Expanding Domain-specific Knowledge Graphs with Unknown Facts

Miao Hu (✉)[0000−0001−9055−563X], Zhiwei Lin, and Adele Marshall[0000−0001−5306−2756]

School of Mathematics and Physics, Queen's University Belfast, Belfast, United Kingdom
{mhu05, z.lin, a.h.marshall}@qub.ac.uk

Abstract. Many knowledge graphs have been created to support intelligent applications, such as search engines and recommendation systems. Some domain-specific knowledge graphs contain similar contents in nature (e.g., the FreeBase contains information about actors and movies which are the core of the IMDB). Adding relevant facts or triples from one knowledge graph into another domain-specific knowledge graph is key to expanding the coverage of the knowledge graph. The facts from one knowledge graph may contain unknown entities or relations that do not occur in the existing knowledge graphs, but it doesn't mean that these facts are not relevant and hence can not be added to an existing domain-specific knowledge graph. However, adding irrelevant facts will violate the inherent nature of the existing knowledge graph. In other words, the facts that conform to the subject matter of the existing domain-specific knowledge graph only can be added. Therefore, it is vital to filter out irrelevant facts in order to avoid such violations. This paper presents an embedding method called UFD to compute the relevance of the unknown facts to an existing domain-specific knowledge graph so that the relevant new facts from another knowledge graph can be added to the existing domain-specific knowledge graph. A new dataset, called UFD-303K, is created for evaluating unknown fact detection. The experiments show that our embedding method is very effective at distinguishing and adding relevant unknown facts to the existing knowledge graph. The code and datasets of this paper can be obtained from GitHub [1].

Keywords: Domain-specific knowledge graph · Knowledge graph expanding · Unknown facts detection · BERT pre-trained model

1 Introduction

Knowledge graphs are widely used in many information systems such as search engines and recommender systems. A *knowledge graph* (KG) $G = \{(h, r, t)|h, t \in E, r \in R\}$ is a set of triples where E is a set of entities, R is a set of relations between the entities. A triple (h, r, t) denotes that the head entity h has a relation of r with a tail entity t. For example, as shown in Fig. 1, ('Tom Cruise', '/film/producer/film', 'Mission:Impossible') is a triple where the head entity is 'Tom Cruise', the relation is 'film/producer/film', and the tail entity is 'Mission:Impossible'.

[1] https://github.com/MiaoHu-Pro/UFD

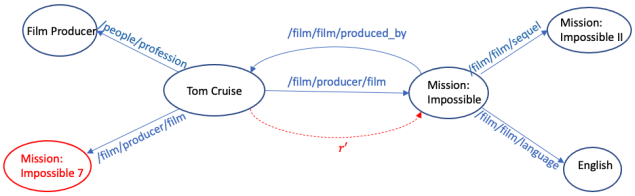

Fig. 1. An example of a sub-graph from FreeBase, where the nodes are entities, and the directed edges represent relationships between the entities. There are 2 relations, '/film/film/produced_by' and '/film/producer/film', between 'Tom Cruise' and 'Mission:Impossible'. And a new relation r' is introduced from out-of-KG. Then, an unknown fact, ('Tom Cruise', r','Mission:Impossible'), is emerging. By analogy, a new entity, 'Mission:impossible 7', is introduced, which constructs an unknown fact ('Tom Cruise', '/film/producer/film','Mission:Impossible 7'). Before adding them to the existing knowledge graph, it is necessary to judge whether these unknown facts constructed by new entities or new relations are relevant to the current knowledge graph.

The *Domain-specific Knowledge Graph* (Ds-KG) is described as "Domain Knowledge Graph is an explicit conceptualisation to a high-level subject-matter domain and its specific subdomains are represented in terms of semantically interrelated entities and relations." in [1]. In other words, Ds-KG is a knowledge representation for a specific domain application. Large-scale domain-specific knowledge graphs (such as FreeBase [2] and IMDB [3]) have played key roles in supporting intelligent question-answering, recommendation systems, and search engines [7]. Some of them may contain similar contents to some extent, for example, the FreeBase knowledge graph also contains facts [4] about movies, TV series, which overlaps with the content in the IMDB knowledge graph. Some of the facts in FreeBase conform to the subject matter of the IMDB, and these facts can be introduced into IMDB from FreeBase, which is a crucial way to enrich the IMDB.

