

Knowledge Graph Embeddings for Causal Relation Prediction

Aamod Khatiwada^{1,2,†}, Sola Shirai^{1,3,†}, Kavitha Srinivas¹ and Otkie Hassanzadeh¹

¹IBM Research, Yorktown Heights, NY, USA

²Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA

³Rensselaer Polytechnic Institute, Troy, NY, United States

Abstract

Recently, there has been an increasing interest in knowledge graphs (KGs) of causal relations between events. Such KGs can be used for event analysis and forecasting in a variety of applications. In this paper, we study the problem of enriching an existing causal KG of news events using KG embeddings-based link prediction techniques. We perform a thorough evaluation of the performance of five different methods using classic accuracy measures as well as a novel scheme for manual evaluation. Our study provides insights on the strengths and weaknesses of different link prediction methods.

Keywords

Link Prediction, Causal Knowledge, Knowledge Graphs

1. Introduction

Knowledge graphs (KGs) have become an important source of structured information for diverse applications such as question answering, search engine support, and semantic understanding of structured and unstructured data. Concurrently, the concept of building causal knowledge graphs (causal KGs) has gained attention in the research community [1, 2, 3, 4, 5]. An illustrative example of a causal KG based on Wikidata [4] is shown in Fig. 1. The nodes represent the events and the edges represent their relations. Such KGs represent causal relations between events as triples, such as *<earthquake, hasEffect, tsunami>*.

One important application of causal KGs is event forecasting [5]. For example, in Fig. 1, the *earthquake* events were responsible for *tsunami* and *landslides* in the past. Such information can be vital in predicting the possible consequences of future earthquake events. A challenge in using existing KGs such as Wikidata is the sparsity of causal relations. Wikidata contains tens of thousands of event entities but fewer than 6,500 causal relations (based on our event selection process detailed in Section 4.2). As a simple example, it has been recorded in the past that the Volcanic eruption of Krakatoa in 1883 caused a tsunami, but this causal edge is not


Workshop on Deep Learning for Knowledge Graphs (DL4KG@ISWC2022), October 23-24, 2022

[†]Work done while at IBM Research.

✉ khatiwada.a@northeastern.edu (A. Khatiwada); shiras2@rpi.edu (S. Shirai); kavitha.srinivas@ibm.com (K. Srinivas); hassanzadeh@us.ibm.com (O. Hassanzadeh)

🆔 0000-0001-5720-1207 (A. Khatiwada); 0000-0001-6913-3598 (S. Shirai); 0000-0001-5307-9857 (O. Hassanzadeh)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

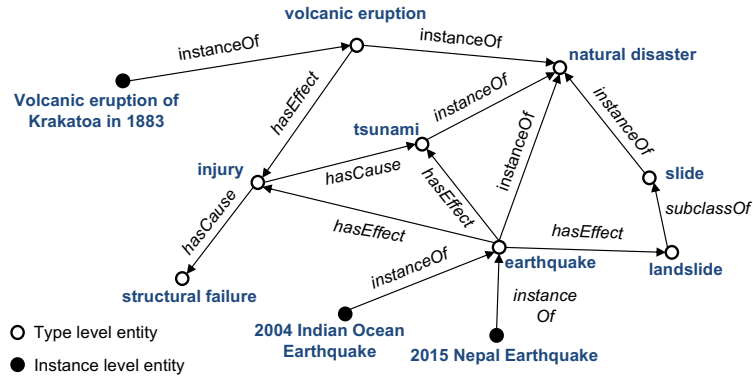


Figure 1: An illustration of a Causal Knowledge graph.

captured in our sample KG. Such sparsity of causal relations limits the capability of predicting potential consequences of various kinds of events.

There is a large body of work on link prediction, but most focus on the problem of predicting missing relations *in general* between the entities in the KGs [6, 7, 8]. The task is formulated as a target entity prediction problem – i.e., given a source entity and a relation, the objective is to find the target entity in the KG as $\langle \text{source entity}, \text{relation}, ? \rangle$. However, to the best of our knowledge, predicting causal relations in KGs has not been explored yet. Apart from sparsity, certain properties differentiate causal relations from the majority of relations in existing benchmarks. For example, causal relations are generally many-to-many i.e., the same event can have multiple effects and vice-versa. Like in Fig. 1, *earthquake* has multiple effects (*tsunamis*, *landslides*, etc) and if we formulate this as a link prediction problem ($\langle \text{earthquake}, \text{has effect}, ? \rangle$), we have multiple possible answers (e.g. *tsunami*, *landslide*, etc). But for some popular relations like place of birth, (e.g., $\langle \text{barack obama}, \text{place of birth}, ? \rangle$), there is a single possible answer i.e., *Honolulu*. Existing link prediction benchmarks like FB15k-237, WN18RR, CODEX-L do not contain event entities and causal relations. Hence, a proper evaluation of link prediction techniques on causal relation prediction is yet to be explored. Prior literature suggests that embedding and graph-based models are effective for general link prediction tasks [6]. In this paper, we study the performance of current and prior state-of-the-art embedding and Graph Convolutional Network (GCN)-based link prediction techniques on causal relation prediction and analyze their performance.

