Eureka: Neural Insight Learning for Knowledge Graph Reasoning

Alex X. Zhang[1], **Xun Liang**[1]*, **Bo Wu**[1], **Xiangping Zheng**[1],
Sensen Zhang[1], **Yuhui Guo**[1], **Jun Wang**[2], **Xinyao Liu**[3]
[1]Renmin University of China
[2]Swinburne University of Technology
[3]Hong Kong University of Science and Technology
{zhangxuanalex, xliang, wubochn, xpzheng,
sensen0126, yhguo}@ruc.edu.cn
junwang@swin.edu.au liuxinyao575599@163.com

Abstract

The human recognition system has presented the remarkable ability to effortlessly learn novel knowledge from only a few trigger events based on prior knowledge, which is called insight learning. Mimicking such behavior on Knowledge Graph Reasoning (KGR) is an interesting and challenging research problem with many practical applications. Simultaneously, existing works, such as knowledge embedding and few-shot learning models, have been limited to conducting KGR in either "seen-to-seen" or "unseen-to-unseen" scenarios. To this end, we propose a neural insight learning framework named Eureka to bridge the "seen" to "unseen" gap. Eureka is empowered to learn the seen relations with sufficient training triples while providing the flexibility of learning unseen relations given only one trigger without sacrificing its performance on seen relations. Eureka meets our expectation of the model to acquire seen and unseen relations at no extra cost, and eliminate the need to retrain when encountering emerging unseen relations. Experimental results on two real-world datasets demonstrate that the proposed framework also outperforms various state-of-the-art baselines on datasets of both seen and unseen relations.

1 Introduction

Human knowledge provides a formal understanding of the world. Knowledge graphs (KGs) that represent structural relations between entities in the form of (*head entity*, *relation*, *tail entity*) have become an increasingly popular research direction towards cognition and human-level intelligence. These triples of KGs are abbreviated using (h, r, t) in this paper. Over the last few years, the works (Bordes et al., 2013; Sun et al., 2019) on knowledge embedding have achieved impressive results in the knowledge graph reasoning (KGR) task. To successfully learn a set of relations, these methods usually require a large number of training triples and cannot infer missing facts of unseen relations due to the sparse interactions, which are essentially transductive learning processes in terms of the relations. Thus, we categorize this line of knowledge embedding research as "seen-to-seen" methods; i.e., reasoning from seen relations to seen relations. Moreover, the representations of the relations in KGs produced by knowledge embedding models always remain fixed after training. They may be sub-optimal since the real-world large-scale KGs dynamically evolve quickly with new relations emerging every day, rather than staying static.

Suppose that we would like to expand the set of relations that the knowledge embedding models can recognize. We need to collect training triples for the emerging (unseen) relations; i.e., those not in the initial training set, and then restart the aforementioned computationally costly training procedure on the enhanced training set. Not to mention the fact that the model may not perform well when only few training examples are available for the unseen relations (Xiong et al., 2018).

To alleviate the above challenge, some few-shot learning methods (Xiong et al., 2018; Zhang et al., 2020) have been proposed, which can be seen as inductive learning approaches. Their basic ideas are to predict new facts in a meta-learning framework in a setting where only few training triples for each unseen relation are available. We term them as "unseen-to-unseen" methods; i.e., reasoning from unseen relations to unseen relations. This is possible since the meta-learning framework can simulate the unseen relations during meta-training, while they are unobservable in conventional learning schemes. However, their performance will reach a plateau as the number of training examples increases, as illustrated in Figure 1. Moreover, they cannot perform as well as knowledge embedding models on the initial seen relations with suffi-

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 4517–4527
October 12–17, 2022.

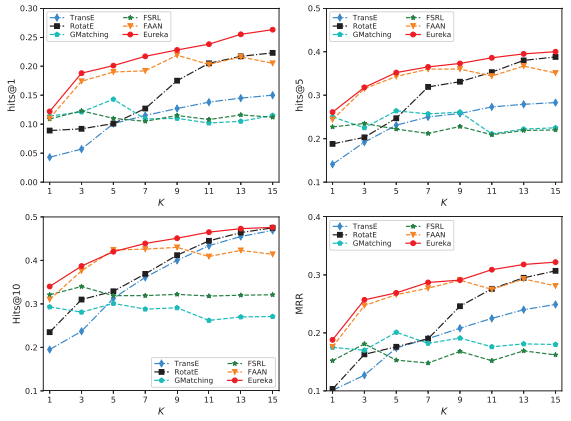

Figure 1: Impact of the size of training triples on NELL dataset. K is the number of training triples per relation. TransE and RotatE are typical knowledge embedding models; GMatching, FSRL, and FAAN are few-shot learning methods; Eureka is our model.

cient training triples, as the few-shot learners are adapted to the common parts of the different meta-tasks and forget the relation-specific information. In other words, the dramatic performance of few-shot learning methods on unseen relations comes at the cost of dysfunction on seen relations with sufficient training examples available. It is also different from the human learning system, where new concepts can be learned from very few examples at no extra cost.

