Knowledge-Based Systems 255 (2022) 109597

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# A comprehensive overview of knowledge graph completion

Tong Shen, Fu Zhang *, Jingwei Cheng

*School of Computer Science and Engineering, Northeastern University, Shenyang, 110169, China*

ARTICLE INFO

*Article history:*
Received 24 March 2022
Received in revised form 24 July 2022
Accepted 3 August 2022
Available online 12 August 2022

*Keywords:*
Knowledge Graph Completion (KGC)
Classification
Comparisons and analyses
Performance evaluation
Overview

ABSTRACT

Knowledge Graph (KG) provides high-quality structured knowledge for various downstream knowledge-aware tasks (such as recommendation and intelligent question-answering) with its unique advantages of representing and managing massive knowledge. The quality and completeness of KGs largely determine the effectiveness of the downstream tasks. But in view of the *incomplete* characteristics of KGs, there is still a large amount of valuable knowledge is missing from the KGs. Therefore, it is necessary to improve the existing KGs to supplement the missed knowledge. *Knowledge Graph Completion* (*KGC*) is one of the popular technologies for knowledge supplement. Accordingly, there has a growing concern over the KGC technologies. Recently, there have been lots of studies focusing on the KGC field. To investigate and serve as a helpful resource for researchers to grasp the main ideas and results of KGC studies, and further highlight ongoing research in KGC, in this paper, we provide a all-round up-to-date overview of the current state-of-the-art in KGC.

According to the information sources used in KGC methods, we divide the existing KGC methods into two main categories: the KGC methods relying on structural information and the KGC methods using other additional information. Further, each category is subdivided into different granularity for summarizing and comparing them. Besides, the other KGC methods for KGs of special fields (including temporal KGC, commonsense KGC, and hyper-relational KGC) are also introduced. In particular, we discuss comparisons and analyses for each category in our overview. Finally, some discussions and directions for future research are provided.



## 1. Introduction

Knowledge Graphs (KGs) describe the concepts, entities, and their relations in a structured triple form, providing a better ability to organize, manage and understand the mass of information on the world [1]. In recent years, KG plays an increasingly important role in lots of knowledge-aware tasks, and especially brings vitality to intelligent question answering, information extraction, and other artificial intelligence tasks [1–3]. There are a number of large-scale KGs such as DBPedia [4], Freebase [5], WordNet [6], and YAGO [7] (as shown in Table 1), which have been widely exploited in many knowledge-aware applications. *Facts* in these KGs are generally represented in a form of *triple*: (*subject*, *predicate*, *object*), which be regarded as the fundamental data structure of KGs and preserves the essential semantic information of KGs [8].

Although KGs are of great value in applications, they are still characterized by **incompleteness** because a large amount of valuable knowledge exists implicitly or misses in the KGs [1]. Some data indicate that the deficiency rate of some common basic relations in the current large KGs was more than 70% [9], while other less universal relations are even more lacking. **Knowledge Graph Completion (KGC)** aims to predict and replenish the missing parts of triples. As one of a popular KGC research direction, *Knowledge Graph Embedding (KGE)* (or *Knowledge Graph Representation Learning*) has been proposed and quickly gained massive attention. KGE embeds KG components (e.g. entities and relations) into continuous vector spaces to simplify the manipulation and preserve the inherent structure of the KG simultaneously [10–15]. Recently, there have been lots of studies focusing on the KGC field. To facilitate the research of the KGC task and follow the development in the KGC field, more and more review articles to sort out and summarize the recent KGC technologies.

Accordingly, several previous overviews on the KGC techniques are provided:

- Wang et al. [16] make the most relevant review with respect to KGC studies from 2012 to 2016. They first coarsely group KGE models according to their input data (the input data including facts only or incorporating additional information. The additional information in [16] involves entity types, relation paths, textual descriptions, logical rules, and a slight mention of several other information, such as entity attributes and temporal information). Then they further make finer-grained categorizations based on the above

* Corresponding author.
*E-mail address:* zhangfu@mail.neu.edu.cn (F. Zhang).

https://doi.org/10.1016/j.knosys.2022.109597
0950-7051/

**Table 1**
Several famous KGs.

| KG | Fact | Entity | Relation | Relying resource |
|---|---|---|---|---|
| DBpedia | 538M | 4.8M | 2813 | Wikipedia, Expertise |
| YAGO | 447M | 9.8M | 114 | WordNet, Wikipedia |
| Freebase | 2400M | 50M | 37 781 | Wikipedia, Expertise, Swarm-intelligence |
| NELL | 0.4M | 2M | 425 | Human-supplied |
| Wikidata | 65M | 15M | 1673 | Freebase, Swarm-intelligence |
| CN-DBPedia | 222M | 16M | – | Wikipedia, Expertise |
| Google KG | 18 000M | 570M | 35 000 | Freebase |

grouping (e.g., the methods that only consider facts involve two categories, distance-based KGC methods and semantic matching-based KGC). However, the work [16] *is not a specific overview for the KGC task, this overview just takes KGC as one of the downstream applications that KGE technologies can support*.

- Gesese et al. [17] make a brief summary of KGC tasks that only adds less than ten more recent articles compared with [16]. Moreover, the work [17] mainly focuses on the *KGE technology related to literal information*. The literal information in [17] indicates the text descriptions, numerical values, images, or their combinations.
- Rossie et al. [18] summarize *16 recent Link Prediction (LP) models based on KG embeddings*. However, the work [18] does not refer to other KGC tasks, such as Triple Classification (TC) and Relation Prediction (RP) (we will give a specifical introduction to these KGC tasks in Section 2.1).
- Also, the other two overviews [19,20] briefly list and state several KGC-related studies. They neither make a thorough and careful introduction to specific KGC technical details nor cover major KGC approaches. In addition, several surveys [21–23] focus on the KG field but do not discuss specific works on KGC.

With the development of technologies such as Transformer and pre-trained language models (e.g., BERT) in the past few years, a large number of novel KGC techniques have appeared, which are either not covered or summarized in detail in the existing surveys. Besides, except for the information mentioned in [16,17], more kinds of additional information such as entity neighbors, multi-hop relation paths, and third-party data sources are used in the KGC field. Intuitively, the KGC methods based on the additional information should be divided into much wider scopes with more details.

**Compared with the overviews above**, in this paper we propose a more comprehensive and fine-grained division overview on Knowledge Graph Completion (KGC). Our paper covers almost all of the mainstream KGC techniques up to now. Our overview provides more careful classification for the different level of KGC categories. In detail, **we make the following main contributions**:

**(1)** From the perspective of *comprehensiveness*, *we provide a more comprehensive and systematic survey* about the KGC field. We pay particular attention to the literature from 2017 to now, which is either not summarized in [16] or not detailedly introduced in the other previous overviews [17,19,20], and [18]. Also, we consider some special KGC techniques, including Temporal Knowledge Graph Completion (TKGC), CommonSense Knowledge Graph Completion (CSKGC), and Hyper-relational Knowledge Graph Completion (HKGC).

**(2)** From the perspective of detailed *classification* and *summarization*, we summarize the recent KGC researches into *two main categories* depending on whether rely on the additional information of KGs: KGC merely with the *structural* information of KGs and KGC with the *additional* information. For the *former* category, KGC methods are reviewed under three categories: Tensor/matrix factorization models, Translation models, and Neural Network models. For the *latter* category, we further divide it into two sub-categories: KGC methods based on the internal information inside KGs and KGC methods relying on the extra information outside KGs. When we introduce the internal information-based KGC methods, we take account of five categories of information, including node literals, entity-related information, relation-related information, neighborhood information, and relational path information. Moreover, extra information-based KGC includes two families: rule-based KGC and KGC based on third-party data sources.

**(3)** From the perspective of *comparison* and *analysis*, for each KGC category, we carry on the detailed comparison of diverse granularity in *both theory and experiment* of introduced KGC methods. We also make thorough analysis and summary on it. On this basis, we give a global discussion and prospect for the *future research directions* of KGC.

The remainder of the paper is structured as follows: we first give an overview of KG notations, definitions, technological process, datasets, evaluation criteria, as well as our categorization criteria in Section 2; then we discuss the two categories of KGC methods relying on the structural information of KG and using the additional information in Section 3 and Section 4; next, our review goes to three special technologies of KGC in Section 5. In Section 6, we make a discussion on outlook research directions. Finally, we make a conclusion in Section 7.

## 2. Notations of knowledge graph completion and our categorization criterion

We first give some notations of KGC in Section 2.1. Then we further introduce a general process of KGC (see Section 2.2), where several key steps of KGC are provided. Further, we summarize the main KGC datasets and evaluation criteria for KGC in Section 2.3. We also briefly introduce the knowledge graph refinement (KGR) technique, which is related to KGC (see Section 2.4). Final, we give our categorization criterion (see Section 2.5).

### 2.1. Notations of KGC

To conveniently introduce various KGC models, this paper gives some notations of KGC as follows: we define a knowledge graph (KG) as $G = (E, R, C)$, where $E = (e_1, e_2, \ldots, e_{|E|})$ is the set of all entities contained in the KG. The total number of entities is $|E|$. $R = (r_1, r_2, \ldots, r_{|R|})$ represents the set of all relations in KG with counts of $|R|$. $\mathcal{T} \subseteq E \times R \times E$ represents the whole triple set in the KG. Each triple is represented as $(h, r, t)$, $h$ and $t$ mean the head entity and the tail entity, and $r$ is the relation between the head entity and the tail entity. During KGE, the entities $h$, $t$ and relations $r$ in the KG are mapped to the constantly low dimensional vectors: $v_h$, $v_r$ and $v_t$. We define the scoring function of KGC models as $s(h, r, t)$ to estimate the plausibility of any fact $(h, r, t)$. In training phase, we formally define their loss objective as $\mathcal{L}$.

KGC can be divided into three subtasks: **triple classification**, **link prediction** and **relation prediction**. Triple classification is an important task in KGC which determines whether to add a triple to KGs by estimating whether this triple is true or not. Link prediction task refers to the process of finding the missing entity when the head entity or tail entity in the triple is missing. Relation prediction judges the probability of establishing the specific

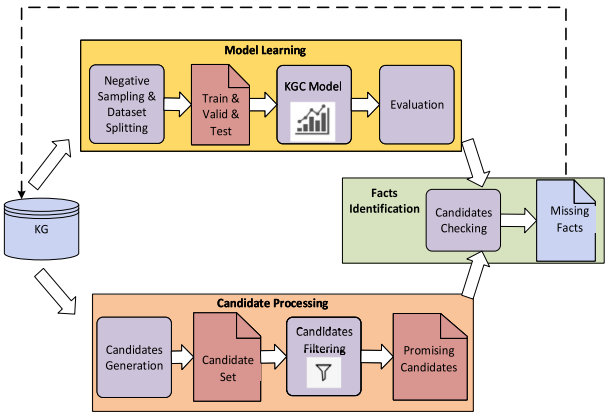



**Fig. 1.** The general KGC process.

relations between two entities. The three subtasks of KGC can be formulated as follows:

**Triple classification (TC)**: Given a triple $(e_i, r_k, e_j)$, the goal is to determine whether the current triple is true.

**Link prediction (LP)**: Given a triple $(?, r_k, e_j)$ or $(e_i, r_k, ?)$, the goal is to predict the missing head entity or tail entity "?".

**Relation prediction (RP)**: Given the partial triple $(e_i, ?, e_j)$, the goal is to predict missing relations between $e_i$ and $e_j$.

### 2.2. An overview of KGC process

In this part, we give a general introduction to the whole technological process of KGC in Section 2.2.1. In addition, we describe two training techniques: *negative sampling* and *ranking setting* in Section 2.2.2. Fig. 1 illustrates the typical workflow of a KGC process.

#### 2.2.1. General KGC procedure

As it can be seen in Fig. 1, generally, a KGC process involves three parts: **model learning**, **candidate processing** and **facts identification**.

• **Model learning.** First, before building a KGC model, there is usually a *pre-processing* step in charge of data preparation, which includes *negative sampling* (sometimes it is not a necessary step, it also can be done online during model training) and *datasets splitting*. The *negative sampling* aims to add a variable amount of negative examples into the original KG to respond to the problem that KGs only contain positive examples [24]. The *datasets splitting* is responsible for splitting the pending KG data into a training set, a validation set, and a testing set. The split datasets will next be used to train and evaluate the KGC model. Then, the *KGC model* usually is a classification model or a ranking model, whose target is predicting whether a candidate triple is correct or not for a KG. Generally, the learned KGC model tends to undergo an *evaluation process* to be assessed through a variety of evaluation metrics. A satisfied assessed result usually means a good KGC model.

• **Candidate processing.** The candidate processing aims to obtain verifiable triples. Those triples will be checked by the learned KGC model in **model learning**. The candidate processing starts with *candidate set generation*, which generates a candidate set (the set of candidates are the triples that possibly be correct but are not present in the KG) relying on algorithms or manual works. Since the initial generated candidate set tends to be very large regardless of whether the candidates are promising or not, it has to further subsequently go through a *candidate filtering* [25] step to preemptively remove those unlikely candidates and simultaneously keep as many promising candidates as possible. Usually, the filtering work is accomplished by generating several filtering rules (also known as "pruning strategies") and applying these rules to the candidate set to produce the condensed set of most promising candidates [26].

• **Facts identification.** Finally, the learned KGC model in **model learning** is applied to the above set of promising candidates generated by **candidate processing**, resulting in the set of missing triples that are considered correct and are likely to be added into the KG [26].

#### 2.2.2. Two training techniques: Negative sampling and ranking setting

***(1) Negative sampling***

**Basic idea of negative sampling.** The existing triples in a given KG are all correct triples, i.e., $(h, r, t) \in \mathcal{T}$, where $\mathcal{T}$ means a positive triple set. Since a KGC model needs to be trained and verified with the help of negative triples, it is necessary to perform negative sampling, i.e., to construct negative triples and build a negative triple set $\mathcal{T}'$. In general, the negative sampling is to replace one entity or relation (there are two options in practice: replace only entity elements or replace both entities and relations in a triple) randomly from a correct triple to make it become an incorrect triple. For example, for the case of (*Bill Gates*, *gender*, *male*), when we replace "Bill Gates" with other random entities in the KG, such as "Italy", a negative triple (*Italy*, *gender*, *male*) is formed, and it is a negative triple (whose label is "false"). However, sometimes the triples formed after random replacement are still true. For example, in the above example, if the head entity is randomly replaced with another entity "Steve Jobs" in the KG, we find that the triple becomes (*Steve Jobs*, *gender*, *male*) and it is still valid. Under this situation, we normally consider filtering out this kind of "negative triple" from $\mathcal{T}'$.

**Sampling strategy.** We introduce three kinds of common sampling strategies: *uniform sampling ("unif")*, *Bernoulli negative sampling method ("bern")* [15], and *generative adversarial network (GAN)-based negative sampling* [27].

• *Uniform sampling ("unif")* is a comparatively simple sampling strategy, which aims to sample negative triples according to the uniform distribution of sampling. In this way, all entities (or relations) are sampled in the same probability.

• *Bernoulli negative sampling method ("bern")* [15]: due to the unbalanced distribution of the number of head entities and tail entities corresponding to a certain relation, i.e., the existence of multiple types of relations including "one to many", "many to one", and "many to many" relations, it is not reasonable to replace the head entities or the tail entities with a uniform manner. Therefore, the "bern" strategy [15] replaces the head entity or the tail entity of a triple under different probabilities. Formally, for a certain relation $r$, "bern" counts the average number of head entities corresponding to per tail entity (denoted as *hpt*) and the average number of tail entities corresponding to per head entity (denoted as *tph*) in all triples with the relation $r$, and then it samples each head entity with probability $\frac{tph}{tph+hpt}$, similarly, it samples each tail entity with probability $\frac{hpt}{tph+hpt}$. The "bern" sampling technique performs well in many tasks, it can reduce false-negative tags than "unif".

• *GAN-based negative sampling* [27]: Inspired by the wide application of generative adversarial network (GAN) [27], Cai et al. [28] change the way of generating negative samples to GAN-based sampling in a reinforcement learning way, in which GAN

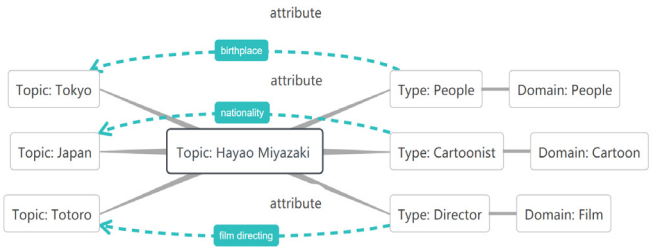



**Fig. 2.** The data example in Freebase.

generator is responsible for generating negative samples, while the discriminator can use translation models to obtain the vector representation of entities and relations, then it scores the generated negative triples and feeds related information back to the generator to provide experience for its negative samples generation. Recently, there have appeared a series of negative sampling techniques based on GAN (e.g. [29–31]), relevant experiments have shown that this kind of methods can obtain high-quality negative samples, which are conducive to classify triples correctly in the training process of knowledge representation model.

***(2) Ranking setting***

In link prediction (LP) task, the evaluation is carried out by performing head prediction or tail prediction on all test triples, and computing for each prediction how the target entity ranks against all the other ones. Generally, the model expects the target entity to yield the highest plausibility. When computing the predicted ranks, two different settings, raw and filtered scenarios, are applied. Actually a prediction may have more than one valid answer: taking an example with the tail predicting for (*Barack Obama*, *parent*, *Natasha Obama*), a KGC model may associate a higher score to *Malia Obama* than to *Natasha Obama*, i.e., there may exist other predicted fact that has been contained in the KG, e.g. (*Barack Obama*, *parent*, *Malia Obama*). Depending on whether valid answers should be considered acceptable or not, two separate settings have been devised [18]:

• *Raw setting*: in this case, valid entities outscoring the target one are considered as mistakes [18]. Thus for a test fact $(h, r, t)$ in a testing set, the raw rank $rank_h$ of the target head entity $h$ is computed as follows (analogous for the tail entity):

$$rank_h = |e \in E \backslash \{h\} : s(e, r, t) > s(h, r, t)| + 1$$

• *Filtered setting*: in this case, valid entities outscoring the target one are not considered as mistakes [18], they are filtered out when computing the rank: for the test fact $(h, r, t)$, the filtered rank $rank_h$ of the target head entity $h$ is computed as (analogous for the tail entity):

$$rank_h = |e \in E \backslash \{h\} : s(e, r, t) > s(h, r, t) \wedge (e, r, t) \notin \mathcal{T}| + 1$$

### 2.3. Datasets and evaluation metrics

Here we introduce some most frequently used datasets for KGC (see Section 2.3.1) and several evaluation metrics for KGC (see Section 2.3.2).

#### 2.3.1. Datasets

We describe the datasets mainly developed on two KGs: Freebase and WordNet, and report some of their important attributes in Table 2.

• **Freebase**: Freebase is a public KG, whose content is added all by users. Moreover, Freebase also extracts knowledge from opening KGs as a supplement [26]. The fundamental data items in Freebase including "Topic", "Type", "Domain", "Property" and so on. We give a demonstration to illustrate the data in the Freebase as Fig. 2. Topic *Miyazaki Hayao* is a *cartoonist* in the field of *cartoon* domain, but a *director* in *movie* domain. It can be seen that Freebase is a database consists of multiple domains expanded by topics, the graph structure of every topic is controlled by its type and type properties. Typically, the subset FB15k and FB13 of Freebase, as well as the improved FB15k-237 based on FB15k, are generally used as experimental benchmarks for method detection in KGC:

(1) **FB15k**: FB15K is created by selecting the subset of entities that are also involved in the Wikilinks database and that also possess at least 100 mentions in Freebase [11]. In addition, FB15K removes reversed relations (where reversed relations like '!/people/person/nationality' just reverses the head and tail compared to the relation '/people/person/nationality'). FB15k describes the ternary relationship between synonymous sets, and the synonym sets that appear in the verification set and testing set also appear in the training set. Also, FB15k converts n-ary relations represented with reification into cliques of binary edges, which greatly affected the graph structure and semantics [18]. FB15K has 592,213 triples with 14,951 entities and 1345 relations which were randomly split as shown in Table 2.

(2) **FB15k-237** is a subset of FB15k built by Toutanova and Chen [32], which is aroused to respond to the test leakage problem due to the presence of near-identical relations or reversed relations FB15k suffering from. Under this background, FB15k-237 was built to be a more challenging dataset by first selecting facts from FB15k involving the 401 largest relations and removing all equivalent or reverse relations. Then they ensured that none of the entities connected in the training set are also directly linked in the validation and testing sets for filtering away all trivial triples [18].

• **WordNet** [6]: WordNet is a large cognitive linguistics based KG ontology, also can be regarded as an English Dictionary knowledge base, whose construction process considers the alphabetic order of words and further form semantic web of English words. In WordNet, entities (called synsets) correspond to semantics, and relational types define the lexical relations between these semantics. Besides, WordNet not only contains multiple types of words such as polysemy, categories classification, synonymy and antonymy, but also includes the entity descriptions. Furthermore, there are various post-produced subset datasets extracted from WordNet, such as WN11, WN18, and WN18RR:

(1) **WN11**: it includes 11 relations and 38 696 entities. What is more, the train set, the validation set, and the test set of WN11 contain 112 581, 2609, and 10 544 triples, respectively [11].

(2) **WN18**: it uses WordNet as a starting point and then iteratively filters out entities and relationships with too few mentions [11, 18]. Note that WN18 involves reversible relations.

(3) **WN18RR**: WN18RR is built by Dettmers et al. [33] for relieving test leakage issue in WN18 that test data being seen by models at training time. It is constructed by applying a pipeline similar to the one employed for FB15k-237 [32]. Recently, they acknowledge that 212 entities in the testing set do not appear in the training set, making it impossible to reasonably predict about 6.7% test facts.

**Table 2**
Common KGC benchmarks and their attributes.

| Benchmark | Entity | Relation | #Training | #Validation | #Test |
|---|---|---|---|---|---|
| WN11 | 38 696 | 11 | 112 581 | 2609 | 10 544 |
| WN18RR | 40 493 | 11 | 86 835 | 3034 | 3134 |
| FB13 | 75 043 | 13 | 316 232 | 5908 | 23 733 |
| FB15k | 14 951 | 1345 | 48 3142 | 50 000 | 59 071 |
| FB15k-237 | 14 541 | 237 | 272 115 | 17 535 | 20 466 |

**Table 3**
Detailed computing formulas of evaluation metrics for KGC.

| Metrics | Computing formula | Notation definition | Task |
|---|---|---|---|
| MRR | $MRR = \frac{1}{\|Q\|}\sum_{\|Q\|}^{i=1}\frac{1}{rank_i}$ | $Q$: query sets; $\|Q\|$: queries numbers; $rank_i$: the rank of the first correct answer for the $i$th query | LP, RP |
| MR | $MRR = \frac{1}{\|Q\|}\sum_{\|Q\|}^{i=1} rank_i$ | $Q$: query sets; $\|Q\|$: queries numbers; $rank_i$: the rank of the first correct answer for the $i$th query | LP, RP |
| Hits@n | $Hits@n = \frac{1}{\|Q\|}Count(rank_i \le n)$, $0 < i \le \|Q\|$ | $Count()$: the hit test number in the top $n$ rankings among test examples; $Q$: query sets; $\|Q\|$: queries numbers; $rank_i$: the rank of the first correct answer for the $i$th query | LP, RP |
| MAP | $MAP = \frac{1}{\|Q\|}\sum_{q\in Q} AP_q$ | $AP_q$: average precision of the query $q$; $Q$: query sets; $\|Q\|$: queries numbers | LP, RP |
| Accuracy | $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ | $TP$: true positive; $FP$: false positive; $FN$: false negative; $TN$: true negative | TC |
| Precision | $Precision = \frac{TP}{TP+FP}$ | $TP$: true positive; $FP$: false positive | TC |
| Recall | $Recall = \frac{TP}{TP+FN}$ | $TP$: true positive; $FN$: false negative | TC |
| F1 score | $F1 = \frac{2*Recall*Precision}{Recall+Precision}$ | — | TC |

#### 2.3.2. *Evaluation metrics*

In this section, we recommend evaluation metrics generally used in KGC. Table 3 shows detailed computing formulas of these mentioned metrics.

**Mean Reciprocal Rank (MRR)**: MRR is widely used in the ranking problem which tends to return multiple results, such as LP and RP task for KGC. When dealing with such problems, the evaluation system will rank the results by their scores from high to low. MRR evaluates a ranking algorithm according to its ranking of the target answer. The higher the target answer ranks, the better the ranking algorithm. In a formulaic view, for a query, if the target answer ranks $n$th, then the MRR score is calculated as $\frac{1}{n}$ (if there is no target answer among returned results, the score is 0).

**Mean-Rank (MR)** and **Hits@n**: Similar to MRR and generally used in the Top-K ranking problem, MR and Hits@n are common metrics in KGC evaluation, especial in LP and RP tasks. MR represents the average ranks of target entity (or relation) in the testing set; Hits@n (usually, $n = 1, 3, 10$) indicates the proportion in the testing set that predicted target entities (or relations) ranks in the top $n$. The ranks are computed according to each prediction's scoring.

**Accuracy**: Accuracy refers to the ratio of correctly predicted triples to the total predicted triples, it usually is applied to evaluate the quality of classification models in TC task for KGC, its calculation formula is demonstrated in Table 3.

**Other evaluation metrics**: There are other evaluation metrics for KGC tasks, such as **Mean Average Precision (MAP)** pays attention to the relevance of returned results in ranking problem. Some metrics closely related to "accuracy" in measuring the classification problems, like **"recall"**, **"precision"** and **"F1 score"**. Compared with MR, MRR, Hits@n, and "accuracy", these metrics are not continually employed in the field of KGC. The detailed computing formulas of these mentioned metrics can be found in Table 3.

### 2.4. Knowledge Graph Refinement (KGR) vs. KGC

The construction process of large-scale KGs results that the formalized knowledge in KGs cannot reasonably reach both "full coverage" and "fully correct" simultaneously. KGs usually need a good trade-off between completeness and correctness. Knowledge Graph Refinement (KGR) is proposed to infer and add missing knowledge to the graph (i.e., KGC), and identify erroneous pieces of information (i.e., error detection) [24]. Recently, KGR is incorporated into recommender systems [34]. Tu et al. [34] exploit the KG to capture target-specific knowledge relationships in recommender systems by distilling the KG to reserve the useful information and refining the knowledge to capture the users' preferences.

Basically, KGC is one of the KGR subtasks to conduct inference and prediction of missing triples. Error detection (e.g., [35,36]) is another KGR subtask for identifying errors in KGs. Jia et al. [36] establish a knowledge graph triple trustworthiness measurement model that quantifies the semantic correctness of triples and the true degree of the triples expressed. But note that KGC is a relatively independent task to increase the coverage of KGs for alleviating the incompleteness of KGs. In our current overview, we focus on the KGC techniques, and the issues about KGR can refer to [24,34].

### 2.5. *Our categorization principle*

The main full-view categorization of our review on KGC studies is shown in Fig. 3.

To follow the experienced rapid development of KGC models, we provide wide coverage on emerging researches for advanced KGC technologies. We include the main literature since the beginning of KGC research as comprehensive as possible and take care of the far-reaching and remarkable approaches in detail. We divide KGC methods into two main categories according to whether using additional information: **Structure (triple) information-based KGC methods** and **Additional information-based KGC methods** (the additional information typically refers to some other information that included inside or outside of KGs except for the structure information, such as text description, artificial rules). Moreover, we further consider the source of additional information — depending on whether it comes from the inner KG, we *classify the additional information into two finer subclasses*: **internal side information inside KGs** and **external extra information outside KGs**. In addition, we introduce some KGC techniques targeting certain fields, like **Temporal Knowledge Graph Completion (TKGC)**, **CommonSense KGC (CSKGC)** and

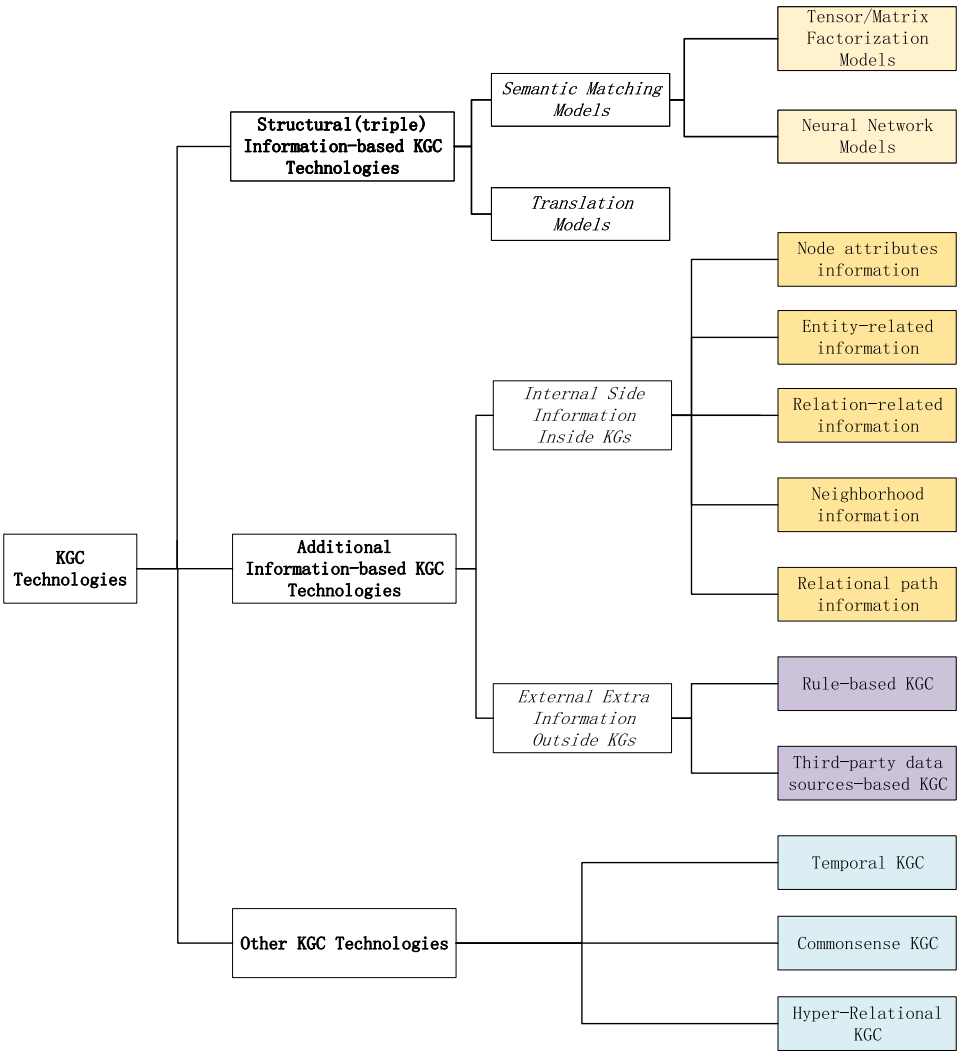



**Fig. 3.** Our Categorization Principle.

**Hyper-relational KGC (HKGC)**. We also make a detailed comparison and summary among the methods of each small category. We give a global discussion and prospect for the future research directions of KGC. Specifically, our categorization principle is as follows:

• **Structure information-based KGC methods**: which only use the structure information of internal facts in KGs. For this category, KGC is reviewed under **semantic matching models** and **translation models** according to the nature of their scoring functions. The semantic matching models generally use semantic matching-based scoring functions and further consists of *tensor/matrix factorization models* and *neural network models*. The translation models apply distance-based scoring function;

• **Additional information-based KGC methods**: which cooperate with additional information (the inside or outside information of KGs except for the structure information) to achieve KGC. For this category, we further propose fine-grained taxonomies respective into two views about the usage of inside information or outside information:
(1) **Internal side information inside KGs** involved in KGs, including *node attributes information*, *entity-related information*, *relation-related information*, *neighborhood information*, *relational path information*;
(2) **External extra information outside KGs** outside KGs, mainly including two aspects: *rule-based KGC* and *third-party data sources-based KGC*.

• **Other KGC technologies**: we take additional attention on some other KGC techniques, such as **Temporal KGC**, **CommonSense KGC** and **Hyper-relational KGC**.

## 3. Structural information-based KGC technologies

In this section, we focus on KGC technologies relying on structure information only, give an account of several categories of methods belonging to this kind of KGC technologies: *Semantic Matching models* in Section 3.1 and *Translation models* in Section 3.2.

### 3.1. Semantic matching models

Semantic Matching models is a kind of models which *compute semantic matching-based scoring functions* by measuring the semantic similarities of entity or relation embeddings in latent embedding space. In this category, we introduce two subclasses: *Tensor/Matrix Factorization Models* (see Section 3.1.1) and *Neural Network Models* (see Section 3.1.2).

**Table 4**
Characteristics of Tensor Factorization (TF) KGC methods.

| Model | Highlight | Score function | Loss function[a] | Parameters & Constrains |
|---|---|---|---|---|
| Tucker-based TF methods | | | | |
| TuckER [37] | Tucker decomposition, multi-task learning | $s(h,r,t) = \mathcal{W} \times_1 v_h \times_2 v_r \times_3 v_t$ | Bernoulli $\mathcal{L}_{log}$ | $\mathcal{W} \in \mathbb{R}^{d_e \times d_r \times d_e}$, $v_h, v_t \in \mathbb{R}^{d_e}$, $v_r \in \mathbb{R}^{d_r}$ |
| DEDICOM-based TF methods | | | | |
| RESCAL [13] | Three-way bilinear TF | $s(h,r,t) = v_h^T M_r v_t$ | $\mathcal{L}_2$ | $v_h, v_t \in \mathbb{R}^d, M_r \in \mathbb{R}^{d \times d}$ |
| LFM [38] | Bilinear TF, decomposing the relation matrix $R_j$, decreasing parameters of RESCAL | $s(h,r,t) \triangleq y^T M_r y' + {v_t}^T M_r z + z'^T M_r v_t + {v_t}^T M_r v_t$, $R_j = \sum_{r=1}^{d} \alpha_r^j \Theta_r = u_r v_t^T$ | $\mathcal{L}_{log}$ | $R_j \in \mathbb{R}^{p \times p}$, $y, y', z, z' \in \mathbb{R}^p$ $u_r, v_r \in \mathbb{R}^p, \alpha^j \in \mathbb{R}^d$ |
| Tatec [39] | 2-way and 3-way interactions models, hard regularization, soft regularization | $s(h,r,t) = s_1(h,r,t) + s_2(h,r,t)$ $s_1(h,r,t) = v_{r_1}^T v_{h_1} + v_{r_2}^T v_{t_1} + v_{h_1}^T D_{diag} v_{t_1}$ $s_2(h,r,t) = v_{h_2}^T M^r v_{t_2}$ | $\mathcal{L}_{marg}$ $+\Delta_{soft/hard}$ | $v_{h_i}, v_{t_i} \in \mathbb{R}^{d_i}, i = 1,2$ $v_{r_1}, v_{r_2} \in \mathbb{R}^{d_1}$, $M_r \in \mathbb{R}^{d_2 \times d_2}$ |
| ANALOGY [40] | Bilinear TF, normality relation matrix commutativity relation matrix | $s(h,r,t) = v_h^T M_r v_t$ $M_r M_r^T = M_r^T M_r, \ \forall r \in \mathbb{R}$, $M_r M_{r'} = M_{r'} M_r, \ \forall r \in \mathbb{R}$. | $\mathcal{L}_{logistic}$ | $h, t \in \mathbb{R}^d, M_r \in \mathbb{R}^{d \times d}$ |
| REST [41] | Subgraph tensors building, RW-based SGS, predicate sparsification operator, Focused Link Prediction (FLP) | $for\ quary\ (h,r,?): \ v_e = v_h^T M_r A$ $s(h,r,t) = v_h^T M_r v_t$ | $\mathcal{L}_2$ | $v_h, v_t \in \mathbb{R}^d, M_r \in \mathbb{R}^{d \times d}$ $A \in \mathbb{R}^{N_e \times d}$ |
| CP-based TF methods | | | | |
| DistMult [42] | RESACL + diagonal matrices | $s(h,r,t) = v_h^T M_{r\,diag} v_t$ | $max\ \mathcal{L}_{marg}$ | $M_{r\,diag} = diag(r), r \in \mathbb{R}^d$ |
| ComplEx [43] | Complex values CP-based TF model | $s(h,r,t) = Re(v_h^T M_{r\,diag} \bar{v}_t)$ $= Re(\sum_{i=0}^{d-1} [v_r]_i \cdot [v_h]_i \cdot [\bar{v}_t]_i)$ | $\mathcal{L}_{nll} + \Delta_{L_2}$ | $v_h, v_t \in \mathbb{C}^d$, $M_{r\,diag} = diag(v_r), v_r \in \mathbb{C}^d$ |
| SimplE [44] | Bilinear TF model, utilizing inverse relations, fully expressive-evaluation metric | $s(h,r,t) = \frac{1}{2}(s_{CP}(h,r,t) + s_{CP}(h,r^{-1},t))$ $s_{CP} = \sum_{i=0}^{d-1} [v_r]_i \cdot [v_h]_i \cdot [v_t]_i$ | $\mathcal{L}_{nll} + \Delta_{L_2}$ | $v_h, v_t \in \mathbb{R}^d, v_r \in \mathbb{R}^d$ |
| DrWT [45] | Fine-grained types inference, domain knowledge modeling, leverages additional data outside KG, 4*th*-order TF | $s(E,F,G,H) = \chi$ $= C_{diag} \times_s E \times_p F \times_o G \times_d H$ | $\mathcal{L}_2$ | $chi \in \mathbb{R}^{S \times O \times P \times D}$, $\mathcal{C}_{diag} \in \mathbb{R}^{d \times d \times d \times d}$, $E \in \mathbb{R}^{S \times d}, F \in \mathbb{R}^{O \times d}$ $G \in \mathbb{R}^{P \times d}, H \in \mathbb{R}^{D \times d}$ |
| TriVec [46] | ComplEx with three components score function, three parts entity/relation -representation | $s(h,r,t) = \sum_{i=0}^{d-1}([v_h^1]_i[v_r^1]_i[v_t^3]_i$ $+[v_h^2]_i[v_r^2]_i[v_t^2]_i + [v_h^3]_i[v_r^3]_i[v_t^1]_i)$ | $\mathcal{L}_{ls} + \Delta_{N_3}$ | $v_h, v_t \in \mathbb{C}^d, v_r \in \mathbb{C}^d$ |
| Additional training technologies | | | | |
| Ensemble DistMult [47] | Reproduces DistMult, parameter adjustment, fine tuning technology | $s(h,r,t) = v_h^T \cdot M_{r\,diag} \cdot v_t$ $s'(h,r,t) = P(t\|h,t) = \frac{exp(s(h,r,t))}{\sum_{\bar{t} \in \epsilon_{h,t}} exp(s(h,r,\bar{t}))}$ | $max\ \mathcal{L}_{marg}$ | $M_{r\,diag} = diag(r), r \in \mathbb{R}^d$ |
| Regularizer -Enhanced Model [48] | R1 multiplicative- L1 regularizer | $s(h,r,t) = Re(v_h^T M_{r\,diag} \bar{v}_t)$ $= Re(\sum_{i=0}^{d-1} [v_r]_i \cdot [v_h]_i \cdot [\bar{v}_t]_i)$ | $\mathcal{L}_{nll}$ $+\Delta_{R_1 mL_1}$ | $M_{r\,diag} = diag(v_r), v_r \in \mathbb{C}^d$, $R_1(\Theta) =$ $\sum_{r \in R} \sum_{i=0}^{d-1} \lvert Re([v_r]_i) \cdot Im([v_r]_i) \rvert$, $R_2(\Theta) = \lVert \Theta \rVert_2^2$ |
| Constraints -enhanced Model [49] | NNE constraints, AER constraints[b] | $s(h,r,t) = Re(v_h^T M_{r\,diag} \bar{v}_t)$ $= Re(\sum_{i=0}^{d-1} [v_r]_i \cdot [v_h]_i \cdot [\bar{v}_t]_i)$ | $\mathcal{L}_{nll} + \Delta_{L_2}$ | $M_{r\,diag} = diag(r), r \in \mathbb{C}^d$; $0 \le Re(e), Im(e) \le 1$; $s(e_i, r_p, e_j) \le s(e_i, r_q, e_j)$, $\forall e, e_i, e_j \in E$ |

(*continued on next page*)

### 3.1.1. *Tensor/matrix factorization models*

Here we introduce a series of Tensor Factorization (TF) models in detail and make a summary table (Table 4) for conveniently exhibiting the characteristics of these models. Recently, tensors and their decompositions are widely used in data mining and machine learning problems [13]. In KG field, the large-scale tensor factorization has been paid more and more attention for KGC tasks.

Based on a fact that KG can be represented as tensors (shown in Fig. 4), KGC can be framed as a 3*rd*-order binary tensor completion problem, tensors can also be regarded as a general method to replace common methods, such as graphical models [50]. For KGC, the relational data can be represented as a $\{0,1\}$-valued third-order tensor $Y \in \{0,1\}^{|E| \times |R| \times |E|}$, if the relation $(h,r,t)$ is true there meets $Y_{h,r,t} = 1$, and the corresponding three modes properly stand for the subject mode, the predicate mode and the object mode respectively. TF algorithms aim to infer a predicted tensor $X \in \mathbb{R}^{|E| \times |R| \times |E|}$ that approximates $Y$ in a sense. Validation/test queries $(?,r,t)$ are generally answered by ordering candidate entities $h'$ through decreasing values of $X_{h',r,t}$, yet queries $(h,r,?)$ are answered by ordering entities $t'$ with decreasing values of $X_{h,r,t'}$. In that context, numerous literature have considered link prediction as a low-rank tensor decomposition problem.

**Table 4** (*continued*).

| Model | Highlight | Score function | Loss function[a] | Parameters & Constrains |
|---|---|---|---|---|
| N3 regularizer [50] | CP + p-norms regularizer | $s(h,r,t) = s_{CP} = \sum_{i=0}^{d-1}[v_r]_i \cdot [v_h]_i \cdot [v_t]_i$ | $\mathcal{L}_{nll} + \Delta_{N_3}$ | $v_h, v_t \in \mathbb{R}^d$, $v_r \in \mathbb{R}^d$<br>$\Omega_p^{\alpha}(v) = \frac{1}{3}\sum_{r=1}^{R}\sum_{d=1}^{3}\|v_r^{(d)}\|_p^{\alpha}$ |
| B-CP [51] | CP + binary value parameters, Bitwise Operations | $s(h,r,t) = \chi = \sum_{i\in[d]} v_{hi}^{(b)} \otimes v_{ti}^{(b)} \otimes v_{ri}^{(b)}$<br>$v_{hi}^{(b)} = Q_{\triangle}(v_{hi})$, $v_{ti}^{(b)} = Q_{\triangle}(v_{ti})$,<br>$v_{ri}^{(b)} = Q_{\triangle}(v_{ri})$<br>$Q_{\triangle}(x) = \triangle sign(x) = \begin{cases} +\triangle & if\ x \geq 0, \\ -\triangle & if\ x < 0 \end{cases}$ | $\mathcal{L}_{CE}$ | $chi \in \{0,1\}^{N_e \times N_e \times N_r}$<br>$v_{hd}, v_{td} \in \{+\triangle, -\triangle\}^d$<br>$v_{rd} \in \{+\triangle, -\triangle\}^d$ |
| QuatE [52] | ComplEx in hyper-complex space | $s(h,r,t) = Q_h^{rotation} \cdot Q_t$,<br>$Q_x = \{a_x + b_x\mathbf{i} + c_x\mathbf{j} + d_x\mathbf{k}\}$,<br>$W_r = \{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}\}$,<br>$Q_h^{rotation} = Q_h \otimes W_r^{\triangleleft}$,<br>$W_r^{\triangleleft} = \frac{W_r}{\lvert W_r \rvert}$ | $\mathcal{L}_{nll} + \Delta_{L_2}$ | $Q \in \mathbb{H}^{N_e \times d}$, $W \in \mathbb{H}^{N_r \times d}$;<br>$x = h, t$;<br>$a_h, b_h, c_h, d_h \in \mathbb{R}^d$;<br>$a_t, b_t, c_t, d_t \in \mathbb{R}^d$;<br>$a_r, b_r, c_r, d_r \in \mathbb{R}^d$ |
| JoBi [53] | Joint learning: bilinear TF model + auxiliary model (using entity -relation co-occurrence pairs) | JoBi ComplEx:<br>$s_{bi}(h,r,t) = Re(v_h^T diag(r_{bi})\bar{v}_t)$,<br>$s_{tri}(h,r,t) = Re(v_h^T diag(r_{tri})\bar{v}_t)$ | $\mathcal{L}_{nll}$ | $v_h, v_t, v_r \in \mathbb{R}^d$ |
| Linear & Quadratic Model [54] | 'Linear + Regularized', 'Quadratic + Regularized', 'Quadratic + Constraint' 'Linear + Constraint' | $s(h,r,t) = v_h^T M_r v_t$ | $\mathcal{L}_{quad} + C/R^c$ | $v_h, v_t \in \mathbb{R}^d$, $M_r \in \mathbb{R}^{d\times d}$ |

[a] $\mathcal{L}_{ll}$ ($\mathcal{L}_{nll}$), $\mathcal{L}_{ls}$, $\mathcal{L}_2$, $\mathcal{L}_{quad}$, $\mathcal{L}_{marg}$ and $\mathcal{L}_{CE}$ are *(negative) log likely-hood loss, log softmax loss,* $L_2$ loss, quadratic loss, *margin-based ranking loss* and *cross entropy loss* respectively, and $\Delta$ indicates the regularization terms in loss function.

[b] 'NNE' and 'AER' represents non-negativity constraints and approximate entailment constraints.

[c] 'C/R' means Constraints and Regularations in [54].

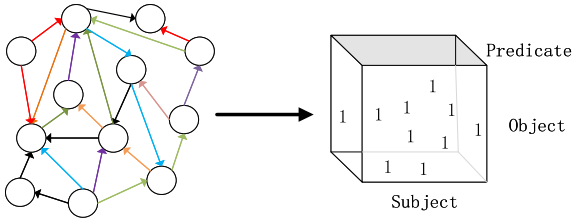



**Fig. 4.** Knowledge Graph as Tensors [41].

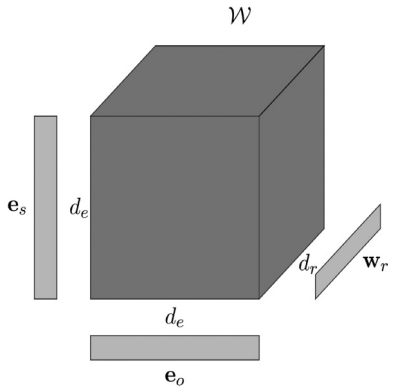



**Fig. 5.** Visualization of the TuckER architecture [37].

*3.1.1.1. Tucker-based TF methods.* The well-known TF approach **Tucker** [55] decomposes the original tensor $\chi \in \mathbb{R}^{N_1\times N_2\times N_3}$ into three matrices $A \in \mathbb{R}^{N_1\times M_1}$, $B \in \mathbb{R}^{N_2\times M_2}$, $C \in \mathbb{R}^{N_3\times M_3}$ and a smaller core tensor $\mathcal{Z} \in \mathbb{R}^{M_1\times M_2\times M_3}$, specifically in the form of

$$\chi \approx \mathcal{Z} \times_1 A \times_2 B \times_3 C,$$

where $\times_n$ denotes the tensor product along the $n$th mode. Factor matrices $A$, $B$ and $C$ can be considered as the principal components in each mode if they are orthogonal. Typically, since $M_1$, $M_2$, $M_3$ are smaller than $N_1$, $N_2$, $N_3$ respectively, thus $\mathcal{Z}$ can be regarded as a compressed version of $\chi$, whose elements express the interaction level between various components.

**TuckER** [37] based on Tucker decomposition to the binary tensor representation, it is a powerful linear model with fewer parameters but obtains consistent good results, this is because it enables multi-task learning across relations. By modeling the binary tensor representation of a KG according to Tucker decomposition as Fig. 5, TuckER defines the score function as:

$$s(h,r,t) = \mathcal{W} \times_1 v_h \times_2 v_r \times_3 v_t$$

where $\mathcal{W} \in \mathbb{R}^{d_e\times d_r\times d_e}$ indicates the core tensor, $v_h, v_t \in \mathbb{R}^{d_e}$ and $v_r \in \mathbb{R}^{d_r}$ represent the head entity embedding, tail entity embedding and relation embedding respectively. It is worth noting that TuckER can derive sufficient bounds on its embedding dimensionality, and adequate evidence can prove that several linear models (such as RESCAL [13] and DistMult [42] that will be mentioned later) can be viewed as special cases of TuckER.

However, Kolda and Bader [56] indicated that Tucker decomposition is not unique because the core tenser $\mathcal{Z}$ can be transformed without affecting the fit if we conduct the inverse transformation to $A$, $B$ and $C$.

*3.1.1.2. Decomposition into directional components (DEDICOM)-based TF methods.* Contrary to Tucker decomposition, the rank-$r$ **DEDICOM** decomposition [57] is capable of detecting correlations between multiple interconnected nodes, which can be captured through singly or synthetically considering the attributes, relations, and classes of related entities during a learning process. DEDICOM decomposes a three-way tensor $\chi$ as:

$$\chi_k \approx AD_kRD_kA^T, \ for\ k = 1, \ldots, m$$

where the matrix $A \in \mathbb{R}^{n\times r}$ indicates the latent components, the asymmetric matrix $R \in \mathbb{R}^{r\times r}$ reflects the global interactions between the latent components, whereas the diagonal matrix $D_k \in \mathbb{R}^{r\times r}$ models the participation situation of the latent components in the $k$th predicate. Under this circumstance, DEDICOM is suitable for the case where there exists a global interaction model

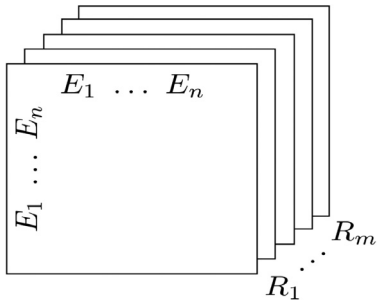



**Fig. 6.** The there-way tensor model for relation data [13].

for the latent components and its variation in the third mode can be described by diagonal factors [13].

**RESCAL** [13] is an early three-way DEDICOM-based model for KGC, which interprets the inherent structure of dyadic relational data. By employing a three-way tensor $\chi$ (as shown in Fig. 6), where two modes are identically formed by the concatenated entity vectors of the domain and the third mode holds the relations matrix in the domain. The score of a fact $(h, r, t)$ is defined by a bilinear function:

$$s(h, r, t) = v_h^T M_r v_t, \tag{1}$$

where $v_h, v_t \in \mathbb{R}^d$ are entity embeddings, and $M_r \in \mathbb{R}^{d\times d}$ is an asymmetric matrix associated with the relation that models the interactions between latent factors.

