Rule-Based Link Prediction over Event-Related Causal Knowledge in Wikidata

Sola Shirai[1,2], Aamod Khatiwada[1,3], Debarun Bhattacharjya[1] and Oktie Hassanzadeh[1]

[1]*IBM Research, Yorktown Heights, NY, United States*

[2]*Rensselaer Polytechnic Institute, Troy, NY, United States*

[3]*Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA*

Abstract

Rich semantic information contained in Wikidata about newsworthy events and their causal relations may serve as a valuable resource to perform event analysis and forecasting. However, prior work in leveraging methods such as link prediction over causal event data in knowledge graphs has been limited. In this work we share our methods and findings to curate a dataset of newsworthy events with cause-effect relations and apply rule-based link prediction models. We find that the performance of such models can vary greatly among the various relations contained in our curated data, and we identify several points of consideration for both the data curation process and model performance when using knowledge about events that are currently present in Wikidata.

Keywords

Link Prediction, Causal Knowledge, Knowledge Graphs

1. Introduction

Data about past newsworthy events, as well as subsequent events they caused, can serve as a valuable source of information to reason about ongoing events and make forecasts about the future. For example, past data about major earthquake events may indicate the occurrence of consequent events such as a tsunami, economic recessions, or even lynchings.[1] The ability to analyze events and make forecasts using causal relations for such events can be invaluable for proactive decision making by various organizations.

One method to represent information about such newsworthy events is to use a knowledge graph (KG), like Wikidata. Capturing events in this way has the benefit of being unambiguous, enabling interoperability among different KGs, and providing rich semantic information about entities and relationships. For some events, Wikidata also includes explicit cause-effect relationships like has_cause (P828) and has_effect (P1542) – such causal relations can exist

---

[1]E.g., the Kantō Massacre, caused by civil unrest and misinformation in the wake of the 1923 Great Kantō earthquake
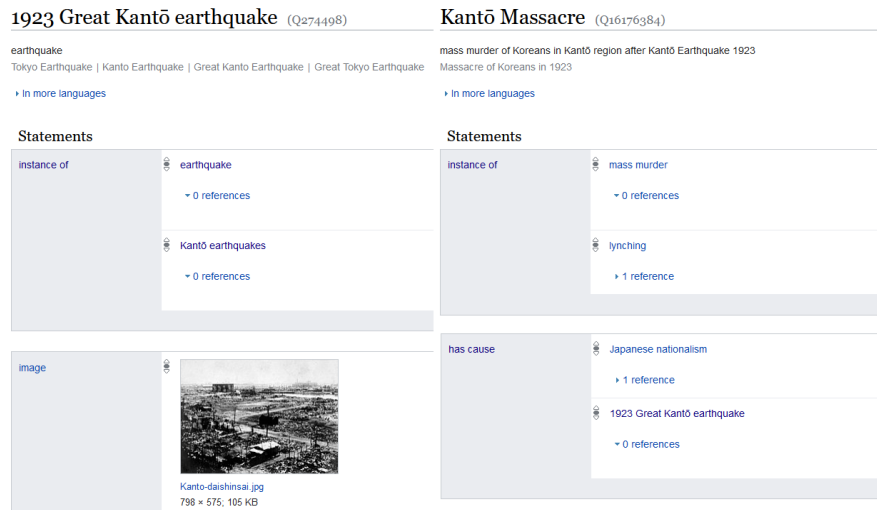
Figure 1: An example of a major earthquake event and its consequent event within Wikidata.

between both specific instances of events and general classes of events (e.g., (`earthquake`, `has_effect`, `landslide`)). An example of such an event in Wikidata can be seen in Figure 1.

While Wikidata contains many causal relations, it is still far from complete. To address this, prior work has applied knowledge extraction methods over text documents to enrich Wikidata with causal relations expressed in Wikipedia articles [1]. An alternative approach is to utilize link prediction methods. To predict the effect of an `earthquake` event, for example, we can try to predict the tail entity of a link as (`earthquake`, `has_effect`, `?t`). While a large variety of link prediction models have been developed in recent years [2], in this work we investigate the use of rule-based models. Besides including inherent interpretability due to their use of semantic relations rather than latent features, rule-based models also are capable of performing inductive link prediction (i.e., link prediction for entities that were unseen during training). These two factors are especially valuable for event analysis and forecasting, allowing event forecasts to be performed for entirely new events while providing a level of explainability to decision makers.

