Conservative Knowledge Graph Completion Using Semantically Enriched Information

Anonymous ACL submission

Abstract

In this paper, we present a novel conservative completion approach for Knowledge Graphs (KGs), designed to address the shortcomings of current knowledge completion methods, particularly their failure to guarantee the accuracy of completion results. Our method uniquely utilizes semantically enriched information inherent in KGs to construct a reasoner based on description logic. By integrating this reasoner with Link Prediction (LP) models, we ensure the correctness of the knowledge completion. Experimental findings show that a substantial proportion of predictions from diverse LP models can undergo conservative completion. Additionally, the volume of conservatively completable results escalates with the increase in semantically enriched information in the KGs.

1 Introduction

In the field of AI, KGs serve as structured frameworks for representing real-world information. In these graphs, nodes represent entities such as individuals, locations, and concepts, while edges represent the relationships between pairs of entities. These relationships are further characterized by labels that specify their nature. Despite the heterogeneous nature of large-scale KGs like DBPedia (Lehmann et al., 2015), Freebase (Bollacker et al., 2008), WordNet (Miller, 1995), YAGO (Suchanek et al., 2008), Wikidata (Vrandecic and Krötzsch, 2014), and Google KG and Microsoft Satori (Noy et al., 2019), a common issue of data incompleteness persists (Min et al., 2013; Dong et al., 2014; Galárraga et al., 2017). Studies highlight significant gaps in these datasets. For example, in FreeBase, over 70% of person entities lack recorded birthplaces, and over 99% have no known ethnicity. In Wikidata, the coverage of relational data was notably sparse, with records of paternal links for only 2% of all individuals in the knowledge base (KB) — a stark contrast to the universal reality of parentage. Similarly, DBpedia stores merely 6 recipients of the prestigious Dijkstra Prize, whereas the actual count stands at 35. This trend of underrepresentation extends to other dimensions as well, as evidenced by YAGO's data suggesting an implausibly low average number of children per person at 0.02. A broader examination reveals a pervasive pattern across popular KBs: between 69% and 99% of instances in these KBs are missing at least one common attribute within their category (Suchanek et al., 2011).

As pointed by (Peng et al., 2023), the incompleteness of KGs introduces a multifaceted set of challenges that significantly impede their effectiveness and reliability. For instance, such incompleteness not only undermines the accuracy of query results, leading to potential misinformation (Pflueger et al., 2022) and misguided decision-making (Yu et al., 2023) but also presents obstacles in natural language processing tasks, impairing the system's ability to understand context and provide accurate responses (Zhang et al., 2023; Li et al., 2019). In recommendation systems, this leads to poor personalization, failing to accurately reflect user preferences (Cao et al., 2019). User trust diminishes when encountering consistent data gaps, affecting the adoption and efficacy of KG-driven systems. Moreover, the challenge of integrating incomplete KGs with other data systems complicates data alignment, while the ongoing efforts required for data curation and validation increase maintenance costs (Huang et al., 2022b). These limitations underscore the necessity for rigorous approaches to address and mitigate the impact of incompleteness in KGs, ensuring their full potential is harnessed in AI and other technical domains.

The pursuit of enhancing KGs has driven researchers to develop a wide range of techniques for augmenting these graphs with absent information, a task referred to as KG Completion (KGC) (Bordes et al., 2013; Paulheim, 2017; Schlichtkrull et al., 2018; Sun et al., 2019). KGC is essentially a learning endeavor aimed at enriching a KG by incorporating missing triples that are likely to hold. This expansion can be achieved either by integrating new facts from external data sources or by inferring the missing information from the existing structure and data composition of the KG. This paper primarily focuses on the latter strategy, known as Link Prediction (LP) (Liben-Nowell and Kleinberg, 2007). Specifically, LP addresses queries of the form $(h, r, ?)$ or $(?, r, t)$, where one entity is unknown. In this context, the former query type aims at tail prediction, and the latter focuses on head prediction.

In an LP model, irrespective of whether external knowledge is employed in training, the model assigns scores to all potential prediction results. Performance evaluation is based on how the actual data score ranks among these possibilities. This rank-based method reflects, to some extent, the model's learning ability from the data. However, under the open-world assumption, this method merely indicates if the correct result ranks within the top-k predictions, without assessing the accuracy of each individual prediction. This leads to the inclusion of uncertain predictions in the KG. To address this issue, (Pezeshkpour et al., 2020) introduced the YAGO3-TC dataset, an extension of YAGO3-10(Mahdisoltani et al., 2014), incorporating both positive and negative examples. The limitation of YAGO3-TC is its reliance on manual completion methods, which hampers its wider applicability.

The KG itself contains solely the information regarding relationships between entities. However, Yet, more comprehensive forms of information can be superimposed on it. For example, YAGO3 incorporates details about the transitivity of relationships between entities, entity types, and hierarchical relationships between types. In this paper, we introduce a novel *Conservative KG Completion* (Conservative KGC) approach for KGC and LP tasks. This approach diverges from those that convert external knowledge into embeddings for training. Instead, we utilize external knowledge like YAGO3-TC as a verifier, applying entailment verification to ensure the accuracy of prediction results. Specifically, each prediction is assessed as a potential fact, akin to an ABox assertion, with the external KB acting as the ontology (i.e., TBox axioms). We check if these assertions lead to any inconsistencies in the KB. Should inconsistencies arise, we exclude high-scoring facts from the KG, regardless of their score. In contrast, lower-scoring facts that are supported by the external KB are included because their correctness is verified. Our method is careful to add only verified, correct facts, assuming that the KB represents the ground truth. This cautious and accuracy-focused approach is why it is termed as Conservative KGC.

