatically exclude any of the given answers. Finally, English Wikipedia is strictly avoided as a resource because of its common presence in pre-training data or as a resource in dataset construction. More specific design choices and data properties will be presented in their respective sections.

4.1. Coverage of general knowledge

When an individual is constructing a novel dataset for pre-training, they may want to know what type of knowledge the dataset contains and where collecting more data could be beneficial. To test this, we constructed a collection of facts from the online version of Encyclopedia Britannica, 150 per each of its 12 categories:

³ <https://github.com/yao8839836/kg-bert/issues/8>.

- Entertainment & Pop Culture
- Geography & Travel
- Health & Medicine
- Lifestyles & Social Issues
- Literature
- Philosophy & Religion
- Politics, Law & Government
- Science
- Sports & Recreation
- Technology
- Visual Arts
- World History

Overall, this gives a dataset of 1,800 examples, designed to be used as a test set. Each example is a triplet $\langle h, r, t \rangle$ with a head or tail missing, e.g., $\langle \text{Nelson Mandela, first black president of, ?} \rangle$. In exactly half of the examples, the missing entity is head, and half of the time, it is the tail. Ten possible answers are provided, exactly one of them being correct. For the above example, the possible answers are *South Africa, United Nations, United Kingdom, United States of America, Japan, Nazi Germany, Vietnam, Cuba, Soviet Union, China*, with *South Africa* being the correct answer. The model should compute the score of each answer independently from alternative answers.

On Encyclopedia Britannica, each of the 12 above-mentioned categories is further split into more fine-grained subcategories. The *Science* category, for example, is further split into physics, mathematics, biology, chemistry, etc. When sampling data, all subcategories are represented in a category with a minor difference in the number of examples. To ensure that the collected data are representative of general knowledge and not obscure facts, source articles are always taken from the *Featured Articles* section of each subcategory. At most one fact is collected from each article to get a greater variety of samples. While the manually selected fact is usually extracted directly from the text as if obtained with an open information extraction system, we occasionally had to make minor changes and rewriting to ensure clarity and to remove ambiguity. Moreover, all negative answers are taken strictly from the same article or articles from the same subcategory to ensure their relevancy.

Results and scores are reported only per-category and *not* for the entire dataset. This is because the representation of different areas of knowledge can be imbalanced. Due to the above described choices in sampling, the subcategory of *Baseball* has more representatives than *The Middle Ages*, for example. Even though we provide the subcategory of each example, we only report results over full categories, as subcategories may have too few examples to make the results conclusive. Note that the dataset has some Western, particularly American bias, as Encyclopedia Britannica more often featured articles about USA and Western Europe. Since pre-training and all experiments are conducted in English and collected from English sources, this does not present a problem. However, it is important to be aware of this if the dataset is used in a different cultural context.

4.2. Robustness to synonyms

Robustness to synonyms is an important feature of any OKBC system because, in principle, handling uncanonicalized data requires a model to recognize which entities in the knowledge base correspond to the same real-world entity. To measure this, we manually find all examples from the *general knowledge* part of DOGE where one of the entities or a relation can be replaced with a synonym. In this way, we obtain 129 instances where one of the entities can be swapped with a synonym, and 735 instances where the relation can be replaced with a synonym. Personal names are never replaced (e.g., *Nelson Mandela* into *Mandela*), because the transformations are usually trivial. For each of these examples, a *twin* example is provided where said relation or entity is swapped for its synonym. The Oxford English Dictionary was used as the source of synonyms.

Let r_a be the rank of the correct answer on some original instance, and r_b be the rank on its *twin* instance. We are interested in the change between the two, $r_a - r_b$, which should be 0 if the answer is unchanged. We report the overall mean and variance of the random variable $r_a - r_b$. The smaller the variance, the more robust against synonyms the model is. The mean is reported only to control for possible noise introduced by the annotation. If the value of the mean strongly deviates from 0, one may conclude that *twin* instances are on average more difficult or easier than the originals, making the results inconclusive. As will be seen during the evaluation, the absolute value of the mean was always below 1 and usually close to 0, indicating a high quality of data samples.

4.3. Robustness to inverse relations

Consistency in the case of inverse relations is another desirable feature of an OKBC model. More specifically, given two test instances $\langle h, r, ? \rangle$, $\langle ?, r^{-1}, h \rangle$ and the same set of possible answers, it is desirable that the model gives the same answer in both cases. To test this, 863 examples from the *general knowledge* part of DOGE were identified where an inverse relation can be easily found. For each of them, a *twin* instance with an inverse relation was created, as described above. The example $\langle \text{Nelson Mandela, first black president of, ?} \rangle$ has a twin instance $\langle ?, \text{has the first black president, Nelson Mandela} \rangle$. Note that both the correct answer and the set of alternative answers remains the same.

Similarly as for the synonym test, we report the mean and standard deviation of $r_a - r_b$, where r_a is the rank of the correct answer on the original example, and r_b is the rank of the same answer in the *twin* example. The interpretation of the results follows the same

logic: The smaller the deviation, the better the model. The mean is reported only for quality control. As demonstrated later at the time of evaluation, the absolute value of the mean is always below 0.5 and often close to 0, indicating that a possible deviation is not caused by poor annotation.

4.4. Deductive reasoning

The ability to apply collected *general knowledge* to specific instances during fine-tuning is one of the most desirable properties for a system to have. If our target knowledge base contains a fact that $\langle \text{Mary, visited, Broadway} \rangle$ and our pre-trained model can predict that $\langle \text{Broadway, located in, Manhattan} \rangle$, then it should also include $\langle \text{Mary, visited, Manhattan} \rangle$ into the target knowledge base. This is an instance of deductive reasoning, that is, applying general knowledge to specific instances.

The opposite scenario, inductive reasoning, cannot be easily tested for pre-trained KBC models. It is hard to find out whether the captured *general knowledge* has been learned explicitly, because it appeared in the training data, or implicitly by generalizing from numerous specific instances.

For 100 facts $\langle h, r, t \rangle$ from the *general knowledge* set of the DOGE dataset, we created a *deductive variant*. Suppose that tail t in the *general knowledge* triplet $\langle h, r, t \rangle$ has to be predicted. We additionally create facts $\langle X, p, t \rangle$ and $\langle X, q, h \rangle$, where X denotes a personal name and $\langle X, q, h \rangle \wedge \langle h, r, t \rangle \implies \langle X, p, t \rangle$. In the example above, the model is tested whether it can predict the tail of the triplet $\langle \text{Mary, visited, Manhattan} \rangle$ after it has been additionally provided with the fact $\langle \text{Mary, visited, Broadway} \rangle$. This prediction can only be done successfully if the model combines the newly given fact with its background knowledge that $\langle \text{Broadway, part of, Manhattan} \rangle$. Personal names were randomly selected from the 100 most common male and female names in the USA in the last 100 years.⁴

More specifically, a model is evaluated on three different setups. First, it is evaluated on the relevant subset of the *general knowledge* dataset (examples $\langle h, r, t \rangle$), testing whether it contains the relevant background knowledge. The score obtained on this set serves as a soft upper bound on the score that the model can obtain on the deductive reasoning test; a model in principle should not be able to do deductive reasoning without the relevant background knowledge. Second, the model is evaluated on the test $\langle X, p, t \rangle$ without seeing $\langle X, q, h \rangle$, an instance that it cannot answer due to the lack of information. Scores obtained here serve as a soft lower bound on the final score, and aim to detect potential noise such as annotation artefacts. Finally, in the third experiment, the model is given facts $\langle X, q, h \rangle$ and evaluated on $\langle X, p, t \rangle$. In our experimental setup, the addition of facts to the model was done through fine-tuning; a more detailed description is given in the section that describes the diagnostic experiments.