Despite the potential of leveraging KGs to represent and forecast causal relations among events, the use of KG-completion and link prediction methods for event forecasting has been limited [3]. Link prediction literature, on the other hand, has tended to focus on improving state-of-the-art performance over common experimental datasets rather than applying it to KGs with causal relations among events.

In this paper, we aim to bridge this gap by investigating the performance of rule-based link prediction methods in a KG of causal events extracted from Wikidata. We begin by describing our data curation methods. Next, we apply two rule-based link prediction models to our dataset to study their performance. Lastly, we provide an analysis of the results obtained by these models to identify factors that affect their performance as well as characteristics of event-related data that is currently captured in Wikidata.

2. Related Work

The task of link prediction in KGs has garnered significant interest in recent years, with a number of surveys [2, 4, 5] presenting the breadth and variety of models that have been developed. Most commonly, this task is performed using machine learning models which aim to learn some latent embedding for the entities and relationships present in the KG. One of the shortcomings of many such models is that they can only perform link prediction in the transductive setting – i.e., the model can only learn embeddings for entities that are present in the training data. Additionally, it remains questionable whether such models are able to appropriately capture the underlying semantic information [6]. On the other hand, rule-based methods for link prediction are naturally able to handle inductive link prediction, as they learn symbolic rules based on entities and their relations. While relatively less attention has been given to rule-based link prediction models, recent analyses have demonstrated that such models can exhibit state-of-the-art performance while requiring significantly less training time than many embedding-based models [7].

For existing link prediction work, most analysis and experiments for link prediction have been conducted on standard benchmarks such as FB15k-237 [8] and WN18RR [9]. While such datasets may serve as a useful means to compare different models, it is unclear if or how these models might perform on a KG that captures causal relations and events.

Rule-based link prediction models aim to learn Horn rules which imply a target relation. Such methods, including AMIE [10], AMIE+ [11], AMIE3 [12], DRUM [13], RUDIK [14], and RuleN [15], learn rules and compute a confidence for each rule based on factors such as the number of correct groundings in the background knowledge or the coverage of those rules. The most notable rule-based link prediction model from recent years is AnyBURL [16], which learns rules efficiently using a bottom-up strategy. In addition to continued development to the AnyBURL model by its authors, further work has also been conducted which aims to apply the learned rules more effectively [17].

An alternative approach to learning rules is to utilize case-based reasoning (CBR). The key motivation for CBR approaches [18, 19] is to make predictions for a specific type of relation by identifying alternative paths through the KG. These alternative paths function equivalently to how "rules" are viewed in the aforementioned methods. Such alternative paths are chosen based on searching for similar entities in the KG. A key difference between the CBR models and rule learning models is that CBR does not require any prior training or learning step (although for practical considerations, the implementation of such methods precompute and store various metrics). These CBR approaches have also been extended to performing question answering over KGs using natural language [20].

3. Event Data Collection

3.1. Causal Event Selection

To curate our dataset of newsworthy events and their causal relations from Wikidata, we first must decide what events are "newsworthy" and what constitute "causal relations" that we wish to query. For an example of what we might consider a newsworthy event, consider Wikidata's

entry for the 1923 Great Kantō earthquake (Q274498). While we can observe that such an event is an instance of the earthquake class, there is no distinguishing property or class type[2] that can provide us with a meaningful method to filter out what classes to consider as our events of interest. Attempting to simply query for all entities that have a causal relation also is not a viable option, as Wikidata contains a large amount of non-event related entities with such relations (such as reports (Q10429085) and motions (Q96739634) in the Swedish legislature, which include thousands of causal relations). Therefore, we chose to select a set of classes of interest by querying Wikidata for all entities that have a link to Wikinews as a means of identifying events that have received news coverage. Despite the fact that many newsworthy events do not have Wikinews coverage and/or links, the existing Wikinews links help us identify the majority of event types (classes) for newsworthy events and their causes and effects.