**LFM** [38] is a bilinear TF model extending RESCAL, to overcome the relational data growing issue thus to model large multi-relational datasets. Similar to RESCAL, LFM embeds entities in $d-$dimension vectors, encodes each relation into a matrix $M_{rj}$ as a bilinear operators among the entities, where $1 \le j \le N_r$, $M_r \in \mathbb{R}^{d\times d}$. For efficiently modeling large relational factor, $(h, r, t)$, LFM first redefines the previous linear score items as the following form to take account of the different interaction order including unigram, bigram, and trigram orders between $h, t$ and $r$:

$$s(h, r, t) \triangleq y^T M_r y' + {v_h}^T M_r z + {z'}^T M_r v_t + {v_h}^T M_r v_t \tag{2}$$

where the parameters $y, y', z, z' \in \mathbb{R}^d$, which participate in the calculation $yM_r y'$, ${v_h}^T M_r z + {z'}^T M_r v_t$ terms, together with ${v_h}^T M_r v_t$, these three terms represents uni-, bi- and trigram orders of interactions between $h, t$ and $r$. The another improvement on RESCAL is decomposing the relation matrix $M_r$ over a set of $p$-rank matrices $\Theta_r (1 \le r \le p)$ with:

$$M_r = \sum_{r=1}^{p} \alpha_r^j \Theta_r \tag{3}$$

where $\Theta_r = u_r w_r^T$ for $u_r, w_r \in \mathbb{R}^d, \alpha^j \in \mathbb{R}^p$. The $\Theta_r$ constrained by the outer product operator efficiently decreases the number of the overall parameters compared with the general relation matrix parameterization process in RESCAL, which greatly speeds up the computations relying on traditional linear algebra. LFM normalizes the terms appearing in formulas (2) and (3) by minimizing the negative log-likelihood over a specific constraint set.

**Tatec** [39] cooperates with both 2-way and 3-way interactions models to capture different data patterns in respective embedding space, which obtains a better performance outstripping the best of either constituent. Different from the closest relative model LFM, Tatec combines the 3-way model and constrained 2-way model but pre-trains them separately. Tatec learns distinct embeddings and relation parameters for the 2-way and the 3-way interaction terms so that it avoids the problem of reducing the expressiveness of the 2-way interaction terms caused by joint parameterization. The combinatorial score function of Tatec as:

$$s(h, r, t) = s_1(h, r, t) + s_2(h, r, t)$$

where $s_1()$ and $s_2()$ correspond to the 2-way and 3-way term as the following forms:

$$s_1(h, r, t) = v_{r_1}^T v_{h_1} + v_{r_2}^T v_{t_1} + v_{h_1}^T D_{diag} v_{t_1}$$

$$s_2(h, r, t) = v_{h_2}^T M^r v_{t_2}$$

where $v_{h_i}, v_{t_i}$ are embeddings of head and tail entities in $\mathbb{R}^{d_i}$ space $(i = 1, 2)$, $v_{r_1}, v_{r_2}$ are vectors in $\mathbb{R}^{d_1}$, while $M^r \in \mathbb{R}^{d_2\times d_2}$ is a mapping matrix, and $D$ is a diagonal matrix that is independent of the input triple. Depending on whether jointly update (or fine-tune) the parameters of 2-way and 3-way score terms in a second phase, Tatec proposes two term combination strategies to effectively combine the bigram and trigram scores, fine tuning (*Tatec-ft*) and linear combination (*Tatec-lc*), the former simply adding $s_1$ term and $s_2$ term and fine-tuned overall parameters in $s$, while the latter combines twos in a linear way. Besides, Tatec attempts hard regularization or soft regularization for the *Tatec-ft* optimization problem.

**ANALOGY** [40] is an extended version of RESCAL, it is interested in explicitly modeling analogical properties of both entity and relation embeddings, applies a bilinear score function used in RESCAL (shown in formula (1)) but further stipulates the relation mapping matrices must to be normal as well as mutually commutative as:

$$normality: M_r M_r^T = M_r^T M_r, \ \forall r \in \mathbb{R}$$

$$commutativity: M_r M_{r'} = M_{r'} M_r, \ \forall r \in \mathbb{R}$$

The relation matrices can be simultaneously block-diagonalized into a set of sparse almost-diagonal matrices, each decomposed matrix equips $O(d)$ free parameters. Besides, ANALOGY carries out the training process by formulating a differentiable learning objective, thus allows it to exhibit a favorable theoretical power and computational scalability. Relevant evidence has shown that multiple TF methods, such as DistMult [42], HolE [58], and ComplEx [43] that will be mentioned later can be regarded as special cases of ANALOGY in a principled manner.

**REST** [41] has fast response speed and good adaptability to evolve data and yet obtains comparable or better performance than other previous TF approaches. Based on the TF model, REST uses Random Walk (RW)-based semantic graph sampling algorithm (SGS) and predicate sparsification operator to construct Ensemble Components, which samples a large KG tensor in its graph representation to build diverse and smaller subgraph tensors (the Ensemble Architecture as Fig. 7), then uses them in conjunction for focused link prediction (FLP) task. Experimental results show that FLP and SGS are helpful to reduce the search space and noise. In addition, the predicate sparsification can improve the prediction accuracy. REST can deliver results on demand, which makes it more suitable for the dynamic and evolutionary KGC field.

*3.1.1.3. CANDECOM/PARAFAC (CP)-based TF methods.* The most well known canonical tensor decomposition method relevant to KGC field might be the **CANDECOM/PARAFAC (CP)** [59], in which a tensor $\chi \in \mathbb{R}^{N_1\times N_2\times N_3}$ was represented as a sum of $R$ rank one tensors $x_r^{(1)} \bigotimes x_r^{(2)} \bigotimes x_r^{(3)}$, thus:

$$\chi = \sum_{r=1}^{R} x_r^{(1)} \cdot x_r^{(2)} \cdot x_r^{(3)}$$

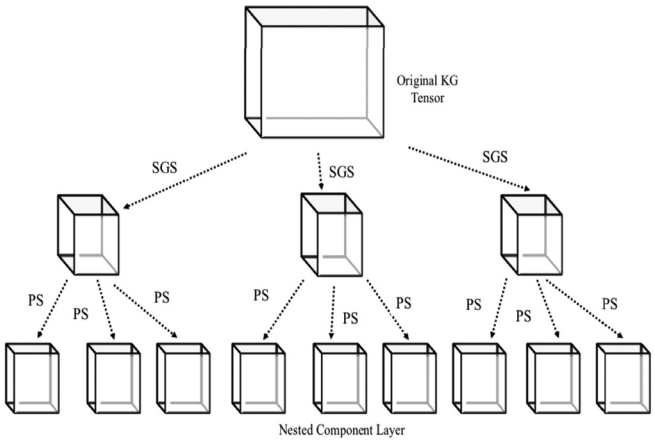



**Fig. 7.** Ensemble Architecture in REST [41].

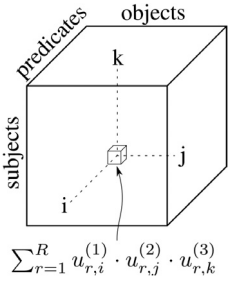



**Fig. 8.** The tensor operated via CP and the score of a triple $(h, r, t)$ [50].

where $\cdot$ denotes the tensor product, $r \in \{1, \ldots, R\}$, and $x_r^{(i)} \in \mathbb{R}^{N_i}$. Fig. 8 shows the representation of CP decomposition and the scoring of a given triplet. In particular, the smallest $r$ contained in the decomposition of given $\chi$ is called the canonical rank of $\chi$. Although the current implementations of CP on standard KGC benchmarks are known to perform poorly compared to more specialized methods, CP owns a surprising expressive ability, thus a series of works attempt to understand the limits of CP and further extend it for KGC.

**DistMult** [42] replaces the dense matrix in RESACL [13] with diagonal matrices to significantly reduce parameters of RESACL, the score function is defined as follows:

$$s(h, r, t) = v_h^T M_{r\,diag} v_t \tag{4}$$

where $M_{r\,diag} = diag(v_r)$. However, DistMult represents embedding vectors with real values may make it model symmetric representation of relations only due to the symmetric nature of the product operator on real numbers.

Moreover, **Holographic Embeddings (HolE)** [58] is not a CP-based method, but it can model asymmetric relations as well as RESCAL, and also achieves the same simplicity as DisMult by only perform an efficient circular correlation. With the commutative circular correlation of tensors, HolE can generate compositional fixed-width representations, i.e., it allows $\mathbb{R}^d \times \mathbb{R}^d = \mathbb{R}^d$, which significantly reduces the parameter number but remains high scalability, economical computing capability, and easy training.

**ComplEx** [43] first uses complex values embeddings in a complex space to handle both symmetric and antisymmetric binary relations, where each embedding is represented using two vectors (real and imaginary numbers). In addition to that, ComplEx also represents those tail entities as the complex conjugate of them, so that it can encode both symmetric and asymmetric relations. What is more, shown as formula (5), ComplEx's bilinear energy function consists of various interaction parts unlike DisMult with only one bilinear product component.

$$s(h, r, t) = Re(v_h^T M_{r\,diag} \bar{v}_t) = Re(\sum_{i=0}^{d-1} [v_r]_i \cdot [v_h]_i \cdot [\bar{v}_t]_i) \tag{5}$$

where $v_h, v_t, v_r \in \mathbb{C}^d$, $Re()$ indicates an operate to obtain the real part of a complex value, $[v_x]_i$ represents the $i$th element of $v_x$ and $\bar{v}_t$ means the conjugate of $v_t$. The spacial style of every components with combination of real numbers and imaginary numbers interprets the ability of ComplEx to model antisymmetry relations in KGs. Noting that HolE is shown to be a special case for ComplEx whose conjugate symmetry is imposed on embeddings.

**SimplE** [44]: Inspired by Canonical Polyadic (CP) [59], SimplE improves it by utilizing the inverse of relations to handle the poor performance problem in CP caused by the independence of the entity vectors. SimplE considers two vectors $v_r, v_r^{-1}$ for each relation $r$, the similarity score function of SimplE for a triple $(e_i, r, e_j)$, $e_i, e_j \in E$ is defined as the average of the CP scores for $(e_i, r, e_j)$ and $(e_j, r^{-1}, e_i)$, this setup allows the two embedding of each entity to be learned dependently and makes the SimplE be considered as a bilinear model, scores each triplet as:

$$s(h, r, t) = 1/2(s_{CP}(h, r, t) + s_{CP}(h, r^{-1}, t))$$

$$s_{CP} = \sum_{i=1}^{d} [v_h]_i \cdot [v_r]_i \cdot [v_h]_i$$

SimplE also use a log-likelihood loss to avoid over-fitting. SimplE model is not only fully expressive, it performs very well empirically despite (or maybe because of) its simplicity.

**DrWT** [45] aims at fine-grained types inference in KGs, it explicitly models domain knowledge and leverages additional data outside KG, the anchor linked Wikipedia page document of entities, and the extra relations mapped from additional data sources. DrWT uses CP based 4*th*-order Tensor Factorization which factorizes each 4*th*-order domain-relevance weighted tensor $\chi \in \mathbb{R}^{S \times O \times P \times D}$ as:

$$s(E, F, G, H) = \chi = \mathcal{C}_{diag} \times_S E \times_P F \times_O G \times_D H$$

where the diagonal core tensor $\mathcal{C} \in \mathbb{R}^{d \times d \times d \times d}$ and the feature matrices $E \in \mathbb{R}^{S \times d}$, $F \in \mathbb{R}^{O \times d}$, $G \in \mathbb{R}^{P \times d}$ and $H \in \mathbb{R}^{D \times d}$ are the model parameters that have to be learned, and the scoring $s(E, F, G, H)$ is the tensor product to multiply a matrix on dimension $x$ with a tensor. DrWT is an attempt for explicitly leveraging domain knowledge in KG, and for utilizing the additional large amount of interactions among multiple entities and text descriptions. On the other hand, it further discusses probabilistic inference based on collective multilevel type classification and latent similarity of typed entities.

**TriVec** [46] is an efficient novel TF-based KG embedding model for stand benchmark datasets and/or more challenging datasets for practical application scenarios. TriVec improves the ComplEx by replacing the four-parts embedding score function of ComplEx with three components style and representing each entity and relation utilizing three parts, which enables TriVec to deal with both symmetric and asymmetric relations. Moreover, TriVec adapts a kind of combined form loss function for training, where applies the traditional ranking loss with the squared error and the logistic loss, and the multi-class configuration with negative-log *softmax* loss simultaneously. TriVec also prepares a new benchmark dataset, NELL239, and produces a real biological application dataset based on the Comparative Toxicogenomics Database (CTD) database especially, which aims at assessing the practical significance of TriVec.

*3.1.1.4. Additional training technologies.* The scalable and efficient performance of these bilinear models have encouraged lots of studies to investigate boosting the DistMult and the ComplEx models by exploiting different training objectives and regularization constraints [47,50].

**Ensemble DistMult** [47] reproduces DistMult [42] through simple parameter adjustment and fine-tuning technology, and gets better scores than most previous KGC methods. Ensemble DistMult employs a *softmax* function normalizing, imposes it on the original score function in DistMult, which turn the formula (4) into:

$$s(h,r,t) = v_h^T \cdot M_{r\,diag} \cdot v_t$$

$$s'(h,r,t) = P(t|h,t) = \frac{exp(s(h,r,t))}{\sum_{\bar{t}\in\epsilon_{h,t}} exp(s(h,r,\bar{t}))}$$

where $\epsilon_{h,t}$ is the candidate answer entities set for the $(h,r,?)$ query.

Ensemble DistMult concludes that increasing the number of negative instances can have a positive impact on the results, and the batch size also has an impact that a larger iteration batch size can promote the model effect. It is highlighted that the question in doubt that whether a model is achieved through better algorithms in theoretical or merely through more extensive parametric search. By the way, since the *filtered* scenario assumes that there is only one correct answer among the candidates in the KG, which is unrealistic, Ensemble DistMult puts forward a proposal that it is necessary to pay more attention to the original *raw scenario* rather than the *filtered* setting, however, this requires the use of other information retrieval metrics, such as Mean Average Precision (MAP).

**Regularizer-Enhanced Model** [48] also aims to improve ComplEx by designing a novel L1 regularizer called R1 multiplicative L1 regularizer, which can support modeling both symmetric and antisymmetric relations. The regularizer R1 in a form of an L1-norm penalty to allow the sparsity of pairwise products. More specifically, this L1 penalty term is expected to help guide learning a vector for relation $r$ in accordance with whether $r$ is symmetric, antisymmetric, or neither of them, as observed in the training data due to the real and imaginary parts of a relation vector govern the symmetry/antisymmetry of the scoring function for the relation. Since parameters are coupled componentwise, the proposed model can also deal with non-symmetric, non-antisymmetric relations which have varying degrees of symmetry/antisymmetry. Setting the vector component items in vector $\Theta$ represents the overall parameters of the model, the regularizer terms as follows:

$$R_1(\Theta) = \sum_{r\in R}\sum_{i=0}^{d-1} |Re([v_r]_i)\cdot Im([v_r]_i)|,\ \ v_r \in \mathbb{C}^d$$

$$R_2(\Theta) = \|\Theta\|^2$$

Although the non-convex R1 term makes the optimization harder, experiments reports that multiplicative L1 regularization not only outperforms the previous standard one in KGC, but also is robust enough against random initialization.

**Constraints-enhanced Model** [49] imposes simple constraints on KGC, introduces *non-negativity constraints (NNE)* on entity representations to form compact and interpretable representations for entities, and *approximate entailment constraints (AER)* on relation representations for further encoding regularities of logical entailment between relations into their distributed representations, these two constraints are:

$$NNE:\ 0 \le Re(v_e), Im(v_e) \le 1, \forall e \in E, v_e \in \mathbb{C}^d \tag{6}$$

$$AER:\ s(e_i,r_p,e_j) \le s(e_i,r_q,e_j), \forall e_i,e_j \in E \tag{7}$$

In the formula (6), *non-negativity constraints* are imposed on both the real part and the imaginary part of the entity vector, which states that only positive properties will be stored in entity representations. Note that 0 and 1 are all-zeros values and all-ones values of d-dimensional vectors, and $\ge, \le, =$ denote the entry-wise comparisons. In the formula (7), it formally describes that when there has a strict entailment $r_p \to r_q$, then the triple score must meet one request that if $(e_i, r_p, e_j)$ is a true fact with a high score $s(e_i, r_p, e_j)$, then the triple $(e_i, r_q, e_j)$ with an even higher score should also be predicted as a true fact.

As Lee and Seung [60] pointed out, non-negativity, in most cases, will further induce sparsity and interpretability. Except for improving the KG embedding, the proposed simple constraints impose prior beliefs upon the embedding space structure but do not significantly increase the space or time complexity.

**Weighted Nuclear 3-Norm Regularizer Model (N3)** [50] also improves basic CP model by testing a novel tensor nuclear p-norms based regularizer, it first indicated that the regularizer based on the square Frobenius norms of the factors [42,43] mostly used in the past is not a tensor norm since it is un-weighted. Then this paper introduces a variational form of the nuclear $3-norm$ to replace the usual regularization at no additional computational cost with the form of:

$$\Omega_p^\alpha(v) = 1/3 \sum_{r=1}^{R}\sum_{d=1}^{3} \|v_r^{(d)}\|^\alpha$$

where $p = 3$ when it is a nuclear 3-norm, and $v_r^{(d)}, d = 1,2,3$ means the tensor of subject mode, the predicate mode and the object mode respectively. Finally, Lacroix et al. [50] discuss a weighting scheme analogous to the weighted trace-norm proposed in Srebro and Salakhutdinov [61] as:

$$Weighted(\Omega_p^\alpha(v)) = \frac{1}{3}\sum_{r=1}^{R}\sum_{d=1}^{3} \|\sqrt[p]{q^{(d)}} \odot v_r^{(d)}\|^\alpha$$

where $\sqrt{q^{(d)}}$ represents the weighting implied by this regularization scheme. Surprisingly, under the using of the nuclear p-norms [62] and the *Reciprocal* setting, the tensor regularizer recreates a much successful result of CP decomposition (even better than advanced ComplEx), and this reflects a phenomenon that although the effect of optimization parameters is well known, neither the effect of the formula nor the effect of regularization has been properly evaluated or utilized. This work suggests one possibility: when each model is evaluated under appropriate optimal configuration, its performance may make great progress, this observation is very important to assess and determine the direction for further TF study for KGC.

**Binarized Canonical Polyadic Decomposition (B-CP)** [51] extends the CP model by replacing the original real-valued parameters with binary values. Only conducts the bitwise operation for score computation, B-CP has been proven a successful technique obtains more than one order of magnitude while maintaining the same task performance as the real-valued CP model. Specifically, setting $D$ is the number of rank-one tensors, and $\triangle$ is a positive constant value, B-CP binarizes the original factor matrices $A, B \in \mathbb{R}^{N_e\times d}$ and $C \in \mathbb{R}^{N_r\times d}$ in CP with: $A^{(b)}, B^{(b)} \in \{+\triangle, -\triangle\}^{N_e\times d}$, $C \in \{+\triangle, -\triangle\}^{N_r\times d}$, thus the original boolean tensor $\chi_{hrt} \in \{0,1\}^{N_e\times N_e\times N_r}$ is turned into:

$$s(h,r,t) = \chi_{hrt} = \sum_{i\in[d]} v_{hi}^{(b)} \cdot v_{ti}^{(b)} \cdot v_{ri}^{(b)}$$

where $v_{hi}^{(b)} = Q_\triangle(v_{hi})$, $v_{ti}^{(b)} = Q_\triangle(v_{ti})$, $v_{ri}^{(b)} = Q_\triangle(v_{ri})$ are binarized through:

$$Q_\triangle(x) = \triangle sign(x) = \begin{cases} +\triangle \ if \ x \geq 0, \\ -\triangle \ if \ x < 0, \end{cases}$$

Here the binarization function can be further extended to vectors: $Q_\triangle(x)$ means a vector with $i$th element is $Q_\triangle(x_i)$.

By deriving a bound on the size of its embeddings, B-CP is proved to be fully expressive.

**QuatE** [52] is an extension to ComplEx on hyper-complex space. QuatE creatively introduced hyper-complex representations to learn KG embeddings more expressively, it uses quaternion embeddings, hyper-complex-valued embeddings with three imaginary components to model entities and considers the rotations in the quaternion space to represent relations. In QuatE, each entity embedding is represented by a quaternion matrix $Q \in \mathbb{H}^{N_e \times k}$, and the relation embeddings are denoted by $W \in \mathbb{H}^{N_r \times k}$, where $k$ is the dimension of the embedding. For a triplet $(h, r, t)$, denotes $Q_h = \{a_h + b_h\mathbf{i} + c_h\mathbf{j} + d_h\mathbf{k} : a_h, b_h, c_h, d_h \in \mathbb{R}^k\}$ and $Q_t = \{a_t + b_t\mathbf{i} + c_t\mathbf{j} + d_t\mathbf{k} : a_t, b_t, c_t, d_t \in \mathbb{R}^k\}$ as the head entity $h$ and tail entity $t$ respectively, while the relation $r$ is expressed in $W_r = \{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k} : a_r, b_r, c_r, d_r \in \mathbb{R}^k\}$ (in a quaternion $Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, the $a$, $b$, $c$, $d$ are real numbers and $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are imaginary units and are square roots of $-1$). Then the scoring function with the use of quaternion inner product:

$$s(h, r, t) = Q_h' \cdot Q_t$$

The $Q_h'$ means the head entity rotation conducted with the Hamilton product:

$$Q_h' = Q_h \otimes W_r^\triangleleft$$

and $W_r^\triangleleft = p + q\mathbf{i} + u\mathbf{j} + v\mathbf{k}$ is a unit quaternion by normalizing the relation quaternion $W_r$, which calculated by:

$$W_r^\triangleleft(p, q, u, v) = \frac{W_r}{|W_r|} = \frac{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}}$$

Compared to the complex Hermitian operator and the inner product in Euclidean space, the Hamilton operator provides a greater extent of expressiveness, it can aptly capture latent interdependencies (between all components), and support a more compact interaction between entities and relations. It is also worth mentioning that the rotation over four-dimensional space has more degree of freedom than complex plane rotation. Since QuatE is a generalization of ComplEx on hyper-complex space but offers better geometrical interpretations, it also satisfies the key request of symmetry, anti-symmetry, and inversion relations learning. Noting that when the coefficients of the imaginary units $j$ and $k$ are all set to zero, the obtained complex embeddings will be the same as in ComplEx yet the Hamilton product will also degrade to complex number multiplication, while even obtains the DistMult case when it further removes the normalization of the relational quaternion.

**JoBi** [53] designs an auxiliary model using entity-relation co-occurrence pairs for joint learning with the base model (can be any bilinear KGE models). The occurrences of entity-relation pairs would overcome data sparsity well, and also bias the model to score plausible triples higher. JoBi creatively contains two copies of a bilinear model, the base triple model is trained about the triple's labels, while the pair model is trained on occurrences of entity-relation pairs within the triples. For the triple $(h, r, t)$, the scoring functions $s_{bi}$ and $s_{tri}$ for the pair and triple models respectively are shown as:

$$s_{bi}(h, r, t) = Re(v_h^T diag(v_{r_{bi}})\bar{v}_t)$$

$$s_{tri}(h, r, t) = Re(v_h^T diag(v_{r_{tri}})\bar{v}_t)$$

where $\bar{v}_t$ denotes the complex conjugate of $v_t$, and $Re(x)$ denotes the real part of the complex vector $x$. The two modules are jointly optimized during training, but during test time it merely uses $s_{tri}$ models, that is the reason why the additional auxiliary module does not affect the number of final parameters of the trained model. JoBi also utilizes entity-relation pair occurrences to improve the distribution of negative examples for contrastive training, which allows the model to learn higher quality embeddings with much fewer negative samples. Finally, a negative log-likelihood loss of softmax and a binary cross-entropy loss are used for $s_{tri}$ and $s_{bi}$ respectively, further the two losses are combined via a simple weighted addition with a tunable hyper-parameter $\alpha$:

$$\mathcal{L} = \mathcal{L}_{tri} + \alpha\mathcal{L}_{bi}$$

**Linear & Quadratic Model** [54] presents a group of novel methods for embedding KGs into real-valued tensors, including four modules, 'Linear + Regularized', 'Quadratic + Regularized', 'Quadratic + Constraint' and 'Linear + Constraint', where two of the models optimize a linear factorization objective and two for a quadratic optimization. All in all, it reconstructs each of the $k$ relation slices of the order-3 tensor $\chi$ as:

$$\chi_k \approx A_\alpha R_k A_\beta^T \tag{8}$$

where $A$ is the collection of p-dimensional entity embeddings, $R$ is the collection of relation embeddings. The matrices $A_\alpha$ and $A_\beta$ are elements contained in $A$ which meet $A_\alpha, A_\beta \in \mathbb{R}^{N_e \times d}$ with $d$ is dimension of both entity and relation embeddings. The whole augmented reconstruction minimized loss objection are formed as:

$$\begin{aligned} L = & min_{A,R} f(A, R) + g(A, R) + f_s(A, R, C) \\ & + f_\rho(A, R, C) + f_{Lag}(A, R, C) \end{aligned} \tag{9}$$

where $f(A, R)$ means the reconstruction loss reflecting each of the $k$ relational criteria in the formula (8), the $g(A, R)$ term represents the standard numerical regularization of the embeddings, $f_s(A, R, C)$ using similarity matrix $C$ proposed in this work to conduct knowledge-directed enrichment with extra knowledge. Additionally, the two terms $f_\rho(A, R, C)$+$f_{Lag}(A, R, C)$ in the formula (9) respectively reflects the added knowledge-directed enrichment items about new regulars and constraints. This work can easily use prior background knowledge provided by users or extracted automatically from existing KGs, providing more robust and provably convergent, linear TF methods for KG embedding.

*3.1.1.5. Performance analysis about TF models.* We integrate experimental results on WN18 and FB15K datasets from most of the over-mentioned models (as shown in Table 5). Fig. 9 shows the performance of TF models on WN18RR and FB15K-237 datasets for further analysis.

***a. Preliminary Performance Analysis***

From Table 5, we can see that the improved extension methods based on the original linear tensor decomposition models (such as Complex) have achieved high competitive *MRR*, *Hits*@10 and accuracy results, which can be summarized as follows:

(1) **Regularization analysis:** In WN18, Manabe et al. [48] and Lacroix et al. [50] try to use different regularization techniques to improve the traditional TF method, they both obtain satisfying performance. In [48], the proposed multiplicative L1 regularizer ('ComplEx w/ m L1') emerges powerful comparability and even exceeds the previous baselines. The method in [50] performs well because it applies the nuclear 3-norm regularizer ('ComplEx-N3-R'). Additionally, the work [50] resets multi-class log-loss and

**Table 5**
Statistic about experimental results of TF models on WN18 and FB15K. We use the bold and italic to mark the scores ranking first and second under the same metrics respectively.

| | WN18 | | | | | FB15K | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MR | MRR | Hits@1 | Hits@3 | Hits@10 | MR |
| RESCAL [13] | 0.89 | 0.842 | 0.904 | 0.928 | – | 0.354 | 0.235 | 0.409 | 0.587 | – |
| DistMult [42] | 0.83 | – | – | 0.942 | – | 0.35 | – | – | 0.577 | – |
| Single DistMult [47] | 0.797 | – | – | 0.946 | 655 | 0.798 | – | – | 0.893 | 42.2 |
| Ensemble DistMult [47] | 0.79 | – | – | 0.95 | 457 | *0.837* | – | – | *0.904* | 35.9 |
| ComplEx [43] | 0.941 | 0.936 | 0.945 | 0.947 | – | 0.692 | 0.599 | 0.759 | 0.84 | – |
| ComplEx w/std L1 [48] | 0.943 | 0.94 | 0.945 | 0.948 | – | 0.711 | 0.618 | 0.783 | 0.856 | – |
| ComplEx w/mul L1 [48] | 0.943 | 0.94 | 0.946 | 0.949 | – | 0.733 | 0.643 | 0.803 | 0.868 | – |
| ComplEx-NNE[c] [49] | 0.941 | 0.937 | 0.944 | 0.948 | – | 0.727 | 0.659 | 0.772 | 0.845 | – |
| ComplEx-NNE+AER[c] [49] | 0.943 | 0.94 | 0.945 | 0.948 | – | 0.803 | *0.761* | 0.831 | 0.874 | – |
| ANALOGY [48] | 0.942 | 0.939 | 0.944 | 0.947 | – | 0.725 | 0.646 | 0.785 | 0.854 | – |
| RESCAL + TransE [31] | 0.873 | – | – | 0.948 | 510 | 0.511 | – | – | 0.797 | 61 |
| RESCAL + HolE [31] | 0.94 | – | – | 0.944 | 743 | 0.575 | – | – | 0.791 | 165 |
| HolE + TransE [31] | 0.938 | – | – | 0.949 | 507 | 0.61 | – | – | 0.846 | 67 |
| RESCAL + HolE + TransE [31] | 0.94 | – | – | 0.95 | 507 | 0.628 | – | – | 0.851 | 52 |
| SimplE [44] | 0.942 | 0.939 | 0.944 | 0.947 | – | 0.727 | 0.66 | 0.773 | 0.838 | – |
| ComplEx-N3-S[a] [50] | *0.95* | – | – | *0.96* | – | 0.8 | – | – | 0.89 | – |
| CP [51] | 0.942 | 0.939 | 0.945 | 0.947 | – | 0.72 | 0.659 | 0.768 | 0.829 | – |
| CP-FRO-R[b] [50] | *0.95* | – | – | 0.95 | – | **0.86** | – | – | **0.91** | – |
| CP-N3-R[b] [50] | *0.95* | – | – | *0.96* | – | **0.86** | – | – | **0.91** | – |
| ComplEx-FRO-R[b] [50] | *0.95* | – | – | *0.96* | – | **0.86** | – | – | **0.91** | – |
| ComplEx-N3-R[b] [50] | *0.95* | – | – | *0.96* | – | **0.86** | – | – | **0.91** | – |
| B-DistMult [51] | 0.841 | 0.761 | 0.915 | 0.944 | – | 0.672 | 0.558 | 0.76 | 0.854 | – |
| B-CP [51] | 0.945 | 0.941 | 0.948 | 0.956 | – | 0.733 | 0.66 | 0.793 | 0.87 | – |
| QuatE [52] | 0.949 | 0.941 | *0.954* | *0.96* | 388 | 0.77 | 0.7 | 0.821 | 0.878 | 41 |
| QuatE-N3-R [52] | *0.95* | 0.944 | *0.954* | **0.962** | – | 0.833 | **0.8** | **0.859** | 0.9 | – |
| QuatE+TYPE[c] [52] | *0.95* | *0.945* | *0.954* | 0.959 | 162 | 0.782 | 0.711 | *0.835* | 0.9 | 17 |
| TuckER [37] | **0.953** | **0.949** | **0.955** | 0.958 | – | 0.795 | 0.741 | 0.833 | 0.892 | – |

[a] "S" means the *Standard learning*.
[b] "R" denotes the *Reciprocal learning*.
[c] "NNE", "AER" and "TYPE" denote the *non-negativity constraints*, *approximate entailment constraints* [49] and the *type constraints* [52], respectively.

selects a larger rank scope for a more extensive search about optimization/regularization parameters, which are also the reasons for the good performance of it. Approaches that apply nuclear 3−*norm* regularizer still show extraordinary talents in FB15K, but most of the improvements are statistically significant than those on WN18.

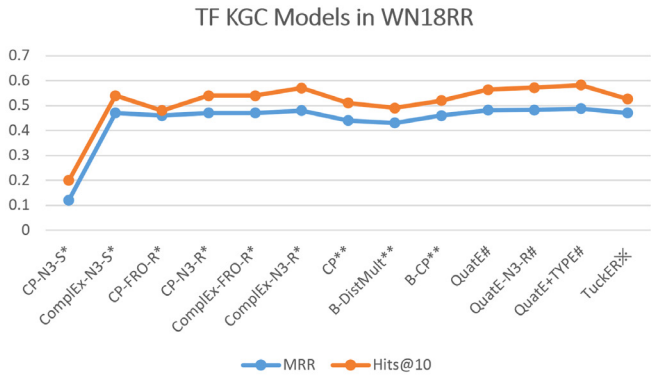
(2) **Constraints on entities and relations:** The results of 'ComplEx-NNE+AER' [49] demonstrate that imposing the non-negativity and approximate entailment constraints respectively for entities and relations indeed improves KG embedding. In Table 5, 'ComplEx-NNE' and 'ComplEx-NNE+AER' perform better than (or as equally well as) ComplEx in WN18. *We can find an interesting sight* that by introducing these simple constraints, 'ComplEx-NNE+AER' can beat strong baselines, including the best performing basic models like ANALOGY and those previous extensions of ComplEx, but can be derived such axioms directly from approximate entailments in [49]. Exerting proper constraints to the original linear TF models is also very helpful for KGC, just as in WN18, the constraints used 'ComplEx-NNE+AER' also out-performs ComplEx and other traditional TF models.

(3) **Different dimension space modeling:** In addition, the explorations of new tensor decomposition mode in different dimension space also achieve inspiring success. From Table 5 we can observe that on WN18, the quaternion-valued method QuatE performs competitively compared to the existing state-of-the-art models across all metrics and deservedly outperforms the representative complex-valued basic model ComplEx, which is because that quaternion rotation over the rotation in the complex plane has advantages in modeling complex relations. Besides, the *N3 regularization* and reciprocal learning in QuatE or the *type constraints* in QuatE also play an important role in QuatE's success. Another eye-catching method TuckER takes account of the binary tensor representation of KGs, which outperforms almost all linear TF models along with their relevant extension versions on all metrics in WN18. TuckER consistently obtains better results than those lightweight models ComplEx and SimplE that are famous for simplicity and fewer parameters equipment, which is because the TuckER allows knowledge sharing between relations through the core tensor so that it supports multi-task learning. In comparison, the same multi-task learning benefited 'ComplEx-N3' [50] forces parameter sharing between relations by ranking regularization of the embedding matrices to encourage a low-rank factorization, which uses the highly non-standard setting $de = dr = 2000$ to generate a large number of parameters compared with TuckER, resulting slightly lower grades than TuckER. Additionally, both QuatE and TuckER also achieve remarkable results on FB15K, especially QuatE on *Hist*@1, outperforms state-of-the-art models while the second-best results scatter amongst TuckER and 'ComplEx-NNE+AER'. Unlike the constraints-used methods that target applying prior beliefs to shrink the solution space, QuatE achieves high grades relying on effectively capturing the symmetry, antisymmetry, and inversion relation patterns, which take a large portion in both WN18 and FB15K. On FB15K, TuckER obtains lackluster performance across *MRR* and *Hits*@10 metrics but excesses on the toughest *Hits*@1 metric.

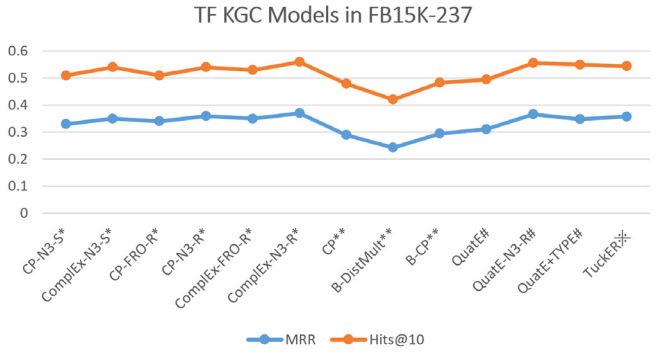
(4) **Considering hyper-parameters setting:** It is notable that on FB15K, the Ensemble DistMult also performs high results across both *MRR* and *Hits*@10, this is because it further improves DistMult only with proper hyper-parameters settings. This work helps us to solve the doubt: whether an algorithm was achieved due to a better model/algorithm or just by a more extensive hyper-parameter search. On the other hand, the good results of DistMult reported in Ensemble DistMult also because of using a large negative sampling size (i.e., 1000, 2000).

***b. Further Performance Verification:***

We have analyzed the effects of many factors on performance, especially the effectiveness of constraints or regularization techniques. To further evaluate the efficacy, we select the experimental results evaluated on WN18RR and FB15K-237 for illustration.




(a) TF Models on WN18RR.




(b) TF Models on FB15K-237.

**Fig. 9.** MRR, Hits@10 of TF methods on WN18RR and FB15K-237. "*", "**", "#" and "※" respectively indicate results from [37,50–52]. "S", "R" are *Standard learning* and *Reciprocal learning*, respectively.

We naturally plot experimental data on WN18RR and FB15K-237 datasets as Fig. 9, from which we can easily discover that both 'ComplEx-N3' and QuatE perform excellently in all metrics, the observation demonstrates the two models own great generality and scalability. Besides, the success of QuatE also enlightens us to explore the potential cooperation mode about useful techniques, such as *N3 regularization*, *reciprocal learning*, *non-negativity constraints (NNE)*, and *approximate entailment constraints (AER)*.

*3.1.1.6. Discussion about TF models.* Based on the above detailed introduction and a series of comparison and analysis on experimental results of these mentioned Tensor Factorization (TF) models, we further make some conclusive discussions:

**1. Regularization and Constraints.** Generally speaking, either imposing proper regularization or constraints on linear tensor factorization models is beneficial for KGC.

**2. High-dimensional Spaces Modeling.** Using rotation or other operations to model entities and relationships in high-dimensional spaces (such as QuatE and TuckER) with higher degrees of freedom may be a good attempt and a nice choice for further exploration on KGC.

**3. Multi-task Learning.** TuckER not only achieves better results than those of other linear models but also better than the results of many complex algorithms belonging to other categories, such as deep neural network models and reinforcement learning used architectures, e.g. ConvE [33] and MINERVA [63]. Still, since the good achievements on TuckER along with 'ComplEx-N3', we can deduce that although they are different in specific method details, they all enjoy the great benefit of multi-task learning.

We also conclude from TuckER that the simple linear models have valuable expressive power and are still worth to be served as a baseline before moving onto more elaborate models. Overall, we can see that the linear TF methods still have potential to be further improved by appropriate constraints, regularization, and parameter settings.

**4. Potential Threatens.** However, when exploring new improved methods, we should pay attention to the potential threatens. For example, *N3 normalization* will require larger embedded dimensions, and the number of Tucker parameters will increase linearly with the number of entities or relations in KGs, so that the scalability and economy of the algorithm should to be considered.

### 3.1.2. Neural network models

We will give a detailed introduction about *Neural Network* models on KGC study. A summary table for exhibiting the general features of introduced neural network KGC methods can be found in Table 6.

In recent years, distributed representations that map discrete language units into continuous vector space have gained significant popularity along with the development of neural networks [64,73–75]. However, human-like reasoning remains as an extremely challenging problem partially because it requires the effective encoding of world knowledge using powerful models [64]. At the same time, it has been found that neural networks can intelligently capture the semantic features of entities and relations and reasonably model the semantic relationships between discrete entities, which can help learn more accurate embeddings of KGs. Meanwhile, more and more complex and effective deep neural network structures have been developed so far, leading to a large amount of studies that apply these novel neural network frameworks to KGC field which obtained successful KGC results. We call this category of KGC approaches as *Neural Network-based KGC Models* in our summary, it also can be referred as the *non-linear models* in other literatures because the nonlinear function in neural network structures, e.g., *softmax* function, *sigmoid* activation function, etc.

*3.1.2.1. Traditional neural network-based KGC models.* **Neural Tensor Networks (NTN)** [14] The primitive NTN averages word vectors in entity name to generate the entity vector, so that entities with similar names can share the text information. NTN can explicitly reason relations between two entity vectors in KGs. In NTN, the standard linear neural network layer is replaced by a bilinear tensor layer, which is used to directly associate two entity vectors in multiple dimensions and calculate a score to represent the possibility of two entities $v_h, v_t$ having a certain relation $r$:

$$g(h, r, t) = u^T f(v_h^T W_r^{[1:k]} v_t + V_r [v_h, v_t]^T + b_r)$$

where $f = tanh()$ is a standard nonlinearity activation function, and $W_r^{[1:k]} \in \mathbb{R}^{d \times d \times k}$, $V_r \in \mathbb{R}^{k \times 2d}$, $b_r \in \mathbb{R}^k$ are parameter tensors, the bilinear tensor product $v_h^T W_r^{[1:k]} v_t$ results in a vector $v_e \in \mathbb{R}^k$.

**Multi Layer Perceptron (MLP)** [9] is a simplified version of NTN, it serves a multi-source Web-scale probabilistic knowledge base: Knowledge Vault built by [9], which is much bigger than other existing automatically constructed KGs. To extract reliable facts from the Web, MLP replaces the NTN's interaction function with a standard multi-layer perceptron.

**Neural Association Model (NAM)** [64] possesses multi-layer nonlinear activations in its deep neural nets, the objective of this spacial framework is detecting association conditional probabilities among any two possible facts. NAM can be applied to several probabilistic reasoning tasks such as triple classification,

**Table 6**
Summarization and comparison of recent popular Neural Network models for KGC.

| Model | Technique | Score Function | Loss function[a] | Notation | Datasets |
|---|---|---|---|---|---|
| Traditional neural network models: | | | | | |
| NTN [14] | Bilinear tensor layer | $g(h,r,t)=u^T f(p_1+p_2+b_r)$, $p_1=v_h^T W_r^{[1:k]} v_t, p_2=V_r[v_h,v_t]^T$ | $max\ \mathcal{L}_{marg}$ | $W_r^{[1:k]}\in\mathbb{R}^{d\times d\times k}$, $V_r\in\mathbb{R}^{k\times 2d}$ $f=tanh()$ | WordNet, Freebase |
| MLP [9] | Improves NTN; Standard Multi-layer Perceptron | $s(h,r,t)=w^T f(p_1+p_2+p_3)$, $p_1=M_1 v_h, p_2=M_2 v_r, p_3=M_3 v_t$ | – | $v_h,v_r,v_t\in\mathbb{R}^d$, $M_i\in\mathbb{R}^{d\times d}$, $w\in\mathbb{R}^d$, $f=tanh()$ | KV |
| NAM [64] | Multi-layer nonlinear activations; probabilistic reasoning | $s(h,r,t)=g(v_t^T u^{(L)})$ $u^{(l)}=f(W^{(l)}u^{(l-1)}+b^{(l)})$ $u^{\{0\}}=[v_h,v_r]$ | $\mathcal{L}_{ll}$ | $v_h,v_t,v_r\in\mathbb{R}^d$, $g=sigmoid()$, $f=ReLU()$ | WN11, FB13 |
| SENN [65] | Embedding shared fully connected neural network; adaptively weighted loss mechanism | $s(h,t)=v_r A_R^T, s(r,t)=v_h A_E^T, s(h,r)=v_t A_E^T$, $v_r=f(f(...f([h;t]W_{r,1}+b_{r,1})...))W_{r,n}+b_{r,n}$ $v_h=f(f(...f([r;t]W_{h,1}+b_{h,1})...))W_{h,n}+b_{h,n}$ $v_t=f(f(...f([h;r]W_{t,1}+b_{t,1})...))W_{t,n}+b_{t,n}$ | Joint adaptively weighted loss | $v_h,v_t,v_r\in\mathbb{R}^d$, $A_E\in\mathbb{R}^{\lvert E\rvert\times d}$, $A_R\in\mathbb{R}^{\lvert R\rvert\times d}$ $f=ReLU()$ | WN18, FB15K |
| ParamE [66] | MLP; CNN; gate structure; embed relations as NN parameters | $s(h,r,t)=((f_{nn}(v_h;v_r))W+b)v_t$ $v_r=Param_{f_{nn}}$ | $\mathcal{L}_{BCE}$ | $v_h,v_t,v_r\in\mathbb{R}^d$, $W\in\mathbb{R}^{d\times n}, b\in\mathbb{R}^d$, $g=sigmoid()$, $f=ReLU()$ | FB15k-237, WN18RR |
| CNN-based KGC models: | | | | | |
| ConvE [33] | Multi-layer 2D CNN; 1-N scoring programs | $s(h,r,t)=f(vec(f(concat(\widehat{v_h},\widehat{v_r})*\Omega))W)\cdot v_t$[b] | $\mathcal{L}_{BCE}$ | $v_h,v_t\in\mathbb{R}^d$, $\widehat{v_h},\widehat{v_r}\in\mathbb{R}^{d_w\times d_h}$; $v_r\in\mathbb{R}^{d'}$, $d=d_w d_h$; $f=ReLU()$; $\Omega$ : filter sets | WN18, FB15k, YAGO3-10, Countries, FB15k-237 |
| InteractE [67] | Feature Permutation; Checkered Reshaping; Circular Convolution | $s(h,r,t)=g(vec(f(\phi(\mathcal{P}_k)\circ w))W)v_t$[c] $\mathcal{P}_i=[(v_h^1,v_r^1);...;(v_h^i,v_r^i)]$ | $\mathcal{L}_{BCE}$ | $v_h,v_t,v_r\in\mathbb{R}^d$, $d=d_w dh$; $f=ReLU()$, $g=sigmoid()$; $w$: a filter | FB15K-237, WN18RR, YAGO3-10 |
| ConvKB [68] | 1D CNN; Transitional characteristic; L2 regularization | $s(h,r,t)=concat(g([v_h,v_r,v_t]*\Omega))\cdot W$[b] | $\mathcal{L}_{nll}$ | $v_h,v_t,v_r\in\mathbb{R}^d$ $g=ReLU()$, $\Omega$ : filter sets; | WN18RR, FB15k-237 |
| CapsE [69] | ConvKB; capsules networks | $s(h,r,t)=\lVert cap(g([v_h,v_r,v_t]*\Omega))\rVert$[b] | $\mathcal{L}_{nll}$ | $v_h,v_r,v_t\in\mathbb{R}^d$; $g=ReLU()$; $\Omega$ : filter sets; $cap()$ : Capsule-Network | WN18RR, FB15k-237 |
| GCN-based KGC Models: | | | | | |
| R-GCN [70] | Basis decomposition; block-diagonal-decomposition; end-to-end framework: encoder: R-GCN, decoder: DistMult | $s(h,r,t)=v_h^T W_r v_t$ | $\mathcal{L}_{BCE}$ | $v_h,v_t\in\mathbb{R}^d$; $W_r\in\mathbb{R}^{d\times d}$ | WN18RR, FB15k, FB15k-237 |
| SACN [71] | End-to-end framework: encoder: WGCN, decoder: Conv-TransE | $s(h,r,t)=f(vec(M(v_h,v_r))W)v_t$ | – | $f=ReLU()$; $W\in\mathbb{R}^{Cd\times d}$; $M(v_h,v_r)\in\mathbb{R}^{C\times d}$; $C$: kernels number | FB15k-237, WN18RR, FB15k-237 -Attr |
| COMPGCN [72] | Entity-relation-composition operators; end-to-end framework: encoder: COMPGCN, decoder: ConvE, DistMult, etc. | $s_{ConvE}$, $s_{DistMult}$, etc. | – | – | FB15k-237, WN18RR |

(*continued on next page*)

recognizing textual entailment, especially responds well for commonsense reasoning.

**Shared Embedding based Neural Network (SENN)** [65] explicitly differentiates the prediction tasks of head-entities, relations, and tail-entities by use of three respective substructures with fully-connected neural networks in an embedding sharing manner. Then the prediction-specific scores gained from substructures are employed to estimate the possibility of predictions. An adaptively weighted loss mechanism enables SENN to be more efficient in dealing with diverse prediction tasks and various mapping styles during the training process.

**ParamE** [66] is an expressive and translational KGC model which regards neural network parameters as relation embeddings, while the head entity embeddings and tail entity embeddings are regarded as the input and output of this neural network respectively. To confirm whether ParamE is a general framework for different NN architectures, this paper designs three different NN architectures to implement ParamE: multi-layer perceptrons (MLP), convolution layers, and gate structure layers, called ParamE-MLP,

**Table 6** (*continued*).

| Model | Technique | Score Function | Loss function[a] | Notation | Datasets |
|---|---|---|---|---|---|
| GAN-based KGC Models: | | | | | |
| KBGAN [28] | Discriminator+generator; negative sampling; reinforcement learning | $s_{trans}$[d] | $\mathcal{L}_{marg}$ | – | FB15k-237, WN18, WN18RR |
| IGAN [31] | Discriminator+generator; negative sampling; reinforcement learning; non-zero loss | $s_{trans}$ or $s_{sem}$[d] | $\mathcal{L}_{marg}$ | – | FB15K, FB13, WN11, WN18 |
| KSGAN [29] | Discriminator + generator + selector; negative sampling; reinforcement learning; non-zero loss | $s_{sem}$[d] | $\mathcal{L}_{marg}$ | – | FB15k-237, WN18, WN18RR |

[a] $\mathcal{L}_{ll}$ ($\mathcal{L}_{nll}$), $\mathcal{L}_{marg}$ and $\mathcal{L}_{BCE}$ are (negative) log likely-hood loss, margin-based ranking loss and binary cross entropy loss respectively.
[b] '$*$' means a convolution operator.
[c] '$\circ$' means depth-wise circular convolution.
[d] '$s_{trans}$' and '$s_{sem}$' are respective the score function of translation models and semantic matching models.

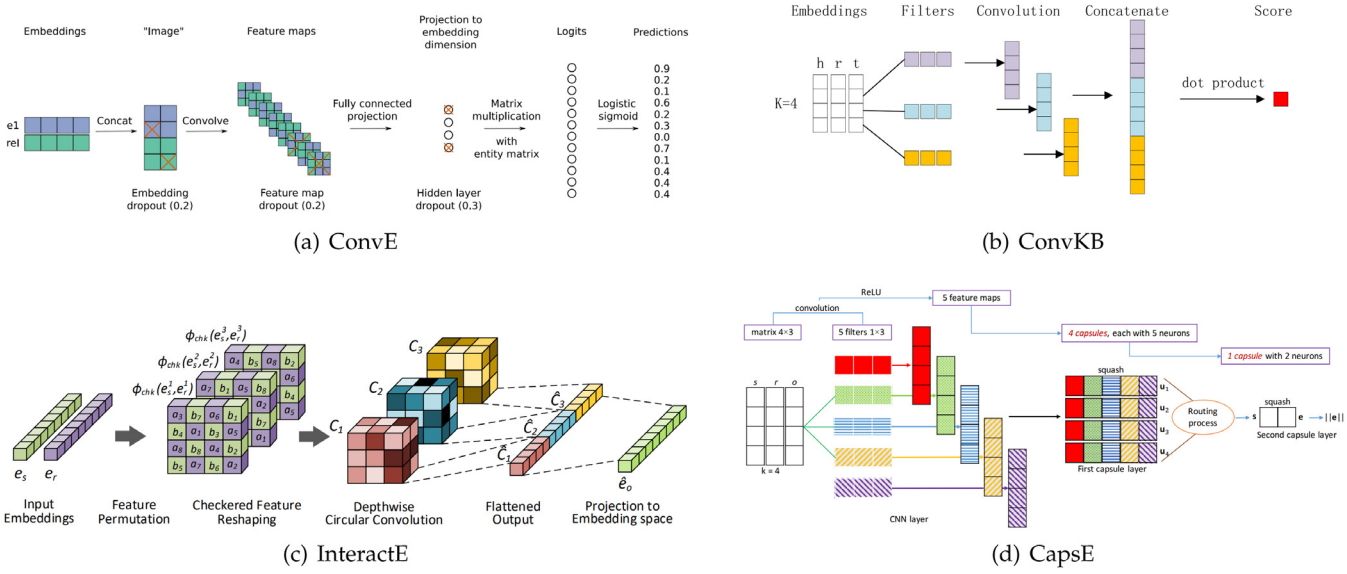



**Fig. 10.** The summarized frameworks of several CNN-based KGC models.
*Source:* Figures are extracted from [33,67–69].

ParamE-CNN, ParamE-Gate. Significantly, ParamE embeds the entity and relation representations in feature space and parameter space respectively, this makes entities and relations be mapped into two different spaces as expected.