Adding relevant facts from one knowledge graph U into another knowledge graph G is instrumental as this will help to enrich the content in the knowledge graph G and consequently improve the intelligent systems that use the knowledge graph G. Adding facts from one knowledge graph U into another existing domain-specific knowledge graph G is not easy, as not all the facts in U are relevant to the subject matter of the contents in G. Adding irrelevant facts may violate the inherent nature of the existing knowledge graph. In other words, we hope to introduce new facts that conform to the subject matters of the existing domain-specific knowledge graph [1]. For example, the triple of ('Geoffrey Hinton', 'recipient of', '2018 ACM A.M. Turing Award') about a computer scientist is certainly irrelevant to the IMDB knowledge graph as the IMDB knowledge graph contains triples about movies, TV programs and actors. Such facts should not be added to the IMDB knowledge graph to avoid violation. The facts from U

[2] www.freebase.com

[3] IMDB is the world's most popular and authoritative source for movie, TV and celebrity content (https://www.imdb.com/).

[4] 'facts' and 'triples' are used interchangeably without confusion in this paper

may contain entities or relations that are not in G. Such facts constructed by new entities or relations are denoted as unknown/new facts for graph G in this paper [5].

Making sure as many relevant facts from U as possible are added to an existing knowledge graph G is key to enriching G but also preventing violating the inherent subject matter of the knowledge graph. This paper proposes to address this issue by learning embeddings to detect if an unknown fact is relevant to the subject matter of an existing knowledge graph so that only those relevant facts are added to an existing domain-specific knowledge graph.

For a fact $(h', r', t') \in U$, this paper focus on the following cases for this fact to be added to an existing knowledge graph $G = (E, R)$:

1. $r' \in R$, $h' \notin E$, or $t' \notin E$: at least one of the head or tail entities is unknown to the existing knowledge graph G;
2. $r' \notin R$: the relation is unknown to the existing knowledge graph G. This should also include the case where $r' \notin R$, $h' \notin E$, and $t' \notin E$.

For example, in Fig. 1, there are 2 relations between 'Tom Cruise' and 'Mission:Impossible'. There may be a new relation r' between the two entities, and the r' does not exist in R. On the other hand, the title of the new film 'Mission:Impossible 7', the latest movie starring 'Tom Cruise', that will be released in 2023, is a new entity, and it is now not in the entity set E. Therefore, it is necessary to detect whether the unknown facts constructed by new entities or new relations, such as ('Tom Cruise' , '/film/producer/film' ,'Mission:Impossible 7'), are relevant or not to the subject of knowledge graph G before adding them into G.

Manually adding unknown facts into the existing knowledge graph is time-consuming and makes it difficult to validate if the unknown facts should belong to the knowledge graph. This paper seeks to show how to expand an existing domain-specific knowledge graph by adding relevant facts obtained from other knowledge graphs. The contributions of this work are as follows:

1. A novel knowledge graph expansion strategy is proposed, using unknown facts from other knowledge graphs;
2. A new embedding method via the composition of word information is introduced to embed a given fact and to judge if it should be added to the existing knowledge graph;
3. A new large dataset, UFD-303K, is created for this task. The experiments with UFD-303K show that our embedding method is effective for determining whether unknown facts are relevant and whether they should be added to the existing knowledge graph.

2 Related Work

Many KGs have been built but they are still incomplete [13,19] due to missing entities or relations. Therefore, adding new entities or relations had been instrumental to improving the completeness of the existing graphs. This section presents key notation and related work about improving knowledge graph completeness.

