Do Temporal Knowledge Graph Embedding Models Learn or Memorize Shortcuts?

Jiaxin Pan
University of Stuttgart
jiaxin.pan@ki.uni-stuttgart.de

Mojtaba Nayyeri
University of Stuttgart
mojtaba.nayyeri@ki.uni-stuttgart.de

Yinan Li
University of Stuttgart
yinan9721@gmail.com

Steffen Staab
University of Stuttgart, University of Southampton
steffen.staab@ki.uni-stuttgart.de

Abstract

Temporal Knowledge Graph Embedding models predict missing facts in temporal knowledge graphs. Previous work on static knowledge graph embedding models has revealed that KGE models utilize shortcuts in test set leakage to achieve high performance. In this work, we show that a similar test set leakage problem exists in widely used temporal knowledge graph datasets ICEWS14 and ICEWS05-15. We propose a naive rule-based model that can achieve start-of-the-art results on both datasets without a deep-learning process. Following this consideration, we construct two more challenging datasets for the evaluation of TKGEs[1].

1 Introduction

Temporal Knowledge Graphs (TKG) represent facts in the real world with their time information to indicate if a triple exists during a time interval or at a time point. Facts are stored in the quadruple form (s, p, o, τ), where s and o denote the head entity and tail entity a.k.a as 'nodes' in TKGs. p denotes the relation between entities, a.k.a as 'edges' and τ specifies the fact is valid at this time. To deal with the incompleteness problem in temporal knowledge graphs, temporal knowledge graph embedding (TKGE) models adopt link prediction tasks and predict missing facts based on observed patterns in temporal knowledge graphs. Although impressive progress on performance gain has been achieved by models from TTransE (Leblay and Chekol, 2018) to TLT-KGE (Zhang et al., 2022), fine-grained analysis of experiment results is still unexplored. TKGs exhibit various kinds of patterns in nature. For instance, (France, host a visit, Angela Merkel, 2014-02-03) reveals (Angela Merkel, make a visit, France, 2014-02-03) as the relation "host a visit" and "make a visit" are inverse relations.

In static knowledge graph datasets, Toutanova and Chen (2015); Dettmers et al. (2018) discovered that the FB15k and WN18 datasets suffer from test set leakage because of inverse relations, i.e., 80.9% and 94.0% test set triples in FB15k and WN18 could find their inverse relations in the train set Toutanova and Chen (2015). Dettmers et al. (2018) proposes a simple rule-based model based on pure inverse patterns of relations and achieves state-of-the-art results on both WN18 and FB15k. To mitigate the test set leakage problem, FB15k-237 and WN18RR are constructed by keeping only one of a set of inverse or duplicate relations. However, a similar test set leakage problem has not been studied in temporal knowledge graph datasets yet. Compared to static knowledge graphs, the popular temporal knowledge graphs ICEWS14 and ICEWS05-15 exhibit two different characteristics: 1) Patterns in TKGs are related to temporal information. 2) In TKGs, the same event may happen repeatedly historically. To investigate the test leakage problem in these two datasets, we construct

[1]The datasets and code are provided in https://github.com/NacyNiko/naive_rule

Temporal Graph Learning Workshop @ NeurIPS 2023, New Orleans.

a naive rule-based model based on the aforementioned characteristics which achieves comparable results with state-of-the-art models. Subsequent analysis demonstrates that current models heavily rely on symmetry/inverse patterns and repeated facts in TKGs to make predictions. We construct two more challenging datasets from ICEWS14 and ICEWS05-15 and test state-of-the-art models on the new datasets. The vast decrease in performance shows that more sophisticated forms of inference such as multi-hop query are severely needed for temporal knowledge graph completion datasets.