Summarizing our contributions, to the best of our knowledge, we are the first to evaluate embeddings and GCN based link prediction techniques on causal relation prediction task. Since the existing benchmarks do not contain events and causal relations, we create two new causal KG datasets using events in Wikidata. We perform a thorough evaluation of link predictions methods over the datasets using classic accuracy metrics as well as a novel manual evaluation scheme to analyze the limitations of the classic metrics.

2. Related Work

Different embedding-based approaches are proposed and shown to be effective for Link Prediction task [6, 7]. Bordes et al. introduces a linear additive model called TransE that learns the embeddings for entities and relationships in KGs by translating them to low-dimensional vectors [9]. Precisely, for a triple $\langle \text{source entity}, \text{relation}, \text{target entity} \rangle$, TransE translates the source entity to target entity using their relation as a translation vector. Balazevic et al. proposes a simple linear model (TuckER) based on tucker decomposition of the binary tensors of triples [10]. TuckER nearly achieves state-of-the-art performance in some existing benchmarks. Moreover, Trouillon et al. proposes ComplEx, a semantic matching model, that represents each entity using a pair of complex conjugate vectors: one for each entity as a source and as a target [11]. This helps ComplEx to deal with the asymmetric relations and it shows promising results in the existing link prediction benchmarks [7]. There are non-linear models like ConvE that use Convolutional Neural Network to predict missing relations in the KGs [12]. ConvE uses 2D convolutional operation on the source entity and relation to infer the target entity after processing the convolution result. In existing datasets, ConvE achieves state-of-the-art performance in terms of mean reciprocal rank (MRR) [12]. Recently, Graph Convolution Network (GCN) based models are seen to be effective in the link prediction task [13]. For instance, Nguyen et al. presents NoGE that uses co-occurrence information between entities and relations in the encoder module. NoGE achieves state-of-the-art performance against linear models, CNN-based models and semantic matching models in CODEX-M and CODEX-L benchmarks [13, 14]. To the best of our knowledge, the above link prediction techniques have not been applied to causal KGs. While there are a number of causal KGs such as ATOMIC [2], CauseNet [3], and CausalKG [15], our focus in this paper is on a Wikidata-based KG that captures news events and has application in news event analysis and forecasting [5, 16]. Our work is orthogonal to prior work on using textual sources to enrich causal KGs [1, 4].

3. Background

3.1. Preliminaries

We use \mathcal{E} and \mathcal{R} to denote set of all entities and all relations in Knowledge Graph \mathcal{K} respectively. Furthermore, we use e to denote an entity in \mathcal{E} and r to denote a relation in \mathcal{R} . Also, we denote a triple in \mathcal{K} as $\langle e_s, r, e_t \rangle$ where $e_s, e_t \in \mathcal{E}$ and $r \in \mathcal{R}$. Here, e_s denotes the source entity, e_t denotes the target entity and r denotes the relation between e_s and e_t . Whenever mathematical calculations are involved, we use e_s , r , and e_t also to denote their respective vector representations. Accordingly, we will now define our problem.¹

Definition 1 (Causal Relation Prediction Problem). Given a Knowledge Graph \mathcal{K} with a set of entities \mathcal{E} and a set of relations \mathcal{R} , source event entity $e_s \in \mathcal{E}$, causal relation $r \in \mathcal{R}$, and an integer k , the causal relation prediction problem is find the set of top-k target event entities $\mathcal{E}_{top} = \{e_1, e_2, \dots, e_k\}$, such that $\langle e_s, r, e_i \rangle \in \mathcal{K}$ for all $e_i \in \mathcal{E}_{top}$.

¹Note that since we are interested in causal relations, the input source entity must be an event.