[5] 'unknown facts' and 'new facts' are used interchangeably in this paper.

2.1 Knowledge Graph Completion

There has been work about *knowledge graph completion* for 'closed' knowledge graphs using link prediction. The *translation-based models*, proposed in TransE [2], interpret each relation as a translating operation from a head entity to a tail entity for a triple $(h, r, t) \in G$, i.e, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^n$ are used to denote the embeddings for h, r, t, respectively. The learning objective is to minimise the loss of the score function $f_{\mathbf{r}}(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ for all the triples in G. The TransH [21] projects h and t to the relationship-specific hyperplane to allow entities to play different roles in different relationships. The RotatE [15] treats the relation r as a rotating operation from h to t. The RESCAL [11] represents each relation as a full rank matrix and defines the score function as $f_{\mathbf{r}}(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{M}_{\mathbf{r}} \mathbf{t}$. As full rank matrices are prone to over-fitting, recent work turns to make additional assumptions on $\mathbf{M}_{\mathbf{r}}$. For example, DistMult [24] assumes $\mathbf{M}_{\mathbf{r}}$ to be a diagonal matrix, which also utilizes the multi-linear dot product as the scoring function. To better model asymmetric and inverse relations, DistMult was extended by introducing complex-valued embeddings, followed by the proposal of ComplEx [17].

The above models are referred to as shallow models as they cannot capture the potential connection between entities and relationships well. *CNN-based* approaches have been proposed to capture the expressive features, such as, ConvE [4] and ConvKB [10]. ConvE and ConvKB take advantage of CNNs, which improves the expressive power by increasing the interactions between entities and relations.

However, the above models ignore the neighbourhood information in the process of embedding. The *Graph Convolutional Network-based methods* (GCNs) were proposed to address this issue, such as R-GCN [12], which is the first to show that the GCNs can be applied to model relational data. To explicitly and sufficiently model the Semantic Evidence into knowledge embedding, a new method SE-GNN [8] was proposed, where the three-level Semantic Evidence (entity level, relation level and triple-level) are modelled explicitly by the corresponding neighbour pattern and merged sufficiently by the multi-layer aggregation, which contributes to obtaining more extrapolative knowledge representation.

2.2 Knowledge Graph Completion with Unknown Entities

The above methods only focus on a closed knowledge graph, which enriches the knowledge graph by complementing the relationships between entities. However, with the growing volume of data on the Internet, new entities are constantly emerging over time. The entities obtained from the out-of-knowledge graph (called out-of-KG for short in this paper) have been used to enrich the existing knowledge graph [20,22,14,13].

Zhang [20] first proposed a novel method of jointly embedding entities and words into the same continuous vector space, resulting in the prediction of facts containing entities that comes from the out-of-KG. In order to enhance the entities' semantic information, the entity description was used to help with knowledge graph embedding. For example, DKRL [22] employed two encoder methods, continuous Bag-of-words and convolutional neural network (CNN), to embed entity description and then to train models based TransE framework. The ConMask [14] used the CNN attention mechanism to mark which words in the entity description are related to the relation,

and then generating target entity embedding. Shah et al. [13] proposed an open-word knowledge graph completion framework, OWE, based on any pre-trained embedding model, such as TransE. This framework aims to establish a mapping between entity descriptions and their pre-trained embeddings.

The above embedding methods use new entities from the out-of-KG to enrich the existing knowledge graph via word embeddings. However, they are not able to tell whether the new facts constructed with unknown entities are relevant to the existing knowledge graph and hence the new facts may violate the inherent coherence of the existing knowledge graph.

The entity alignment approach [23,9] aims to expand an existing knowledge graph by linking or aligning two entities from two different knowledge graphs that describe the same real-world object. For example, let G_1 and G_2 be two knowledge graphs to be aligned. If an entity e_1 in G_1 corresponds to another entity e_2 in G_2, we call (e_1, e_2) an alignment pair. The task of entity alignment is to find all alignment pairs across two knowledge graphs. Related work also includes extracting facts from texts to enrich an existing knowledge graph [25,6]. However, most of the existing work relies on the pre-defined relations, which are used to guide the extraction of facts from texts.