*3.1.2.2. Convolutional Neural Network (CNN)-based KGC models.* We summarize some CNN-based KGC methods and draw a related figure (Fig. 10) for exhibiting the whole architecture of them, from which we can clearly know the learning procedure of these models.

**ConvE** [33] describes a multi-layer 2D convolutional network model for LP task, which is the first attempt that uses 2D convolutions over graph embeddings to explore more valuable feature interactions. ConvE defines its score function by a convolution over 2D shaped embeddings as:

$$s(h,r,t) = f(vec(f([\bar{v}_h;\bar{v}_r] * \omega))W)v_t$$

where the relation parameter $v_r \in \mathbb{R}^k$, $\bar{v}_h$ and $\bar{v}_r$ represent the 2D reshaping of $v_h$ and $v_t$ respectively, which conform to: both the $\bar{v}_h, \bar{v}_r \in \mathbb{R}^{k_w \times k_h}$ when $v_h, v_r \in \mathbb{R}^k$, where $k = k_w k_h$ in which the $k_h, k_w$ denotes the width and height of the reshaped 2D matrix. The $vec()$ means the vectorization operation, while $f()$ indicates the basic nonlinear transformation function, rectified linear units, for faster training [76]. ConvE owns much fewer parameters but is significantly efficient when modeling high-scale KGs with high degree node numbers. This work also points out the test set leakage issue of WN18 and FB15k datasets, performing a comparative experiment on their robust variants: WN18RR and FB15K-237.

**InteractE** [67] further advances ConvE by increasing the captured interactions to heighten LP's performance. InteractE chooses a novel input style in a multiple permutation manner and replaces simple feature reshaping of ConvE with the checked reshaping. Additionally, its special circular convolution structure is performed in a depth-wise manner.

**ConvKB** [68] is proposed after ConvE—the main difference between ConvKB and ConvE is that ConvKB uses 1D convolution expecting to extract global relations over the same dimensional entries of an input triple matrix, which indicated that ConvKB concerns at the transitional characteristics of triples. According to the evaluation on two benchmark datasets: WN18RR and FB15k-237, ConvKB performs better grades compared with ConvE and some other past models, which may be due to the efficient CNN

structure as well as the design for extracting the global relation information so that ConvKB will not ignore the transitional characteristics of triples in KGs.

**CapsE** After ConvKB, Nguyen et al. [69] next present CapsE to model triples by employing the capsule network [77], a network whose original intention is capturing entities in images. It is the first attempt at applying a capsule network for KGC. The general framework of CapsE is shown in Fig. 10(d), from which we can see after feeding to a convolution layer with multiple filters sets $\Omega$ as ConvKB dose, the 3-column triple matrix then is transformed into different feature maps, and these feature maps are later reconstructed by two capsule layers. A routing algorithm extended from Sabour et al. [77] guides the routing process between these two capsule layers. To that end, a continuous vector was produced whose length can be used to compute the score function of the triple:

$$s(h, r, t) = \|capsnet(g([v_h, v_r, v_t] * \Omega))\|$$

where *capsnet* and $*$ mean the capsule network operator and convolution operation respectively. Experimental results confirm that the CapsE model performs better than ConvKB [68] on WN18RR and FB15k-237.

*3.1.2.3. Graph Convolution Network (GCN)-based KGC models.* Graph Convolution Network (GCN) [78] was introduced as a generalization of Convolutional Neural Networks (CNNs),[1] which are a popular neural network architecture defined on a graph structure [70,80,82]. Recently, lots of researchers have employed GCNs to predict missing facts in KGs.

**R-GCN** [70] is presented as an extension of GCNs that operate on local graph neighborhoods to accomplish KGC tasks. R-GCN uses relation-specific transformations different from regular GCNs as the encoder side. For the LP task, the DisMult model was chosen to be the decoder to perform a computation of an edge's score. To avoid over-fitting on sparse relations and massive growth of model parameters, this work utilizes block-diagonal-decomposition methods to regularize the weights of R-GCN layers. R-GCN can act as a competitive, end-to-end trainable graph-based encoder (just like SACN [71] shows), i.e., in LP task, the R-GCN model with DistMult factorization as the decoding component outperformed direct optimization of the factorization model and achieved competitive results on standard LP benchmarks.

**Structure-Aware Convolutional Network (SACN)** [71] is an end-to-end model, where the encoder uses a stack of multiple **W-GCN** (Weighted GCN) layers to learn information from both graph structure and graph nodes' attributes, the W-GCN framework addresses the over-parameterization shortcoming of GCNs by assigning a learnable relational specific scalar weight to each relation and multiplies an incoming "message" by this weight during GCN aggregation. The decoder **Conv-TransE** is modified based on ConvE but abolishes the reshape process of ConvE, and simultaneously keeps the translational property among triples. In summary, the SACN framework efficiently combines the advantages of ConvE and GCN, thus obtain a better performance than the original ConvE model when experimenting on the benchmark datasets FB15k-237, WN18RR.

**COMPGCN** [72] Although R-GCN and W-GCN show performance gains on KGC task, they are limited to embedding only the entities of the graph. On this basis, COMPGCN systematically leverages entity-relation composition operations from KGE techniques to jointly embed entities and relations in a graph. Firstly, COMPGCN alleviates the over-parameterization problem by performing KGE composition ($\phi(u, r)$) of a neighboring node $u$ with respect to its relation $r$, to substitute the original neighbor parameter $v_u$ in the GCNs, therefore COMPGCN is relation-aware. Additionally, to ensure that COMPGCN scales with the increasing number of relations, COMPGCN shares relation embeddings across layers and uses basis decomposition based on the basis formulations proposed in R-GCN. Different from R-GCN which defines a separate set of basis matrices for each GCN layer, COMPGCN defines basis vectors and only for the first GCN layer, while the later layers share the relations through the relation embedding transformations performed by a learnable transformation matrix. This makes COMPGCN more parameter efficient than R-GCN.

Recently more and more novel effective GCN methods are proposed to conduct the graph analytical tasks. To efficiently exploit the structural properties of relational graphs, some recent works try to extend multi-layer GCNs to specific tasks for obtaining proper graph representation. For example, Bi-CLKT [83] and JKT [84], which are both knowledge tracing methods [85], apply two-layer GCN structure to encode node-level and global-level representations for relational subgraphs exercise-to-exercise (E2E) and concept-to-concept (C2C), respectively. The utilization of two-layer GCN can effectively learn the original structural information from multidimensional relationship subgraphs. Besides, ie-HGCN [86] try to learn interpretable and efficient task-specific object representations by using multiple layers of heterogeneous graph convolution on the Heterogeneous Information Network (HIN) [87]. Based on these works, a possible direction of future research is to explore the multi-layer GCN to efficiently capture different levels of structural information of KGs for the KGC task.

*3.1.2.4. Generative adversarial network (GAN)-based KGC models.* Generative adversarial network (GAN) [88] is one of the most promising methods for unsupervised learning on complex distribution in recent years, whose intention is originally proposed for generating samples in a continuous space such as images. GAN usually consists of at least two modules: a *generative module* and a *discriminative module*, the former accepts a noise input and outputs an image while the latter is a classifier that classifies images as "true" (from the ground truth set) or "fake" (generated by the generator), these two parts train and learn together in a confrontational way. However, it is not possible to use the original version of GANs for generating discrete samples like natural language sentences or knowledge graph triples since gradients from propagation back to the generator are prevented by the discrete sampling step [28] until SEQGAN [89] firstly gives successful solutions to this problem by using reinforcement learning — it trains the generator using policy gradient and other tricks. Likewise, there have been arisen lots of KGC works that incorporated the GAN framework in knowledge representation learning. Table 7 shows the general information about the GAN-based negative sampling methods. Intuitively, we place Fig. 11 to reveal the frame structure of GAN-based models.

**KBGAN** [28] aims to employ *adversarial learning* to generate high-quality negative training samples and replace formerly used uniform sampling to improve Knowledge Graph Embedding (KG embedding). As Fig. 11(a) shows, KBGAN takes KG embedding models that are probability-based and have a log-loss function as the generator to supply better quality negative examples, while the discriminator uses distance-based, margin-loss KG embedding models to generate the final KG embeddings. More specifically, it expects the generator to generate negative triples $(h', R, t')$ that obey the probability distribution of $p_G(h', r, t'|h, r, t)$, and

[1] Whereas CNNs require regular structure data, such as images or sequences, GCNs allow for irregular graph-structured data [79]. GCNs can learn to extract features from the given node (entity) representations and then combine these features together to construct highly expressive entity vectors, which can further be used in a wide variety of graph-related tasks, such as graph classification [80] and generation [81].

**Table 7**
Characteristic of several GAN-based negative sampling technologies for KGC.

| Models | KBGAN [21] | IGAN [31] | KSGAN [29] |
|---|---|---|---|
| Modules | Generator, discriminator | Generator, discriminator | Generator, discriminator, knowledge selector |
| Generator | Semantic matching models with softmax probabilistic models | Neural network | Translational distance models with softmax probabilistic models |
| Discriminator | Translational distance models | KGE models | Semantic matching models |
| Generator reward function | $R_G = \sum_{(h,r,t)\in\mathcal{T}} \mathbb{E}_{(h',r,t')\sim p_G}[R]$ | $J(\theta) = \mathbb{E}_{e\sim p(e\|\cdot;\theta)}[R]$ | $R_G = \sum_{(h,r,t)\in\mathcal{T}} \sum_{(h',r,t')\in\mathcal{T}_{s'}} \mathbb{E}_{(h',r,t')\sim p_G}[R]$ |
| Discriminator reward function | $R = -f_D(h',r,t') - b(h,r,t)$ | $R = tanh(f_r(h,t) - f_r(h',t') + \gamma)$ | $R = f_D(h',r,t')$ |
| Probability distribution of sampling | $p_G(h',r,t'\|h,r,t) = \frac{exp\ s_G(h',r,t')}{\sum exp\ s_G(h^*,r,t^*)}$ | $p(e\|(h,r,t),z;\theta) = z\cdot(e\|t,r;\theta) + (1-z)\cdot(e\|h,r^{-1};\theta)$ | $p_G(h',r,t'\|h,r,t) = \frac{exp\ f_G(h',r,t')}{\sum exp\ f_G(h^*,r,t^*)}$ |
| Selector | – | – | $f_{sel}(h',r,t') = \max_{(h',r,t')\in\mathcal{T}_{s'}} (f_D(h',r,t'))$ |

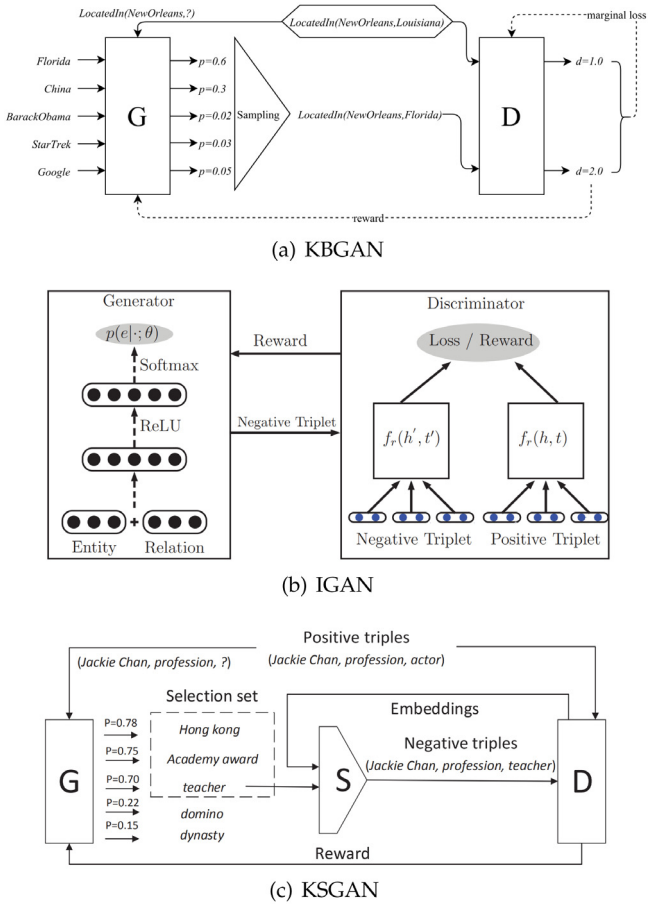



**Fig. 11.** Several GAN-based KGC methods about negative sampling.
*Source:* Figures are adapted from [28,29,31]

assumes that the score function of the discriminator is $s_D(h,r,t)$, then the objective of the discriminator is to minimize the margin loss function as follows:

$$L_D = \sum_{(h,r,t)\in\mathcal{T}} [s_D(h,r,t) - s_D(h',r,t') + \gamma]_+,$$

$$(h',r,t') \sim p_G(h',r,t'|h,r,t)$$

while the objective function of the generator is defined as a negative distance expectation:

$$R_G = \sum_{(h,r,t)\in\mathcal{T}} \mathbb{E}[-s_D(h',r,t')], \quad (h',r,t') \sim p_G(h',r,t'|h,r,t)$$

Note that those negative samples are created by the generator and its probability distribution $p_G$ is modeled with:

$$p_G(h',r,t'|h,r,t) = \frac{exp\ s_G(h',r,t')}{\sum exp\ s_G(h^*,r,t^*)},$$

$$(h^*,r,t^*) \in Neg(h,r,t)$$

where the $s_G(h',r,t')$ means the generator's score function and the candidate negative triples set are:

$$Neg(h,r,t) \subset \{(h',r,t)|h' \in E\} \cup \{(h,r,t')|t' \in E\}$$

To enable backpropagation of errors in the generator, KBGAN relies on policy gradient, a variance-reduction REINFORCE method with a one-step reinforcement learning setting, to seamlessly integrate the generator module and discriminator module. On the one hand, KBGAN enhances the KG embedding considering adversarial learning, on the other hand, this framework is independent of specific embedding models so that it can be applied to a wide range of KG embedding models and without the need for external constraints.

**GAN-based framework (IGAN)** [31] is also an answer to negative sampling in the KGC procedure, which can obtain quality negative samples to provide non-zero loss situation for discriminator, thus it makes full use of discriminator to operate with a margin-based ranking loss. Different from [28], IGAN obeys a probability distribution of the entity set $E$ as:

$$p(e|(h,r,t),z;\theta) = z\cdot p(e|t,r;\theta) + (1-z)\cdot p(e|h,r^{-1};\theta)$$

where the binary flag $z \in \{1,0\}$ reflects whether to replace head entity or tail entity. By the way, the GAN-based model is also flexible with good adaptive capacity to be extended to various KG embedding models. The general process of IGAN is shown in Fig. 11(b).

**KSGAN** [29] further advances KBGAN by adopting a selective adversarial network to generate better negative examples for training as shown in Fig. 11(c). The proposed new knowledge selective adversarial network adds a new knowledge selector module to the previous adversarial network structure to enhance the performance of discriminator and generator in KBGAN, purposely picks out corrupted triples are of high quality from generator who has the high discriminator score:

$$s_{sel}(h',r,t') = \max_{(h',r,t')\in\mathcal{T}_{s'}} (s_D(h',r,t'))$$

where the picked corrupted triples compose a selection set $\mathcal{T}_{s'}$, thus the selector selects negative triples with correct semantic information or close distance which can help the discriminator to avoid zero loss during the training process.

**Table 8**
Published results of Neural Network-based KGC methods. Best results are in bold.

| Model | WN18RR | | | | | FB15K-237 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MR | MRR | Hits@1 | Hits@3 | Hits@10 | MR | MRR | Hits@1 | Hits@3 | Hits@10 |
| ConvE [33][b] | 4464 | 0.456 | 0.419 | 0.470 | 0.531 | 245 | 0.312 | 0.225 | 0.341 | 0.497 |
| ConvKB [68][a] | 3433 | 0.249 | – | – | 0.524 | 309 | 0.243 | – | – | 0.421 |
| CapsE [69][a] | **718** | 0.415 | – | – | **0.559** | 403 | 0.150 | – | – | 0.356 |
| InteractE [67] | 5202 | 0.463 | 0.430 | – | 0.528 | **172** | 0.354 | 0.263 | – | 0.535 |
| R-GCN [70][b] | 6700 | 0.123 | 0.080 | 0.137 | 0.207 | 600 | 0.164 | 0.100 | 0.181 | 0.300 |
| SACN [71] | – | 0.470 | 0.430 | 0.480 | 0.540 | – | 0.350 | 0.260 | 0.390 | 0.540 |
| Conv-TransE [71] | – | 0.460 | 0.430 | 0.470 | 0.520 | – | 0.330 | 0.240 | 0.370 | 0.510 |
| SACN with FB15k-237-Attr [71] | – | – | – | – | – | – | 0.360 | 0.270 | 0.400 | 0.550 |
| COMPGCN [72] | 3533 | 0.479 | 0.443 | 0.494 | 0.546 | 197 | 0.355 | 0.264 | 0.390 | 0.535 |
| ParamE-MLP [66] | – | 0.407 | 0.384 | 0.429 | 0.445 | – | 0.314 | 0.240 | 0.339 | 0.459 |
| ParamE-CNN [66] | – | 0.461 | 0.434 | 0.472 | 0.513 | – | 0.393 | 0.304 | 0.426 | **0.576** |
| ParamE-Gate [66] | – | **0.489** | **0.462** | **0.506** | 0.538 | – | **0.399** | **0.310** | **0.438** | 0.573 |
| KBGAN [28] | – | 0.215 | – | – | 0.469 | – | 0.277 | – | – | 0.458 |
| KSGAN [29] | – | 0.220 | – | – | 0.479 | – | 0.280 | – | – | 0.465 |

[a] Resulting numbers are re-evaluated by [90].
[b] Resulting numbers are reported by [91], and others are taken from the original papers.

*3.1.2.5. Performance analysis about neural network-based KGC models.* We report the published results of Neural Network-based KGC approaches in Table 8 and make a simple comparison between them. From Table 8 we have the following findings:

**1.** Among the first four CNN-based KGC models, CapsE performs well on the WN18RR because (1) in CapsE, the length and orientation of each capsule in the first layer can help to model the important entries in the corresponding dimension, so that CapsE is good at handling much sparser datasets, like WN18RR. (2) CapsE uses pre-trained Glove [92] word embeddings for initialization and uses additional information.

**2.** R-GCN, SACN and its variants, and COMPGCN are all the extensions of GCNs, both SACN and COMPGCN make use of the weighted GCN to aggregate the neighbor information by the learnable weights, therefore they all perform relatively consistent excellent results on all datasets. Besides, "SACN with FB15k-237-Attr" uses additional attribute information in the FB15k-237 dataset, which further results in higher results on the FB15k-237.

**3.** We observe that the "ParamE-Gate" basically outperforms all the other neural network models, obviously reflects in the MRR, Hits@1, and Hits@3 metrics on both datasets. Note that ConvE and ParamE-CNN have similar network architectures, but ParamE-CNN achieves a substantial improvement over ConvE. ParamE-CNN takes parameters in itself as relation embeddings, which can capture the intrinsic property and is more reasonable [66]. The performance comparison among "ParamE-MLP", "ParamE-CNN" and "ParamE-Gate" shows that MLP has a weaker modeling ability than convolution layers and the gate structure. Moreover, although convolution layers are good at extracting features, "ParamE-CNN" performs worse than "ParamE-Gate" because the gate structure can optionally let some useful information through. In addition, although the differences between the FB15k-237 dataset and the WN18RR dataset let some models get un-balanced performance for the two datasets, ParamE-Gate can work well in both datasets.

*3.1.2.6. Discussion on Neural Network Models.* Also be known as non linear models, the neural network KGC models relying on neural network structure (along with the non-linear Activation Function, such as *sigmoid* function, *tanh* function, *Rectified Linear Unit (ReLU)* function etc., this situation can be seen from Table 6) to learn deep potential features.

Many literatures on KGE use neural networks to represent KGs in low-dimensional continuous space [11,14,15,64]. It can effectively extract hidden latent features needed for knowledge reasoning with strong accuracy, high reasoning scalability, and efficiency. However, neural network KGC models *rely on a large number of training data, which is a kind of data-driven works*, therefore they usually do not perform well when dealing with sparse KG data because of its great dependence on data. Moreover, these kinds of models have some other shortcomings, such as low interpretation, too many parameters, and poor performance in handling sparse KGs.

With the diversity research of the KGC method, more additional information is used in the completion work. It should be noted that there are several models we previously discussed making use of some additional information besides structural information. For example, the typical neural network KGC model SACN [71] applies a weighted graph convolutional network (WGCN) as its encoder, which utilizes node attributes and relation types information.

The widely known CNN-based KGC models have effective performance that benefit from the strong expressiveness of neural networks. Typically, the ConvE and ConvKB tend to be applied as the decoder model in lots of KGC methods (such as [72,91]) to conduct KGC. So also, there are other various neural network families that have been widely applied working with different additional information for conducting KGC. Take the recurrent neural network (RNN) as an example, because of its superior ability to learn sequence features, RNN often is used in the relational path-based KGC methods and also be exploited to deal with long text information (e.g., entity description text) for KGC. Similarly, CNN can be regarded as a feature extractor for textual feature modeling in KGC procedure substituting RNN structure (e.g., [93–95]). Zia et al. [96] is also an example that involves GAN structure combined with path information, which will be introduced in detail in the subsequent additional information based KGC methods.

### 3.2. Translation models

As a family of methods concentrating on distributed representation learning for KGs, *translation models* are both straightforward and have satisfied performance on KGC, they are promising to encode entities as low dimensional embeddings and relations between entities as translation vectors. This kind of model usually defines a *relation-dependent translation scoring function* to measure the probability of a triple through the distance metric. In the ordinary sense, the distance score reflects the correctness of a triple $(h, r, t)$, and more generally, it collocates with a margin-based ranking loss for learning the translation relation between entities. We also list a brief table about the basic characteristics of introduced translation models in Table 9.

**Table 9**
Summarization and comparison about Translation models for KGC.

| Model | Highlights | Score Function | Notion Difination | Loss Objective[a] | Datasets[b] |
|---|---|---|---|---|---|
| TransE Extensions: | | | | | |
| TransE [11] | Precursory translation method | $s(h,r,t) = \|v_h + v_r - v_t\|$ | $v_h, v_r, v_r \in \mathbb{R}^d$ | $\mathcal{L}_{marg}$ | LP: WN, FB15K, FB1M |
| TransH [15] | Performs translation in relation-specific hyperplane | $s(h,r,t) = \|v_{h_\perp} + v_r - v_{t_\perp}\|$, $v_{h_\perp} = v_h - w_r^T v_h w_r$, $v_{t_\perp} = v_t - w_r^T v_t w_r$ | $v_h, v_r, v_r \in \mathbb{R}^d$; $w_r \in \mathbb{R}^d$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15k; TC: WN11, FB13, FB15K |
| TransR [12] | Converts *entity space* to *relation space* relational space projection | $s(h,r,t) = \|M_r v_h + v_r - M_r v_t\|$ | $v_h, v_t \in \mathbb{R}^d$, $v_r \in \mathbb{R}^k$; $M_r \in \mathbb{R}^{k\times d}$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15k; TC: WN11, FB13, FB15K |
| TransD [97] | Different relational mapping matrix to head and tail entity; vector multiplication | $s(h,r,t) = \|M_{rh} v_h + v_r - M_{rt} v_t\|$ $M_{rh} = v_{r_p} v_{h_p}^T + I^{k\times d}$, $M_{rh} = v_{r_p} v_{h_p}^T + I^{k\times d}$ | $v_h, v_t, v_{h_p}, v_{t_p} \in \mathbb{R}^d$; $v_r, v_{r_p} \in \mathbb{R}^k$; $M_{rh}, M_{rt} \in \mathbb{R}^{k\times d}$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15k; TC: WN11, FB13, FB15k |
| lppTransD [98] | Role-specific projection | $s(h,r,t) = \|M'_{rh} v_h + v_r - M'_{rt} v_t\|$ $M'_{rh} = v_{r_{ph}} v_{h_p}^T + I^{k\times d}$, $M'_{rt} = v_{r_{pt}} v_{t_p}^T + I^{k\times d}$ | $v_h, v_t, v_{h_p}, v_{t_p} \in \mathbb{R}^d$; $v_r, v_{r_{ph}}, v_{r_{pt}} \in \mathbb{R}^k$; $M_{rh}, M_{rt} \in \mathbb{R}^{k\times d}$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15K; TC: WN11, FB13, FB15K |
| TransF [99] | Light weight and robust; explicitly model basis subspaces of projection matrices | $s(h,r,t) = \|M_{rh} v_h + v_r - M_{rt} v_t\|$, $M_{rh} = \sum_{i=1}^{f} \alpha_r^{(i)} U^{(i)} + I$, $M_{rt} = \sum_{i=1}^{f} \beta_r^{(i)} V^{(i)} + I$ | $v_h, v_r \in \mathbb{R}^d$, $v_r \in \mathbb{R}^k$; $U^{(i)}, V^{(i)} \in \mathbb{R}^{k\times d}$ $M_{rh}, M_{rh} \in \mathbb{R}^{k\times d}$ | $\mathcal{L}_{marg}$ | LP: FB15k, WN18; TC: FB15k-237, WN18RR |
| STransE [100] | SE+TransE | $s(h,r,t) = \|W_{r,1} v_h + v_r - W_{r,2} v_t\|$ | $v_h, v_r, v_r \in \mathbb{R}^d$, $W_{r,1}, W_{r,2} \in \mathbb{R}^{d\times d}$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15k |
| Trans-FT [101] | Flexible translation modeling | $s(h,r,t) =$ $(v_{h_r} + v_r)^T v_{t_r} + v_{h_r}^T (v_{t_r} - v_r)$, $v_{h_r} = M_r v_h$, $v_{t_r} = M_r v_t$ | $v_h, v_t, v_{h_r}, v_{t_r} \in \mathbb{R}^d$; $v_r, v_{r_p} \in \mathbb{R}^k$, $M_r \in \mathbb{R}^{k\times d}$ | $\mathcal{L}_{marg}$ | LP: WN18,FB15k; TC: WN11, FB13, FB15K |
| Translation Models Using Attention Mechanism: | | | | | |
| TransM [102] | Relational mapping; property-specific weight | $s(h,r,t) = w_r \|v_h + v_r - v_t\|$ $w_r = \frac{1}{\overline{log(h_r pt_r + t_r ph_r)}}$ | $v_h, v_r, v_t \in \mathbb{R}^d$, $w_r \in \mathbb{R}$ $h_r pt_r$: heads per tail, $t_r ph_r$: tails per head | $\mathcal{L}_{marg}$ | LP: WN18, FB15K |
| ITransF [103] | Sparse attention mechanism; relation concepts sharing | $s(h,r,t) = \|v_{h_a tt} + v_r - v_{t_a tt}\|$ $v_{h_a tt} = \alpha_r^H \cdot D \cdot v_h$ $v_{t_a tt} = \alpha_r^T \cdot D \cdot v_t$ $\alpha_r^X = SparseSoftmax(v_r^X, I_r^X)$, $X = H, T$ | $v_h, v_r, v_t \in \mathbb{R}^d$; $\alpha_r^X \in [0,1]^m$, $I_r^X \in \{0,1\}^m$, $v_r^X \in \mathbb{R}^m, X =$ $H, T$; $D \in \mathbb{R}^{m\times d\times d}$ | $\mathcal{L}_{marg}$ | WN18 and FB15k |
| TransAt [104] | Relation-related entities categories; relation-related attention | $s(h,r,t) = P_r(h) + v_r - P_r(t)$ $P_r(h)^c = P_r(\sigma(r_h) v_h)$ $P_r(t)^c = P_r(\sigma(r_t) v_t)$ $P_r(x) = a_r * v_x, x = h, t$ | $v_h, v_r, v_t \in \mathbb{R}^d$; $a_r \in \{0,1\}^d$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15k; TC: WN11, FB13 |
| TransGate [105] | Gate structure; shared discriminate mechanism | $s(h,r,t) = \|v_{h_r} + v_r - v_{t_r}\|$ $x_r = x \odot \sigma(z)$, $z(x) = W_x \odot x + W_{rx} \odot r + b_x$ $x = h, t$ | $v_h, v_r, v_t \in \mathbb{R}^d$ | $\mathcal{L}_{marg}$ | LP: WN18RR, FB15K, FB15K-237; TC: WN11, FB13 |

(*continued on next page*)

### 3.2.1. *TransE extensions*

We introduce several prominent translation KGC models in TransE [11] family, which are frequently summarized and cited in lots of literature. We draw a comprehensive figure exhibiting some representative translation models (shown as Fig. 12).

**TransE** [11] as a pioneer translation KGC model, can balance both effectiveness and efficiency compared to most traditional methods. TransE projects entities and relations together into a continuous low-dimensional vector space, where the tail-entity $t$ in triple $(h, r, t)$ can be viewed as the translation operator results among the head entity $h$ and relation $r$, that are:

$$v_h + v_r \approx v_t$$

and it defines its score function as:

$$s(h,r,t) = \|v_h + v_r - v_t\|_{l_{1/2}}$$

However, the over-simplified translation assumption TransE holds might constraint the performance when modeling complicated relations, which leads to a weak character that TransE can only model pure $1-1$ relations in KGs. To effectively learn

**Table 9** (*continued*).

| Model | Highlights | Score Function | Notion Difination | Loss Objective[a] | Datasets[b] |
|---|---|---|---|---|---|
| Modification to Loss Objection of Translation-based KGC: | | | | | |
| TransRS [106] | Upper limit score function for positive triplets; limit-based scoring loss | $s(h,r,t) = \|v_h + v_r - v_t\|$<br>$f_r(h,t) \le \gamma'$ | $v_h, v_r, v_t \in \mathbb{R}^d$ | $\mathcal{L}_{marg}$<br>$+\mathcal{L}_{limit}$ | LP: WN18, FB15k;<br>TC: WN11, FB13, FB15K |
| TransESM [107] | Trans-RS+TransE's score function<br>Soft Margin loss | $s(h,r,t) = \|v_h + v_r - v_t\|$<br>$f_r(h,t) \le \gamma_1$,<br>$f_r(h',t') \ge \gamma_2 - \xi^r_{h,t}$,<br>$\xi^r_{h,t} \ge 0$ | $v_h, v_r, v_t \in \mathbb{R}^d$;<br>$\gamma_2 \ge \gamma_1 \ge 0$;<br>$(h',r',t') \in \mathcal{T}'$,<br>$(h,r,t) \in \mathcal{T}$ | *soft* $\mathcal{L}_{marg}$ | A scholarly KG |
| Transition Models in Novel Vector Space: | | | | | |
| TransA [108] | Adaptive metric approach;<br>elliptical surfaces modeling | $s(h,r,t) =$<br>$(\|v_h + v_r - v_t\|)^T W_r(\|v_h + v_r - v_t\|)$<br>$\|x\| \doteq (\|x_1\|, \|x_2\|, \ldots, \|x_n\|)$,<br>$x_i = v_{h_i} + v_{r_i} - v_{t_i}$ | $v_h, v_r, v_t \in \mathbb{R}^d$ | $\mathcal{L}_{marg}$ | LP: WN18, FB15K;<br>TC: WN11, FB13 |
| TorusE [109] | TransE+Torus | $s(h,r,t) =$<br>$\min_{(x,y)\in([h]+[r])\times[t]} \|x - y\|$ | $[h], [r], [t] \in T^n$<br>$T$ is a torus space | $\mathcal{L}_{marg}$ | LP: WN18, FB15K |
| RotatE [110] | Entire complex space $\mathbb{C}$;<br>self-adversarial negative sampling | $s(h,r,t) = \|v_h \circ v_r - v_t\|$ | $v_h, v_r, v_t \in \mathbb{C}^d$;<br>$v_{r_i} = \mathbb{C}, \|v_{r_i}\| = 1$ | $\mathcal{L}_{ns}$ | LP: FB15k, WN18, FB15k-237, WN18RR |

[a]Put simply, the $\mathcal{L}_{ns}$ and $\mathcal{L}_{marg}$ are *negative sampling loss* and *margin-based ranking loss* respectively, also, $\mathcal{L}^c_{marg}$ means a *Confidence-aware margin-based ranking loss* [111], and $\mathcal{L}_{limit}$ refers to the *Limit-based Scoring Loss* in [106], while the $\mathcal{L}_{HRS}$ is the *HRS-aware loss function* in [112].
[b]When we describe the datasets, we apply the shorthand for: 'LP' means Link Prediction task, while 'TC' means Triple Classification task.
*** The $v_{r_c}$, $v'_r$ and $v_{r_s}$ are respective the relation cluster embedding, relation-specific embedding and sub-relation embedding in [112].
[c]$P_r()$ is a projection function.

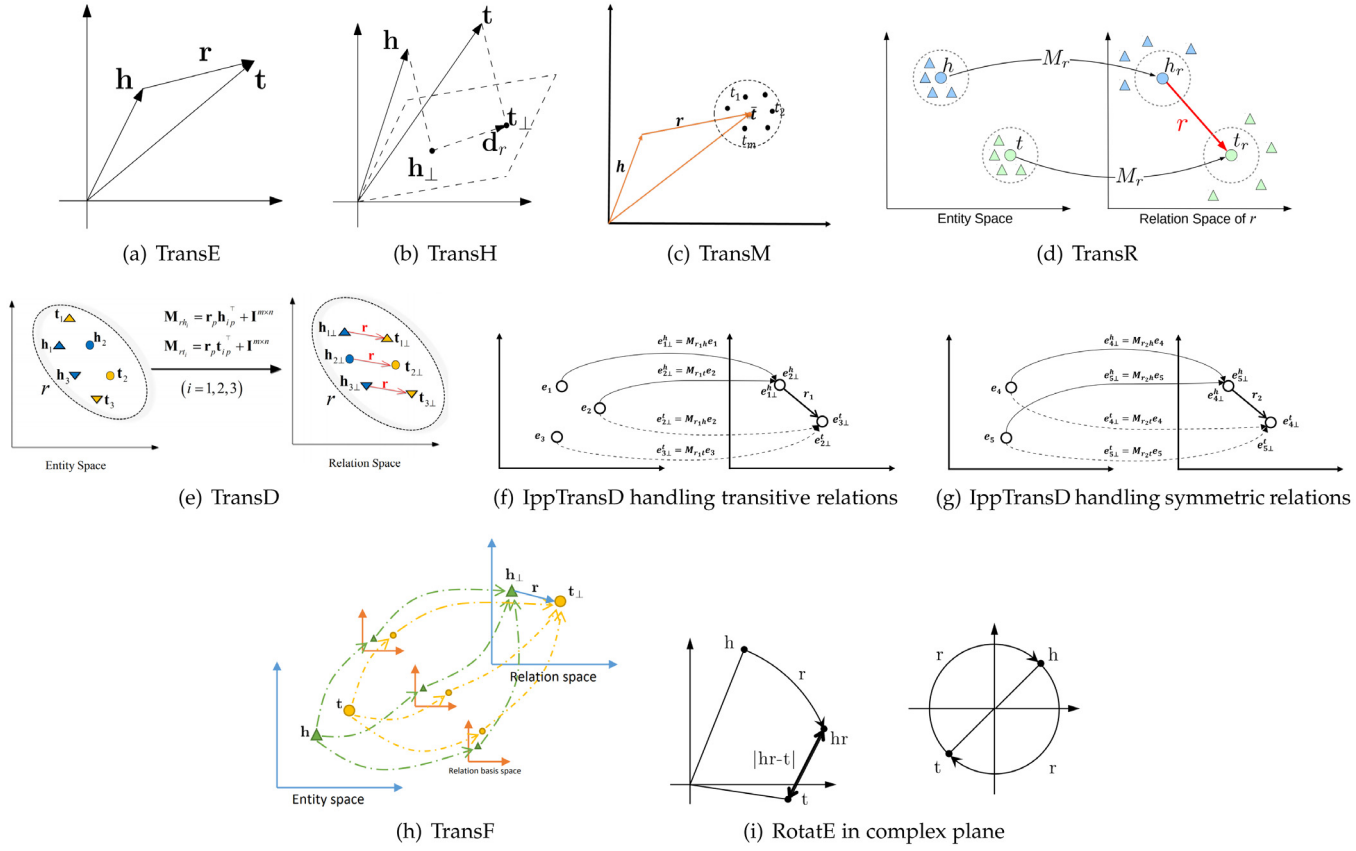



**Fig. 12.** TransE and its extension models. These pictures are referred to [12,15,97–99,102,110].

complex relation types and model various KG structures, a series of enhanced translation-based KGC models continuously improve the TransE.

**TransH** [15] projects the entities onto the relation-specific hyperplane $w_r$ (the normal vector) by $h_\perp = v_h - w_r^T v_h w_r$ or $t_\perp = v_t - w_r^T v_t w_r$ and then performs translation actions on this hyperplane, so that the score function is defined as follows:

$$s(h, r, t) = \|h_\perp + v_r - t_\perp\|_2^2$$

which can model the $1-n$, $n-1$ even $n-n$ relations availably.

**TransR** [12] considers that there are semantic differences between entities and relations so that they should be in different semantic spaces. Moreover, different relations should constitute different semantic spaces. It converts the entity space to corresponding relation space through a relational projection matrix $M_r \in \mathbb{R}^{d\times k}$, the translation performed in relation space is:

$$v_h M_r + v_r \approx v_t M_r$$

In order to better model internal complicated correlations within diverse relation type, this work also extends TransR by incorporating the idea of piecewise linear regression to form **Cluster-based TransR (CTransR)**, they introduce cluster-specific relation vector $r_c$ for each entity pairs cluster and matrix $M_r$. However, although TransR performs well in handling complicated relation patterns, it involves too many additional parameters to result in poor robustness and scalability issues for large KGs learning.

**TransD** [97] further advances TransR by assigning different relational mapping matrix $M_{rh}, M_{rt} \in \mathbb{R}^{m\times n}$ to head and tail entity respectively:

$$M_{rh} = r_p h_p^T + I^{m\times n}$$

$$M_{rt} = r_p t_p^T + I^{m\times n}$$

$$h_\perp = M_{rh} h, \quad t_\perp = M_{rt} t$$

The subscript $p$ marks the projection vectors. Then it scoring a triple $(h, r, t)$ by defining the following function:

$$s(h, r, t) = -\|h_\perp + r - t_\perp\|_2^2$$

Thus each objects in KGs is equipped with two vectors. Additionally, TransD replaces matrix multiplication with vector multiplication which significantly increases the speed of operation.

**lppTransD** [98] is an extension of TransD, which accounts for different roles of head and tail entities. They indicated that logical properties of relations like transitivity and symmetry cannot be represented by using the same projection matrix for both head and tail entities [99]. To preserve these logical properties, the *lpp-series* ideas consider a role-specific projection that maps an entity to a distinct vector according to its role in a triple, whether is a head entity or a tail entity. The concrete mapping matrices are designed as:

$$M'_{rh} = r_{ph} h_p^T + I^{m\times n}$$

$$M'_{rt} = r_{pt} t_p^T + I^{m\times n}$$

**TransF** [99] is similar to lppTransD, which also applies the same idea to compute the projection matrices for head and tail entities separately. The difference between lppTransD and TransF is that TransF mitigates the burden of relation projection by explicitly modeling the basis subspaces of projection matrices with two separate sets of basis matrices $U^{(i)}$, $V^{(i)}$, and the two factorized projection matrices are calculated as:

$$M_{r,h} = \sum_{i=1}^{s} \alpha_r^{(i)} U^{(i)} + I$$

$$M_{r,t} = \sum_{i=1}^{s} \beta_r^{(i)} V^{(i)} + I$$

Inspired by TransR, TransF is robust and lightweight enough to deal with the large-scale KGs through easily learning multiple relations by explicitly modeling the underlying subspace of the relation's specific projection matrix.

**STransE** [100] properly combines insights from SE [113] and TransE [11], draws on the experience of relation-specific matrices in SE for relation-dependent identification of both head entity and tail entity, also follows the basic translation principle in the TransE model.

**Trans-FT** [101] develops a general principle called *Flexible Translation (FT)*, which enables it to model complex and diverse objects in KGs unlike those previous translation models only concentrate on strict restriction of translation among entities/relations (such as TransE). Experiment adapts FT to existing translation models, TransR-FT gets the best performance compared to other two baselines (TransE-FT and TransH-FT).

### 3.2.2. *Translation models with attention mechanism*

**TransM** [102] is an appropriate solution to the inflexible issue in TransE. They focus more on the diverse contribution (i.e. various relational mapping properties) of each training triple to the final optimization target, therefore TransM decides to develop a weighted mechanism, with which each training triple can be assigned a pre-calculated distinct weight according to its relational mapping property. In other words, we can regard this weighted operation as an attention mechanism that takes every training example as a impact attention to tackle well with the various mapping properties of triplets.

**ITransF** [103] To make full use of the shared conceptions of relations and apply it to perform knowledge transfer effectively, ITransF outfits with a sparse attention mechanism to discover sharing regularities for learning the interpretable sparse attention vectors, which fully capture the hidden associations between relations and sharing concepts.

**TransAt** [104] effectively learns the translation-based embedding using a reasonable attention mechanism, it exploits a piecewise evaluation function which divides the KGC problem into a two-stage process: checking whether the categories of head and tail entities with respect to a given relation make sense firstly, and then considering for those possible compositions, whether the relation holds under the relation-related dimensions (attributes). During this two-stage process, TransAt uses *K-means* to cluster generating categories for generality. TransAt sets the projection function by computing the variances between head (tail) entities associated with relation $r$ in the training set for each dimension, additionally, it designs a threshold to determine whether a dimension should be retained. In consideration of the ORC structure problems [114], TransAt utilizes an asymmetric operation on both head entity and tail entity, therefore the same entities will have different representations of head position and tail position.

**TransGate** [105] pays close attention to inherent relevance between relations. To learn more expressive features and reduce parameters simultaneously, TransGate follows the thought of parameter sharing using gate structure and then integrates the shared discriminate mechanism into its architecture to ensure

that the space complexity is the same as indiscriminate models. The shared gates above-mentioned also be reconstructed with weight vectors to avoid matrix–vector multiplication operations, impelling the model to be more effective and scalable.

### 3.2.3. *Modification to loss objection of translation-based KGC*

Some translation models try to improve KGC by *modifying the objective functions* [106,107]. In order to facilitate the comparison among these improved loss programs, we can directly see Table 9, from which we can easily pick out them by their distinctive loss objectives.

**TransRS** [106] explores a limit-based scoring loss $L_S$ to provide an upper limit score of a positive triple and then adds this limit-based scoring loss item into the original loss function as a new loss function for optimizations. By this mean, the modified loss objective including two terms, a limit-based scoring loss as well as the original margin-based ranking loss $L_R$, that is:

$$L_{RS} = L_R + \lambda L_S, \ (\lambda > 0)$$

When applied the loss to the traditional translation baselines such as TransE and TransH (i.e., TransE-RS and TransH-RS), it achieves remarkable performance improvements compared with initial models.

**TransESM** [107] not only changes the score function and loss function of Trans-RS into TransE's score function with *Soft Margins (Margin Ranking Loss)* where soft margins allow false-negative samples to slightly slide into the margin, mitigating the adverse effects of false-negative samples, but also indicates that most existing methods are tested on datasets such as Freebase and WordNet, which may prevent the development of KGC technology. Therefore, they verify the TransESM and compares TransE with other models on the specific field datasets (faculty KG, academic KG), then found that TransE is better than ComplEx [43], TransH [15] and TransR [12] on these specific field datasets.

### 3.2.4. *Transition models in novel vector space*

Most of the translation distance models tend to leverage spherical equipotential hyper-surfaces with different plausibility. Unfortunately, the over-simplified loss metric they use limits their ability about modeling complex relational data in KGs. As shown in Fig. 13, on the equipotential hyper-surfaces, more near to the center, more plausible the triple is, thus it is difficult to correctly identify the matched answer entities from unmatched ones. As the common scene in KGs, complex relations (including $1{-}to{-}n$, $n{-}to{-}1$, and $n{-}to{-}n$ relations) always require complex embedding topologies techniques. Although complex embedding is an urgent challenge, the existing translation methods are not satisfied for this task because of the inflexibility of spherical equipotential hyper-surfaces.

**TransA** [108] More than modeling on a traditional spherical surface, TransA applies an adaptive and flexible metric on an elliptical surface for KG embedding. TransA not only represents the complex embedding topologies induced by complex relations well, but also can suppress the noise from unrelated dimensions as the TransA itself could be treated as weighting transformed feature dimensions in Adaptive Metric Approach.

**TorusE** [109] transforms the real vector space into a *torus* (a *compact Abelian Lie group* painted as Fig. 14), and keeps the same principle as TransE simultaneously. TorusE is proposed to overcome the TransE's regularization flaw that regularization conflicts with the translated-embedding principle and reduces the accuracy in LP task, meanwhile. A *Lie group* is a group that is also a finite-dimensional smooth manifold, in which the group operations of multiplication and inversion are smooth maps, while

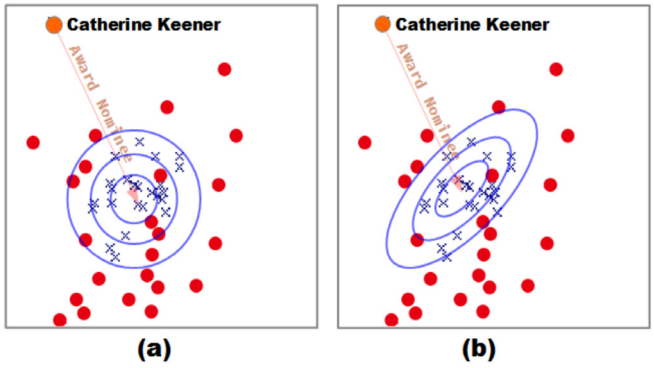



**Fig. 13.** Visualization of TransE embedding vectors for Freebase with PCA dimension reduction. The navy crosses are the matched tail entities for an actor's award nominee, while the red circles are the unmatched ones. TransE applies Euclidean metric and spherical equipotential surfaces and making seven mistakes as (a) shows, while TransA takes advantage of adaptive Mahalanobis metric and elliptical equipotential surfaces, avoiding four mistakes in (b) [108].

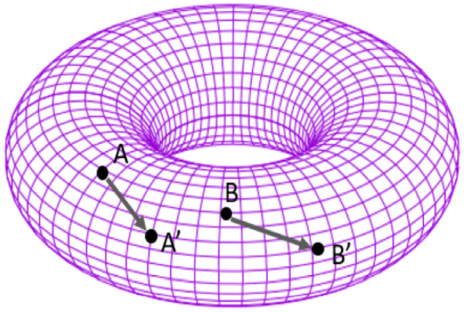



**Fig. 14.** Visualization of embeddings on 2-dimensional torus obtained by TorusE. Embeddings of the triples $(A, r, A')$ and $(B, r, B')$ are illustrated. Note that $[A'] - [A]$ and $[B'] - [B]$ are similar on the torus [109].

the *Abelian Lie group* is a special case of *Lie group* when the operation of multiplication is commutative, and it satisfies all the conditions that an embedding space should require according to TransE's embedding strategy above all. TorusE defines three types of scoring functions $f_{L_1}, f_{L_2}$ and $f_{eL_2}$ exploiting the distance functions. TorusE has good performance in the LP task, in addition to that, it has some other excellent characters, for instance, it not only has good computing performance but also possesses high scalability.

**RotatE** [110] Inspired by Euler's identity, RotatE is defined on an *entire complex space*, which has much more representation capacity than the above Lie group-based ToursE model, whereas the latter sets its embeddings to be fixed, which can be regarded as a special case of RotatE. RotatE can uniformly model and infer three relation patterns: symmetry/antisymmetry, inversion, and composition, and defines each relation as a rotation on the *entire complex space*. Moreover, RotatE develops a novel self-adversarial negative sampling technique to train the model effectively.

### 3.2.5. *Performance analysis about translation models*

We make a simple comparison on those translation models for KGC and report the results of them in Table 10, from which we have the following findings:

**1.** From the experimental results of the TransE and its extension models TransH, TransR, TransD, IppTransD, TransF, STransE, and

**Table 10**
Published link prediction results of translation models. Best results are in bold.

| Model | WN18 | | | FB15K | | | WN18RR | | | FB15K-237 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MR | MRR | Hits@10 | MR | MRR | Hits@10 | MR | MRR | Hits@10 | MR | MRR | Hits@10 |
| TransE [11] | 251 | – | 0.892 | 125 | – | 0.471 | **2300**[a] | 0.243[a] | 0.532[a] | 323[a] | 0.279[a] | 0.441[a] |
| TransH [15] | 303 | – | 0.867 | 87 | – | 0.644 | – | – | – | – | – | – |
| TransR [12] | 225 | – | 0.920 | 77 | – | 0.687 | – | – | – | – | – | – |
| TransD [97] | 212 | – | 0.922 | 91 | – | 0.773 | – | – | – | – | – | – |
| lppTransD [98] | 270 | – | 0.943 | 78 | – | 0.787 | – | – | – | – | – | – |
| TransF [99] | 198 | 0.856 | 0.953 | 62 | 0.564 | 0.823 | 3246 | **0.505** | 0.498 | 210 | 0.286 | 0.472 |
| STransE [100] | 206 | 0.657 | 0.934 | 69 | 0.543 | 0.797 | – | – | – | – | – | – |
| Trans-FT [101] | 342 | – | 0.953 | 49 | – | 0.735 | – | – | – | – | – | – |
| TransM [102] | 281 | – | 0.854 | 94 | – | 0.551 | – | – | – | – | – | – |
| ITransF [103] | 223 | – | 0.952 | 77 | – | 0.814 | – | – | – | – | – | – |
| TransAt [104] | **157** | – | 0.950 | 82 | – | 0.782 | – | – | – | – | – | – |
| TransRS [106] | 357 | – | 0.945 | 77 | – | 0.750 | – | – | – | – | – | – |
| TransA [108] | 392 | – | 0.943 | 74 | – | 0.804 | – | – | – | – | – | – |
| TorusE [109] | – | 0.947 | 0.954 | – | 0.733 | 0.832 | – | – | – | – | – | – |
| RotatE [110] | 309 | **0.949** | **0.959** | 40 | 0.797 | 0.884 | 3340 | 0.476 | **0.571** | **177** | 0.388 | 0.533 |
| TransGate [105] | – | – | – | **33** | **0.832** | **0.914** | 3420 | 0.409 | 0.510 | **177** | **0.404** | **0.581** |

[a]Resulting numbers are reported by [91] and others are taken from the original papers.