This paper presents a novel approach to KG completion, focusing on the conservative completion of KGs using LP models. Our method emphasizes the importance of entailment checks in verifying the correctness of predicted results, distinguishing between conservative and non-conservative completions. We employ five LP models, including both embedding-based (ComplEx, HAKE, MEIM, and ConEx) and rule-based (AnyBURL) models, to generate candidate predictions. The performance of these models is evaluated in terms of the number of correct predictions, revealing interesting insights into the relationship between standard evaluation metrics like Mean Reciprocal Rank (MRR) and Hits@K, and the actual accuracy of predictions. Through experimental analysis on widely used datasets like YAGO3-10 and Radiology Lexicon (RadLex), both in their RDFS and OWL forms, we demonstrate that the number of complements increases with the expansion of KG information. This paper also delves into the logical equivalence of KGs and their conservative completions, highlighting the potential inconsistencies and detrimental effects of non-conservative completions on model learning. Our findings suggest that while high-performance scores in traditional metrics are valuable, they do not necessarily correlate with the accuracy of KG completion. This underlines the significance of conservative completion, which ensures the logical consistency and reliability of the augmented KG.

Due to space constraints, the introduction to related LP models has been included in the Appendix.

2 Preliminaries

Existing LP models can be broadly classified into two types: embedding-based and rule-based.

Embedding-Based Models: In these models, each entity and relation is embedded into a vector space. Let $\mathbf{e}$ and $\mathbf{r}$ denote the embeddings of an entity e and a relation r, respectively. These models employ a scoring function ϕ to estimate the probability of a fact's validity. For any given fact (h, r, t), a higher $\phi(\mathbf{h}, \mathbf{r}, \mathbf{t})$ score suggests a greater likelihood of the fact being true. To predict missing entities in facts represented as $(h, r, ?)$ or $(?, r, t)$, the model computes:

$$h = \underset{e}{\operatorname{argmax}}\, \phi(\mathbf{e}, \mathbf{r}, \mathbf{t}),\ (e, r, t) \notin G$$

$$t = \underset{e}{\operatorname{argmax}}\, \phi(\mathbf{h}, \mathbf{r}, \mathbf{e}),\ (h, r, e) \notin G$$

Crucially, the model is calibrated to discover unknown facts, necessitating the filtering of these predicted results

Rule-Based Models: These models aim to learn specific patterns of rules from existing facts, which can be used for prediction. For instance, given a rule like speaks$(p, l) \leftarrow$ lives(p, c), lang(c, l), where p, c, and l are variables representing entities, the model can infer new facts. If it knows lives(Tom, NewYork) and lang(NewYork, English), it would predict speaks(Tom, English). These inferred facts are scored based on the confidence level of the underlying rules. For queries such as $(h, r, ?)$ or $(?, r, t)$, rule-based models offer probable predictions absent in the KG, with $\psi(h, r, t)$ representing the rule's confidence score:

$$h = \underset{e}{\operatorname{argmax}}\, \psi(e, r, t),\ (e, r, t) \notin G$$

$$t = \underset{e}{\operatorname{argmax}}\, \psi(h, r, e),\ (h, r, e) \notin G$$

Interestingly, despite their distinct methodologies, these two approaches share a striking similarity in their formal structures. If we abstract the LP model as a black box and ignore how it scores the predicted results, we can represent the predictions in a unified manner by directly using the expression of rule-based models.

Expanding our focus beyond the single most likely result to encompass the top k results, our expression adapts as follows:

$$h_1, ..., h_k = \text{argtopk}_e \psi(e, r, t),\ (e, r, t) \notin G$$

$$t_1, ..., t_k = \text{argtopk}_e \psi(h, r, e),\ (h, r, e) \notin G$$

Here, argtopk entails sorting all potential predictions by their likelihood in descending order and extracting the top k. This is particularly relevant for non-functional relations, where multiple valid predictions may exist for a single query. For instance, in the query *(Chatou, isLocatedIn, ?)*, both *France* and *Europe* are correct results.

Due to space constraints, the introduction to Description Logic $\mathcal{ALC}$ has been deferred to the Appendix.

3 Issues in Existing KG Completion

Despite their strong performance in rank-based metrics, LP models face practical application challenges. This section focuses on the correctness of predictions. Our analysis reveals two key insights: the top-scoring predictions are not always correct, and lower-scoring predictions can also be correct.

3.1 Uncertainty of Highest Prediction

Under the open world assumption, the limited knowledge available impedes definitive judgments about the correctness of predicted results in LP models. This necessitates a cautious approach, where predicted results should not be added to the existing KG without proper verification. Nonetheless, the richness of TBox knowledge in datasets like YAGO3 enhances our capability to evaluate predictions. YAGO3's TBox knowledge encompasses various types, including:

- $C \sqsubseteq D$ (e.g., books as a subset of artifacts).
- Dom(r) (e.g., the domain of 'hasChild' is person).
- Range(r) (e.g., the range of 'hasCapital' is city).
- Trans(r) (e.g., isLocatedIn is transitive).
- $C \sqcap D \sqsubseteq \bot$ (e.g., person and city are mutually exclusive).

This comprehensive KB aids in assessing the correctness of predictions. For instance, with the relation *isLocatedIn*, we identify *PermanentlyLocatedEntity* as its domain and *GeoEntity* as its range. Knowing that *GeoEntity* and *Organization* are disjoint, a prediction like *(Ann_Arbor, isLocatedIn, University_of_Michigan)* is incorrect, as *University_of_Michigan* is categorized as an organization.

To quantify incorrect predictions, we modified the YAGO3-10 test set by removing tail entities, creating a prediction set. We then used three models to make predictions on this set, evaluating their correctness. The results, presented in Table 1, reveal error occurrences in each model's predictions.

While the number of incorrect predictions in LP models may be small, specific properties of relations, such as the transitivity of *isLocatedIn*, can amplify these inaccuracies. For instance, if the *isLocatedIn* relations are established as *(A, isLocatedIn, B)* and *(B, isLocatedIn, C)*, we logically

Model	#Incorrect Predictions
AnyBURL	1
ComplEx	35
HAKE	5

Table 1: Number of Incorrect Predictions

infer *(A, isLocatedIn, C)*. However, if either of the initial relations is incorrect, this leads to a propagated error in the inferred relations, which we term a “hidden error”. As the volume of existing knowledge grows, so does the potential for such hidden errors. In the context of the YAGO3-10 dataset, which contains 89,524 *isLocatedIn* relationships, we propose a method to estimate the frequency of hidden errors. This involves sampling these relationships and progressively increasing the sample size. The aim is to model the increase in hidden errors as the KG expands. Our approach includes integrating the model’s incorrect predictions into the sample, calculating the transitive closure, and then quantifying the inaccuracies within this closure. To illustrate the effect of increasing KG size on the prevalence of hidden errors, we conducted experiments using the incorrect predictions from three distinct models. The findings are detailed in Figure 1, demonstrating the relationship between KG size and hidden error frequency.