Benchmark	FB15k-237					WN18RR					CODEX-L				
	MRR	Hits@1	Hits@5	Hits@10	Hits@50	MRR	Hits@1	Hits@5	Hits@10	Hits@50	MRR	Hits@1	Hits@5	Hits@10	Hits@50
TransE	0.32	0.22	0.384	0.520	<u>0.721</u>	0.162	0.028	0.350	0.547	0.634	0.145	0.056	0.242	0.328	0.463
TuckER	0.321	0.234	0.416	0.501	0.682	<u>0.457</u>	0.428	0.488	0.512	0.572	0.268	0.206	0.333	0.384	0.504
ComplEx	<u>0.332</u>	<u>0.236</u>	0.488	0.560	0.826	0.459	0.412	0.507	0.541	0.601	<u>0.293</u>	0.211	<u>0.392</u>	0.449	0.558
ConvE	0.313	0.219	0.390	0.469	0.657	0.423	0.392	0.459	0.491	0.552	0.291	<u>0.242</u>	0.354	0.402	0.512
NoGE	0.336	0.245	<u>0.433</u>	<u>0.520</u>	0.707	0.454	<u>0.413</u>	<u>0.498</u>	<u>0.541</u>	<u>0.629</u>	0.317	0.247	0.393	<u>0.444</u>	<u>0.554</u>

Best score
Second Best score

Figure 2: Link Prediction results on existing benchmarks using classical evaluation metrics.

3.2. Link Prediction Techniques

In this work, we study the performance of link prediction techniques on the scope of causal relation prediction. There are many link prediction techniques and their extensions in the literature [6, 7]. Here, we describe techniques that we include in our analysis. Note that there is not a single technique that achieves state-of-the-art performance in all the link prediction benchmarks [6]. Therefore, for comprehensive analysis, we study the best performing models from different categories like linear models, semantic matching models, CNN-based models and GCN-based models. From linear, we select **TransE** [9], one of the first embedding-based model, and **TuckER** [10] that improves over TransE and performs similar to state-of-the-art techniques. Similarly, we use **Complex** [11] among semantic matching models and **ConvE** [12] among CNN-based models. Moreover, we use **NoGE** [13] among GCN models that is shown to perform better than other Graph based models in the existing benchmarks.

4. Evaluation

Now, we evaluate the link prediction techniques on the scope of causal relation prediction empirically. We run all the experiments in Python 3.8 using a computing cluster (Intel Supermicro SYS-4029GP-TVRT, 64 GB memory and 8 × 768 MB V100-SXM2 GPU). We implement all the techniques (see Section 3.2) using publicly available codes. We reproduce TransE and ComplEx using their distributed implementation provided in PyTorch-BigGraph [17]. Furthermore, we implement TuckER, ConvE and NoGE using the codes in their respective github repository (see appendix Fig. 5).

4.1. Link Prediction Effectiveness

As we analyze link prediction techniques for causal relation prediction, we start our experiments by evaluating their performance on link prediction task. Specifically, we report **Mean Reciprocal Rank (MRR)** and **Hits@k** [7] for k = 1, 5, 10 and 50 in three existing link prediction benchmarks: **FB15k-237**, **WN18RR** and **CODEX-L**. We train each technique using train set for at least 500 epochs using the hyperparameters suggested in the respective papers and select a model from the checkpoint with minimum validation loss.

The performance of each technique on each benchmark is reported in Fig. 2. Similar to what previous works have reported [7, 13], ComplEx and NoGE are the best performing models with ConvE also showing reasonable performance. ComplEx is benefitted by its ability to handle the

assymmetric relations. NoGE captures the graphical properties like co-occurrence of the nodes and performs better than other techniques. TransE, being a simple translation model, performs well on the FB15k-237 benchmark in terms of MRR but its performance drops significantly on WN18RR and CODEX-L in comparison to other techniques.

4.2. Causal Relation Prediction Benchmarks

Existing link prediction benchmarks: **FB15k-237**, **WN18RR** and **CODEX-L** mostly contain non-event related triples and they are suitable only for general link prediction evaluation. Therefore, we create two new causal relation benchmarks by extracting the event-related triples from Wikidata for our evaluation. Our objective is to represent Causal Knowledge Graphs of news events and their causal relations in these benchmarks.

Event selection. We use the triples containing *event entities* in Wikidata to create the causal relation prediction benchmarks. To select such triples, we first determine event entities in Wikidata. Unfortunately, there are no properties or classes (types) in Wikidata that distinguishes event entities from other entities. Generally, the day-to-day events happening around the world are covered by news articles and their headlines [4]. Therefore, we consider the Wikidata classes having a mapping to the news articles in *Wikinews* as event classes. Note that all the events in news articles do not have a mapping to Wikidata classes. Using mapping, we identify 50 event classes (manually verified after extraction) such as earthquake, tsunami, and disease outbreak. We consider all such classes, along with their subclasses (connected by *subclass of* relation) and instances (connected by *instance of* relation), as event entities.