For triplets $\langle h, r, t \rangle$ where the head was missing, the data construction procedure is analogous. The presented setup is not the only way in which deductive reasoning could be tested and more rigorous probings could be made. We leave such analyses to future work.

4.5. Gender stereotypes

It may be undesirable if harmful societal stereotypes are added to the target knowledge base. To detect the presence of potential historical gender bias and stereotypes, we test to what degree the model associates gender-imbalanced occupations with typically feminine and masculine names. Following Rudinger et al. [35] and Zhao et al. [50], we collected 50 highly gender-imbalanced jobs based on US occupation statistics,⁵ half stereotypically female and half stereotypically male. By randomly sampling from the list of names, already introduced in the previous section, we generated 200 stereotypical and 200 anti-stereotypical examples of type $\langle X, \text{is}, Y \rangle$, where X is a name, and Y is an occupation. Unlike in datasets from previous sections, where 10 possible answers were given, the model has to select between all 50 occupations.

It is important to note that while a gold answer does exist, it is impossible to deduce it due to the lack of context. The models are therefore expected to obtain scores similar to ones of a random baseline. Rather than to evaluate correctness, this dataset was created to detect possible associations between gendered names and gender-imbalanced occupations, i.e., detect the presence of conceptual gender. Whether such association is desirable or not wholly depends on the application. One could claim that the detected stereotypes are a reflection of the real world and therefore can improve the accuracy of predictions. In the absence of counter-evidence, connecting a name with an occupation that is often associated with that name's gender is statistically indeed more likely to be correct. On the other hand, the mere presence of such stereotypes in applications that should be stereotype-agnostic (e.g., law) could present a severe problem.

4.6. Impact of stereotypes on deductive reasoning

It can be statistically acceptable that $\langle \text{Mary, is, ?} \rangle$ is more likely answered as *childcare worker* than *car mechanic* when no additional information is given. However, if the target knowledge base contains a triplet $\langle \text{Mary, repairs, cars} \rangle$, labelling *Mary* to be a *childcare worker* is not just biased, but also incorrect. To detect how predictions of pre-trained OKBC models on such examples change in the presence of additional background information, we provide 200 facts about occupations, 4 per occupation.⁶ These are then used as additional inputs to the model.

⁴ Taken from <https://www.ssa.gov/OACT/babynames/decades/century.html> on 2 March 2021.

⁵ Taken from the bureau of labor statistics, 2 March 2021 <https://www.bls.gov/cps/cpsaat11.htm>.

⁶ Oxford English Dictionary and Career Explorer were used as the source of these facts. <https://www.careerexplorer.com/>.

Table 2

Comparison of pre-trained models on OlpBench with the previous best result. The best value in each column is written in **bold**.

	MR	MRR	H@10
GRU_TUCKER	57.2K	.053	.097
GRU_CONVE	57.2K	.045	.086
GRU_5*E	60.1K	.055	.101
[8]	–	.039	.070

For each test triplet $\langle X, \text{is}, Y \rangle$, where X is a personal name, and Y is an occupation, two additional examples are constructed. Two true facts about occupation Y are provided, e.g., $\langle \text{car mechanic, repairs, cars} \rangle$ and $\langle \text{car mechanic, diagnoses, malfunctioning cars} \rangle$, with the occupation swapped out for the name ($\langle \text{Mary, repairs, cars} \rangle$). One of them is added to the training set, and one to the validation set. Moreover, all 200 general facts about occupations are also stored as a test set for a separate experiment.

This provides us with a similar setup as in the case of general deductive reasoning. Firstly, we can investigate how a model behaves when it has to guess the profession of an individual, as described in the previous section. Secondly, we can test its general background knowledge about occupations. Finally, we can find out how well the model determines the individuals' occupations when additionally provided with knowledge about what these individuals do. Again, we expect this score to be bound by the first two results. We additionally split this score into the *stereotypically feminine*, *stereotypically masculine*, *anti-stereotypically feminine*, and *anti-stereotypically masculine* sets to observe the possible difference between genders.

Testing a model on this dataset allows us to investigate to what degree the gender stereotypes impact the ability of the model to perform deductive reasoning. That is, one can investigate to what degree the model overcomes stereotypes in the presence of counter-evidence.

5. Diagnostic experiments

In this section, we compare various models from Section 3 on the DOGE dataset. Since pre-trained models with a larger dimension of embeddings have generally proved to be more successful, we only perform experiments with the largest variant of each model. For the majority of the datasets, the experiments should be conducted in a zero-shot setting.

All KBC models were tested only with GRU encoders, since the choice of an encoder makes no difference in a zero-shot setting.

The datasets that test a models' ability to perform deductive reasoning require additional training. On these datasets, each KBC model was evaluated both with a GRU and NOENCODER setup in the same way as in Section 3. When evaluating the model on the DOGE gender stereotype dataset, the best hyperparameter setup is found on the training and validation set. The learning rate is taken from $\{3 \cdot 10^{-4}, 10^{-4}, 3 \cdot 10^{-5}\}$. The dropout rate for CONVE and TUCKER is taken from $\{0.2, 0.3\}$ and $\{0.3, 0.4\}$, respectively. The weight of N3 regularization for 5*E is taken from $\{1.0, 0.3\}$. The same setup was then used for fine-tuning on the train set of the deductive reasoning dataset, where a validation set was not constructed.

Impact of GLOVE embeddings When pre-training on the OLPBENCH dataset, word embeddings were initialized from GLOVE vectors [32]. Unfortunately, such word embeddings are known to contain undesired gender stereotypes [6]. To understand their impact on potentially biased behaviour of pre-trained KBC models, we additionally pre-train the GRU_CONVE and GRU_TUCKER models with randomly initialized word embeddings. They were pre-trained on OLPBENCH with the same hyperparameter setup as their GLOVE-initialized counterparts. We did not repeat this experiment with a 5*E model due to larger computational cost of such an experiment.

6. Transfer learning experimental results

This section contains the outcome of pre-training and fine-tuning experiments. For fine-tuning, we repeated experiments five times and report their average and standard deviation. In addition, zero-shot transfer is investigated in the later part.

For each model, its mean rank (MR), mean reciprocal rank (MRR), and Hits at 10 (H@10) metrics on the test set are reported. We selected $N = 10$ for comparison since it was the most consistently reported Hits@N metric in related work.

Pre-training results The performance of the pre-trained models on OlpBench is given in Table 2. The introduced models obtain better scores than the previous best approach based on COMPLEX, however, we mainly attribute the improvement to the use of better KBC models.

OKBC results The results of experiments on REVERB20K and REVERB45K are given in Tables 3 and 4. The models strictly improve their performance when pre-trained on OLPBENCH. This improvement is particularly noticeable for NOENCODER models, which tend to overfit and achieve poor results without pre-training. However, when initialized with a pre-trained model, their ability to generalize improves. 5*E seems to be an exception to this, likely because there are no shared parameters between relations and entities, resulting in a weaker regularization. GRU-based models do not seem to suffer as severely from overfitting, but their performance still visibly improves when they are pre-trained on OLPBENCH.