Trans-FT, we can conclude that: (1) Based on the translation idea of TransE, for a triple $(h, r, t)$, it is necessary to further consider the semantic differences between entities and relations. (2) TransF achieves a clear and substantial improvement over others in this series. The reason is that TransF factorizes the relation space as a combination of multiple sub-spaces for representing different types of relations in KGs. Besides, TransF is more robust and efficient than congeneric methods by modeling the underlying subspace of the relation's specific projection matrix for explicitly learning various relations.

**2.** The attention-based methods TransM, ITransF, and TransAt almost consistently outperform TransE. Specifically, ITransF performs better on most of the metrics of WN18 and FB15k, while TransM has a poor result on the sparser WN18 dataset. The reason is that ITransF employs a sparse attention mechanism to encourage conceptions of relations sharing across different relations, which primarily benefit facts associated with rare relations. TransAt focuses on the hierarchical structure among the attributes in an entity, so it utilizes a two-stage discriminative method to achieve an attention mechanism. It suggests that the proper attention mechanism can help to fit the human cognition of a hierarchical routine effectively.

**3.** Both TorusE and RotatE get good performance on the WN18 and FB15k. RotatE is good at modeling and inferring three types of relation patterns: the symmetry pattern, the composition pattern, and the inversion pattern, by defining each relation as a rotation in complex vector spaces. By comparison, TorusE focuses on the problem of regularization in TransE. Although TorusE can be regarded as a special case of RotatE since it defines KG embeddings as translations on a compact Lie group, the modulus of embeddings in TorusE are set fixed, while in RotatE is defined on the entire complex space, which is very critical for modeling and inferring the composition patterns. Therefore, RotatE has much more representation capacity than TorusE, which may help explain why RotatE gains better performance than TorusE on the WN18 and FB15k.

**4.** TransGate achieves excellent performance on four datasets, especially in the metrics of FB15k and FB15k-237. These results show the appropriateness of sharing discriminate parameters and the great ability of gate structure. Actually, TransGates is a better trade-off between the complexity and the expressivity by following the parameter sharing strategy. With the help of the shared discriminate mechanism based on the gate structure, TransGate can optimize embeddings and reduce parameters simultaneously. However, TransGate has a poorer performance on the WN18RR, since that WN18RR removes reverse relations and destroys the inherent structure of WordNet, which results in low relevance between relations and further reduces the effect of parameter sharing [105].

#### 3.2.6. *Discussion on translation models*

In summary, the translation models based on internal structure information are simple but surprisingly effective when solving the KGC problems. Additionally, the translation models only need few parameters. *At present, translation models usually are served as the basis for extended models that exploit a wider variety of additional information sources, which benefits from the easy-to-use translation transformation hypothesis*. Ordinarily, collaborate transitional characteristics with additional information to conduct KGC is an ongoing trend. This bunch of methods take account of other useful information instead of only utilizing the inner structure information, based on the translation distance classic baselines or follow the basic translation assumption thought. For instance, **OTE** [115] advances RotatE in two ways: (1) leveraging orthogonal transforms [116] to extend the RotatE from 2D complex domain to high dimension space for improving modeling ability, and (2) making use of the context information of nodes. **PTransE** (path-based TransE) [12] and **PTransD** [117] are both the path-augmented translation based models, while **TransN** [31] considers the dependencies between triples and incorporates neighbor information dynamically. On the other hand, people begin to explore how to implement the basic translation transformation of entities and relations in a more effective and reasonable modeling space to easily model complex types of entities and relations and various structural information. Under this case, the improvement and optimization of the loss function is also a promising research direction.

## 4. Additional information-based KGC technologies

The research on additional information-based KGC has received increasing attention in recent years. The techniques as surveyed in Section 3 perform KGC mainly relying on the structure information of KGs (i.e., the simple triple structure information), of course, several methods mentioned in Section 3 also simultaneously utilize the additional information for KGC. For example, KBAT [91] considers the multi-hop neighborhood information of a given entity to capture entity and relation features, and DrWT [45] leverages the additional Wikipedia page document of entities outside KGs. In this section, we focus on the additional

information-based KGC techniques, and make a comprehensive and fine-grained summarization and comparison.

We focused specifically on the incorporation of *two types* of additional information, including *internal side information inside* KGs and *external extra information outside* KGs:

- We introduce the usage of internal side information inside KGs in Section 4.1, which consists of five subclasses: node attributes information (in Section 4.1.1), entity-related information (in Section 4.1.2), relation-related information (in Section 4.1.3), neighborhood information (in Section 4.1.4) and path information (in Section 4.1.5).
- The investigations on incorporating external information outside KGs are in Section 4.2, which involves two aspects of contents: rule-based KGC in Section 4.2.1, and third-party data sources-based KGC in Section 4.2.2.

### 4.1. *Internal side information inside KGs*

The inherent rich information (i.e., internal information) inside KGs often is used during KG learning, these non-negligible information plays an important role in capturing useful features of knowledge embeddings for KGC and knowledge-aware applications. In general, the common internal side information inside KGs includes *node attributes information*, *entity-related information*, *relation-related information*, *neighborhood information*, and *relational path information*.

#### 4.1.1. *Node attributes information*

Nodes in KGs usually carry rich attribute information, this information often explains and reflects the characteristics of entities. For example, the *gender*, *age* and *appearance* of a person are respectively corresponding to the *textual* attribute, *non-discrete digital* attribute, and *image* attribute — they are the mainstream attribute information, which are usually exploited by cooperating with structure information of KGs to jointly learn KG embeddings. Although attribute information of entities is important to understand the entity and may help to alleviate the inherent sparsity and incompleteness problem that are prevalent in KGs [41], there is still less literature concern about attribute information when performing KGC task. We summarize KGC methods using node attribute, pay close attention to the usage of the numeric attribute, text attribute, and image attribute. The general characteristics of these methods are compared and listed in Table 11.

*4.1.1.1. Numeric atrribute information.* Numeric attribute information is a kind of available internal information for KG learning. Many popular KGs such as Freebase, YAGO, or DBPedia maintain a list of non-discrete attributes for each entity. Intuitively, these attributes such as height, price, or population count are able to richly characterize entities in KGs. Unfortunately, *many state-of-the-art KGC models ignore this information* due to the challenging nature of dealing with non-discrete data in inherently binary-natured KGs.

**KBLRN** Garcia et al. [118] firstly integrate latent, relational and numerical features of KGs for KGC with the support of new proposed end-to-end model KBLRN.

**MTKGNN** [119] is a multi-task learning approach constructed by a deep learning architecture, which not only leverages non-discrete attribute information in KGC but also aims to predict that numerical attributes.

**TransEA** [120] consists of two component modules, a structure embedding model and an attribute embedding model. TransEA extends TransE [11] with numeric attributes embedding by adding a numerical attribute prediction loss to the original relational loss of TransE.


**Fig. 15.** Some examples of entity images. This image is referred from [122].

*4.1.1.2. Text attribute information.* As an important supplement to structured information in KGs, internal semantic information, e.g., text attribute information such as literal names of nodes or edges, is adapted in many KGC studies. Earlier **NTN** [14] for KGC leverages entity names by averaging the embeddings of words involved in them, hoping to achieve semantic sharing among those learned vectors. Inspired by this idea, many relative KGC works have sprouted up to explore the usage of text attributes.

**JointAS** [15] and **JointTS** [121] propose novel KGE methods which jointly embed entities and words in entity names into a same continuous vector space.

*4.1.1.3. Image atrribute information.* Since image attributes associated with entities could provide significant visual information for KG learning, entity images also have been used to enhance KG embedding in some works. Fig. 15 demonstrates some examples of entity images. In KGs, each entity may have multiple images that intuitively describe the appearances and behaviors of this entity in a visual manner.

The representative **IKRL** [122] designs a specialized image encoder to generate the image-based representation for each image instance and jointly learn the KG representations with translation-based methods. To consider all image instances of an entity and further aggregate their image-based representation for each entity, they use an attention-based method to construct the aggregated entity embeddings. There also exists some literature that employs multiple kinds of attribute information of nodes involves image information, such as the similar translation-based method **Visual and Linguistic Representation Model (VALR)** [123] combines linguistic representations and visual representations of entities to learn entity embeddings.

*4.1.1.4. Multi-model atrribute information.* Some literatures attempt to learn KG embedding utilizing multi-model data including various factors: text, images, numerical values, categorical values, and etc.

**VALR** [123] considers multi-modal information for learning entity embeddings. Based on the work of [122], apart from the entity images, VALR integrates linguistic representation of entities, it builds the score function upon the foundations of TransE and designs it as the sum of sub-energy functions that leverage both multi-modal (visual and linguistic) and structural information, which may properly learn new multi-modal representations. VALR builds an easily extensible neural network architecture to train the model.

**Multi-modal knowledge base embeddings (MKBE)** [124] focuses on the multimodel relational data for link prediction task, introduced a novel link prediction model named multi-modal knowledge base embeddings (MKBE). MKBE consists of an encoder and a decoder, the encoder employs multiple different neural structures according to the different multimodel evidence

**Table 11**
Characteristics of KGC methods using nodes' attributes information.

| Model | Highlights | Nodes information | Jointly learning expression | Datasets |
|---|---|---|---|---|
| KGC using numeric atrribute information | | | | |
| KBLRN [118] | End-to-end jointly training model; multi-task learning; feature types-combining approach | Numerical attributes | $L = -\sum_{(h,r,t)\in\mathcal{T}} logp((h,r,t)\|\theta_1,\ldots,\theta_n)$[a] | FB15k-num, FB15k-237-num |
| MTKGNN [119] | End-to-end multi-task NN | Numeric attributes | $L_{attr} = L_{head} + L_{tail}$<br>$L_{head} = MSE(g_h(a_i),(a_i)^*)$<br>$L_{tail} = MSE(g_t(a_j),(a_j)^*)$ | YG24K, FB28K |
| TransEA [120] | TransE + numerical attributes | Numeric attributes | $L = (1-\alpha)\cdot L_{TransE} + \alpha \cdot L_A$<br>$L_{TransE}$: TransE loss; $L_A$: attribute loss | YG58K, FB15K |
| KGC using textual atrribute information | | | | |
| JointAs [15] | Jointly neural network model | Node's name, anchors | $L = L_K + L_T + L_A$<br>$L_K$: KGC loss;<br>$L_T$: Text model loss;<br>$L_A$: Alignment loss | Freebase |
| JointTs [121] | Replaces anchors in JointAs with text description | Node's name | $L = L_K + L_T + L_{A_T}$<br>$L_{A_T}$: text description-aware Alignment loss | Freebase |
| KGC using image atrribute information | | | | |
| IKRL [122] | Neural image encoder; translation-based decoder; attention mechanism | Image attributes | $s(h,r,t) = s_{SS} + s_{SI} + s_{IS} + s_{II}$<br>$s_{XY} = \|h_X + r - t_Y\|$,<br>$S(I)$: structure(image)-based representations | WN9-IMG |
| KGC using multi-model atrribute information | | | | |
| VALR [123] | Linguistic embeddings; neural network architecture; multi-model additional energy function | Text attributes, image attributes | $s(h,r,t) = s_S + s_{M1} + s_{M2} + s_{SM} + s_{MS}$<br>$s_S = \|h_S + r_S - t_S\|$,<br>$s_{M1} = \|h_M + r_S - t_M\|$,<br>$s_{M2} = \|(h_M + h_S) + r_S - (t_M + t_S)\|$,<br>$s_{SM} = \|h_S + r_S - t_M\|$,<br>$s_{MS} = \|h_M + r_S - t_S\|$<br>$S/M$: structure/multi-modal representations | FB-IMG, WN9-IMG |
| MKBE [124] | Feature type specific encoders/decoders; DistMult/ConvE; multi-modal KGs modeling; VGG pretrained network on ImageNet | Text attributes, images attributes, numeric attributes | $L = \sum_{(h,r)}\sum_t l_t^{h,r} log(p_t^{h,r}) + (1-l_t^{h,r}) log(1-p_t^{h,r})$<br>$p_t^{h,r} = \sigma s(h,r,t)$,<br>$l_t^{h,r}$: a binary label | YAGO-10 |
| MMKG [125] | Relational reasoning across different entities and images | Numeric attributes, images attributes | $L = -\sum_{(h,r,t)\in\mathcal{T}} logp((h,r,t)\|\theta_1,\ldots,\theta_n)$[a] | DB15K, YAGO15K, FB15K |
| LiteralE [126] | End-to-end universal extension module | Numeric attributes, text attributes | $s_X(h,r,t) \rightarrow s_X(g(h,l_h),r,g(t,l_t))$<br>$g()$: a gated function; $X$: specific KGE models | FB15k, FB15k-237, YAGO-10 |

[a] $\theta_i$: the parameters of individual model.

types to embed multimodel data that link prediction task used, while different neural decoders distinguished by missing multi-model relational data types use the learned entity embeddings to achieve multimodel attributes recovery. Experiments demonstrate the effectiveness of MKBE based on both the Distmult and the ConvE scoring functions on two new datasets generated by extending the exiting datasets, YAGO-10 and MovieLens-100k. This paper proves a variety of relational data types can provide abundant evidence for link prediction task, and made a successful attempt to use the multimodel information in a unified model.

**Multi-Modal Knowledge Graphs (MMKG)** [125] is a visual-relational resource collection of three KGs for KGC, which is constructed relying on FB15K and is enriched with numeric literals and image information. MMKG extends KBLRN [118] by adding image information to this learning framework.

**LiteralE** [126] also attaches importance to rich literal attributes of nodes, especially non-discrete values, and learns entity embeddings by incorporating attribute information via a portable parameterized function. Although LiteralE plays emphasis on numerical attributes, it points out that textual or image feature can fit the incorporation principle as well for jointly learning literal-enriched embedding. Additionally, LiteralE explores the effect of utilizing multiple attribute features among relational data, and constructs a new large-scale dataset for multi-modal KGC based on Freebase.

*4.1.1.5. Discussion on KGC methods using node's attribute information.* ***Datasets:*** From Table 12, we dabble in several datasets which are rich in attribute data. Liu et al. [125] introduce a collection of Dbpedia15K, YAGO15K, and FB15K that contain both numerical features and image links for all entities in KGs. The WN9-IMG dataset in [122] contains a subset of WordNet synsets, which are linked according to a pre-defined set of linguistic relations, e.g. hypernym. Based on Freebase, Mousselly-Sergieh et al. [123] develop a novel large-scale dataset, FB-IMG, for multimodal KGC. The FB-IMG dataset can better resemble the characteristics of real KG because it has a much larger number of relations, entities, and triples compared to WN9-IMG (cf. Table 12). Besides, Garcia-Duran et al. [118] create two special datasets referred to as FB15k-num and FB15k-237-num by adding numerical features on the original KGC benchmark FB15K.

***Jointly learning:*** In this end, we discuss the general situation of jointly learning in these KGC works using attribute data. We can easily find that the attribute of nodes is used less singly. On the contrary, they tend to be combined and interacted with each other as mentioned by multi-modal data. The above-mentioned IKRL [122] is a classic multimodal data used method that incorporates both visual and structural information. Since the node's attribute features are a kind of additional diversified information, such KGC works tend to jointly learn original structure models and additional attribute models. As a consequential result, they

**Table 12**
Statistics of several nodes' attributes datasets.

| Dataset | Entity | Relation | #Rel KG | #Numeral | #Images |
|---|---|---|---|---|---|
| Dbpedia15K | 14 777 | 279 | 99 028 | 46 121 | 12 841 |
| YAGO15K | 15 283 | 32 | 122 886 | 48 405 | 11 194 |
| Dataset | Entity | Relation | #Train | #Valid | #Test |
| WN9-IMG | 6555 | 9 | 11 741 | 1337 | 1319 |
| FB-IMG | 11 757 | 1231 | 285 850 | 29 580 | 34 863 |
| FB15k-num | 14 951 | 1345 | 483 142 | 5156 | 6012 |
| FB15k-237-num | 14 541 | 237 | 272 115 | 1058 | 1215 |

are likely to design a combined scoring system or optimize a joint loss objective. We uniformity summarize their loss functions in Table 11, from which we can easily conclude that they mostly develop their loss objective or energy function in a composition form, usually extend the original definition of triple energy (distance energy or similarity energy and so on) to consider the new multimodal representations.

#### 4.1.2. *Entity-related information*

Entity-related information includes *entity types* and semantic *hierarchical taxonomic* information of entities. We uniformly summary this part of works in Table 13.

In KGs, entity types are the side information that commonly exists and dictates whether some entities are legitimate arguments of a given predicate [127]. For instance, suppose the interest relation is *bornin*, which denotes the birth location of a person, naturally we expect the asked candidate entity pairs are *person-location* type to own this relation. What is more, entity type information is readily available and gives assistance in avoiding unnecessary computation led by incompatible entity-relation.

##### 4.1.2.1. *Entity types information.*

**TRESCAL** [127] is a conventional tensor decomposition approach, it regards relation extraction (RE) as a KGC task, indicating that entity type information relates to KG can provide additional valuable relational domain knowledge for KGC. The novel paradigm focuses on the relevance between RE and KGC, which enables the learning process to spend less time than other traditional approaches (i.e, TransE and RESCAL).

**TCRL** [128] considers entity types as hard constraints in latent variable models for KGs. With type information, the type-constraint model selects negative samples according to entity and relation types. However, the type information is not explicitly encoded into KG representations, and their method does not consider the hierarchical structure of entity types. Moreover, hard constraints may have issues with noises and incompleteness in type information, which is pretty common in real-world KGs.

**TransT** [129] combines structure information with type information and takes into account the ambiguity of entities, it dynamically generates multiple semantic vectors according to the context of the entity. Moreover, TransT constructs relation types relying on entity types, also add similarity between relative entities and relations as the prior knowledge to guide KG embedding algorithm.

**Feature-Rich Networks (FRNs)** [130] also leverages entity type information and additional textual evidence for KGC on the FB15k-237 dataset. They learn embeddings for manifold types, along with entities and relations from noisy resources. Their method to incorporate with type information has a (small) contribution towards improving performance in predicting unseen facts.

**Ontology-Based Deep Learning Approach (OBDL)** [131] recently adds ontological information (where ontological information refers to those type hierarchy features of a given entity, they are shared among similar entities) into KG embedding in a deep learning framework, which enables it to predict unseen facts in the training process (referred as fresh entities).

##### 4.1.2.2. *Entity hierarchy taxonomic information.*

The entity hierarchy taxonomic information is a hierarchy of entity categories. Categories in different levels reflect the similarity in different granularities. Each vertex is assigned a path (from the root to a leaf) in the hierarchy [129]. The neighborhood structure of a vertex is usually closely related to an underlying hierarchical taxonomy: the vertices are associated with successively broader categories that can be organized hierarchically [134]. The hierarchical taxonomic of entity allows the information to flow between vertices via their common categories so that it provides an effective mechanism for alleviating data scarcity.

**Entity Hierarchy Embedding (EHE)** [132] learns distribution representation for entity hierarchy by designing a distance matrix for each entity node. The aggregated metrics encode entity hierarchical information to obtain hierarchy embeddings, which can significantly capture abundant semantic for KGC.

**Semantically Smooth Embedding (SSE)** [133] takes advantage of additional semantic information, e.g., entity semantic categories, and restrains the geometric structure of the embedding space to be consistent with observed facts. They semantically smooth under a smoothness assumption that leverages two various learning algorithms Laplacian Eigenmaps [137] and Locally Linear Embedding [138]. On the one hand, the proposed smoothness assumption is portable and well-adapted in a wide variety of KG embedding models. On the other hand, SSE regularization terms can be constructed by other useful additional features in other possible embedding tasks.

**NetHiex** [134] is a network embedding algorithm that incorporates hierarchical taxonomy into network embeddings thus modeling hierarchical taxonomy aware entity embeddings. NetHiex uses a nonparametric probabilistic framework to search the most plausible hierarchical taxonomy according to the nested Chinese restaurant process, and then recover the network structure from network embeddings according to the Bernoulli distribution. This framework is implemented by an efficient EM algorithm with linear time complexity of each iteration, which makes NetHiex a scalable model. Besides, NetHiex learns an entity representation consists of multiple components that are associated with the entity's categories of diverse granularity, which alleviates data scarcity with effect.

**Guided Tensor Factorization Model (GTF)** [135] pays attention to more challenging completion of generics KGs. It applies a knowledge guided TF method considering the taxonomy hierarchy of entities and the corresponding relation schema, appending guided quantification constraints and schema consistency on triple facts.

**SimplE+** [136] also concentrates on background taxonomic information about knowledge facts. [136] points out that the existing fully expressive TF models are less expressive in utilizing taxonomic features, which is very instructive to guide LP. Considering the taxonomic information in forms of subclass and sub-property, SimplE+ advances SimplE [44] by adding non-negativity constraints to further inject subsumption content into the original LP method, which is a simple but effective attempt for KGC.

#### 4.1.3. *Relation-related information*

The majority of facts in KGs possess comprehensive semantic relations, which often include transitivity and symmetry properties, as well as the type hierarchical characteristic. Take the

**Table 13**
Summarization of introduced KGC methods using Entity-related information.

| Model | Technology | Entity information | Dataset |
|---|---|---|---|
| KGC using entity types information: | | | |
| TRESCAL [127] | a. base on RESCAL;<br>b. low computational complexity;<br>c. entity-type constraints | Entity type information; textual data | NELL |
| TCRL [128] | a. entity-type constraint model;<br>b. under closed-world assumption | Entity type information | Dbpedia-Music, FB-150k,YAGOc-195k |
| TransT [129] | a. dynamical multiple semantic vectors;<br>b. entities-relations similarity as prior knowledge | Structured information; entity type information | FB15k, WN18 |
| FRNs [130] | a. jointly modeling KGs and aligned text;<br>b. a composition and scoring function parameterized by a MLP | Entity type information; additional textual evidence | FB15k-237 |
| OBDL [131] | a. deep learning framework (NTN);<br>b. a new initialization method for KGE;<br>c. unseen entity prediction | Entity type hierarchy feature; ontological information | WordNet, Freebase |
| KGC using entity hierarchy taxonomic information: | | | |
| EHE [132] | a. distance matrix;<br>b. entity similarity measuring | Entity hierarchy information | Wikipedia snapshot |
| SSE [133] | Portable smoothness assumption:<br>a. Laplacian Eigenmaps<br>b. Locally Linear Embedding. | Entity semantic categories | NELL_L, NELL_S, NELL_N 186 |
| NetHiex [134] | a. a nonparametric probabilistic framework<br>b. nested Chinese restaurant process<br>c. EM algorithm | Hierarchical taxonomy information | BlogCatalog, PPI, Cora, Citeseer |
| GTF [135] | a. knowledge guided tensor factorization method;<br>b. guided quantification constraints;<br>c. imposing schema consistency | Entity taxonomy hierarchy; corresponding relation schema | Animals, Science |
| SimplE+ [136] | SimplE with non-negativity constraints | Subclass and subproperty taxonomic information of entity | WN19, FB15K, Sport, Location |

**Table 14**
Characteristics of introduced KGC methods using relation-related information.

| Model | Technologies | Relation-related information | Datasets[a] |
|---|---|---|---|
| TranSparse [139] | Complex relation-related transformation matrix | Heterogeneous and imbalance characteristics of relations | LP: FB15k, WN18, FB15k-237, WN18RR |
| AEM [140] | Relation weight | Asymmetrical and imbalance characteristics of relations | LP: WN18, FB15K;<br>TC: WN11, FB13, FB15K |
| Trans-HRS [112] | TransE/TransH/DistMult + HRS structure | Three-layer HRS structure information of relations | LP: FB15K, WN18 |
| On2Vec [141] | a. Component-specific Model encoder<br>b. Hierarchy Model | Hierarchical relations | RP: DB3.6K,CN30K, YG15K,YG60K |
| JOINTAe [142] | a. autoencoder<br>b. considers relation inverse characteristic<br>c. based on RESAC<br>d. relations composition in [143] | Compositional information of relations | LP: WN18, FB15k, WN18RR, FB15k-237 |
| Riemannian-TransE [144] | a. multi-relational graph embedding<br>b. Non-Euclidean Space modeling<br>c. based on TransE<br>d. non-Euclidean manifold | Multi-relational (hypernym and synonym) information of relations | TP: WN11, FB13 |
| TRE [145] | Relation inference based on the triangle pattern of knowledge base | Entity-independent transitive relation patterns | LP: FB15K, WN18,<br>RP: FB15K, WN18, DBP |

[a]'LP', 'RP' and 'TC' respectively refer to Link Prediction task, Relation Prediction task and Triple Classification task.

transitivity relation pattern as an example in Fig. 16, three entities $a$, $b$, $c$ are connected through relations $r_1$, $r_2$, $r_3$. If these three relations, no matter connected with which entity, often appear together, then we can treat that as a transitivity relation pattern, this pattern can be applied to an incomplete triangle to predict the missing relation between entities $d$ and $f$. Here we set out the applications of relation-related information among KGC methods.

Table 14 gives a systematical summary for the KGC studies using relation-related features.

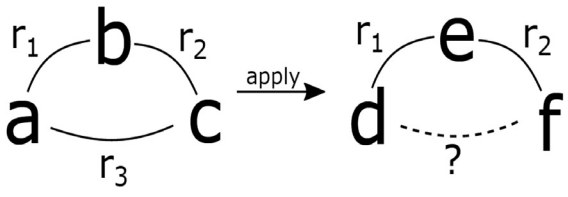



**Fig. 16.** An example of transitivity relation pattern excerpting from [145].

*4.1.3.1. Methods.* **TranSparse** [139] Since *relations* in KGs are *heterogeneous*, and *imbalance*, TranSparse is proposed to address this issue by introducing complex relation-related transformation

matrix [99]. TranSparse believes that the transformation matrix should reflect the heterogeneity and unbalance of entity pair, it changes the transformation matrix into two adaptive sparse transfer matrices corresponding to the head entities and tail entities.

**Asymmetrical Embedding Model (AEM)** [140] pays attention to the *asymmetrical* and *imbalanced* characteristics of relations and conducts supplement research for KGC. AEM weights each entity vector by corresponding relation vectors according to the role of this entity in a triple. Significantly, AEM weights each dimension of the entity vectors, whose impact is similar to TransA [108], can accurately represent the latent properties of entities and relations.

**Trans-HRS** [112] learns knowledge representations by exploiting the three-layer HRS relation structure information as an extension of existing KG embedding models TransE, TransH, and DistMult.

**On2Vec** [141] is a translation-based model for dealing with specialized semantic relation facts in ontology graphs, technically models comprehensive relations in terms of various relation properties, such as transitivity, symmetry, and hierarchical. On2Vec consists of two sub-structures, one of them is a *Component-Specific Model* which is charges for preserving relation properties, the another named the *Hierarchy Model* aims to handle hierarchy relations specifically. On2Vec is an effective ontology relation prediction model which can nicely operate ontology population by exploiting those properties or sub-properties of semantic relations properly.

**JOINTAe** [142] explores a dimension reduction technique jointly training with an auto-encoder, to better learn low dimension interpretable relations, especially for compositional constraints. As for the compositional constraints on relations, JOINTAe adapts mentioned approach in [143]. Moreover, JOINTAe considers inverse relations in the training procedure and amends the score function based on RESACL.

**Riemannian TransE** Recently, *Multi-Relation Embedding* is a popular hot-spot to KGC. At this basis, Riemannian TransE [144] exploits a non-Euclidean manifold in a *Non-Euclidean Space* to operate multi-relational graph embedding. It allots particular dissimilarity criteria to each relation according to the distance in *Non-Euclidean space*, replaces parallel vector fields in TransE with vector fields with an attractive point to get better embedding results, and inherits TransE's characteristic of low complexity parameter at the same time.

**TRE** [145] is invented for completing sparse KGs, which effectively leverages entity-independent transitive relation patterns to find the patterns for infrequent entities. Though TRE briefly learns representations of relations instead of entity representation learning as previous KGC methods, it gets high effectiveness in predicting missing facts with low computational expensive but high interpretability.

*4.1.3.2. Discussion on relation-related information for KGC.* Why are the relation characteristics evidence becoming popular in the KGC field? *Firstly*, the relation patterns are independent of entities, so that it can predict missing relations of uncommon entities, which is helpful to alleviate the sparsity problem by improving the completion of infrequent entities through frequent relation patterns [145], the conventional embedding method is hard to achieve it. *Secondly*, compared with the embedding methods, the computational cost of identifying relation patterns is lower [146], because it does not need to learn the embedded representation of individual entities. Last but not least, relation patterns are highly interpretable.

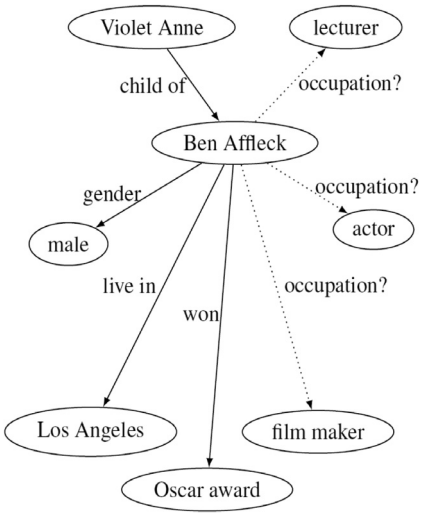



**Fig. 17.** A neighborhood subgraph example of a KG [147].

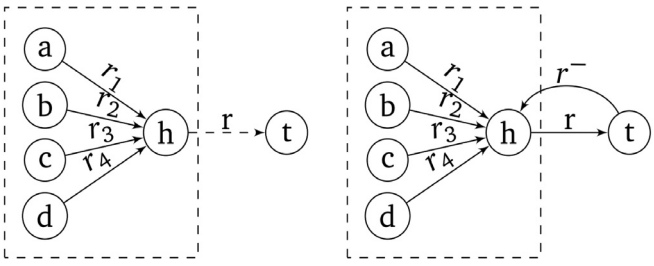



**Fig. 18.** A general process using neighbor information for KGC [149]. The subgraphs $G$ in the dashed boxes is the neighborhood graph of triple $(h, r, t)$, and triples in $G$ are represented by a solid edge, and triples (e.g., candidate triples) not in $G$ are represented by a dashed edge. Note that any "head prediction" problem $(?, r, t)$ can be converted to the "tail prediction" problem $(t, r^{-}, ?)$.

*4.1.4. Neighborhood information*

The neighbors of entity are new kinds of additional information containing both semantic and topological features, which could be exploited for KGC. For instance, consider a KG fragment example given in Fig. 17 [147]. If we know that *BenAffleck* has won an *Oscaraward* and *BenAffleck* lives in *LosAngeles*, we prefer to predict that *BenAffleck* is an *actor* or a *filmmaker*, rather than a *teacher* or a *doctor*. Further, if we additionally know that Ben Affleck's gender is *male* then there is a higher probability for him to be a *filmmaker*. Mostly, the neighbors are utilized to form a relation-specific mixture representation as an entity vector to assist in entity learning, the general thought is shown in Fig. 18. Although the well-known Graph Convolution Networks (GCNs) [70,82] and Graph Attention Networks (GATNs) [148] also learn neighborhood-based representations of nodes, they suffered from expensive computation and did not learn sub-optimal query-dependent compositions of the neighborhood. We make a presentation for entity neighbor information aware KGC methods except for GCNs or GATNs. Table 15 exhibits general KGC methods using neighbor information.

*4.1.4.1. Aggregating neighbors with attention mechanism.* **A2N** [150] Opposed to the early method **NMM** [147] (where NMM incorporates TransE with neighbors information to crystallize into a TransE-MRR version but only learns a fixed mixture over neighbors), A2N embeds query-dependent entities with corresponding neighbors into the same space via *bi-linear attention* on the graph neighborhood of an entity, to generate neighborhood informed

**Table 15**
Characteristics of introduced KGC methods using neighbor information.

| Model | Technology | Additional information | Datasets |
|---|---|---|---|
| Aggregating neighbors with attention mechanism: | | | |
| A2N [150] | DistMult + attention scoring | Neighbor structure information | FB15K-237, WN18RR |
| LENA [149] | Windowed Attentions;<br>Cross-Window Pooling | Neighbor structure information | FB15K, FB15K-237, WN18, WN18RR |
| LAN [151] | Logic Attention Network;<br>end-to-end model:<br>Encoder: LAN, Decoder: TransE | Relation-level information;<br>neighbor-level information | Subject-10 and Object-10 in FB15K |
| G2SKGEatt [152] | Graph2Seq network;<br>attention mechanism;<br>end-to-end model:<br>Encoder: Graph2Seq, Decoder: ConvE | Neighbor structure information | FB15K, FB15K-237, WN18, WN18RR |
| KBAT [91] | Generalized GAT;<br>end-to-end model:<br>Encoder: KBAT, Decoder: ConvKB | Entity's multi-hop neighborhood | FB15K-237, WN18RR, NELL-995, Kinship |
| RGHAT [153] | GNN;<br>hierarchical attention mechanism;<br>end-to-end model:<br>Encoder: RGHAT, Decoder: ConvE | Entity's multi-hop neighborhood | FB15K, WN18, FB15K-237, WN18RR |
| Other technologies for KGC using neighbor information: | | | |
| GMatching [154] | Permutation-invariant network;<br>LSTM;<br>end-to-end model:<br>Encoder: neighbor encoder,<br>Decoder: matching processor | Neighbor structure information | NELL-One, Wiki-One |
| GMUC [155] | Gaussian metric learning;<br>few-shot UKGC;<br>end-to-end model:<br>Encoder: Gaussian neighbor encoder,<br>Decoder: LSTM-based matching networks | Neighbor structure information | NL27K-N0, NL27K-N1, NL27K-N2 and NL27K-N3 |
| NKGE [31] | Dynamic Memory Network;<br>gating mechanism;<br>end-to-end model:<br>Encoder: DMN, Decoder: TransE/ConvE | Structure representation;<br>neighbor representation | FB15K, FB15K-237, WN18, WN18RR |
| CACL [93] | Contextual information collection;<br>context-aware convolutional | Multi-hop neighborhoods structure information | FB13, FB15K, FB15K-237, WN18RR |
| OTE [115] | RotatE;<br>orthogonal transforms | Graph contexts representations | FB15K-237, WN18RR |
| CNNIM [156] | Concepts of Nearest Neighbors;<br>Dempster–Shafer theory | Neighbors information | FB15k-237, JF17k, Mondial |
| CAFE [157] | Neighborhood-aware feature set;<br>feature grouping technique | Neighborhood-aware features | FB13-A-10, WN11-AR-10, WN18-AR-10, NELL-AR-10 |

representation. For the attention scoring, A2N uses the DistMult function to project the neighbors in the same space as the target entities.

Inspired by the thought of aggregating neighbors with attention mechanism in [150], there has generated a lot of closely relevant studies:

**Termed locality-expanded neural embedding with attention (LENA)** [149] is introduced to filter out irrelevant messages among neighborhoods with the support of an attentional setting. This work indicates that the KG embedding relying on even sufficient structure information is deficient since the graph data tend to be heterogeneous. Therefore, LENA emphasizes that information involved in the graph neighborhood of an entity plays a great role in KG embedding in especially with complex heterogeneous graphs.

**Logic Attention Network (LAN)** [151] is a novel KG-specific neighborhood aggregator that equips attention mechanism to aggregate neighbors in a weighted combination manner. This work designs two mechanisms for modeling relation-level and neighbor-level information respective from coarse to fine: *Logic Rule Mechanism* and *Neural Network Mechanism*, in the end, a *double-view attention* is employed to incorporate these two weighting mechanisms together in measuring the importance of neighbors. LAN meets all three significant properties: *Permutation Invariant*, *Redundancy Aware* and *Query Relation Aware*.

**G2SKGEatt** [152] develops a information fusion mechanism *Graph2Seq* to learn embeddings that fuses sub-graph structure information of entities in KG. To make fusion more meaningful, G2SKGEatt formulates an attention mechanism for fusion. The 1-N scoring strategy proposed by ConvE [33] is used to speed up the training and evaluation process.

**KBAT** [91] is also an attention-based KGE model which captures both entity and relation features in the multi-hop neighborhood of given entity. KBAT uses ConvKB [68] as its decoder module and specifically caters to the relation prediction (RP) task. **RGHAT** [153] designs a novel hierarchical attention mechanism to compute different weights for different neighboring relations and entities. Consider that the importance of different relations differ greatly in indicating an entity and to highlight the importance of different neighboring entities under the same relation, the hierarchical attention mechanism including two-level attention mechanisms: a relation-level attention and an entity-level attention. The relation-level attention firstly indicate an entity by computing the weights for different neighboring relations of it, then the entity-level attention computes the attention scores for different neighboring entities under each relation. Finally, each

entity aggregates information and gets updated from its neighborhood based on the hierarchical attentions. RGHAT can utilize the neighborhood information of an entity more effectively with the use of hierarchical attention mechanism.

*4.1.4.2. Other technologies for KGC using neighborhood information.* Some other works concern different technologies to make use of the neighborhood information.

**GMatching** [154] takes those one-shot relations which usually contain valuable information and make up a large proportion of KGs into consideration, and introduces an intelligent solution to the problem of KG sparsity caused by long-tail relations. GMatching learns knowledge from one-shot relations to solve the sparsity issue and further avoid retraining the embedding models when new relations are added into existing KGs. This model consists of two components: a neighbor encoder and a matching processor, which are responsible for encoding the local graph structure to represent entities and calculating the similarity of two entity pairs respectively.

**GMUC** [155] is a Gaussian metric learning-based method that aims to complete few-shot uncertain knowledge graphs (UKGs, such as NELL and Probase, which model the uncertainty as confidence scores related to facts). As the first work to study the few-shot uncertain knowledge graph completion (UKGC) problem, GMUC uses a Gaussian neighbor encoder to learn the Gaussian-based representation of relations and entities. Then a Gaussian matching function conducted by the LSTM-based matching networks is applied to calculate the similarity metric. The matching similarity can be further used to predict missing facts and their confidence scores. GMUC can effectively capture uncertain semantic information by employing the Gaussian-based encoder and the metric matching function.

**NKGE** [31] uses a *End-to-End Memory Networks (MemN2N)* based *Dynamic Memory Network (DMN)* encoder [158] to extract information from entity neighbors, and a gating mechanism is utilized to integrate the structure representations and neighbor representations. Based on TransE [11] and ConvE [33] respectively, NKGE designs two kinds of architectures to combine structure representation and neighbor representation. Experimental results show that the TransE-based model outperforms many existing translation methods, and the ConvE-based model gets state-of-the-art metrics on most experimental datasets.

**Context-aware convolutional learning (CACL)** [93] is a study of exploring the connection modes between entities using their neighbor contexts information, which facilitates the learning of entity and relation embeddings via convoluting deep learning techniques directly using the connection modes contained in each multi-hop neighborhood.

**Orthogonal transform embedding (OTE)** [115] advances RotatE [110] in two ways: (1) leveraging orthogonal transforms [116] to extend RotatE from 2D complex domain to high dimension space in order to raise modeling ability, and (2) OTE takes account of the neighbor contexts information, effectively learns entity embeddings by fusing relative graph contexts representations. Experiments contrast that with RotatE, R-GCN and A2N revealing great availability of OTE.

**Concepts of Nearest Neighbors-based Inference Model (CNNIM)** [156] performs LP recognizing similar entities among common graph patterns by the use of *Concepts of Nearest Neighbors* [159], from where *Dempster–Shafer theory* [160] is adapted to draw inferences. CNNIM only spends time in the inference step because it abolishes training time-wasting to keep a form of instance-based learning. The application of graph pattern instead of numerical distances makes the proposed method interpretable.

**CAFE** [157] completes KGs using the sets of neighborhood-aware features to evaluate whether a candidate triple could be added into KGs. The proposed set of features helps to transform triples in the KG into feature vectors which are further labeled and grouped for training neural prediction models for each relation. These models help to discern between correct triples that should be added to the KG, and incorrect ones that should be disregarded. Note that since CAFE exploits the highly connected nature of KGs rather than requiring pre-processing of the KG, it is especially suitable for ever-growing KGs and dense KGs.

*4.1.4.3. Discussion on KGC models using neighborhood information.* From the above introduction and comparison about neighbor-used KGC literature, we further make a basic discussion and analysis as follows:
**(1)** To better obtain the neighborhood graph information, we need to select an appropriate fusion strategy to collect useful surrounding neighbor contexts.
**(2)** we find that most models tend to use the encoder-to-decoder (*end-to-end*) architecture when *learning neighbor information* for KGC, in other words, the neighbor learning part is portable which could be applied to various KGE models such as translation models (e.g., TransE [11], TransH [15], TransR [12]) and Bilinear models [42,161]. We give a presentation about these end-to-end structures in Table 15, and show them in Fig. 19 to illustrate this intuition.
**(3)** The embedding parameters for every entity-relation pair may be prohibitively large when the learned neighbor fusion is fixed, which led to the adaptable mixture methods based on the different query are more and more popular over recent years.

### *4.1.5. Relational path information*

In KGs, there are substantial *multiple-step relation paths* between entities indicating their semantic relations, these relation paths reflect complicated inference patterns among relations in KGs [12], it helps to promote the rise of the path-based relation inference, one of the most important approaches to KGC task [168]. We generally list these path-based KGC works in Table 16.

***multi-hop KGC (mh-KGC)***: We refer to the definition in [163], the mh-KGC aims at performing KGC based on existing relation paths. For example in Fig. 20, for the relation path *Microsoft* → *IsBasedIn* → *Seattle* → *IsLocatedIn* → *Washington* → *IsLocatedIn* → *United States* (as the blue lines shows), the task is to predict whether (or what) there exists direct relations that connects $h$ and $t$; i.e., (*Microsoft*, *CountryOfHQ*, *United States*) in this case. This kind of reasoning lets us infer new or missing facts from KGs. Sometimes there can exist multiple long paths between two entities, thus in this scene, the target relation may be inferrable from not only one path.

*4.1.5.1. Multi-hop KGC using atomic-path features.* **Path Ranking Algorithm (PRA)** [164] is the first work that emerges as a promising method for learning inference paths in large KGs, it uses *random walks* to generate relation paths between given entity pairs by depth-first search processes. The obtained paths then are further encoded as relational features and combined with a logistic regression model to learn a binary log-linear classifier to decide whether the given query relation exists between the entity pairs.

However, millions of distinct paths in a single classifier are generated by the PRA method, it may supervene with **feature explosion** problem because each path is treated as an atomic feature, which makes the atomic-path idea difficult to be adopted by KGs with increasing relation types [166]. Additionally, since

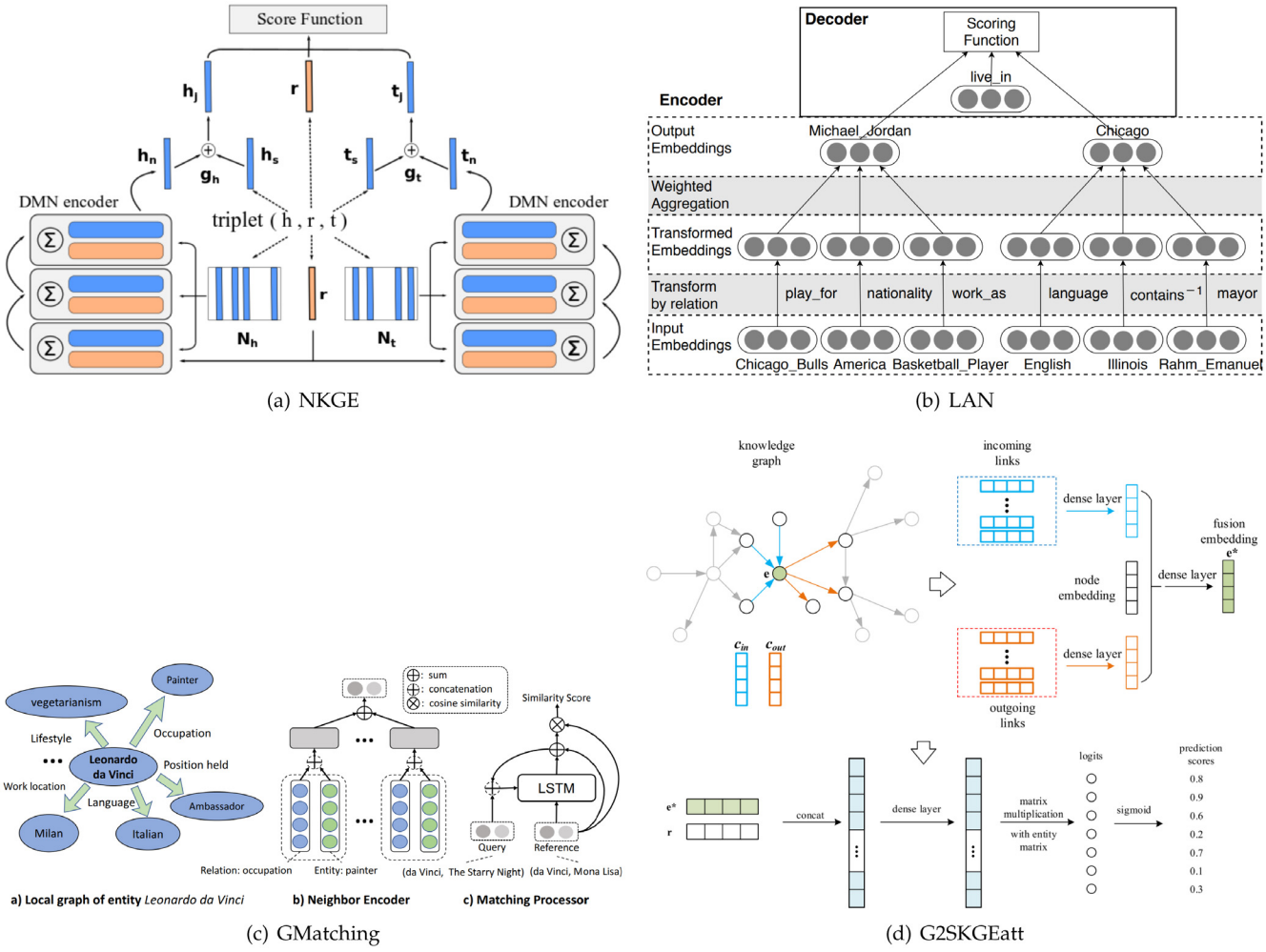



**Fig. 19.** Several end-to-end KGC models using neighborhood information.
*Source:* These figures are extracted from [31,152,154,162].

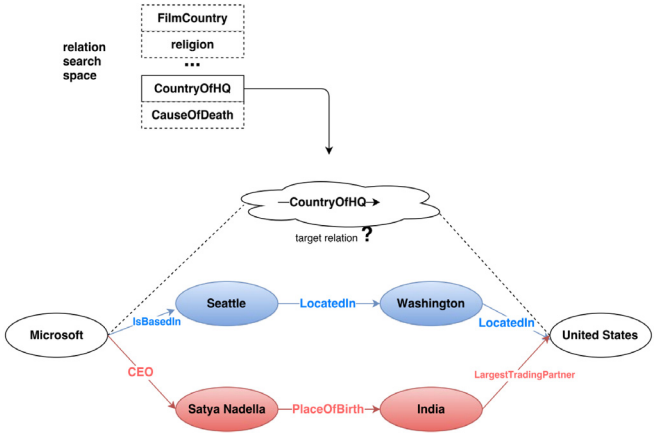



**Fig. 20.** An illustration of knowledge reasoning over paths [163].

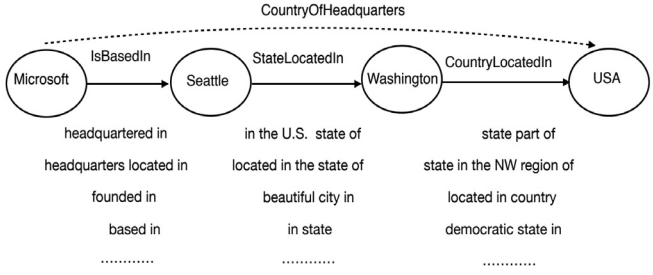



**Fig. 21.** A sketch map to path feature explosion [166].

PRA must compute random walk probabilities associated with each path type and entity pair, resulting in proportional computation amount increase with the path number and path length. The feature explosion issue is shown in Fig. 21.

Therefore, **new versions of PRA** [165,177,178] try to develop a series of more efficient and more expressive models related to PRA. Both the first two use pre-trained vector representations of relations to alleviate the feature explosion problem [166]: In the work of [177], many paths are folded by clustering the paths according to the embedding degree of the relation between paths, then it uses cluster ID to replace the original relation type. The work [178] maps unseen paths to nearby paths seen at training time, where the nearness is measured using the embeddings. The work [165] defines a simpler feature matrix generation algorithm called **subgraph feature extraction (SFE)**, it conducts a more exhaustive search, a breadth-first search instead of random walks, to characterize the local graph. Without the random walk probabilities computation, SFE can extract much more expressive features, including features that are not representable as paths in

**Table 16**
Characteristics of introduced KGC methods using relational path information.

| Model | Technology | Additional information | Path selection strategy | Datasets |
|---|---|---|---|---|
| Mh-KGC using atomic-path features: | | | | |
| PRA [164] | Random walks | Atomic path feature | A single path | NELL |
| SFE [165] | Breadth-first search | Atomic path feature | A single path | NELL |
| Non-atomic multi-hop reasoning: | | | | |
| PATH-RNN [166] | RNN + PRA, zero-shot reasoning | Non-atomic and compositional path feature, arbitrary-length path | Max pooling | Freebase + ClueWeb |
| Trans-COMP [143] | Compositional training, path compositional regularizer | Non-atomic path feature | A single path | WordNet, Freebase |
| Path-augmented translation models: | | | | |
| PTransE [12] | PCRA[a] + TransE + path scoring | Non-atomic path feature | PCRA | FB15K |
| RTransE [39] | TransE + regularization composition | Non-atomic path feature | Focused on "unambiguous" paths: $\ell_1$: 1-to-1 or 1-to-many relations, $\ell_2$: 1-to-1 or many-to-1 relations | FB15K, FAMILY |
| PTransD [117] | Path-augmented TransD | Path | PCRA | FB15K |
| Modeling paths using neural networks: | | | | |
| Single-Model [167] | Path-RNN; Shared Parameter Architecture | Path, intermediate nodes, entity-types | Scoring pooling: Top-K, Average and LogSumExp | Freebase + ClueWeb |
| APCM [168] | RNN + Attention | Path, entity type | Attentive Path Combination | FC17 |
| IRNs [169] | Shared memory + controller | Path, structured relation information | Controller determines the length of paths | WN18, FB15K |
| ROHP [163] | Three ROPs architectures: GRUs | Path | Arbitrary-length path | Freebase + ClueWeb |
| PRCTA [170] | RNN; constrained type attention; relation-specific type constraints | Path, entity and relation types | Path-level attention | Freebase + ClueWeb |
| mh-RGAN [96] | RNN reasoning models + GAN | Non-atomic path feature | Generator G of GAN | WordNet, FreeBase |
| Combine path information with type information: | | | | |
| All-Paths [171] | Dynamic programming, considers intermediate nodes | Path, relation types | Dynamic programming | NCI-PID and WordNet. |
| RPE [172] | Relation-specific type constraints; path-specific type constraints | Path, relation type | Reliable relation paths-selection strategy | LP: FB15K; TC: FB15K, FB13, WN11 |
| APM [173] | Abstract graph + path | Abstract paths, strongly typed relations | Paths in abstract graph | Freebase, NELL |
| Leveraging order information in paths: | | | | |
| OPTransE [174] | TransE + Ordered Relation Paths | Path, relation orders | Path fusion: two layer pooling strategy | WN18 and FB15K |
| PRANN [175] | CNN + BiLSTM | Path + entities/relations orders, entity types | Path-level Attention | NELL995, FB15k-237, Countries, Kinship |

[a]'PCRA': path-constraint resource allocation algorithm [176].

the graph at all — but the core mechanism of these three works continues to be a classifier based on atomic-path features. Besides, neither one can perform zero-shot learning because there must be a classifier for each predicted relation type in their approaches.