Meanwhile, the human learning system exhibits the remarkable ability to effortlessly discover novel concepts during the "Eureka moment", with only one or few examples as the trigger. For example, a child, having accumulated enough knowledge (seen relations) like "CEO", "President" and so on, can easily learn and generalize the unseen concept of "Leader"from only a single knowledge set like (Gandhi, Leader_of, India) by analogy. Mimicking this behavior in artificial reasoning systems is an interesting and very challenging research problem with many practical advantages, such as developing real-time knowledge reasoning systems for downstream applications, such as language models.

Motivated by the limitations of knowledge embedding methods and few-shot KGR models, we mimic human insight learning modes in machine learning and propose a neural insight learning framework termed Eureka for KGR tasks. The whole structure of Eureka is illustrated in Figure 2. Eureka aims to tackle the problem knowledge embedding methods and few-shot KGR models encountered, under a more realistic setting, where a large set of training triples are assumed to exist for seen relations; and using these data as the sole input, we want to develop a KGR model that, is not only capable of recognizing these seen relations, but also learning unseen relations from only one training example (provided only at the testing time) without sacrificing the performance on seen relations or requiring to be retrained). We also devise a cross-domain attention (CDA) network to model the semantic interactions and bridge the gap between unseen and seen relations; for example, the unseen relation "Leaderof" has semantic similarity with the seen relations "Presidentof" and "CEOof" and modeling such similarity will help to make up for the lack of training information of unseen relations and represent the unseen relation more accurately to some extent.

Compared to prior approaches, we believe that Eureka resembles more closely the human learning behavior (w.r.t. how it learns novel concepts). Eureka is also more suitable in the real-world scenario where unseen relations do not emerge one by one but may emerge simultaneously as a set, with only few triples available for each new unseen relation.

To summarize, the contributions are as follows:

- To the best of our knowledge, Eureka is the first neural insight learning framework for KGR by mimicking human learning behaviors, which can efficiently learn new unseen relations based on one given trigger and the learned seen relations at no extra cost.
- In contrast to the previous works, Eureka bridges the "seen" to "unseen" gap with the CDA networks and provides the flexibility of inferring missing facts for both seen and unseen relations in a unified protocol.
- The extensive experimental results on two real-world datasets show the superiority of Eureka compared with the state-of-the-art baselines on both seen and unseen relations.

2 Related Work

The neural insight learning framework draws on the previous research in knowledge embedding and meta-learning.

2.1 Knowledge Embedding

Knowledge embedding aims to model multi-relational data and automatically inferring missing facts in knowledge graphs. Many of them encode both entities and relations into a continuous low dimensional vector space. TransE (Bordes et al., 2013) is a classic work that encodes both entities

and relations into a 1-D vector space. DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) attempt to mine latent semantics to benefit their KG embeddings. CoKE (Wang et al., 2019) presents a novel paradigm that takes into account KGs' contextual nature and learns contextualized knowledge graph embedding based on the transformer. There are also other effective models like ConvE (Dettmers et al., 2018a), Rotate (Sun et al., 2019) and UniKER (Cheng et al., 2021). These embedding-based models rely heavily on extensive collections of training instances, and they are not able to deal with sparse triples, as presented in (Xiong et al., 2018).

2.2 Meta-Learning

Meta-learning, commonly known as *learning to learn* (Lake et al., 2015), refers to the process of improving the learning algorithm itself over multiple learning episodes. Contrary to conventional machine learning approaches where tasks are handled from scratch using a fixed learning algorithm, meta-learning provides an opportunity to dynamically adapt to new tasks with the learned algorithm.

One line of meta-learning research, which is closely related to our work, is few-shot learning. Few-shot learning methods seek to learn novel concepts with only a small number of labeled examples. Recent deep learning based few-shot learning algorithms can be classified into three groups. The first group is **model-based approaches**, which depend on a specially designed part, like memory, to quickly optimize the model parameters given few-shot training instances. MetaNet (Munkhdalai and Yu, 2017), a typical model-based approach, learns meta knowledge across tasks and generalizes rapidly via its fast parameterization. The second group is **metric-based approaches**, which try to learn a generalizable metric and the corresponding matching functions among a set of training instances. For example, prototypical networks (Snell et al., 2017) classify each instance by calculating its similarity to the prototype representation of each class, whose idea is similar to some nearest neighbor algorithms. GMatching (Xiong et al., 2018), FSRL (Zhang et al., 2020), and FAAN (Sheng et al., 2020) can also be considered as metric-based approaches. The third group is **optimization-based approaches**, which aim to learn faster by changing the optimization methods on few-shot reference instances. One example is the model-agnostic meta-learning (MAML) (Finn et al., 2017), which first proposed a framework of parameter updating for a task-specific learner and performing meta-optimization across tasks by using the above updated parameters. MetaR (Chen et al., 2019), MetaP (Jiang et al., 2021) and GANA (Niu et al., 2021) can be regarded as optimization-based approaches for few-shot KGR.

3 Preliminaries

In this section, we formally describe neural insight learning in the KGR scenario and leave technical details to the next section. According to the Gestalt theory of learning, insight learning occurs spontaneously when people discover new knowledge within their prior knowledge as a result of reasoning or problem-solving processes that reorganize or restructure that knowledge (Ash et al., 2012). In other words, there are two key points about machine insight learning in the KGR scenario. One is that the model could learn new unseen relations with a trigger based on the prior seen relations it learned, the other is that the model should achieve good performance on both seen and unseen relations as human beings do.