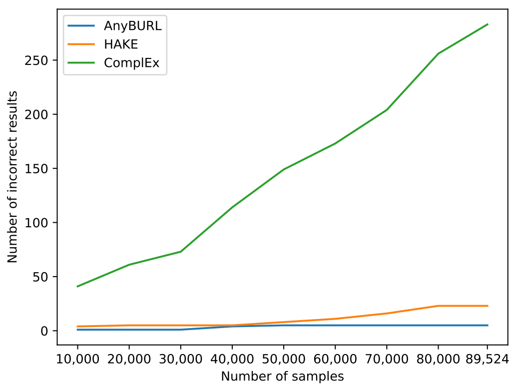


Figure 1: Number of incorrect results with different size of sample.

The data in the figure clearly indicates an increase in hidden errors as the volume of knowledge grows. Furthermore, note that this analysis is restricted to knowledge pertaining to the *isLocatedIn* relation. Broadening the scope to include other relations, especially given the comprehensive nature of TBox knowledge, would likely reveal a further increase in hidden errors. Additionally, the inferential capability of these models allows us to determine the correctness of predictions more effectively. For example, from the relationships *(A, isLocatedIn, B)* and *(B, isLocatedIn, C)*, we deduce *(A, isLocatedIn, C)*. This deduction process enables us to assess the correctness of the model’s predictions more rigorously. The results of this assessment are detailed in Table 2.

Model	#Correct Predictions
AnyBURL	272
ComplEx	116
HAKE	113

Table 2: Number of truly Correct Predictions by Different Models.

3.2 Validity of Non-Highest Prediction

In the preceding subsection, we examined the accuracy of the highest-probability, top-ranked predictions from various models. However, it is crucial to recognize that lower-probability predictions can also be correct.

To explore the accuracy among the top k predictions for different models, we conducted experiments with varying k values, focusing exclusively on knowledge associated with the *isLocatedIn* relation. We utilized the YAGO3-10 dataset, computing the transitive closure of its 89,524 *isLocatedIn*-related knowledge statements. This closure encompassed all relevant YAGO3-10 knowledge statements about *isLocatedIn*, based on existing information. For each model, we assessed whether the top k predictions fell within this transitive closure. Predictions included in the closure were deemed correct. Notably, this analysis excluded the highest probability prediction, already discussed previously.

Figure 2 illustrates the number of correct predictions for various models when k is set at 10 and 100. The data reveal a significant count of correct predictions among the top k results, excluding the highest probability prediction. Furthermore, increasing k from 10 to 100 markedly enhances the number of correct predictions.

Note that this analysis is limited to the *isLocatedIn* relation. Including additional knowledge would likely yield an even higher number of accurate predictions among the top results.

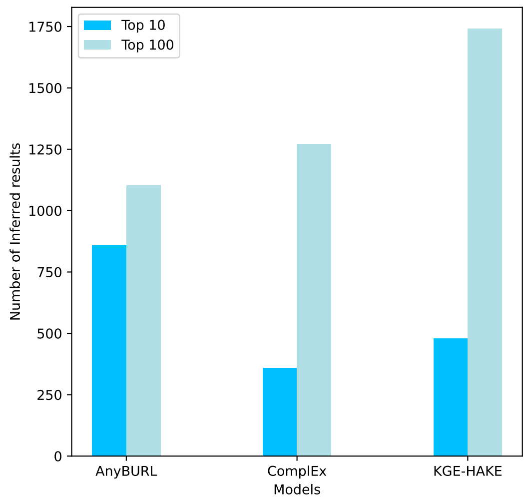


Figure 2: Number of truly correct top k results with different models

4 Conservative Completion with LP model

In this section, we elaborate on the application of LP model for conservative KGC.

Definition 1 (Conservative KG Completion)**.** *Consider a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where E represents the set of entities and R the set of relations in $\mathcal{K}$. For a predicted result (h, r, t), either in the form of $(h, r, ?)$ or $(?, r, t)$ with $h, t \in E$ and $r \in R$, if $\mathcal{K}$ entails (h, r, t) and this triple is not already present in $\mathcal{K}$, then the union of $\mathcal{K}$ and (h, r, t) is defined as a conservative completion of $\mathcal{K}$, and (h, r, t) is termed a complement.*

The essence of conservative completion is to augment the KG exclusively with predictions that are verifiably accurate within the existing framework of $\mathcal{K}$. In principle, for a missing fact $(h, r, ?)$, one could iterate over all potential results (h, r, t), verifying whether each (h, r, t) is entailed by $\mathcal{K}$ but not already in $\mathcal{K}$. However, the task of checking $\mathcal{K}$'s entailment for (h, r, t) is PSPACE-complete (Donini and Massacci, 2000). Given the potentially vast number of entities in $\mathcal{K}$, this iterative process is impractically time-consuming.

A more feasible approach involves initially using an LP model to generate a top-k candidate set of predicted results. Subsequently, these candidates undergo individual verification to accomplish conservative KGC. Specifically, for a missing fact $(h, r, ?)$, the LP model first predicts the top-k tail entities. For each candidate tail entity t_i, we check if (h, r, t_i) is both entailed by $\mathcal{K}$ and not present in $\mathcal{K}$. If these conditions are satisfied, we incorporate (h, r, t_i) into $\mathcal{K}$. The procedure for this approach is detailed in Algorithm 1.