*4.1.5.2. Non-atomic multi-hop reasoning.* Some works explore to utilize path information as non-atomic features during a KGC procedure.

**PATH-RNN** [166] can not only jointly reason on the path, but also deduce into the vector embedded space to reason on the elements of paths in a non-atomic and combinatorial manner. Using recursive neural networks (RNNs) [179] to recursively apply a composite function to describe the semantics of latent relations over arbitrary length paths (in Fig. 22(a)), PATH-RNN finally produces a homologous path-vector after browsing a path. PATH-RNN can infer from the paths not seen in the training during the testing process, and can also deduce the relations that do not exist in the KGs.

**TransE-COMP** [143] suggests a new compositional training objective that dramatically improves the path modeling ability of various traditional KGC models to answer path queries. This technique is applicable to a broad class of combinable models that include the bilinear model [13] and TransE [11], i.e, the score function:

$$s(s/r, t) = \mathbb{M}(\mathbb{T}_r(x_s), x_t)$$

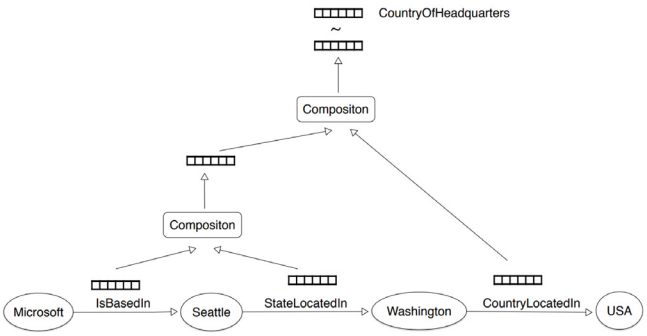
represents a combinatorial form where the traversal operator $\mathbb{T}_r(x_s)$ means a path query $(x_s, r, ?)$ following $\mathbb{T}_r : \mathbb{R}^d \to \mathbb{R}^d$, and operator $\mathbb{M}$ illustrates the incorporable model's score operation follows $\mathbb{M} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, for example, when cooperates with TransE, the traversal operator becomes to $\mathbb{T}_r(x_s) = x_s + w_r$ and the score function then turns into:

$$s(s/r, t) = \mathbb{M}(T_r(x_s), x_s) = -\|T_r(x_s) - x_s\|_2^2$$

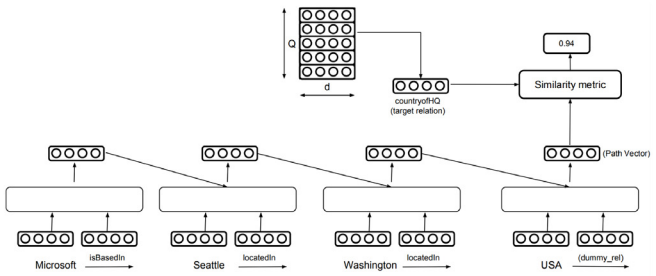
so that it can handle a path query $q = s/r_1/r_2/.../r_k$ by:

$$s(q, t) = -\|x_s + w_{r_1} + \cdots + w_{r_k} - x_t\|_2^2$$

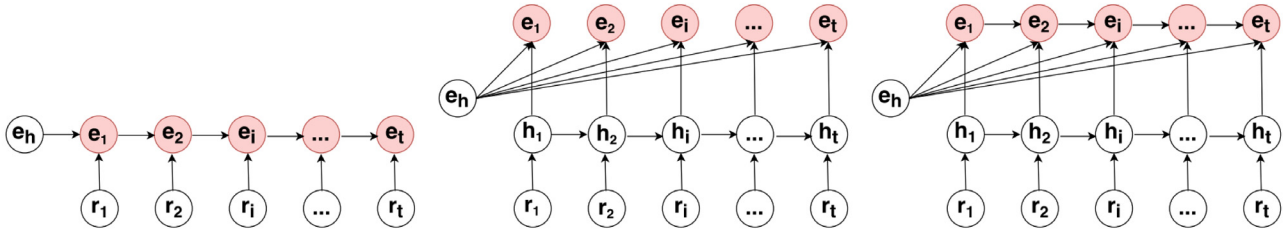
The compositional training is regarded as providing a new form of structural regularization for existing models since it substantially reduces cascading errors presented in the base vector space model.



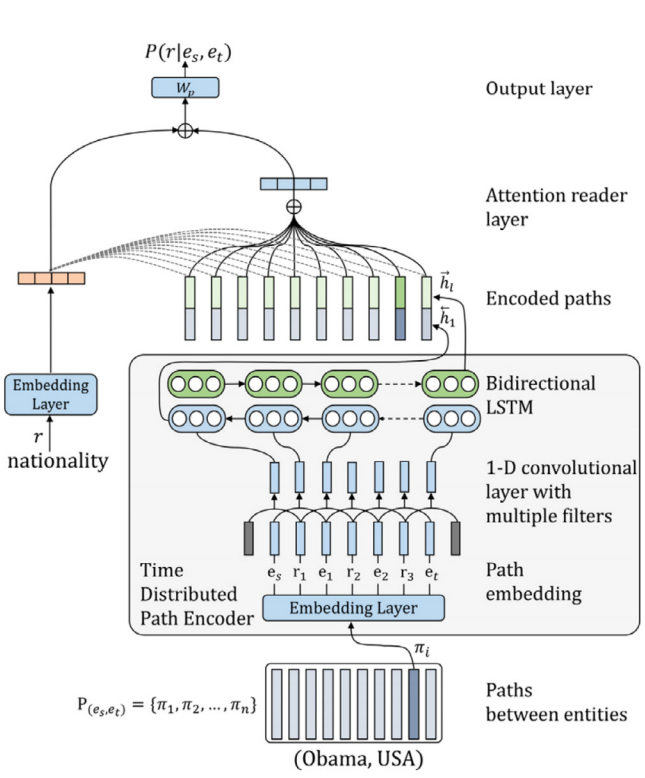

(a) PATH-RNN only models relations in a path.



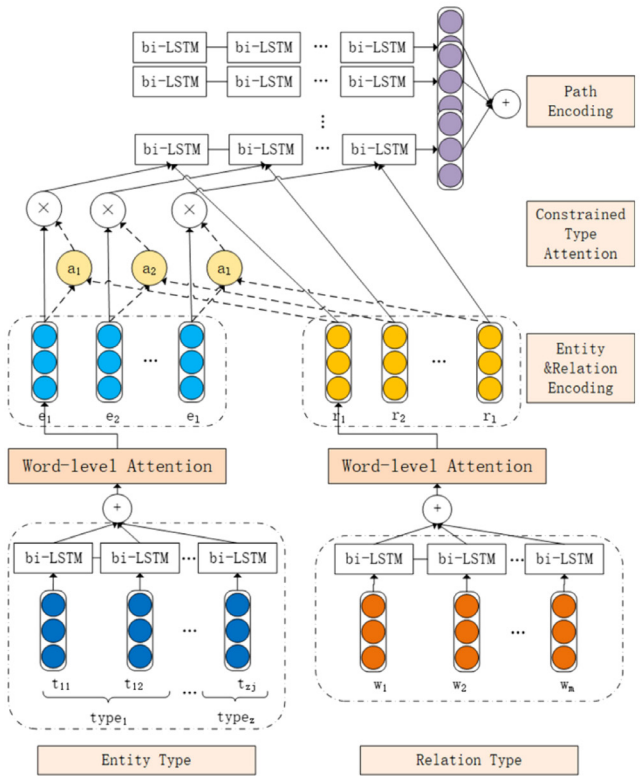

(b) Single-Model models both entities and relations in a path.



(c) Three ROP architectures in ROHP. They all consider both entities and relations in a path.




(d) PRANN uses Bi-LSTM and CNNs to encode path representation.




(e) PRCTA uses multi-layer Bi-LSTMs for learning path information.

**Fig. 22.** Several RNN-structure KGC models using path information.
*Source:* These figures are from [96,163,166–168,170,175].

*4.1.5.3. Path-augmented translation models.* The path-augmented translation methods, which introduce multi-step path information into classical translation models, are developed.

**PTransE** [12] uses path information in its energy function as:

$$s(h, r, t) = E(h, r, t) + E(h, P, t)$$

which the latter item $E(h, P, t)$ models the inference correlations between relations with multi-step relation path triples. In PTranasE, relation paths $p \in P(h, t)$ are represented via semantic composition of relation embeddings, by perform *Addition*, *Multiplication* or *RNN* operation:

$$Addition: \ p = r_1 + \cdots + r_l$$

*Multiplication* : $p = r_1 \cdot \ldots \cdot r_l$

*RNN* : $c_1 = r_1, \ldots, \ p = c_n$

Simply put, PTransE doubles the number of edges in the KG by creating reverse relations for each existing relation. Then PTransE uses a *path-constraint resource allocation algorithm (PCRA)* [176] to select reliable input paths within a given length constraint.

**RTransE** [39] learns compositions of relations as sequences of translations in TransE by simply reasoning among paths, in this process, RTransE only considers a restricted set of paths of length two. This paper augments the training set with relevant examples of the above-mentioned compositions, and training so that sequences of translations lead to the desired result.

**PTransD** [117] is a path-augmented TransD, it thinks relation paths as translation between entities for KGC. Similar to TransD, PTransD considers entities and relations into different semantics spaces. PTransD uses two vectors to represent each entity and relations, where one of them represents the meaning of a(n) entity (relation), and another one is used to construct the dynamic mapping matrix.

*4.1.5.4. Modeling paths using neural networks.* We can see that *neural network* is handy in *modeling path*, especially the RNN application lines:

**Single-Model** [167] Based on the PATH-RNN [166], Single-Model discusses path-based complex reasoning methods extended by RNN and jointly reasoning with within-path relations, entities, and entity types in the paths.

**Attentive Path Combination Model (APCM)** [168] first generates path representations using an RNN architecture, then it assigns discriminative weights to each path representations to form the representation of entity pair, finally, a dot-product operation between the entity pair representation and the query relation representation is designed to compute the score of a candidate query relation, so that it allows entity pair to get representation with respect to query relations in a dynamic manner.

**Implicitly ReasonNets (IRNs)** [169] designs a network architecture, which performs multi-hop reasoning in vector space based on shared memory. The key highlight is the employment of shared memory that intelligently saves relevant large-scale structured relations information in an implicit manner, thus it can avoid explicit human-designed inference. IRNs reasons according to a controller to stipulate the inference step during the whole inference procedure simultaneously gets proper interaction with shared memory. This work performs an excellent function on KGC about complex relations.

**Recurrent one-hop predictor Model (ROHP)** [163] explores three ROHP architectures with the capability of modeling KG paths of arbitrary lengths by using recurrent neural networks (GRUs [180]) to predict entities in the path step by step for multi-hop KG reasoning.

**Path-based Reasoning with Constrained Type Attention (PRCTA)** equipped with a constrained type attention mechanism for multi-hop path reasoning [170]. On the one hand, PRCTA encodes type words of both entities and relations to extract abundant semantic information by which partly improves the sparsity issue, on the other hand, for reducing the impact of noisy entity types, constrained type attention is designed to softly select contributing entity types among all the types of a certain entity in various scenarios, meanwhile, relation-specific type constraints are made full use for enhancing entity encoding.

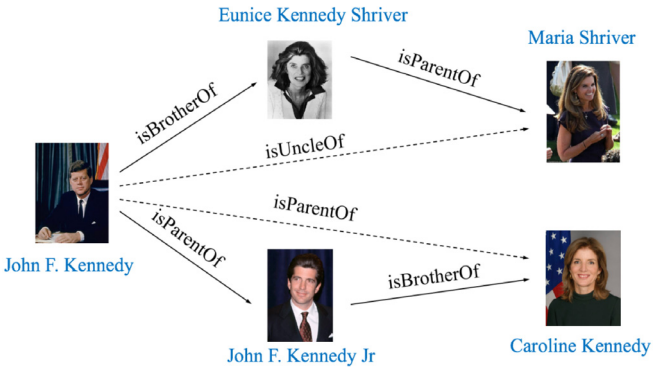



**Fig. 23.** Example of the meaning change when the order of relations is altered.

Final path encoding leverages path-level attention to combine useful paths and produces path representations.

We collect some representative structures of methods that *model path information* for KGC using *RNNs* in Fig. 22. There are other path-based KGC models using other neural network frameworks:

**Multi-hop Relation GAN (mh-RGAN)** [96] considers multi-hop (mh) reasoning over KGs with a generative adversarial network (GAN) instead of training RNN reasoning models. The mh-RGAN consists of two antagonistic components: a generator $G$ with respect to composing a *mh-RP*, and a discriminator $D$ tasked with distinguishing real paths from the fake paths.

*4.1.5.5. Combining path information with type information.* Some methods consider type information of entities and relations when modeling path representations, such as [167,168], and [170].

**Relational Path Embedding model (RPE)** Lin et al. [172] extend the relation specific type constraint to the new path specific type constraint, both two type constraints can be seamlessly incorporated into RPE to improve the prediction quality. In addition, RPE takes full advantage of the semantics of the relation path to explicitly model KGs. Using the composite path projection, RPE can embed each entity into the proposed path space to better handle the relations with multiple mapping characteristics.

**Abstract Path Model (APM)** [173] focuses on the generation of abstract graph depending on the strongly typed relations and then develops a traversal algorithm for mining abstract paths in the produced intensional graph. Those abstract paths tend to contain more potential patterns to execute KG tasks such as LP. The proposed abstract graph drastically reduces the original graph size, making it becomes more tractable to process various graphs.

*4.1.5.6. Leveraging order information in paths.* The *order* of relations and entities in paths is also important for reasoning. As Fig. 23 shows, the meaning will change when the order of relations is altered [174].

**OPTransE** [174] attaches importance to the order information of relations in relation paths via projecting each relation's head entity and tail entity into different vector spaces respectively. To capture the complex and nonlinear features hidden in the paths, OPTransE designs a multi-flow of min-pooling layers. It was experimentally validated that OPTransE performs well in LP task, directly mirroring the vital role of relation order information in relation paths for KGC.

**Path-based Reasoning with Attention-aware Neural Network (PRANN)** [175] also uses the ordering of the local features to learn

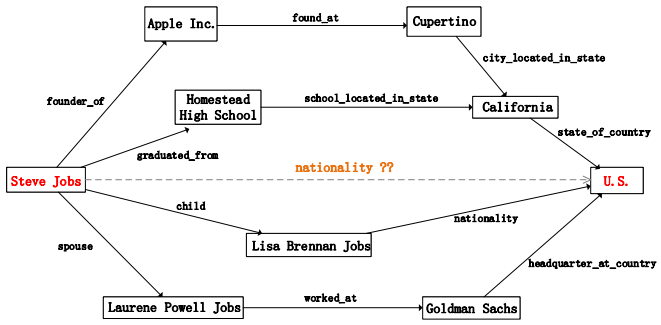



**Fig. 24.** An illustration that relations between an entity pair can be inferred by considering information available in multiple paths collectively [168].

about the entities and the relation orderings of each path. PRANN explores a novel path encoding framework including a bidirectional long short-term memory (BiLSTM) followed by a CNN and fusion paths works by a path-level attention mechanism. The structure of PRANN can result in an efficient path representation which leads to excellent LP results. Considering multi-step reasoning on paths, this paper further designs a memory module for storing the time-distributed encoded path, which can repeatedly extract path features to enhance the prediction performance. The author indicates that it is necessary to develop external memory storage for storing overall paths between every entity pairs to meet the increased needs of entity pairs in current KGs.

*4.1.5.7. Path selection strategy.* As we mentioned above, the reasoning path set of an entity pair usually contains more than one path, so that when we conduct KGC, we should ponder over how we make use of those multiple paths, *should we choose one path or consider all the paths*? And *if we choose only one path, what selection strategies do we need to follow*? Thus it is a noteworthy issue that how to formulate an appropriate method of finding the most informative path under the mh-KGC task. For an example in Fig. 24, none of the four paths directly contains evidence that the *nationality* of *Steve Jobs* is *U.S.*, but when we jointly consider these paths together, we will get much more information to support the fact (*Steve Jobs*, *nationality*, *U.S.*).

**Trans-COMP** [143] models only a single path between an entity pair, moreover, **PATH-RNN** [166] uses Max operator to select the path with the largest predictability at each training/testing iteration [168]. The previous KGC methods [12,143] using relation paths neither take account of intermediate nodes nor model all the relation paths since the computational expense is too expensive to enumerate all possible paths, especially in graphs containing text [171]. Whereby, **All-Paths** [171] improves upon them by additionally modeling the intermediate entities in the path and modeling multiple paths. For a given path type referred to in the PRUNED-PATHS approach, All-Paths uses dynamic programming to exactly build the sums of all path representations over node sequences.

However, in their method they have to store scores for intermediate path length for all entity pairs, making it prohibitive to be used in large-scale KGs. **Single-Model** [167] is presented to improve the performance of Path-RNN [166]. Rather than the "max" pooling, Single-Model leverages various score pooling strategy: *Top-K*, *Average* and *LogSumExp*, and among which the *LogSumExp* pooling performs best. *LogSumExp* pooling is deemed to play the same role as attention mechanism and can integrate every path in trainable proportion.

Unfortunately, none of these methods can simulate scenarios in which relations can be inferred only by considering multiple information paths [168]. On the other hand, each of these path combination operations works at a score-level, and has its deficiency:

(1) *Max*: Only one path is used for reasoning, while all other information paths are ignored.

(2) *Average*: As is often the case that the path sets connecting an entity pair are very large, and only a few paths may be helpful for reasoning. Therefore, the model is often affected by noise.

(3) *Top-K*: Different entity pairs may have different optimal K values. Moreover, not all Top-k paths contribute equally to reasoning.

(4) *LogSumExp*: This is a smooth approximation of the "Max" operator, which can be seen as 'soft' attention, but cannot effectively integrate evidence from multiple paths.

The unsatisfactory path combination situation promotes a series of effective approaches that begin to spring up. For an entity pair and the set of relation paths between them, **Attentive Path Combination Model (APCM)** [168] assigns discriminative weights to each path to further combine these weighted path representations into an entity pair level representation, **Path-based Reasoning with Constrained Type Attention (PRCTA)** [170] uses a constrained type attention mechanism for multi-hop path reasoning, which mainly considers to alleviate the negative influence of graph sparsity and entity type noise when conducting the reasoning procedure.

**Table 17**
Statistics of path-based KGC datasets FC (Freebase + ClueWeb) and FC17.

| Datasets | FC | FC17 |
|---|---|---|
| Entities | 18M | 3.31M |
| Freebase triples | 40M | 35M |
| ClueWeb triples | 12M | 104M |
| Relations | 25,994 | 23 612 |
| Relation types tested | 46 | 46 |
| Avg. paths/relation | 2.3M | 3.19M |
| Avg. training positive/query relation | – | 6621 |
| Avg. training negative/query relation | – | 6622 |
| Avg. training facts/relation | 6638 | – |
| Avg. positive test instances/relation | 3492 | 3516 |
| Avg. negative test instances/relation | 43,160 | 43 777 |

*4.1.5.8. Performance analysis about path-based KGC.* ***Datasets:*** We introduce a famous dataset, **Freebase + ClueWeb** (called **FC** for convenience) [166], for path reasoning over KGs. FC is a large-scale dataset of over 52 million triples, it involves preprocessing for multi-hop KGC (mh-KGC). The dataset is built from the combination of Freebase [5] and Google's entity linking in ClueWeb [181], which contains entities and relations from Freebase and is enriched with ClueWeb text. FC is widely applied by several path-based KGC methods [163,166,167,170], rather than Gardner's 1000 distinct paths per relation type, it have over 2 million [166]. FC can be downloaded from http://iesl.cs.umass.edu/downloads/inferencerules/release.tar.gz. **FC17** is a more recently released version to FC, in which the number of paths between an entity pair ranges drastically from 1 to 900 or more, so the robust of methods in comparison can be better evaluated with this dataset. Compared with the older version, FC17 has far more ClueWeb triples. Statistics of both FC and FC17 is listed in Table 17.

***Performance Comparison:*** We report the existing published experimental performance of several path-based KGC models according to different evaluation datasets in Table 18, Tables 19 and 20. By the way, we also give some analysis about presented results.

(1) **On FC and FC17 datasets:** Table 18 shows experimental results of path-based KGC methods on FC and FC17 datasets. On FC, it can be observed that: overall, PRCTA outperforms all the

**Table 18**
Experimental results of path-based KGC methods on FC and FC17 datasets. MAP on FC and FC17 are reported by [170] and [168], respectively. Best results are in bold.

| Model | FC | FC17 |
|---|---|---|
| | MAP (%) | MAP (%) |
| PRA [164] | 64.43 | 55.48 |
| Path-RNN [166] | 68.43 | 52.37 |
| Single-Model [167] | 70.11 | 58.28 |
| Single-Model + Type [167] | 73.26 | 63.88 |
| Att-Model [168] | 71.21 | 59.89 |
| Att-Model + Type [168] | 73.42 | **65.03** |
| PRCTA [170] | **76.44** | – |

other methods in the table, which indicates the effectiveness of leveraging textual types and entity type discrimination for mh-KGC as PRCTA does on the task of multi-hop reasoning. Besides, as mentioned in [170], textual types and all attention mechanisms contribute to the ablation test, except that, conducting entity type discrimination with constrained type attention also provides a greater performance boosting. Notably, in terms of noise reduction, the result shows that the attention mechanisms adopted in PRCTA can significantly reduce noise. Specifically, the word-level attention alleviates the representation sparseness by reducing noise in the whole type context, while constrained type attention further reduces noisy entity types and thus alleviates inefficiency on entities with a large number of types [170]. On FC17, there lacks of relevant data of PRCTA.

Over FC17 dataset, the model "Att-Model + Type" [168] achieves the best performance. Not only the 'AttModel', using relations in the path, outperforms other methods that also use relation only, but also the proposed method 'Att-Model+Types', further considering the entities in the path by adding their types into RNN modeling, still achieves considerable improvements than its main opponent 'Single-Model+Types'. All the comparison results above-mentioned indicate the importance of proper attention mechanisms.

(2) **On NELL995, FB15K-237, Kinship and Countries datasets:** Further, we report the data comparison on NELL995 and FB15k-237 in Table 19 and observe that PRANN [175] can more accurately predict missing links on the large datasets compared with other methods. Note that when it compared with the existing non-path models to verify the competitiveness of the approach in the KGC task, PRANN have achieved comparable results to the state-of-the-art methods across all evaluation metrics, in especial the *MRR* and *Hits@k* scores of MINERVA, a path-based KGC model which is similar to that of PRANN. It is notable that on the KG such as FB15k-237 with a large number of diverse relations, PRANN performs better compared to other models in the experiment. On the contrary, MINERVA [63] was giving slightly better results on the dataset with a fewer number of relations, such as the Countries dataset. From Table 19 we can observe the experimental results on the small datasets, Kinship, and Countries. PRANN also achieves excellent results on the Kinship dataset because this dataset was created to evaluate the reasoning ability of logic rule learning systems with more predictable paths compared to other datasets. However, on the Countries dataset, PRANN shows lower results compared to MINERVA, relevant explanation is because the number of training triples in the Countries dataset is too small to efficiently train our model.

(3) **On WN18 and FB15K datasets:** Table 20 presents the experimental results on WN18 and FB15K, numbers in bold mean the best results among all methods. The evaluation results of baselines are from their original work, and "–" in the table means there is no reported result in prior work. According to the table, IRN significantly outperforms other baselines, regardless of whether other approaches use additional information or not. Specifically, on FB15k, the *Hit*@10 of IRN surpasses all previous results by 5.7%. From Table 20 we could observe that: (a) Both PTransE and RPE perform better than their basic model TransE and TransR, which indicates that additional information from relation paths between entity pairs is helpful for link prediction. Also, OPTransE outperforms baselines which do not take relation paths into consideration in most cases. These results demonstrate the effectiveness of taking advantage of the path features inside of KGs in OPTransE. (b) Except for the *Hits*@10 scores of IRN, OPTransE almost performs best on all metrics compared to previous path-based models like RTransE, PTransE, PaSKoGE, and RPE, which implies that the order of relations in paths is of great importance for knowledge reasoning, and learning representations of ordered relation paths can significantly improve the accuracy of link prediction [174]. Moreover, the proposed pooling strategy which aims to extract nonlinear features from different relation paths also contributes to the improvements of performance.

*4.1.5.9. Discussion on relational path information in KGC.* **1. Limitation of path information:** In multi-hop KGC (mh-KGC), the path extracted from KGs mainly stores the structural information of facts, which is inherently incomplete. This incompleteness can affect the process in different ways, e.g. it leads to representations for nodes with few connections that are not very informative, it can miss relevant patterns/paths (or derive misleading patterns/paths) [173]. The limited information will lead to the representation sparseness of entities and relations, resulting in low discrimination for intermediate nodes, which constitutes a potential obstacle to the improvement of mh-KGC performance. Therefore, it is necessary to consider other information to assist reasoning, such as the semantic information of nodes in the path (e.g., textual type attributes, entity or relation order information, et al.). Intuitively, incorporating knowledge from textual sources by initializing the entity embeddings with a distributional representation of entities [182] could improve path-based relation reasoning results further.

**2. Neglection on entity/relation types in mh-KGC:** Although previous works have introduced entities and relations types into relational path reasoning tasks, they only consider single type entities while actually, entities have more than one type, especially in different contexts, the same entities often have different types and semantics. Additionally, they do not distinguish entity types in different triples which may pose noisy entity types to limit the final performance.

**3. More efficient attention mechanism:** More flexible and effective attention mechanism over mh-KGC tasks need to be explored. For example, previous methods often applied a similar approach using the dot product to measure the match between weighted path vectors and a target relation, although calculating the dot product attention is faster and space-efficient, in some cases, more intelligent handling technologies such as additional scaling factors are needed to compute the correct attention weights. For example, in [175], an additive attention function using a feed-forward network that scales well to smaller values are applied to act attention mechanism, it exhibits better performance compared to the dot product and efficiently scales to large values [183]. What is more, it is fully differentiable and trained with standard back-propagation.

**4. More efficient path encoder and more proper training objective:** Path reasoning in KG is still in continuous development, especially with the emergence of various coding structures, such as Bert, XLNet, etc. We can try to use more effective encoders to encode path features. In addition, when combined with the traditional methods, we can learn from previous experience (e.g., [166]

**Table 19**
Experimental results of path-based KGC methods on NELL995, FB15k-237, Kinship and Countries datasets. The public performance data in this table comes from [175]. Best results are in bold.

| Model | NELL995 | | | FB15k-237 | | | Kinship | | | Countries | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | MRR | Hits@1 | Hits@3 | MRR | Hits@1 | Hits@3 | MRR | Hits@1 | Hits@3 |
| PRA [164] | 0.696 | 0.637 | 0.747 | 0.412 | 0.322 | 0.331 | 0.799 | 0.699 | 0.896 | 0.739 | 0.577 | 0.9 |
| Single-Model [167] | 0.859 | 0.788 | 0.914 | 0.575 | 0.512 | 0.567 | 0.804 | 0.814 | 0.885 | 0.941 | 0.918 | 0.956 |
| MINERVA [63] | 0.879 | 0.813 | 0.931 | 0.615 | 0.49 | 0.659 | 0.824 | 0.71 | 0.937 | **0.96** | **0.925** | **0.995** |
| PRANN [175] | **0.898** | **0.838** | **0.951** | **0.66** | **0.544** | **0.708** | **0.952** | **0.918** | **0.984** | 0.947 | 0.916 | 0.986 |

**Table 20**
Link prediction results of path-based KGC methods on WN18 and FB15k datasets. All the data in the table comes from [174]. Best results are in bold.

| Model | WN18 | | FB15k | |
|---|---|---|---|---|
| | Hits@10 | MR | Hits@10 | MR |
| RTransE [39] | – | – | 0.762 | 50 |
| PTransE (ADD, 2-step) [174] | 0.927 | 221 | 0.834 | 54 |
| PTransE (MUL, 2-step) [174] | 0.909 | 230 | 0.777 | 67 |
| PTransE (ADD, 3-step) [12] | 0.942 | 219 | 0.846 | 58 |
| PTransD (ADD,2-step) [117] | – | – | 0.925 | 21 |
| RPE (ACOM) [172] | – | – | 0.855 | 41 |
| RPE (MCOM) [172] | – | – | 0.817 | 43 |
| IRN [169] | 0.953 | 249 | **0.927** | 38 |
| OPTransE [174] | **0.957** | **199** | 0.899 | **33** |

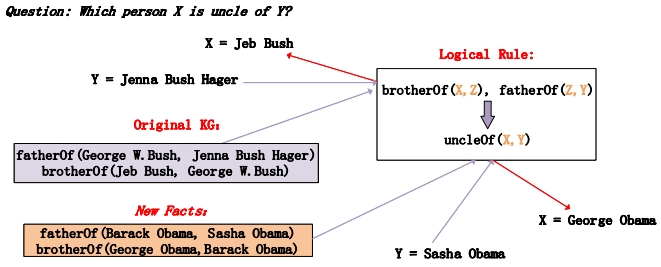



**Fig. 25.** Example of rules for KGC. The picture refers to [184].

has shown that the non-linear composition function outperforms linear functions (as used by them) for relation prediction tasks) to select and expand the appropriate linear or non-linear model.

### 4.2. External extra information outside KGs

In this section we comb KGC studies which exploit external information and mainly include two aspects: rule-based KGC in Section 4.2.1 and third-party data source-auxiliary KGC in Section 4.2.2.

#### 4.2.1. Rule-based KGC

*Logical rules* in KGs are non-negligible in that they can provide us expert and declarative information for KGC, they have been demonstrated to play a pivotal role in inference [185–187], and hence are of critical importance to KGC. In this section we give a systemic introduction of KGC tasks working with various rules, we also list a summary table for rule-based KGC methods as shown in Table 21.

*4.2.1.1. Introduction of logical rules.* An example of KGC with logical rules is shown in Fig. 25. From a novel perspective [192], KGs can be regards as a collection of conceptual knowledge, which can be represented as a set of rules like $BornIn(x, y) \wedge Country(y, z) \rightarrow Nationality(x, z)$, meaning that if person $x$ was born in city $y$ and $y$ is just right in country $z$, then $x$ is a citizen of $z$. Rules are explicit knowledge (compared to a neural network), thus reasonable use of logic rules is of great significance to handle problems in KGs. Rule-based KGC allows knowledge transfer for a specific domain by exploiting rules about the relevant domain of expertise, which

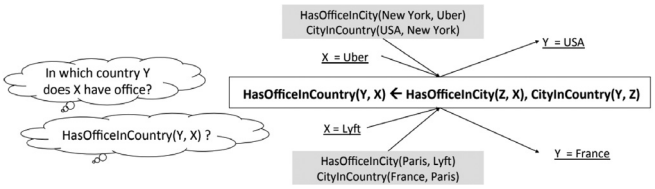



**Fig. 26.** An example of the robustness of rule reasoning shown in [191].

makes rule-based reasoning achieve high accuracy. Moreover, logical rules are *interpretable* enough to provide insight into the results of reasoning, and in many cases, this excellent character will lead to the robustness of the KGC transfer task. For example, conducting rule reasoning over an increasing KG can avoid parts of retraining work due to the addition of new nodes, which is more adaptable than models modeled for certain entities within a specific KG. Consider the scenario in Fig. 26, when we add some new facts about more companies or locations to this KG, the rules with respect to '*HasOfficeInCountry*' will still be usefully accurate without retraining. The same might not be workable for methods that learn embeddings for specific KG entities, as is done in TransE. In other words, *logical rule-based learning can be applied to those "zero-shot" entities that cannot be seen during training.*

The rules are manually or automatically constructed as various logic formulas, each formula learns a weight by sampling or counting grounding from existing KGs. These weighted formulas are viewed as the long-range interactions across several relations [185]. Manual rules are not suitable for large-scale KGs, on the other hand, it is hard to cover all rules in the specific domain KG by hand. Recently, rule mining has become a hot research topic, since it can automatically induce logical rules from ground facts, i.e., captures co-occurrences of frequent patterns in KGs to determine logical rules [207,209] in a machine-readable format.

*4.2.1.2. Definition about logical rules based KGC.* Formulaically, the KGC over rules we consider here consists of a query, an entity tail that the query is about, and an entity head that is the answer to the query [191]. The goal is to retrieve a ranked list of entities based on the query such that the desired answer (i.e., head) is ranked as high as possible.

**Formulation of Logical Rules:** In terms of first-order logic [210, 211], given a logical rule, it is first instantiated with concrete entities in the vocabulary $E$, resulting in a set of ground rules. Suppose $X$ is a countable set of variables, $C$ is a countable set of constants. A rule is of the form $head \leftarrow body$ as follows formula, where **head** $query(Y, X)$ is an atom over $R \cup X \cup C$ and **body** $R_n(Y, Z_n) \wedge \ldots \wedge R_1(Z_1, X)$ is a conjunction of positive or negative atoms over $R \cup X \cup C$.

$$query(Y, X) \leftarrow R_n(Y, Z_n) \wedge \ldots \wedge R_1(Z_1, X)$$

where $R_1, \ldots, R_n$ are relations in the KGs.

**Ground Atom & Rule's Grounding:** A triple $(e_i, r_k, e_j)$ can be taken as a ground atom which applies a relation $r_k$ to a pair of entities $e_i$ and $e_j$. When replacing all variables in a rule with concrete

**Table 21**
Characteristics of introduced KGC methods using rules.

| Model | Technology | Information | Rules | Dataset |
|---|---|---|---|---|
| Markov Logic Network (MLNs) series: | | | | |
| LRNNs [188] | Standard feed-forward NN,<br>weighted first-order rules | First-order rules | Function-free first-order logic | 78 RL benchmarks |
| MLN-based KGC [189] | Markov Logic Network,<br>mathematical axiom proof | Rules | – | – |
| ExpressGNN [190] | GNNs+MLN,<br>solving zero-shot problem,<br>EM algorithm,<br>mean-field approximation inference | Logic rules,<br>entity information | First order logical rules in MLN | FB15K-237 |
| End-to-end differentiable framework: | | | | |
| NuralLP [191] | TensorLog,<br>neural controller system (LSTM),<br>attention mechanism | First-order logical rules | Weighted chain-like logical rules | WN18,<br>FB15K,<br>FB15KSelected |
| RLvLR [192] | Improves NuralLP,<br>RESCAL,<br>target oriented sampling | First-order logical rules | CP rule: closed path rules | FB75K,<br>WikiData |
| NTPs [193] | RNN,<br>backward chaining algorithm,<br>RBF kernel,<br>ComplEx | First-order logical rules | Function-free first-order logic rules,<br>parameterized rules,<br>unify rule,<br>OR rule,<br>AND rule | Countries,<br>Kinship,<br>Nations,<br>UMLS |
| NTP2.0 [194] | NTPS,<br>max pooling strategy,<br>Hierarchical Navigable Small World (HNSW, a ANNS structure) | First-order logical rules | Function-free first-order logic rules;<br>parameterized rules;<br>unify rule;<br>OR rule;<br>AND rule | Countries,<br>Nations,<br>Kinship,<br>UMLS |
| DRUM [184] | Open World Assumption,<br>confidence score,<br>BIRNN | First-order logical rules | – | Family,<br>UMLS,<br>Kinship |
| Combining rule and embedding approach: | | | | |
| a. A shallow interaction: | | | | |
| r-KGE [185] | ILP,<br>RESCAL/TRESCAL/TransE,<br>four rules | Logical rules,<br>physical rules | Rule 1 (noisy observation);<br>Rule 2 (argument type expectation);<br>Rule 3 (at-most-one restraint);<br>Rule 4 (simple implication). | Location,<br>Sport |
| INS [195] | MLNs,<br>INS-ES,<br>TransE | Paths,<br>rules | path rules | FB15K |
| ProRR-MF [196] | ProPPR,<br>matrix factorization,<br>BPR loss | First-order logical rules | First-order logical rules | FB15K,<br>WordNet |
| b. Explore further combination style: | | | | |
| KALE [197] | Translation hypothesis,<br>t-norm fuzzy logic | Logic rules | Horn logical rules | WN18,<br>FB122 |
| Trans-rule [198] | TransE/TransH/TransR,<br>first-order logic space transformer,<br>encode the rules in vector space,<br>confidence score with a threshold | First-order logical rules | Inference rules;<br>transitivity rules;<br>antisymmetry rules | WN18,<br>FB166, FB15K |
| c. Iteration interactions: | | | | |
| RUGE [199] | Iterative model,<br>soft label prediction,<br>embedding rectification,<br>confidence score | Soft rules,<br>logic rules | Soft rules;<br>Horn logical rules | FB15K,<br>YAGO37 |
| ItRI [200] | KG embedding model,<br>iteratively learning,<br>pruning strategy,<br>hybrid rule confidence measures | Feedback information of KG embedding model text corpus,<br>non-monotonic rules | Non-monotonic rules with negated atoms;<br>non-monotonic rules with partially-grounded atoms | FB15K,<br>Wiki44K |
| IterE [201] | Iterative model,<br>embedding representation,<br>axiom induction,<br>axiom injection,<br>confidence score,<br>linear mapping hypothesis | OWL2 Language,<br>axioms information | 7 types of object property expression;<br>ontology axioms;<br>Horn logical rules | WN18-s,<br>WN18RR-s,<br>FB15k-s,<br>FB15k-237-s[a] |

(*continued on next page*)

**Table 21** (*continued*).

| Model | Technology | Information | Rules | Dataset |
|---|---|---|---|---|
| pLogicNet [202] | MLN,<br>EM algorithm,<br>amortized mean-field inference,<br>KG embedding model<br>(TransE/ComplEx) | First order logical rules | First order logical rules in MLN:<br>Composition Rules,<br>Inverse Rules,<br>Symmetric Rules,<br>Subrelation Rules | FB15k,<br>FB15k-237,<br>WN18,<br>WN18RR |
| Text + Logic rules: | | | | |
| FEP-AdTE [203] | Knowledge verification system,<br>TEP-based abductive text evidence,<br>remote supervision | Logical information,<br>text information | First-order logic rules | FGCN[b] KGs |
| Rules + Paths + Embedding approaches: | | | | |
| AnyBURL [204] | Aleph's bottom-up rule learning | Fuzzy rules,<br>uncertain rules,<br>path | Straight ground path rule:<br>$AC_1$ rules,<br>$AC_2$ rules,<br>C rules | FB15(k),<br>FB15-237,<br>WN18,<br>WN18RR,<br>YAGO03-10 |
| ELPKG [205] | KGE model,<br>breadth first search for paths,<br>probabilistic logical framework PSL | Path information,<br>logic rules | Probabilistic soft logic rules | YAGO,<br>NELL,<br>YAGO-50,<br>YAGO-rest |
| RPJE [206] | KGE model,<br>confidence score,<br>compositional representation learning | Logical rules,<br>path | Horn rules for two modules:<br>$R_1$: relation pairs association,<br>$R_2$: paths composition | FB15K,<br>FB15K-237,<br>WN18, NELL-995 |
| Filtering candidate triples: | | | | |
| AMIE+ [207] | Open-world assumption,<br>pruning operations | First-order logical rules | Single chain of variable rules<br>for Confidence approximation;<br>PCA;<br>typed rules | YAGO2 core,<br>YAGO2s,<br>DBpedia 2.0,<br>DBpedia 3.8,<br>Wikidata |
| CHAI [26] | Complex rules normalizer | Rules | Complex rules base on relation<br>domain and distance;<br>4 types filtering candidates criteria | FB13,<br>WN18,<br>NELL,<br>EPSRC |
| About evaluation: | | | | |
| RuleN [208] | An unify evaluation framework,<br>evaluated with AMIE model | Logical rules | Path rules $P_n$;<br>C rules | WN18,<br>FB15k,<br>FB15k-237 |

[a]'-s' means the '-sparse' series datasets.
[b]The 'FGCN' means Four Great Chinese Novels in China.

entities in KG, we get a grounding of the rule. A logical rule is encoded, for example, in the form of $\forall x, y : (x, r_s, y) \rightarrow (x, r_t, y)$, reflecting that any two entities linked by relation $r_s$ should also be linked by relation $r_t$ [197]. For example, a universally quantified rule $\forall x, y : (x, CapitalOf, y) \rightarrow (x, LocatedIn, y)$ might be instantiated with the concrete entities of *Paris* and *France*, forming the ground rule $(Paris, CapitalOf, France) \rightarrow (Paris, LocatedIn, France)$. A grounding with all triples existing in the KG is a support of this rule, and the ground rule can then be interpreted as a complex formula, constructed by combining ground atoms with logical connectives (e.g. $\wedge$ and $\rightarrow$).

**Logical Rules for KGC:** To reason over KGs, for each query it is usually interested in learning weighted chain-like rules of a form similar to stochastic logic programs [212]:

$$\alpha\ query(Y, X) \leftarrow R_n(Y, Z_n) \wedge \ldots \wedge R_1(Z_1, X)$$

where $\alpha \in [0, 1]$ means the confidence associated with this rule. In a generic sense, the inference procedure will define the score of each $y$ implies query $(y, x)$ as the sum of the confidence of the rules for the given entity $x$, and we will return a ranked list of entities where higher the score implies higher the ranking [191].

*4.2.1.3. Rule mining.* Inferring the missing facts among existing entities and relations in the growing KG by rule-based inference approaches has become a hot research topic, and how to learn the rules used for KGC also catches people's eye. There is a lot of literature that takes many interests in rule learning technology.

*(1). Inductive logic programming (ILP) for rule mining:*

**Inductive logic programming (ILP)** [213] (i.e. XAIL) is a type of classical statistical relational learning (SRL) [214], it proposes new logical rules and is commonly used to mine logical rules from KGs. Although ILP is a mature field, mining logical rules from KGs is difficult because of the open-world assumption KGs abide by, which means that absent information cannot be taken as counterexamples.

*(2). Markov Logic Networks (MLNs) and its extensions:*

Often the underlying logic is a probabilistic logic, such as **Markov Logic Networks (MLNs)** [215] or **ProPPR** [216]. The advantage of using probabilistic logic is that by equipping logical rules with probability, one can better statistically model complex and noisy data [191].

**MLNs** combines hard logic rules and probabilistic graphical models. The logic rules incorporate prior knowledge and allow MLNs to generalize in tasks with a small amount of labeled data, while the graphical model formalism provides a principled framework for dealing with uncertainty in data. However, inference in MLN is computationally intensive, typically exponential in the number of entities, limiting the real-world application of MLN. Also, logic rules can only cover a small part of the possible combinations of KG relations, hence limiting the application of models that are purely based on logic rules.

**Lifted Relational Neural Networks (LRNNs)** [188] is a lifted model that exploits weighted first-order rules and a set of relational facts work together for defining a standard feed-forward

neural network, where the weight of rules can be learned by stochastic gradient descent and it constructs a separate ground neural network for each example.

**A Theoretical study of MLN-based KGC (MLN-based KGC)** [189] explores the possibility that using MLN for KGC under the maximum likelihood estimation, it discusses the applicability of learning the weights of MLN from KGs in the case of missing data theoretically. In this work, it is proved by mathematical axiom proof that the original method, which takes the weight of MLNs learning on a given and incomplete KG as meaningful and correct (i.e. using the so-called closed world assumption), and predicts the learned MLN on the same open KGs to infer the missing facts is feasible. Based on the assumption that the missing triples are independent and have the same probability, this paper points out that the necessary condition for the original reasoning method is that the learning distribution represented by MLN should be as close as possible to the data generating distribution. In particular, maximizing the log-likelihood of training data should lead to maximizing the expected log-likelihood of the MLN model.

**ExpressGNN** [190] explores the combination of *MLNs* and popular *GNNs* in KGC field, and applies GNNs into MLN variational reasoning. It uses GNNs to explicitly capture the structural knowledge encoded in the KG to supplement the knowledge in the logic formula for predicting tasks. The compact GNNs allocates similar embedding to similar entities in the KG, while the expressible adjustable embedding provides additional model capacity to encode specific entity information outside the graph structure. ExpressGNN overcomes the scalability challenge of MLNs through efficient stochastic training algorithm, compact posterior parameterization and GNNs. A large number of experiments show that ExpressGNN can effectively carry out probabilistic logic reasoning, and make full use of the prior knowledge encoded within logic rules while meet data-driven requirement. It achieves a good balance between the representation ability and the simplicity of the model. In addition, it not only can solve the zero-shot problem, but also is a general enough which can balance the compactness and expressiveness of the model by adjusting the dimensions of GNNs and embedding.

*(3). End-to-end differentiable rule-based KGC methods:*

Based on these proposed basic rule-mining theories, a large amount of end-to-end differentiable rule-based KGC methods are developed according to these types of rules.

**Neural Logic Programming (NeuralLP)** [191] is an end-to-end differentiable framework which combines first-order rules inference and sparse matrix multiplication, thus it allow us learn parameters and structure of logical rules simultaneously. Additionally, this work establishes a neural controller system using attention mechanism to properly allot confidences to the logical rules in the semantic level, rather than merely "softly" generate approximate rules as mentioned in previous works [217–220], and the main function of the neural controller system is controlling the composition procedure of primitive differentiable operations of TensorLog [221] in the memory of LSTM to learn variable rule lengths.

**RLvLR** [192] aims at tackling the main challenges in the scalability of rule mining. Learning rules from KGs with the RESCAL embedding technique, RLvLR guides rules mining by exploring in predicates and arguments embedding space. A new target-oriented sampling method makes huge contributions to the scalability of RLvLR in inferring over large KGs, and the assessment work for candidate rules is handled by a suit of matrix operations referred to [207,209]. RLvLR shows a good performance both in the rules's quality and the system scalability compared with NeuralLP.

**NTPs** [193] is similar to NeuralLP, it focuses on the fusion of neural networks and rule inferring as well, but models neural networks following a backward chaining algorithm referred in *Prolog*, performing inference by recursively modeling transitivity relations between facts represented with vectors or tensors using RNN. NTPs makes full use of the similarity of similar subsymbolic representations in vector space to prove queries and induce function-free first-order logical rules, the learned rules are used to perform KGC even further. Although NTPs demonstrates better results than ComplEx in a majority of evaluation datasets, it has less scalability compared to NeuralLP as the limitation of computation complexity which considers all the proof paths for each given query.

**NTP 2.0** [194] whereby scales up NTPS to deal with real-world datasets cannot be handled before. After constructing the computation graph as same as NTPs, NTPs 2.0 employs a pooling strategy to only concentrate on the most promising proof paths, reducing the solutions searching procedure into an *Approximate Nearest Neighbor Search (ANNS)* problem using *Hierarchical Navigable Small World (HNSW)* [222,223].

**DRUM** [184], an extensible and differentiable first-order logic rule mining algorithm, further improves NeuralLP by learning the rule structure and the confidence score corresponding to the rule, and establishes a connection between each rule and the confidence score learned by tensor approximation, uses BIRNN to share useful information when learning rules. Although it makes up for the shortcomings of the previous inductive LP methods that have poor interpretability and cannot infer unknown entities, DRUM is still developed on the basis of the Open World Assumption of KGs and is limited to positive examples in training. In the following research, it is necessary to further explore improved DRUM methods suitable for negative sampling, or try to explore the same combination of representation learning and differential rule mining as methods [63,224].

*4.2.1.4. Combining rule-based KGC models with KGE models.* The *rule-based KGC models* provide interpretable reasoning and allows domain-specific knowledge transfer by using the rules about related professional fields. Compared to the representation model, the rule-based models do not need a lot of high-quality data but can achieve high accuracy and strong interpretability. However, they often face efficiency problems in large-scale search space; while the *embedding-based KGC models*, i.e., the KGE models, have higher scalability and efficiency but they have a flaw in dealing with sparse data due to their great dependence on data. We summarize the advantages and disadvantages of rule-based KGC and embedding-based KGC methods in a simplified table (Table 22). Therefore, there is no doubt that *combining rule-based reasoning with KGE models* to conduct KGC will be noteworthy. Please see Fig. 27 for a rough understanding of the researches of combining rule information with KGE models.

*(1). A shallow interaction:*

There are already some simple integrating works in the earlier attempts:

**r-KGE** [185] is one of these methods, it tries to utilize ILP to integrate the embedding model (three embedding models: RESCAL, TRESCAL, TransE) and four rules (including logical rules and physical rules): rules are expressed as constraints of the maximization problem, by which the size of embedding space is greatly reduced. r-KGE employs relaxation variables to model the noise explicitly, and a simple noise reduction method is used to reduce the noise of KGs. But there are some disadvantages in this work: it cannot solve $n-to-n$ relations and the reasoning process is too time-consuming, especially for large KGs, which makes the algorithm has poor scalability.