From this step, we identified a set of 307 classes as our event classes of interest. Next, we identified causal relations that we would use to query and collect the initial set of cause-effect event pairs. From the set of Wikidata properties for causal relations,[3] we use the `has_cause`, `has_immediate_cause`, and `has_contributing_factor` relations, as well as their inverses (`has_effect`, `immediate_cause_of`, and `contributing_factor_of`, respectively). We opt to include the inverses of the causal relations because Wikidata only contained triples for many cause-effect pairs in one direction. For example, the Kantō Massacre (Q16176384) refers to the 1923 Kantō earthquake as its cause, but the earthquake's entry does not contain a `has_effect` relation to the massacre event. Further details about how these relations are used in Wikidata to model causes can be found at wikidata.org/wiki/Help:Modeling_causes.

Our final query thus searched for all entities that are an instance of one of our event classes of interest and connected to another event of interest by one of our 6 causal relations. This query resulted in a total of **1,953** causal event pairs spanning **157** unique event types. We note that causal relations can be one-to-one, one-to-many, or many-to-many, and events may simultaneously have `has_effect` and `has_cause` relations. Our 1,953 event pairs were made up of 538 unique event entities in total, with **284** unique cause events and **311** unique effect events.

3.2. Data Curation

Having selected our newsworthy events with cause-effect relations, our second step was to collect additional relevant triples from Wikidata. Starting from our 538 unique event entities, we collected triples up to a distance of 3-hops away from each entity. The decision to only collect up to the 3-hop neighborhood is to keep in line with how most experiments for rule-based link prediction methods, which follow paths through the KG, tend to limit themselves to only search up to 3-hops. Additionally, to keep our dataset at a manageable size and quality, we only collect the 3-hop neighborhood that can be reached by outgoing relations. Several classes that were reachable from our events had an extremely large amount of incoming triples (notably the human (Q5) class, which had over 4 million incoming `instance_of` relations, and the `taxon` (Q16521) class, with nearly 3.5 million `instance_of` relations).

---

[2]We interchangably refer to the `instance_of` (P31) relation in Wikidata as the "type" or "class" of the event.
[3]https://www.wikidata.org/wiki/Wikidata:List_of_properties/causality

Lastly, we removed literals (numeric values, date-times, and strings) from the KG. Our final dataset consisted of roughly 1,080,000 triples, encapsulating 444,228 unique entities and 1,263 unique relations. Due to the manner in which we collected an outgoing 3-hop neighborhood around events, we also found that a significant number of entities in our dataset contained very few triples – roughly 70% of the entities only had one triple in the KG. While this would be problematic if we simply perform link prediction evaluation over all entities, our focus is on the events with causal relations, all of which have their 3-hop neighborhood of connections available in our dataset.

Most Common Cause Event Types	Count	Most Common Effect Event Types	Count
Disease Outbreak	264	Disease Outbreak	237
Disease	96	Disease	206
Infectious Disease	44	Closing of Educational Institutions	146
Rare Disease	34	Social Distancing	112
Shooting	32	Declaration of Public Health Emergency	111
War	32	Infectious Disease	93
Biological Process	31	Travel Restriction	77
Conflict	26	Clinical Sign	76
Phenomenon	25	Aviation Accident	62
Homicide	24	Lockdown	57

Table 1
The top 10 most common cause and effect event types in our dataset.

Some of the most common types of cause and effect events in our dataset can be seen in Table 1. A large number of events related to diseases are included, which we can likely attribute to entries related to the Covid-19 pandemic. Many similar Covid-related events have highly similar entries made for each country, resulting in the majority of our most common effect types also being Covid-related. On the other hand, the most common causes include some more variety, such as the "war" and "shooting" types.