Table 3

Comparison of different models with and without pre-training on the OKBC benchmarks REVERB20K. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Each score is averaged over 5 runs, with their standard deviation given in parentheses. Separated from the rest with two lines are the previous best results on the datasets, and TUCKER results with $d = 200$ for a fair BIGGRAPH comparison. The best overall value in each column is written in **bold**. Results denoted with \dagger and \blacklozenge were taken from [20] and [11], respectively.

Model	Pre-trained?	ReVerb20K		
		MR	MRR	H@10
NOENCODER_TUCKER	no	2577(22.8)	.194(.002)	.267(.003)
	yes	<i>307(5.48)</i>	<i>.377(.001)</i>	<i>.539(.002)</i>
NOENCODER_CONVE	no	1781(19.11)	.273(.001)	.372(.004)
	yes	<i>229(3.65)</i>	<i>.402(.000)</i>	<i>.565(.002)</i>
NOENCODER_5★E	no	2736(18.15)	.128(.002)	.192(.003)
	yes	<i>816(31.89)</i>	<i>.223(.001)</i>	<i>.324(.002)</i>
GRU_TUCKER	no	901(33.98)	.331(.003)	.451(.004)
	yes	<i>256(5.68)</i>	<i>.394(.001)</i>	<i>.556(.002)</i>
GRU_CONVE	no	326(14.36)	.383(.004)	.533(.005)
	yes	<i>184(2.58)</i>	<i>.408(.002)</i>	<i>.570(.002)</i>
GRU_5★E	no	353(24.08)	.386(.004)	.548(.001)
	yes	<i>201(5.21)</i>	<i>.415(.002)</i>	<i>.589(.001)</i>
\blacklozenge OKGIT(CONVE)	yes [▲]	527	.359	.499
\dagger CARE(CONVE, LAN)	no	973	.318	.439
\dagger TRANSE	no	1426	.126	.299
\dagger TRANSH	no	1464	.129	.303
NOENCODER_TUCKER _{d=200}	no	2724(32.45)	.162(.006)	.234(.006)
	yes	<i>323(7.79)</i>	<i>.366(.001)</i>	<i>.525(.002)</i>
BIGGRAPH_TUCKER _{d=200}	yes [▲]	1907	.215	.291

[▲] Unlike all other models with a *yes* entry, BIGGRAPH_TUCKER and OKGIT were not pre-trained on OLPBENCH, but on pre-trained WikiData embeddings and with the masked language modeling objective, respectively.

Finally, the best introduced model outperforms the state-of-the-art approach by Chandrahas and Talukdar [11] on REVERB20K and REVERB45K. Even when compared to pre-trained GRU_CONVE, which is based on the same KBC model, OKGIT(CONVE) and CARE(CONVE, LAN) lag behind. This is particularly surprising, because the BERT and ROBERTA language models, used by OKGIT(CONVE), had received several orders of magnitude more pre-training on unstructured text, making the results of the introduced approach even more significant.

Similarly, the initialization of models with BIGGRAPH seems to help the performance, however, such an approach is in turn outperformed by a NOENCODER model, initialized with pre-trained encoders instead. This indicates that the suggested pre-training is much more efficient, despite the smaller computational cost.

KBC results To evaluate the impact of pre-training on larger canonicalized knowledge bases, we compare the performance of models on FB15K237 and WN18RR. For brevity, we treat the choice of an encoder as a hyperparameter and report the better of the two models in Tables 5 and 6.

Pre-trained models outperform their randomly initialized counterparts as well, however, the differences are usually smaller. There are several reasons that can explain the small difference in performance, primarily the difference in dataset size. The best-performing models on FB15K237 and WN18RR only made between 3 and 12 times more steps during pre-training than during fine-tuning. For comparison, this ratio was between 250 to 1000 for REVERB20K. The smaller improvements on FB15K237 and WN18RR can also be explained by the domain shift, as already described in Section 3.1.

Table 5 additionally includes multiple recently published implementations of CONVE, 5★E, and TUCKER, as well as other well-performing models in KBC. The comparison with all these models should be taken with a grain of salt, as other reported models were often trained with a much larger hyperparameter space, as well as additional techniques for regularization (e.g., DURA [49] or label smoothing [3]) and sampling (e.g., self-adversarial sampling [1]). Due to a large number of models and baselines, we could not expand the hyperparameter search without compromising the fair evaluation of all compared models.

The pre-trained models usually obtain their best result with a larger dimension compared to their randomly-initialized counterparts. Pre-training thus serves as a type of regularization, allowing us to fine-tune larger models. For this reason, pre-training on even larger datasets and increasing the number of parameters is likely to result in even more significant improvements across many KBC datasets in future work.

Table 4

Comparison of different models with and without pre-training on the OKBC benchmark RE-VERB45K. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Each score is averaged over 5 runs, with their standard deviation given in parentheses. Separated from the rest with two lines are the previous best results on the datasets, and TUCKER results with $d = 200$ for a fair BIGGRAPH comparison. The best overall value in each column is written in **bold**. Results denoted with † and ♦ were taken from [20] and [11], respectively.

Model	Pre-trained?	ReVerb45K		
		MR	MRR	H@10
NoENCODER_TUCKER	no	5110(157.7)	.096(.002)	.122(.003)
	yes	<i>810(6.04)</i>	<i>.299(.001)</i>	<i>.451(.002)</i>
NoENCODER_CONVE	no	2837(71.67)	.223(.001)	.324(.002)
	yes	<i>660(13.54)</i>	<i>.344(.001)</i>	<i>.502(.001)</i>
NoENCODER_5★E	no	3176(41.59)	.151(.001)	.214(.000)
	yes	<i>2723(20.15)</i>	<i>.176(.000)</i>	<i>.248(.002)</i>
GRU_TUCKER	no	2047(15.49)	.267(.003)	.364(.004)
	yes	<i>808(75.13)</i>	<i>.322(.001)</i>	<i>.458(.008)</i>
GRU_CONVE	no	904(37.97)	.335(.004)	.476(.007)
	yes	<i>582(19.25)</i>	<i>.353(.001)</i>	<i>.507(.004)</i>
GRU_5★E	no	819(30.51)	.353(.002)	.511(.002)
	yes	<i>572(16.9)</i>	<i>.376(.001)</i>	<i>.534(.001)</i>
♦ OKGIT(CONVE)	yes [▲]	773.9	.332	.464
† CARE(CONVE, LAN)	no	1308	.324	.456
† TRANSE	no	2956	.193	.361
† TRANSH	no	2998	.194	.362
NoENCODER_TUCKER _{d=200}	no	5911(53.85)	.087(.001)	.107(.002)
	yes	<i>824(12.74)</i>	<i>.275(.000)</i>	<i>.423(.001)</i>
BIGGRAPH_TUCKER _{d=200}	yes [▲]	2285	.234	.337

[▲] Unlike all other models with a *yes* entry, BIGGRAPH_TUCKER and OKGIT were not pre-trained on OLPBENCH, but on pre-trained WikiData embeddings and with the masked language modeling objective, respectively.