**INS** [195] is a data-driven inference method which naturally incorporates the logic rules and TransE together through *MLNs*

**Table 22**
Statistics of pros and cons of rule-based KGC methods and embedding-based KGC methods.

| Category | Advantage | Disadvantage |
|---|---|---|
| Rule-based KGC | 1. Consider explicit logical semantics<br>2. Strong explainability and accuracy<br>3. Data dependency<br>4. Can be applied to both transductive and inductive problems<br>5. High robustness avoiding re-training | 1. Poor Scalability<br>2. Noise sensitive<br>3. High computational complexity |
| Embedding-based KGC | 1. High scalability<br>2. High efficiency<br>3. Not affected by huge candidate sets | 1. Data-driven<br>2. Poor explainability<br>3. Hard to model the interaction of different relations<br>4. Cannot handle inductive scenarios |

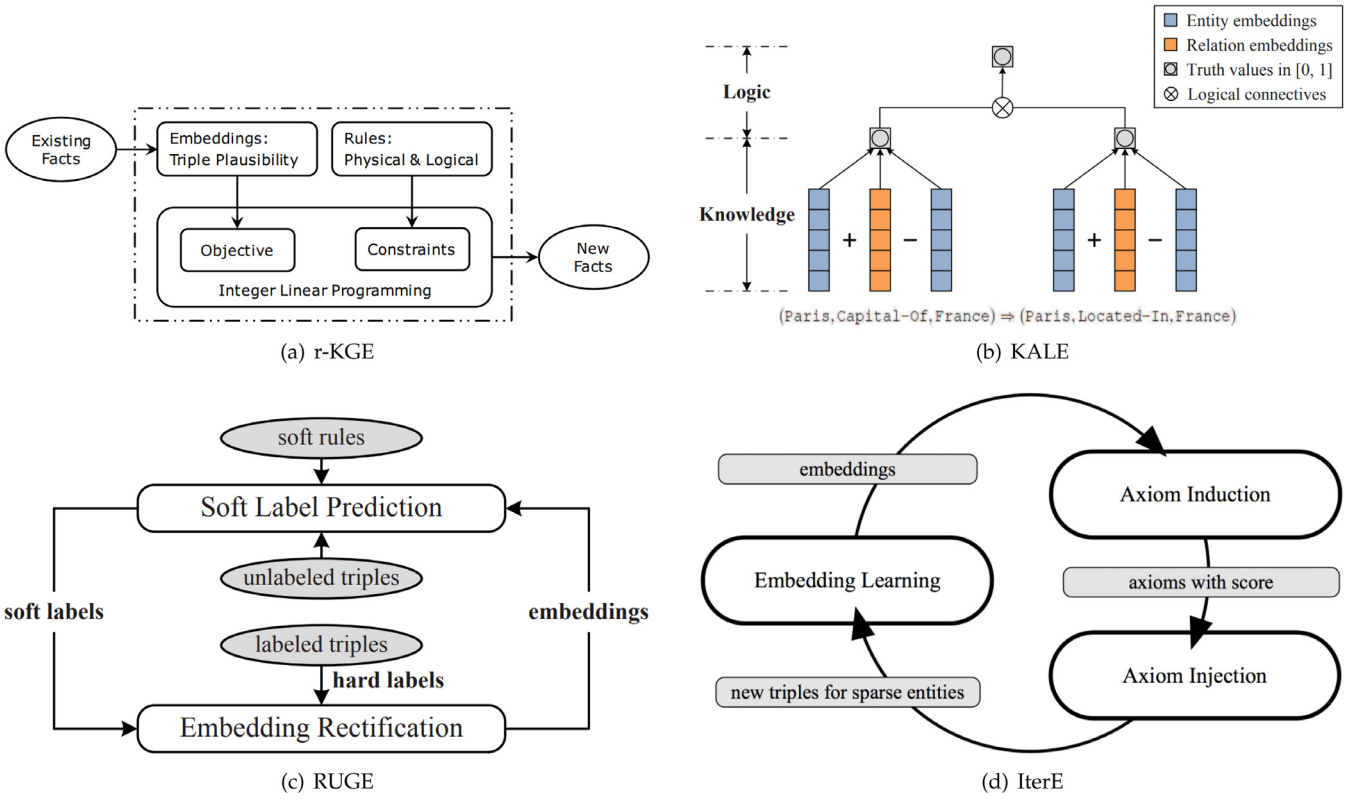



**Fig. 27.** Several typical KGC models which *combine logical rules and embedding models*, the development from (a) to (d) shows the process of deepening interaction between rules and embedding models.
*Source:* These pictures are extracted from [52,185,197,199]

to conduct KGC, where TransE calculates the similarity score between the candidate and the correct tag, so as to take the top-N instances selection to form a smaller new candidate set, which not only filters out the useless noise candidates, but also improves the efficiency of the reasoning algorithm. The calculated similarity score is used as a priori knowledge to promote further reasoning. For these selected candidate cases, INS and its improved version **INS-ES** [195] algorithm adopted in MLN network is proposed to consider the probability of transition between network sampling states during reasoning, therefore, the whole reasoning process turns into supervised. It is worth noting that INS greatly improves the Hits@1 score in FB15K dataset.

**A Matrix Factorization Based Algorithm utilizing ProPPR (ProRR-MF)** [196] tries to construct continuous low dimensional embedding representation for first-order logics from scratch, and is interested in learning the potential and distributed representation of horn clause. It uses scalable probabilistic logic structure (ProPPR in [216]) learning to construct expressive and learnable logic formulas from the large noisy real-world KGs, and applies a matrix factorization method to learn formula embedding. This work is the first formal research on low dimensional embedding learning of first-order logic rules. However, it is still in a dilemma in predicting new knowledge since it has not combines entity, relation and rule embedding to cooperate symbolic reasoning with statistical reasoning.

Nevertheless, although these several KGC methods jointly model with logical rules and embeddings, the rules involved in them are used merely as the post-processing of the embedding methods, which leads to less advance in the generation of better embedding representation [197].

*(2). Explore further combination style:*

Different from previous approaches, the latter literatures expect to explore more meaningful combination ways, rather than just jointly working on the surface level.

**KALE** [197] is a very simple KGC model which combines the embedding model with the logical rules, but pays attention to the deep interaction between rules and embedding methods. The main idea of KALE is to represent triples and rules in a unified framework, in which triples are represented by atomic formulas and modeled by translation hypothesis; rules are represented by

complex formulas and modeled by t-norm fuzzy logic. Embedding can minimize the overall loss of atomic formulas and complex formulas. In particular, it enhances the prediction ability of new facts that cannot be inferred directly from pure logic inference, and it has strong generality for rules.

**Trans-rule** [198] is also a translation-based KG embedding and logic rules associative mode, what distinguishes this model from previous similar works is that, it concerns about rules having confidence above a threshold, including *inference rules*, *transitivity rules* and *antisymmetry rules*, these rules and their confidences are automatically mined from triples in KG, then they are placed together with triples into a unify first-order logic space which allow rules encoded in it. Additionally, to avoid algebraic operations inconsistency problem, it maps all triples into first-order logics, and also defines kinds of interaction operations for rules to keep the form of rules encoding to 1-to-1 mapping relation.

*(3) Iteration interactions*

With the emergence of new completion demands, a new way of jointly learn rules and embeddings for KGC in a iteration manner comes into being.

**RUGE** [199] is a novel paradigm of KG embedding model which combines the embedding model with logic rules and exploits guidance from soft rules in an iterative way. RUGE enables the embedding model to learn both labeled and unlabeled triples in exiting KG, the soft rules with different confidence levels can be acquired automatically from the KG at the same time. Vary from the previous studies, this work first applies a iterative manner to deeply capture the interactive nature between embedding learning and logical inference. The iterative procedure can automatically extracted beneficial soft rules without extensive manual effort that are needed in the conventional attempts which always use hard rules in a one-time injection manner. Each iteration contains two stage: soft label prediction and embedding rectification, the two partial responsible for approximately reasoning, predicting and updating the KG with the newly predicted triples for further better embeddings in the next iteration respectively. Though the whole iteration procedure, this flexible approach can fully divert the rich knowledge contained in logic rules to the learned embeddings. Moreover, RUGE demonstrates the usefulness of automatically extracted soft rules according a series of experiments.

**Iterative Rules Inducing (ItRI)** [200] iteratively extends induced rules guided by feedback information of the KG embedding model calculated in advance (including probabilistic representations of missing facts) as well as external information sources, such as text corpus, thus the devised approach not only learns high quality rules, but also avoids scalability problems. Moreover, this machinery is more expressive through supporting non-monotonic rules of negated atoms and partially grounded atoms.

**IterE** [201] recursively combines the embedding model and rules to learn the embedding representation as well as logic rules. IterE mainly consists of three parts: embedding representation, axiom induction, axiom injection, and the training is carried out by interactive iteration among these three parts so that rules and embedding can promote each other to the greatest extent, forming the final reasoning framework. Specifically, on the one hand, the embedding model learns from the existing triples in KGs as well as the triples inferred from the rules. On the other hand, the confidence score of axioms derived from the pruning strategy should be calculated on the learned relational embeddings according to the linear mapping hypothesis, and then new triples can be inferred by the axioms. Finally, the new triples are linked into KGs for following entity embedding learning. The recursive operation designed by IterE not only alleviates the sparsity of KGs but also pays attention to the influence of semantics on rules. IterE proposes a new form of combining rule and embedding representation, which provides a new idea for KGC research combining different types of methods.

**pLogicNet** proposed by [202] is the product of cooperation between KG embedded model and *MLN logic rules*. Similar to IterE, the operation process of pLogicNet is also carried out under the deep interaction between embedding and rules. The difference is that in pLogicNet, a first-order Markov logic network is used to define the joint distribution of all possible triples, then applies the variant algorithm of *EM algorithm* to optimize pLogicNet. In the E step of the variant EM algorithm, the probability of unobserved triples is deduced by using amortized mean-field inference, and the variation distribution is parameterized as the parameter of the KG embedding model; in M-step, the weights of the logic rules are updated by defining the pseudo-likelihood on both the observed triples and the triples inferred from the embedding model. PLogicNet can effectively use the stochastic gradient descent algorithm to train. The training process iteratively performs E-step and M-step until convergence, and the convergence speed of the algorithm is very satisfactory.

*4.2.1.5. Cooperating rules with other information. (1) Cooperating with abductive text evidence*

**TEP-based Abductive Text Evidence for KGC (FEP-AdTE)** [203] combines logical information and text information to form a new knowledge verification system, adding new fact triples to KGs. The main idea of this paper is to define the explanation of triples — the form of (triples, windows) abductive text evidence based on TEP, in which the sentence window $w$ explains the degree of the existence of the triple $\tau$, and uses the remote supervision method in relation extraction to estimate the abductive text evidence. FEP-AdTE considers only the subset-minimal abductive explanation (called Mina explanation) to make the explanation as concise as possible and applies the hypothesis constraint to limit the number of Mina explanations to be calculated to make the interpretation work possible. It is worth mentioning that this paper has developed KGs corresponding to the text corpus of four Chinese classics to evaluate the new knowledge verification mechanism of this paper. However, the triple interpretation in this paper does not contain valuable entity-type attributes. In future work, we can consider adding pragmatic interpretation of entity types to further enhance the verification effect of new knowledge and make contributions to KGC.

*(2) Cooperating with path evidence*

**AnyBURL** [204] can learn logic rules from large KGs in a bottom-up manner at any time. AnyBURL is further designed as an effective KG rule miner, the concept of example is based on the interpretation of path in KGs, which indicates that KGs can be formed into a group of paths with edge marks. In addition, AnyBURL learns fuzzy, uncertain rules. Because the candidate ranking can be explained by the rules that generate the ranking, AnyBURL has good explanatory power. In addition to the other advantages of rule-based KGC, the additional advantages of AnyBURL are its fast running speed and less use of resources. In addition, AnyBURL proves that rule learning can be effectively applied to larger KBs, which overturns the previous bias against the rule-based KGC method.

**ELPKG** [205] combines path information, embedding representation and soft probability logic rules together. In a word, the KG embedding model is used to train the representation of inter-entity relation, and breadth-first search is used to find the path between entity nodes. The representation of entity/relation based on path information is combined with the representation based

**Table 23**
Candidate filtering works in rule reasoning.

| Model | Candidate filtering strategies |
|---|---|
| AMIE, AMIE+ [207] | Maximum rule length, Perfect rules, Simplifying projection queries, confidence threshold *minConf* |
| INS [195] | Instance selection using TransE |
| NTP 2.0 [194] | Approximate Nearest Neighbor Search |
| RLvLR [192] | *MinSC* and *MinHC* |
| IterE [201] | Traversing and random selection |
| DRUM [184], RUGE [199], Trans-rule [198], RPJE [206], ItRI [200] | Confidence measure for pruning rules |
| CHAI [26] | Filtering candidates criteria in KGs: $exists_{KG}((h,r,t)) \leftrightarrow \exists e \in \mathcal{E}\|(h,r,e) \in \mathcal{T}$, $dom_{KG,rel}((h,r,t)) \leftrightarrow \exists e \in \mathcal{E}\|(t,rel,e) \in \mathcal{T}$, $ran_{KG,rel}((h,r,t)) \leftrightarrow \exists e \in \mathcal{E}\|(e,rel,t) \in \mathcal{T}$, $distance_{KG,i}((h,r,t)) \leftrightarrow dist(KG,h,t) \leqslant i$ |

on the embedding vector to generate relational representation between entities. On this basis, the probability soft logic is applied to deduce and predict the relation probability between entities to perform KGC, which solves the problems of knowledge inconsistency and knowledge conflict. Finally, the method is used to complete the relation between KG entities. ELPKG not only ensures the efficiency of it but also shows the high accuracy of LP. Because it makes full use of the existing facts of KG, it does not need external auxiliary knowledge.

**RPJE** [206] also combines path and semantic level associate relations by Horn rules. Firstly, it mines and encodes logical rules of Horn sub-sentence forms with different lengths from the knowledge graph, and then uses the rules with length 2 to accurately combine paths, and explicitly makes length 1 rules to create a semantic association between relations and constrain relations vector representation. In addition, the confidence degree of each rule is also considered in the optimization process to ensure that the rule should be effective in representation learning. RPJE combines logic rules and paths to embed KG, which fully benefits the interpretability and accuracy of logic rules-based KGC methods, the generalization of KG embedding, and the semantic structure information provided by paths. The combination strategy of this paper is simple so that it is worth trying to adopt more complex combination methods, such as using the LSTM with an attention mechanism suitable for long-path modeling. In addition, learn from the interaction between embedding and rules in IterE and pLogicNet to explore how to use a well-designed closed-loop system to push embedded information back from RPJE to rule learning also deserves people's attention.

*4.2.1.6. Candidate filtering in rule reasoning.* Some rules are proposed for *filtering candidate triples* (called filtering rules) in the context of the KGC process by combining a number of criteria in such a way that it optimizes a given fitness function, the produced rules can be applied to the initial set of candidates and generate a reduced set that contains only the more promising candidate triples rather than using the full set possible missing candidate triples (and thus provide no filtering) or applying very basic rules to filter out unlikely candidates most current approaches do, which may have a negative effect on the completion performance as very few candidate triples are filtered out [26]. A summary table about candidate filtering are listed as Table 23.

**AMIE+** [207] presents a series of pruning strategies including formulating *Maximum rule length*, *Perfect rules* and *Simplifying projection queries*. Besides, they prune rules with a confidence threshold *minConf* and conduct confidence approximations that allow the system to explore the search space much more efficiently.

**Inferring via Grounding Network Sampling (INS)** [195] employs an embedding-based model (TransE) to conduct the instance selection and form much smaller candidate sets for subsequent fact inference, whose aim is not only narrowing the candidate sets but also filtering out part of the noise instances.

**NTP 2.0** [194] shows that searching answer facts over KGs that best explain a query can be reduced to a k-nearest neighbor problem, for which efficient exact and approximate solutions exist [79].

**RLvLR** [192] sets the *MinSC* and *MinHC* which represent the minimum values of standard confidence and head coverage for learned rules, respectively, to further filter the candidate rules.

**IterE** [201] utilizes a pruning strategy combining traversing and random selection to generate a pool of possible axioms and then assigns a score to each axiom in the pool based on a calculation between relation embeddings according to rule conclusions from linear map assumption.

The work in [184,198–200,206] tend to devise the confidence measures that capture rule quality better for pruning out not promising rules, thus improve the ranking of rules.

**CHAI** [26] At the same time, CHAI focuses on the filtering method of candidate triples in the KGC process. It points out that the previous KGC method considers all candidate triples or filters candidate sets roughly, which is not reasonable. To solve these problems, CHAI considers more complex rules based on relation domain and distance to normalize the candidate set and effectively selects the most promising candidate triples to form the smallest candidate set, so as to improve the performance of KGC. Although this method provides a good idea for filtering candidate triples, it is not suitable for large relational KGs and sparse KGs, which can be further improved in the future. In the experiment, it is compared with [25], whose candidate set filtering proposal is replacing the target entity with the entities within the range of all relations of the existing triples, so as to generate candidate triples.

*4.2.1.7. Evaluation and datasets of rule-based KGC methods.* ***About Evaluation:*** Mining rules have traditionally relied on predefined statistical measures such as support and confidence to assess the quality of rules [192]. These are fixed heuristic measures. For example, to assess the quality of mined rules, the common measures that are used to rule learning mostly evaluate candidates rules according to their *Standard Confidence (SC)* and *Head Coverage (HC)*. If entity pair $(e, e')$ satisfies the body of $r$ (denoted as $body(r)(e, e')$), and $(e, e')$ satisfies the head of $r$ (denoted as $R_t(e, e')$), for the entities $e_1, \ldots, e_{n-1}$ and the facts $R_1(e, e_1), R_2(e_1, e_2), \ldots, R_n(e_{n-1}, e')$ in KG, when there exists $R_t(e, e')$ in the KG, the computation of SC and HC are as follows:

$$SC(r) = \frac{supp(r)}{\#(e, e') : body(r)(e, e')}$$

$$HC(r) = \frac{supp(r)}{\#(e, e') : R_t(e, e')}$$

where $supp(r)$ is the support degree of rule $r$:

$$supp(r) = \#(e, e') : body(r)(e, e') \wedge R_t(e, e')$$

Whereas these measures maybe are not optimal for various use cases in which one might want to use the rules. For instance, using SC is not necessarily optimal for statistical relational learning. Therefore, the work in [207] develops *PCA confidence* to allow the counterexamples generation in a less restrictive way than SC.

**Table 24**
An KGC example using rules referred to [208]. In this instance, four relevant rules for the completion task $(h, r, ?)$ resulting in the ranking $(g(0.81), d(0.81), e(0.23), f(0.23), c(0.15))$. A rule can generate one candidate (fourth row), several candidates (first and third row), or no candidate (second row).

| Rule | Type | Confidence | Result |
|---|---|---|---|
| $r(x, y) \leq s(y, x)$ | P1 | 0.81 | {d,g} |
| $r(x, y) \leq r(y, x)$ | P1 | 0.7 | $\phi$ |
| $r(x, y) \leq t(x, z) \wedge u(z, y)$ | P2 | 0.23 | {e,f,g} |
| $r(x, c) \leq \exists y r(x, y)$ | C | 0.15 | {c} |

Besides, the work in [184] uses two theorems to learn rule structures and appropriate scores simultaneously. However, this is a challenge because the method needs to find an optimal structure in a large discrete space and simultaneously learn proper score values in a continuous space. Due to the process of evaluating candidate rules in a rule mining system is generally challenging and time-consuming, [192] reduces its computation to a series of matrix operations. This efficient rule evaluating mechanism allows the rule mining system to handle massive benchmarks efficiently. Meilicke et al. [208] presents a unified fine-grained evaluation framework that commonly assesses rule-based inferring models over the datasets generally used for embedding-based models, making the effort to observe the valuable rules and interesting experiences for KGC. Consider the rule's confidence as well, since when we use relevant rules for the complete task $(h, r, ?)$, a rule can generate a variable number of candidate, and the possible ways of aggregating the results generated by the rules are various. The work in [208] defines the final score of an entity as the maximum confidence scores of all rules that generated this entity. Furthermore, if a candidate has been generated by more than one rule, they use the amount of these rules as a secondary sorting attribute among candidates with the same (maximum) score. For instance in the Table 24, if there are four relevant rules for completing $(h, r, ?)$ and resulting in the final ranking $(g(0.81), d(0.81), e(0.23), f(0.23), c(0.15))$. To support the evaluation system, this paper designs a simplified rule-based model called **RuleN** for assessing experiments and evaluated together with the AMIE model. With the inspiring results of experiments showing that models integrating multiple different types of KGC approach deserve to be attracted attention in KGC task, this paper further classifies test cases of datasets for fine-grained evaluation according to the interpretation generated by the rule-based method, then gets a series of observations about the partitioning of test cases in datasets.

***Datasets:*** Table 25 list the basic statistics information about common used datasets for rule-based KGC research. Here we introduce several datasets in detail.

**NELL**: NELL datasets (http://rtw.ml.cmu.edu/rtw/resources) and its subsets are likely to be used as experimental data, including NELL-995 [206], Location and Sport [185].

**FB122**: composed of 122 Freebase relations [197] regarding the topics of "people", "location", and "sports", extracted from FB15K. FB122's test set are further split into two parts test-I and test-II, where the former contains triples that cannot be directly inferred by pure logical inference, and the latter the remaining test triples.

**Countries**: a dataset introduced by [225] for testing reasoning capabilities of neural link prediction models [193]. Triples in Countries are $(countries(c), regions(r), subregions(sr))$ and they are divided into train, dev and test datasets which contain 204, 20 and 20 countries data.

**KGs about Four great classical masterpieces of Chinese literature (FGCN)**: new KGs and the corresponding logical theories are constructed from existing text corpora in a domain about character relationships in the four great classical masterpieces of Chinese literature, namely *Dream of the Red Chamber (DRC)*, *Journey to the West (JW)*, *Outlaws of the Marsh (OM)*, and *Romance of the Three Kingdoms (RTK)* [203]. Triples in those KGs are collected on character relationships from e-books for these masterpieces, yielding four KGs each of which corresponds to one masterpiece.

**Table 25**
Statistics about other datasets for KGC using rules.

| Dataset | Entity | Relation | Fact | | |
|---|---|---|---|---|---|
| | | | #Train | #Valid | #Test |
| NELL-995 [206] | 75,492 | 200 | 123,370 | 15,000 | 15,838 |
| DRC [203] | 388 | 45 | 333 | – | 34 530 |
| JW [203] | 104 | 21 | 106 | – | 27 670 |
| OM [203] | 156 | 38 | 178 | – | 34 010 |
| RTK [203] | 123 | 30 | 132 | – | 29 817 |
| FB122 [197] | 9738 | 122 | 91,638 | 9595 | 5057+6186 |
| FB166 [198] | 9658 | 166 | 100,289 | 10,457 | 12,327 |
| YAGO [205] | 192 628 | 51 | 192 900 | | |
| NELL [205] | 2 156 462 | 50 | 2 465 372 | | |
| YAGO-50 [205] | 192 628 | 50 | 100 774 | | |
| YAGO-rest [205] | 192 628 | 41 | 92 126 | | |
| Sport [185] | 447 | 5 | 710 | | |
| Location [185] | 195 | 5 | 231 | | |
| Countries [225] | 244+23 | 5 | 1158 | | |

*4.2.1.8. Analysis of rule-based KGC methods.* In summary, we analyze some tips about experiment rule-based KGC methods on the common benchmark. Referring to the generated results in [208], which allow for a more comprehensive comparison between various rule-based methods and embedding-based approaches for KGC, employing a global measure to rank the different methods. On this basis, we gained several interesting insights:

**1.** Both AMIE and RuleN perform competitively to embedding-based approaches for the most common benchmarks. This holds for the large majority of models reported about in [226]. Only a few of these embedding models perform slightly better.

**2.** Since the rule-based approaches can deliver an explanation for the resulted ranking, the characteristic can be helpful to conduct fine-grained evaluations and understand the regularities within and the hardness of a dataset [204].

**3.** The traditional embedding-based KGC methods may have matters in solving specific types of completion tasks whereas it can be solved easily with rule-based approaches, this tip becomes even more important when the situations looking solely at the top candidate of the filtered ranking.

**4.** One reason for the good results of rule-based systems is the fact that most standard datasets are dominated by rules such as symmetry and (inverse) equivalence (except for those especially constructed datasets, e.g., FB15k-237).

**5.** It is quite possible to leverage both families of approaches by learning an ensemble [185,195–199,202] to achieve better results than any of its members. The overall ensemble models tend to contain a closed-loop operation, which indicates that the embedding expression and rules are mutual achievements with each other. In the future, it is necessary to explore more effective interaction ways for integrating these two categories approaches.

**6.** Recently, novel effective but complex KG encoding models emerge in endlessly, which also provides alternative techniques for KGC to combine knowledge embedding and rules in the future.

*4.2.2. Third-party data sources-based KGC*

Some related techniques learn entity/relation embeddings from triples in a KG jointly with third-party data sources, in particular with the additional textual corpus (e.g., Wikipedia articles) for getting help from related rich semantic information.

**Table 26**
Statistics of popular KGC models using third-party data sources.

| Models | Technology | Information (Data Source) | Datasets |
|---|---|---|---|
| Joint alignment model: | | | |
| JointAS [15] | TransE, skip-gram, words co-occurrence, entity-words co-occurrence | Structural information, entities names, Wikipedia anchors | Freebase subset; English Wikipedia |
| JointTS [121] | TransE, skip-gram, JointAS | Structural information, entities names Wikipedia text descriptions, textual corpus | FB15K, Freebase subset; Wikipedia articles |
| DKRL [94] | TransE, CBOW, CNN, max/mean-pooling | Structural information, multi-hop path, entity descriptions | FB15K, FB20K |
| SSP [227] | TransE, topic extraction, Semantic hyperplane Projection | Structural information, entity descriptions | FB15K; Wikipedia corpuses |
| Prob-TransE (or TransD) JointE (or JointD) [228] | TransE/TransD, CNN, semantics-based attention mechanism | Structural information, entity descriptions, anchor text, textual corpus | FB15K; NYT-FB15K |
| JOINER [229] | TransE, regularization, JointAS | Structural information, textual corpus, Wikipedia anchors | Freebase subset; English Wikipedia |
| ATE [230] | TransE, BiLSTM, Skip-Gram, mutual attention mechanism | Relation mentions and entity descriptions, textual corpus | Freebase, WordNet; English Wikipedia (Wiki) |
| aJOINT [162] | TransE, collaborative attention mechanism | KG structural information, textual corpus | WN11, WN18, FB13, FB15k; Wikipedia articles |
| KGC with Pre-trained Language Models (PLMs): | | | |
| JointAS [15], DESP [121], DKRL [94] | word2vec | Structural information, textual information | FB15K, FB20K |
| LRAE [231] | TransE, PCA, word2vec | Structural information, entity descriptions | FB15k, WordNet |
| RLKB [232] | Probabilistic model, single-layer NN | Structural information, entity descriptions | FB500K, EN15K |
| Jointly-Model [233] | TransE, CBOW/LSTM, Attention, Gate Strategy | Structural information, entity descriptions | FB15K, WN18 |
| KGloVe-literals [234] | Entity recognition, KGloVe | Textual information in properties, textual corpus | Cities, the AAUP, the Forbes, the Metacritic Movies, the Metacritic Albums; DBpedia abstracts |
| Context Graph Model [235] | Context graph, CBOW, Skip-Gram | Analogy structure, semantic regularities | DBpedia |
| KG-BERT [236] | BERT, sequence classification | Entity descriptions, entity/relation names, sequence order in triples, textual corpus | WN11, FB13, FB15K, WN18RR, FB15k-237, UMLS; Wikipedia corpuses |
| KEPLER [237] | RoBERTa [238], masked language modeling (MLM) | KG structural information, entity descriptions, textual corpus | FB15K, WN18, FB15K-237, WN18RR; Wikidata5M |
| BLP [239] | BERT, holistic evaluation framework, inductive LP, TransE, DistMult, ComplEx, and SimplE | KG structural information, entity descriptions, textual corpus | FB15K-237, WN18RR; Wikidata5M |
| StAR [240] | RoBERTa/BERT, multi-layer perceptron (MLP), Siamese-style textual encoder | KG structural information, entity descriptions, textual corpus | WN18RR, FB15k-237, ULMS, NELL-One; Wikipedia paragraph |

Next, we will systematically introduce KGC studies that use third-party data source, we also list them in Table 26 for a direct presentation.

*4.2.2.1. Research inspiration.* This direction is inspired by these three key items: **Firstly**, pre-training language models (PLMs) such as Word2Vec [75], ELMo [241], GPT [242], and BERT [243], have caused the upsurge in the field of natural language processing (NLP) which can effectively capture the semantic information in text. They originated in a surprising found that word representations that are learned from a large training corpus display semantic regularities in the form of linear vector translations [75], for example, $king - man + woman \approx queen$. Such a structure is appealing because it provides an interpretation of the distributional vector space through lexical-semantic analogical inferences. **Secondly**, under the *Open-world Assumption*, a missing fact often contains entities out of the KG, e.g., one or more entities are phrases appearing in web text but not included in the KG yet [15]. While only relying on the inner structure information is hard to model this scene, the third-party textual datasets can provide satisfied assistance for dealing with these out-of-KG facts. **Thirdly**, similar to the last point, auxiliary textual information such as entity descriptions can help to learn sparsity entities, which act as the supplementary information of these entities lacking sufficient messages in the KG to support learning.

The most striking textual information is entity description, very few KGs contain a readily available short description or definition for each of the entities or phrases, such as **WordNet**

and **Freebase**, and usually it needs the additional lexical resources to provide textual training. For instance, in a medical dataset with many technical words, the Wikipedia pages, dictionary definitions, or medical descriptions via a site such as 'medilexicon.com' could be leveraged as lexical resources [236].

*4.2.2.2. Joint alignment model.* **JointAS** [15] *jointly embeds entities and words* into the same continuous vector space. Entity names and Wikipedia anchors are utilized to align the embeddings of entities and words in the same space. Numerous scale experiments on Freebase and a Wikipedia/NY Times corpus show that jointly embedding brings promising improvement in the accuracy of predicting facts, compared to separately embedding KGs and text. Particularly, JointAS enables the prediction of facts containing entities out of the KG, which cannot be handled by previous embedding methods. The model is composed of three components: the knowledge model $L_K$, text model $L_T$, and alignment model $L_A$ which make the use of entity names $L_{AN}$ and *Wikipedia anchors* $L_{AA}$, thus the overall objective is to maximize this jointly likelihood loss function:

$$L = L_K + L_T + L_A$$

where $L_A$ could be $L_{AA}$ or $L_{AN}$ or $L_{AN} + L_{AA}$, and the score function $s(w, v) = b - \frac{1}{2}(\|w - v\|^2)$ of a target word $w$ appearing close to a context word $v$ (within a context window of a certain length) for text model while the score function $s(h, r, t) = b - \frac{1}{2}(\|v_h + v_r - v_t\|^2)$ for KG model, in which the $b$ is a bias constant.

Although this alignment model goes beyond previous KGE methods and can perform prediction on any candidate facts between entities/words/phrases, it has drawbacks: using entity names severely pollutes the embeddings of words; using Wikipedia anchors completely relies on the special data source and hence the approach cannot be applied to other customer data.

**JointTS** [121] takes these above-mentioned issues into consideration, without dependency on anchors, it improves alignment model $L_A$ based on *text descriptions of entities* by considering both conditional probability of predicting a word $w$ given entity $e$ and predicting a entity $e$ when there is a word $w$. This model learns the embedding vector of an entity not only to fit the structured constraints in KGs but also to be equal to the embedding vector computed from the text description, hence it can deal with words/phrases beyond entities in KGs. Furthermore, the new alignment model only relies on the description of entities, so that it can obtain rich information from the text description, thus well handles the issue of KG sparsity.

**DKRL** [94] is the first work to build entity vectors directly applying *entity description* information. The model combines triple information with entity description information to learn vectors for each entity. The model efficiently learns the semantic embedding of entities and relations relying on the *CBOW* and CNN mechanism and encodes the original structure information of triples with the use of TransE. Experiments on both KGC and entity classification tasks verify the validity of the DKPL model in expressing new entities and dealing with zero-shooting cases. But it should not be underestimated that DKRL tune-up needs more hyper-parameters along with extra storage space for inner layers' parameters.

**Semantic Space Projection (SSP)** [227] is a method for KGE with text descriptions modifying TransH. SSP jointly learns from the symbolic triples and textual descriptions, which builds interaction between these two information sources, at the same time textual descriptions are employed to discover semantic relevance and offer precise semantic embedding. This paper firmly convinced that triple embedding is always the main procedure and textual descriptions must interact with triples for better embedding. SSP can model the strong correlations between symbolic triples and textual descriptions by performing the embedding process in a semantic subspace.

**Prob-TransE and Prob-TransD** [228] jointly learns the representation of the entities, relations, and words within a unified parameter sharing semantic space. The KG embedding process incorporates TransE and TransD (called Prob-TransE and Prob-TransD) as representative in the framework to handle representation learning of KGs, while the stage of representation learning of textual relations applies CNN to embed textual relations. A reciprocal attention mechanism consists of *knowledge based attention* and the *semantics attention (SATT)* are proposed to enhance the KGC. The attention mechanism can be simply described as follows: during the KG embedding process, semantic information extracted from text models can be used to help explicit relations to fit more reasonable entity pairs, similarly, additional logical knowledge information can be utilized to enhance sentence embedding and reduce the disadvantageous influences of noisy generated in the process of distant supervision. The experiments use *anchor text annotated in articles* to align the entities in KG and entities mentions in the vocabulary of the text corpus, and build the alignment between relations in KGs and text corpus with the idea of distant supervision. A series of comparative experiments prove that the joint models (**JointE+SATT** and **JointD+SATT**) have effective performances through trained without strictly aligned text corpus. In addition to that, this framework is adaptable and flexible which is open to existing models, for example, the partial of TransE and TransD can be replaced by the other KG embedding methods similar to them such as TransH and TransR.

**JOINER** [229] jointly learns text and KG embeddings via regularization. Preserving word–word co-occurrence in a text corpus and transition relations between entities in a KG, JOINER also can use regularization to flexibly control the amount of information shared between the two data sources in the embedding learning process with significantly less computational overhead.

**ATE** [230] carries out KGE using both specific relation mention and entity description encoded with a BiLSTM module. A mutual attention mechanism between relation mentions and entity descriptions is designed to learn more accurate text representation, to further improve the representation of KG. In the end, the final entity and relation vectors are obtained by combining the learned text representation and the previous traditional translation-based representation. This paper also considers the fuzziness of entity and relation in the triple, filters out noisy text information to enrich KG embedding accurately.

**aJOINT** [162] proposes a new cooperative attention mechanism, based on this mechanism, a text-enhanced KGE model was proposed. Specifically, aJOINT enhances KG embeddings through the text semantic signal: the multi-directional signals between KGE and text representation learning were fully integrated to learn more accurate text representations, so as to further improve the structure representation.

*4.2.2.3. KGC with pre-trained language models.* Recently, pre-trained language models (PLMs) such as ELMo [241], Word2Vec [75], GPT [242], BERT [243], and XLNet [244] have shown great success in NLP field, they can learn contextualized word embedding with large amount of free text data and achieve excellent performance in many language understanding tasks [236].

According to the probable usage of PLMs in KGC tasks, the related approaches can be roughly divided into two categories [236]: **feature-based** and **fine tuning approaches**. Traditional feature-based word embedding methods like Word2Vec and

Glove [92] aim to learn context-independent word vectors. ELMo generalized traditional word embedding to context-aware word embedding, where word polysemy can be properly handled. Mostly, these word embeddings learned from them are often used as initialization vectors during the KGC process. Different from the former method, fine-tuning approaches such as GPT and BERT use the pre-trained model structure and parameters as the starting point of specific tasks (KGC task we care about). The pre-trained model learns rich semantic patterns from free text.

**Lexical Resources Auxiliary Embedding Model (LRAE)** [231] explores methods to provide vector initialization for TransE by using the semantic information of entity description text. LRAE exploits entity descriptions that are available in WordNet and Freebase datasets. The first sentence of a given entity description is first selected and then decomposed into a series of word vectors (the first sentence is often most relevant to the described entity, which avoids noise interference and large-scale computation from lengthy description text), next all those vectors are averaged to form embeddings that represent the overall description semantics of the entity, where word vectors are computed by Word2vec [75] and GloVe [92]. These processed descriptive text vectors are used as the initialization vectors of the translation model and are input to TransE for training. LRAE provides initialization vectors for all entities, even including those not present in the data, thus it alleviates the entity sparse issue. Also, LRAE is very versatile and can be applied directly to other models whose input is represented by solid vectors.

**RLKB** [232] modifies DKRL by developed a single-layer probabilistic model that requires fewer parameters, which measures the probability of each triple and the corresponding entity description, obtains contextual embeddings of entities, relations, and words in the description at the same time by maximizing a logarithmic likelihood loss.

**Jointly-Model** [233] proposes a novel deep architecture to utilize both structural and textual information of entities, which contains three neural models to encode the valuable information from the text description of entity: Bag-of-Words encoder, LSTM encoder and Attentive LSTM encoder, among which an attentive model can select related information as needed, because some of the words in an entity's description may be useful for the given relation, but may be useless for other relations. The Jointly-Model chooses a gating mechanism to integrate representations of structure and text into a unified architecture.

**Including Text Literals in KGloVe (KGloVe-literals)** [234] combines the text information in entity attributes into KG embeddings, which is a preliminary exploration experiment based on KGloVe: it firstly performs KGloVe step to create a graphical co-occurrence matrix by conducting a personalized PageRank (PPR) on the (weighted) graph; at the same time, it extracts information from the DBpedia summary by performing Named Entity Recognition (NER) step, in which the words representing the entity are replaced by the entity itself, and the words surrounding it (and possibly other entities) are contained in the context of the entity; then the text co-occurrence matrix is generated in collaboration with the list of entities and predicates generated in the KGloVe step. Finally, a merge operation is performed to combine the two co-occurrence matrices to fuse the text information into the latent feature model. Although the gain of this work is very small, it can provide new ideas for the joint learning of attribute text information and KG embedding.

**Context Graph Model** [235] finds hidden triples by using the observed triples in incomplete graphs. This paper is based on the neural language embedding of context graph and applies the similar structure extracted from the relation similarity to infer new unobserved triples from existing triples. Excerpts from large input graphs are regarded as the simplified and meaningful context of a group of entities in a given domain. Next, based on the context graph, CBOW [75] and Skip-Gram [245] models are used to model KG embedding and perform KGC. In this method, the semantic rules between words are preserved to adapt to entities and relationships. Satisfactory results have been obtained in some specific field.

The well-known BERT [243] is a prominent PLM by pre-training the bidirectional Transformer encoder [246] through masked language modeling and next sentence prediction. It can capture rich linguistic knowledge in pre-trained model weights. As this basis, a number of KGC models try to exploit BERT or its variants for learning knowledge embedding and predicting facts:

**KG-BERT** [236] treats entity and relation descriptions of triples as textual sequences inputting to BERT framework, and naturally regards KGC problems as corresponding sequence classification problems. KG-BERT computes the scoring function of serialized triples with a simple classification layer. During the BERT fine-tuning procedure, they can obtain high-quality triple representations, which contain rich semantic information.

**KEPLER** [237] encodes textual entity descriptions with RoBERTa [238] as their embedding, and then jointly optimizes the KG embeddings and language modeling objectives. As a PLM, KEPLER can not only integrate factual knowledge into language representation with the supervision from KG, but also produce effective text-enhanced KG embeddings without additional inference overhead compared to other conventional PLMs.

**BLP** [239] proposes a holistic evaluation framework for entity representations learned via the inductive LP. Consider entities not seen during training, BLP learns inductive entity representations based on BERT, and performs LP in combination with four different relational models: TransE, DistMult, ComplEx, and SimplE. BLP also provides evidence that the learned entity representations transfer well to other tasks (such as entity classification and information retrieval) without fine-tuning, which demonstrates that the entity embeddings act as compressed representations of the most salient features of an entity. This is additionally important because having generalized vector representations of KGs is useful for using them within other tasks.

**Structure-augmented text representation (StAR)** [240] augments the textual encoding paradigm with KGE techniques to learn KG embeddings for KGC. Following translation-based KGE methods, StAR partitions each triple into two asymmetric parts. These parts are then encoded into contextualized representations by a Siamese-style textual encoder. To avoid combinatorial explosion of textual encoding approaches, e.g., KG-BERT, StAR employs a scoring module involves both deterministic classifier and spatial measurement for representation and structure learning respectively, which also enhances structured knowledge by exploring the spatial characteristics. Moreover, StAR presents a self-adaptive ensemble scheme to further boost the performance by incorporating triple scores from existing KGE models.

*4.2.2.4. Discussion on KGC using third-party data source.* Based on the above introduction of KGC using the third-party data source (almost all are textual corpus), we give our corresponding analysis as follows:

**1.** In a narrow sense, this part of KGC studies emphasize the utilize of additional data source outside KGs, but you may be aware that these literals tend to apply PLMs in their works, which takes us to think about the application of 'third party data' in a broader sense: these PLMs either possess plenty of parameters which have trained on large scale language corpus, or provide ready-made

**Table 27**
Statistics of a part of TKGC technologies.

| Model | Loss function[a] | Whether consider time periods | Datasets |
|---|---|---|---|
| Temporal order dependence models: | | | |
| TransE-TAE [252] | $\mathcal{L}_{marg}$ | no | YAGO2 |
| Diachronic embedding models: | | | |
| DE-Simple [253] | Sampled $\mathcal{L}_{mll}$ | No | ICEWS14, ICEWS15-05, GDELT15/16 |
| ATiSE [254] | Self-adversarial $\mathcal{L}_{ns}$ | Yes | ICEWS14, ICEWS05-15, Wikidata12k, YAGO11k |
| Temporal Information embedding models | | | |
| TTransE [255] | $\mathcal{L}_{marg}$ | No | Wikidata |
| HyTE [256] | Sampled $\mathcal{L}_{mll}$ | Yes | Wikidata12k, YAGO11k |
| ConT [257] | $\mathcal{L}_{BRL}$ | No | ICEWS14,GDELT |
| TA-DisMult [258] | Sampled $\mathcal{L}_{mll}$ | No | YAGO-15k, ICEWS14, ICEWS05-15, Wikidata |
| TNT-ComplEx [259] | Instantaneous $\mathcal{L}_{mll}$ | Yes | ICEWS14, ICEWS15-05, YAGO-15k, Wikidata40k |
| Dynamic evolution models: | | | |
| Know-Evolve [260] | Conditional intensity function | No | GDELT, ICEWS14 |
| RE-NET [261] | Total classification $\mathcal{L}_{CE}$ | No | ICEWS18, GDELT18 |
| GHN [262] | Total classification $\mathcal{L}_{CE}$ | No | ICEWS18, GDELT15/16 |
| TeMP [263] | Sampled $\mathcal{L}_{mll}$ | No | ICEWS14, ICEWS05-15, GDELT |

[a]As usual, $\mathcal{L}_{CE}$, $\mathcal{L}_{marg}$ and $\mathcal{L}_{ns}$ refers to cross entropy loss, margin-based ranking loss and negative sampling loss respectively. Besides, the $\mathcal{L}_{mll}$ means the multiclass log-loss, and $\mathcal{L}_{BRL}$ refers to the binary regularized logistic loss.

semantically-rich word embeddings, thus when we say a KGC work uses a PLM, we would think about it gets assistance from the additional language information (from other large language corpora, on which the PLM has been fully trained). In other words, we should not judge a KGC model whether use third-party data source merely according to their used datasets, it is especially important to focus on the details of the model most of the time.

**2.** As we have discussed in 4.2.2.1, PLMs have an important role in capturing rich semantic information which is helpful to KGC. Along with a growing number of assorted PLMs are proposed, in particular, the models jointly learn language representation from both KGs and large language corpus, some PLM models introduce structure data of KGs into the pre-training process through specific KGC tasks to obtain more reasonable language model parameters (such as ERNIE [247], CoLAKE [248–251]). In the future, to explore an efficient joint learning framework derive entity representations from KGs and language corpus may be needed, and the key point is how to design the interaction between these two data source, an iterative learning manner, just as the Rule-KG embedding series worked, maybe a possible future direction. What is needed is a method to derive entity representations that work well for both common and rare entities.

## 5. Other KGC technologies

In this part we focus on several other KGC techniques oriented at the special domain, including Temporal Knowledge Graph Completion (TKGC) in Section 5.1 that concerns time elements in KGs; CommonSense Knowledge Graph Completion (CSKGC) which is a relatively new field about commonsense KGs studying (see Section 5.2), and Hyper-relational Knowledge Graph Completion (HKGC) that pays attention to n-ary relation form instead of usual 2-nary triples in KGs (see Section 5.3).

### 5.1. Temporal Knowledge Graph Completion (TKGC)

At present, many facts in KGs are affected by temporal information, owing to the fact in the real world are not always static but highly ephemeral such as (*Obama*, *Presidentof*, *USA*) is true only during a certain time segment. Intuitively, temporal aspects of facts should play an important role when we perform KGC [252]. In this section, we briefly introduce some famous TKGC models. Naturally, a summary table is made to sum up all the TKGC methods introduced in our overview (Table 27).

**Temporal Knowledge Graphs (TKGs) and TKGC:** For such KGs with temporal information, we generally call them TKGs. Naturally, the completion of such KGs is called TKGC, and the original triples are redefined as quadruples $(h, r, t, T)$ where $T$ is the time (which can be a timestamp or a time span as $[T_{start}, T_{end}]$) [252]. With studying time-aware KGC problems, it helps to achieve more accurate completion results, i.e., in LP task, we can distinguish which triple is real in a given time condition, such as (*Barack Obama*, *President of*, *USA*, 2010) and (*Bill Clinton*, *President of*, *USA*, 2010). In addition, some literature also proposes *time prediction task* that predicting the most likely time for the given entity and relation by learning the time embeddings $v_T$.

According to the usage manner of temporal information, we roughly categorize recent TKGC methods into four groups: **temporal order dependence model**, **diachronic embedding model**, **temporal information embedding model** and **dynamic evolution model**.

#### 5.1.1. Temporal order dependence models

The mentioned temporal order information indicates that under the time condition, some relations may follow a certain order timeline, such as *BornIn* → *WorkAt* → *DiedIn*.

**TransE-TAE** [252] firstly incorporates two kinds of temporal information for KG completion: (a) temporal order information and (b) temporal consistency information. To capture the temporal order of relations, they tend to design a temporal evolving matrix $M_T$, with which a prior relation can evolve into subsequent relation (as Fig. 28 shows). Specifically, given two facts having same head entity $(e_i, r_1, e_j, T_1)$ and $(e_i, r_2, e_k, T_2)$, it assumes that prior relation $r_1$ projected by $M_T$ should be near subsequent relation $r_2$, i.e., $r_1 M_T \approx r_2$. In this way, TransE-TAE allows to separate prior relation and subsequent relation automatically during training. Note that the temporal order information finally is treated as a regularization term injected into original loss function, being optimized together with KG structural information.

#### 5.1.2. Diachronic embedding models

This kind of models often design a mapping function from time scalar to entity or relation embedding, input both time and entity/relation into a specific diachronic function framework to

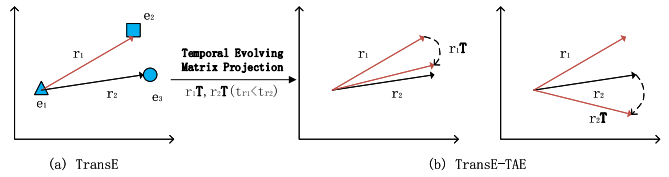



**Fig. 28.** Simple illustration of Temporal Evolving Matrix $T$ in the time-aware embedding (TAE) space [252].

get time-aware entity/relation embedding, which can be directly combined with the existing KGE models.

**DE-SimplE** [253] extends the previous static model SimplE with the diachronic entity embedding function (DEEMB, which provides the characteristics of the entity at any time point), whose input is entity and time stamp while output is entity's hidden representation at that time-step. This embedding method is called diachronic embedding (DE). Any static KGE methods can be extended to the relevant TKGE (Temporal KGE) models by using DEEMB as follows:

$$z_v^T[n] = \begin{cases} a_v[n]\sigma(w_v[n]T + b_v[n]), & if \quad 1 \le n \le \gamma d \\ a_v[n], & if \quad \gamma d \le n \le 0 \end{cases}$$

where $z_v^T[n]$ represents the $n$th element of the $d$-dimensional entity vector, which is calculated in two parts: the first part captures the temporal characteristics of the entity, and the function adopts $sin()$ to learn a set of parameters $a, w, b$ for each entity; the second part captures the static characteristics of the entity, i.e., to keep the original entity embedding unchanged. In other words, DE-SimplE can learn how to open and close entity time-series features at different time points with the use of $sin()$, so as to accurately predict their time at any time. At the same time, by combining SimplE [44] (static KGE model) with DE, DE-SimplE achieves fully expressive (an important standard measuring the quality of KGE model proposed in SimplE).

**ATiSE** [254] introduces additive time series decomposition to function on this basis. ATiSE thinks that the evolution of entity and relation representation is random because the entity characteristics at a certain time are not completely determined by the past information, thus they map the entity and relation into a multi-dimensional Gaussian distribution, the mean vector of each entity at a certain time step represents the current expectation, and the covariance represents the uncertainty of time (the constant diagonal matrix is used to improve efficiency). For the problem that DE-SimplE only considers time points, ATiSE extends to the time span, which means a triple whose time-step within the begin time point and end time point is regarded as a positive triple. The diachronic embedding function of entities in the current time step $T$ is as follows:

$$e_{i,T} = e_i + \alpha_{e,i} w_{e,i} T + \beta_{e,i} sin(2\pi w'_{e,i} T) + (0, \sum e, i)$$

The entity embedding calculated by the above formula will be regarded as the mean value vector $\bar{e}_{s,T}$ in multi-dimensional Gaussian distribution $P_{s,T} \sim \mathcal{N}(\bar{e}_{s,T}, \sum_s)$ of the certain entity. Similar to DE-SimplE, ATiSE also can extend any traditional static KGC model developed to the TKGC model, but it cannot give full play to the ability of time expression.

### 5.1.3. Temporal information embedding models

Temporal information embedding models introduce temporal information into a specific traditional KGC baseline, like translation model or tensor decomposition model, for learning time-aware embeddings and training time-aware scoring function.

Concerning the earlier work TransE-TAE [252] (which learns non-explicit time-aware embeddings as it did not directly introduce temporal information into embedding learning), **TTransE** [255] and **HyTE** [256] integrate time embedding into the distance-based score function with the idea of TransE and TransH, the former explores three methods of introducing time factor into basic TransE, among them the vector-based TTransE performs excellent results which directly models time embedding as same as entity or relation embeddings, i.e., for a quadruples $(h, r, t, T)$, $score = -\|v_h + v_r + v_T - v_t\|$, while the latter HyTE applies a time-aware KG embedding method based on time hyperplane, after projected onto certain time hyperplane at timestamp $T$, $P_T(e_i)$ in time $T$, each entity or relation is represented as the follows form:

$$P_T(v_x) = v_x - (w_T^\top v_x) w_T$$

where $w_T$ means the corresponding normal vector of current time hyperplane, then defines a score function of quadruples $(h, r, t, T)$ as:

$$f_T(h, r, t) = \|P_T(v_h) + P_T(v_r) - P_T(v_t)\|$$

which follows the transitional characteristics.

**ConT** [257] is an extension of Tucker [37] defining a core tensor $w$ for each time stamp. **TADisMult** [258] combines tokenized time and relation into predicate sequence which input into RNN to learn temporal relation representation while **TNTComplEx** [259] adopts unfolding of 4-way tensor modes.

### 5.1.4. Dynamic evolution models

Dynamic evolution models dynamically learn entity embeddings along with time steps. This kind of methods like **Know-Evolve** [260] calls the phenomenon that entities and relations change dynamically over time as *knowledge evolution*, and it models nonlinear evolution representation of entities under this scene. Know-Evolve is used in the reasoning of TKGs, which designs a novel RNN structure for dynamic evolution representation learning of entity and sets a specific loss function based on relational score function, like RESCAL [13]. Besides, recent works use neighborhood aggregation information to predict probability of event occurrence including **RE-NET** [261], **GHN** [262] and **TeMP** [263] by Graph Convolution Network (GCN) [82].