4. Experiments

4.1. Link Prediction Models

To investigate the performance of rule-based link prediction methods, we apply two models from recent years – CBR [18] and AnyBURL [16]. CBR performs link prediction for a triple $(h, r_q, ?t)$ by first searching for k similar entities in the KG. For each similar entity h_s, CBR samples the KG for m alternative paths up to n hops in length that can be used to reach the target entity for the relation r_q – i.e., for a triple (h_s, r_q, t_s) in the KG, a path of relations $p = (r1, ..., r_n)$ connecting h_s to t_s is identified, where $p \neq (r_q)$. These sampled paths are then applied to the query triple, following each path p starting from the query entity h to reach candidate tail entities. Candidates are scored based on how many such sampled paths reach them. The CBR method is very simple, but demonstrated comparable performance to common KG embedding models in the original publication while requiring no training step.

On the other hand, AnyBURL performs efficient rule mining using a bottom-up approach.

AnyBURL first learns "bottom rules," which are Horn rules whose variables are grounded to specified instances in the KG. The bottom rules are then iteratively generalized, and confidence scores for each generalization are computed based on the number of body groundings in the KG that make the rule true. AnyBURL is trained for a set amount of time (up to 1,000 seconds by default) and has shown competitive performance to state-of-the-art link prediction models on standard benchmarks[4].

4.2. Experiments

We analyze the performance of CBR and AnyBURL by following standard evaluation procedures for link prediction tasks in KGs. Our dataset is split into training, validation, and test sets containing 70%, 15%, and 15% of the overall data, respectively. We run CBR over the training set using hyperparameter choices to select 5 similar cases and sample 20 paths (similar numbers to those used in the original publication), and we run AnyBURL using default configurations up to 1,000 seconds. Performance is measured over the test set using Hits@K for K=1 and 10 as well as Mean Reciprocal Rank (MRR).

Additionally, we perform experiments to assess the performance of the two link prediction models as the amount of training data decreases. One benefit of rule-based methods is that they can often learn rules from a small number of samples, whereas embedding-based models tend to require a large amount of training data and time to learn useful embeddings. To test this, we repeat the above experiment while reducing the training dataset size to 90%, 80%, 60%, and 40% of the original, evaluating against the test set for each size decrease.

5. Results and Discussion

5.1. General Model Performance

All Test Triples					Event-Related Test Triples				
Model	Train %	H@1	H@10	MRR	Model	Train %	H@1	H@10	MRR
CBR	100%	0.126	0.216	0.158	CBR	100%	0.139	0.202	0.161
	90%	0.114	0.198	0.143		90%	0.127	0.187	0.148
	80%	0.102	0.180	0.129		80%	0.114	0.169	0.134
	60%	0.081	0.142	0.102		60%	0.081	0.122	0.095
	40%	0.055	0.095	0.069		40%	0.055	0.082	0.065
AnyBURL	100%	**0.243**	**0.362**	**0.280**	AnyBURL	100%	**0.214**	**0.329**	**0.249**
	90%	0.230	0.347	0.267		90%	0.201	0.311	0.234
	80%	0.216	0.330	0.251		80%	0.181	0.287	0.214
	60%	0.186	0.294	0.219		60%	0.146	0.245	0.176
	40%	0.150	0.234	0.180		40%	0.115	0.199	0.141

Table 2
Model performance comparisons for the Hits@K (H@K) and MRR performance metrics.

---

[4]The most recent version of AnyBURL can be accessed at https://web.informatik.uni-mannheim.de/AnyBURL/

Table 2 displays the performance of the CBR and AnyBURL models over our dataset. For each model, the performance is compared with decreasing training data size (indicated as the Train % column). Additionally, we compare the performance of the models over all triples in the test set (shown on the left of the table) versus the performance when considering only triples whose head or tail entity are one of our events of interest (on the right).

In general we can observe that AnyBURL shows superior performance to the simple CBR model – this is an expected result, considering the previously reported performance metrics. However, when we specifically focus on triples in the test set related to events, we can observe that the performance of AnyBURL generally decreases while CBR very slightly increases. This result is interesting, as we would expect that event-related entities have more complete training data avaiable due to our data curation methods.

