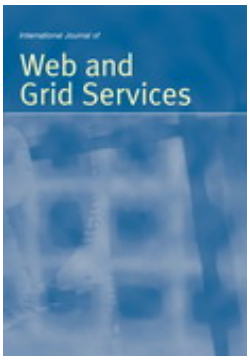


International Journal of Web and Grid Services

ISSN online: 1741-1114 - ISSN print: 1741-1106
https://www.inderscience.com/ijwgs

A survey on knowledge graph evolution: proliferation, dynamic embedding, and versioning

Xiongnan Jin, Zhilin Wang, Manni Duan, Yan Shao, Xingyun Hong, Yongheng Wang, Byungkook Oh

DOI: 10.1504/IJWGS.2025.10069835

Article History:

Received:	29 December 2023
Last revised:	06 January 2024
Accepted:	27 September 2024
Published online:	14 March 2025

A survey on knowledge graph evolution: proliferation, dynamic embedding, and versioning

Xiongnan Jin

National Engineering Laboratory for
Big Data System Computing Technology,
Shenzhen University,
Shenzhen, China
Email: xiongnanjin@szu.edu.cn

Zhilin Wang

Alibaba Group,
Hangzhou, China
Email: wzl446229@alibaba-inc.com

Manni Duan

Zhejiang Laboratory,
Hangzhou, China
Email: duanmanni@gmail.com

Yan Shao

China Mobile (Hangzhou) Information Technology Co., Ltd.,
Hangzhou, China
Email: shaoyan@cmhi.chinamobile.com

Xingyun Hong and Yongheng Wang*

Zhejiang Laboratory,
Hangzhou, China
Email: xyhong@zhejianglab.org
Email: wangyh@zhejianglab.org
*Corresponding author

Byungkook Oh

Computer Science and Engineering,
Konkuk University,
Seoul, South Korea
Email: bkoh@konkuk.ac.kr

Abstract: In the era of large language models (LLMs), knowledge graphs (KGs) can play a pivotal role in enhancing LLMs by providing a structured representation of knowledge, relationships, and entities. This knowledge is essential for LLMs to understand and interpret information in a coherent and contextually relevant manner. KGs must undergo continuous evolution with minimal human intervention to remain effective. Organisations often employ automated techniques, such as web scraping, natural language processing, and machine learning algorithms, to accomplish this continuous evolution. However, there is a lack of reviews covering recent advances in KG evolution. In this survey, we first give an overview and then describe the methods of KG evolution. Afterward, we review and analyse the evaluation metrics, datasets, and experimental performances. Finally, we provide findings and future directions from the investigation and conclude with a discussion.

Keywords: knowledge graph evolution; KG proliferation; fact validation; property error detection; PED; rule mining; KG dynamic embedding; KG versioning; large language model; LLM.

Reference to this paper should be made as follows: Jin, X., Wang, Z., Duan, M., Shao, Y., Hong, X., Wang, Y. and Oh, B. (2025) 'A survey on knowledge graph evolution: proliferation, dynamic embedding, and versioning', *Int. J. Web and Grid Services*, Vol. 21, No. 1, pp.88–111.

Biographical notes: Xiongnan Jin is an Assistant Professor at the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, China. Before that, he was a senior researcher at the Zhejiang Laboratory in China and a post-doctoral researcher at the National Institute of Standards and Technology in the United States. He received his Bachelor's in Software Engineering from Zhejiang University, China, and a PhD in Computer Science from Yonsei University, South Korea. His research interests include knowledge graphs, large language models, and multi-modal data processing.

Zhilin Wang is a Algorithm Engineer at Alibaba Group, China. Before that, he was an Algorithm Engineer with Zhejiang Laboratory in China. He received his MS in Data Science from George Washington University, DC, USA. His research interests include data mining, deep learning, natural language processing, and knowledge graph.

Manni Duan is a Senior Engineering Expert at Zhejiang Laboratory, China. Before that, she worked as a Senior Algorithm Expert (P9) at Alibaba Group in China. She was responsible for the group's first content-based image monitoring system. Besides, she has led multiple large-scale artificial intelligence engineering projects, including the original product image protection, the one-stop visual model production, and FashionAI innovation systems. She received her PhD in Signal Processing and Information Systems from the University of Science and Technology of China. Her main research areas are computer vision, multimodal knowledge computing, and application architecture.

Yan Shao is a Senior Engineer at China Mobile (Hangzhou) Information Technology Co., Ltd., China. Before that, he was an Engineering Expert at Zhejiang Laboratory and an Algorithm Expert at Alibaba Group in China.

He received his PhD from Uppsala University, Sweden. He has published 10+ research papers and holds five authorised patents. His research interests include natural language processing and large language models for science.

Xingyun Hong is an Algorithm Engineer at Zhejiang Laboratory, China. She received her MS in Control Science and Technology from Zhejiang University, China. Her research interests include machine learning, data mining, and deep learning.

Yongheng Wang is the Deputy Director of Research Center for Astronomical Computing, Zhejiang Laboratory, China. He received his PhD in Computer Science and Technology from the National University of Defense Technology, China. His research interests include big data analytics, machine learning, and intelligent decision-making.

Byungkook Oh is an Assistant Professor at Konkuk University, South Korea. Before that, he was a Research Scientist at Samsung Research and a post-doctoral researcher at Yonsei University in South Korea. He received his PhD in Computer Science from Yonsei University, South Korea. His research interests include knowledge representation and knowledge-based applications such as knowledge-enhanced NLP applications, information retrieval, and recommendation.

1 Introduction

Knowledge graphs (KGs) model human knowledge in a graph structure that enables machines to understand the semantics and logic. With the development of knowledge modelling and construction technologies, more and more KGs are built and made public on the web, such as Wikidata (https://www.wikidata.org/) and BaiduBaike (https://baike.baidu.com/), supporting various KG-based applications such as question answering, reasoning, and information retrieval (Ahn et al., 2023; Liu et al., 2022; Zillner and Winiwarter, 2005; Yang et al., 2006).

With the emergence of large language models (LLMs) such as GPT-4 (OpenAI, 2023), traditional information technologies are facing revolutionary change. For example, ChatGPT assists people in answering questions, creating travel plans in the way of conversation. However, problems still exist for LLMs, such as generating realistic incorrect answers [i.e., hallucination (Ji et al., 2023)], unreliable logical reasoning, and lacking vertical domain knowledge (Sun et al., 2023).

KGs have advantages over LLMs regarding trustworthiness, explainability, and extensibility (Yang et al., 2023). Therefore, KGs can evaluate the generating ability, guide the controllable constraint creation, and assist the domain adaption in improving LLMs' performance. However, KGs are always flawed since data and human knowledge are continuously generated and updated. Thus, KGs require frequent refinement to evolve up to date.

Various KG evolution methods have been proposed in the literature. *KG proliferation* methods verify new facts and properties by computing measurement scores to decide whether to integrate into an existing KG. *KG dynamic embedding* approaches are

proposed to avoid re-learning the full KG each time it is proliferated. *KG versioning* techniques are developed for comparing, tracking, and reverting evolutionary KGs.

There are several literature surveys on KG research (Ji et al., 2021; Zhang et al., 2021) covering the topics of KG embedding (mainly static), extraction, reasoning, and applications. Paulheim (2017) provides a survey for KG completion and error detection, including methods and evaluation details. However, it is now more than six years old, and there is still a lack of surveys covering recent advances in KG evolution.

To provide a systematic view of KG evolution, we look for papers published in the past three years of KG-associated major conferences via exploring related KG sessions. If there are no sessions, keyword searching is employed using 'evolve, update, trust, validate, reliable, dynamic, knowledge, etc.' In this way, 18 seed papers are collected. Finally, references and citations of seed papers are investigated. The main contributions of this survey are summarised as follows:

- A temporal view of the KG evolution procedure is given. KG evolution methods are divided into proliferation, dynamic embedding, and versioning. KG proliferation is further categorised into fact validation (FV), property error detection (PED), and rule mining.
- Evaluation metrics, datasets, and performance of KG evolution methods are summarised and analysed. Besides, datasets are collected and made accessible online (https://pan.baidu.com/s/1O76I7LLwzyyEX4dIVwEvhQ?pwd=j1wa).
- Findings and potential future directions are described.

The remainder of this survey is organised as follows. Section 2 gives an overview. Section 3 introduces the KG evolution approaches in the literature. Section 4 describes the evaluation metrics, datasets, and performance of current KG evolution methods. Section 5 gives the future directions, and Section 6 concludes the article.