2 Related Work

Previous test set leakage analyses only focus on static knowledge graph models. (Toutanova and Chen, 2015) detect a huge number of near-duplicate and inverse relations exist in FB15k and WN18 datasets. They construct a new dataset FB15k-237 by removing the aforementioned relations from FB15K which is difficult for models based on simple observed features. Following this work, (Dettmers et al., 2018) systematically investigated the influence of reported inverse relation leakage and found that a single rule-based model could achieve comparable performance with state-of-the-art models on FB15k and WN18. To mitigate this problem, they designed a new dataset WN18RR from WN18 similar to (Toutanova and Chen, 2015). However, no analysis of the test set leakage problem has been conducted on temporal knowledge graph datasets. (Han et al., 2021) classify temporal knowledge graph embedding methods into two types: (1) timestamp embedding methods in which entity embeddings and time embedding are represented separately. (2) time-dependent entity embedding methods which entity embeddings evolve over time. They discovered that when trained appropriately, timestamp embedding methods could achieve comparable or even better performance than time-dependent entity embedding methods. Our work replenishes their observations with the discovery that shortcuts in the TKG datasets are not so relevant to time.

3 A Naive Rule-based Model for Temporal Knowledge Graph Completion

We consider four simple rules to infer the missing entities in the test set: 1) Static symmetry/inverse pattern detection, 2) Dynamic symmetry/inverse pattern detection, 3) Repeated facts detection, and 4) Connected entities detection. We will explain our proposed approach in the following. For further details, we refer to the Algorithm 1 in the Appendix.

3.1 Patterns in Temporal Knowledge Graph

Patterns such as symmetry and inverse in TKGs have been studied in previous work Chen et al. (2022); Xu et al. (2020) and could serve as direct hints for link prediction. For example, if *Angela Merkel Consult Barack Obama* on *2014/08/29*, we could easily infer that *Barack Obama Consult Angela Merkel* on *2014/08/29* also holds as *Consult* is a symmetric relation. We generalize and go beyond these approaches and consider *static temporal patterns* and *dynamic temporal patterns*. If a pattern holds *regardless of time information* as in traditional knowledge graphs, we call it a *static temporal pattern*. Otherwise, we call it a *dynamic temporal pattern*. Figure 1 shows examples of patterns and the detection procedure of defined patterns is provided in Appendix A.

Definition 1 *A temporal relation p is static symmetric at all points in time iff* $\forall s, o, \tau : (s, p, o, \tau) \rightarrow (o, p, s, \tau)$.

Definition 2 *A temporal relation p_1 is the static inverse of temporal relation p_2 at all points in time iff* $\forall s, o, \tau : (s, p_1, o, \tau) \rightarrow (o, p_2, s, \tau)$.

Definition 3 *A temporal relation p is temporal symmetric iff* $\forall s, o, \tau_1 : \exists \tau_2 : (s, p, o, \tau_1) \rightarrow (o, p, s, \tau_2)$.

Definition 4 *A relation p_1 at time τ_1 is the dynamic inverse of relation p_2 at time τ_2 iff* $\forall s, o : \exists \tau_1, \tau_2 : (s, p_1, o, \tau_1) \rightarrow (o, p_2, s, \tau_2)$.

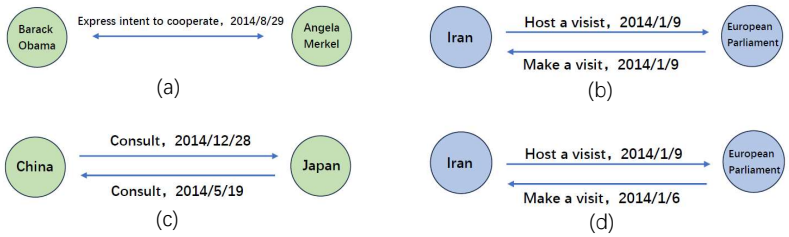

Figure 1: Patterns in temporal knowledge graphs. Sub-figures (a), (b), (c), and (d) present examples of static symmetric, static inverse, dynamic symmetric, and dynamic inverse patterns respectively.

3.2 Naive Rule-based Model

Rule 1:Static Symmetry/inverse Pattern Detection For query $(s, p, ?, \tau)$[2], if p is static symmetry or inverse relation, we detect if its reversed quadruple $(?, p, s, \tau)$ or $(?, p^{-1}, s, \tau)$ exists in the train set and return a sorted list of subject entities based on their occurrence frequency. Occurrence frequency here serves as an intuitive signal for interaction probability.