5.2. Performance for Relations of Interest

Table 3 shows a breakdown of the two models' performance in terms of MRR for the most common relation types in the test dataset. Test Count refers to the number of triples in the test dataset for each relation type, while Training Count refers to the number of triples for each relation that were present in the training set. Note that these counts and performance metrics are only considering the event-related test triples.

Most Common Relations	Test Count	Training Count	AnyBURL MRR	CBR MRR
subclass of (P279)	1239	26,574	0.125	0.105
instance of (P31)	961	59,420	0.458	0.229
has part(s) (P527)	354	15,178	0.179	0.283
has cause (P828)	318	1,797	0.258	0.296
drug used for treatment (P2176)	247	3,687	0.602	0.372
has effect (P1542)	237	1,369	0.282	0.384
described by source (P1343)	232	18,862	0.398	0.075
part of (P361)	222	9,930	0.422	0.356
medical condition treated (P2175)	210	3,604	0.741	0.481
topic's main category (P910)	207	22,243	0.304	0.773

Table 3
Model performance on the top 10 most common relations for event-related triples.

We are able to observe some large differences in performance between the two models for specific relations. For example, while CBR's performance is incredibly poor compared to AnyBURL for the `described_by_source` relation, it performs significantly better for the `topic's_main_category` relation. Of particular interest is their respective performance on the causal relations between events, where we see slightly better performance by CBR on both the `has_effect` and `has_cause` relations.

One of the reasons for this discrepancy in performance may be due to how CBR relies on effectively finding similar entities in the KG to base its predictions on. As our data curation method revolved around events with causal relations, it is reasonable to assume that adequate data has been collected on such entities to identify good "similar" entities for the case of the `has_effect` relation. On the other hand, many other relations for which CBR performed

poorly were most likely unable to find good matches, or there were insufficient alternatively paths through the KG to make good predictions. This especially is the case for any nodes that are on the outer edge of the 3-hop neighborhood which was used to collect our event dataset.

These results lead to an important consideration, both for data curation, and for performance analysis on data from Wikidata. When methods for link prediction rely on training data, it is necessary to consider how the methods used for data curation might lead to an imbalance of available data or completeness of data surrounding each entity. Additionally, when comparing performance between models, it is important to understand exactly how the different models perform better than each other. From our results, while AnyBURL shows superior performance for link prediction in general, analysis of performance for each relation would suggest that CBR is in fact superior for causal link prediction.

5.2.1. Performance of Inverse Relations

Among the best performing relations from Table 3, several pairs of inverse relations can be found (such as `has_cause` and `has_effect`, `has_part(s)` and `part_of`, and `P2176` and `P2175`). The inclusion of inverse relations have been criticized as a source of indirect data leakage in analyses of link prediction methods [21], suggesting that the presence of inverse relations artificially inflates the performance of link prediction models while limiting their usefulness in practical applications. For the task of event forecasting, this is an important point to consider because new events that we are trying to forecast will not have such inverse relations present in the KG.

To test this, we filter out inverse relations from the training data and re-run the AnyBURL model. We focus on assessing the performance for just the `has_effect` relation in two experiments – first, removing all inverse relations between entities present in `has_effect` triples in the test set, and second, removing all inverse relations among *all* entities in the training set (removing any inverse of a triple contained in the test set, and otherwise selecting which inverse to remove at random) to compare general performance.

We find that when removing inverse connections in the training triples, the performance of AnyBURL significantly decreases for this relation. When removing inverses of just the `has_effect` relation, AnyBURL's MRR for predicting `has_effect` links decreases from 0.282 in the original results to 0.072. When removing all inverses from the training data, the MRR further decreases to 0.057.

For the general training set, the MRR and Hits@10 decrease to 0.153 and 0.242, respectively, when all inverse relations are removed. This indicates that AnyBURL was able to effectively make use of the inverse relations contained in our dataset – further, it also indicates that our methods for data curation lead to a large number of inverse relations being included. We also extend the training of AnyBURL up to 10,000 seconds to observe if better performance can be achieved. With the 10-fold increase in training time, the model showed very small improvements, with the MRR increasing to 0.161 and Hits@10 to 0.252.