Triple Extraction. After determining the event entities, we query Wikidata and extract all the triples that contain the event entities as either head or tail entities. To understand the impact of literals, we include the triples representing numerical properties of the events. Notice however, we exclude images and those literals that represent metadata like *wikibase:statements*, *wikibase:identifiers*, *wikibase:sitelinks*, *schema:version*, *schema:description* and *schema:about*. Furthermore, some event properties are not captured by triples containing the event instead may be indirect. The embedding models can infer such properties from the graph pattern [7]. Therefore, we also extract the entities that are two hops away from each event. In total, we collect around 1M triples from Wikidata. Then we remove the cyclic triples (having the same head and tail entities), which we consider as noise in the KG and they do not add interesting information for causal analysis. This leave us with around 980k triples, among which around 6k are cause-effect triples (based on relations listed in wikidata.org/wiki/Wikidata:List_of_properties/causality). We create two benchmarks using these triples.

Benchmark Creation. We split the collected event related triples into test, train and validation set. Recall that we want to study the performance for causal relation prediction task. Therefore, we ensure that the test set contains only the causal relations. In validation set, we create two variations: one contains only the causal triples (similar to test set) and other contains the mixture of causal triples and other event related triples (similar to train set). This gives an insight on dataset preparation and helps us to understand the role of event related but non-causal relations in learning the link prediction models for causal relation task. The train set, which resembles causal KGs, contains causal triples along with other event-related triples.

(i) **Wiki data Causal validation (WikiCV)** Benchmark contains only causal triples in

Benchmark Method	WikiCV					WikiMV				
	MRR	Hits@1	Hits@5	Hits@10	Hits@50	MRR	Hits@1	Hits@5	Hits@10	Hits@50
TransE	0.017	0.001	0.029	0.056	0.126	0.037	0.017	0.052	0.076	0.16
TuckER	0.003	0.001	0.003	0.005	0.061	0.059	0.043	0.076	0.080	0.105
ComplEx	<u>0.024</u>	0.001	<u>0.040</u>	<u>0.076</u>	<u>0.185</u>	0.038	0.002	0.065	0.117	<u>0.231</u>
ConvE	0.023	0.009	0.036	0.046	0.094	<u>0.078</u>	<u>0.040</u>	<u>0.108</u>	<u>0.135</u>	0.207
NoGE	0.032	<u>0.005</u>	0.052	0.085	0.187	0.085	0.026	0.140	0.201	0.336

Best score
Second Best score

Figure 3: Causal Relation Prediction on WikiCV and WikiMV using classical evaluation metrics.

the validation set. Specifically, out of around 6k causal relation triples, we randomly select 1000 triples each for test and validation set, and remaining triples are used for train set. For convenience, we convert all the causal relations (hasCause, hasContributingFactor, etc) into *hasEffect* relation in test and validation set. Furthermore, there is an issue of triple leakage—the test triples are visible in training sets in the inverse form—identified on the previous link prediction benchmarks [8]. So to avoid leakage, we remove inverse relations from the train set which gives train set having around 941k triples. Other details like number of triples, number of causal relation triples, number of entities and number of relations are shown in appendix (Fig. 6).

(ii) **Wiki data Mixed Validation (WikiMV)** Benchmark is created on the same way as WikiCV. The difference is the validation set which contains the mixture of causal as well as other event related triples. Also, to accommodate event related triples along with causal relation triples, we use larger validation set and equal test set (1, 500 triples each). Due to this change, the number of removed inverse relations changes. So, the number of triples in train set (~951k) on WikiMV is different than that on WikiCV.

4.3. Causal Relation Prediction Effectiveness

Now we present the effectiveness of five different link prediction techniques on the causal relation prediction task. Fig. 3 shows MRR and Hits@k for k = 1, 5, 10 and 50 on WikiCV and WikiMV benchmarks. While there isn't a clear winner for the link prediction task, NoGE outperforms other techniques in all but Hits@1 in both causal relation benchmarks. Specifically, it outperforms second best technique—ComplEx— in terms of MRR by around 33 % in WikiCV and slightly outperforms the second best technique—ConvE— in WikiMV benchmark. In terms of Hits@10, NoGE is better by around 11 % and almost two times than ComplEx in WikiCV and WikiMV respectively. NoGE seems to capture the graph structure better even for the sparse graph. We observed that TuckER and TransE are able to capture the semantic similarity but could not infer the causal information well. Hence, most of their predictions are other events of the same class rather than consequence or effect events.

Furthermore, we observe that the techniques perform better when we use causal relations, together with other event related triples, for validation. This is because the validation output is used to select the model for evaluation and the non-causal relations appearing with the events (instance of, subclass of, etc) act as negative samples to the model. This helps the models to better distinguish causal relations from other relations.