Rule 2:Dynamic Symmetry/inverse Pattern Detection If p is dynamic symmetry or inverse relation, we follow a similar procedure as in Rule 1. However, as time is flexible for Rule 2, we detect reversed quadruple $(?, p, s, -)$ or $(?, p^{-1}, s, -)$. Here $-$ is a placeholder any detected time.

Rule 3: Repeated Facts In TKGs, some facts appear to happen many times during a long period. For example, *China* may *Consult Japan* many times during 2005-2015. Previous work Zhu et al. (2021) utilizes this feature for the temporal knowledge graph extrapolation task and achieves good performance. However, this feature has not been studied for the temporal knowledge graph interpolation task yet. In Rule 3, for missing facts $(s, p, ?, \tau)$, we relax the time constraint to $(s, p, ?, -)$ and obtain the object entity set from training graph G_{train}. A sorted entity list based on occurrence frequency is returned as facts happening frequently tend to happen again in the future.

Rule 4: Connected Entities In rule 4, we relax the constraint of relation and assume entities showing more co-occurrences by means of all relations tend to co-occur again in the test set. Therefore, for missing facts $(s, p, ?, \tau)$, we obtain the object entity set from $(s, -, ?, -)$ in G_{train} and return a sorted object entity list based on occurrence frequency.

Ranking Strategy From Rule 1 to 4, the constraints for searching quadruples are getting more relaxed and inaccurate. Therefore, we merge the sorted ranking list E_{rank} from Rule 1 to 4. At test time, we check if the correct entity e_{target} for $(s, p, ?, \tau)$ is in E_{rank}. If yes, the rank corresponding to e_{target} in E_{rank} will be the final rank. Otherwise, we select a random rank between $|E_{rank}| + 1$ and $|E|$. It's noted that we remove repeated entities when building an entity list for each rule.

4 Experiments

Datasets ICEWS14 (Garcia-Duran et al., 2018) and ICEWS05-15 (Garcia-Duran et al., 2018) are two popular temporal knowledge graph benchmark datasets. They are two subset datasets from the Integrated Conflict EarlyWarning System (ICEWS)(O'brien, 2010), which contain news facts in 2014 and between 2005 and 2015 respectively. Table 3 in the Appendix shows the details of the datasets.

Baselines We select state-of-the-art TKGE models, TTransE(Leblay and Chekol, 2018), TA-DistMult(Garcia-Duran et al., 2018), TeRo(Xu et al., 2020), T(NT)ComplEx(Lacroix et al., 2019), RotateQVS(Chen et al., 2022) and TLT-KGE(Zhang et al., 2022) as baselines.

Evaluation Metrics We adopt the link prediction task for evaluation. Link prediction infers the missing entities for incomplete facts. During the test step, we follow the procedure of (Xu et al.,

[2]For simplicity, we only use query $(s, p, ?, \tau)$ to represent incomplete quadruple in the following part

Table 1: Link prediction results on ICEWS14 and ICEWS05-15. Naive-Rules show a stable performance in 10 runs with variances on all metrics less than 0.0001.

Model	ICEWS14				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TTransE(Leblay and Chekol, 2018)	25.5	7.4	-	60.1	27.1	8.4	-	
TADistMult(Garcia-Duran et al., 2018)	47.7	36.3	-	68.6	47.4	34.6	-	72.8
TeRo(Xu et al., 2020)	56.2	46.8	62.1	73.2	58.6	46.9	66.8	79.5
RotateQVS(Chen et al., 2022)	59.1	50.7	64.2	75.4	63.3	52.9	70.9	81.3
TComplEx(Lacroix et al., 2019)	61.9	54.2	66.1	76.7	66.5	58.3	71.6	81.1
TNTComplEx(Lacroix et al., 2019)	60.7	51.9	65.9	77.2	66.6	58.3	71.8	81.7
TLT-KGE(dim=100)(Zhang et al., 2022)	54.9	46.7	59.2	70.9	58.4	49.9	63.0	74.6
TLT-KGE(dim=1200)(Zhang et al., 2022)	63.0	54.9	67.8	77.7	68.6	60.7	73.5	83.1
Naive-Rules	57.3	49.4	61.5	72.6	58.5	49.6	63.4	76.0