5.2.2. Patterns in Performance

In general, the performance of these rule-based models shows large variance between different relation types. Some factors that we believed might be influencing this performance were (1) the variety of entities connected by certain relations, and (2) the number of paths connecting entities in relations. Regarding the variety of entities, we mean this to indicate how many different tail entities a particular relation leads to – for example, the `continent` relation would only have 7 tail entities it leads to, while the `country` relation can connect to a much larger number of entities. In a trivial case, if a particular relation only ever leads to one entity, a model such as AnyBURL could learn a rule that always leads to the same entity. On the other hand, the number of paths connecting entities may be relevant to consider because the rules learned by AnyBURL are, essentially, paths through the KG. It might serve to reason that the number of such paths connecting entities leads to differences in performance – a smaller number of paths might indicate greater precision in the rules, or a larger number of paths might allow for a better chance of learning a relevant rule.
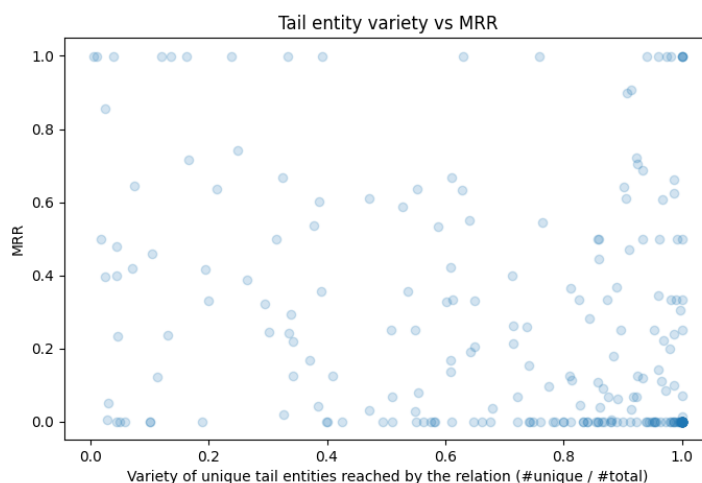

Figure 2: AnyBURL's MRR versus the variety of tail entities reached by event-related relations.

Figure 2 shows a scatterplot of AnyBURL's performance, where each point represents the model's performance for a single relation. The X-axis captures the variety of tail entities reached by the relation, where points further to the right have a greater variety of tail entities. We can see a cluster of relations with very poor performance at the bottom-right corner (i.e., relations which always lead to different tail entities). We also can observe that even relations with very little variety can show poor performance, as seen by some points on the bottom-left.

To consider the second point, Figure 3 shows a plot of AnyBURL's MRR for relations versus the number of paths connecting entities in the relations. Here, we limit the relations to only consider the top 20 best performing relations, where the relations occured at least 10 times in the test data. For each triple, the number of paths between the head and tail entity in the training data was counted, and the average such count for each triple of a particular relation is
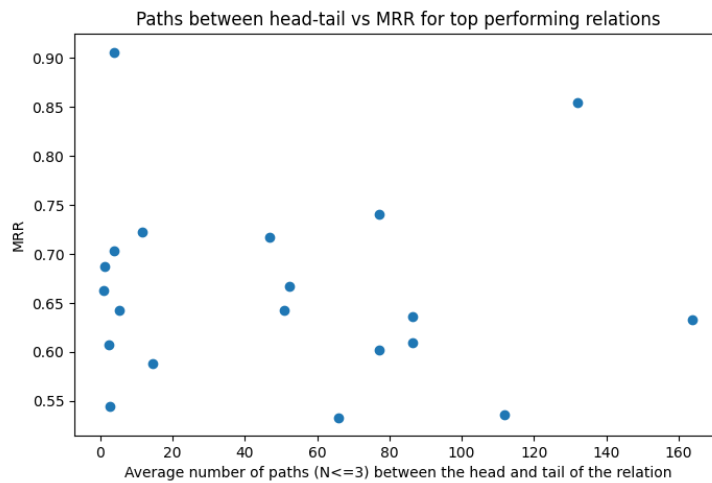
Figure 3: Top 20 best performing relations versus the number of paths connecting the head and tail entity in the training data.