Algorithm 1 Conservative Completion

Input: A KB $\mathcal{K}$; To predict missing facts Q; An LP model M
Parameter: k
Output: Conservative completion of $\mathcal{K}$

for missing fact $q \in Q$ **do**
  R be the top-k predictions of model M on q.
  for result $r \in R$ **do**
    if $r \notin \mathcal{K}$ and $\mathcal{K} \models r$ **then**
      $\mathcal{K} = \mathcal{K} \cup r$
    end if
  end for
end for
return $\mathcal{K}$

The algorithm illustrates that the number of complements generated is influenced by both the capabilities of the LP model and the content of the KB $\mathcal{K}$. We now demonstrate that an increase in the knowledge encompassed by $\mathcal{K}$ correlates with a rise in the number of complements.

Theorem 1. *Given two KBs, $\mathcal{K}_1$ and $\mathcal{K}_2$, with $\mathcal{K}_1 \subseteq \mathcal{K}_2$, any axiom α of the form $C \sqsubseteq D$, $r \sqsubseteq s$, $a : C$, or $(a, b) : r$ that is entailed by $\mathcal{K}_1$ is also entailed by $\mathcal{K}_2$.*

Corollary 1. *Enriching $\mathcal{K}$ with additional knowledge leads to an increase in the number of complements generated by LP models.*

5 Experiments and Analysis

5.1 Experimental Settings

5.1.1 Datasets

To effectively use LP models, we require KG datasets with extensive TBox information. Among commonly used KGs, YAGO3-10 fits this criterion. To enhance the diversity of our experimental data, we also incorporate the Radiology Lexicon (RadLex) dataset. YAGO3-10's knowledge aligns entirely with the RDFS framework, while RadLex contains elements that extend beyond RDFS. To illustrate how additional knowledge aids in the conservative completion of more complements, we processed RadLex to isolate knowledge compatible with RDFS (McBride, 2004), creating a subset termed RadLex_rdfs. The full RadLex dataset, including its OWL components (Bechhofer, 2018), is denoted as RadLex_owl.

For model training and evaluation, datasets must be divided into training, validation, and test sets. YAGO3-10, being a well-established dataset, is already pre-partitioned, eliminating the need for further division. For RadLex, we adhere to a rule ensuring that all entities and relations in the validation and test sets have prior appearances in the training set. This partitioning strategy ensures model robustness and reliability. The specifics of these datasets, including their partitioning, are detailed in Table 3.

5.1.2 LP Models

For generating candidate predictions, we evaluated five LP models: ComplEx, HAKE, MEIM, and ConEx, which are embedding-based models, and AnyBURL, a rule-based model. All models operate under the "Filtered" setting, which disregards existing valid triples in the prediction process.

5.1.3 Predict Sets

To assess LP models' performance, we created a prediction dataset consisting of incomplete triples $(h, r, ?)$ and $(?, r, t)$. For each $(h, r, ?)$, models predict k possible tail entities, and for $(?, r, t)$, they predict k possible head entities.

Given a KG with n entities and m relations, there are $n * m$ potential $(h, r, ?)$ and $(?, r, t)$ triples. However, not all combinations are meaningful. For example, a triple like *(Chatou, playsFor, ?)* in YAGO3-10 is illogical as *Chatou* is a city and cannot have a *playsFor* relationship. To ensure practical relevance in predictions, we utilized the test set to generate the prediction dataset. This involved masking the head and tail entities in each test data to create corresponding prediction data.

5.1.4 Model Pre-Warming

Before predictions, each LP model must be well-trained on the training set to ensure prediction accuracy. This training is followed by an evaluation on the test set, using metrics like Mean Reciprocal Rank (MRR) and Hits at K (H@K). The goal here is not to maximize these metrics, but to ensure they are reasonably high. Post-training, the evaluation metrics of various LP models on the test set are summarized in Table 4.

5.1.5 Reasoner

For entailment checking, a capable reasoner is essential to check whether $\mathcal{K} \models r$ is valid. HermiT (Glimm et al., 2014) fits this requirement. It stands out as the first publicly-available OWL reasoner that utilizes an efficient hypertableau calculus.

5.2 Results and Analysis

5.2.1 Conservative Completion on YAGO3-10

Using the pre-warmed model, we generated predictions for each entry in the predict sets, focusing on the top 10 and top 100 results. These predictions were subjected to entailment checks using HermiT applied to the corresponding ontology, to identify if they qualify as complements. The conservative completion results on the YAGO3-10 dataset are presented in Table 5. The table reveals that a substantial number of predictions from all models qualify as complements, with the count of complements significantly increasing as we expand our consideration from the top 10 to the top 100 results.

5.2.2 Correlation Experiment

It is important to note the lack of a strong correlation between the number of complements and standard evaluation metrics like MRR (Mean Reciprocal Rank) and Hits@K. Notably, the model with the highest Hits@10 score, MEIM, does not yield the most complements in tail prediction. Conversely, AnyBURL, which has the lowest Hits@10 score, produces the highest number of complements for tail prediction. To delve deeper into the relationship between the number of complements and the Hits@K metric, we propose modifying the learning conditions of the models. For AnyBURL, this involves adjusting its rule learning duration, and for ComplEx, altering the number of training epochs. We assess both head and tail prediction results under these revised learning conditions. The evaluation will track Hits@10 for the model's learning performance and utilize the top 10 predictions to evaluate the number of complements.

Figure 3 shows the results of AnyBURL and ComplEx. The graph shows that both AnyBURL and ComplEx exhibit an upward trend in the Hits@10 metric with increasing time or training epochs. However, this rise in Hits@10 does not consistently correspond to an increase in the number of complements. Specifically, while AnyBURL shows an increase in head prediction complements

dataset	#Entity	#Relation	#TBox	#TR	#VA	#TE
YAGO3-10	123,182	37	475,961	1,079,040	5,000	5,000
RadLex_rdfs	24,029	43	46,877	31,199	8,410	5,604
RadLex_owl	24,029	43	128,998	31,199	8,410	5,604

Table 3: Dataset Statistics. Here, #Entity denotes the total count of entities, and #Relation denotes the number of relations. #TBox denotes the number of axioms within the TBox. #TR, #VA, and #TE denote the sizes of the training, validation, and test sets, respectively.