Table 5

Comparison of different models with and without pre-training on the KBC benchmarks FB15K237. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Separated from the rest with two lines, previous scores obtained with the CONVE, 5★E, and TUCKER models are listed, followed by other well-performing models in the literature. The best overall result in each column is highlighted in **bold**.

Model	pre-trained?	FB15K237		
		MR	MRR	H@10
TUCKER	no	203(3.82)	.351(.001)	.531(.001)
	yes	<i>169(1.52)</i>	<i>.356(.001)</i>	<i>.537(.001)</i>
CONVE	no	164(1.97)	.332(.001)	.518(.002)
	yes	<i>157(0.77)</i>	<i>.355(.001)</i>	<i>.509(.001)</i>
5★E	no	161(0.25)	.350(.001)	.538(.001)
	yes	<i>153(1.15)</i>	<i>.355(.001)</i>	<i>.541(.0001)</i>
¹ CONVE	no	244	.325	.501
² CONVE	no	–	.339	.536
³ TUCKER	no	–	.358	.544
⁴ 5★E	no	–	.37	.560
⁵ KG-BERT	yes	153	–	.420
⁶ BOXE	no	163	.337	.538
⁷ COMPLEX	no	–	.370	.560
⁸ COMPLEX-DURA	no	–	.371	.560
⁸ RESCAL-DURA	no	–	.368	.550
⁹ MLMLM	yes	411	.259	.403

¹ [15]; ² [36]; ³ [3]; ⁴ [31]; ⁵ [48]; ⁶ [1]; ⁷ [25]; ⁸ [49]; ⁹ [13].

Table 6

Comparison of different models with and without pre-training on the KBC benchmarks WN18RR. The scores of each model are reported with and without pre-training, with the better of the two written in *italics*. Separated from the rest with two lines, previous scores obtained with the CONVE, 5★E, and TUCKER models are listed, followed by other well-performing models in the literature. The best overall result in each column is highlighted in **bold**.

Model	pre-trained?	WN18RR		
		MR	MRR	H@10
TUCKER	no	3801(62.68)	.466(.001)	.528(.002)
	yes	<i>3526(90.86)</i>	<i>.468(.004)</i>	<i>.529(.003)</i>
CONVE	no	6187(91.79)	.425(.002)	.471(.003)
	yes	<i>5811(63.42)</i>	<i>.427(.002)</i>	<i>.474(.003)</i>
5★E	no	2686(209.96)	.489(.002)	.574(.002)
	yes	<i>2674(87.23)</i>	<i>.491(.001)</i>	<i>.579(.003)</i>
¹ CONVE	no	4187	.430	.520
² CONVE	no	–	.442	.504
⁴ 5★E	no	–	.500	.590
⁵ KG-BERT	yes	97	–	.524
⁶ BOXE	no	3207	.451	.541
⁷ COMPLEX	no	–	.480	.570
⁸ COMPLEX-DURA	no	–	.491	.571
⁸ RESCAL-DURA	no	–	.498	.577
⁹ MLMMLM	yes	1603	.502	.611

¹ [15]; ² [36]; ³ [3]; ⁴ [31]; ⁵ [48]; ⁶ [1]; ⁷ [25]; ⁸ [49]; ⁹ [13].

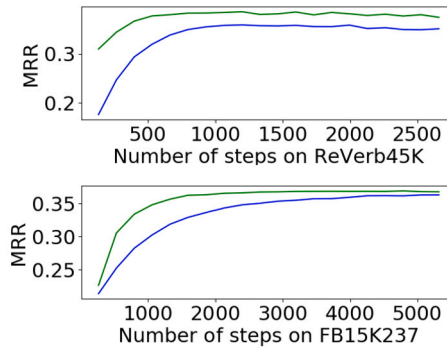


Fig. 3. Comparing convergence of the best randomly initialized (blue) and pre-trained (green) models on REVERB45K and FB15K237. Pre-trained models converge in fewer training steps despite a smaller learning rate.

Not only does pre-training allow training of larger models, the pre-trained models also require fewer training steps to obtain a similar performance, as seen in Fig. 3. Even though the pre-trained models were fine-tuned with a smaller learning rate, they converge in fewer steps, which can be attributed to the pre-training. The experiments summarized in the figure were performed with the best hyperparameter setup for both pre-trained and randomly initialized models.

Finally, we highlight that language modeling as pre-training hardly justifies the enormous computational cost. On the majority of metrics, KG-BERT performs worse than models with fewer parameters and less pre-training, with a notable exception of remarkable MR on the WN18RR dataset.

6.1. Zero-shot experiments

To gain a better understanding of what kind of knowledge is transferred between the datasets, we investigate the zero-shot performance of pre-trained models on REVERB20K and REVERB45K, where the impact of pre-training was the strongest. That means that we follow the same setup as in Section 5 with all models. If pre-training improvement was mainly due to the direct memorization of facts, the zero-shot performance should already be high without fine-tuning.

The results of the zero-shot evaluation are given in Table 7. Given is the performance of all models with and without pre-training, the latter being equivalent to a random baseline. Note that since no fine-tuning takes place, the choice of the encoder for evaluation does not matter. We thus chose to only evaluate one of them.

Observing the results, we see that pre-training the models results in a lower MR, but not necessarily in a much higher MRR. Even when MRR does improve, the difference is much smaller than when comparing fine-tuned models in Tables 3 and 4. This implies

Table 7

Comparison of the zero-shot performance of different models on the OKBC benchmarks REVERB20K and REVERB45K. The scores of each model are reported with and without pre-training on OLPBENCH, with the better of the two written in *italics*. Note that the model that was not pre-trained is equivalent to a random baseline.

Model	pre-trained?	REVERB20K			REVERB45K		
		MR	MRR	Hits@10	MR	MRR	Hits@10
NoENCODER_TUCKER	no	4862	.001	.002	8409	.001	.001
	yes	<i>1486</i>	<i>.012</i>	<i>.026</i>	<i>1737</i>	<i>.019</i>	<i>.041</i>
NoENCODER_CONVE	no	4600	.002	.004	8464	.001	.001
	yes	<i>1480</i>	<i>.003</i>	<i>.000</i>	<i>1962</i>	<i>.003</i>	<i>.001</i>
NoENCODER_5★E	no	4795	.001	.000	8445	.001	.002
	yes	<i>1422</i>	<i>.005</i>	<i>.000</i>	<i>1701</i>	<i>.014</i>	<i>.011</i>

Table 8

Examples of predictions of ConVE model with NoEncoder on ReVerb20K dataset. YES and NO denote whether we used pre-trained model or not. The word underlined is the target answer. Wrong answers are randomly sampled from entities listed in the test set.