### 5.1.5. Performance comparison of TKGC models

***Datasets:*** There are part of datasets specialized in TKGC task and several TKGC datasets are shown in Table 28. We make a brief introduction about them as follows:

**ICEWS** The Integrated Conflict Early Warning System (ICEWS) [264] is a natural episodic dataset recording dyadic events between different countries, which was first created and used in [265], where a semantic tensor is generated by extracting consecutive events that last until the last timestamp. After that, **Icews14**, **icews05-15** and **icews18** are subsets of ICEWS, corresponding to the facts of 2014, 2005–2015 and 2018 respectively. These three datasets are filtered by only selecting the most frequent entities in the graph, and all the time labels inside them are time points.

**GDELT** The Global Database of Events, Language and Tone (GDELT) [264] monitors the world's news media in broadcast, print, and web formats from all over the world, daily since January 1, 1979. As a large episodic dataset, the data format inside it is similar to ICEWS, i.e., $(e_s, e_p, e_o, e_t)$ quadruples, these events also usually be aggregated into an episodic tensor. **GDELT15-16**, **GDELT18** are subsets of GDELT.

**YAGO15K** is created firstly using **FB15K** [11] by aligning entities from FB15K to **YAGO** [266] with *SAMEAS* relations contained in a

**Table 28**
Statistic of several Temporal Knowledge Graph datasets.

| Dataset | Entity | Relation | Fact | | | Timestamps |
|---|---|---|---|---|---|---|
| | | | #Train | #Valid | #Test | |
| Time Slot-based dataset | | | | | | |
| Wikidata [258] | 11 134 | 95 | 121 442 | 14 374 | 14 283 | 1726 (1 year) |
| Wikidata12K [256] | 12 554 | 24 | 32.5k | 4k | 4k | 232 (1 year) |
| YAGO11K [256] | 10 623 | 10 | 16.4k | 2k | 2k | 189 (1 year) |
| YAGO15K [258] | 15 403 | 34 | 110 441 | 13 815 | 13 800 | 198 (1 year) |
| Fact-based dataset | | | | | | |
| ICEWS 14 [253] | 7128 | 230 | 72 826 | 8941 | 8963 | 365 (1 day) |
| ICEWS 18 [253] | 23 033 | 256 | 373 018 | 45 995 | 49 545 | 304 (1 day) |
| ICEWS 05-15 [253] | 10 488 | 251 | 386 962 | 46 275 | 46 092 | 4017 (1 day) |
| GDELT(15-16) [253] | 500 | 20 | 2 735 685 | 341 961 | 341 961 | 366 (1 day) |
| GDELT(18) [261] | 7691 | 240 | 1,734,399 | 238,765 | 305,241 | 2751 (15 min) |

**Table 29**
Evaluation results of TKGC on ICEWS14, ICEWS05-15 and GDELT datasets. Best results are in bold.

| | ICEWS14 | | | | ICEWS05-15 | | | | GDELT | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 |
| TransE [11] | 0.280 | 0.094 | – | 0.637 | 0.294 | 0.090 | – | 0.663 | 0.113 | 0.0 | 0.158 | 0.312 |
| DisMult [42] | 0.439 | 0.323 | – | 0.672 | 0.456 | 0.337 | – | 0.691 | 0.196 | 0.117 | 0.208 | 0.348 |
| SimplE [44] | 0.458 | 0.341 | 0.516 | 0.687 | 0.478 | 0.359 | 0.539 | 0.708 | 0.206 | 0.124 | 0.220 | 0.366 |
| ComplEx [43] | 0.456 | 0.343 | 0.516 | 0.680 | 0.483 | 0.366 | 0.543 | 0.710 | 0.226 | 0.142 | 0.242 | 0.390 |
| TTransE [255] | 0.255 | 0.074 | – | 0.601 | 0.271 | 0.084 | – | 0.616 | 0.115 | 0.0 | 0.160 | 0.318 |
| HyTE [256] | 0.297 | 0.108 | 0.416 | 0.655 | 0.316 | 0.116 | 0.445 | 0.681 | 0.118 | 0.0 | 0.165 | 0.326 |
| TA-DistMult [258] | 0.477 | 0.363 | – | 0.686 | 0.474 | 0.346 | – | 0.728 | 0.206 | 0.124 | 0.219 | 0.365 |
| ConT [257] | 0.185 | 0.117 | 0.205 | 0.315 | 0.163 | 0.105 | 0.189 | 0.272 | 0.144 | 0.080 | 0.156 | 0.265 |
| DE-TransE [253] | 0.326 | 0.124 | 0.467 | 0.686 | 0.314 | 0.108 | 0.453 | 0.685 | 0.126 | 0.0 | 0.181 | 0.350 |
| DE-DistMult [253] | 0.501 | 0.392 | 0.569 | 0.708 | 0.484 | 0.366 | 0.546 | 0.718 | 0.213 | 0.130 | 0.228 | 0.376 |
| DE-SimplE [253] | 0.526 | 0.418 | 0.592 | 0.725 | 0.513 | 0.392 | 0.578 | 0.748 | 0.230 | 0.141 | 0.248 | 0.403 |
| ATiSE [254] | 0.545 | 0.423 | 0.632 | 0.757 | 0.533 | 0.394 | 0.623 | 0.803 | – | – | – | – |
| TeMP-GRU [263] | 0.601 | 0.478 | 0.681 | 0.828 | **0.691** | 0.566 | **0.782** | **0.917** | **0.275** | **0.191** | **0.297** | **0.437** |
| TeMP-SA [263] | 0.607 | 0.484 | **0.684** | **0.840** | 0.680 | 0.553 | 0.769 | 0.913 | 0.232 | 0.152 | 0.245 | 0.377 |
| TNTComplEx [259] | **0.620** | **0.520** | 0.660 | 0.760 | 0.670 | **0.590** | 0.710 | 0.810 | – | – | – | – |

YAGO dump, and kept all facts involving those entities. Then, this collection of facts are augmented with time information from the **yagoDateFacts** dump. Contrary to the ICEWS data sets, YAGO15K does contain temporal modifiers, namely, 'occursSince' and 'occursUntil' [258]. What is more, all facts in YAGO15K maintain time information in the same level of granularity as one can find in the original dumps these datasets come from, this is different from [255].

**YAGO11k** [256] is a rich subgraph from **YAGO3** [267], including top 10 most frequent temporally rich relations of YAGO3. By recursively removing edges containing entities with only a single mention in the subgraph, YAGO11k can handle sparsity effectively and ensure healthy connectivity within the graph.

**Wikidata** Similar to YAGO11k, Wikidata contains time interval information. As a subset of Wikidata, **Wikidata12k** is extracted from a preprocessed dataset of Wikidata proposed by [255], its created procedure follows the process as described in YAGO11k, by distilling out the subgraph with time mentions for both start and end, it ensures that no entity has only a single edge connected to it [256], but it is almost double in size to YAGO11k.

***Performance Results Comparison:*** We report some published experimental results about TKGC methods in Table 29, from which we find that TeMP-SA and TeMP-GRU achieve satisfying results on all three datasets across all evaluated metrics. Compared to the most recent work TNTComplex [259] — which achieves the best performance on the ICEWS datasets before TeMP, are 8.0% and 10.7% higher on the *Hits*@10 evaluation. Additionally, TeMP also achieves a 3.7% improvement on GDELT compared with DE, the prior state-of-the-art on that dataset, while the results of the AtiSEE and TNTComplEx methods on the GDELT dataset are not available.

#### 5.1.6. Analysis of TKGC models

Inspired by the excellent performance of translation model and tensor factorization model in traditional KGC, temporal knowledge graph completion (TKGC) mainly introduces temporal embedding into the entity or relation embedding based on the above two kinds of KGC ideas. Recently, with the wide application of GCN in heterogeneous graphs, more and more TKGC methods adopt the idea of "subgraph of a TKG" [261] we call it **temporal subgraph**, which aggregate the neighborhood information at each time, and finally collaborate with the sequence model RNN to complete the time migration between subgraphs. Future methods may continue to explore the construction of temporal subgraphs and show solicitude for the relevance between time subgraphs. In addition, more attention may be paid to the static information that existed in TKG, so as to promote the integration of TKGC and traditional KGC methods.

### 5.2. CommonSense Knowledge Graph Completion (CSKGC)

**CommonSense knowledge** is also referred as background knowledge [268], it is a potentially important asset towards building versatile real-world AI applications, such as visual understanding for describing images (e.g., [269–271]), recommendation systems or question answering (e.g., [272–274]). Whereby a novel kind of KGs involve CommonSense knowledge is emerged, **CommonSense knowledge graphs (CSKGs)**, we naturally are interested in the complement of CSKGs, here give a presentation of series **CommonSense Knowledge Graph Completion (CSKGC)** techniques. The corresponding summary table involves described CSKGC methods shown in Table 30.

**CommonSense knowledge graphs (CSKGs)** almost provide a confidence score along with every relation fact, for representing

**Table 30**
Statistics of recent popular CommonSense KGC technologies.

| Model | Technology | Information | Datasets |
|---|---|---|---|
| Language Auxiliary CSKGC Models with Pre-trained Language Models: | | | |
| NAM [64] | Neural Association Model,<br>neural networks:<br>DNN and relation-modulated neural nets (RMNN),<br>probabilistic reasoning,<br>PLMs: skip-gram | Large unstructured texts | CN14 |
| DNN-Bilinear [275] | DNN,<br>Bilinear architecture,<br>averaging the word embeddings (DNN AVG, Bilinear AVG),<br>max pooling of LSTM (DNN LSTM, Bilinear LSTM),<br>PLMs: skip-gram | Text phrases | ConceptNet 100K |
| CSKGC-G [268] | DNN AVG in [275],<br>attention pooling of DNN LSTM,<br>bilinear function,<br>defining CSKG generation task | Text phrases | ConceptNet 100K,<br>JaKB |
| COMET [276] | Automatic CSKG generation,<br>adaptable framework,<br>GPT,<br>multiple transformer blocks of multi-headed attention | CSKG structure and relations | ConceptNet,<br>ATOMIC |
| MCC [277] | End-to-end framework,<br>encoder: GCNs + fine-tuned BERT,<br>decoder: ConvTransE,<br>A progressive masking strategy | Graph structure of local neighborhood,<br>semantic context of nodes in KGs | ConceptNet,<br>ATOMIC |
| CSKGC with Logical Rules: | | | |
| UKGEs [278] | Uncertain KGE,<br>probabilistic soft logic | Structural and uncertainty information<br>of relation facts | ConceptNet,<br>CN15k,<br>NL27k,<br>PPI5k |
| DICE [279] | ILP (Integer linear programming),<br>weighted soft constraints,<br>the theory of reduction costs of a relaxed LP,<br>joint reasoning over CommonSense,<br>knowledge statements sets | CommonSense knowledge statements (four dimensions),<br>taxonomic hierarchy related concepts | ConceptNet,<br>Tuple-KB,<br>Qasimodo |

**Table 31**
ConceptNet tuples with left term "soak in hotspring"; final column is confidence score [275].

| Relation | Right term | conf. |
|---|---|---|
| MOTIVATEDBYGOAL | Relax | 3.3 |
| USEDFOR | Relaxation | 2.6 |
| MOTIVATEDBYGOAL | Your muscle be sore | 2.3 |
| HASPREREQUISITE | Go to spa | 2 |
| CAUSES | Get pruny skin | 1.6 |
| HASPREREQUISITE | Change into swim suit | 1.6 |

the likelihood of the relation fact to be true. Some famous uncertain KGs include ProBase [280], ConceptNet [281] and NELL [282], among which the ConceptNet [281] is a multilingual uncertain KG for CommonSense knowledge that is collected via crowdsourcing [278], and the confidence scores in ConceptNet mainly come from the co-occurrence frequency of the labels in crowdsourced task results. The curated commonsense resource ConceptNet contains tuples consisting of a left term, a relation, and a right term, this form about some examples just like Table 31 shows. The relations come from a fixed set. While terms in Freebase tuples are entities, ConceptNet terms can be arbitrary commonsense phrases. Normally, for the examples in Table 31, a NLP application may wish to query this kind of commonsense phase collections for information about "soaking in a hotspring", but may use distinct words from those contained in the existing tuples.

**Data format:** Facts in CSKGs is often represented in RDF-style triples $(h, r, t)$, where $h$ and $t$ are arbitrary words or phrases, and $r \in R$ is a relation between $h$ and $t$ [268]. Taking triple $(go\ to\ restaurant, subevent, orderfood)$ for an instance, it means a commonsense: "order food" happens as a sub-event of "go to restaurant".

*5.2.1. Commonsense Knowledge Graph Completion*

As the existing CommonSense knowledge in CSKGs is far from sufficient and thorough, it is natural to introduce the **Commonsense Knowledge Graph Completion (CSKGC)** task. While there has been a substantial amount of work on KGC for conventional KGs such as Freebase [277], relatively little work exists for KGC for CSKGs such as ATOMIC [283] and ConceptNet [284].

The work in [285] enters into meaningful discussions with the rationality and possibility of KGC models for *mining* CommonSense knowledge (**CSKM**), through a series of complex analysis about multiple KGC baseline models: the Factorized model, the Prototypical model, and the DNN model, and designs the compared model as the Bilinear model of [275]. They propose a novelty metric to re-evaluate these KGC models aforementioned and analyze splitting candidate triples for the mining task. In a word, the abundant analysis with respect to the potential correlation between existing KGC models and CSKGC task and several first steps towards a more principled evaluation methodology will provide helpful experiences for further exploration about CSKM. More specifically, based on the distinct goals, many researchers identify unique challenges in CSKGC and further investigate effective methods to address these challenges. Here, we will introduce the currently CSKGC methods according to their used technologies in two main categories: **Language Auxiliary CSKGC Models with Pre-trained Language Models** and **CSKGC with Logical Rules** as shown in Table 30.

*5.2.2. Language auxiliary CSKGC models with pre-trained language models*

**Neural association model (NAM)** [64] applies a deep learning framework to model the association between any two events

in a domain by computing a conditional probability of them. The work conducts two case studies to investigate two NAM structures, namely deep neural networks (DNN) and relation-modulated neural nets (RMNN). In the experiment, this work evaluates CSKGC task across ConceptNet CSKG, the results highly appreciated that both DNNs and RMNNs perform equally well and they can significantly outperform the conventional methods available for these reasoning tasks. Moreover, to further prove the effectiveness of the proposed models when reasoning new CommonSense knowledge, the work tries to apply NAMs to solve challenging Winograd Schema (WS) problems and the subsequent experiments performances prove that NAMs have the potential for commonsense reasoning.

**DNN-Bilinear Model** [275] attempts to use bilinear model and DNN for CSKGC study. Specifically, they designs two strategies of both two structures to model commonsense phrases: directly averaging the word embeddings (called *DNN AVG*, *Bilinear AVG*) or using max pooling of LSTM (called *DNN LSTM*, *Bilinear LSTM*). Formally, they define the score function of a triplet $(h, r, t)$ about both bilinear and DNN models respectively as follows:

$$score_{bilinear}(h, r, t) = u_h^T M_r u_t$$

$$u_x = a(W^{(B)} v_x + b^{(B)}), i = h, t$$

and:

$$score_{DNN}(h, r, t) = W^{(D_2)}(a(W^{(D_1)} v_{in} + b^{(D_1)})) + b^{(D_2)}$$

$$v_{in} = concat(v_{ht}, v_r) \in \mathbb{R}^{d_e + d_r}$$

where $v_h, v_t \in \mathbb{R}^{d_e}$ is the vector representing $h$ and $t$, and $v_r$ is the relation embedding. $M_r \in \mathbb{R}^{d_r \times d_r}$ means the parameter matrix for relation $r$, and $v_{ht} \in \mathbb{R}^{d_e}$ is a phrase representation of concatenating $h$ and $t$. The function $a()$ is a nonlinear activation function and the $W^{(B)}, W^{(D_x)}; b^{(B)}, b^{(D_x)}$ $(x = h, t)$ are weight matrix and bias matrix of bilinear model and DNN model, respectively.

**Completion and Generation Model (CSKGC-G)** [268] further improves [275] by replacing the max pooling to attention pooling in DNN LSTM structure and adding a bilinear function, the phrase embedding of $(h, r, t)$ is formulated into:

$$hidden_x^j = BiLSTM(v_x^j, h_{j-1}^i), \ (x = h, t)$$

$$v_x = \sum_{j=1}^{J} \frac{exp(e_j)}{\sum_{K=1}^{J} exp(e_k)} hidden_x^j, \ (x = h, t)$$

$$e^k = w^T tanh(Whidden_x^k), \ (x = h, t)$$

$$v_{ht} = Bilinear(v_h, v_t)$$

$$v_{in} = concat(v_{ht}, v_r)$$

Except for the commonly used variable, the $J$ means the word length of phrase $h$ (or $t$), $w$ is a linear transformation vector for calculating the attention vector. Besides, $v_x^j$ and $hidden_x^j$ are the $j$th word embedding and hidden state of the LSTM for phrase $x$, $(x = h, t)$. Another *highlight* in [268] is that it develops a commonsense knowledge *generation* model which shares information with the CSKGC part, its framework is shown in Fig. 29. This devised model jointly learns the completion and generation tasks, which improves the completion task because triples generated by the generation model can be used as additional training data for the completion model. In this way, this work allows to increase the node size of CSKGs and increase the connectivity of CSKGs.

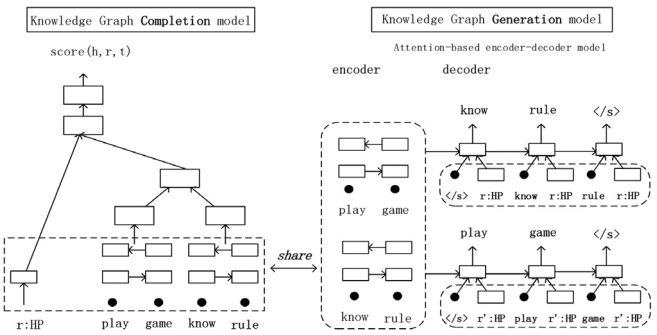



**Fig. 29.** Architecture of CSKGC-G Model. The completion part estimates the score of ($h$ = 'play game'; $r$ = 'HasPrerequisite (HP)'; $t$ = 'know rule'), and the generation module generates $t$ from $(h; r)$ and $h$ from $(t; r')$. $r'$: HP denotes the reverse direction of 'HasPrerequisite' [268].

**COMET** [276] is an automatic generation model for CSKGs. This adaptation framework constructs CSKG by using a seed set of existing knowledge tuples, where contain rich information of KG structure and relations, and operates a large-scale transformer language model (GPT in [242]) with multiple transformer blocks of multi-headed attention among these prepared seed sets to produce CommonSense knowledge tuples.

**Machine Commonsense Completion (MCC)** [277] performs CSKGC by utilizing structure and semantic context of nodes in KGs. CSKGs have significantly sparser and magnitude larger graph structures compared with conventional KGs, therefore it throws a major challenge for general KGC approaches that assume densely connected graphs over a relatively smaller set of nodes. In this work, a joint model is presented with a Graph Convolutional Networks (GCNs) [78] and a fine-tuned BERT [243] model as the encoder side to learn information from the graph structure. ConvTransE [71] is chosen as the decoder side to get a tuple's strong score. As for the encoder process, the GCN model first integrates the representation of a node according to its local neighborhood via the synthetic semantic similarity links, and fine-tune BERT is used to then transfer learning from text to KGs. A progressive masking strategy further ensures that the model appropriately utilizes information from both sources.

### 5.2.3. CSKGC with logical rules

**uncertain KGEs (UKGEs)** [278] explores the uncertain KGE approaches, including CSKGC research. Preserving both structural and uncertainty information of triples in the embedding space, UKGEs learns embeddings according to the confidence scores of uncertain relation facts and further applies probabilistic soft logic to infer confidence scores for unseen relation facts during training.

**Diverse CommonSense Knowledge (DICE)** [279] is a multi-faceted method with weighted soft constraints to couple the inference over concepts (that are related in a taxonomic hierarchy) for deriving refined and expressive CommonSense knowledge. To capture the refined semantics of noisy CommonSense knowledge statements, they consider four dimensions of concept properties: *plausibility*, *typicality*, *remarkability* and *saliency*, and model the coupling of these dimensions by a soft constraint system, which expresses inter-dependencies between the four CommonSense knowledge dimensions with three kinds of logical constraints: *Concept-dimension dependencies*, *Parent–child dependencies* and *Sibling dependencies*, enabling effective and scalable joint reasoning over noisy candidate statements. Note that the

**Table 32**
Statistic of CommonSense Knowledge Graph datasets.

| Dataset | Entity | Relation | Fact | | | |
|---|---|---|---|---|---|---|
| | | | #Train | #Val1 | #Val2 | #Test |
| ATOMIC | 256,570 | 9 | 610,536 | – | – | – |
| CN14 | 159,135 | 14 | 200,198 | 5000 | – | 10,000 |
| JaKB | 18,119 | 7 | 192,714 | 13,778 | – | 13,778 |
| CN-100K | 78,088 | 34 | 100,000 | 1200 | 1200 | 2400 |
| CN15k | 15,000 | 36 | | 241,158 | | |
| NL27 | 27,221 | 404 | | 175,412 | | |
| PPI5k | 4999 | 7 | | 271,666 | | |

**Table 33**
Summary about CSKGC models.

| Model | Completion | Generation |
|---|---|---|
| NAMs [64] | √ | |
| DNN-Bilinear [275] | √ | √ |
| CKGC-G [268] | √ | √ |
| COMET [276] | √ | √ |
| MCC [277] | √ | |
| UKG embedding [278] | √ | |
| Dice [279] | √ | |

over-mentioned reasoning is then cast into an integer linear programming (ILP), and they also leverage the theory of reduction costs of a relaxed LP to compute informative rankings. After experiments on large CommonSense knowledge collections, ConceptNet, TupleKB, and Quasimodo, as long as human judgments, it finally results in a publicly available CSKG containing more than 1.6M statements about 74*k* concepts.

### 5.2.4. Performance analysis of CSKGC models

***Datasets:*** We list some CSKG datasets in Table 32 to show their basic data statistics.

**ConceptNet** As we have introduced before, ConceptNet [284] is a large-scale and multi-lingual CSKG. The evaluation set, which is created from a subset of the whole ConceptNet, consists of data only in English and contains many short phrases including single words [268]. CN14, CN-100K and CN15k are all the subsets of ConceptNet.

**ConceptNet-100K (CN-100K)** [275] contains general commonsense facts about the world. The original version contains the Open Mind Common Sense (OMCS) entries from ConceptNet, whose nodes contain 2.85 words on average. Its dataset splits are shown as Table 32. Following this original splits from the dataset, [277] combines the two provided development sets to create a larger development set, thus the development and test sets consisted of 1200 tuples each.

**CN14** Liu et al. [64] uses the original ConceptNet [286] to construct CN14. When building CN14, they first select all facts in ConceptNet related to 14 typical commonsense relations and then randomly divide the extracted facts into three sets, Train, Dev, and Test. In the end, to create a test set for classification, they randomly switch entities (in the whole vocabulary) from correct triples and get a total of $2\times$*#Test* triples (half are positive samples and half are negative examples).

**CN15k** is a subgraph of ConceptNet, it matches the number of nodes with FB15k [11], and contains 15,000 entities and 241,158 uncertain relation facts in English [278].

**ATOMIC** contains social CommonSense knowledge about day-to-day events [268]. This dataset specifies the effects, requirements, intentions, and attributes of the participants in the event. The average phrase length of nodes (4.40 words) is slightly higher than that of CN-100k, and there may be multiple targets in the source entity and source relation. Tuples in this graph may also contain none targets when the relation type does not need to be annotated. The original dataset segmentation is created to make the seed entity sets between training and evaluation segmentation mutually exclusive. Due to the CSKGC work requires entities to be viewed at least once, [268] creates a new random 80-10-10 partition for the dataset with development set and test set consisting of 87k tuples.

**NL27k** is extracted from NELL [282], an uncertain KG obtained from web-page reading.

**PPI5k** [287] labels the interactions between proteins with the probabilities of occurrence. PPI5k is a subset of STRING, it is a denser graph with fewer entities but more relation facts than NL27 and CN15K.

**Ja-KB** The open-domain Ja-KB (Japanese CommonSense knowledge) is created using crowdsourcing like in Open Mind Common Sense (OMCS) [288] to evaluate the robustness of CSKGC models in terms of the language and long phrases [268]. By limiting the relation types often containing nouns and verbs, Ja-KB owns fewer relation labels than that of ConceptNet. The relation set of Ja-KB including *Causes*, *MotivatedBy*, *Subevent*, *HasPrerequisite*, *ObstructedBy*, *Antonym*, and *Synonym*, and its average length of phrases is longer than in ConceptNet. Since data annotated by crowd workers is usually noisy, the Ja-KB created procedure performed a two-step data collection process to eliminate noisy data, a data creating step, and an evaluation step.

**TupleKB** is extracted from web sources with focus on the science domain, with comparably short and canonicalized triples [279].

**Qasimodo** is a web-extracted general-world CommonSense knowledge collection with focus on saliency [279].

***Analysis about CSKGC models:*** Here we throw out a plain analysis on CSKGC models. The generation models can produce appreciated new explicit knowledge from original diverse and noisy commonsense phrase collections, in general, they are affected by language corpus or pre-trained language models to generalize commonsense language representations, whose target is to add novel nodes and edges to the seed CSKGs. Generative models such as COMET can generate novel knowledge that approaches human performance. This research pointed out a plausible alternative to extractive methods that using generative commonsense models for automatic CSKGC. By comparison, the CSKGC models tend to search potential valid edges in existing CSKGs. An intuitive table is shown in Table 33, which roughly sums up completion and generation models. However, the main finding in [277] about the generative model to completion task is that such generative model cannot easily be re-purposed to rank tuples for KGC, the experimental results as evidence shown in Table 34, this may because of the problems associated with using log-likelihood as an estimate for the truth of a tuple. Nevertheless, generative models such as COMET have several merits. These models possess faster training speed, require lower storage memory, and are transductive naturally. Furthermore, the work in [277] indicates that reasoning models that rely on KGs could favor discriminative approach towards CSKG induction since that would make the graph denser without adding new nodes.

Saito et al. [268] exhibits a shared model that may help promote the CSKGC effect by jointly learning with a generation module, in this case, the generation module can generate augmented

**Table 34**
CommonSense KGC (CSKGC) evaluation on CN-100K and ATOMIC with subgraph sampling [277]. The baselines are presented in the top of the table, the middle part shows the KGC results of COMET and the bottom half are the model implementations in [277].

| Model | CN-100K | | | | ATOMIC | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| DISTMULT | 8.97 | 4.51 | 9.76 | 17.44 | 12.39 | 9.24 | 15.18 | 18.3 |
| COMPLEX | 11.4 | 7.42 | 12.45 | 19.01 | 14.24 | 13.27 | 14.13 | 15.96 |
| CONVE | 20.88 | 13.97 | 22.91 | 34.02 | 10.07 | 8.24 | 10.29 | 13.37 |
| CONVTRANSE | 18.68 | 7.87 | 23.87 | 38.95 | 12.94 | 12.92 | 12.95 | 12.98 |
| COMET-NORMALIZED | 6.07 | 0.08 | 2.92 | 21.17 | 3.36 | 0 | 2.15 | 15.75 |
| COMET-TOTAL | 6.21 | 0 | 0 | 24 | 4.91 | 0 | 2.4 | 21.6 |
| BERT + CONVTRANSE | 49.56 | 38.12 | 55.5 | 71.54 | 12.33 | 10.21 | 12.78 | 16.2 |
| GCN + CONVTRANSE | 29.8 | 21.25 | 33.04 | 47.5 | 13.12 | 10.7 | 13.74 | 17.68 |
| SIM + GCN + CONVTRANSE | 30.03 | 21.33 | 33.46 | 46.75 | 13.88 | 11.5 | 14.44 | 18.38 |
| GCN + BERT + CONVTRANSE | 50.38 | 38.79 | 56.46 | 72.96 | 10.8 | 9.04 | 11.21 | 14.1 |
| SIM + GCN + BERT + CONVTRANSE | 51.11 | 39.42 | 59.58 | 73.59 | 10.33 | 8.41 | 10.79 | 13.86 |

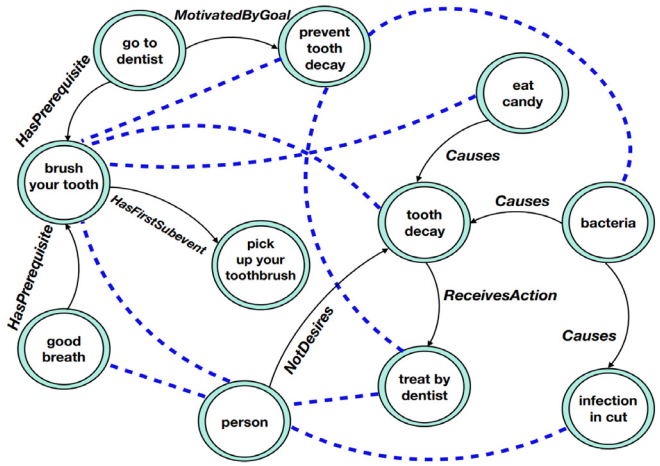



**Fig. 30.** Subgraph from ConceptNet illustrating semantic diversity of nodes, which is represented by non standardized free-form text. Dashed blue lines represent potential edges to be added to the graph [277].

reasonable knowledge to further improve CSKGC. In other words, the loss function of generation module as a good constraint for the CSKGC model.

### 5.2.5. Challenges of CSKGC

As a kind of novel KGs, CSKGs have a series of inherently challenging features:

**1. Resource Scarcity in CSKGs:** Although researchers have developed lots of techniques for acquiring CSKGs from raw text with patterns [289], it has been pointed out that some sorts of knowledge are rarely expressed explicitly in textual corpora [290]. Therefore, researchers have developed curated CSKG resources by manual annotation [281]. Although manually created knowledge has high precision, these resources mostly suffer from coverage shortage [268].

**2. Sparsity of CSKGs:** The key challenge in completing CSKGs is the sparsity of the graphs [277]. Different from traditional KGs, CSKGs are composed of nodes represented by non standardized free-form text, as shown in Fig. 30. For example, nodes "prevent dental caries" and "dental caries" are conceptually related, but not equivalent, so they are represented as different nodes. This conceptual diversity and graphic expressiveness are essential for expressing commonsense, which whereas means that the number of nodes is several orders of magnitude larger, and the graphics are much sparse than traditional KGs. For example, encyclopedias like FB15K-237 [32] owns 100*x* the density of KB than ConceptNet and ATOMIC.

**3. Difficulty to model Uncertain KG using KGE models:** It is a difficult problem to use ordinary KG embedding to obtain uncertain information such as CommonSense knowledge facts [278]. This is a very important task for several reasons. *Firstly*, compared with the deterministic KG embedding, the uncertain KG embedding needs to encode additional confidence information to keep the uncertainty characteristic. *Secondly*, the existing KG embedding models cannot capture the subtle uncertainty of invisible relational facts, because they assume that all invisible relational facts are false beliefs and minimize the credibility measures of relational facts. For uncertain KG embedding learning, one of the main challenges is to correctly estimate the uncertainty of invisible relational facts.

**4. Irrationality of Structural Information in CSKGs:** Another limitation of existing CommonSense knowledge datasets is that they organize statements in a fat, one-dimensional way, and the only rank according to the confidence score [279]. It not only lacks information about whether an attribute is applicable to all or some instances of a concept but also is short of awareness of which attributes are typical and which are prominent from a human point of view. Take an example in [279], the idea that hyenas drink milk (when they were young, all mammals drink milk) is true, but not typical. It is typical for hyenas to eat meat, but it is not obvious that humans will spontaneously name it as a major feature of hyenas. In contrast, the carcass eaten by hyenas is remarkable because it distinguishes hyenas from other African carnivores (such as lions or leopards), which many people would list as a prominent asset. Previous work on CommonSense knowledge has omitted these reference and expression dimensions.

## 5.3. Hyper-Relational Knowledge Graph Completion (HKGC)

Despite existing embedding techniques have obtained promising successes across most commonly KGs, they are all developed based on the *assumption* of a *binary relation* that knowledge data instances each involving *two entities* (such as "Beijing is the capital of China"), such binary relational triples are in the form of (head entity, relation, tail entity). However, a large portion of the knowledge data is from non-binary relations (such as "Benedict Cumberbatch played Alan Turing in the movie The Imitation Game") [291], although these *n*-ary relational facts usually are decomposed into multiple triples via introducing virtual entities, such as the Compound Value Type (CVT) entities in Freebase. For example, in Freebase [5], more than 1/3 of the entities that participate are non-binary relations. Noting that some studies [8] has indicated that the triple-based representation of a KG often

**Table 35**
Statistics of recent popular hyper-relational KGC technologies.

| Model | Hyper-relational fact representation (for n-ary fact$(h, r, t)$ with $(k_i, v_i)$) | Information | Technology | Task |
|---|---|---|---|---|
| m-TransH [291] | $\{(r_h, r_t, k_1, \ldots, k_n), (h, t, v_1, \ldots, v_n)\}$ | n-ary key–value pairs | A direct modeling framework for embedding multifold relations, fact representation recovering, TransH | Predict entities |
| RAE [112] | $\{(r_h, r_t, k_1, \ldots, k_n), (h, t, v_1, \ldots, v_n)\}$ | n-ary key–value pairs | m-TransH, relatedness between entities, instance reconstruction | Predict entities |
| NaLP [292] | $\{r_h : h, r_t : t, k_i : v_i\}$, $i = 1, \ldots, n$ | n-ary key–value pairs | CNN, key–value pairs relatedness | Predict entities predict relations |
| HINGE [8] | $(h, r, t)$, $\{r_h : h, r_t : t, k_i : v_i\}$, $i = 1, \ldots, n$ | Triple data, n-ary key–value pairs | CNN, triple relatedness, key–value pairs relatedness | Predict entities predict relations |

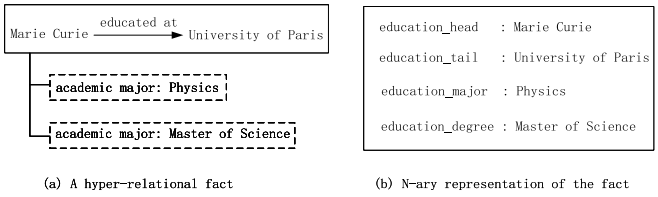



**Fig. 31.** An example of a hyper-relational fact and its corresponding n-ary representation [8].

oversimplifies the complex nature of the data stored in the KG, in particular for hyper-relational data, so that it calls for a necessary investigation of embedding techniques for KGs containing *n*-ary relational data (**HKGs**), we call it Hyper-Relational Knowledge Graph Completion (**HKGC**). Table 35 is a overview of several HKGC technologies introduced in this paper.

### 5.3.1. Definition of facts in hyper-relational KG

Formally, a commonly used representation scheme for HKG's fact transforms a hyper-relational fact into an n-ary representation [112,291,292], i.e., a set of key–value (relation-entity) pairs $r_h : h, r_t : t, k_1 : v_1, \ldots, k_n, v_n$ for the n-ary hyper-relational fact $(h, r, t)$. A simple n-ary fact example and its n-ary representation are shown in Fig. 31. Specifically, by this formula definition, a relation (binary or n-ary relation) is defined by the mappings from a roles sequence corresponding to this type of relation, to their values, and each specific mapping is an instance of this relation [291]. Each hyper-relational fact $(h, r, t)$ with $(k_i, v_i)$, $i = 1, \ldots, n$ is firstly associated with a meta-relation represented as an ordered list of keys (relations), such as $R := (r_h, r_t, k_1, \ldots, k_n)$, the fact is then represented as a list of ordered values associated with the over-mentioned meta-relation as: $\{R, (h, t, v_1, \ldots, v_n)\}$. However, this form of hyper-relational fact pattern (as a set of key–value pairs without triplets) treats each key–value pair in the fact equally, which is not compatible with the schema used by modern KGs [8]. To avoid the wastage of essential information in triples, Rosso et al. [8] decides to preserve the original triple schema of n-ary relational data, i.e., it contains a base triple $(h, r, t)$ and a set of associated key–value pairs $(k_i, v_i)$, $i = 1, \ldots, n$, while a commonly triple fact only contains a triple $(h, r, t)$. In other words, this definition emphasizes the non-negligible characteristic of basic triplet structure even in hyper-relational fact sets.

### 5.3.2. Specific HKGC models

Base on the over-mentioned hyper-relational fact representation, we further discuss the HKGC models meticulously.

**m-TransH** [291] is an earlier work that focuses on HKGs concerning the n-ary relations (so-called multi-fold relations), it models the interaction between entities involved in each fact for predicting missing links in KGs. At the basis of hyper-relational fact definition, according to the translation idea of TransH, m-TransH defines its score function of an instance by the weighted sum of the projection results from its values to its relation hyperplane, in which the weights are the real numbers projected from its roles. However, the primary m-TransH does not take care of the relatedness of the components inside the same n-ary relational fact [292], so that this method does not make full use of the possible inner relative semantic information in the predefined fact structure. On the other hand, since m-TransH learns merely from sets of entities in meta-relations (taking no account of the exact relations in each meta-relation), it can be applied to conduct the link prediction (LP) task for predicting missing entities only.

**RAE** [112] further improves m-TransH by complimentary modeling the relatedness of values, which means the likelihood that two values co-participate in a common instance. The work [112] adds this relatedness loss with a weighted hyper-parameter to the embedding loss of m-TransH and learns the relatedness metric from RAE. When we return to the two issues m-TransH suffered, we find that RAE attempts to solve the first "relatedness modeling" problem by taking the additional modeling of the relatedness of values into account. Although RAE surely achieves favorable performance which outperforms m-TransH, it does not consider the roles explicitly when evaluating the above likelihood [292], whereas roles are also a fundamental aspect for complex relation modeling and taking them into consideration may make a difference because, under different sequences of roles (corresponding to different relations), the relatedness of two values tends to be greatly different. Taking an example from [292], Marie Curie and Henri Becquerel will be taken more related under the role sequence (*person*, *award*, *point in time*, *together with*), than under the role sequence (*person*, *spouse*, *start time*, *end time*, *place of marriage*) due to they won Nobel Prize in Physics in 1903 together.

For the second problem, RAE learns from the pairwise relatedness between entities in each n-ary relational data to perform instance reconstruction, i.e., predicting one or multiple missing entities [8]. Similar to m-TransH, RAE can only be used to perform LP.

**NaLP** [292] whereby designs a relatedness evaluation module to explicitly model the relatedness of the role-value (i.i.e, key–value or relation-entity) pairs involved in the same n-ary relational fact via a neural network pipeline, which supports the prediction of either a missing key (relation) or a missing value (entity). Until now, the above-mentioned two concerned problems are all solved by [292]. In summary, m-TransH, RAE, and NaLP pay attention to the set of key–value pairs of an n-ary fact, resulting in suboptimal models.

**HINGE** [8] aims to directly learn from hyper-relational facts by not only distilling primary structure information from triple data

**Table 36**
Statistic of popular hyper-relational datasets.

| Dataset | Entity | Relation | #Train | | | #Valid | | | #Test | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Binary | N-ary | Overall | Binary | N-ary | Overall | Binary | N-ary | Overall |
| JF17K | 28,645 | 322 | 44,210 | 32,169 | 76,379 | – | – | – | 10,417 | 14,151 | 24,568 |
| WikiPeople1 | 47,765 | 707 | 270,179 | 35,546 | 305,725 | 33,845 | 4378 | 38,223 | 33,890 | 4391 | 38,281 |
| WikiPeople2 | 34839 | 375 | 280520 | 7389 | 287918 | – | – | – | 36597 | 971 | 37586 |

but also extracting further useful information from corresponding key–value pairs simultaneously. HINGE also applies a neural network framework equipped with convolutional structures, just likes the network of [292].

#### 5.3.3. Negative sampling about hyper-relational data

A commonly adopted negative sampling process on HKGs is randomly corrupting one key or value in a true fact. For example, for an n-ary relational fact representation $\{r_h : h, r_t : t, k_i : v_i\}, i = 1, \ldots, n$, when corrupting the key $r_h$ by a randomly sampled $r'_h(r, r')$, the negative fact becomes $\{r'_h : h, r_t : t, k_i : v_i\}, i = 1, \ldots, n$. However, this negative sampling process is not fully adaptable to its n-ary representation of hyper-relational facts, it is unrealistic in especial for keys $r_h$ and $r_t$, as $r'_h$ is not compatible with $r_t$ while only one relation $r$ (or $r'$) can be assumed between $h$ and $t$ in a hyper-relational fact [8]. Therefore, an improved negative sampling method is proposed to fix this issue in [8]. Specifically, when corrupting the key $r_h$ by a randomly sampled $r'_h(r, r')$, the novel negative sampling approach also corrupts $r_t$ by $r'_t$, resulting in a negative fact $\{r'_h : h, r'_t : t, k_i : v_i\}, i = 1, \ldots, n$. Subsequently for this negative fact, only a single relation $r'$ links $h$ and $t$. Similarly, when corrupting $r_t$, we also corrupt $r_h$ in the same way. This new process is more realistic than the original one.

#### 5.3.4. Performance analysis of HKGC models

***Datasets:*** As we have discussed, the hyper-relational data is one natural fact style in KGs. For uniformly modeling and learning, a KG usually is represented as a set of binary relational triples by decomposing n-ary relational facts into multiple triples relying on adding virtual entities, such as in Freebase, a so-called particular "star-to-clique" (S2C) conversion procedure to transform non-binary relational data into binary triplets on filtered Freebase data [291]. Since such procedures have been verified to be irreversible [291], so that it causes a loss of structural information in the multi-fold relations, in other words, this kind of transformed traditional triple datasets are no longer adaptable to n-ary relational fact learning. Therefore the special datasets for HKGs embedding and completion are built as follows:

**JF17K** [291] extracts from Freebase. After removing the entities involved in very few triples and the triples involving String, Enumeration Type, and Numbers, JF17K recovers a fact representation from the remained triples. During fact recovering, it firstly removes facts from meta-relations which have only one single role. Then JF17K randomly selects 10000 facts from each meta-relation containing more than 10000 facts. According to two instance representation strategies, JF17K further constructs two instance representations $T_{id}(F)$ and $T(F)$ where $F$ means the resulting fact representation from previous steps. Next, the final dataset is built by further applying filtering on $T_{id}(F)$ and $T(F)$ into $G$, $G_{id}$, randomly splitting along with original instance representation operation $s2c(G)$. These resulting datasets are uniformly called JF17K, we give their statistics in Table 36.

**WikiPeople** [292] extracts WikiPeople from Wikidata and focuses on entities of type human without any specific filtering to improve the presence of hyper-relational facts. The original WikiPeople dataset version in [292] also contains literals (used as tails) in some facts, Rosso et al. [8] further filters out these non-entity literals and the corresponding facts. Table 36 involves the main statistics of these two versions of WikiPeople datasets. Each of these datasets contains both triple facts and hyper-relational facts.

***Performance Comparison of HKGC Models:*** To get an understanding of the HKGC performance of existing models, we refer to the newest public KGC results for learning from hyper-relational facts in [8] (shown in Table 37). We observe that HINGE [8] consistently outperforms all other models when learning hyper-relational facts, even performs better than the best-performing baseline NaLP-Fix [292], which shows a 13.2% improvement on the link prediction (LP) task, and a 15.1% improvement on the relation prediction (RP) task on WikiPeople (84.1% and 23.8% on JF17K, respectively). Also, from Table 37 we can see NaLP shows better performance than m-TransH and RAE, since it learns the relatedness between relation-entity pairs while m-TransH and RAE learn from entities only.

Moreover, Rosso et al. [8] noted that m-TransH and RAE result in very low performance on WikiPeople, which may be probably due to the weak presence of hyper-relational facts in WikiPeople while m-TransH and RAE are coincidentally designed for hyper-relational facts. Besides, it is obvious that NaLP-Fix (with a fixed negative sampling process) consistently shows better performance compared to NaLP, with a slight improvement of 2.8% in head/tail prediction, and a tremendous improvement of 69.9% in RP on WikiPeople (10.4% and 15.8% on JFK17K, respectively), this result verifies the effectiveness of fixed negative sampling process proposed in [8], in particular for RP.

In addition, the baseline methods learning from hyper-relational facts (i.e., m-TransH, RAE, NaLP and NaLP-Fix) surprisingly yield worse performance in many cases than the best-performing baseline which learns from triples only [8]. They further explain that the ignorance of the triple structure results in this subpar performance, because the triple structure in KGs preserves essential information for KGC.

## 6. Discussion and outlook

### 6.1. Discussion about KGC studies

According to a series of systematic studies about recently KGC works, we discuss several major lights as follows:

1. **About Traditional KGC Models:** With the KGC technology going to be mature, the traditional translation model, decomposition model and neural network model in this field tend to become more and more commonly used as baseline KGC tools to integrate other technologies for promising efficient and effective KGC research.
2. **About Optimization Problem:** It is absolutely necessary to pay attention to the optimization method. A proper optimization method can make it faster or more accurately to get solution. The modeling of optimization objective also determines whether the KGC problem has a global or local optimal solution, or in some cases, it can improve the situation that is easy to fall into the local optimal solution (suboptimal solution), which is not conducive to the KGC task.

**Table 37**
The performance of several HKGC methods on WikiPeople and JF17K [8].

| Method | WikiPeople | | | | | | JF17K | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Head/Tail prediction | | | Relation prediction | | | Head/Tail prediction | | | Relation prediction | | |
| | MRR | Hit@10 | Hit@1 | MRR | Hit@10 | Hit@1 | MRR | Hit@10 | Hit@1 | MRR | Hit@10 | Hit@1 |
| m-TransH | 0.0633 | 0.3006 | 0.0633 | N/A | | | 0.206 | 0.4627 | 0.206 | N/A | | |
| RAE | 0.0586 | 0.3064 | 0.0586 | N/A | | | 0.2153 | 0.4668 | 0.2153 | N/A | | |
| NaLP | 0.4084 | 0.5461 | 0.3311 | 0.4818 | 0.8516 | 0.3198 | 0.2209 | 0.331 | 0.165 | 0.6391 | 0.8215 | 0.5472 |
| NaLP-Fix | 0.4202 | 0.5564 | 0.3429 | 0.82 | 0.9757 | 0.7197 | 0.2446 | 0.3585 | 0.1852 | 0.7469 | 0.8921 | 0.6665 |
| HINGE | 0.4763 | 0.5846 | 0.4154 | 0.95 | 0.9977 | 0.9159 | 0.4489 | 0.6236 | 0.3611 | 0.9367 | 0.9894 | 0.9014 |

3. **About Regularization and Constraints:** During a specific model learning, proper regularization and constraints, as well as the skills of super-parameter tuning can make the trained model achieves unexpected results. Although this is an empirical work step even maybe with potential threatens (for example, $N3$ normalization [50] will require larger embedded dimensions, some optimization techniques (e.g., Tucker [55]) may require a large number of parameters, and thus the resulting scalability or economical issues need to be considered), we should attach important to the model tuning works. Relevant attention has been raised in previous works [50], officially doubting the question that whether the parameters are not adjusted well or the problem of the model itself should be responsible for a bad performance needs to be studied and experimented continuously, emphasizing that model tune-up works are as important as optimization model itself.

4. **About Joint Learning Related to KGC:** We conclude that the joint KGC models that jointly learn distinct components tend to develop their energy function in a composition form. The Joint KGC methods usually extend the original definition of triple energy (distance energy, similarity energy, etc.) to consider the new multimodality representations.

5. **About Information Fusion Strategies:** We also conclude several common experiences here. One of them is that when it comes to the internal combination of the same kind of information (such as collecting useful surrounding graph context as effective as possible for learning the proper neighbor aware representation, the combination between different paths of an entity pair, etc.), attention mechanism along with various neural network structure is an appropriate fusion strategy at the most cases. Moreover, draw lessons from NLP field, RNN structure is suitable for dealing with sequence problems. For example, when considering the path modeling, the general applied neural network structure is RNN [96,166–168], and [163], as well as in the situation that utilizing textual information (especially the long text sequence) for KGC.

6. **Embedding-based Reasoning and Rule-based Reasoning:** As we have introduced and analyzed in our work, both rule-based reasoning and embedding-based reasoning have their separate advantages and disadvantages. Under this case, researchers tend to make the cooperation between these two kinds of KGC models expecting to exert both of their superiorities sufficiently.

### 6.2. Outlook on KGC

We give the following outlooks depending on our observation and overview in this paper:

1. **A Deep-level Interaction is Beneficial for KGC.** In the aspect of adding additional information for KGC, especially those extra information outside KGs, such as the rules and external text resources we mentioned, a peeping research trend is exploring a deep-level interactive learning between external knowledge and internal knowledge. That is, designing a model jointly with a combination of parameter sharing and information circulation, even employing an iterative learning manner to achieve the goal of enriching the knowledge of the internal KG with external information, which in turn feeds back the training information to the encoding side module based on both external information and internal KG's data while continuously replenishing the "knowledge" of the KG.

2. **Rule-based KGC is Promising.** As are introduced in our paper, rule-based approaches perform very well and are a competitive alternative to popular embedding models. For that reason, they have promise to be included as a baseline for the evaluation of KGC methods and it has been recommended that conducting the evaluation on a more fine-grained level is necessary and instructive for further study about KGC field in the future.

3. **Try the New PLMs is Feasible.** Obviously, the endless new pre-training language models (PLMs) make it unlimited possibilities to combine effective language models with various text information for obtaining high-quality embeddings and capturing abundant semantic information to complete KGs.

4. **There is a Plenty of Scopes for Specific-Fields-KGC.** The emergence of new KGs in various specific fields stimulate the completion research on the specific field KGs. Although the existing KGC works concerning the KGs for specific fields and demands is yet relatively rare (for example, there are only a few or a dozen of literature studying the completion of CommonSense KGs and Hyper-Relational KGs), KGC for specific field KGs is exactly meaningful with great practical application value, which will be further developed in the future.

5. **Capture Interaction between Distinct KGs will be Helpful to KGC.** A series of tasks have emerged with the need of interaction between various KGs, such as entity alignment, entity disambiguation, attribute alignment and so on. When it comes to the multi-source knowledge fusion, the research of *heterogeneous graph embedding (HGE)* and *multilingual Knowledge Graph Embedding (MKGE)* has gradually attracted much attention, which are not covered in our current review. KGC under multi-KGs interaction could evolve as a sub-direction for the future development of KGC, which may create some inspiring ideas by studying the unified embedding and completion of different types and structures of knowledge. By the way, the KGC work with respect to multilingual KGs is insufficient, it is worth launching this research direction to replenish the multilingual KGs demanded in real applications.

6. **Select More Proper Modeling Space.** A novel opinion indicates that modeling space of KG embedding does not have to be limited in European space as most literatures do (TransE and its extensions), on the contrary, as KGs possess an intrinsic characteristic of presenting power-law (or scale-free) degree distributions as many other networks [293,294], there have been shown that scale-free networks naturally emerge in the hyperbolic space [295] . Recently, the hyperbolic geometry was exploited in various works [296–298] as a means to provide high-quality embeddings for hierarchical structures instead of in ordinary European space. The work in [295] illustrated that hyperbolic space has the potential to perform significant role in the task of KGC since it offers a natural way to take the KG's topological information into account. This situation inspires researchers to explore more

effective and reasonable embedding vector space for KGC to implement the basic translation transformation or tensor decomposition of entities and relations, the expected model space could be able to easily model complex types of entities and relations, along with various structural information.