2.3 Remarks

This paper proposes to expand a domain-specific knowledge graph G by adding unknown facts from another knowledge graph U, where either the entities or the relations from U may be unknown to G. This is significantly different from and more challenging than the above mentioned work. As the unknown facts may contain new entities or relations which did not exist in G, we need to make sure that only the facts that are relevant to the subject matter of G from U can be added into G to avoid potential violation of the inherent subject matter of G.

3 Unknown Fact Detection

This section introduces a novel approach by learning word embeddings for the entities and relations, in order to detect if the unknown facts from U are relevant to the existing knowledge graph so that as many relevant facts from U as possible are added into G.

3.1 Constructing Description for Facts

We use a sub-graph (shown in Fig. 2) from the created UFD-303K dataset as an example. For each entity, a brief description (known as Mention in this paper) was obtained from WiKidata [6]. Each Mention (usually in a phrase or a sentence) provides a brief explanation for its associated entity. For example, the entity 'Titanic' has a Mention with '1997 American romantic disaster film directed by James Cameron'.

A triple (h, r, t) in a knowledge graph is used to denote a fact, i.e., the head entity h has a relation of r with the tail entity t. For example, as shown in Fig. 2, a triple τ

[6] https://www.wikidata.org/wiki/Wikidata:Main_Page

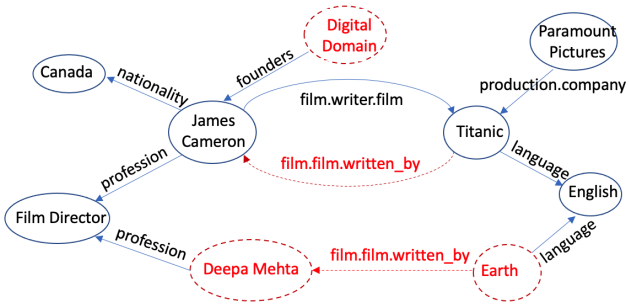

Fig. 2. A sub-graph of the UFD-303K dataset, the new dataset proposed by this work and the more details will be given in Section 4.1. The data use the format of (entity name, mention). For example, the format for entity 'James Cameron' is ('James Cameron', 'Canadian film director') and the format for entity 'Titanic' is ('Titanic', '1997 American romantic disaster film directed by James Cameron'). The new entities and new relations are introduced marked as red.

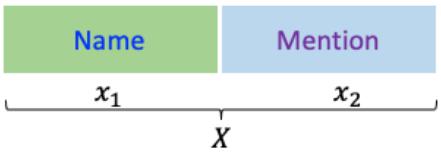

Fig. 3. The structure for representing entity or relation X.

= ('James Cameron', 'film.writer.film', 'Titanic') can be described as $D(\tau)$ = (`James Cameron has a relation of film.writer.film with Titanic.`). In order to enhance the semantic information for the triple, the external explanation information, such as the Mention, can be used to construct a meaningful description for the triple.

In this work, an entity or relation representation consists of 2 components (Name, Mention), as shown in Fig. 3. Here, the Name will refer to either the actual entity or relation and the Mention is usually a phrase or a sentence to interpret an entity. In order to increase the interaction between entities and the relation, we use the sentence template {r`, which is between` h `and` t`.`} to create a corresponding Mention for a relation within the facts. Let X be an entity or a relation, and X contains 2 components as shown in Fig. 3,

$$X = (x_1, x_2), \tag{1}$$

where x_1 is a list of words for Name and x_2 is a list of words for Mention as shown in Fig. 3. For example, given an entity, 'Titanic', x_1 refers to itself, and x_2 denotes its Mention, '1997 American romantic disaster film directed by James Cameron'. Then, the entity 'Titanic' can be described as $X_{Titanic}$ = (`Titanic, 1997 American romantic disaster film directed by James Cameron.`). As a result, given a