We first present the definition of KGR, then formalize neural insight learning in the KGR scenario. The difference between Eureka and previous relevant learning theories such as meta-learning (Hospedales et al., 2021), transfer learning (Pan and Yang, 2010), one-shot learning (Wang et al., 2020), and one-pass learning (Zhou et al., 2016) are also discussed.

Definition 1. *Given an incomplete knowledge graph $\mathcal{G}$ presented as $\{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and relation set, the KG reasoning task aims at finding a set of missing triples; i.e., predicting relations r between two existing entities: $(h, ?, t)$, or predicting the tail entity t given the head entity and the relation: $(h, r, ?)$, or predicting the head entity h given the relation and the tail entity: $(?, r, t)$.*

In practical experimental settings, it is reasonable to predict the tail entity to test a KGR model. Usually, we aim to rank the triples with the true tail entity higher than those with the false tail entities.

Eureka consists of two stages: prior knowledge learning and trigger learning. The former mimics where the human beings acquire basic prior knowledge. The latter aims at learning new knowledge given only one trigger as the training example,

based on the prior knowledge they gained. Taking the KGR task as an example, the objective of prior knowledge learning can be presented as follows:

$$\min_{\theta} \mathbb{E}\left[\sum_{(h,r,t)\in\mathcal{G}_a\cup\bar{\mathcal{G}}_a} \ell_\theta\left(h,r,t \mid \mathcal{G}_a,\bar{\mathcal{G}}_a\right)\right], \quad (1)$$

where $\mathcal{G}_a$ is a KG full of triples containing seen relations and $\bar{\mathcal{G}}_a$ is a set of invalid triples generated by polluting the tail entities of valid triples in $\mathcal{G}_a$; $\ell_\theta\left(h,r,t \mid \mathcal{G}_a,\bar{\mathcal{G}}_a\right)$ is an arbitrary ranking loss function, and θ is the parameter of Eureka including the embeddings of entities and seen relations. This stage is very similar to conventional knowledge embedding models.

Trigger learning imitates the human ability of fast learning new knowledge based on their prior knowledge after being stimulated by a new phenomenon. We sample one new training triple episodically as a trigger for the model to acquire the unseen relation. The objective of trigger learning is defined as:

$$\min_{\varphi} \mathbb{E}_{D_{r'}}\left[\sum_{(h_i,r',t_i)\in T^{\text{test}}_{r'}\cup\bar{\mathcal{G}}_b} \frac{\ell_\varphi\left(h_i,r',t_i \mid \theta^*,\bar{\mathcal{G}}_b,T^{\text{train}}_{r'}\right)}{\left|T^{\text{test}}_{r'}\right|}\right], \quad (2)$$

where r' is a unseen relation and $D_{r'} = \{T^{\text{train}}_{r'}, T^{\text{test}}_{r'}\}$ is sampled from $\mathcal{G}_b \cup \bar{\mathcal{G}}_b$; $\mathcal{G}_b$ is a KG of unseen relations and $\bar{\mathcal{G}}_b$ is a set of invalid triples generated by polluting tail entities of triples in $\mathcal{G}_b$. The relations in $\mathcal{G}_b$ and $\mathcal{G}_a$ are disjointed; i.e., the relations in $\mathcal{G}_b$ are the unseen relations for the model trained on $\mathcal{G}_a$. Each $T^{\text{train}}_{r'}$ contains only one training triple (h_0, r', t_0) as a trigger. The $T^{\text{test}}_{r'} = \{(h_i, r', t_i)\}$ is comprised of the testing triples of r' with ground-truth tail entity and the invalid tail entities for each query (h_i, r'). θ^* is the learned optimal parameter of prior knowledge learning stage. $\ell_\varphi\left(h_i,r',t_i \mid \theta^*,\bar{\mathcal{G}}_b,T^{\text{train}}_{r'}\right)$ is the loss function of trigger learning stage and φ is the parameter to learn.

There are also some learning theories that neural insight learning looks a bit similar to. We list them as follows:

1) **Meta-Learning** (Hospedales et al., 2021): Meta-learning, also termed as *learning to learn*, refers to the paradigm of improving a learning algorithm given the experience of multiple learning episodes.

2) **Transfer Learning** (Pan and Yang, 2010): Transfer learning focuses on storing knowledge gained from a source domain and applying it to a different but related target domain.

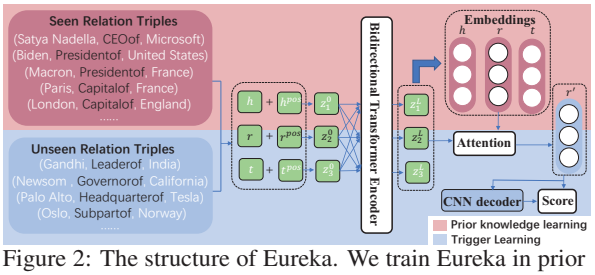

Figure 2: The structure of Eureka. We train Eureka in prior knowledge learning and trigger learning stages. Specifically, Eureka acquires the embeddings of entities and seen relations in the prior knowledge learning stage through a bidirectional transformer encoder and CNN-based decoder. With the learned representations of seen relations, Eureka then learns the unseen relations efficiently through a cross-domain attention network. Note that Eureka shares the same encoder and decoder in the two stages.