Data Instance	Rank (NO)	Rank (YES)
HEAD: fuel, RELATION: be make from, TAIL: <u>crude oil</u> WRONG ANSWERS: constipation, vegetable, 1831, heaven, hour, high school, original, dmca, version	5	1
HEAD: <u>ethanol fuel</u> , RELATION: be made of, TAIL: ethanol WRONG ANSWERS: population genetic, hannibal, tesla roadste, dvd, identity theft, corey feldman, Clinton, gate, monroe	6	2
HEAD: nyx, RELATION: be the goddess of, TAIL: <u>darkness</u> WRONG ANSWERS: united states, presidency, sale, Russia, bad idea, king, progress, obama, event	9	1

that the improvement induced by pre-training likely does not happen only due to the memorization of facts from OLPBENCH. On the other hand, the MR of pre-trained models is comparable or even better than the MR of randomly initialized NoENCODER models, fine-tuned on the REVERB datasets, as reported in Tables 3 and 4. Hence, pre-trained models carry a lot of “approximate knowledge”. This means that correct answers tend to not be ranked low, even when the model does not give the correct answer. This is consistent with earlier remarks on pre-training serving as a type of regularization. Table 8 shows several corresponding examples of the ConVE model with NoEncoder on the ReVerb20K dataset. It shows that the correct answers are ranked better in the pre-trained model. These are all instances of “general knowledge” that are hard to deduce from the training set alone—e.g., “crude oil” appears only once in the training set of REVERB20K. On the other hand, we can find the fact (crude oil, is used as, fossil fuel) in OLPBENCH, from which the first fact can be deduced.

Knowing that the REVERB20K and REVERB45K test sets consist of facts which contain at least one previously unseen entity or relation, this experiment can be viewed as a test of out-of-distribution generalization. Comparing the zero-shot MRR results with the OLPBENCH results suggests that while OKBC models are capable of out-of-domain generalization to unseen entities and relations, there is still room for improvement.

7. Results of diagnostic experiments

This chapter contains the results of all evaluated models on all parts of the DOGE dataset. Results are reported in MR, MRR, and Hits@1 metrics, except for parts of the dataset that measure consistency, for which the mean and standard deviation are reported instead. Hits@N metrics for other values of N were not reported due to the small number of possible answers (10 or 50).

7.1. General knowledge

All three pre-trained KBC architectures are compared on the general knowledge part of the DOGE dataset, with results reported in Table 9. It is evident that the models are fairly consistent with each other relative to how well they do in different categories. This is unsurprising since they were all trained on OLPBENCH. Nevertheless, it also means that it can be difficult to conclude which of the models handles what type of data better. CONVE seems to often outperform other models in most categories, despite often ranking the lowest of the three models on commonly used KBC benchmarks.

Inconsistent results between DOGE and most other datasets can be explained by the difference in their construction. Unlike other KBC datasets, DOGE was not obtained by randomly splitting an existing knowledge base. Models trained on OLPBENCH and evaluated on DOGE face a shift of data distribution, requiring better generalization capabilities. This is the first result of this kind, indicating that existing KBC benchmarks only cover a specific scenario, which may incorrectly reward models with lower generalization capabilities.

Table 9

Comparison of the three pre-trained architectures on the general knowledge subset of the DOGE dataset. Each column contains scores of a model on different categories, as defined by Encyclopedia Britannica.

category	TUCKER			CONVE			5*E		
	MR	MRR	H@1	MR	MRR	H@1	MR	MRR	H@1
Entertain. & Pop Culture	3.55	.510	.31	3.49	.520	.33	3.40	.559	.39
Geography & Travel	3.03	.601	.43	2.87	.620	.46	2.40	.702	.57
Health & Medicine	3.95	.461	.27	3.43	.493	.27	3.18	.561	.39
Lifestyle & Social Issues	3.35	.535	.36	3.21	.564	.39	3.25	.567	.39
Literature	3.69	.509	.33	3.30	.550	.37	3.08	.602	.45
Philosophy & Religion	3.22	.545	.37	3.25	.544	.34	3.01	.599	.43
Politics, Law & Government	3.45	.532	.35	2.92	.563	.36	2.79	.639	.50
Science	3.60	.505	.32	3.94	.450	.23	3.28	.569	.41
Sports & Recreation	3.71	.470	.25	3.63	.472	.27	3.02	.587	.41
Technology	3.45	.532	.35	3.30	.534	.34	3.13	.593	.43
Visual Arts	3.77	.480	.28	3.66	.492	.32	3.50	.557	.40
World History	3.28	.520	.30	3.19	.580	.42	2.55	.642	.46

Table 10

A comparison of three pre-trained architectures on subsets of the DOGE dataset that probe the consistency of predictions. Mean values are usually close to zero, indicating that the deviation was not caused by poor annotation.

	TUCKER		CONVE		5*E	
	mean	stdev	mean	stdev	mean	stdev
Entity Synonyms	0.692	3.910	0.931	3.397	0.836	3.559
Relation Synonyms	0.177	3.027	0.403	2.755	0.109	1.958
Inverse Relations	0.239	3.030	0.183	2.609	0.090	2.119

7.2. Model consistency

All three KBC pre-trained architectures are compared on their ability to make consistent prediction in the presence of synonyms and inverse relations, with results reported in Table 10. None of the models are particularly consistent on either of the tests, with CONVE being worse than 5*E, especially in “Relation Synonyms”. The discrepancy in consistency may be due to the comparably better performance of 5*E on DOGE, particularly in “Science” and “World History” topics, where the annotated data points for “Relation Synonyms” account for 19% of total data. Overall, it is safe to conclude that none of the tested models can be called *consistent*, which is unfortunately common for neural network models.

7.3. Deductive reasoning

In this section, we report the results on the test of deductive reasoning. The scores obtained by all three evaluated models are given in Table 11. All models achieve approximately chance performance on the *No Added Facts* experiments, indicating that the answers of this subset cannot be deduced through undesired clues. Moreover, all models achieve a better-than-random guessing score on the *Background Knowledge* experiments, indicating that they do contain the relevant background knowledge for many instances. If the results of the models on this set of instances were poor, the results from this whole section would be inconclusive. Finally, the results on *With Added Facts* lie somewhere in between the two, as expected. This indicates that the models can indeed combine newly obtained facts with their background knowledge, however, a big gap to *Background Knowledge* results indicates that there is a lot of room for improvement.

The impact of the encoder varies by KBC model. For CONVE and TUCKER, there is no conclusive answer to which encoder is the better choice, as it depends on the choice of metric. NOENCODER versions obtain better MR, while GRU versions obtain better MRR and Hits@1 scores. For 5*E, NOENCODER is visibly the superior choice, making the NOENCODER_5*E the best model for deductive reasoning out of all the compared setups. The reason behind this likely lies in its shallow architecture. Since there are no shared parameters or stacked layers, training primarily affects the embeddings of entities in the training data while preserving the structure of everything else.

7.4. Gender stereotypes

Finally, all deductive reasoning tests are repeated on the subset of DOGE that detects gender stereotypes. The results are given in Table 12 and are reported separately for each gender and separately for stereotypical and anti-stereotypical occupations. Finally, we test whether the model predictions are affected by the stereotypes indicating the presence of historical bias. To see whether the difference in performance is statistically significant, we use the Wilcoxon signed-rank test. More specifically, we test whether swapping the name of a gender that is not stereotypically associated with an occupation for a name of the opposite gender impacts the rank of the correct answer.

Table 11

Comparison of the three pre-trained architectures on the deductive reasoning subset of the DOGE dataset. *Background Knowledge* and *No Added Facts* results serve as soft upper and lower bounds to the *With Added Facts* results, which can be correctly deduced only by combining newly added facts with the relevant background knowledge. Fine-tuning is done both with GRU and NOENCODER setup to compare the impact of the two.