2020) to generate candidate quadruples. From a test quadruple (s, p, o, τ), we replace s with $\bar{s} \in \mathcal{E}$ and o with $\bar{o} \in \mathcal{E}$ to get candidate quadruples $(s, p, \bar{o}, \tau) \cup (\bar{s}, p, o, \tau)$. All candidate quadruples will be ranked by their scores using a time-aware filtering strategy (Goel et al., 2020). We evaluate our models with five metrics: Mean Rank(MR), Mean Reciprocal Rank (MRR), the mean of the reciprocals of predicted ranks of correct quadruples, and Hits@(1/3/10), the percentage of ranks not higher than 1/3/10. For MR, the smaller the better. For others, the higher the better.

5 Results

Main Result Table 1 presents the link prediction results of baselines and our proposed naive-rules model. Strikingly, without using any embeddings or deep learning, Naive Rules achieves comparable results with state-of-the-art methods on many different metrics for both ICEWS14 and ICEWS05-15. It validates our assumption that shortcuts also exist in temporal knowledge graph datasets. To a large extent, the performance of current neural network models could be attained by simple patterns rather than complicated patterns in the TKGs. Moreover, the result of TLT-KGE(dim=100) and TLT-KGE(dim=1200) shows that current neural network models require high embedding dimensions and massive computing resources to transcend the performance of naive rules.

Re-dividing Dataset by Complete Symmetry or Inverse Pairs ICEWS14 and ICEWS05-15 present test data leakage from 1) symmetry relations 2) inverse relations. In static knowledge graphs, Toutanova and Chen (2015) and Dettmers et al. (2018) remove one relation from the inverse relation pairs to construct FB15k237 and WN18RR. For example, when detecting the relation /award/award_nominee is inverse of /award_nominee/award, only one of the relation will be kept. However, this strategy does not work for ICEWS14 and ICEWS05-15 for two reasons: 1) Symmetry relations such as *Consult* form the pair by itself and could not be removed. 2) Relations in TKGs have closer connections compared to relations in KGs. For example, *Discuss by Telephone* is a strong signal for *Consult*. Simply removing relations may devastate the inference path between quadruples. Therefore, we propose a novel re-division strategy for symmetry and inverse relations. As shown in Figure 2, ICEWS14 and ICEWS05-15 divide the quadruple randomly to train/valid/test sets. Consequently, the edges which form a complete symmetry or inverse pair may appear in different subsets and the model could easily infer the missing link in valid or test set by the symmetry or inverse quadruple known in the train set. To fix this problem, we divide the two quadruples which form a complete symmetry or inverse pair simultaneously to train/valid/test sets. In this way, models have to predict the whole symmetry or inverse pair from evidence provided by other quadruples. Table 3 presents the details of the constructed datasets ICEWS14RR and ICEWS05-15RR. We argue that this re-division strategy is more realistic and universal than the removing strategy by (Toutanova and Chen, 2015) and Dettmers et al. (2018). It can also be extended to static KG datasets.

Results on ICEWS14RR and ICEWS05-15RR Table 2 presents the link prediction results on more challenging dataset ICEWS14RR and ICEWS05-15RR. As expected, the performance of all models dropped a lot. Moreover, the performance gap between the rule-based model and neural network model enlarges marginally. This demonstrates that existing models highly rely on easy patterns for prediction and only possess weak ability to do multi-step inference such as discover unobserved relations between entities from other known facts and then infer its symmetry or inverse pair. Interestingly, T(NT)ComplEx performs much better than TLT-KGE on ICEWS14RR and