triple $\tau = (h, r, t)$, its new description can be represented as:

$$D(\tau) = (X_h; X_r; X_t), \tag{2}$$

where X_h, X_r, and X_t represent the word sequence of h, r, and t, which is initialized by Eq. (1). Finally, for τ = ('James Cameron', 'film.writer.film', 'Titanic'), it will be described by the new description constructed by Eq. (2), i.e., $D(\tau)$ = (`James Cameron, Canadian film director; film.writer.film, which is between James Cameron and Titanic; Titanic, 1997 American romantic disaster film directed by James Cameron.`).

3.2 Unknown Fact Detection Model

BERT [5] is a pre-trained language model based on a multilayer bidirectional Transformer encoder [18]. In this work, we fine-tune the pre-trained BERT for Unknown Fact Detection, known as UFD. The triple description will be concatenated together into a single sequence as input. The first token of every sequence is always a special classification token (`[CLS]`). The final hidden state corresponding to this token is used as the aggregated sequence representation for the classification task. Each component of the triple description is separated with a special token (`[SEP]`). For example, the triple description $D(\tau)$ (Eq. (2)) contain 3 components, X_h, X_r, and X_t, which are separated with a special token (`[SEP]`). Then, the input sequence will be represented as S = (`[CLS]` X_h `[SEP]` X_r `[SEP]` X_t `[SEP]`), and N is the number of tokens in the sequence, $N = |S| = m + n + k + 4$, where m, n, and k denote the number of tokens in X_h, X_r, and X_t, respectively. The pre-trained WordPice embeddings was used to initialize input tokens [5]. The final hidden vector of the special `[CLS]` token as $C_\tau \in \mathbb{R}^H$, where H is the hidden state size in pre-trained BERT. The only new parameters introduced during fine-tuning are classification layer weights $W \in \mathbb{R}^{K \times H}$, where K is the number of labels. Then, we compute a standard classification loss with C_τ and W (Eq. (3)).

3.3 Training

Finally, we train the model by optimizing a cross entropy loss:

$$L = \sum_{\tau \in G \cup G^-} (y_\tau \log(p_\tau) + (1 - y_\tau) \log(1 - p_\tau)), \tag{3}$$

where

$$p_\tau = \text{softmax}(C_\tau W^T), \tag{4}$$

$p_\tau \in \mathbb{R}^2$, which is two probability values $p_\tau = [p_{\tau_1}, p_{\tau_2}]$, indicating relevant probability and irrelevant probability, respectively. If the former p_{τ_1} is greater than the latter p_{τ_2}, the triplet is considered to be related to the existing knowledge graph, and this triple can be introduced into the graph. $y_\tau \in \{0, 1\}$ indicates a negative or positive label, and G^- is the set of negative triples (fake triples) that are constructed by positive triples, and these all 'negative' triples that are irrelevant to the existing knowledge G. This set is obtained

using $G^- = G_1^- \cup G_2^- \cup G_3^- \cup G_4^- \cup G_5^-$ by considering 5 different cases as shown below to replace the entities or relations, for $\forall(h, r, t) \in G$:

$$
\begin{aligned}
G_1^- &= \{(\overline{h_i}, r, t) | 1 \leq i \leq m\}, \\
G_2^- &= \{(h, r, \overline{t_i}) | 1 \leq i \leq m\}, \\
G_3^- &= \{(\overline{h_i}, r, \overline{t_i}) | 1 \leq i \leq m\}, \\
G_4^- &= \{(h, \overline{r_i}, t) | 1 \leq i \leq n\}, \\
G_5^- &= \{(\overline{h_i}, \overline{r_i}, \overline{t_i}) | 1 \leq i \leq n\},
\end{aligned}
$$

where $\overline{h_i} \neq h$, and $\overline{t_i} \neq t$ are random samples from E, and $\overline{r_i} \neq r$ are random samples from R. In our experiments, we set $m = 1$ and $n = 2$.