3) **One-Shot Learning/Few-Shot Learning** (Wang et al., 2020): Whereas most machine learning algorithms require training on hundreds or thousands of samples and very large datasets, one-shot/few-shot learning aims to learn information about object categories from only one/few training samples.

4) **One-Pass Learning** (Zhou et al., 2016; Hou and Zhou, 2018): One-pass learning is proposed to predict new coming samples' label and update the model based on the prediction, where coming samples are used only once and never stored.

It is obvious that we adopt a meta-learning framework to formalize and implement Eureka. However, we use a more strict setting where only one training example can be seen and the trained model is capable of performing well on the unseen relations without sacrificing its performance on the seen relations. Our neural insight learning can be termed as *trigger to learn* as the meta-learning is also known as *learning to learn*. Eureka can be also seen as a subset of transfer learning with more restrictions since it could learn unseen relation representations in the target domain based on the prior knowledge acquired from the source domain. Although one-shot learning and Eureka both use one training instance as input, the goal of one-shot learning is to learn the common knowledge across the tasks and forget the relation-specific prior knowledge while Eureka wants to remember prior knowledge. One pass learning basically refers to learning by seeing the data once. So if we learn by taking data as a single instance, mini-batch, or large batches as long as we go over them once (epoch=1), they qualify as one pass learning. However, our neural insight learning only takes one training example as input and could also satisfy the evolving streaming data

nature in the real world.

4 Neural Insight Learning

4.1 Overall Architecture

To fully mimic human insight learning behaviors, Eureka is built on a two-stage learning framework. Figure 2 shows an overview of Eureka. In the prior knowledge learning stage, Eureka takes triples sampled from the $\mathcal{G}_a$ as input, similar to the knowledge embedding models (Wang et al., 2019) based on deep neural networks. We use a bidirectional transformer encoder and a CNN-based decoder to learn the dynamic embeddings of entities and relations of $\mathcal{G}_a$ and the parameters of the encoder-decoder model. The second stage, termed trigger learning, is designed to learn new knowledge with only one training example as a trigger based on the prior knowledge Eureka gained. Thus, we adopt a meta-learning framework to implement trigger learning. During meta-training step, the trigger sampled from $\mathcal{G}_b$ only contains one training triple (h_0, r', t_0) for each unseen relation r'. The representations of r' can be produced by the trained encoder-decoder model. However, only a single training example cannot guarantee the accurate representations of r' as the previous research on knowledge embedding (Xiong et al., 2018) claimed. We use a CDA network to make up for the lack of training information of unseen relations. The CDA mechanism incorporates the embeddings of the relevant seen relations with the calibrated embeddings of r' to acquire accurate representations of unseen relations. The same CNN-based decoder is then applied to judge whether the query is true or not with the given trigger during the meta-testing step.

4.2 Prior Knowledge Learning

We expect that Eureka to learn new relations with only one trigger and preserves good performance on seen relations; i.e. the embeddings of entities and relations should ideally evolve with the newly added triggers. Thus, we need an encoder to produce dynamic embeddings for every component given its graph contexts. The pre-trained language models such as BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), have recently made great progress in learning contextualized word embeddings with transformers (Vaswani et al., 2017). Inspired by these techniques, we employ a bidirectional transformer encoder to model graph contexts and produce the dynamic contextual embeddings of entities and seen relations. In contrast to previous sequential left-to-right or right-to-left encoding strategies for the elements in a triple (Guo et al., 2019), our model applies a multi-head self-attention mechanism to model context information, which allows each element to pay attention to all elements in the sequence. Given a triple (h, r, t), we obtain a sequence $X = (x_1, x_2, x_3)$, where x_1, x_2, x_3 represent the head entity, relation, and tail entity, respectively. Since we aim to acquire the well-learned embeddings among triples, we follow the settings of most knowledge embedding models; i.e., modeling the semantic knowledge among triples instead of paths and walks outstretching from the entity and relation like CoKE (Wang et al., 2019). For each element x_i in X, the input of our transformer encoder is constructed as:

$$\mathbf{m}_i^0 = \mathbf{x}_i^{\text{ele}} + \mathbf{x}_i^{\text{pos}}, \tag{3}$$

where $\mathbf{x}_i^{\text{ele}}$ and $\mathbf{x}_i^{\text{pos}}$ denote the element embedding and the position embedding, respectively. The former is used to identify the current element, and the latter represents its position in the sequence. We feed the input vectors into a stack of L transformer blocks to encode X:

$$\mathbf{m}_i^l = \text{Transformer}\left(\mathbf{m}_i^{l-1}\right), l = 1, 2, \cdots, L, \tag{4}$$

where $\mathbf{m}_i^l$ is the hidden state of x_i after l-th layer.