2 Overview

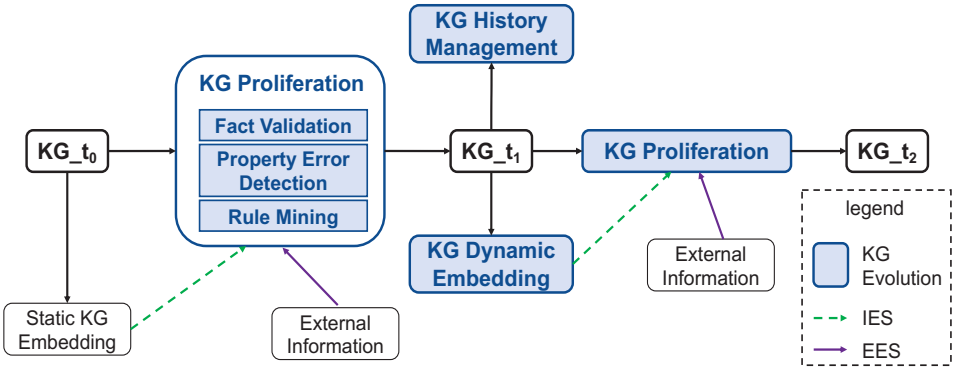
Ontologies are the backbone of KGs. An ontology is a formal representation of knowledge about a particular domain. It defines the concepts, relationships, and properties within that domain and provides a structured framework for organising and understanding that knowledge.

KGs are instantiations of one or multiple ontologies. A KG represents human knowledge in the way of a heterogeneous graph structure, which consists of a set of facts (also called triples) $\mathcal{KG} = (h, r, t)$. h and t indicate head and tail entities or concepts, and r denotes a relation between two entities or an entity and a concept. For example, (`ChatGPT`, `productOf`, `OpenAI`) represents a fact that the ChatGPT is a product of OpenAI, and (`OpenAI`, `hasType`, `company`) means OpenAI belongs to a concept of the company.

KGs need evolution to fulfill the requirements of KG-based applications. Concept-level changes are rare after initial construction compared with entity-level changes. Moreover, the concept-level fact scale is much smaller than entity-level facts. Thus, this survey focuses on the challenging problem of entity-level KG evolution, including KG proliferation, dynamic embedding, and versioning.

Figure 1 shows a temporal view of KG evolution. KG_t_0 indicates the initially constructed KG. There are two kinds of information sources for KG evolution: internal evolution sources (IESs) and external evolution sources (EESs). IESs are typically provided by KG embedding, also called knowledge representation learning. Static KG embedding learns the full KG_t_0 and maps the entities and relations into low-dimensional vector spaces. A major downstream task of KG embedding is link prediction, which generates missing facts. EESs can be produced by knowledge extraction techniques such as semantic table interpretation (Suhara et al., 2022) from the Web. Knowledge extraction is typically time-consuming and requires knowledge modelling, entity recognition, relation extraction, and entity linking. The emergence of LLMs is accelerating the procedure.

Figure 1 A temporal view of KG evolution (see online version for colours)



Notes: KG_t_n indicates a KG at a certain timestamp t_n. IES and EES denote the internal and external evolution sources.

KG proliferation methods are employed to verify the knowledge from IESs and EESs to be aggregated into KG_t_0 and then evolve it as KG_t_1. We categorise the research for KG proliferation into FV, PED, and rule mining. FV or fact-checking discriminates the knowledge triples by designing measurements and computing corresponding scores. PED aims to find out the literal or numerical errors contained in properties of entities and relations. Rule mining discovers logical rules from a KG, and the rules assist KG proliferation in turn.

Then, KG dynamic embedding and versioning approaches are performed on KG_t_1 for handling changes. Unlike static embedding, dynamic embedding learns the representation incrementally or adaptively. KG versioning aims to discover and visualise the changes and evolution and enable rollback to KG_t_0 if needed.

3 Methods for KG evolution

3.1 KG proliferation

For *KG proliferation*, *FV* is a crucial aspect of KG update, focusing on measuring the veracity and trustworthiness of relational facts. The procedures of truth inference

and source credibility estimation are viewed as closely linked. Therefore, specific FV strategies use an iterative mechanism (Yin et al., 2007; Galland et al., 2010; Zhang et al., 2019a; Yan et al., 2021). In this process, the steps of inferring truthfulness and assessing the weight of sources are repeatedly performed in a cycle until a point of agreement or stability is reached.

Optimisation-based FV approaches usually construct a function to evaluate the disparity between the given information and the ascertained truths. The goal during training is to reduce this function's value, thereby aligning the collated truths more closely with sources of significant credibility. ETA^2 (Zhang et al., 2019b) utilises semantic analysis to infer user expertise for optimising task assignment and cost reduction in truth estimation. CTD (Ye et al., 2021) applies denial constraints in truth discovery, treating it as an optimisation problem by modelling relationships between different entity attributes. Xiao and Wang (2022) redefine truth discovery as a joint maximum likelihood estimation problem, using profile likelihood techniques for determining truth in claims and source reliability. Zheng et al. (2022) present an optimised network flow approach for crowd workers in knowledge base validation, focusing on expertise, fact utility, and evolving knowledge bases. Li et al. (2023) propose a globally optimal, uncertainty-aware truth discovery model for mineral prospectivity mapping to enhance decision-making through quantitative assessment and uncertainty visualisation.

Some FV approaches utilise probabilistic graphical models (PGMs), which are graphical depictions of probability distributions, capitalising on conditional independencies to encode complex distributions efficiently. The primary forms of PGMs include Bayesian networks and Markov networks, also known as Markov random fields. MBM (Wang et al., 2015) is a Bayesian-based model proposed to address the multi-truth-finding problem, incorporating unique adaptations for source/value grouping, source dependency, and inter-value mutual exclusion. BWA (Li et al., 2019) is a generative Bayesian model that transforms a binary discrete problem into a continuous regression task by utilising a normal likelihood function and a conjugate inverse-Gamma prior. Wang et al. (2021) introduce an online location-aware crowdsensing system that adeptly handles both numerical and categorical tasks, using a PGM to infer truth and assess worker expertise dynamically.

PED identifies and rectifies property fact (i.e., literal fact) errors or inconsistencies in the graph. By detecting conflicting or missing types or values, PED helps maintain the integrity and quality of the KGs during the updating process. Yao and Barbosa (2021) put forward an active algorithm that maximises the use of both accurate and noisy labels. Their method leverages diverse information from multiple sources and is tailored for detecting typing errors. Beyond a specific type property, TKGC (Huang et al., 2022) is proposed to handle the problems of PED and FV simultaneously by leveraging multi-sourced noisy data and existing KG facts. It introduces a graph neural network with a holistic scoring function to evaluate the plausibility of facts with different value types. PTrustE (Ma et al., 2022) further abstracts the relational/property facts into paths to detect and represent noisy information in high-level, complex path facts. PTrustE learns the quality of each path in KG and uses this information to determine if a triple is accurate or contains noise.

However, these approaches cannot discover simple, straightforward errors that violate common sense. For example, given two facts (`Afghanistan`, `hasCapital`, `Afghanistan`) and (`Mike`, `hasAge`, `-1`), a human annotator could quickly tell these

facts are false. The first fact violates the domain type restriction, i.e., the capital of a country should be a city. The second fact does not satisfy the range limit that the age of a human or animal cannot be a negative number.

Rule mining techniques are used in KG proliferation to uncover meaningful patterns and relationships. These methods extract logical rules from KGs to fill gaps and discover hidden knowledge. Common sense can be injected into models by converting mined rules into structural languages such as SWRL (Horrocks et al., 2004). Chen et al. (2022) introduce a method in which a reinforcement learning agent is trained to generate rules and guide the rule generation process using its value function. The reinforcement learning-based approach discovers more rules than traditional deep learning methods but may need more time and iterations to converge. TyRuLe (Wu et al., 2022) is a KG rule learning method for typed rules built upon path- and embedding-based strategies. Experimental results show that integrating typed rules improves the performance of rule mining in downstream tasks, e.g., link prediction. Colt (Loster et al., 2021) is a framework for few-shot rule-based knowledge validation, which allows users to validate a few facts entailed by a rule to estimate rule quality and validate facts accurately. ChatRule (Luo et al., 2023) mines logical rules over KGs by leveraging the power of LLMs. ChatRule comprises an LLM-based rule generator, rule ranking, and KG reasoning modules.

3.2 KG dynamic embedding

KG dynamic embedding aims to dynamically map high-dimensional evolutionary KGs into low-dimensional vectors to speed up the computation while preserving the structural information. KG incremental embedding and temporal knowledge graph (TKG) embedding are two typical methods of KG dynamic embedding.

KG incremental embedding aims to update the KG vector representations without investigating the entire KG structure. ABIE (Dong et al., 2021) is an anchors-based incremental model for dynamic KG embedding. The underlying hypothesis is that some key entities in a KG, called anchors, can locate the primary embedding space. RotatH (Wei et al., 2021) is an approach for incremental KG embedding, which leverages relation-specific hyperplanes and rotation to update embeddings efficiently while ensuring freshness and accuracy. Recently, Wei et al. (2021) extended their approach to embed multi-modal KGs (MMKGs) by adopting a gated multi-modal encoder and decoding using RotatH. Then, the approach fuses unseen modal information of entities and incrementally embeds the new entities. MMKG is a specialised KG, which part of its knowledge has data items in modalities other than text, such as image, sound, and video (Zhu et al., 2022). However, these methods overlook the temporal information, which widely exists in KGs and may lead to imprecise KG representations.