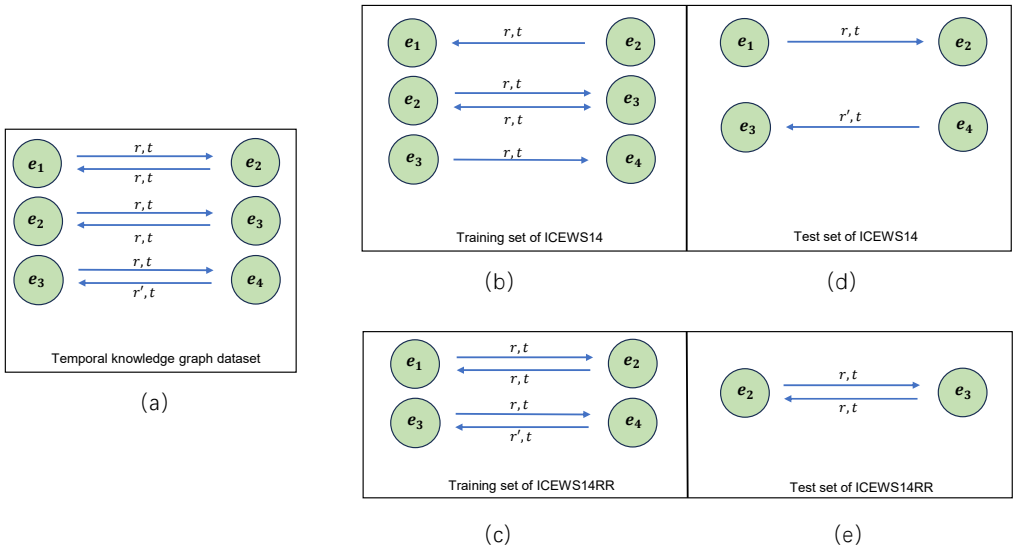

Figure 2: In Sub-figure(a), e_1 and e_2, e_2 and e_3 form two pairs of symmetry relation. e_3 and e_4 form one pair of inverse relation. As shown by sub-figure (b) and (d), ICEWS14 randomly divide training set and test set based on edges between entities. Therefore, TKGE models could easily complete the missing edges in test set through known edges in training set. In contrary, our proposed new division strategy will assign the entire pairs to training set or test set as shown in sub-figure (c) and (e). Therefore, models need to infer the connection between entities from other facts and capture the temporal patterns at the same time.

ICEWS05-15RR although TLT-KGE is the best model on ICEWS14 and ICEWS05-15, which validates the new challenging datasets could provide new insights of TKGC models.

Table 2: Link prediction results on ICEWS14RR, ICEWS05-15RR. The decrease of performance compared to original datasets is shown in (-).

Model	ICEWS14RR				ICEWS05-15RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TComplEx	48.4(-13.5)	38.6	54.0	66.7	53.3(-13.2)	43.4	59.1	71.9
TNTComplEx	48.2(-12.5)	37.9	54.1	67.4	54.3(-12.3)	44.1	60.3	73.4
TLT-KGE	47.3(-15.7)	36.7	53.2	66.8	51.9(-16.7)	41.3	57.9	72.0
Naive-Rules	41.6(-15.7)	32.1	48.5	62.9	46.1(-17.3)	34.1	50.1	65.4

6 Conclusion

This work studies the test set leakage problem in temporal knowledge graph datasets. We construct a naive rule-based model and achieve comparable performance with state-of-the-art models on ICEWS14 and ICEWS05-15. To alleviate the problem that existing neural network models heavily rely on symmetry or inverse patterns to make prediction, we construct two more challenging datasets from ICEWS14 and ICEWS05-15. In the future, we plan to construct temporal knowledge graph datasets with more inference types.

Acknowledgement

This research was funded by the German Research Foundation (DFG) via grant agreement number STA 572/18-1 (Open Argument Mining) and the German Federal Ministry for Economic Affairs and Climate Action under Grant Agreement Number 01MK20008F (Service-Meister). We would also like to thank the valuable advice from Daniel Hernández and San San.