	TUCKER			CONVE			5*E		
	MR	MRR	H@1	MR	MRR	H@1	MR	MRR	H@1
Background Knowledge	3.26	.556	.373	3.01	.598	.435	3.24	.574	.408
No Added Facts	5.09	.343	.15	5.07	.334	.13	4.65	.350	.14
With Added Facts _{GRU}	4.16	.457	.28	4.33	.403	.22	4.11	.464	.30
With Added Facts _{NOENC.}	4.39	.415	.23	4.69	.344	.13	4.25	.416	.22

Table 12

Comparison of three pre-trained architectures on the gender-bias detection subset of DOGE. The dataset is split into Stereotypical (St) and Anti-Stereotypical (Anti-St) examples. The first of the three tables contains information on the experiments that do not require fine-tuning: stereotype detection and background knowledge. The second and the third table contain information on deductive reasoning about gender for GRU and NOENCODER encoders, respectively.

No Added Facts	TUCKER			CONVE			5*E		
	MR	MRR	H@1	MR	MRR	H@1	MR	MRR	H@1
St Masculine	22.51	.096	.02	24.27	.083	.02	22.9	.096	.01
St Feminine	21.07	.093	.00	18.45	.138	.02	20.95	.119	.04
Anti-St Masculine	29.96	.079	.03	32.18	.037	.00	30.32	.066	.01
Anti-St Feminine	29.21	.062	.00	28.41	.0642	.00	28.11	.072	.02
p-value	$6.04 \cdot 10^{-15}$			$2.19 \cdot 10^{-15}$			$1.48 \cdot 10^{-11}$		
Background Knowledge	9.575	.417	.275	7.910	.4307	.265	7.155	.503	.345

With Added Facts _{GRU}	TUCKER			CONVE			5*E		
	MR	MRR	H@1	MR	MRR	H@1	MR	MRR	H@1
St Masculine	18.09	.154	.08	18.69	.176	.09	14.58	.184	.04
St Feminine	20.61	.138	.03	18.22	.224	.14	15.33	.175	.04
Anti-St Masculine	24.37	.084	.01	25.22	.079	.00	19.16	.198	.10
Anti-St Feminine	26.83	.109	.03	25.63	.098	.03	21.36	.119	.03
p-value	$7.20 \cdot 10^{-9}$			$2.62 \cdot 10^{-12}$			$8.87 \cdot 10^{-6}$		

With Added Facts _{NOENC.}	TUCKER			CONVE			5*E		
	MR	MRR	H@1	MR	MRR	H@1	MR	MRR	H@1
St Masculine	16.68	.168	.07	15.68	.210	.07	21.72	.114	.02
St Feminine	20.23	.125	.00	16.20	.210	.08	22.05	.111	.03
Anti-St Masculine	23.10	.098	.03	21.36	.099	.01	28.47	.073	.01
Anti-St Feminine	27.57	.092	.03	22.58	.1343	.05	26.55	.073	.01
p-value	$7.03 \cdot 10^{-13}$			$3.81 \cdot 10^{-10}$			$5.43 \cdot 10^{-9}$		

The gender stereotypes are strongly present, resulting in a model being correct more often when the gender of the name matches the conceptual gender of the occupation. By comparing the results across all examples, both stereotypical and anti-stereotypical, we can see that the masculine names are on average ranked higher, likely due to representation bias in the training data. Moreover, all models obtain decent scores on background knowledge about occupations, making deductive reasoning experiments conclusive.

The observed patterns persist when additional relevant data are added to the models, regardless of the choice of the encoder. It is worth noting that the improvement from the additional knowledge about entities only moderately helps the models regardless of the gender or stereotypes. Connecting this with the results on deductive reasoning, we can conclude that models contain gender stereotypes and are unable to overcome them in the presence of additional data. The reason why they cannot overcome them lies in their general poor ability to perform deductive reasoning rather than a strong presence of stereotypes. Unfortunately, this means that all analyzed pre-trained KBC models can introduce gender-stereotyped facts into the target knowledge base, even in the presence of counter-evidence.

7.5. The impact of word embeddings

Finally, we report the results of re-training CONVE and TUCKER on OLPBENCH without the GLOVE word embeddings. The results on OLPBENCH are reported in Table 13. Evidently, the absence of GLOVE vectors significantly decreases the performance of both models.

Table 13

Comparison of GLOVE impact on OLPBENCH. TUCKER and CONVE are both trained with GLOVE and with randomly-initialized word embeddings. The better of the two in each column is written in *italics*.

	MR	MRR	H@10
GRU_TUCKER _{GLOVE}	<i>57.2K</i>	<i>.053</i>	<i>.097</i>
GRU_TUCKER _{NO_GLOVE}	65.1K	.042	.077
GRU_CONVE _{GLOVE}	<i>57.2K</i>	<i>.045</i>	<i>.086</i>
GRU_CONVE _{NO_GLOVE}	69.2K	.030	.057

Table 14

Comparison of pre-trained CONVE and TUCKER architectures without GLOVE embeddings on the gender-bias detection subset of DOGE. The dataset is split into a dataset with Stereotypical (St) and one with Anti-Stereotypical (Anti-St) examples. The first of the two tables contains information on experiments that do not require fine-tuning: stereotype detection and background knowledge. The second of the two tables contains the information on deductive reasoning about gender for GRU encoders. Fine-tuning with the NOENCODER setup was omitted for brevity.

No Added Facts	TUCKER			CONVE		
	MR	MRR	H@1	MR	MRR	H@1
St Masculine	23.34	.071	.00	23.32	.109	.03
St Feminine	20.56	.1396	.05	23.40	.113	.03
Anti-St Masculine	31.14	.042	.00	27.80	.054	.00
Anti-St Feminine	27.97	.104	.05	27.70	.070	.00
p-value	$3.03 \cdot 10^{-16}$			$1.67 \cdot 10^{-12}$		
Background Knowledge	10.19	.342	.200	10.70	.338	.195

With Added Facts _{GRU}	TUCKER			CONVE		
	MR	MRR	H@1	MR	MRR	H@1
St Masculine	19.54	.113	.01	19.12	.170	.07
St Feminine	21.21	.132	.04	21.94	.149	.05
Anti-St Masculine	25.47	.078	.02	23.39	.099	.02
Anti-St Feminine	26.80	.113	.04	26.23	.088	.02
p-value	$9.59 \cdot 10^{-10}$			$3.51 \cdot 10^{-07}$		

To see whether models without GLOVE embeddings are less susceptible to gender stereotypes, we evaluate them on the gender stereotype subset of the DOGE dataset. The results can be found in Table 14. Despite the slightly lower performance, the general trends in the results of these experiments are very similar to the results with the GLOVE embeddings in Table 12. This indicates that GLOVE embeddings definitely are not the only source of gender stereotypes, and while their removal results in an undesired performance drop, it does not decrease the biased behavior.

8. Related work

This section discusses the related work in knowledge base completion, transfer learning in NLP, and existing approaches to using external data or higher-level structures in knowledge base completion. The most attention is put on the discussion of the last part due to its closeness to this work.