Then we are allowed to obtain a sequence of three encoded vectors $(\mathbf{m}_1^L, \mathbf{m}_2^L, \mathbf{m}_3^L)$ for the triple (h, r, t). (Fan et al., 2020) indicates that a given layer of transformers can only access low-level representations and it restricts the model from fully exploiting the sequential nature of the input. Transformers also have challenges in modeling hierarchical structures (Hahn, 2020). Thus we adopt CNN as the decoder in Eureka since CNN can explore the high-level representations and model the hierarchical structures of the interactions between entities and relations by nonlinear feature learning. The score function for the triple is designed as

$$f(h, r, t) = \text{pooling}(\sigma([\mathbf{m}_1^L, \mathbf{m}_2^L, \mathbf{m}_3^L] * \omega))^\top \mathbf{u}, \tag{5}$$

where pooling is a max-pooling operator and σ denotes an activation function. $[\mathbf{m}_1^L, \mathbf{m}_2^L, \mathbf{m}_3^L] \in \mathbb{R}^{d \times 3}$ is a matrix generated by stacking $\mathbf{m}_1^L$, $\mathbf{m}_2^L$ and $\mathbf{m}_3^L$, and d is the embedding size; $*$ denotes a convolution operator; $\omega \in \mathbb{R}^{s \times 3}$ is a set of filters with s being the number of filters; and $\mathbf{u} \in \mathbb{R}^d$ denotes a weight vector. Unlike ConvE (Dettmers et al., 2018b), the decoder of Eureka stacks ele-

ments of the triple instead of concatenating the relation and entities in the triple. The stack operation for feature maps, which is fed to a Conv2D network, increases the learning ability of latent features. The pooling operator is empowered to capture the most important semantic feature from each feature map and reduces the number of weight parameters.

Depending on the scoring function $f(h,r,t)$, we adopt a binary cross-entropy (BCE) loss as (Nguyen et al., 2018). It applies a softplus (Glorot et al., 2011) to the score of each (positive or negative) triple and uses the cross-entropy between the resulting likelihood and the triple's label as loss:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{G}_a^*} \log\left(1+\exp\left(-l_{(h,r,t)}\cdot f(h,r,t)\right)\right), \tag{6}$$

in which, $l_{(h,r,t)} = \begin{cases} 1 & \text{for } (h,r,t)\in\mathcal{G}_a \\ -1 & \text{for } (h,r,t)\in\bar{\mathcal{G}}_a \end{cases}$ where $\mathcal{G}_a$ and $\bar{\mathcal{G}}_a$ are collections of valid and invalid triples; $\mathcal{G}_a^* = \{\mathcal{G}_a \cup \bar{\mathcal{G}}_a\}$; $\bar{\mathcal{G}}_a$ is generated by corrupting tail entities of valid triples in $\mathcal{G}_a$.

4.3 Trigger Learning

In this stage, we explore how to learn an unseen relation representation with only one trigger. Since we aim to predict new facts on both seen and unseen relations, Eureka shares the same encoder and decoder across the two training stages. In other words, the representation vectors $(\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3)$ for a trigger (h_0, r', t_0) is output through Equation 3 and 4. Note that Eureka cannot see r' in the prior knowledge learning stage. Thus, r' is randomly initialized before being fed to the transformer encoder. In the dynamic scenario, the representations of new relations cannot be sufficiently trained on knowledge embedding models given limited training triples and thus the embeddings of r' output by the transformer encoder can be not accurate. To empower Eureka to adapt well to unseen relations and learn from prior knowledge, we borrow the idea of transfer learning and design a CDA to model the semantic interactions between unseen and seen relations and bridge the "seen-to-unseen" gap.

In the "seen-to-unseen" reasoning scenario, we adopt a meta-learning framework to implement trigger learning. For a specific trigger (h_0, r', t_0), the CDA network can be presented as $\text{CDA}(\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3, \mathbf{W}_a)$, where $\mathbf{m}'_1, \mathbf{m}'_2, \mathbf{m}'_3$ is the output embeddings by the transformer encoder and $\mathbf{W}_a$ is a set of all seen relations' embeddings generated in the prior knowledge learning stage. The semantic interactions between seen relations and r' is modeled as follows:

$$\mathbf{w}'_i = \sum_{t=1}^{K} \text{ATTENTION}\left(\frac{\mathbf{m}'_2}{\|\mathbf{m}'_2\|}, k_t\right)\cdot\frac{\mathbf{w}_a^t}{\|\mathbf{w}_a^t\|}, \tag{7}$$

where $\left\{k_t\in\mathbb{R}^d\right\}_1^K$ is a set of learnable keys (one per seen relation) used for indexing the memory. $\mathbf{w}_a^t$ is the t-th row vector of the prior knowledge matrix $\mathbf{W}_a$, which represents the embedding of the seen relation r_t. Then the final representation vector of unseen relation r' can be computed as:

$$\mathbf{w}_i^* = \lambda_1 \odot \mathbf{m}'_2 + \lambda_2 \odot \mathbf{w}'_i, \tag{8}$$

where λ_1 and $\lambda_2 \in \mathbb{R}^d$ are weight matrices to learn; $\odot$ is the Hadamard product. Through the above process, our model is able to explicitly leverage the acquired semantic knowledge from the seen relations. Note that we only consider a closed set of entities and an open set of relations in this scenario. To be more specific, the testing triples share the same entities with the training triples while the relations of testing triples are disjointed from the relations of training triples.