References

Kai Chen, Ye Wang, Yitong Li, and Aiping Li. 2022. Rotateqvs: Representing temporal information as rotations in quaternion vector space for temporal knowledge graph completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5843–5857.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3988–3995.

Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021. Time-dependent entity embedding is not all you need: A re-evaluation of temporal knowledge graph completion models under a unified framework. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8104–8118, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2019. Tensor decompositions for temporal knowledge base completion. In *International Conference on Learning Representations*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776.

Sean P O'brien. 2010. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International studies review*, 12(1):87–104.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593.

Fuwei Zhang, Zhao Zhang, Xiang Ao, Fuzhen Zhuang, Yongjun Xu, and Qing He. 2022. Along the time: Timeline-traced embedding for temporal knowledge graph completion. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2529–2538.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4732–4740.

Table 3: Statistics for ICEWS14, ICEWS05-15, GDELT, ICEWS14RR and ICEWS05-15RR.

Dataset	ICEWS14	ICEWS05-15	ICEWS14RR	ICEWS05-15RR
Entities	7,128	10,488	7128	10,488
Relations	230	251	230	251
Times	365	4017	365	4017
Train	72,826	368,962	72,581	369,060
Validation	8,941	46,275	9,073	46,134
Test	8,963	46,092	9,076	46,135

A Symmetry/Inverse Relation Detection

To predict missing entities based on the defined patterns, we need to detect symmetry and inverse relations in TKGs. We calculated the probability of each relation exhibiting a particular pattern on the training set. For a given pattern $h \in H$ and relation $p \in R$, if the proportion of instances that satisfy the pattern and are related to relation p in the training exceeds a predefined threshold, this relation will be considered as holding pattern h. The proportion is calculated as:

$$\mathbb{P}_{p,h} = \frac{|satisfies((s,p,o,\tau),h)|}{|(s,p,o,\tau)|} \tag{1}$$

where $(s,p,o,\tau) \in G_{train}$. In the experiment, we set the threshold as 0.5.

Algorithm 1: Naive Rule-based Model

Data: G: temporal knowledge graph, $(s,p,?,\tau)$: incomplete quadruple from G_{valid} or G_{test}, e_{target}: correct entity for the incomplete quadruple, R_s, R_i: symmetry relation set and inverse relation set, E: Entity set, R: Relation set, T: Time set.

Result: $rank$: The ranking of e_{target}

if p *in* R_s *or* R_i **then**
  if p *in* R_s **then**
    $q = p$
  else
    $q = p^{-1}$
  end
  $E_{rule1} = \{e | e \in E, (e,q,s,\tau) \in G_{train}\}$;
  Sort E_{rule1} based on the occurrence frequency of (e,q,s,τ) in G_{train};
  $E_{rule2} = \{e | e \in E, e \notin E_{rule1}, \tau' \in \mathrm{T}, (e,q,s,\tau') \in G_{train}\}$;
  Sort E_{rule2} based on the occurrence frequency of (e,q,s,τ') in G_{train}
end
$E_{rule3} = \{e | e \in E, e \notin E_{rule1}, e \notin E_{rule2}, \tau' \in \mathrm{T}, (s,p,e,\tau') \in G_{train}\}$;
Sort E_{rule3} based on the occurrence frequency of (s,p,e,τ') in G_{train};
$E_{rule4} = \{e | e \in E, e \notin E_{rule1}, e \notin E_{rule2}, e \notin E_{rule3}, \tau' \in \mathrm{T}, p' \in R, (s,p',e,\tau') \in G_{train}\}$;
Sort E_{rule4} based on the occurrence frequency of (s,p',e,τ') in G_{train};
Concat sorted list $E_{rank} = [E_{rule1}, E_{rule2}, E_{rule3}, E_{rule4}]$;
if $e_{target} \in E_{rank}$ **then**
  $rank$ = the index of e_{target} in E_{rank} +1
else
  $rank$ = a random number between $|E_{rank}| + 1$ and $|E|$
end