4 Experiments

In this section, we evaluate our method for the unknown fact detection task. If a new fact is detected as relevant to the existing knowledge, it can be added to the knowledge graph. Otherwise, it should not be added. As such, this unknown fact detection is a binary classification task which is based on a score (Eq. 4) to tell whether a given fact (h, r, t) conforms to the subject matter of the existing domain-specific knowledge graph.

4.1 Datasets

			#Triples	#Entities		#Relations	
				In-KG	Out-of-KG	In-KG	Out-of-KG
UFD-303K	#Train	E - E - E	243998	49391 (E_t)	0	1734 (R_t)	0
	#Test	E - O - E	3234	16516 (E_k)	47918 (E_u)	1282 (R_k)	3028 (R_u)
		E - $\overleftarrow{O}$ - E	8289				
		O - E - O	11829				
		O - E - E	10026				
		E - E - O	7261				
		O - O - O	8145				
		O - $\overleftarrow{O}$ - O	10787				

Table 1. Statistics of the datasets. E_k , E_t, and E_u are the sets of entities, and R_k, R_t , R_u are the set of relations, where $E_k \subset E_t$, $R_k \subset R_t$, $E_t \cap E_u = \emptyset$, and $R_t \cap R_u = \emptyset$. The dataset contains 97309 ($E_t \cup E_u$) entities and 4762 ($R_t \cup R_u$) relations to construct Train and Test (303569 triples in total). In testing set, E indicates the entity or the relation occurs in the training set while O means they do not occur in the training set.

In this work, we create a new dataset called UFD-303K from FreeBase. Table 1 is a summary of the entities and relations in UFD-303K. This dataset is split into 2 parts,

#Train (training set) and #Test (test set), to represent two knowledge graphs and #Train $\cap$ #Test = $\emptyset$. We assume that #Train is an existing domain-specific knowledge graph, and we introduce facts that conform to the subject matter of #Train from another knowledge graph #Test.

As shown in Table 1, the set of known entities is denoted as E_t and $|E_t| = 49391$. The set of known relations is denoted as R_t and $|R_t| =1734$. Both E_t and R_t are used to build the training set using the facts from FreeBase. R_t does not have any two relations that are inverse relationship to each other. The set of 47918 new entities E_u has no common entities with E_t, and the set of 3028 relations R_u has no common relations with R_t.

For the dataset, the #Test (test set) includes 7 kinds of type triples: E-$\overleftarrow{O}$-E, E-O-E, O-E-O, O-E-E, E-E-O, O-$\overleftarrow{O}$-O, O-O-O, where E indicates the entity or the relation that occurs in the training set, while O means they do not occur in the training set. The $\overleftarrow{O}$ indicates that the new relations have reverse relations in training data. For example, as shown in Fig. 2, ('Earth', 'film.film.written_by', 'Deepa Mehta') is an unknown fact, belonging to the O-$\overleftarrow{O}$-O type, where all elements, 'Earth' $\in E_u$, 'film.film.written_by' $\in R_u$, and 'Deepa Mehta' $\in E_u$, are unknown. The relation 'film.film.written_by' in R_u is an inverse relationship of 'film.writer.film' in R_t.

4.2 Hyper-parameter Settings

We choose the pre-trained BERT-Base model with 12 layers, 12 self-attention heads and H = 768 [5]. Following the original BERT, we set the following hyper-parameters in our model to fine-tune: The batch size is 32; The learning rate is set among {5e-5, 3e-5, 0.001}; The N is set among {100, 200, 300, 400}; The number of epochs is set among {2, 5, 10}. Unknown fact detection is a binary classification task and therefore $K = 2$ in this work.