TKG embedding incorporates time and temporal order information in the representation learning process to address the temporal conflicts or predict future facts. TDG2E (Tang et al., 2020) incorporates temporal evolving processes and preserves structural information. It utilises a GRU-based model to handle the dependency among sub-KGs and introduces an auxiliary loss to supervise learning using structural information. RE-GCN (Li et al., 2021) tackles the challenge of reasoning over TKGs and predicting future facts. It effectively captures the structural dependencies and sequential patterns in TKGs by learning evolutional representations of entities and relations through recurrently modelling the KG sequence. CENET (Xu et al., 2023) is an event forecasting

model based on historical contrastive learning on temporal KGs. CENET can predict repetitive, periodic, and brand-new events by considering historical and non-historical dependencies. However, TKG embedding must improve its scalability since it commonly requires re-learning the full KG when new facts come. A straightforward solution is to learn only a particular recent period of the KG. Nevertheless, the appropriate period is hard to decide since it depends on specific KG and applications.

Some works try to *combine the TKG embedding and incremental embedding* to benefit from both advantages (Wu et al., 2021; Jia et al., 2023). TIE (Wu et al., 2021) integrates TKG representation learning, experience replay, and temporal regularisation to address time-aware incremental embedding. AIR (Jia et al., 2023) is another framework for the adaptive update of dynamic KG embeddings that contain timestamps. These approaches demonstrate efficiency compared to full learning methods but still have a significant gap in accuracy on specific datasets. IncDE (Liu et al., 2024) is a continual KG embedding framework that learns and preserves knowledge with explicit graph structure. The proposed hierarchical ordering computes the adequate learning order, and incremental distillation and two-stage training preserve decent old knowledge.

3.3 KG history management

KG history management and versioning handles the KG changes to ensure consistency between different versions and track the evolution over time (Hartung et al., 2013; Cardoso et al., 2020; Diaz Benavides et al., 2022). COnto-Diff (Hartung et al., 2013) manages the evolution of life science ontologies. It determines the difference between ontology versions by matching and transforming basic change operations into more complex ones. HKG (Cardoso et al., 2020) addresses the problem of maintaining annotation quality for evolving domain knowledge by capturing the evolutionary aspects of knowledge in a structured manner. DynDiff (Diaz Benavides et al., 2022) detects and classifies alterations between different versions of ontologies. Besides, an ontology of changes is proposed to provide a context for improving comprehensiveness towards machines and humans. However, these works are mainly designed for ontology-level changes, which are relatively small and rare compared to entity-level changes. It remains a challenge to compare, manage versions, and visualise the evolution paths of large-scale entity-level KGs.

4 Metric, dataset and performance

4.1 Evaluation metrics

In this section, we review the metrics used in the literature for evaluating KG evolution approaches. Table 1 lists the related metrics, value preferences, and corresponding tasks. In the following, descriptions of the metrics will be given.

For *FV*, typically modelled as a classification problem, rank-based metrics are widely used for probabilistic outputs. New facts are ranked in ascending order of scores, computed by the KG proliferation models. A score indicates the possibility of a fact being authentic. Thus, the top-ranked facts are more likely to be wrong. Suppose that an evaluation dataset $\mathcal{D}$ consists of a positive fact set $\mathcal{D}^+$ and a negative fact set $\mathcal{D}^-$, and $rank_i$ denotes the rank of i^{th} fact. $\mathcal{D}^-$ should be ranked higher so that the average

ranking of $\mathcal{D}^-$ could tell the performance of a KG proliferation model. The metric *mean rank* is defined as:

$$\frac{1}{|\mathcal{D}^-|}\sum_{i=1}^{|\mathcal{D}^-|} rank_i \tag{1}$$

Table 1 Analysis of major metrics for KG evolution

Metric	*Mean rank* ↓	*Precision@k* ↑	*Precision* ↑	*Recall* ↑
Task	FV, DE	FV	FV, PED, KGV	FV, PED, KGV
Metric	F1-score ↑	RMSE ↓	MAE ↓	MAP ↑
Task	FV, PED	PED	PED	PED
Metric	Hits@k ↑	MRR ↑	#(high quality) rule ↑	Execution time ↓
Task	Rule, DE	Rule, DE	Rule	FV, rule, DE, KGV

Notes: Methods of FV, DE, PED, and KGV are short for FV, dynamic embedding, PED, and KG versioning, respectively. Pre denotes the value preference, ↑ means a higher value is preferred, and ↓ vice versa.

Mean filtered rank is a variant of Mean Rank that removes the correct predictions with higher rankings, formally:

$$\frac{1}{|\mathcal{D}^-|}\sum_{i=1}^{|\mathcal{D}^-|} rank_i - i \tag{2}$$

Another rank-based metric *Precision@k* measures the percentage of incorrect facts among top-k ranks, formally:

$$\frac{1}{k}\sum_{i=1}^{k} prec(i) \tag{3}$$

where $prec(i)$ equals to 1 if i^{th} ranked fact belongs to $\mathcal{D}^-$, otherwise 0.

When evaluating *non-probabilistic* FV results, traditional accuracy metrics such as *precision*, *recall*, and *F1-score* are adopted to test the fact classification ability of KG proliferation models. Positive (negative) means a KG proliferation model predicts a fact should (should not) be integrated into an existing KG. True (false) denotes a verified result of the model prediction. Besides, there are also specific metrics that are tailored for particular methods. *Utility* metric (Zheng et al., 2022) is proposed to evaluate the overall utility of a worker's correctly validated facts in expectation.

For evaluating the generated explanations for FV, *entity overlap* (Vedula and Parthasarathy, 2021) is used to measure the quality of explanations generated by a KG proliferation model, which compares the overlap of entity phrases between the generated and ground truth explanations. Conventional metrics for machine translation and document understanding, such as *BLUE* (Papineni et al., 2002), *METEOR* (Banerjee and Lavie, 2005), and *ROUGE* (Lin and Hovy, 2002), are also adopted to measure the quality of explanations generated by FV models. To comprehensively evaluate the generated explanations, more dedicated metrics designed for explainable AI systems, e.g., explanation goodness and satisfaction (Hoffman et al., 2018), are suggested to be considered.

For *PED*, *root mean square error (RMSE)* and *mean absolute error (MAE)* are commonly used to test the model performance on property-level verification. RMSE and MAE are defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{4}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y_i})| \tag{5}$$

where y_i and $\hat{y_i}$ indicate the ground truth and the predicted value. RMSE and MAE measure the performance of the prediction by computing the error amount corresponding to the ground truth. The difference between the two metrics is that RMSE penalises the bigger mistakes more strictly by employing the quadratic operation.

Each entity can have multiple properties, and each KG proliferation source contains a set of entities. Thus, PED is also modelled as a multi-classification problem, similar to the object detection task in computer vision, and uses *mean average precision (MAP)* to measure the performance of a set of classifiers.

For *rule mining*, two phases are rule learning and rule utilisation. *Execution time*, *number of rules found*, and *number of high-quality rules* are adopted to evaluate the efficiency, quantity, and quality factors of rule learning. High-quality rules are commonly defined as rules with pre-defined confidence, such as ≥ 0.7 (Omran et al., 2018; Pirrò, 2020). More rigorous metrics are needed to measure the quality of mined rules. In the phase of rule utilisation, rules are used for the link prediction task. Therefore, link prediction metrics such as *Hits@k* and *mean reciprocal rank (MRR)* are adopted, which are defined as:

$$Hits@k = \frac{1}{k}\sum_{i=1}^{k} hits(i) \tag{6}$$

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank(i)} \tag{7}$$

$hits(i)$ equals 1 if the golden answer appears in top-k model predictions; otherwise, 0. $rank(i)$ is the rank of the golden answer among the predictions ranked in descending orders with regard to the model confidence.

For *KG dynamic embedding*, the downstream task is also link prediction. The main focus is to update the embeddings using incremental or adaptive learning while ensuring accuracy. The performance of dynamic embedding is commonly compared with that of static embedding conducted on full KG re-learning. Hits@k and MRR are widely used metrics for accuracy. Another essential metric is execution time since the dynamic embedding should support continuous changes.

For *KG versioning*, execution time is adopted to test the efficiency. For effectiveness, *recall* is the primary metric to measure how many interested entities are missing. Note that the precision is 100% as claimed in Cardoso et al. (2020). The metric *%changeTools* (Diaz Benavides et al., 2022) is proposed to handle the cases without ground truth.