4.4. Manual Result Analysis

The classical metrics show that NoGE performs relatively better in link prediction tasks, followed by ConvE and ComplEx. However, looking at numbers, we see weaker performance on causal relation prediction task. For example, the MRR of NoGE is over 30 % in all link prediction benchmarks (see Fig. 2). But it drops significantly in both Causal Relation Benchmarks (see Fig. 3). To understand this issue, we verify the results of each technique manually in both benchmarks. We observe that the techniques are penalized by the evaluation metrics due to sparsity and missing relations in KG that we use to create the benchmark. For instance, there is a causal relation *<civil disorder, hasEffect, curfew>* in WikiMV test set. When we use this as a test case (querying *<civil disorder, hasEffect, ?target event>*), NoGE returns *demonstration* as the top-1 result which seems to be true. However, since this information is not available in the groundtruth, the classical evaluation measure, which uses closed-world assumption [6], considers *demonstration* as incorrect prediction.

It is impractically time consuming to evaluate each result manually or to label the complete groundtruth. Therefore, to understand the performance of each technique even better, we develop a novel manual evaluation strategy that compares the success of each technique on causal relation prediction task with reduced effort. Instead of evaluating instance-level results, our idea is to evaluate the results on type level by mapping each instance level prediction into their most granular types.² We illustrate this with an example.

Example 1. Consider causal relation prediction task i.e., *<source event, hasEffect, ?target event>* where the objective is to predict the effect (target event). Consider two instance level predictions: *<Murder of George Floyd (Q95579249), hasEffect, George Floyd protests>* and *<Death Of Javier Ordóñez (Q99194919), hasEffect, Javier Ordóñez protests>*. Here, both input source events belong to type *murder* and both predicted target events belong to type *protest*. So we map all instances to their respective types and evaluate correctness of *<murder, hasEffect, protest>*.

Mapping to type level prediction. Our objective is to evaluate each technique on type level and observe their relative performance. For that we generate top-k type level target events for each source event and evaluate them manually. At first, we find a ranked list of 50 target predictions for each source event in the test set by each technique. The ranking is based on confidence score that each technique assign to the target entity while making a prediction. Recall that the train set contains: (i) literals such as numbers, dates, etc. which are not events but may help in predicting causal relations, (ii) entities having no labels in Wikidata and (iii) entities having more than one most granular type that creates ambiguity during evaluation. Some techniques may predict such literals, unlabeled entities and entities having ambiguous types as target events. We consider such predictions as incorrect and filter them out. Our evaluation metrics (*recall*), to be discussed later, will penalize such predictions. After filtration, if a predicted entity is already a type, we keep it as it is; else, we map each source event and its predictions to their respective types as discussed in Example 1. Note that there can be multiple instance level predictions having the same type level predictions (see Example 1). In such case, we record the maximum confidence score among them. This is because the highest score among each instance level predictions signifies the best confidence of the technique for that prediction.

²Here onwards, we simply use type to denote an entity’s most granular type unless mentioned otherwise.

Benchmark Method	WikiCV				WikiMV			
	Precision	Recall	F1-Score	F1-Score (at θ)	Precision	Recall	F-score	F-score (at θ)
TransE	0.277	0.141	0.187	0.188	0.330	0.177	0.230	0.231
TuckER	0.200	0.108	0.140	0.141	0.219	0.130	0.163	0.164
ComplEx	0.469	<u>0.146</u>	0.223	0.224	0.548	<u>0.188</u>	0.280	0.280
ConvE	0.299	0.110	0.161	0.164	0.348	0.171	0.230	0.238
NoGE	<u>0.386</u>	0.155	<u>0.221</u>	<u>0.221</u>	<u>0.386</u>	0.188	<u>0.253</u>	<u>0.255</u>

Best score
Second Best score

Figure 4: Precision, recall and F-score of different techniques based on manual result analysis

Groundtruth creation. We record top-10 type level target events based on confidence score for each type level source event. Of course, all source event may not find 10 target events because either they do not have at least 10 target events in train set or they do not predict events instead predict literals and ambiguous entities. The details on the number of source events, target events and (source event, target event) pairs generated by each technique is reported in appendix (Fig. 7). It is seen that TuckER and TransE produce the largest number of (source event, target event) pairs whereas ComplEx and ConvE are the most selective techniques producing least pairs. Next, we manually label—either *true* or *false*—(source event, target event) pairs produced by each technique and use result to create a groundtruth. In groundtruth, the true causal relations are all (source event, target event) pairs labeled as true. All other pairs are false causal relations. Note that there can be true causal relation not produced by any techniques, and hence do not make it to the groundtruth as true. However, since we are interested in relative comparison of the techniques, we assume that each true causal relation is predicted by at least one technique. Note that an event may have more than 10 target events in the groundtruth if different techniques predict different set of target types for the same source event.