shown. Here, we once again see no clear pattern in the performance – while the best performing relation has a low average number of paths connecting its head and tail entities, the second best performing relation has a relatively high number of connecting paths.
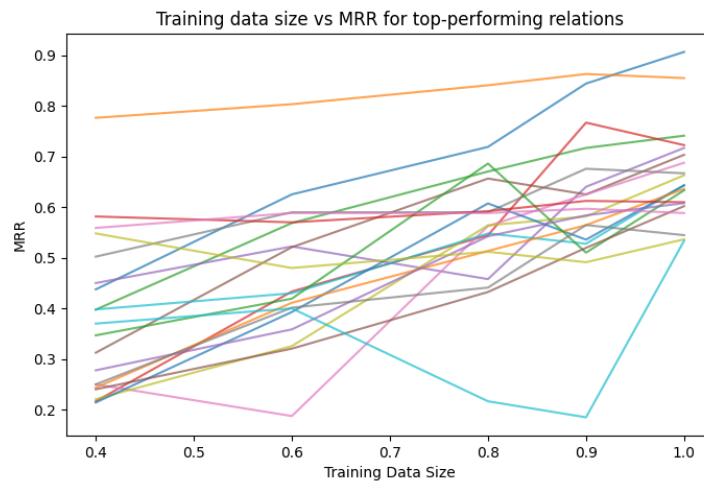

Figure 4: A comparison of the MRR for individual relation types vs the training data size, where 1.0 refers to the "full" training dataset, 0.9 refers to 90%, and so on.

5.3. Impact of Training Data Size

Lastly, we consider the impact of training data size on individual relations. While we saw an expected general pattern of decreasing performance for the overall evaluation, at the scope of individual relations we in fact see some interesting results. Figure 4 plots the performance of AnyBURL's prediction for 20 relations (once again selecting the top performing relations with over 10 triples). While we can see a general trend of increased performance as training data increases for many relations, for several relations we can see that increasing dataset size sometimes leads to decreased performance. This could be caused by a number of factors, as our decreased training data was selected randomly. For example, as AnyBURL relies on learning and mining rules, it is possible that some of the larger datasets included a triple that lead to an unreliable rule.

6. Conclusion

In this paper, we investigated utilizing link prediction methods over causal event-related knowledge in Wikidata, as a means of analysis of newsworthy events and event forecasting. In Wikidata, we found that the availability of event entities with causal relations was quite limited considering the large scale of the KG. In applying two rule-based link prediction methods – a case-based reasoning model and AnyBURL – we are able to observe limited success in link prediction. These models show wide variance in their performance across the various relations found in our curated dataset, with many trivial relations showing high performance while some relations could not be predicted at all by the models. We also can observe that increasing the amount of available data does not necessarily lead to improved performance in some experimental settings. We find that the presence of inverse relations heavily impacts the performance of the rule-based models, leading to nearly a 75% decrease in performance for relations such as has_effect. This shows that link prediction methods can provide a reliable solution for enriching event-related knowledge, but may have limited application in event forecasting.

Towards realizing the potential of using such data for more sophisticated event forecasting, a major challenge exists in determining how to collect data with a wider coverage of events as well as how to appropriately apply and evaluate models. Furthermore, without careful curation and analysis of test data, typical evaluation metrics that average together the performance over various relations can lead to models whose performance is inflated by over-performing on a number of relations. In the future, we plan to experiment with link prediction over Wikidata knowledge that is enriched through knowledge extraction from Wikipedia articles [1] as well as performing more thorough explorations into applying various link prediction models to causal KGs [22]. We hope to develop a robust link prediction framework that can reliably derive certain kinds of event-related relations, and contribute the outcome to Wikidata in the form of triples with explanations on how they have been derived.

References

[1] O. Hassanzadeh, Building a knowledge graph of events and consequences using wikidata, in: Proceedings of the 2nd Wikidata Workshop (Wikidata 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24, 2021, volume 2982, 2021.

[2] 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) 2724–2743.