%changeTools measure the difference ratio between the outputs of the proposed method and COnto-Diff (Hartung et al., 2013), whose results are regarded as the ground truth. Besides, recall and precision are also used to test the ability of information retrieval and concept prediction in case versioning methods edit the original KGs. Furthermore, *storage* is employed to measure the space usage of KGs before and after versioning since versioning methods may edit KGs using evolution relations.

4.2 Datasets

We analysed the datasets used in the experiments for KG evolution methods. Table 2 shows the statistical counts, data factors such as fact label and temporal information, accessibility, and target tasks of the datasets. We collected and made the available datasets accessible online to facilitate the research on KG evolution. Please follow the original licenses while handling the datasets. The descriptions of typical datasets are given as the following.

- *YAGO-Loster* (Loster et al., 2021) YAGO2 contains 1,517 rules, of which 928 produced more than 5,500 triples. Loster et al. (2021) carefully selected 26 rules and corresponding 23,324 facts, then manually labelled the facts.
- *NELL-314* (Wang et al., 2020) is a subset of NELL constructed in the following way. Firstly, entities that appear at least five facts are chosen. Then, facts containing the chosen entities are collected. Finally, facts with reverse or 'generalisations' relations are removed to reduce the size and ambiguity.
- *OKELE* (Cao et al., 2020) includes ten popular classes of *Freebase*, each with 1,200 entities. OKELE contains 191,759 facts, and Cao et al. (2020) manually labelled the fact authenticity.
- *DBpedia-C* (Yao and Barbosa, 2021) is a subset of DBpedia with a balanced type distribution and a coarse typing granularity. DBpedia-C is for evaluating typing error detection. Entities of DBpedia-C have 17 distinct types, and 3,847 entities are assigned with gold labels.
- *DBpedia-F* (Yao and Barbosa, 2021) is a fine-grained dataset for typing error detection, which consists of 83 fine-grained types from DBpedia, 5,889 positive and 3,395 negative samples.
- *WN18RR* (Dettmers et al., 2018) is a subset of WordNet enhanced against inverse relation leakage. The dataset requires modelling of the whole KG to execute KG completion.
- *FB15-237* (Toutanova and Chen, 2015). FB15K (Bordes et al., 2013) is a subset of Freebase, which contains about 15K entities and 1,345 relations. FB15-237 is constructed by reducing the kinds of relations to 237 frequently used ones.
- *YAGO3-10* (Galárraga et al., 2015) is another subset of YAGO that only contains entities that appear in more than ten facts.
- *YAGO11K* (Tang et al., 2020) is extracted from YAGO facts assigned with time annotations.

- *Wikidata12k* (Tang et al., 2020) is extracted from Wikidata facts assigned with time annotations.
- *YAGO-increment* (Wei et al., 2021) includes three subsets of YAGO snapshots at different times: YAGO-1SP, 2SP, 3SP. The number of relation types is fixed at 32, and the number of entities and facts increases with time. YAGO-1SP, 2SP, 3SP contains 26,897, 27,031, 27,166 entities and 129,974, 131,364, and 132,754 facts, respectively.
- *DBLP-increment* (Wei et al., 2021) includes three subsets of DBLP snapshots at different times: DBLP-1SP, 2SP, 3SP. The number of relation types is fixed at 8, and entities and facts increase with time. DBLP-1SP, 2SP, 3SP contains 113,848, 114,413, 114,866 entities and 171,654, 172,666, and 173,690 facts, respectively.

Table 2 Analysis of datasets for KG evolution

Dataset	*#Fact*	*#Ent.*	*#R*	*FL*	*Acc.*	*T*	*E*	*Task*
YAGO-Loster (Loster et al., 2021)	23,324			◯	◯	X	P	FV, rule
NELL-314 (Wang et al., 2020)	148,001	13,965	314	P			X	FV
OKELE (Cao et al., 2020)	191,759	1,200		◯	◯	X	X	FV, PED
DBpedia-C (Yao and Barbosa, 2021)		3,847		X	◯	X	-	PED
DBpedia-F (Yao and Barbosa, 2021)		9,284		X	◯	X	-	PED
WN18RR (Dettmers et al., 2018)	93,003	40,493	11	X	◯	X	-	Rule
FB15-237 (Toutanova and Chen, 2015)	310,116	14,541	237	X	◯	X	-	Rule, DE
YAGO3-10 (Galárraga et al., 2015)	1,079,040	123,143	37				-	Rule, DE
YAGO11K (Tang et al., 2020)	20,400	10,623	10			◯	-	DE
Wikidata12k (Tang et al., 2020)	40,500	12,554	24	X	◯	◯	-	DE
YAGO-increment (Wei et al., 2021)	132,754	27,166	32	X	◯	X	-	DE
DBLP-increment (Wei et al., 2021)	173,690	114,866	8	X	◯	X	-	DE

Notes: #Fact, #Ent., and #R denote a dataset's number of facts, unique entities, and unique relations. FL, Acc., T, and E. indicate the fact label, dataset accessibility, temporal factor, and explain, where ◯, X, P, and blank means available, not available, partially available, and not sure, respectively. Dataset accessibility is verified using the links provided in the corresponding papers. Tasks of DE, FV, and PED are short for dynamic embedding, FV, PED, and dynamic query processing.

To summarise, the datasets are mainly derived from large-scale KGs, such as YAGO (https://yago-knowledge.org/), DBpedia (https://www.dbpedia.org/), Freebase (Bollacker et al., 2008), and NELL (Mitchell et al., 2018), with strategies focusing on specific KG evolution tasks. The datasets containing more than 1 million facts are regarded as large-scale datasets, and super-large ones with more than 10 million facts typically need distributed parallel computing. Super-large datasets in our review merely describe the generation procedures and do not provide the data directly. YAGO-Loster and OKELE are publicly available FV datasets with labels. Nevertheless, the facts are in different formats. The facts in YAGO-Loster are presented in the way (`subject`, `predicate`, `object`) and those of OKELE are expressed in natural language. According to our investigation, quite a few mistakes (17 out of 42 samples assigned with wrong labels) exist in YAGO-Loster's FV labels, maybe due to its focus on rules.

4.3 Performance

In this section, we present and analyse the performance of KG evolution approaches regarding tasks such as dynamic embedding, FV, PED, rule mining, and versioning. The scores are mainly taken and merged from recent KG evolution works (Huang et al., 2022; Wang et al., 2020; Yao and Barbosa, 2021; Wei et al., 2021). The blank cells indicate no score is recorded, and the bold score and method denote the best performance w.r.t. a metric on a dataset. Up to six methods and corresponding scores are presented for each metric-dataset pair. The representative methods are selected according to the category and scores. For example, if methods are divided into two categories, then three top-scored methods of each category are chosen.

Table 3 Performance of dynamic embedding methods on DBLP-increment (Wei et al., 2021) and YAGO-increment (Wei et al., 2021) datasets

	Dynamic embedding		
	Method	*Hits@10* ↑	*ET (min)* ↓
DBLP-1SP	RotatE (Sun et al., 2018)	0.693	*48*
	HAKE (Zhang et al., 2020)	0.711	52
	RotatH (Wei et al., 2021)	*0.716*	69
	Method	*Hits@10* ↑	*ET (min)* ↓
DBLP-2SP	RotatE (Sun et al., 2018)	0.699	47
	HAKE (Zhang et al., 2020)	0.718	52
	RotatH (Wei et al., 2021)	*0.721*	*8*
	Method	*Hits@10* ↑	*ET (min)* ↓
DBLP-3SP	RotatE (Sun et al., 2018)	0.701	49
	HAKE (Zhang et al., 2020)	0.701	49
	RotatH (Wei et al., 2021)	*0.718*	*8*
	Method	*MRR* ↑	*ET (min)* ↓
YAGO-1SP	HAKE (Zhang et al., 2020)	0.483	*45*
	RotatE (Sun et al., 2018)	0.494	53
	RotatH (Wei et al., 2021)	*0.500*	90
	Method	*MRR* ↑	*ET (min)* ↓
YAGO-2SP	HAKE (Zhang et al., 2020)	0.498	46
	RotatE (Sun et al., 2018)	0.498	58
	RotatH (Wei et al., 2021)	*0.504*	*10*
	Method	*MRR* ↑	*ET (min)* ↓
YAGO-3SP	HAKE (Zhang et al., 2020)	0.509	46
	RotatE (Sun et al., 2018)	0.515	59
	RotatH (Wei et al., 2021)	*0.518*	*15*

Note: ET indicates the execution time.