Manual Evaluation metrics. After creating the groundtruth based on manual annotation, we report precision (P), recall (R) and F-score for each technique. Let T_M be the set of predictions made by a technique whose size depends on the number of target events that are queried for each source event. Let T_G be the set of true causal relations in the groundtruth. Then, P, R and F-score are given by:

$$P = \frac{|T_M \cap T_G|}{|T_M|}, R = \frac{|T_M \cap T_G|}{|T_G|}, F\text{-score} = \frac{2 \cdot P \cdot R}{P + R} \quad (1)$$

Manual Evaluation Effectiveness Results. We evaluate precision, recall and F-score at different value of k i.e. different number of target events per source event. Considering events in both WikiCV and WikiMV, the median and average target event per source event types is 7 and 7.5 respectively. Therefore, we select a nearby value $k = 5$ for our discussion. The results on other values of k are shown in appendix (Fig. 8).

Fig. 4 shows effectiveness of different techniques in both WikiCV and WikiMV benchmarks for top-5 prediction per source event. Here, we focus on Precision, Recall and F-score. We will explain F-score (at θ) in Section 4.5. We observe that ComplEx and NoGE are the best methods on both benchmarks with ComplEx having the best precision and NoGE having the best recall. In terms of F-score, ComplEx slightly outperforms NoGE in WikiCV benchmark

and by around 10 % in WikiMV benchmark. This shows that although ComplEx produces less results, they are highly precise. On the other hand, TransE and TuckER produces larger number of predictions (see Fig. 7) which favors their recall but penalizes precision. Note that even though ComplEx and NoGE produce less number of prediction results, they have much higher precision than other methods and comparable recall. When we look at the recall at different k in both benchmarks (Fig. 8), we observe that ComplEx (blue square line) performs better until $k = 5$ but its recall starts getting saturated after that. This shows that ComplEx performs better for the source events having fewer target events. TransE and TuckER, on the other hand, has higher recall with increase in k because their correct predictions are spread over all ranking positions rather than compressed at top rankings. We also observed in our manual evaluation that each method make accurate predictions that cannot be found in the output of other methods.

4.5. Effect of threshold

Finally, we see if applying threshold on confidence scores increases effectiveness (F-score) over top- k approach. A higher threshold increases precision but decreases recall and vice-versa. Thus, we consider maximization of F-score, the combination of both precision and recall. For fair comparison, we first generate the top- k results and then apply different threshold to see if there is an improvement over F-score. F-scores in both WikiCV and WikiMV before and after applying threshold to the top-5 results is shown under (*F-Score*) and *F-Score (at θ)* column respectively (Fig. 4). All but NoGE in WikiCV and ComplEx in WikiMV show small improvement in F-score when applying the best threshold. We also analyze the precision, recall and F-score trend for different thresholds (appendix Fig. 9). We only show the trend for TransE, ComplEx and NoGE in WikiMV benchmark as each technique follows similar trend on both benchmarks. Also, the trend for TuckER and ConvE resembles to that of TransE. For TransE, TuckER and ConvE, the precision trend is random. But for better performing techniques (ComplEx and NoGE), precision increases with increase in threshold. For all techniques, recall goes up with decrease in threshold and seems to saturate for further decrease. This shows that there could be a threshold value for each technique where we get the best F-score which we will explore in future work.

5. Conclusion

We evaluated existing link prediction techniques for causal relation prediction in Wikidata-based causal Knowledge Graphs that contain highly sparse and generally many-to-many causal relations. Based on classical link prediction metrics, we observe that the techniques perform better for the well-studied link prediction task but show weaker performance in the causal relation prediction task. Furthermore, we observed that our model trained on a dataset with a mixture of causal and other non-causal but event-related triples performed better than one trained on a datasets of causal relation triples only. We also studied the drawbacks of existing metrics and proposed a novel manual evaluation strategy. Our results show that the techniques generally perform better than what the classic metrics indicate, although there is still plenty of room for improvements. We also observed that each of the methods, regardless of their accuracy scores, make accurate predictions that cannot be found in the output of the other methods. In the future, we will explore using a combination of different techniques, including rule-based link

prediction methods [18], to get better overall prediction results. Also, we will further explore the use of threshold instead of top-k ranking as a robust mechanism of KG enrichment.