Since we aim to predict the missing links in a unified protocol for both seen and unseen relations, we adopt the same decoder and loss function as the prior knowledge learning does. To be more specific, we replace $[\mathbf{m}_1^L, \mathbf{m}_2^L, \mathbf{m}_3^L]$ in Equation 5 with $[\mathbf{m}'_1, \mathbf{w}_i^*, \mathbf{m}'_3]$ to constructed trigger learning's score function and adopt the same loss function as Equation 6 implemented in $\mathcal{G}_b$.

5 Experiments

We investigate three issues with Eureka: (1) Could Eureka improve the performance of KGR on unseen relations at no extra cost? (2) Is each component in Eureka necessary? (3) How CDA works? To answer these questions, we conduct experiments on two KG datasets and systematically analyze the corresponding results.

5.1 Datasets

We use two public datasets for experiments, NELL and Wiki, which are released by (Xiong et al., 2018). NELL is derived from a system that can continuously acquire diverse structured knowledge (Mitchell et al., 2015). Wiki is constructed based on Wikidata (Vrandečić and Krötzsch, 2014). The dataset statistics are shown in Table 1. We randomly select a number of relations with more

Table 1: Statistics of the Datasets. # Entities, # Relations and # Triples denote the number of unique entities, relations and triples in the datasets, respectively.

Dataset	#Entities	# Relations	# Triples
NELL	68,545	358	181,109
Wiki	4,838,244	822	5,859,240

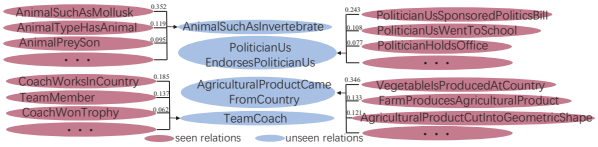

Figure 3: The most contributive seen relations in different tasks for unseen relations in NELL. Here we present top 3 seen relations and their attention weights.

than 1000 triples as seen relations, and the relations less than 500 but more than 50 triples as unseen relations. There are 67 and 183 unseen relations in NELL and Wiki data, respectively. Besides, we use 51/5/11 unseen relations for training/validation/testing in NELL and the division is set to 133/16/34 in Wiki during the few-shot learning stage. The datasets used in the prior knowledge learning stage are constructed by assigning triples of each seen relation in the ratio of 7:1:2 to the training/validation/testing set.

5.2 Baseline Methods

We select two kinds of baseline methods including knowledge embedding models and few-shot learning models: 1) **Knowledge Embedding Models.** This line of research models multi-relational structure in KGs and encodes both entities and relations into a continuous low dimensional vector space. We consider four widely used baselines as follows: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019) and UniKER(Cheng et al., 2021). For fair comparison, we consider the transformer-based model termed CoKE (Wang et al., 2019) as the baseline method. All training triples of the seen relations, as well as the trigger triples of unseen relations, are used during training. 2) **Few-Shot Learning Models.** These models concentrate on predicting new facts in KGs with only few-shot reference triples. We select four typical models; i.e., GMatching (Xiong et al., 2018), MetaR (Chen et al., 2019), FSRL (Zhang et al., 2020), FAAN (Sheng et al., 2020) and GANA (Niu et al., 2021). Note that we adopt one-shot setting for these methods for fair comparison since Eureka only gets one available trigger.

5.3 Implementation Details

At the prior knowledge learning stage, Eureka is trained, evaluated, and tested solely on the triples of seen relations. We vary the number of transformer layers in $\{2, 3, 4\}$, the number of transformer heads in $\{2, 3, 4, 5, 6\}$, the head size in $\{128, 256, 512, 1024\}$, the number of filters in $\{128, 256, 512, 1024\}$. We also apply dropout to the transformer layers with the rate in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ to avoid over-fitting. For parameter updates, we use Adam (Kingma and Ba, 2015) with the initial learning rate of 0.005 and we have the learning rate decay 5 times for each 50k training step. At the trigger learning stage, Eureka is trained, evaluated, and tested solely on the triples of unseen relations with one training data available. We also use Adam (Kingma and Ba, 2015) with the initial learning rate of 0.001 to optimize our model. Then we have the learning rate decay 5 times for each 10k training step. We evaluate all methods for every 10k training steps, and select the best models leading to the highest Hits@10 on the validation set within 500k steps as (Xiong et al., 2018); and then we get the optimal hyper-parameters, and report the final results on the testing set.

5.4 Evaluation Metrics

Following the widely used evaluation metrics in KGR tasks (Bordes et al., 2013), we adopt Hits@k, i.e., the proportion of correct entities ranked in the top k, and MRR, i.e., the mean reciprocal rank, to evaluate the overall performance of Eureka. Generally, the higher MRR and Hits@k indicate the better performance. k is set to 1, 5, and 10.