Table 4 Performance of FV and PED models on OKELE (Cao et al., 2020), NELL-314 (Wang et al., 2020), DBpedia-C (Yao and Barbosa, 2021), and DBpedia-F (Yao and Barbosa, 2021) datasets

	Fact validation						*Property error detection*					
	OKELE (Cao et al., 2020) (FV, PED)		*NELL-314 (Wang et al., 2020) (FV)*		*DBpedia-C (Yao and Barbosa, 2021)(PED)*		*DBpedia-F (Yao and Barbosa, 2021) (PED)*	
	Method	*Score*	*Score*	*Method*	*Method*	*Score*	*Method*	*Score*
Precision ↑ (FV, PED)	Majority voting (Zheng et al., 2017)	0.321			SSNM (ERR) (Yao and Barbosa, 2021)	0.72	Random	0.386
	PooledInvestment (Pasternack and Roth, 2010)	0.397			*SSNM (US) (Yao and Barbosa, 2021)*	*0.74*	RDF2Vec (Ristoski and Paulheim, 2016) + IF (Liu et al., 2008)	0.470
	CATD (Li et al., 2014)	0.432					Wikipedia2Vec (Yamada et al., 2020) + IF (Liu et al., 2008)	0.593
	BWA (Li et al., 2019)	0.414					*ReprLearning (Yao and Barbosa, 2021) + IF (Liu et al., 2008)*	*0.697*
	TKGC (Huang et al., 2022)	*0.524*						
Precision @500 ↑ (FV)			PTransE (Zhu et al., 2017)	0.580				
			CKRL (Xie et al., 2018)	0.586				
			Analogy (Liu et al., 2017)	0.600				
			DistMult (Yang et al., 2015)	0.604				
			CrossVal (Wang et al., 2020)	*0.704*				
Recall ↑ (FV, PED)	Majority voting (Zheng et al., 2017)	0.419	TransE (Bordes et al., 2013)	0.548	SSNM (ERR) (Yao and Barbosa, 2021)	0.46	Wikipedia2Vec (Yamada et al., 2020) + IF (Liu et al., 2008)	0.170
	PooledInvestment (Pasternack and Roth, 2010)	0.380	Analogy (Liu et al., 2017)	0.587	*SSNM (US) (Yao and Barbosa, 2021)*	*0.55*	RDF2Vec (Ristoski and Paulheim, 2016) + IF (Liu et al., 2008)	0.259
	CATD (Li et al., 2014)	0.423	SimplE (Kazemi and Poole, 2018)	0.591			ReprLearning (Yao and Barbosa, 2021) + IF (Liu et al., 2008)	0.288
	TKGC (Huang et al., 2022)	0.491	PTransE (Zhu et al., 2017)	0.609			*Random*	*0.513*
	MBM (Wang et al., 2015)	*0.539*	*CrossVal (Wang et al., 2020)*	*0.645*				
F1-score ↑ (FV, PED)	Majority voting (Zheng et al., 2017)	0.364			SSNM (ERR) (Yao and Barbosa, 2021)	0.56	Wikipedia2Vec (Yamada et al., 2020) + IF (Liu et al., 2008)	0.222
	PooledInvestment (Pasternack and Roth, 2010)	0.388			*SSNM (US) (Yao and Barbosa, 2021)*	*0.63*	RDF2Vec (Ristoski and Paulheim, 2016) + IF (Liu et al., 2008)	0.261
	MBM (Wang et al., 2015)	0.417					ReprLearning (Yao and Barbosa, 2021) + IF (Liu et al., 2008)	0.359
	CATD (Li et al., 2014)	0.427					*Random*	*0.395*
	TKGC (Huang et al., 2022)	*0.507*						
Mean rank ↓ (FV)			CKRL (Xie et al., 2018)	941				
			SimplE (Kazemi and Poole, 2018)	879				
			Analogy (Liu et al., 2017)	874				
			PTransE (Zhu et al., 2017)	870				
			CrossVal (Wang et al., 2020)	*797*				
MAP ↓ (PED)							ReprLearning (Yao and Barbosa, 2021) + IF (Liu et al., 2008)	0.669
							Wikipedia2Vec (Yamada et al., 2020) + IF (Liu et al., 2008)	0.585
							RDF2Vec (Ristoski and Paulheim, 2016) + IF (Liu et al., 2008)	0.462
							Random	*0.421*
MAE ↓ (PED)	Majority voting (Zheng et al., 2017)	0.134						
	CATD (Li et al., 2014)	0.127						
	PooledInvestment (Pasternack and Roth, 2010)	0.091						
	LTM (Zhao et al., 2012)	0.071						
	TKGC (Huang et al., 2022)	*0.054*						
RMSE ↓ (PED)	Majority voting (Zheng et al., 2017)	0.173						
	CATD (Li et al., 2014)	0.145						
	PooledInvestment (Pasternack and Roth, 2010)	0.108						
	LTM (Zhao et al., 2012)	0.093						
	TKGC (Huang et al., 2022)	*0.062*						

Notes: Metric/dataset (tasks) indicates the metric or dataset is used for the tasks. Up to five scores are represented for each metric-dataset pair.

4.3.1 Dynamic embedding

Table 3 shows the performance of dynamic embedding methods on two incremental datasets. The downstream task is link prediction. The results of three rotation-based embedding methods are demonstrated. RotatH (Wei et al., 2021) recorded the highest accuracy scores in all cases because the hyperplane enhances the ability to capture complex KG patterns. However, adopting a hyperplane increased execution time on the first snapshots of both datasets. By incorporating incremental learning, i.e., merely re-training a few updated triples, RotatH (Wei et al., 2021) reduced execution time and ensured its high accuracy on second and third snapshots.

Table 5 Performance of rule mining methods on WN18RR (Dettmers et al., 2018), FB15-237 (Toutanova and Chen, 2015), YAGO3-10 (Galárraga et al., 2015), and DBpedia (Shiralkar et al., 2017) datasets

Rule mining				
	Method	*MRR* ↑	*Hits@1* ↑	*Hits@10* ↑
WN18RR (Dettmers et al., 2018)	RARL (Pirrò, 2020)	0.360	0.351	0.409
	AMIE+ (Galárraga et al., 2015)		0.358	0.388
	RuleN (Meilicke et al., 2018)		0.427	0.536
	AnyBURL (Meilicke et al., 2019)	*0.470*	*0.441*	*0.552*
	Method	*MRR* ↑	*Hits@1* ↑	*Hits@10* ↑
FB15-237 (Toutanova and Chen, 2015)	RuleN (Meilicke et al., 2018)		0.182	0.420
	RLvLR (Omran et al., 2018)	0.240		
	AnyBURL (Meilicke et al., 2019)	0.310	0.233	0.486
	RARL (Pirrò, 2020)	*0.320*	*0.251*	*0.491*
	RotatE (Sun et al., 2018) (SE)	0.338	0.241	0.533
	RotatH (Wei et al., 2021) (SE)	0.344	0.249	0.536
	HAKE (Zhang et al., 2020) (SE)	*0.346*	*0.250*	*0.542*
	Method	*MRR* ↑	*Hits@1* ↑	*Hits@10* ↑
YAGO3-10 (Galárraga et al., 2015)	RLvLR (Omran et al., 2018)	0.240		0.393
	AnyBURL (Meilicke et al., 2019)	0.540	0.477	0.673
	RARL (Pirrò, 2020)	*0.560*	*0.482*	*0.693*
	RotatE (Sun et al., 2018) (SE)	0.495	0.402	0.670
	HAKE (Zhang et al., 2020) (SE)	0.545	0.462	*0.694*
	RotatH (Wei et al., 2021) (SE)	*0.555*	*0.475*	*0.694*
	Method	*#R (QR)*↑	*ET (min)* ↓	
DBpedia (Shiralkar et al., 2017)	RLvLR (Omran et al., 2018)	855 (183)	351	
	AnyBURL (Meilicke et al., 2019)	20,564 (3,246)	*91*	
	RARL (Pirrò, 2020)	*13,561 (3,564)*	138	

Notes: #R (QR) denotes the number of mined rules (quality rules) and ET is short for execution time. Methods assigned with (SE) mean the approach is originally designed for static embedding of KGs.

4.3.2 FV and PED

Table 4 illustrates the performance of FV and PED approaches on different datasets regarding various metrics. On OKELE, TKGC (Huang et al., 2022) achieved superior performance compared with direct computation, truth inference approaches (Zheng et al., 2017; Li et al., 2014; Pasternack and Roth, 2010; Zhao et al., 2012; Wang et al., 2015), except for recall. It shows that leveraging existing KG facts and subtle comparisons between entities can benefit the FV process. On NELL-314, cross-KG embedding approach CrossVal (Wang et al., 2020) outperformed multiplicative (Yang et al., 2015; Liu et al., 2017; Kazemi and Poole, 2018) and translational (Zhu et al., 2017; Xie et al., 2018; Bordes et al., 2013) embedding method. It indicates that cross-KG negative sampling helps transfer information from a human-curated KG to a target KG and thus improves the FV. Besides, validating facts of OKELE is more challenging than NELL-314, according to the recall scores. The best F1-score on OKELE is 0.507, which needs significant improvement to satisfy the real-world application requirements.