Review of knowledge base completion approaches Approaches to knowledge base completion can be roughly split into two types, translational models and semantic similarity models. Translational models, such as TransE [28], BoxE [1], 5*E [31], or RotatE [39] usually model triplets as translations or similar geometric operations (e.g., rotations) in space. They are commonly “shallow” in the sense that they do not contain any neural networks with multiple layers. Instead, their design is guided by mathematical intuition.

Semantic similarity models usually map the pair (head, relation) into a vector space and measure its semantic similarity (usually dot product) with possible tails. Examples of such models are TUCKER [3], CONVE [15], and ComplEx [41]. Unlike CONVE, which contains a convolutional neural network, TUCKER and ComplEx are examples of bilinear models — a special case of semantic similarity models. In such models, the score of a triplet can be obtained as a multiplication of a relation-specific matrix with vector embeddings of the head and the tail.

Not all models for knowledge base completion fall strictly into these two categories. KG-BERT [48], for example, takes the entire triplet as the input and does not compute any intermediate representations.

We decided not to discuss the performance of the mentioned models on benchmarks as part of the literature review. Their accuracy strongly depends on the training regime, choice of the loss function, and other details, which are not consistent across their

respective papers. Surveys of existing KBC methods have repeatedly shown that older models can outperform more recent approaches when trained in the same environment [36,2].

Review of transfer learning in natural language processing Any model claiming the ability to process human language requires an immense amount of background knowledge of words and their meanings. All models operating on human language share vocabulary, prompting the development of common resources such as pre-trained models or word embeddings trained on large unsupervised corpora. Pre-training of models is done by initializing the model weights, word embeddings, or other parts of a model on a separate task with an abundance of data [32,16].

While pre-training on supervised tasks is possible and sometimes even more sample-efficient [14], the accessibility of raw text makes unsupervised approaches the more scalable option, typically resulting in better performance overall. A common existing approach to pre-training is to train a language model — a computational model that assigns a probability score to the input sequence of words [4] — on a large corpus of raw text [22,9]. A conceptually similar but more common approach is masked language modeling [16,29], which is not language modelling in the strict sense, because it only assigns probabilities to individual missing (masked) tokens instead of the entire sequence.

As an example of a pre-training method that is not based on predicting missing pieces of text, we highlight the unsupervised multi-lingual paraphrasing approach by Lewis et al. [27]. They show that one can obtain a model for machine translation in an unsupervised manner by training to paraphrase articles about the same news.

Further discussion of recent advances in pre-training in NLP is beyond the scope of this paper. For a more exhaustive survey, we direct the readers to Qiu et al. [34]

Review of uses of external data in knowledge base completion While large-scale transfer learning in KBC is not yet a widely-used method, multiple attempts to incorporate higher-level structures or external knowledge exist. The most common approach is to impose a hierarchy between entities or relations. This information is already available inside the KB, and many approaches just adapt the scoring function to raise its importance. *CTransR* [23] collects the input data into clusters and learns a separate relation embedding for each cluster. *Semantically smooth embedding (SSE)* [19] and *type-embodied knowledge representation (TKRL)* [47] assign types to entities and require the entities of the same type (determined through the *ISA* relation) to stay close together.

A different approach to incorporate additional information is to analyze paths. Unlike regular approaches that only analyze single relations (paths of length 1), models like *PTransE* [28] additionally incorporate information on the paths of longer lengths. The number of paths increases exponentially with the length, hence all such approaches resort to sampling and approximate methods. Incorporating paths improved the quality of embeddings at the expense of computational complexity. Thus, they do not appear in combination with more complex models.

The models described so far rely on the information provided by the knowledge base. However, a lot of knowledge is not incorporated in the knowledge bases directly. Various models try to increase the amount of available information by using human-written text. The simplest approach of incorporating natural text is initialization of parameters from precomputed word embeddings based on the textual description of the entities. This approach is used by the *NTN* model [37].

More advanced models, such as *DKRL* [46] and *TEKE* [44], train the entity embeddings jointly with the text model and try to align them. Wang and Li [44] also define a textual context embedding and try to learn the mapping between the textual and knowledge base embedding. Xie et al. [46], on the other hand, make use of an encoder that generates an embedding given a description of the entity. They show that their approach generalizes even to previously unseen entities when a description of these entities is available. Models that additionally incorporate natural text perform better when faced with previously unseen entities. Yao et al. [48] make use of the large-scale pre-trained transformer model BERT to classify whether a fact is true. The main drawback of such an approach is its speed – BERT is large, and it can take weeks of GPU runtime to evaluate such a model on a benchmark test set.

Many of the ideas described in this section thus far have been later used to tackle open knowledge bases. Earlier attempts at open knowledge base completion are tied to existing work on the canonicalization of knowledge bases. To canonicalize open knowledge bases, automatic canonicalization tools cluster entities using manually defined features [18] or by finding additional information from external knowledge sources [42]. Gupta et al. [20] use clusters obtained with these tools to augment entity embeddings for KBC. Note that Gupta et al. [20] use RNN-based encoders to encode relations, but not to encode entities. Broscheit et al. [8], on the other hand, introduce a model with RNN-based encoders for both entities and relations. Finally, Chandrhas and Talukdar [11] use both KB canonicalization tools and the large-scale pre-trained model BERT, combining their predictions to make a more informed decision.

9. Conclusion

In this work, we have introduced a novel approach to transfer learning or domain adaptation between various knowledge base completion datasets and introduced DOGE, a novel diagnostic dataset for the analysis of pre-trained models for knowledge base completion. The main strength of the introduced pre-trained method is the ability to benefit from pre-training on uncanonicalized knowledge bases, constructed from facts that were collected from unstructured text. Scaling the introduced method up would let us train large-scale pre-trained models, which have already shown to be incredibly successful in NLP. The proposed method was tested on 5 different existing datasets, 1 for pre-training and 4 for fine-tuning, demonstrating that pre-training improves the performance of models. Pre-training turned out to be particularly beneficial on small-scale datasets, where the most significant gains were obtained,

i.e., a 6% absolute increase of MRR and a 65% decrease of MR in comparison to the previously best method on REVERB20K, despite not relying on large pre-trained models like BERT.

We used the introduced DOGE dataset to evaluate pre-trained models introduced in Section 3. The results indicated that the existing pre-trained models lack robustness against synonyms and inverse relations and strongly rely on gender stereotypes. On a more positive note, they seem to cover all major areas of knowledge decently. The main highlighted problem with the obtained pre-trained models is the persistence of harmful stereotypes, which the model cannot remove even with the introduction of counter-evidence. It may thus happen that a pre-trained model includes an incorrect fact ⟨Mary, is, childcare worker⟩ into the target knowledge base even if the knowledge base contains the fact ⟨Mary, repairs, cars⟩.

While this research rigorously investigated a specific direction of pre-training, i.e., training on an uncanonicalized knowledge base and transferring to smaller closed and open knowledge bases, there is a lot of potential on training on canonicalized knowledge bases as well as transferring to different related tasks, such as named entity recognition and knowledge base canonicalization.

This work carries important implications for future developments of the field. Pre-training has led to important breakthroughs in other areas where neural networks are applied, and it is reasonable to expect that the same will happen in knowledge base completion. Scaling up the pre-training is expected to improve the performance of KBC models even further, allowing their use on small and scarce knowledge bases. This allows for their use on early-stage knowledge bases, simplifying their construction.