5.5 Results

We first evaluate our model on unseen relations, which predicts new facts on a query set where no seen relations are included. As shown in Table 2, Eureka shows a significant improvement margin over both knowledge embedding models and few-shot learning models. Taking the best performing few-shot learning model on NELL (FAAN) as an example, the improvement (%) of Eureka on testing MRR and Hits@10 are 23.9% and 14.5%, respectively. It, to some extent, confirms the effectiveness of the idea that the unseen relation representation can benefit from semantic interactions of seen relations and even be composed as a linear combination of the similar seen relation embeddings.

We also perform a KGR experiment on the seen

Table 2: The overall results of seen and unseen relations on testing datasets. We present the best baseline results by underline and highlight the best results of all methods in bold. For coping with the space limitation, we shortened the names of some evaluation metrics, e.g., Hits@10 is shortened as H@10. The notations are the same in all tables.

	NELL								Wiki							
	Seen relations				Unseen relations				Seen relations				Unseen relations			
Model	MRR	H@10	H@5	H@1	MRR	H@10	H@5	H@1	MRR	H@10	H@5	H@1	MRR	H@10	H@5	H@1
TransE	.254	.475	.284	.158	.101	.195	.141	.043	305	.464	.378	.267	.033	.052	.041	.022
DistMult	.235	.426	.256	.147	.095	.177	.125	.065	.285	424	.357	.221	.050	.102	.069	.019
ComplEx	.289	.453	.285	.215	.131	.223	.156	.086	.324	.468	.381	.295	.069	.122	.089	.036
RotatE	.314	**.482**	.392	.226	.103	.235	.188	.089	.337	.481	.408	.299	.055	.083	.055	.033
UniKER	.299	.463	.390	.232	.107	.230	.176	.075	.321	.480	.373	.290	.051	.101	.053	.039
CoKE	.289	.466	.384	.235	.082	.155	.092	.037	.322	.477	.395	.280	.042	.051	.032	.024
GMatching	.181	.295	.261	.131	.175	.293	.250	.114	.269	.388	.341	.205	.201	.335	.272	.123
MetaR	.231	.384	.291	.175	.172	295	.236	.096	.324	.420	.390	.281	.193	.291	.237	.155
FSRL	.181	.322	.219	.103	.152	.321	.227	.109	.161	.298	.212	.103	.197	.318	.255	.119
FAAN	.268	.421	.357	.202	.176	.310	.244	.110	.321	.466	.395	.281	.239	.380	.309	.170
GANA	.242	.389	.299	.193	.176	.317	.247	.112	.320	.435	.388	.270	.223	.370	.262	.155
Eureka (Ours)	**.332**	**.482**	**.407**	**.267**	**.188**	**.340**	**.261**	**.122**	**.339**	**.498**	**.412**	**.305**	**.257**	**.397**	**.323**	**.192**

Table 3: Results of model variants on unseen relations of NELL. The best results are highlighted in bold.

Model	MRR	H@10	H@5	H@1
AS_1.1.1	.103	.201	.147	.055
AS_1.1.2	.114	.228	.162	.061
AS_1.2.1	.108	.249	.151	.058
AS_1.2.2	.135	.261	.197	.105
AS_2	.179	.330	.250	.118
AS_3.1	.157	.295	.231	.112
AS_3.2	.160	.304	.227	.115
Eureka	**.188**	**.340**	**.261**	**.122**

relation dataset. The experimental results show that Eureka surpasses the prior state-of-the-art few-shot learning approaches, which demonstrates that Eureka is able to remember the original knowledge learned from seen relations when having acquired unseen relations. Moreover, Eureka is still competitive compared with best-performed knowledge embedding models on both NELL and Wiki. It is also worth noting that the performance of the few-shot learning model does not improve significantly with sufficient training examples as other models do. It indicates that the learning capacity of the few-shot learning approaches is limited even though the number of their training examples increases.

Thus, we have so far answered the first question; i.e., Eureka can be well adapted into the KGR task of unseen relations and outperform both embedding models and few-shot learning models by incorporating the knowledge learned from the seen relation embeddings without sacrificing performance on seen relations.

5.6 Ablation Study

In this section, we inspect effectiveness of the model components. Experimental results of model variants is shown in Table 3 :

1) We verify the significance of the transformer encoder. We remove the transformer encoder and replace it with TransE and RotatE, respectively. We conduct two group of experiments with TransE as an alternative of the transformer encoder; i.e., AS_1.1.1 and AS_1.1.2. In AS_1.1.1, the entity and relation embeddings produced in the prior knowledge learning stage remain static when used in the trigger learning stage. In AS_1.1.2, these entity and relation embeddings are fine-tuned in the trigger learning stage. AS_1.2.1 and AS_1.2.2 share the same settings with RotatE as an alternative of the transformer encoder. Experimental results demonstrate that the transformer encoder is an essential component of our model due to its ability to model the semantic interactions between entities and relations and produce the dynamic embeddings. By comparing the results of AS_1.1.1 with AS_1.1.2, we can find that it is better to allow dynamic embeddings rather than static embeddings for the encoder of Eureka since Eureka absorbs new knowledge (unseen relation triples) in the trigger learning stage and could adjust embeddings of seen relation triples to adapt the unseen domain. The comparison between AS_1.2.1 and AS_1.2.2 also leads to the same conclusion.