For PED on DBpedia-C, there are records of two variants of SSNM (Yao and Barbosa, 2021) that apply different active learning strategies: uncertainty sampling (US) and error rate reduction (ERR). SSNM (US) achieved better precision, recall, and F1-scores due to the intensive and complicated computation of ERR, which may not work well in the early stage of model training. On DBpedia-F, representation learning (Ristoski and Paulheim, 2016; Yamada et al., 2020; Yao and Barbosa, 2021) combined with outlier detection (Liu et al., 2008) methods tried to resolve the fine-grained typing error detection problem. However, random baseline represented the best scores in three out of four metrics, demonstrating that current PED methods are far from real applications.

Moreover, there are many blanks in Table 4. An approach may perform well on a dataset (metric) but work poorly on another one. It would provide a more comprehensive view of KG evaluation model performances if a benchmark framework is provided for testing upon unified and diverse datasets and metrics.

4.3.3 Rule mining

Table 5 shows the performance of rule mining methods. The values are taken from Pirrò (2020). The downstream task employed to test the ability of rule mining methods is link prediction, which is the same as static embedding approaches. Thus, the performance of static embedding approaches is also included in Table 5 for comparison. On DBpedia (Shiralkar et al., 2017), Tbox-driven RARL (Pirrò, 2020) found the most QRs both in quantity and QR ratio than other Abox-based approaches (Meilicke et al., 2019; Omran et al., 2018). RARL was the best rule learning approach regarding accuracy scores on complex-schema KGs [FB15-237 (Toutanova and Chen, 2015) and YAGO3-10 (Galárraga et al., 2015)] and even competitive to static embedding methods. However, on relatively simple-schema dataset WN18RR (Dettmers et al., 2018), Abox-based rule learning approach AnyBURL (Meilicke et al., 2019) achieved higher scores, and its execution time was shorter by one-third. According to Pirrò (2020), the static embedding approach could not finish running within five hours on the DBpedia dataset. Consequently, Tbox (Abox)-driven rule mining methods better fit complex (simple) schema KGs.

4.3.4 Versioning

Table 6 shows the performance of the KG versioning method HKG Cardoso et al. (2020) on various datasets. For storage, the size of HKG (Cardoso et al., 2020) was smaller than the original ones and saved 61% space on average since the evolution mechanism can arrange and merge KG elements. In addition, HKG (Cardoso et al., 2020) outperformed the baseline in forward and backward query processing in terms of recall. It indicates that information retrieval tasks can benefit from adopting temporal concept evolution, and the KG evolution does not harm information integrity. The precision metric measures the ability of concept prediction, i.e., to determine whether a semantic annotation will evolve and which concept will be used at a specific moment. The precision scores were poor for two reasons. First, several identical attribute values come from different concepts. For example, in NCIt 'salt' is the label of the concept with the code 'C822' and the concept of 'C29974', which causes ambiguity. The other reason was that the heuristic implementation of HKG (Cardoso et al., 2020) sped up the search but caused information loss.

Table 6 Performance of KG versioning method on SNOMED-CT, MeSH, NCIt, ICD9-DM (Cardoso et al., 2020) datasets

	Versioning		
	Method	*Storage (MB)* ↓	*Precision* ↑ *(CP)*
SNOMED-CT (Cardoso et al., 2020)	Origin	1517	
	HKG (Cardoso et al., 2020)	*700*	0.630
	Method	*Storage (MB)* ↓	*Precision* ↑ *(CP)*
NCIt (Cardoso et al., 2020)	Origin	275	
	HKG (Cardoso et al., 2020)	*121*	0.250
	Method	*Storage (MB)* ↓	*Precision* ↑ *(CP)*
ICD9CM (Cardoso et al., 2020)	Origin	88	
	HKG (Cardoso et al., 2020)	*37*	0.700
	Method	*Storage (MB)* ↓	*Precision* ↑ *(CP)*
MeSH (Cardoso et al., 2020)	Origin	229	
	HKG (Cardoso et al., 2020)	*54*	0.450
	Method	*Recall* ↑ *(IR)*	
MeSH (Cardoso et al., 2020)	SimpleMatchQeury (forward)	0.964	
	HKG (Cardoso et al., 2020) (forward)	0.976	
	SimpleMatchQeury (backward)	0.988	
	HKG (Cardoso et al., 2020) (backward)	*0.994*	

Notes: IR and CP are short for information retrieval and concept prediction. Forward (backward) means using terms from MeSH version 2014 (2016) for querying documents stored in 2016 (2014), which are annotated with terms in version 2016 (2014).

5 Findings and future directions

We obtain interesting observations by taking a closer look at the reviewed contents and provide potential future directions.

5.1 Human-in-the-loop

Semi-supervised learning emerged as a favored method in KG proliferation, indicating that a proportion of human supervision is still required. In addition, the performance of model verification is far from practical in its application to real-world data. It calls for novel human-in-the-loop KG evolution mechanisms that enable humans to interact with KG evolution algorithms to reduce annotation costs and improve the model performance iteratively. Active learning would be a promising direction that allows a model to choose annotation targets for low cost and high accuracy.

5.2 One-size-fits-all KGP

Few KG proliferation methods showed versatility by applying to both FV and PED, showing the potential to address multiple aspects of KG evolution. A one-size-fits-all solution leveraging LLMs is a promising research direction to provide adequate initialisation and reduce the burden of human-machine interaction. For example, integrate rules and a few FV examples in prompts, leverage the world knowledge involved in LLMs to judge the correctness of relational or property facts, and provide explanations and evidence.

5.3 Beyond conventional KG

Current KG evolution approaches focus on traditional triple (h, r, t) and temporal quadruple $(h, r, t, time)$, in which data modality is mainly text in a graph structure. MMKG evolution would be an exciting and complicated research area that demands a novel combination of techniques in different fields, such as KG, NLP, and computer vision. Applying existing KG evolution methods to MMKGs is not straightforward. For example, validating a multi-modal fact requires semantically comparing the texts/images between a fact and a validation source (maybe a KG) and then analysing the relatedness among texts and images. In addition, digital media forensic techniques, e.g., determining the authenticity of the image or video, should be considered. If the authenticity is judged as fake, the image/video provenances (Jin et al., 2022) can be high-quality candidates for explanations. Image/video provenance retrieves a set of related images for a target image and constructs the sequence of manipulation operations among retrieved images.

5.4 Efficiency metrics

Various metrics were used in the reviewed papers. Effectiveness-related metrics, such as precision, recall, and Hits@k, achieved far more attention than efficiency metrics. However, recent advances in LLMs are accelerating the KG extraction procedure, requiring KGs to evolve more frequently. More attention should be drawn to efficiency metrics such as execution time, throughput, and required iteration numbers.

5.5 Benchmark dataset with fact label and time

For evaluation datasets, various large-scale KGs are selected as the basis, e.g., YAGO, DBpedia, Freebase, NELL, and Wikidata. YAGO has the most variants. YAGO-Loster and OKELE are accessible datasets assigned with verified fact labels. OKELE contains a reasonable scale of 192K verified facts. YAGO11K and Wikidata12k datasets have timestamps assigned with fact. However, to the best of our knowledge, there is no single accessible dataset containing both fact labels and time. Temporal information is essential for KG evolution since the facts and properties could have temporal dependencies, and some facts have time validity, such as the president of a country.

5.6 Benchmark dataset with explanation

Explanations for FV predictions would help annotators or systems to do cross-validation. There are no explanations for validation results in the datasets listed in Table 2. The closest datasets are FEVER and MultiFC (Vedula and Parthasarathy, 2021) in the NLP domain. However, the inputs are in natural language, requiring KG extraction techniques to acquire triples, which may result in information loss propagation.

5.7 LLM-based KG proliferation

Existing KG proliferation methods typically adopt supervised or semi-supervised learning that requires plenty of high-cost human labels. Unsupervised learning is urgent in dealing with explosively generating information. LLM-based KG proliferation is a promising research direction that could leverage LLMs' powerful language understanding and inference abilities to realise zero-shot or few-shot learning.

A straightforward way is to construct a prompt consisting of a KG proliferation task description, input fact, and desired output such as veracity, explanation, and evidence source, and then deliver the prompt to an LLM to obtain an FV answer. Through our initial preliminary experiments, LLM-based FV achieved superior performance compared to the SOTA method TKGC (Huang et al., 2022) on the OKELE dataset.

According to our test, an interesting observation is that ChatGPT (OpenAI, 2022) verified a fact (`Afghanistan`, `hasCapital`, `Afghanistan`) as true. Then we said a capital cannot be a country, and the LLM apologised and corrected its answer. However, after a few rounds of FV, ChatGPT still made the same mistake. Another one is the fact in Chinese (德川家茂, 妻子, 和宫亲子内亲王), which means Tokugawa Lemochi's wife Princess Kazu no Miya, was verified correctly according to the evidence '德川家茂的御台所 和宫亲子内亲王'. The point is that 御台所 is the Japanese historical calling of 妻子 (wife), and ChatGPT captured the cross-lingual and across-time text meaning.