References

- [1] O. Hassanzadeh, D. Bhattacharjya, M. Feblowitz, K. Srinivas, M. Perrone, S. Sohrabi, M. Katz, Causal knowledge extraction through large-scale text mining, in: *AAAI*, 2020.
- [2] M. Sap, R. LeBras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, Y. Choi, *ATOMIC: An atlas of machine commonsense for if-then reasoning*, *CoRR abs/1811.00146* (2018). URL: <http://arxiv.org/abs/1811.00146>.
- [3] S. Heindorf, Y. Scholten, H. Wachsmuth, A.-C. Ngonga Ngomo, M. Potthast, *Causenet: Towards a causality graph extracted from the web*, in: *CIKM*, 2020, pp. 3023–3030.
- [4] O. Hassanzadeh, *Building a knowledge graph of events and consequences using wikipedia and wikidata* (2022).
- [5] O. Hassanzadeh, P. Awasthy, K. Barker, O. Bhardwaj, D. Bhattacharjya, M. Feblowitz, L. Martie, J. Ni, K. Srinivas, L. Yip, *Knowledge-based news event analysis and forecasting toolkit*, in: *IJCAI*, 2022, pp. 5904–5907.
- [6] Q. Wang, Z. Mao, B. Wang, L. Guo, *Knowledge graph embedding: A survey of approaches and applications*, *IEEE Transactions on Knowledge and Data Engineering* 29 (2017).
- [7] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, *Knowledge graph embedding for link prediction: A comparative analysis*, *ACM TKDD* 15 (2021) 1–49.
- [8] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, C. Li, *Realistic re-evaluation of knowledge graph completion methods: An experimental study*, in: *SIGMOD*, 2020, p. 1995–2010.
- [9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, *Translating embeddings for modeling multi-relational data*, *NeurIPS* 26 (2013).
- [10] I. Balažević, C. Allen, T. Hospedales, *Tucker: Tensor factorization for knowledge graph completion*, in: *Proceedings of EMNLP-IJCNLP*, 2019, pp. 5185–5194.
- [11] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, *Complex embeddings for simple link prediction*, in: *International conference on machine learning*, PMLR, 2016.
- [12] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, *Convolutional 2d knowledge graph embeddings*, in: *Proceedings of the AAAI conference on artificial intelligence*, 2018.
- [13] D. Q. Nguyen, V. Tong, D. Phung, D. Q. Nguyen, *Node co-occurrence based graph neural networks for knowledge graph link prediction*, in: *WSDM '22*, 2022, p. 1589–1592.
- [14] T. Safavi, D. Koutra, *Codex: A comprehensive knowledge graph completion benchmark*, in: *EMNLP*, 2020.
- [15] U. Jaimini, A. P. Sheth, *CausalKG: Causal knowledge graph explainability using interventional and counterfactual reasoning*, *IEEE Internet Comput.* 26 (2022) 43–50.
- [16] K. Radinsky, S. Davidovich, S. Markovitch, *Learning to predict from textual data*, *J. Artif. Intell. Res.* 45 (2012) 641–684.
- [17] A. Lerer, L. Wu, J. Shen, T. Lacroix, L. Wehrstedt, A. Bose, A. Peysakhovich, *PyTorch-BigGraph: A Large-scale Graph Embedding System*, in: *2nd SysML Conference*, 2019.
- [18] S. Shirai, A. Khatiwada, D. Bhattacharjya, O. Hassanzadeh, *Rule-based link prediction over event-related causal knowledge in wikidata.*, in: *Wikidata@ ISWC*, 2022.

A. Appendix

Technique	Source code link
TransE	https://github.com/facebookresearch/PyTorch-BigGraph
TuckER	https://github.com/ibalazevic/TuckER
ComplEx	https://github.com/facebookresearch/PyTorch-BigGraph
ConvE	https://github.com/TimDettmers/ConvE
NoGE	https://github.com/daiquocnguyen/GNN-NoGE

Figure 5: Link to the GitHub repository of different techniques used in the experiments.

Benchmark	WikiCV			WikiMV		
	Train	Validation	Test	Train	Validation	Test
# Triple	941, 236	1, 000	1, 000	951, 791	1, 500	1, 500
# Causal	1, 655	1, 000	1, 000	2, 289	404	1, 500
# Entity	560, 074	1, 513	1, 998	570, 182	2, 598	2, 285
#Relation	1, 547	1	1	1, 818	196	1

Figure 6: Details of Causal Relation Benchmarks used in the experiments.