	YAGO3-10				RadLex			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
AnyBURL	.544	.485	.583	.673	.582	.536	.615	.671
ComplEx	.576	.501	.621	.710	.571	.523	.602	.658
HAKE	.545	.461	.599	.694	.569	.497	.611	.700
MEIM	.584	.512	.626	.712	.566	.524	.587	.647
ConEx	.554	.477	.602	.692	.540	.487	.569	.640

Table 4: Evaluation metrics of various models after pre-warming.

	YAGO3-10			
	head prediction		tail prediction	
	top 10	top 100	top 10	top 100
AnyBURL	579	3386	1106	2278
ComplEx	890	4709	515	1583
HAKE	579	3114	594	1855
MEIM	749	4289	514	1657
ConEx	622	3099	381	1115

Table 5: Number of complements across different models on the YAGO3-10 dataset. Top 10 (Top 100) indicates the evaluation based on the 10 (100) highest-probability predictions.

with higher Hits@10, its tail prediction complements decrease. In contrast, ComplEx displays fluctuating trends in both head and tail prediction complements as Hits@10 increases, suggesting a lack of a direct linear correlation between the number of complements and the Hits@K metric.

5.2.3 Conservative Completion on RadLex

Experiments were also carried out for conservative completion on RadLex_rdfs and RadLex_owl. Table 6 details the number of complements obtained under various experimental settings for both RadLex_rdfs and RadLex_owl. Similar to the YAGO3-10 results, many predictions are identified as complements under different settings. Additionally, there is an observable increase in the number of complements as the focus shifts from the top 10 to the top 100 predictions. Furthermore, given the larger TBox in RadLex_owl compared to RadLex_rdfs, a comparison of the number of complements under identical experimental conditions is possible. This comparison, as shown in Table 6, indicates that under all settings, RadLex_owl consistently achieves equal or higher numbers of complements than RadLex_rdfs. This trend is evident in all cases except for the top 10 results in tail prediction, affirming the conclusion of Corollary 1.

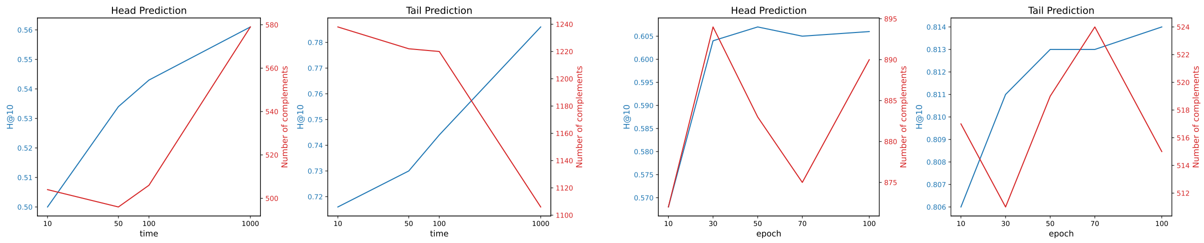
5.2.4 Conservative vs. Non-conservative

We define completions lacking entailment checks as *non-conservative completions*. Our analysis compares these with conservative completions from both KB and KG perspectives.

When a KB $\mathcal{K}$ and its conservative completion $\mathcal{K}_1$ are logically equivalent, this means that $\mathcal{K} \models \alpha$ if and only if $\mathcal{K}_1 \models \alpha$ for any axiom α. In contrast, $\mathcal{K}$ and its non-conservative completion $\mathcal{K}_2$ do not exhibit logical equivalence, and $\mathcal{K}_2$ may potentially become inconsistent, thereby compromising the reliability of any reasoning derived from it.

From a KG Perspective, where the focus is on relationships between entities, the distinction between conservative and non-conservative completions essentially boils down to a set difference. However, their impact can be compared by examining the variations in the learning behavior of LP models under both completion types.

We evaluate this by using ComplEx's completion results on YAGO3-10. Here, non-conservative com-



(a) AnyBURL

(b) ComplEx

Figure 3: Correlation between Hits@10 and the number of complements on AnyBURL and ComplEx. The two figures in (a) represent the results on AnyBURL, while the two figures in (b) represent the results on ComplEx. The left figures in both (a) and (b) represent the results of head prediction, while the right graphs represent the results of tail prediction. The blue line represents the variation of the Hits@10 metric with the model learning time, while the red line represents the change in the number of complements.

	RadLex_rdfs				RadLex_owl			
	head prediction		tail prediction		head prediction		tail prediction	
	top 10	top 100	top 10	top 100	top 10	top 100	top 10	top 100
AnyBURL	127	128	537	591	133	136	537	594
ComplEx	125	129	496	576	133	137	496	579
HAKE	130	135	539	593	137	146	539	598
MEIM	122	129	482	569	131	138	483	572
ConEx	119	130	519	565	124	139	519	572

Table 6: Number of complements with different models on the RadLex_rdfs and RadLex_owl dataset.

	head prediction				tail prediction			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Base	**.432**	.341	**.489**	**.604**	.719	**.664**	.753	**.816**
Non-conservative completion	.350	.231	.413	.586	.705	.636	.752	.814
Conservative completion	.430	**.342**	.475	.603	**.720**	**.664**	**.754**	.815

Table 7: Evaluation metrics on non-conservative completion and conservative completion.

pletion is derived from the top 1 tail prediction result, whereas conservative completion comes from the top 10 tail prediction results that pass entailment checks. Both completions are learned using ComplEx under identical parameters; their performance is evaluated using metrics on YAGO3-10's validation set.

Table 7 presents the learning performance of ComplEx with non-conservative and conservative completions. It indicates a decline in metrics for non-conservative completion compared to when no completion is used, whereas conservative completion shows partial improvement. This highlights the negative impact of incorporating unverified facts on the model's learning efficiency.