However, due to LLMs' hallucination issue, the above FV answers contained explanations that were either nonsensical or not grounded in reality. For instance, the apparent fake evidence link https://www.example.com and invalid links resulted in 404 or 400 HTTP errors. Another issue is the outdated knowledge of LLMs. It takes weeks or months to retrain an LLM entirely, which causes the stale knowledge cut-off date. Furthermore, this old knowledge or evidence may lead to inaccurate FV answers.

6 Conclusions

In this paper, we present a survey on KG evolution methods. We distinguish KG evolution into KG proliferation, dynamic embedding, and versioning. KG proliferation is categorised into FV, PED, and rule mining. Evaluation metrics, datasets, and performance of KG evolution methods are also systematically reviewed. Finally, the findings are summarised, and future directions are given.

This work has several limitations. We mainly describe approaches in meaningful categories, not in detail. We hope this work can serve as guidance where specifics can be found in corresponding papers. Moreover, this is a pure survey without any empirical experiments, and it would be helpful to conduct comparative experiments for in-depth analysis and direction. We leave this for future work.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 62306287) and Zhejiang Provincial Natural Science Foundation of China (Grant No. LY23F020012).

References

Ahn, Y., Yoo, S. and Jeong, O. (2023) 'Polarisx2: auto-growing context-aware knowledge graph', *International Journal of Web and Grid Services*, Vol. 19, No. 2, pp.137–155.

Banerjee, S. and Lavie, A. (2005) 'Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments', in *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp.65–72.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J. (2008) 'Freebase: a collaboratively created graph database for structuring human knowledge', in *Int'l Conf. on Management of Data (SIGMOD)*, pp.1247–1250.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. and Yakhnenko, O. (2013) 'Translating embeddings for modeling multi-relational data', in *Int'l. Conf. on Neural Information Processing Systems (NIPS)*, pp.2787–2795.

Cao, E., Wang, D., Huang, J. and Hu, W. (2020) 'Open knowledge enrichment for long-tail entities', in *ACM the Web Conference (WWW)*, pp.384–394.

Cardoso, S.D., Da Silveira, M. and Pruski, C. (2020) 'Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies', *Knowledge-Based Systems*, Vol. 194, No. 105508, pp.1–10.

Chen, L., Jiang, S., Liu, J., Wang, C., Zhang, S., Xie, C., Liang, J., Xiao, Y. and Song, R. (2022) 'Rule mining over knowledge graphs via reinforcement learning', *Knowledge-Based Systems*, Vol. 242, No. 108371, pp.1–13.

Dettmers, T., Minervini, P., Stenetorp, P. and Riedel, S. (2018) 'Convolutional 2D knowledge graph embeddings', in *AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 32.

Diaz Benavides, S., Cardoso, S.D., Da Silveira, M. and Pruski, C. (2022) 'Dyndiff: a tool for comparing versions of large ontologies', in *SeWebMeDa Workshop at Extended Semantic Web Conf. (ESWC)*.

Dong, L., Zhao, D., Zhang, X., Li, X., Kang, X. and Yao, H. (2021) 'Anchors-based incremental embedding for growing knowledge graphs', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 4, pp.3458–3470.

Galárraga, L., Teflioudi, C., Hose, K. and Suchanek, F.M. (2015) 'Fast rule mining in ontological knowledge bases with AMIE+', *The VLDB Journal*, Vol. 24, No. 6, pp.707–730.

Galland, A., Abiteboul, S., Marian, A. and Senellart, P. (2010) 'Corroborating information from disagreeing views', in *ACM Int'l Conf. on Web Search and Data Mining (WSDM)*, pp.131–140.

Hartung, M., Groß, A. and Rahm, E. (2013) 'Conto-diff: generation of complex evolution mappings for life science ontologies', *Journal of Biomedical Informatics*, Vol. 46, No. 1, pp.15–32.

Hoffman, R.R., Mueller, S.T., Klein, G. and Litman, J. (2018) *Metrics for Explainable AI: Challenges and Prospects*, arXiv preprint arXiv:1812.04608.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M. et al. (2004) 'SWRL: a semantic web rule language combining owl and ruleml', *W3C Member Submission*, Vol. 21, No. 79, pp.1–31.

Huang, J., Zhao, Y., Hu, W., Ning, Z., Chen, Q., Qiu, X., Huo, C. and Ren, W. (2022) 'Trustworthy knowledge graph completion based on multi-sourced noisy data', in *ACM the Web Conference (WWW)*, pp.956–965.

Ji, S., Pan, S., Cambria, E., Marttinen, P. and Philip, S.Y. (2021) 'A survey on knowledge graphs: representation, acquisition, and applications', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 2, pp.494–514.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A. and Fung, P. (2023) 'Survey of hallucination in natural language generation', *ACM Computing Surveys*, Vol. 55, No. 12, pp.1–38.

Jia, Z., Li, H. and Chen, L. (2023) 'Air: adaptive incremental embedding updating for dynamic knowledge graphs', in *Int'l Conf. on Database Systems for Advanced Applications (DASFAA)*, Springer, pp.606–621.

Jin, X., Lee, Y., Fiscus, J., Guan, H., Yates, A.N., Delgado, v and Zhou, D.F. (2022) 'MFC-PROV: media forensics challenge image provenance evaluation and data analysis on large-scale datasets', *Neurocomputing*, Vol. 470, pp.76–88.

Kazemi, S.M. and Poole, D. (2018) 'Simple embedding for link prediction in knowledge graphs', *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31, pp.1–12.

Li, N., Yin, S., Li, C., Wang, Y., Xiao, K., Cao, R., Hua, W., Chu, W., Song, X. and Li, C. (2023) 'An uncertainty analysis method based on a globally optimal truth discovery model for mineral prospectivity mapping', *Mathematical Geosciences*, Vol. 56, pp.249–278.

Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W. and Han, J. (2014) 'A confidence-aware approach for truth discovery on long-tail data', *Proceedings of the VLDB Endowment*, Vol. 8, No. 4, pp.425–436.

Li, Y., Rubinstein, B.I.P. and Cohn, T. (2019) 'Truth inference at scale: a Bayesian model for adjudicating highly redundant crowd annotations', in *ACM the Web Conference (WWW)*, pp.1028–1038.

Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y. and Cheng, X. (2021) 'Temporal knowledge graph reasoning based on evolutional representation learning', in *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pp.408–417.

Lin, C-Y. and Hovy, E. (2002) 'Manual and automatic evaluation of summaries', in *Workshop on Automatic Summarization*, pp.45–51.

Liu, F.T., Ting, K.M. and Zhou, Z-H. (2008) 'Isolation forest', in *2008 Eighth IEEE International Conference on Data Mining*, IEEE, pp.413–422.

Liu, H., Wu, Y. and Yang, Y. (2017) 'Analogical inference for multi-relational embeddings', in *Int'l Conf. on Machine Learning (ICML)*, PMLR, pp.2168–2178.

Liu, L., Du, B., Xu, J., Xia, Y. and Tong, H. (2022) 'Joint knowledge graph completion and question answering', in *ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD)*, pp.1098–1108.

Liu, J., Ke, W., Wang, P., Shang, Z., Gao, J., Li, G., Ji, K. and Liu, Y. (2024) 'Towards continual knowledge graph embedding via incremental distillation', in *AAAI Conf. on Artificial Intelligence (AAA))*, Vol. 38, pp.8759–8768.

Loster, M., Mottin, D., Papotti, P., Ehmüller, J., Feldmann, B. and Naumann, F. (2021) 'Few-shot knowledge validation using rules', in *ACM the Web Conference (WWW)*, pp.3314–3324.

Luo, L., Ju, J., Xiong, B., Li, Y-F., Haffari, G. and Pan, S. (2023) *ChatRule: Mining Logical Rules with Large Language Models for Knowledge Graph Reasoning*, arXiv preprint arXiv:2309.01538.

Ma, J., Zhou, C., Wang, Y., Guo, Y., Hu, G., Qiao, Y. and Wang, Y. (2022) 'Ptruste: a high-accuracy knowledge graph noise detection method based on path trustworthiness and triple embedding', *Knowledge-Based Systems*, Vol. 256, No. 109688, pp.1–14.

Meilicke, C., Fink, M., Wang, Y., Ruffinelli, D., Gemulla, R. and Stuckenschmidt, H. (2018) 'Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion', in *Int'l Semantic Web Conf. (ISWC)*, Springer, pp.3–20.