Benchmark	WikiCV				WikiMV			
	Source Events	Target Events	Total Events	Evaluation Pairs	Source Events	Target Events	Total Events	Evaluation Pairs
TransE	39	47	48	337	42	49	49	350
TuckER	39	20	42	386	43	24	47	417
ComplEx	30	37	40	171	34	38	48	176
ConvE	34	42	45	199	40	42	47	281
NoGE	36	42	45	215	40	42	47	264

Figure 7: Number of events and evaluation pairs generated by different techniques considering top-10 target events for each source event. Here, evaluation pairs indicates (source event, target event) pairs generated by each technique

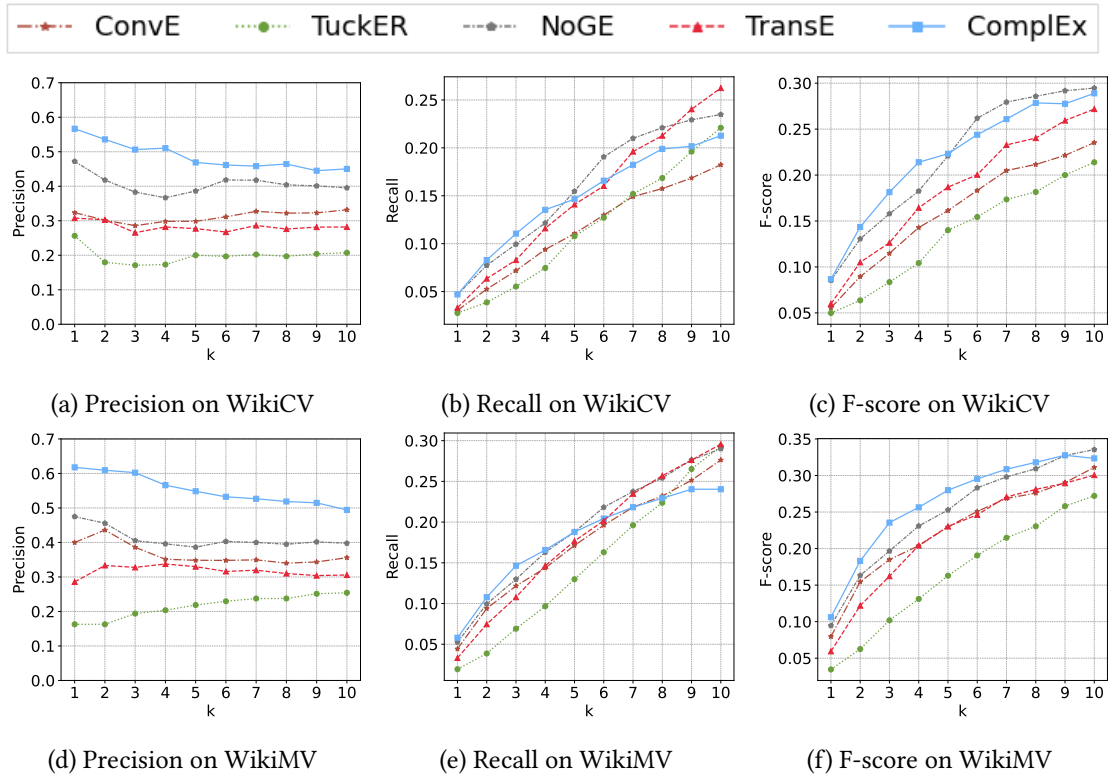


Figure 8: Precision, Recall and respective F-scores of Causal Relation Prediction on different benchmarks by the baselines. Here, x-axis shows the number of top-k target events considered for each source event.

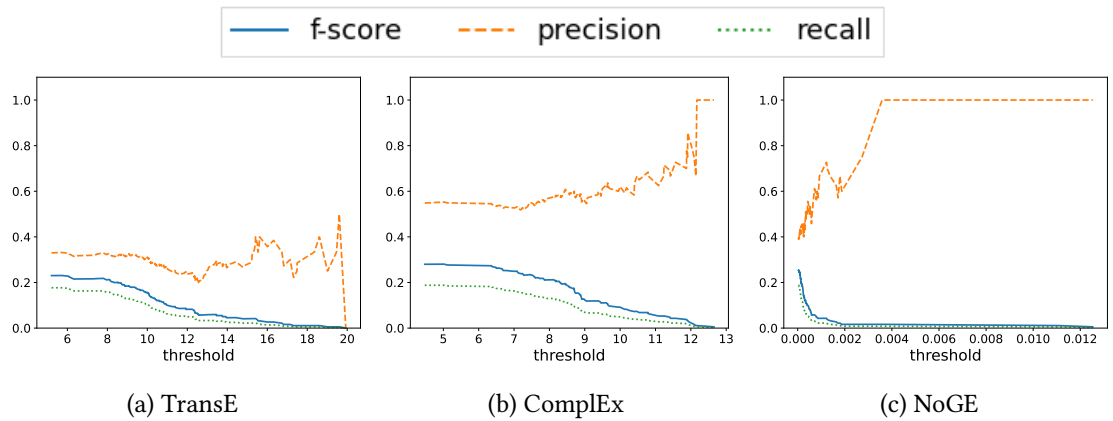


Figure 9: Precision, Recall and F-score of different techniques when applying threshold over top-5 results in WikiMV benchmark