4.3 Unknown Fact Detection Results

In this section, we evaluate the performance of our method on a new dataset, UFD-303K, using accuracy, recall, precision, and F_1. However, the UFD-303K only provides positive triples. A testing set with negative triples are created for the 7 cases as Table 1 shows the number of positive triples. For each positive triple in the testing set, two negative triples are created by replacing head or tail entity with two randomly selected entities from $E_t \cup E_u$. The details are shown in Table 2.

The training set denotes an existing domain knowledge graph and is used to train a model, and the test set simulates the unknown facts obtained from other knowledge graphs. We use the trained model to detect whether these unknown facts are relevant to the subject matter of existing knowledge graph (the training set). Table 3 shows the results of facts detection on 7 test cases. From the results, we observe that the predicted performance of the E-$\overleftarrow{O}$-E case is better than the E-O-E, especially on recall. Also, we observe that the O-$\overleftarrow{O}$-O obtained detection results are better than O-O-O because the relations of the O-$\overleftarrow{O}$-O cases have inverse relationships in the training set. As noted by [16], if the test set contains the inverse relations of the training set, the inverse relations

	$r' \notin R$, and $h', t' \in E$		$r' \in R$, and h' or $t' \notin E$			$r' \notin R$, and $h', t' \notin E$	
7 cases	E - $\overleftarrow{O}$ - E	E - O - E	O - E - O	O - E - E	E - E - O	O - $\overleftarrow{O}$ - O	O - O - O
#Test (positive)	8289	3234	11829	10026	7261	10787	8145
#Test (negative)	16578	6468	23658	20052	14522	21574	16290

Table 2. A testing set with negative triples are created for the 7 cases as Table 1 shows the number of positive triples. For each positive triple in the testing set, two negative triples are created by replacing head or tail entity with two randomly selected entities from $E_t \cup E_u$.

	$r' \notin R$, and $h', t' \in E$		$r' \in R$, and h' or $t' \notin E$			$r' \notin R$, and $h', t' \notin E$	
7 cases	E - $\overleftarrow{O}$ - E	E - O - E	O - E - O	O - E - E	E - E - O	O - $\overleftarrow{O}$ - O	O - O - O
TN	16410	6420	23456	19779	14354	21409	16154
FP	168	48	202	273	168	165	136
FN	840	589	1750	918	630	1942	2328
TP	7449	2645	10079	9108	6631	8845	5817
$Accuracy$	0.9594	0.9343	0.9449	0.9604	0.9633	0.9348	0.8991
$Recall$	0.8986	0.8178	0.8520	0.9084	0.9132	0.8199	0.7141
$Precision$	0.9779	0.9821	0.9803	0.9708	0.9752	0.9816	0.9771
F_1	0.9366	0.8925	0.9117	0.9386	0.9432	0.8935	0.8252

Table 3. Experimental results on unknown fact detection for 7 test cases using true positive (TP), true negative (TN), false positive (FP) and false negative (FN). They are used to calculate accuracy, recall, precision and F_1 score

can help to obtain a good predicting result. This is because the inverse relations may have some common knowledge with the training data. For the result of unknown facts consisting of the new entity (O-E-O, O-E-E, and E-E-O cases), we obtained comparable performance on O-E-E and E-E-O cases, and higher than O-E-O case.

	WN18RR (test set)	YAGO3-10 (test set)
#Test	3134	5000
TN	2571	4269
FP	563	731
TNR	0.8203	0.8538

Table 4. Unknown fact detection on WN18RR and YAGO3-10 test set. TNR (True Negative Rate) indicates the ratio of true negative and total negative, i.e., $TNR = TN/(TN + FP)$

We also need to make sure that our model does not classify the irrelevant triples as relevant to the existing knowledge graph. We use WN18RR [7] [10], and YAGO3-10 [8] [4] as irrelevant triples since they are different KGs from the UFD-303K. We would expect as high TN and TNR as possible as we do not want to add those irrelevant triples from WN18RR and YAGO3-10 into UFD-303K.