[3] L. Zhao, Event prediction in the big data era: A systematic survey, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3450287. doi:10.1145/3450287.

[4] M. Wang, L. Qiu, X. Wang, A survey on knowledge graph embeddings for link prediction, Symmetry 13 (2021) 485. URL: https://www.mdpi.com/2073-8994/13/3/485.

[5] M. Zhang, Graph Neural Networks: Link Prediction, Springer Nature Singapore, 2022, pp. 195–223. URL: https://doi.org/10.1007/978-981-16-6054-2_10. doi:10.1007/978-981-16-6054-2_10.

[6] N. Jain, J.-C. Kalo, W.-T. Balke, R. Krestel, Do embeddings actually capture knowledge graph semantics?, in: ESWC, 2021.

[7] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Trans. Knowl. Discov. Data 15 (2021). URL: https://doi.org/10.1145/3424672. doi:10.1145/3424672.

[8] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, 2015. doi:10.18653/v1/W15-4007.

[9] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: AAAI, 2018.

[10] L. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, Amie: Association rule mining under incomplete evidence in ontological knowledge bases, 2013, pp. 413–422. doi:10.1145/2488388.2488425.

[11] L. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, Fast rule mining in ontological knowledge bases with amie+, The VLDB Journal 24 (2015). doi:10.1007/s00778-015-0394-1.

[12] J. Lajus, L. Galárraga, F. Suchanek, Fast and exact rule mining with amie 3, in: A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, M. Cochez (Eds.), The Semantic Web, Springer International Publishing, Cham, 2020, pp. 36–52.

[13] A. Sadeghian, M. Armandpour, P. Ding, D. Z. Wang, Drum: End-to-end differentiable rule mining on knowledge graphs, in: NeurIPS, 2019.

[14] S. Ortona, V. V. Meduri, P. Papotti, Robust discovery of positive and negative rules in knowledge bases, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), 2018, pp. 1168–1179. doi:10.1109/ICDE.2018.00108.

[15] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, H. Stuckenschmidt, Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion, in: ISWC, volume 11136, Springer, 2018, pp. 3–20. URL: https://doi.org/10.1007/978-3-030-00671-6_1. doi:10.1007/978-3-030-00671-6_1.

[16] C. Meilicke, M. W. Chekol, D. Ruffinelli, H. Stuckenschmidt, Anytime bottom-up rule learning for knowledge graph completion, in: IJCAI, 2019.

[17] S. Ott, C. Meilicke, M. Samwald, SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models, in: 3rd Conference on Automated Knowledge Base Construction, 2021. URL: https://openreview.net/forum?id=jCt9S_3w_S9.

[18] R. Das, A. Godbole, S. Dhuliawala, M. Zaheer, A. McCallum, A simple approach to case-based reasoning in knowledge bases, in: D. Das, H. Hajishirzi, A. McCallum, S. Singh (Eds.), Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020, 2020. URL: https://doi.org/10.24432/C52S3K. doi:10.24432/C52S3K.

[19] R. Das, A. Godbole, N. Monath, M. Zaheer, A. McCallum, Probabilistic case-based reasoning in knowledge bases, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, Association for Computational Linguistics, 2020, pp. 4752–4765. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.427. doi:10.18653/v1/2020.findings-emnlp.427.

[20] R. Das, M. Zaheer, D. Thai, A. Godbole, E. Perez, J. Y. Lee, L. Tan, L. Polymenakos, A. McCallum, Case-based reasoning for natural language queries over knowledge bases, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 9594–9611. URL: https://aclanthology.org/2021.emnlp-main.755. doi:10.18653/v1/2021.emnlp-main.755.

[21] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, C. Li, Realistic re-evaluation of knowledge graph completion methods: An experimental study, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1995–2010. URL: https://doi.org/10.1145/3318464.3380599. doi:10.1145/3318464.3380599.

[22] A. Khatiwada, S. Shirai, K. Srinivas, O. Hassanzadeh, Knowledge graph embeddings for causal relation prediction, in: Workshop on Deep Learning for Knowledge Graphs (DL4KG), 2022.