6 Conclusion and Future Work

This paper introduces a conservative completion approach for KGs, employing an LP model and leveraging rich KG information for correctness checks. Experimental results show many model predictions can be validated through these checks. As KG information is enhanced, the number of valid completions is expected to rise. The comparative analysis between conservative and non-conservative completions from KB and KG perspectives suggests that adding unverified completions to KGs detrimentally affects model learning.

The immediate next step for future research is to find methods to directly use KG's rich information for KGC, avoiding reliance on post-filtering for prediction validation.

References

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. In *Proc. NeurIPS'20*.

Naser Ahmadi, Viet-Phi Huynh, Venkata Vamsikrishna Meduri, Stefano Ortona, and Paolo Papotti. 2020. Mining expressive rules in knowledge graphs. *ACM J. Data Inf. Qual.*, 12(2):8:1–8:27.

Franz Baader, Ian Horrocks, and Ulrike Sattler. 2008. Description logics. *Foundations of Artificial Intelligence*, 3:135–179.

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proc. ACL'19*, pages 4387–4392. Association for Computational Linguistics.

Sean Bechhofer. 2018. OWL: web ontology language. In *Encyclopedia of Database Systems, Second Edition*. Springer.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. SIGMOD'08*, pages 1247–1250. ACM.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proc. NIPS'13*, pages 2787–2795.

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proc. WWW'19*, pages 151–161. ACM.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In *Proc. AAAI'21*, pages 6894–6902. AAAI Press.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022a. Geometry interaction knowledge graph embeddings. In *Proc. AAAI'22*, pages 5521–5529. AAAI Press.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, and Qingming Huang. 2022b. ER: equivariance regularizer for knowledge graph completion. In *Proc. AAAI'22*, pages 5512–5520. AAAI Press.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proc. EMNLP'14*, pages 1568–1579. ACL.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. Pairre: Knowledge graph embeddings via paired relation vectors. In *Proc. ACL/IJCNLP'21*, pages 4360–4369. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proc. AAAI'18*, pages 1811–1818. AAAI Press.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610.

Francesco M. Donini and Fabio Massacci. 2000. EXPTIME tableaux for ALC. *Artif. Intell.*, 124(1):87–138.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Proc. AAAI'18*, pages 1819–1826. AAAI Press.

Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved knowledge graph embedding using background taxonomic information. In *Proc. AAAI'19*, pages 3526–3533. AAAI Press.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting completeness in knowledge bases. In *Proc. WSDM'17*, pages 375–383. ACM.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.*, 24(6):707–730.

Zhiqiang Geng, Zhongkun Li, and Yongming Han. 2018. A novel asymmetric embedding model for knowledge graph completion. In *Proc. ICPR'18*, pages 290–295. IEEE Computer Society.

Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. 2014. Hermit: an owl 2 reasoner. *Journal of automated reasoning*, 53:245–269.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proc. ICML'19*, volume 97 of *Proceedings of Machine Learning Research*, pages 2505–2514. PMLR.

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proc. ACL'15*, pages 84–94. The Association for Computer Linguistics.

Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P. Xing. 2015. Entity hierarchy embedding. In *Proc. ACL'15*, pages 1292–1300. The Association for Computer Linguistics.

Jiacheng Huang, Yao Zhao, Wei Hu, Zhen Ning, Qijin Chen, Xiaoxia Qiu, Chengfu Huo, and Weijun Ren. 2022a. Trustworthy knowledge graph completion based on multi-sourced noisy data. In *Proc. WWW'22*, pages 956–965. ACM.

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022b. Multilingual knowledge graph completion with self-supervised adaptive graph alignment. In *Proc. ACL'22*, pages 474–485. Association for Computational Linguistics.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proc. AAAI'16*, pages 985–991. AAAI Press.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. Adaptive convolution for multi-relational learning. In *Proc. NAACL-HLT'19*, pages 978–987. Association for Computational Linguistics.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Proc. NeurIPS'18*, pages 4289–4300.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proc. ISWC'15*, pages 640–655. Springer.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.*, 81(1):53–67.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Qianyu Li, Xiaoli Tang, Tengyun Wang, Haizhi Yang, and Hengjie Song. 2019. Unifying task-oriented knowledge graph learning and recommendation. *IEEE Access*, 7:115816–115828.

David Liben-Nowell and Jon M. Kleinberg. 2007. The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.*, 58(7):1019–1031.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *Proc. ICML'17*, volume 70 of *Proceedings of Machine Learning Research*, pages 2168–2178. PMLR.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *Proceedings 7th biennial conference on innovative data systems research*. CIDR Conference.

Brian McBride. 2004. The resource description framework (RDF) and its vocabulary description language RDFS. In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 51–66. Springer.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proc. IJCAI'19*, pages 3137–3143. ijcai.org.

Johannes Messner, Ralph Abboud, and İsmail İlkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proc. AAAI'22*, pages 7779–7787. AAAI Press.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proc. HLT-NAACL'13*, pages 777–782. The Association for Computational Linguistics.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proc. NAACL-HLT'18*, pages 327–333. Association for Computational Linguistics.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proc. NAACL-HLT'19*, pages 2180–2189. Association for Computational Linguistics.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proc. NAACL-HLT'16*, pages 460–466. The Association for Computational Linguistics.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proc. AAAI'16*, pages 1955–1961. AAAI Press.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proc. ICML'11*, pages 809–816. Omnipress.

Guanglin Niu, Bo Li, Yongfei Zhang, and Shiliang Pu. 2022. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In *Proc. ACL'22*, pages 2867–2877. Association for Computational Linguistics.

Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *Proc. SIGIR'21*, pages 213–222. ACM.

Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM*, 62(8):36–43.

Simon Ott, Patrick Betz, Daria Stepanova, Mohamed H. Gad-Elrab, Christian Meilicke, and Heiner Stuckenschmidt. 2023. Rule-based knowledge graph completion with canonical models. In *Proc. CIKM'23*, pages 1971–1981. ACM.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artif. Intell. Rev.*, 56(11):13071–13102.