The training set of UFD-303K is used to train the model. After the training process, the test set of WN18RR and YAGO3-10 are detected by the trained model. The WN18RR and YAGO3-10 is O-O-O case according to the division in Table 1, where both entities and relations are unknown to UFD-303K. The experimental results are shown in Table 4, from which we observe that TNR is 82.03 % for WN18RR and 85.38 % for YAGO3-10. From Table 3 and 4, our method can detect the relevant triples (shows in Table 3) from U but also is effective to filter out the irrelevant triples (shows in Table 4) .

5 Conclusion and Future Work

In this paper, we propose a novel strategy to expand an existing domain-specific knowledge graph with relevant unknown facts that may come from other knowledge graphs; A new embedding method, UFD, is introduced to learn embeddings for entities and relations to judge unknown triples. This method is validated using a new dataset (UFD-303K), and the experiments show that our embedding method effectively distinguishes the relevance of unknown facts to an existing domain-specific knowledge graph.

Our future work will include canonicalisation for entities and relations [3] to reduce information redundancy, caused by adding new facts into the existing knowledge graph. For example, if the entity 'James Cameron' occurs in an existing domain-specific knowledge graph, adding a new triple with the entity 'James Francis Cameron' into this existing knowledge graph may potentially duplicate the information when 'James Francis Cameron' and 'James Cameron' in fact refer to the same person.

References

1. Abu-Salih, B.: Domain-specific knowledge graphs: A survey. J. Netw. Comput. Appl. **185**, 103076 (2021)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS). pp. 1–9 (2013)
3. Dash, S., Rossiello, G., Mihindukulasooriya, N., Bagchi, S., Gliozzo, A.: Open knowledge graphs canonicalization using variational autoencoders. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021. pp. 10379–10394 (2021)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. pp. 1811–1818 (2018)

[7] WN18RR is a sub-set of a lexical database of English.

[8] YAGO3-10 is a sub-set of large semantic knowledge base, derived from Wikipedia, WordNet, and other data sources.

5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: 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. pp. 4171–4186 (2019)
6. Hwang, E., Lee, J., Yang, T., Patel, D., Zhang, D., McCallum, A.: Event-event relation extraction using probabilistic box embedding. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 235–244. Association for Computational Linguistics (2022)
7. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2021)
8. Li, R., Cao, Y., Zhu, Q., Bi, G., Fang, F., Liu, Y., Li, Q.: How does knowledge graph embedding extrapolate to unseen data: A semantic evidence view. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI. pp. 5781–5791 (2022)
9. Mao, X., Ma, M., Yuan, H., Zhu, J., Wang, Z., Xie, R., Wu, W., Lan, M.: An effective and efficient entity alignment decoding algorithm via third-order tensor isomorphism. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 5888–5898 (2022)
10. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.Q.: A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,Volume 2 (Short Papers). pp. 327–333 (2018)
11. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011. pp. 809–816 (2011)
12. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European semantic web conference. pp. 593–607 (2018)
13. Shah, H., Villmow, J., Ulges, A., Schwanecke, U., Shafait, F.: An open-world extension to knowledge graph completion models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3044–3051 (2019)
14. Shi, B., Weninger, T.: Open-world knowledge graph completion. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1957–1964 (2018)
15. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, ICLR. pp. 1–18 (2019)
16. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality. pp. 57–66 (2015)
17. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning. pp. 2071–2080 (2016)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems. pp. 5998–6008 (2017)

19. Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., Chang, Y.: Structure-augmented text representation learning for efficient knowledge graph completion. In: WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 1737–1748 (2021)
20. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1591–1601 (2014)
21. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. pp. 1112–1119 (2014)
22. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 2659–2665 (2016)
23. Yan, Y., Liu, L., Ban, Y., Jing, B., Tong, H.: Dynamic knowledge graph alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4564–4572 (2021)
24. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. pp. 1–12 (2015)
25. Zhu, H., Lin, Y., Liu, Z., Fu, J., Chua, T., Sun, M.: Graph neural networks with generated parameters for relation extraction. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. pp. 1331–1339 (2019)