7. **Explore the Usage of RL in KGC.** Reinforcement learning (RL) has seen a variety of applications in NLP including machine translation [299], summarization [300], and semantic parsing [301]. Compared to other applications, RL formulations in NLP and KGs tend to have a large action space (e.g., in machine translation and KGC, the space of possible actions is the entire vocabulary of a language and the whole neighbors of an entity, respectively) [302]. On this basis, more recent work formulates multi-hop reasoning as a sequential decision problem, and exploits reinforcement learning (RL) to perform effective path search [63,141, 303,304]. Under normal circumstances, a RL agent is designed to find reasoning paths in the KG, which can control the properties of the found paths rather than using random walks as previous path finding models did. These effective paths not only can be used as an alternative to Path Ranking Algorithm (PRA) in many path-based reasoning methods, but also mainly be treated as reasoning formulas [303]. In particular, some recently studies apply human-defined reward functions, a foreseeable future is to investigate the possibility of incorporating other strategies (such as adversarial learning [27]) to give better rewards than human-defined reward functions. On the other hand, a discriminative model can be trained to give rewards instead of designing rewards according to path characteristics. Additionally, in the future, RL framework can be developed to jointly reason with KG triples and text mentions, which can help to address the problematic scenario when the KG does not have enough reasoning paths.

8. **Multi-task learning about KGC.** Multi-task learning (MTL) [305] is attracting growing attention which inspires that the combined learning of multiple related tasks can outperform learning each task in isolation. With the idea of MTL, KGC can learn and train with other KG-based tasks (or properly designed ancillary tasks) by the MTL framework, which could gain both representability and generalization by sharing the common information between the tasks in the learning process, to achieve overall performance.

## 7. Conclusion

With this overview, we tried to fill a research gap about a systematic and comprehensive introduction of Knowledge Graph Completion (KGC) works and shed new light on the insights gained in previous years. We make up a novel full-view categorization, comparison, and analyzation of research on KGC studies. Specifically, in the high-level, we review KGs in *three major aspects*: KGC merely with internal structural information, KGC with additional information, and other special KGC studies. For the *first* category, KGC is reviewed under Tensor/matrix factorization models, Translation models, and Neural Network models. For the *second* category, we further propose fine-grained taxonomies into two views about the usage of *inside* information of KGs (including node attributes, entity-related information, relation-related information, neighbor information, and relational path information) or *outside* information of KGs (including rule-based KGC and third-party data sources-based KGC). The *third* part pays attention to other special KGC, such as CommonSense KGC, Temporal KGC, and Hyper-relational KGC. In particular, our survey provides a detailed and in-depth comparison and analysis of each KGC category in the fine-grained level and finally gives a global discussion and prospect for the future research directions of KGC. This paper may help researchers grasp the main ideas and results of KGC, and to highlight an ongoing research on them. In the future, we will design a relatively uniform evaluation framework and conduct more detailed experimental evaluations.

## CRediT authorship contribution statement

**Tong Shen:** Classification, Comparisons and analyses, Performance evaluation, Writing – original draft, Revision responses. **Fu Zhang:** Classification, Writing – review & editing, Revision responses. **Jingwei Cheng:** Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors sincerely thank the editors and the anonymous reviewers for their valuable comments and suggestions, which improved the paper. The work is supported by the National Natural Science Foundation of China (61672139) and the Fundamental Research Funds for the Central Universities, China (No. N2216008).

## References

[1] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, et al., Knowledge graphs, 2020, arXiv preprint arXiv:2003.02320.
[2] Wanli Li, Tieyun Qian, Ming Zhong, Xu Chen, Interactive lexical and semantic graphs for semisupervised relation extraction, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[3] Ming Zhong, Yingyi Zheng, Guotong Xue, Mengchi Liu, Reliable keyword query interpretation on summary graphs, IEEE Trans. Knowl. Data Eng. (2022).
[4] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2) (2015) 167–195.
[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.
[6] George A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[7] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, Yago: a core of semantic knowledge, in: WWW, 2007, pp. 697–706.
[8] Paolo Rosso, Dingqi Yang, Philippe Cudré-Mauroux, Beyond triplets: hyper-relational knowledge graph embedding for link prediction, in: The Web Conference, 2020, pp. 1885–1896.
[9] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, Wei Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: ACM SIGKDD KDD, 2014, pp. 601–610.
[10] Antoine Bordes, Xavier Glorot, Jason Weston, Yoshua Bengio, A semantic matching energy function for learning with multi-relational data, Mach. Learn. 94 (2) (2014) 233–259.
[11] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, Oksana Yakhnenko, Translating embeddings for modeling multi-relational data, in: NIPS, 2013, pp. 1–9.
[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu, Learning entity and relation embeddings for knowledge graph completion, in: AAAI, Vol. 29, 2015.
[13] Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, A three-way model for collective learning on multi-relational data, in: ICML, 2011.
[14] Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Ng, Reasoning with neural tensor networks for knowledge base completion, in: NIPS, Citeseer, 2013, pp. 926–934.
[15] Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen, Knowledge graph embedding by translating on hyperplanes, in: AAAI, Vol. 28, 2014.
[16] Quan Wang, Zhendong Mao, Bin Wang, Li Guo, Knowledge graph embedding: A survey of approaches and applications, TKDE 29 (12) (2017) 2724–2743.
[17] Genet Asefa Gesese, Russa Biswas, Mehwish Alam, Harald Sack, A survey on knowledge graph embeddings with literals: Which model links better literal-ly?, 2019.

[18] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, Paolo Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, TKDD 15 (2) (2021) 1–49.
[19] Mayank Kejriwal, Advanced Topic: Knowledge Graph Completion, 2019.
[20] Dat Quoc Nguyen, An overview of embedding models of entities and relationships for knowledge base completion, 2017.
[21] Hongyun Cai, Vincent W. Zheng, Kevin Chen-Chuan Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, TKDE 30 (9) (2018) 1616–1637.
[22] Palash Goyal, Emilio Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowl.-Based Syst. 151 (2018) 78–94.
[23] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, Philip S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, 2020, arXiv preprint arXiv:2002.00388.
[24] Heiko Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semant. Web 8 (3) (2017) 489–508.
[25] Baoxu Shi, Tim Weninger, Open-world knowledge graph completion, in: AAAI, Vol. 32, 2018.
[26] Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, David Ruiz, Generating rules to filter candidate triples for their correctness checking by knowledge graph completion techniques, in: KCAP, 2019, pp. 115–122.
[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, 2014, arXiv preprint arXiv:1406.2661.
[28] Liwei Cai, William Yang Wang, Kbgan: Adversarial learning for knowledge graph embeddings, 2017, arXiv preprint arXiv:1711.04071.
[29] Kairong Hu, Hai Liu, Tianyong Hao, A knowledge selective adversarial network for link prediction in knowledge graph, in: CCF NLPCC, Springer, 2019, pp. 171–183.
[30] Jinghao Niu, Zhengya Sun, Wensheng Zhang, Enhancing knowledge graph completion with positive unlabeled learning, in: ICPR, IEEE, 2018, pp. 296–301.
[31] Yanjie Wang, Rainer Gemulla, Hui Li, On multi-relational link prediction with bilinear models, in: AAAI, Vol. 32, 2018.
[32] Kristina Toutanova, Danqi Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 57–66.
[33] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Convolutional 2d knowledge graph embeddings, in: AAAI, Vol. 32, 2018.
[34] Ke Tu, Peng Cui, Daixin Wang, Zhiqiang Zhang, Jun Zhou, Yuan Qi, Wenwu Zhu, Conditional graph attention networks for distilling and refining knowledge graphs in recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1834–1843.
[35] Baoxu Shi, Tim Weninger, Discriminative predicate path mining for fact checking in knowledge graphs, Knowl.-Based Syst. 104 (2016) 123–133.
[36] Shengbin Jia, Yang Xiang, Xiaojun Chen, Kun Wang, Triple trustworthiness measurement for knowledge graph, in: The World Wide Web Conference, 2019, pp. 2865–2871.
[37] Ivana Balažević, Carl Allen, Timothy M. Hospedales, Tucker: Tensor factorization for knowledge graph completion, 2019, arXiv preprint arXiv:1901.09590.
[38] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, Guillaume Obozinski, A latent factor model for highly multi-relational data, in: NIPS), 2012, pp. 3176–3184.
[39] Alberto Garcia-Duran, Antoine Bordes, Nicolas Usunier, Yves Grandvalet, Combining two and three-way embeddings models for link prediction in knowledge bases, 2015, arXiv preprint arXiv:1506.00999.
[40] Hanxiao Liu, Yuexin Wu, Yiming Yang, Analogical inference for multi-relational embeddings, in: ICML, PMLR, 2017, pp. 2168–2178.
[41] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, Falk Brauer, Random semantic tensor ensemble for scalable knowledge graph link prediction, in: WSDM, 2017, pp. 751–760.
[42] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng, Embedding entities and relations for learning and inference in knowledge bases, 2014, arXiv preprint arXiv:1412.6575.
[43] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, Guillaume Bouchard, Complex embeddings for simple link prediction, in: ICML, PMLR, 2016, pp. 2071–2080.
[44] Seyed Mehran Kazemi, David Poole, Simple embedding for link prediction in knowledge graphs, 2018, arXiv preprint arXiv:1802.04868.
[45] ABM Moniruzzaman, Richi Nayak, Maolin Tang, Thirunavukarasu Balasubramaniam, Fine-grained type inference in knowledge graphs via probabilistic and tensor factorization methods, in: WWW, 2019, pp. 3093–3100.
[46] Sameh K. Mohamed, Nová Vít, TriVec: Knowledge Graph Embeddings for Accurate and Efficient Link Prediction in Real World Application Scenarios.
[47] Rudolf Kadlec, Ondrej Bajgar, Jan Kleindienst, Knowledge base completion: Baselines strike back, 2017, arXiv preprint arXiv:1705.10744.
[48] Hitoshi Manabe, Katsuhiko Hayashi, Masashi Shimbo, Data-dependent learning of symmetric/antisymmetric relations for knowledge base completion, in: AAAI, Vol. 32, 2018.
[49] Boyang Ding, Quan Wang, Bin Wang, Li Guo, Improving knowledge graph embedding using simple constraints, 2018, arXiv preprint arXiv:1805.02408.
[50] Timothée Lacroix, Nicolas Usunier, Guillaume Obozinski, Canonical tensor decomposition for knowledge base completion, in: ICML, PMLR, 2018, pp. 2863–2872.
[51] Koki Kishimoto, Katsuhiko Hayashi, Genki Akai, Masashi Shimbo, Binarized canonical polyadic decomposition for knowledge graph completion, 2019, arXiv preprint arXiv:1912.02686.
[52] Shuai Zhang, Yi Tay, Lina Yao, Qi Liu, Quaternion knowledge graph embeddings, 2019, arXiv preprint arXiv:1904.10281.
[53] Esma Balkir, Masha Naslidnyk, Dave Palfrey, Arpit Mittal, Using pairwise occurrence information to improve knowledge graph completion on large-scale datasets, 2019, arXiv preprint arXiv:1910.11583.
[54] Ankur Padia, Konstantinos Kalpakis, Francis Ferraro, Tim Finin, Knowledge graph fact prediction via knowledge-enriched tensor factorization, J. Web Semant. 59 (2019) 100497.
[55] Ledyard R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[56] Tamara G. Kolda, Brett W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[57] Richard A. Harshman, Models for analysis of asymmetrical relationships among N objects or stimuli, in: First Joint Meeting of the Psychometric Society and the Society of Mathematical Psychology, Hamilton, Ontario, 1978, 1978.
[58] Maximilian Nickel, Lorenzo Rosasco, Tomaso Poggio, Holographic embeddings of knowledge graphs, in: AAAI, Vol. 30, 2016.
[59] Richard A. Harshman, Margaret E. Lundy, PARAFAC: Parallel factor analysis, Comput. Statist. Data Anal. 18 (1) (1994) 39–72.
[60] Daniel D. Lee, H. Sebastian Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[61] Ruslan Salakhutdinov, Nathan Srebro, Collaborative filtering in a non-uniform world: Learning with the weighted trace norm, 2010, arXiv preprint arXiv:1002.2780.
[62] Shmuel Friedland, Lek-Heng Lim, Nuclear norm of higher-order tensors, Math. Comp. 87 (311) (2018) 1255–1281.
[63] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, Andrew McCallum, Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning, 2017, arXiv preprint arXiv:1711.05851.
[64] Quan Liu, Hui Jiang, Andrew Evdokimov, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, Yu Hu, Probabilistic reasoning via deep learning: Neural association models, 2016, arXiv preprint arXiv:1603.07704.
[65] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, Xueqi Cheng, Shared embedding based neural networks for knowledge graph completion, in: ACM CIKM, 2018, pp. 247–256.
[66] Feihu Che, Dawei Zhang, Jianhua Tao, Mingyue Niu, Bocheng Zhao, Parame: Regarding neural network parameters as relation embeddings for knowledge graph completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 2774–2781.
[67] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, Partha Talukdar, Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions, in: AAAI, Vol. 34, 2020, pp. 3009–3016.
[68] Tu Dinh Nguyen Dai Quoc Nguyen, Dat Quoc Nguyen, Dinh Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: NAACL-HLT, 2018, pp. 327–333.
[69] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, A capsule network-based embedding model for knowledge graph completion and search personalization, 2018, arXiv preprint arXiv:1808.04122.
[70] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, Max Welling, Modeling relational data with graph convolutional networks, in: ESWC, Springer, 2018, pp. 593–607.
[71] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, Bowen Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in: AAAI, Vol. 33, 2019, pp. 3060–3067.
[72] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Partha Talukdar, Composition-based multi-relational graph convolutional networks, in: International Conference on Learning Representations, 2019.
[73] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[74] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (ARTICLE) (2011) 2493–2537.
[75] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.
[76] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS 25 (2012) 1097–1105.
[77] Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton, Dynamic routing between capsules, 2017, arXiv preprint arXiv:1710.09829.
[78] Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann LeCun, Spectral networks and locally connected networks on graphs, 2014.
[79] Na Li, Zied Bouraoui, Steven Schockaert, Ontology completion using graph convolutional networks, in: ISWC, Springer, 2019, pp. 435–452.
[80] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams, Convolutional networks on graphs for learning molecular fingerprints, 2015, arXiv preprint arXiv:1509.09292.
[81] Aditya Grover, Aaron Zweig, Stefano Ermon, Graphite: Iterative generative modeling of graphs, in: ICML, PMLR, 2019, pp. 2434–2444.
[82] Thomas N. Kipf, Max Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[83] Xiangyu Song, Jianxin Li, Qi Lei, Wei Zhao, Yunliang Chen, Ajmal Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.
[84] Xiangyu Song, Jianxin Li, Yifu Tang, Taige Zhao, Yunliang Chen, Ziyu Guan, Jkt: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523.
[85] Albert T. Corbett, John R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, User Model. User-Adapt. Interact. 4 (4) (1994) 253–278.
[86] Yaming Yang, Ziyu Guan, Jianxin Li, Wei Zhao, Jiangtao Cui, Quan Wang, Interpretable and efficient heterogeneous graph convolutional network, IEEE Trans. Knowl. Data Eng. (2021).
[87] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, Tianyi Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, Proc. VLDB Endow. 4 (11) (2011) 992–1003.
[88] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, 2014, arXiv preprint arXiv:1406.2661.
[89] Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, Seqgan: Sequence generative adversarial nets with policy gradient, in: AAAI, Vol. 31, 2017.
[90] Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, Yiming Yang, A re-evaluation of knowledge graph completion methods, 2019, arXiv preprint arXiv:1911.03903.
[91] Deepak Nathani, Jatin Chauhan, Charu Sharma, Manohar Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: ACL, 2019, pp. 4710–4723.
[92] Jeffrey Pennington, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
[93] Byungkook Oh, Seungmin Seo, Kyong-Ho Lee, Knowledge graph completion by context-aware convolutional learning with multi-hop neighborhoods, in: ACM CIKM, 2018, pp. 257–266.
[94] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, Maosong Sun, Representation learning of knowledge graphs with entity descriptions, in: AAAI, Vol. 30, 2016.
[95] Minjun Zhao, Yawei Zhao, Bing Xu, Knowledge graph completion via complete attention between knowledge graph and entity descriptions, in: CSAE, 2019, pp. 1–6.
[96] Tehseen Zia, Usman Zahid, David Windridge, A generative adversarial strategy for modeling relation paths in knowledge base representation learning, 2019.
[97] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao, Knowledge graph embedding via dynamic mapping matrix, in: ACL and IJCNLP (Volume 1: Long Papers), 2015, pp. 687–696.
[98] Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, Se-Young Park, A translation-based knowledge graph embedding preserving logical property of relations, in: NAACL: Human Language Technologies, 2016, pp. 907–916.
[99] Kien Do, Truyen Tran, Svetha Venkatesh, Knowledge graph embedding with multiple relation projections, in: ICPR, IEEE, 2018, pp. 332–337.
[100] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, Mark Johnson, STransE: a novel embedding model of entities and relationships in knowledge bases, in: HLT-NAACL, 2016.
[101] Jun Feng, Minlie Huang, Mingdong Wang, Mantong Zhou, Yu Hao, Xiaoyan Zhu, Knowledge graph embedding by flexible translation, in: KR, 2016, pp. 557–560.
[102] Miao Fan, Qiang Zhou, Emily Chang, Fang Zheng, Transition-based knowledge graph embedding with relational mapping properties, in: PACLIC, 2014, pp. 328–337.
[103] Qizhe Xie, Xuezhe Ma, Zihang Dai, Eduard H. Hovy, An interpretable knowledge transfer model for knowledge base completion, in: ACL (1), 2017.
[104] Wei Qian, Cong Fu, Yu Zhu, Deng Cai, Xiaofei He, Translating embeddings for knowledge graph completion with relation attention mechanism, in: IJCAI, 2018, pp. 4286–4292.
[105] Jun Yuan, Neng Gao, Ji Xiang, TransGate: knowledge graph embedding with shared gate structure, in: AAAI, Vol. 33, 2019, pp. 3100–3107.
[106] Xiaofei Zhou, Qiannan Zhu, Ping Liu, Li Guo, Learning knowledge embeddings by combining limit-based scoring loss, in: ACM on CIKM, 2017, pp. 1009–1018.
[107] Mojtaba Nayyeri, Sahar Vahdati, Jens Lehmann, Hamed Shariat Yazdi, Soft marginal transe for scholarly knowledge graph completion, 2019, arXiv preprint arXiv:1904.12211.
[108] Han Xiao, Minlie Huang, Yu Hao, Xiaoyan Zhu, TransA: An adaptive approach for knowledge graph embedding, 2015, arXiv preprint arXiv:1509.05490.
[109] Takuma Ebisu, Ryutaro Ichise, Toruse: Knowledge graph embedding on a lie group, in: AAAI, Vol. 32, 2018.
[110] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, Jian Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in: ICLR, 2018.
[111] Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, Does william shakespeare really write hamlet? knowledge representation learning with confidence, in: AAAI, Vol. 32, 2018.
[112] Richong Zhang, Junpeng Li, Jiajie Mei, Yongyi Mao, Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding, in: WWW, 2018, pp. 1185–1194.
[113] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, Learning structured embeddings of knowledge bases, in: AAAI, Vol. 25, 2011.
[114] Ziqi Zhang, Effective and efficient semantic table interpretation using tableminer+, Semant. Web 8 (6) (2017) 921–957.
[115] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, Bowen Zhou, Orthogonal relation transforms with graph context modeling for knowledge graph embedding, 2019, arXiv preprint arXiv:1911.04910.
[116] Nitin Bansal, Xiaohan Chen, Zhangyang Wang, Can we gain more from orthogonality regularizations in training deep networks? Adv. Neural Inf. Process. Syst. 31 (2018) 4261–4271.
[117] Shengwu Xiong, Weitao Huang, Pengfei Duan, Knowledge graph embedding via relation paths and dynamic mapping matrix, in: International Conference on Conceptual Modeling, Springer, 2018, pp. 106–118.
[118] Alberto Garcia-Duran, Mathias Niepert, Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features, 2017, arXiv preprint arXiv:1709.04676.
[119] T. Yi, L.A. Tuan, M.C. Phan, S.C. Hui, Multi-task neural network for non-discrete attribute prediction in knowledge graphs, in: CIKM'17, 2017.
[120] Yanrong Wu, Zhichun Wang, Knowledge graph embedding with numeric attributes of entities, in: Workshop on RepL4NLP, 2018, pp. 132–136.
[121] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, Zheng Chen, Aligning knowledge and text embeddings by entity descriptions, in: EMNLP, 2015, pp. 267–272.
[122] Ruobing Xie, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Image-embodied knowledge representation learning, 2016, arXiv preprint arXiv:1609.07028.
[123] Hatem Mousselly-Sergieh, Teresa Botschen, Iryna Gurevych, Stefan Roth, A multimodal translation-based approach for knowledge graph representation learning, in: SEM, 2018, pp. 225–234.
[124] Pouya Pezeshkpour, Liyan Chen, Sameer Singh, Embedding multimodal relational data for knowledge base completion, in: EMNLP, 2018.
[125] Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, David S. Rosenblum, MMKG: multi-modal knowledge graphs, in: ESWC, Springer, 2019, pp. 459–474.
[126] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, Asja Fischer, Incorporating literals into knowledge graph embeddings, in: ISWC, Springer, 2019, pp. 347–363.
[127] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, Christopher Meek, Typed tensor decomposition of knowledge bases for relation extraction, in: EMNLP, 2014, pp. 1568–1579.
[128] Denis Krompaß, Stephan Baier, Volker Tresp, Type-constrained representation learning in knowledge graphs, in: ISWC, Springer, 2015, pp. 640–655.
[129] Shiheng Ma, Jianhui Ding, Weijia Jia, Kun Wang, Minyi Guo, Transt: Type-based multiple embedding representations for knowledge graph completion, in: ECML PKDD, Springer, 2017, pp. 717–733.
[130] Alexandros Komninos, Suresh Manandhar, Feature-rich networks for knowledge base completion, in: ACL (Volume 2: Short Papers), 2017, pp. 324–329.
[131] Elvira Amador-Domínguez, Patrick Hohenecker, Thomas Lukasiewicz, Daniel Manrique, Emilio Serrano, An ontology-based deep learning approach for knowledge graph completion with fresh entities, in: DCAI, Springer, 2019, pp. 125–133.

[132] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, Eric Xing, Entity hierarchy embedding, in: ACL and IJCNLP (Volume 1: Long Papers), 2015, pp. 1292–1300.
[133] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, Li Guo, Semantically smooth knowledge graph embedding, in: ACL and IJCNLP (Volume 1: Long Papers), 2015, pp. 84–94.
[134] Jianxin Ma, Peng Cui, Xiao Wang, Wenwu Zhu, Hierarchical taxonomy aware network embedding, in: ACM SIGKDD KDD, 2018, pp. 1920–1929.
[135] Hanie Sedghi, Ashish Sabharwal, Knowledge completion for generics using guided tensor factorization, Trans. Assoc. Comput. Linguist. 6 (2018) 197–210.
[136] Bahare Fatemi, Siamak Ravanbakhsh, David Poole, Improved knowledge graph embedding using background taxonomic information, in: AAAI, Vol. 33, 2019, pp. 3526–3533.
[137] Mikhail Belkin, Partha Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: NIPS, Vol. 14, 2001, pp. 585–591.
[138] Sam T. Roweis, Lawrence K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[139] Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: AAAI, Vol. 30, 2016.
[140] Zhiqiang Geng, Zhongkun Li, Yongming Han, A novel asymmetric embedding model for knowledge graph completion, in: ICPR, IEEE, 2018, pp. 290–295.
[141] Muhao Chen, Yingtao Tian, Xuelu Chen, Zijun Xue, Carlo Zaniolo, On2vec: Embedding-based relation prediction for ontology population, in: SIAM ICDM, SIAM, 2018, pp. 315–323.
[142] Ryo Takahashi, Ran Tian, Kentaro Inui, Interpretable and compositional relation learning by joint training with an autoencoder, in: ACL (Volume 1: Long Papers), 2018, pp. 2148–2159.
[143] Kelvin Guu, John Miller, Percy Liang, Traversing knowledge graphs in vector space, 2015, arXiv preprint arXiv:1506.01094.
[144] Atsushi Suzuki, Yosuke Enokida, Kenji Yamanishi, Riemannian TransE: Multi-relational graph embedding in non-euclidean space, 2018.
[145] Zili Zhou, Shaowu Liu, Guandong Xu, Wu Zhang, On completing sparse knowledge base with transitive relation embedding, in: AAAI, Vol. 33, 2019, pp. 3125–3132.
[146] Charalampos E. Tsourakakis, Fast counting of triangles in large real networks without counting: Algorithms and laws, in: ICDM, IEEE, 2008, pp. 608–617.
[147] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, Mark Johnson, Neighborhood mixture model for knowledge base completion, 2016, arXiv preprint arXiv:1606.06461.
[148] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.
[149] Fanshuang Kong, Richong Zhang, Yongyi Mao, Ting Deng, Lena: Locality-expanded neural embedding for knowledge base completion, in: AAAI, Vol. 33, 2019, pp. 2895–2902.
[150] Trapit Bansal, Da-Cheng Juan, Sujith Ravi, Andrew McCallum, A2n: Attending to neighbors for knowledge graph inference, in: ACL, 2019, pp. 4387–4392.
[151] Peifeng Wang, Jialong Han, Chenliang Li, Rong Pan, Logic attention based neighborhood aggregation for inductive knowledge graph embedding, in: AAAI, Vol. 33, 2019, pp. 7152–7159.
[152] Weidong Li, Xinyu Zhang, Yaqian Wang, Zhihuan Yan, Rong Peng, Graph2Seq: Fusion embedding learning for knowledge graph completion, IEEE Access 7 (2019) 157960–157971.
[153] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhiping Shi, Hui Xiong, Qing He, Relational graph neural network with hierarchical attention for knowledge graph completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 9612–9619.
[154] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, William Yang Wang, One-shot relational learning for knowledge graphs, 2018, arXiv preprint arXiv:1808.09040.
[155] Jiatao Zhang, Tianxing Wu, Guilin Qi, Gaussian metric learning for few-shot uncertain knowledge graph completion, in: International Conference on Database Systems for Advanced Applications, Springer, 2021, pp. 256–271.
[156] Sébastien Ferré, Link prediction in knowledge graphs with concepts of nearest neighbours, in: ESWC, Springer, 2019, pp. 84–100.
[157] Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, David Ruiz, CAFE: Knowledge graph completion using neighborhood-aware features, Eng. Appl. Artif. Intell. 103 (2021) 104302.
[158] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus, End-to-end memory networks, 2015, arXiv preprint arXiv:1503.08895.
[159] Sébastien Ferré, Concepts de plus proches voisins dans des graphes de connaissances, in: 28es Journées Francophones d'Ingénierie des Connaissances IC 2017, 2017, pp. 163–174.
[160] T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.
[161] Antoine Bordes, Xavier Glorot, Jason Weston, Yoshua Bengio, Joint learning of words and meaning representations for open-text semantic parsing, in: Artificial Intelligence and Statistics, PMLR, 2012, pp. 127–135.
[162] Yashen Wang, Yifeng Liu, Huanhuan Zhang, Haiyong Xie, Leveraging lexical semantic information for learning concept-based multiple embedding representations for knowledge graph completion, in: APWeb and WAIM Joint International Conference on Web and Big Data, Springer, 2019, pp. 382–397.
[163] Wenpeng Yin, Yadollah Yaghoobzadeh, Hinrich Schütze, Recurrent one-hop predictions for reasoning over knowledge graphs, in: COLING, 2018, pp. 2369–2378.
[164] Ni Lao, Tom Mitchell, William Cohen, Random walk inference and learning in a large scale knowledge base, in: EMNLP, 2011, pp. 529–539.
[165] Matt Gardner, Tom Mitchell, Efficient and expressive knowledge base completion using subgraph feature extraction, in: EMNLP, 2015, pp. 1488–1498.
[166] Arvind Neelakantan, Benjamin Roth, Andrew McCallum, Compositional vector space models for knowledge base completion, in: ACL and the IJCNLP (Volume 1: Long Papers), 2015, pp. 156–166.
[167] Rajarshi Das, Arvind Neelakantan, David Belanger, Andrew McCallum, Chains of reasoning over entities, relations, and text using recurrent neural networks, in: EACL (1), 2017.
[168] Xiaotian Jiang, Quan Wang, Baoyuan Qi, Yongqin Qiu, Peng Li, Bin Wang, Attentive path combination for knowledge graph completion, in: ACML, PMLR, 2017, pp. 590–605.
[169] Yelong Shen, Po-Sen Huang, Ming-Wei Chang, Jianfeng Gao, Modeling large-scale structured relationships with shared memory for knowledge base completion, in: Workshop on Representation Learning for NLP, 2017, pp. 57–68.
[170] Kai Lei, Jin Zhang, Yuexiang Xie, Desi Wen, Daoyuan Chen, Min Yang, Ying Shen, Path-based reasoning with constrained type attention for knowledge graph completion, Neural Comput. Appl. (2019) 1–10.
[171] Kristina Toutanova, Xi Victoria Lin, Wen-tau Yih, Hoifung Poon, Chris Quirk, Compositional learning of embeddings for relation paths in knowledge base and text, in: ACL (Volume 1: Long Papers), 2016, pp. 1434–1444.
[172] Xixun Lin, Yanchun Liang, Fausto Giunchiglia, Xiaoyue Feng, Renchu Guan, Relation path embedding in knowledge graphs, Neural Comput. Appl. 31 (9) (2019) 5629–5639.
[173] Vivi Nastase, Bhushan Kotnis, Abstract graphs and abstract paths for knowledge graph completion, in: SEM, 2019, pp. 147–157.
[174] Yao Zhu, Hongzhi Liu, Zhonghai Wu, Yang Song, Tao Zhang, Representation learning with ordered relation paths for knowledge graph completion, 2019, arXiv preprint arXiv:1909.11864.
[175] Batselem Jagvaral, Wan-Kon Lee, Jae-Seung Roh, Min-Sung Kim, Young-Tack Park, Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism, Expert Syst. Appl. 142 (2020) 112960.
[176] Tao Zhou, Jie Ren, Matúš Medo, Yi-Cheng Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (4) (2007) 046115.
[177] Matt Gardner, Partha Talukdar, Bryan Kisiel, Tom Mitchell, Improving learning and inference in a large knowledge-base using latent syntactic cues, in: EMNLP, 2013, pp. 833–838.
[178] Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, Tom Mitchell, Incorporating vector space similarity in random walk inference over knowledge bases, in: EMNLP, 2014, pp. 397–406.
[179] Paul J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560.
[180] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP, 2014.
[181] Dave Orr, Amar Subramanya, Evgeniy Gabrilovich, Michael Ringgaard, billion clues in 800 million documents: A web research corpus annotated with freebase concepts. Google Research Blog, 11.
[182] Yadollah Yaghoobzadeh, Hinrich Schütze, Corpus-level fine-grained entity typing using contextual information, 2016, arXiv preprint arXiv:1606.07901.
[183] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need, in: NIPS, 2017.
[184] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, Daisy Zhe Wang, Drum: End-to-end differentiable rule mining on knowledge graphs, 2019, arXiv preprint arXiv:1911.00055.
[185] Quan Wang, Bin Wang, Li Guo, Knowledge base completion using embeddings and rules, in: IJCAI, 2015.

[186] Shangpu Jiang, Daniel Lowd, Dejing Dou, Learning to refine an automatically extracted knowledge base using markov logic, in: ICDM, IEEE, 2012, pp. 912–917.
[187] Jay Pujara, Hui Miao, Lise Getoor, William W. Cohen, Ontology-aware partitioning for knowledge graph identification, in: AKBC Workshop, 2013, pp. 19–24.
[188] Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezny, Steven Schockaert, Ondrej Kuzelka, Lifted relational neural networks: Efficient learning of latent relational structures, J. Artificial Intelligence Res. 62 (2018) 69–100.
[189] Ondřej Kuželka, Jesse Davis, Markov logic networks for knowledge base completion: A theoretical analysis under the MCAR assumption, in: UAI, PMLR, 2020, pp. 1138–1148.
[190] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, Le Song, Efficient probabilistic logic reasoning with graph neural networks, 2020, arXiv preprint arXiv:2001.11850.
[191] Fan Yang, Zhilin Yang, William W. Cohen, Differentiable learning of logical rules for knowledge base reasoning, 2017, arXiv preprint arXiv:1702.08367.
[192] Pouya Ghiasnezhad Omran, Kewen Wang, Zhe Wang, Scalable rule learning via learning representation, in: IJCAI, 2018, pp. 2149–2155.
[193] Tim Rocktäschel, Deep prolog: End-to-end differentiable proving in knowledge bases, in: AITP 2017, 2017, p. 9.
[194] Pasquale Minervini, Matko Bosnjak, Tim Rocktäschel, Sebastian Riedel, Towards neural theorem proving at scale, 2018, arXiv preprint arXiv:1807.08204.
[195] Zhuoyu Wei, Jun Zhao, Kang Liu, Zhenyu Qi, Zhengya Sun, Guanhua Tian, Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances, in: CIKM, 2015, pp. 1331–1340.
[196] William Yang Wang, William W. Cohen, Learning first-order logic embeddings via matrix factorization, in: IJCAI, 2016, pp. 2132–2138.
[197] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, Li Guo, Jointly embedding knowledge graphs and logical rules, in: EMNLP, 2016, pp. 192–202.
[198] Pengwei Wang, Dejing Dou, Fangzhao Wu, Nisansa de Silva, Lianwen Jin, Logic rules powered knowledge graph embedding, 2019, arXiv preprint arXiv:1903.03772.
[199] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, Li Guo, Knowledge graph embedding with iterative guidance from soft rules, in: AAAI, Vol. 32, 2018.
[200] Vinh Thinh Ho, Daria Stepanova, Mohamed Hassan Gad-Elrab, Evgeny Kharlamov, Gerhard Weikum, Learning rules from incomplete kgs using embeddings, in: ISWC, ceur. ws. org, 2018.
[201] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, Huajun Chen, Iteratively learning embeddings and rules for knowledge graph reasoning, in: WWW, 2019, pp. 2366–2377.
[202] Meng Qu, Jian Tang, Probabilistic logic neural networks for reasoning, 2019, arXiv preprint arXiv:1906.08495.
[203] Jianfeng Du, Jeff Z. Pan, Sylvia Wang, Kunxun Qi, Yuming Shen, Yu Deng, Validation of growing knowledge graphs by abductive text evidences, in: AAAI, Vol. 33, 2019, pp. 2784–2791.
[204] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, Heiner Stuckenschmidt, Anytime bottom-up rule learning for knowledge graph completion, in: IJCAI, 2019, pp. 3137–3143.
[205] Jiangtao Ma, Yaqiong Qiao, Guangwu Hu, Yanjun Wang, Chaoqin Zhang, Yongzhong Huang, Arun Kumar Sangaiah, Huaiguang Wu, Hongpo Zhang, Kai Ren, ELPKG: A high-accuracy link prediction approach for knowledge graph completion, Symmetry 11 (9) (2019) 1096.
[206] Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, Xiaowei Zhang, Rule-guided compositional representation learning on knowledge graphs, in: AAAI, Vol. 34, 2020, pp. 2950–2958.
[207] Luis Galárraga, Christina Teflioudi, Katja Hose, Fabian M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE +, VLDB J. 24 (6) (2015) 707–730.
[208] Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, Heiner Stuckenschmidt, Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion, in: ISWC, Springer, 2018, pp. 3–20.
[209] Yang Chen, Sean Goldberg, Daisy Zhe Wang, Soumitra Siddharth Johri, Ontological pathfinding, in: International Conference on Management of Data, 2016, pp. 835–846.
[210] Tim Rocktäschel, Matko Bosnjak, Sameer Singh, Sebastian Riedel, Low-dimensional embeddings of logic, in: ACL 2014 Workshop on Semantic Parsing, 2014, pp. 45–49.
[211] Tim Rocktäschel, Sameer Singh, Sebastian Riedel, Injecting logical background knowledge into embeddings for relation extraction, in: NAACL: Human Language Technologies, 2015, pp. 1119–1129.
[212] Stephen Muggleton, et al., Stochastic logic programs, in: Advances in Inductive Logic Programming, Vol. 32, Citeseer, 1996, pp. 254–264.
[213] Stephen Muggleton, Inductive Logic Programming, Vol. 38, Morgan Kaufmann, 1992.
[214] Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, Chris Meek, Jennifer Neville, et al., Introduction to Statistical Relational Learning, MIT Press, 2007.
[215] Matthew Richardson, Pedro Domingos, Markov logic networks, Mach. Learn. 62 (1–2) (2006) 107–136.
[216] William Yang Wang, Kathryn Mazaitis, William W. Cohen, Programming with personalized pagerank: a locally groundable first-order probabilistic logic, in: CIKM, 2013, pp. 2129–2138.
[217] Arvind Neelakantan, Quoc V. Le, Martin Abadi, Andrew McCallum, Dario Amodei, Learning a natural language interface with neural programmer, 2016, arXiv preprint arXiv:1611.08945.
[218] Arvind Neelakantan, Quoc V. Le, Ilya Sutskever, Neural programmer: Inducing latent programs with gradient descent, 2015, arXiv preprint arXiv:1511.04834.
[219] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, Dan Klein, Learning to compose neural networks for question answering, 2016, arXiv preprint arXiv:1601.01705.
[220] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al., Hybrid computing using a neural network with dynamic external memory, Nature 538 (7626) (2016) 471–476.
[221] William W. Cohen, Tensorlog: A differentiable deductive database, 2016, arXiv preprint arXiv:1605.06523.
[222] Leonid Boytsov, Bilegsaikhan Naidan, Engineering efficient and effective non-metric space library, in: SISAP, Springer, 2013, pp. 280–293.
[223] Yu A. Malkov, Dmitry A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, PAMI 42 (4) (2018) 824–836.
[224] Richard Evans, Edward Grefenstette, Learning explanatory rules from noisy data, J. Artificial Intelligence Res. 61 (2018) 1–64.
[225] Guillaume Bouchard, Sameer Singh, Theo Trouillon, On approximate reasoning capabilities of low-rank vector spaces, in: AAAI Spring Symposia, Citeseer, 2015.
[226] Baoxu Shi, Tim Weninger, ProjE: Embedding projection for knowledge graph completion, in: AAAI, Vol. 31, 2017.
[227] Han Xiao, Minlie Huang, Lian Meng, Xiaoyan Zhu, SSP: semantic space projection for knowledge graph embedding with text descriptions, in: AAAI, Vol. 31, 2017.
[228] Xu Han, Zhiyuan Liu, Maosong Sun, Neural knowledge acquisition via mutual attention between knowledge graph and text, in: AAAI, Vol. 32, 2018.
[229] Paolo Rosso, Dingqi Yang, Philippe Cudré-Mauroux, Revisiting text and knowledge graph joint embeddings: The amount of shared information matters!, in: 2019 IEEE Big Data, IEEE, 2019, pp. 2465–2473.
[230] Bo An, Bo Chen, Xianpei Han, Le Sun, Accurate text-enhanced knowledge graph representation learning, in: NAACL: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 745–755.
[231] Teng Long, Ryan Lowe, Jackie Chi Kit Cheung, Doina Precup, Leveraging lexical resources for learning entity embeddings in multi-relational data, in: ACL (2), 2016.
[232] Miao Fan, Qiang Zhou, Thomas Fang Zheng, Ralph Grishman, Distributed representation learning for knowledge graphs with entity descriptions, Pattern Recognit. Lett. 93 (2017) 31–37.
[233] Jiacheng Xu, Xipeng Qiu, Kan Chen, Xuanjing Huang, Knowledge graph representation with jointly structural and textual encoding, in: IJCAI, 2017.
[234] Michael Cochez, Martina Garofalo, Jérôme Lenßen, Maria Angela Pellegrino, A first experiment on including text literals in KGloVe, 2018, arXiv preprint arXiv:1807.11761.
[235] Nada Mimouni, Jean-Claude Moissinac, Anh Vu, Knowledge base completion with analogical inference on context graphs, in: Semapro 2019, 2019.
[236] Liang Yao, Chengsheng Mao, Yuan Luo, KG-BERT: BERT for knowledge graph completion, 2019, arXiv preprint arXiv:1909.03193.
[237] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, Jian Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, 2019, arXiv preprint arXiv:1911.06136.
[238] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.
[239] Daniel Daza, Michael Cochez, Paul Groth, Inductive entity representations from text via link prediction, in: Proceedings of the Web Conference 2021, 2021, pp. 798–808.
[240] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, Yi Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, 2021, pp. 1737–1748.

[241] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, Deep contextualized word representations, 2018, arXiv preprint arXiv:1802.05365.
[242] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, Improving language understanding by generative pre-training, 2018.
[243] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[244] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, 2019, arXiv preprint arXiv:1906.08237.
[245] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, Distributed representations of words and phrases and their compositionality, 2013, arXiv preprint arXiv:1310.4546.
[246] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need, 2017, arXiv preprint arXiv:1706.03762.
[247] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, Qun Liu, ERNIE: Enhanced language representation with informative entities, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1441–1451.
[248] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, Zheng Zhang, CoLAKE: Contextualized language and knowledge embedding, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 3660–3670.
[249] Boran Hao, Henghui Zhu, Ioannis Paschalidis, Enhancing clinical bert embedding using a biomedical knowledge base, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 657–661.
[250] Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, Sameer Singh, Barack's wife hillary: Using knowledge graphs for fact-aware language modeling, in: ACL, 2019, pp. 5962–5971.
[251] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, Noah A. Smith, Knowledge enhanced contextual word representations, in: EMNLP-IJCNLP, 2019, pp. 43–54.
[252] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, Zhifang Sui, Encoding temporal information for time-aware link prediction, in: EMNLP, 2016, pp. 2350–2354.
[253] Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, Pascal Poupart, Diachronic embedding for temporal knowledge graph completion, in: AAAI, Vol. 34, 2020, pp. 3988–3995.
[254] Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, Jens Lehmann, Temporal knowledge graph completion based on time series Gaussian embedding, in: ISWC, Springer, 2020, pp. 654–671.
[255] Julien Leblay, Melisachew Wudage Chekol, Deriving validity time in knowledge graph, in: Companion Proceedings of the the Web Conference 2018, 2018, pp. 1771–1776.
[256] Shib Sankar Dasgupta, Swayambhu Nath Ray, Partha Talukdar, Hyte: Hyperplane-based temporally aware knowledge graph embedding, in: EMNLP, 2018, pp. 2001–2011.
[257] Yunpu Ma, Volker Tresp, Erik A. Daxberger, Embedding models for episodic knowledge graphs, J. Web Semant. 59 (2019) 100490.
[258] Alberto García-Durán, Sebastijan Dumančić, Mathias Niepert, Learning sequence encoders for temporal knowledge graph completion, 2018, arXiv preprint arXiv:1809.03202.
[259] Timothée Lacroix, Guillaume Obozinski, Nicolas Usunier, Tensor decompositions for temporal knowledge base completion, 2020, arXiv preprint arXiv:2004.04926.
[260] Rakshit Trivedi, Hanjun Dai, Yichen Wang, Le Song, Know-evolve: Deep temporal reasoning for dynamic knowledge graphs, in: ICML, PMLR, 2017, pp. 3462–3471.
[261] Woojeong Jin, He Jiang, Meng Qu, Tong Chen, Changlin Zhang, Pedro Szekely, Xiang Ren, Recurrent event network: Global structure inference over temporal knowledge graph, 2019, arXiv preprint arXiv:1904.05530.
[262] Zhen Han, Yuyi Wang, Yunpu Ma, Stephan Guünnemann, Volker Tresp, The graph hawkes network for reasoning on temporal knowledge graphs, 2020, arXiv preprint arXiv:2003.13432.
[263] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, William L Hamilton, TeMP Temporal Message Passing for Temporal Knowledge Graph Completion, 2020, arXiv preprint arXiv:2010.03526.
[264] Michael D Ward, Andreas Beger, Josh Cutler, Matthew Dickenson, Cassy Dorff, Ben Radford, Comparing GDELT and ICEWS event data, Analysis 21 (1) (2013) 267–297.
[265] Aaron Schein, John Paisley, David M. Blei, Hanna Wallach, Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts, in: ACM SIGKDD KDD, 2015, pp. 1045–1054.
[266] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Gerhard Weikum, YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia, Artificial Intelligence 194 (2013) 28–61.
[267] Farzaneh Mahdisoltani, Joanna Biega, Fabian Suchanek, Yago3: A knowledge base from multilingual wikipedias, in: CIDR, CIDR Conference, 2014.
[268] Itsumi Saito, Kyosuke Nishida, Hisako Asano, Junji Tomita, Commonsense knowledge base completion and generation, in: CoNLL, 2018, pp. 141–150.
[269] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Vqa: Visual question answering, in: ICCV, 2015, pp. 2425–2433.
[270] Andrej Karpathy, Li Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: CVPR, 2015, pp. 3128–3137.
[271] Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, Jason Weston, Engaging image captioning via personality, in: IEEE/CVF CVPR, 2019, pp. 12516–12526.
[272] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, Richard Socher, Explain yourself! leveraging language models for commonsense reasoning, 2019, arXiv preprint arXiv:1906.02361.
[273] Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, Xu Sun, Enhancing topic-to-essay generation with external commonsense knowledge, in: ACL, 2019, pp. 2002–2012.
[274] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, Minlie Huang, Augmenting end-to-end dialogue systems with commonsense knowledge, in: AAAI, Vol. 32, 2018.
[275] Xiang Li, Aynaz Taheri, Lifu Tu, Kevin Gimpel, Commonsense knowledge base completion, in: ACL (Volume 1: Long Papers), 2016, pp. 1445–1455.
[276] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, Yejin Choi, Comet: Commonsense transformers for automatic knowledge graph construction, 2019, arXiv preprint arXiv:1906.05317.
[277] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, Yejin Choi, Commonsense knowledge base completion with structural and semantic context, in: AAAI, Vol. 34, 2020, pp. 2925–2933.
[278] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, Carlo Zaniolo, Embedding uncertain knowledge graphs, in: AAAI, Vol. 33, 2019, pp. 3363–3370.
[279] Yohan Chalier, Simon Razniewski, Gerhard Weikum, Joint reasoning for multi-faceted commonsense knowledge, 2020, arXiv preprint arXiv:2001.04170.
[280] Wentao Wu, Hongsong Li, Haixun Wang, Kenny Q. Zhu, Probase: A probabilistic taxonomy for text understanding, in: ACM SIGMOD International Conference on Management of Data, 2012, pp. 481–492.
[281] Robyn Speer, Joshua Chin, Catherine Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: AAAI, Vol. 31, 2017.
[282] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhanava Dalvi, Matt Gardner, Bryan Kisiel, et al., Never-ending learning, Commun. ACM 61 (5) (2018) 103–115.
[283] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, Yejin Choi, Atomic: An atlas of machine commonsense for if-then reasoning, in: AAAI, Vol. 33, 2019, pp. 3027–3035.
[284] Robert Speer, Catherine Havasi, ConceptNet 5: A large semantic network for relational knowledge, in: The People's Web Meets NLP, Springer, 2013, pp. 161–176.
[285] Jastrzebski Stanislaw, Dzmitry Bahdanau, Seyedarian Hosseini, Michael Noukhovitch, Yoshua Bengio, Jackie Chi Kit Cheung, Commonsense mining as knowledge base completion? A study on the impact of novelty, 2018, arXiv preprint arXiv:1804.09259.
[286] Hugo Liu, Push Singh, ConceptNet—a practical commonsense reasoning tool-kit, BT Technol. J. 22 (4) (2004) 211–226.
[287] Damian Szklarczyk, John H. Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T. Doncheva, Alexander Roth, Peer Bork, et al., The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible, Nucleic Acids Res. (2016) gkw937.
[288] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, Wan Li Zhu, Open mind common sense: Knowledge acquisition from the general public, in: OTM Confederated International Conferences" on the Move to Meaningful Internet Systems", Springer, 2002, pp. 1223–1237.
[289] Gabor Angeli, Christopher D. Manning, Philosophers are mortal: Inferring the truth of unseen facts, in: CoNLL, 2013, pp. 133–142.
[290] Jonathan Gordon, Benjamin Van Durme, Reporting bias and knowledge acquisition, in: Workshop on AKBC, 2013, pp. 25–30.
[291] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, Richong Zhang, On the representation and embedding of knowledge bases beyond binary relations, in: IJCAI, 2016, pp. 1300–1307.
[292] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, Xueqi Cheng, Link prediction on n-ary relational data, in: WWW, 2019, pp. 583–593.
[293] Michalis Faloutsos, Petros Faloutsos, Christos Faloutsos, On power-law relationships of the internet topology, in: The Structure and Dynamics of Networks, Princeton University Press, 2011, pp. 195–206.

[294] Mark Steyvers, Joshua B. Tenenbaum, The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth, Cogn. Sci. 29 (1) (2005) 41–78.
[295] Prodromos Kolyvakis, Alexandros Kalousis, Dimitris Kiritsis, Hyperkg: Hyperbolic knowledge graph embeddings for knowledge base completion, 2019, arXiv preprint arXiv:1908.04895.
[296] Maximillian Nickel, Douwe Kiela, Learning continuous hierarchies in the lorentz model of hyperbolic geometry, in: ICML, PMLR, 2018, pp. 3779–3788.
[297] Octavian Ganea, Gary Bécigneul, Thomas Hofmann, Hyperbolic entailment cones for learning hierarchical embeddings, in: ICML, PMLR, 2018, pp. 1646–1655.
[298] Frederic Sala, Chris De Sa, Albert Gu, Christopher Ré, Representation tradeoffs for hyperbolic embeddings, in: ICML, PMLR, 2018, pp. 4460–4469.
[299] Sumit Chopra Marc'Aurelio Ranzato, Michael Auli, Wojciech Zaremba, Sequence level training with recurrent neural networks, 2015, CoRR abs/1511.06732.
[300] Romain Paulus, Caiming Xiong, Richard Socher, A deep reinforced model for abstractive summarization, 2017, arXiv preprint arXiv:1705.04304.
[301] Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, Percy Liang, From language to programs: Bridging reinforcement learning and maximum marginal likelihood, 2017, arXiv preprint arXiv:1704.07926.
[302] Peng Lin, Qi Song, Yinghui Wu, Fact checking in knowledge graphs with ontological subgraph patterns, Data Sci. Eng. 3 (4) (2018) 341–358.
[303] Wenhan Xiong, Thien Hoang, William Yang Wang, Deeppath: A reinforcement learning method for knowledge graph reasoning, 2017, arXiv preprint arXiv:1707.06690.
[304] Yelong Shen, Jianshu Chen, Po-Sen Huang, Yuqing Guo, Jianfeng Gao, Reinforcewalk: Learning to walk in graph with monte carlo tree search, 2018.
[305] Rich Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.