Pouya Pezeshkpour, Yifan Tian, and Sameer Singh. 2020. Revisiting evaluation of knowledge base completion models. *Automated Knowledge Base Construction (AKBC)*.

Maximilian Pflueger, David J. Tena Cucala, and Egor V. Kostylev. 2022. GNNQ: A neuro-symbolic approach to query answering over incomplete knowledge graphs. In *Proc. ISWC'22*, pages 481–497. Springer.

Giuseppe Pirrò. 2020. Relatedness and tbox-driven rule learning in large knowledge bases. In *Proc. AAAI'20*, pages 2975–2982. AAAI Press.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proc. ESWC'18*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Manfred Schmidt-Schauß and Gert Smolka. 1991a. Attributive concept descriptions with complements. *Journal of Artificial intelligence*, 48(1):1–26.

Manfred Schmidt-Schauß and Gert Smolka. 1991b. Attributive concept descriptions with complements. *Journal of Artificial intelligence*, 48(1):1–26.

Tengwei Song, Jie Luo, and Lei Huang. 2021. Rotpro: Modeling transitivity by projection in knowledge graph embedding. In *Proc. NeurIPS'21*, pages 24695–24706.

Fabian M. Suchanek, David Gross-Amblard, and Serge Abiteboul. 2011. Watermarking for ontologies. In *Proc. ISWC'11*, volume 7031, pages 697–713. Springer.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A large ontology from wikipedia and wordnet. *J. Web Semant.*, 6(3):203–217.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proc. ICLR'19*. OpenReview.net.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proc. ICML'16'*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Huijuan Wang, Siming Dai, Weiyue Su, Hui Zhong, Zeyang Fang, Zhengjie Huang, Shikun Feng, Zeyu Chen, Yu Sun, and Dianhai Yu. 2022. Simple and effective relation-based embedding propagation for knowledge representation learning. In *Proc. IJCAI'22*, pages 2755–2761. ijcai.org.

Hong Wu, Zhe Wang, Kewen Wang, and Yi-Dong Shen. 2022. Learning typed rules over knowledge graphs. In *Proc. KR'22*.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016a. Representation learning of knowledge graphs with entity descriptions. In *Proc. AAAI'16*, pages 2659–2665. AAAI Press.

Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied knowledge representation learning. In *Proc. IJCAI'17*, pages 3140–3146. ijcai.org.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016b. Representation learning of knowledge graphs with hierarchical types. In *Proc. IJCAI'16*, pages 2965–2971. IJCAI/AAAI Press.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. ICLR'15*.

Donghan Yu, Yu Gu, Chenyan Xiong, and Yiming Yang. 2023. Compleqa: Benchmarking the impacts of knowledge graph completion methods on question answering. In *Proc. EMNLP'23*, pages 12748–12755. Association for Computational Linguistics.

Qianjin Zhang, Ronggui Wang, Juan Yang, and Lixia Xue. 2022. Knowledge graph embedding by reflection transformation. *Knowl. Based Syst.*, 238:107861.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Proc. NeurIPS'19'*, pages 2731–2741.

Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019b. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proc. WSDM'19*, pages 96–104. ACM.

Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023. Variational reasoning over incomplete knowledge graphs for conversational recommendation. In *Proc. WSDM'23*, pages 231–239. ACM.

Zili Zhou, Shaowu Liu, Guandong Xu, and Wu Zhang. 2019. On completing sparse knowledge base with transitive relation embedding. In *Proc. AAAI'19*, pages 3125–3132. AAAI Press.

A Related Work

LP in the context of KGs has seen a variety of approaches over time. Traditional methods often rely on observable features, employing strategies like Rule Mining (Galárraga et al., 2015; Meilicke et al., 2019; Pirrò, 2020; Ahmadi et al., 2020; Wu et al., 2022; Ott et al., 2023) or Path Ranking Algorithm (Lao and Cohen, 2010) to uncover missing triples in the graph. Rule Mining, for instance, deduces patterns such as "If a person lives in a city, then their spouse likely resides in the same city", using these inferred rules to predict new information and identify gaps in the KB. However, these methods are limited to discovering facts between instances already present in the KB, failing to recognize entirely missing entities.

More recently, the advent of machine learning techniques has shifted focus to capturing latent features of graphs. This is achieved through vectorized representations, or embeddings, of graph elements. Embeddings, which are vectors of numerical values, can represent various elements depending on the domain. They are learned automatically based on the patterns of occurrence and interaction of these elements in real-world datasets. One prominent approach for KGC and LP is based on KG embedding models, where the idea is to learn embeddings for entities and relations through training over known facts, and subsequently use the learned embeddings to compute plausibility scores for all possible facts. Embedding-based LP models harness diverse approaches and architectures, each tailored to specific optimization challenges and techniques. Broadly, these models can be categorized into three primary families:

- Tensor Decomposition Models: These models leverage mathematical techniques of tensor factorization. Representative examples include RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), Analogy (Liu et al., 2017), SimplE (Kazemi and Poole, 2018), HolE (Nickel et al., 2016), A2N (Bansal et al., 2019), and EA (Cao et al., 2022b).

- Geometric Models: These models conceptualize entities and relations in a geometric space, often leveraging spatial relationships for prediction. Representative examples include TranE (Bordes et al., 2013), STarnsE (Nguyen et al., 2016), CrossE (Zhang et al., 2019b), TorusE (Ebisu and Ichise, 2018), RotatE (Sun et al., 2019), QuatE (Zhang et al., 2019a), PairRE (Chao et al., 2021), DualE (Cao et al., 2021), MTransH (Niu et al., 2021), RotPro (Song et al., 2021), BoxE (Abboud et al., 2020), BoxTE (Messner et al., 2022), ReflectE (Zhang et al., 2022), and GIE (Cao et al., 2022a).

- Deep Learning Models: These models apply deep learning techniques, particularly convolutional, recurrent, and graph neural networks, to extract complex patterns from graph data. Representative examples include ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018), ConvR (Jiang et al., 2019), CapsE (Nguyen et al., 2019), RSN (Guo et al., 2019), REP (Wang et al., 2022), and TKGC (Huang et al., 2022a).