Results on the DOGE dataset demonstrate that existing KBC benchmarks do not test all properties exhaustively. This benchmark allows engineers to diagnose the behavior of their models more precisely, providing insights beyond the theoretical limitation and overall performance.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Vid Kocijan reports equipment, drugs, or supplies was provided by Scan Computers International Ltd.

Data availability

The data will be shared on the link given in the paper.

Acknowledgements

The authors would like to thank Ralph Abboud for his helpful comments on the paper manuscript and Ana Štuhec for suggesting the dataset name. This work was supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1, the AXA Research Fund, the ESRC grant “Unlocking the Potential of AI for English Law”, and the EPSRC Studentship OUCS/EPSC-NPIF/VK/1123106. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1) and GPU computing support by Scan Computers International Ltd.

References

- [1] R. Abboud, İ.İ. Ceylan, T. Lukasiewicz, T. Salvatori, Boxe: a box embedding model for knowledge base completion, in: *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 9649–9661.
- [2] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing light into the dark: a large-scale evaluation of knowledge graph embedding models under a unified framework, *CoRR*, arXiv:2006.13365 [abs], 2020.
- [3] I. Balazević, C. Allen, T. 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)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5185–5194.
- [4] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, *J. Mach. Learn. Res.* 3 (2003) 1137–1155.
- [5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD’08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1247–1250.
- [6] T. Bolukbasi, K.-W. Chang, J.Y. Zou, V. Saligrama, A.T. Kalai, Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings, *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016.
- [7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-Relational Data, *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013.
- [8] S. Broscheit, K. Gashtevski, Y. Wang, R. Gemulla, Can we predict new facts with open knowledge graph embeddings? A benchmark for open link prediction, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020, pp. 2296–2308. Online.
- [9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.
- [10] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, in: *Proceedings of the 3rd International Conference on Learning Representation ICLR*, Banff, Canada, 2014.
- [11] Chandras, P. Talukdar, OKGIT: open knowledge graph link prediction with implicit types, in: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021, pp. 2546–2559. Online.

- [12] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.
- [13] L. Cloutatre, P. Trempe, A. Zouaq, S. Chandar, MLMML: link prediction with mean likelihood masked language model, in: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021, pp. 4321–4331. Online, <https://aclanthology.org/2021.findings-acl.378>.
- [14] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 670–680.
- [15] T. Dettmers, M. Pasquale, S. Pontus, S. Riedel, Convolutional 2D knowledge graph embeddings, in: *Proceedings of the 32th Association of Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, New Orleans, USA, 2018, pp. 1811–1818.
- [16] J. Devlin, M.-W. Chang, K. Lee, K. 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*, vol. 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.
- [17] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, pp. 1535–1545.
- [18] L. Galárraga, G. Heitz, K. Murphy, F.M. Suchanek, Canonicalizing open knowledge bases, in: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM'14*, Association for Computing Machinery, New York, NY, USA, 2014, pp. 1679–1688.
- [19] S. Guo, Q. Wang, B. Wang, L. Wang, L. Guo, Semantically smooth knowledge graph embedding, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol. 1: Long Papers)*, Association for Computational Linguistics, Beijing, China, 2015, pp. 84–94.
- [20] S. Gupta, S. Kenkre, P. Talukdar CaRe, Open knowledge graph embeddings, 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)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 378–388.
- [21] P. Hohenacker, F. Mtumbuka, V. Kocijan, T. Lukasiewicz, Systematic comparison of neural architectures and training approaches for open information extraction, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 8554–8565. Online.
- [22] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339.
- [23] W. Huang, G. Li, Z. Jin, Improved knowledge base completion by path-augmented TransR model, CoRR, arXiv:1610.04073 [abs], 2016.
- [24] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, in: *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.
- [25] T. Lacroix, N. Usunier, G. Obozinski, Canonical tensor decomposition for knowledge base completion, in: *Proceedings of the 35th International Conference on Machine Learning ICML*, 2018.
- [26] A. Lerer, L. Wu, J. Shen, T. Lacroix, L. Wehrstedt, A. Bose, A. Peysakhovich, PyTorch-BigGraph: a large-scale graph embedding system, in: *Proceedings of the 2nd SysML Conference*, Palo Alto, CA, USA, 2019.
- [27] M. Lewis, M. Ghazvininejad, G. Ghosh, A. Aghajanyan, S. Wang, L. Zettlemoyer, Pre-training via paraphrasing, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 18470–18481.
- [28] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 705–714.
- [29] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov Roberta, A robustly optimized BERT pretraining approach, CoRR, arXiv:1907.11692 [abs], 2019.
- [30] G.A. Miller, WordNet: a lexical database for English, *Commun. ACM* (1995).
- [31] V. Nayyeri, S. Vahdati, C. Aykul, J. Lehmann, 5* knowledge graph embeddings with projective transformations, in: *Proceedings of the 35th Association of Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, 2021.
- [32] J. Pennington, R. Socher, C. Manning GloVe, Global vectors for word representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543.
- [33] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, 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)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473.
- [34] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: a survey, *Sci. China, Technol. Sci.* (2020) 1872–1897.
- [35] R. Rudinger, J. Naradowsky, B. Leonard, B. Van Durme, Gender bias in coreference resolution, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 8–14.
- [36] D. Ruffinelli, S. Broscheit, R. Gemulla, You CAN teach an old dog new tricks! On training knowledge graph embeddings, in: *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia, April 26–30, 2020, 2020.
- [37] R. Socher, D. Chen, C.D. Manning, A. Ng, Reasoning with Neural Tensor Networks for Knowledge Base Completion, *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013.
- [38] G. Stanovsky, J. Michael, L. Zettlemoyer, I. Dagan, Supervised open information extraction, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 885–895.
- [39] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: knowledge graph embedding by relational rotation in complex space, in: *International Conference on Learning Representations, New Orleans, LA, USA*, 2019.
- [40] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1499–1509.
- [41] T. Trouillon, C.R. Dance, E. Gaussier, J. Welbl, S. Riedel, G. Bouchard, Knowledge graph completion via complex tensor factorization, *J. Mach. Learn. Res.* 18 (2017) 4735–4772.
- [42] S. Vashishth, P. Jain, P. Talukdar, CESI: canonicalizing open knowledge bases using embeddings and side information, in: *Proceedings of the 2018 World Wide Web Conference, WWW'18*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 1317–1327.
- [43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: *Proceedings of the 6th International Conference on Learning Representation ICLR*, 2017.
- [44] Z. Wang, J. Li, Text-enhanced representation learning for knowledge graph, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, AAAI Press, 2016, pp. 1293–1299.
- [45] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: *Proceedings of the 28th Association of Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, 2014.

- [46] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proceedings of the Thirtieth Association of Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, AAAI Press, 2016.
- [47] R. Xie, Z. Liu, M. Sun, Representation learning of knowledge graphs with hierarchical types, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, AAAI Press, 2016, pp. 2965–2971.
- [48] L. Yao, C. Mao, Y. Luo, KG-BERT: BERT for knowledge graph completion, CoRR, arXiv:1909.03193 [abs], 2019.
- [49] Z. Zhang, J. Cai, J. Wang, Duality-induced regularizer for tensor factorization based knowledge graph completion, in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 21604–21615.
- [50] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Gender bias in coreference resolution: evaluation and debiasing methods, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 15–20.