Meilicke, C., Chekol, M.W., Ruffinelli, D. and Stuckenschmidt, H. (2019) 'Anytime bottom-up rule learning for knowledge graph completion', in *Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp.3137–3143.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B. et al. (2018) 'Never-ending learning', *Communications of the ACM*, Vol. 61, No. 5, pp.103–115.

Omran, P.G., Wang, K. and Wang, Z. (2018) 'Scalable rule learning via learning representation', in *Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp.2149–2155.

OpenAI (2022) *Introducing ChatGPT*.

OpenAI (2023) *GPT-4 Technical Report*.

Papineni, K., Roukos, S., Ward, T. and Zhu, W-J. (2002) 'Bleu: a method for automatic evaluation of machine translation', in *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.311–318.

Pasternack, J. and Roth, D. (2010) 'Knowing what to believe (when you already know something)', in *Int'l Conf. on Computational Linguistics (Coling)*, pp.877–885.

Paulheim, H. (2017) 'Knowledge graph refinement: a survey of approaches and evaluation methods', *Semantic Web*, Vol. 8, No. 3, pp.489–508.

Pirrò, G. (2020) 'Relatedness and tbox-driven rule learning in large knowledge bases', in *AAAI Conf. on Artificial Intelligence (AAAI)*, pp.2975–2982.

Ristoski, P. and Paulheim, H. (2016) 'RDF2VEC: RDF graph embeddings for data mining', in *Int'l Semantic Web Conf. (ISWC)*, Springer, pp.498–514.

Shiralkar, P., Flammini, A., Menczer, F. and Ciampaglia, G.L. (2017) 'Finding streams in knowledge graphs to support fact checking', in *IEEE Int'l Conf. on Data Mining (ICDM)*, IEEE, pp.859–864.

Suhara, Y., Li, J., Li, Y., Zhang, D., Demiralp, Ç., Chen, C. and Tan, W-C. (2022) 'Annotating columns with pre-trained language models', in *Int'l Conf. on Management of Data (SIGMOD)*, pp.1493–1503.

Sun, Z., Deng, Z-H., Nie, J-Y. and Tang, J. (2018) 'Rotate: knowledge graph embedding by relational rotation in complex space', in *Int'l Conf. on Learning Representations (ICLR)*.

Sun, W., Shi, Z., Gao, S., Ren, P., de Rijke, M. and Ren, Z. (2023) 'Contrastive learning reduces hallucination in conversations', in *AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 37, pp.13618–13626.

Tang, X., Yuan, R., Li, Q., Wang, T., Yang, H., Cai, Y. and Song, H. (2020) 'Timespan-aware dynamic knowledge graph embedding by incorporating temporal evolution', *IEEE Access*, Vol. 8, pp.6849–6860.

Toutanova, K. and Chen, D. (2015) 'Observed versus latent features for knowledge base and text inference', in *Workshop on Continuous Vector Space Models and their Compositionality*, pp.57–66.

Vedula, N. and Parthasarathy, S. (2021) 'FACE-KEG: fact checking explained using knowledge graphs', in *ACM Int'l Conf. on Web Search and Data Mining (WSDM)*, pp.526–534.

Wang, X., Sheng, Q.Z., Fang, X.S., Yao, L., Xu, X. and Li, X. (2015) 'An integrated Bayesian approach for effective multi-truth discovery', in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp.493–502.

Wang, Y., Ma, F. and Gao, J. (2020) 'Efficient knowledge graph validation via cross-graph representation learning', in *ACM Int'l Conf. on Information & Knowledge Management (CIKM)*, pp.1595–1604.

Wang, X., Jia, R., Fu, L., Jin, H., Tian, X., Gan, X. and Wang, X. (2021) 'Online spatial crowdsensing with expertise-aware truth inference and task allocation', *IEEE Journal on Selected Areas in Communications*, Vol. 40, No. 1, pp.412–427.

Wei, Y., Chen, W., Li, Z. and Zhao, L. (2021) 'Incremental update of knowledge graph embedding by rotating on hyperplanes', in *IEEE Int'l Conf. on Web Services (ICWS)*, IEEE, pp.516–524.

Wu, J., Xu, Y., Zhang, Y., Ma, C., Coates, M. and Cheung, J.C.K. (2021) 'Tie: a framework for embedding-based incremental temporal knowledge graph completion', in *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pp.428–437.

Wu, H., Wang, Z., Wang, K. and Shen, Y-D. (2022) 'Learning typed rules over knowledge graphs', in *Int'l Conf. on Principles of Knowledge Representation and Reasoning (KR)*, Vol. 19, pp.494–503.

Xiao, H. and Wang, S. (2022) 'A joint maximum likelihood estimation framework for truth discovery: a unified perspective', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 6, pp.5521–5533.

Xie, R., Liu, Z., Lin, F. and Lin, L. (2018) 'Does william shakespeare really write Hamlet? Knowledge representation learning with confidence', in *AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 32.

Xu, Y., Ou, J., Xu, H. and Fu, L. (2023) 'Temporal knowledge graph reasoning with historical contrastive learning', in *AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 37, pp.4765–4773.

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y. and Matsumoto, Y. (2020) 'Wikipedia2VEC: an efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia', in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp.23–30.

Yan, L., Yang, K. and Yang, S. (2021) 'Reputation-based truth discovery with long-term quality of source in internet of things', *IEEE Internet of Things Journal*, Vol. 9, No. 7, pp.5410–5421.

Yang, S.J.H., Hsieh, J.S.F., Lan, B.C.W. and Chung, J-Y. (2006) 'Composition and evaluation of trustworthy web services', *International Journal of Web and Grid Services*, Vol. 2, No. 1, pp.5–24.

Yang, B., Yih, S.W-t., He, X., Gao, J. and Deng, L. (2015) 'Embedding entities and relations for learning and inference in knowledge bases', in *Int'l Conf. on Learning Representations (ICLR)*.

Yang, L., Chen, H., Li, Z., Ding, X. and Wu, X. (2023) *ChatGPT is Not Enough: Enhancing Large Language Models with Knowledge Graphs for Fact-Aware Language Modeling*, arXiv preprint arXiv:2306.11489.

Yao, P. and Barbosa, D. (2021) 'Typing errors in factual knowledge graphs: severity and possible ways out', in *ACM the Web Conference (WWW)*, pp.3305–3313.

Ye, C., Wang, H., Zheng, K., Kong, Y., Zhu, R., Gao, J. and Li, J. (2021) 'Constrained truth discovery', in *Int'l Conf. on Data Engineering (ICDE)*, IEEE Computer Society, pp.2356–2357.

Yin, X., Han, J. and Yu, P.S. (2007) 'Truth discovery with multiple conflicting information providers on the web', in *ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pp.1048–1052.

Zhang, Y., Ives, Z. and Roth, D. (2019a) 'Evidence-based trustworthiness', in *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.413–423.

Zhang, X., Wu, Y., Huang, L., Ji, H. and Cao, G. (2019b) 'Expertise-aware truth analysis and task allocation in mobile crowdsourcing', *IEEE Transactions on Mobile Computing*, Vol. 20, No. 3, pp.1001–1016.

Zhang, Z., Cai, J., Zhang, Y. and Wang, J. (2020) 'Learning hierarchy-aware knowledge graph embeddings for link prediction', in *AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 34, pp.3065–3072.

Zhang, J., Chen, B., Zhang, L., Ke, X. and Ding, H. (2021) 'Neural, symbolic and neural-symbolic reasoning on knowledge graphs', *AI Open*, Vol. 2, pp.14–35.

Zhao, B., Rubinstein, B.I.P., Gemmell, J. and Han, J. (2012) 'A Bayesian approach to discovering truth from conflicting sources for data integration', *Proceedings of the VLDB Endowment*, Vol. 5, No. 6, pp.550–561.

Zheng, Y., Li, G., Li, Y., Shan, C. and Cheng, R. (2017) 'Truth inference in crowdsourcing: is the problem solved?', *Proceedings of the VLDB Endowment*, Vol. 10, No. 5, pp.541–552.

Zheng, L., Cheng, P., Chen, L., Yu, J., Lin, X. and Yin, J. (2022) 'Crowdsourced fact validation for knowledge bases', in *Int'l Conf. on Data Engineering (ICDE)*, IEEE, pp.938–950.

Zhu, H., Xie, R., Liu, Z. and Sun, M. (2017) 'Iterative entity alignment via joint knowledge embeddings', in *Int'l Joint Conf. on Artificial Intelligence*, Vol. 17, pp.4258–4264.

Zhu, X., Li, Z., Wang, X., Jiang, X., Sun, P., Wang, X., Xiao, Y. and Yuan, N.J. (2022) 'Multi-modal knowledge graph construction and application: a survey', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 2, pp.715–735.

Zillner, S. and Winiwarter, W. (2005) 'Integration of ontological knowledge within the authoring and retrieval of multimedia metaobjects', *International Journal of Web and Grid Services*, Vol. 1, Nos. 3–4, pp.397–415.