The models discussed above rely entirely on factual data from KGs for learning. While this internal KG data is valuable, relying exclusively on it can introduce issues such as biases, overfitting, limited perspectives, and difficulties in capturing complex relationships. Embedding-based models using solely internal KG data risk inaccuracies in representing entity and relation vectors, potentially leading to erroneous predictions, as noted in (Niu et al., 2022). Combining internal data with external sources can alleviate these issues. Several models exemplify this approach. For instance, TRESCAL (Chang et al., 2014), TCRL (Krompaß et al., 2015), TKRL (Xie et al., 2016b), and CAKE (Niu et al., 2022) utilize entity type information. DKRL (Xie et al., 2016a) incorporates textual descriptions. EHE (Hu et al., 2015), SSE (Guo et al., 2015), and SimplE+(Fatemi et al., 2019) leverage entity hierarchical and taxonomic information. TranSparse(Ji et al., 2016), AEM (Geng et al., 2018), and TRE (Zhou et al., 2019) focus on relation-related information. IKRL (Xie et al., 2017) employs images of entities. While these models have demonstrated impressive performances across various datasets, the field continues to grapple with unresolved challenges.

B The Description Logic $\mathcal{ALC}$

KGs contain information about relationships between entities and can encompass more enriched information, such as entity types and specific properties of relationships. Description Logic (DL) (Baader et al., 2008) offers a unified framework for representing both the KG and its supplementary information. As a decidable fragment of first-order logic, DL presents favorable computational properties. For instance, the entirety of the YAGO3 dataset can be expressed using the $\mathcal{ALC}$ (Schmidt-Schauß and Smolka, 1991a) subset of DL. We now delve into the syntax and semantics of $\mathcal{ALC}$ and its application in representing KG information.

Consider N_I, N_C, and N_R as disjoint and countably infinite sets representing individuals, concept names, and role names, respectively. In this context, "individuals" correspond to entities in the KG, "concept names" correspond to entity types, and "role names" correspond to the various relationships. To represent more complex information, $\mathcal{ALC}$-concepts are built through inductive construction, adhering to the following syntactic rules:

$$C, D \longrightarrow \top \mid \bot \mid A \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall r.C \mid \exists r.C,$$

where $A \in N_C, r \in N_R$, and C and D range over concepts.

The semantics of $\mathcal{ALC}$ is defined in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is the domain of the interpretation (a non-empty set), and $\cdot^{\mathcal{I}}$ denotes the interpretation function, which satisfies:

- $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, for all $A \in N_C$
- $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, for all $r \in N_R$

The extension mapping is extended to complex ALC-concept as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{I} \backslash C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\forall r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y.(x,y) \in r^{\mathcal{I}} \to y \in C^{\mathcal{I}}\}$$
$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

For any concept C and D, the expression $C \sqsubseteq D$ is referred to as an $\mathcal{ALC}$ general concept inclusion (GCI). An interpretation $\mathcal{I}$ satisfies a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. GCIs can be used to represent relationships between entity types. For instance, the statement Teacher is Person can be represented as Teacher $\sqsubseteq$ Person, and the statement Courses and people are disjoint can be represented as Course $\sqcap$ Person $\sqsubseteq \bot$. Similarly, we can use $\mathcal{ALC}$ to represent information about relations. For example, $r \sqsubseteq s$ expresses that relation s includes relation r. If relation r is transitive, it can be represented as Trans(r). Dom(r) and Range(r) are used to express the domain concept of r and range concept of r. We define TBox as a collection of GCIs, $r \sqsubseteq s$, Trans(s), and other properties of relations.

In addition to the information about TBox, there are various other types of information, such as entity types and relationships between entities, that need to be represented. We use $a : C$ (concept assertion) to express the type to which an entity belongs and $(a, b) : r$ (role assertion) to express the relationship satisfied between two entities, where C is a concept, r is a role and a, b are individuals. For example, New York is a city can be expressed as New_York : city and New York is located in the United States can be expressed as $(\text{New\_York}, \text{United\_States})$: is_located_in. An interpretation satisfies a concept assertion if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and it satisfies a role assertion if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. The combination of concept assertion and role assertion is referred to as an ABox.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Knowledge Base (KB), where $\mathcal{T}$ represents the TBox and $\mathcal{A}$ represents the ABox. A model of $\mathcal{K}$ is an interpretation that satisfies every axiom in $\mathcal{T} \cup \mathcal{A}$. For an axiom α of the form $C \sqsubseteq D, a : C, (a, b) : r$, we say that $\mathcal{K}$ entails α if every model of $\mathcal{K}$ is also a model of α, denoted as $\mathcal{K} \models \alpha$. Entailment checking allows us to determine which information can be derived from the existing information. By using the tableau algorithm(Schmidt-Schauß and Smolka, 1991b), we can check whether an axiom can be entailed from $\mathcal{K}$ with a PSPACE-complete complexity.

C Missing Proofs

Theorem 1. *Given two KBs, $\mathcal{K}_1$ and $\mathcal{K}_2$, with $\mathcal{K}_1 \subseteq \mathcal{K}_2$, any axiom α of the form $C \sqsubseteq D$, $r \sqsubseteq s$, $a : C$, or $(a, b) : r$ that is entailed by $\mathcal{K}_1$ is also entailed by $\mathcal{K}_2$.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{K}_2$. By definition, $\mathcal{I}$ entails every axiom in $\mathcal{K}_2$. Given that $\mathcal{K}_1 \subseteq \mathcal{K}_2$, $\mathcal{I}$ must also entail every axiom in $\mathcal{K}_1$. Thus, $\mathcal{I}$ serves as a model for $\mathcal{K}_1$. Consequently, if $\mathcal{K}_1 \models \alpha$, indicating that all models of $\mathcal{K}_1$ entail α, it follows that all models of $\mathcal{K}_2$, which include models of $\mathcal{K}_1$, also entail α. Hence, $\mathcal{K}_2 \models \alpha$. $\square$