2) We analyze the contribution of the CNN decoder. We remove the CNN decoder and this makes Eureka in the prior training stage degenerate into a solely transformer-based model similar to CoKE. Experimental results indicate that the model per-

formance can slightly benefit from the CNN-based decoder through modeling the high-level representations and the hierarchical structures of the KGs.

3) **(AS_3)** We evaluate the effectiveness of CDA. We conduct two experiments denoted as AS_3.1, AS_3.2. In AS_3.1, we simply remove CDA to inspect the contribution of seen relations; in AS_3.2, we replace CDA with an average pooling operation on seen relations to evaluate the effectiveness of the attention mechanism. Experimental results show that both unseen relations and attention mechanism are important and contribute consistent improvements to Eureka.

5.7 Case Study for CDA

We investigate how CDA works. Since CDA adopts an attention mechanism to model the interactions between seen and unseen relations and represents unseen relations with the acquired knowledge from seen relations. We randomly select four unseen relations and present their most relevant seen relations according to the attention score in CDA as shown in Figure 3. We could find that the relevant seen relations have semantic similarity with their corresponding unseen relations. For example, for the unseen relation *AnimalSuchAsInvertebrate*, the top 3 of the selected seen relations are *AnimalSuchAsMollusk*, *AnimalTypeHasAnimal*, *AnimalPreySon*. Obviously, most mollusks belong to invertebrate animals, which confirms the effectiveness of CDA and our assumption that unseen relations can benefit from relevant semantic interactions.

5.8 Discussions

We summarize the answers to our three research issues: (1) Eureka surpasses both embedding models and few-shot learning models on seen and unseen relations. Eureka achieves better performance on unseen relations without sacrificing its performance on seen relations. Note that as the number of triggers increases, Eureka still outperforms other baselines, which is shown in Figure 1. (2) The ablation study demonstrates the effectiveness of each model variant of Eureka, i.e., the transformer encoder for allowing dynamic embeddings; CNN decoder for modeling the high-level representations and the hierarchical structures of the KGs; CDA for modeling semantic interactions between seen and unseen relations. (3) The case study shows why CDA brings a dramatic rise for experimental results, i.e., CDA assigns varying attention weights to different seen relations and selects the most relevant seen relations to represent each unseen relation more accurately.

6 Conclusion

In this work, we present a neural insight learning framework (Eureka), which mimics human insight learning modes to bridge the "seen" to "unseen" gap in the KGR tasks. We train Eureka in prior knowledge learning and trigger learning stages. Specifically, Eureka acquires the representations of entities and seen relations in the prior knowledge learning stage, and then learns the unseen relations efficiently through a CDA network with the incorporation of the embeddings of seen relations. Eureka meets our expectation of the model to not only have good performance on both relation types but also eliminate the need to retrain the original training datasets. The experimental results demonstrate that our model outperforms the state-of-the-art baselines on datasets of both seen and unseen relations. The case studies confirm the CDA network is empowered to select relevant seen relations to better represent unseen relations. We plan to investigate enhancing CDA with relations' text descriptions as future directions of this work.

Acknowledgements

This work was supported by National Natural Science Foundation of China (62072463, 71531012), Research Seed Funds of School of Interdisciplinary Studies of Renmin University of China, National Social Science Foundation of China (18ZDA309), and Opening Project of State Key Laboratory of Digital Publishing Technology of Founder Group. Xun Liang is the corresponding author of this paper.

References

Ivan K. Ash, Benjamin D. Jee, and Jennifer Wiley. 2012. Investigating insight as sudden learning. *J. Probl. Solving*, (2).

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning

for few-shot link prediction in knowledge graphs. 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)*.

Kewei Cheng, Ziqing Yang, Ming Zhang, and Yizhou Sun. 2021. Uniker: A unified framework for combining embedding and definite horn rule reasoning for knowledge graph inference. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018a. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018b. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Angela Fan, Thibaut Lavril, Edouard Grave, Armand Joulin, and Sainbayar Sukhbaatar. 2020. Addressing some limitations of transformers with feedback memory. *arXiv preprint arXiv:2002.09402*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, JMLR Proceedings.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*.

Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chenping Hou and Zhi-Hua Zhou. 2018. One-pass learning with incremental and decremental features. *IEEE Trans. Pattern Anal. Mach. Intell.*, (11).

Zhiyi Jiang, Jianliang Gao, and Xinqi Lv. 2021. Metap: Meta pattern learning for one-shot knowledge graph completion. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*.

Diederik P. Kingma and Jimmy Ba. 2015. 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*.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, (6266).

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. 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)*.

Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with

gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, (10).

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, (8).

Jiawei Sheng, Shu Guo, Zhenyu Chen, Juwei Yue, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2020. Adaptive Attentional Network for Few-Shot Knowledge Graph Completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, JMLR Workshop and Conference Proceedings.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, (10).

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, (3).

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. 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*.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. Few-shot knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*.

Zhaoze Zhou, Wei-Shi Zheng, Jianfang Hu, Yong Xu, and Jane You. 2016. One-pass online learning: A local approach. *Pattern Recognit.*