## KG Course South-East

https://github.com/npubird/KnowledgeGraphCourse

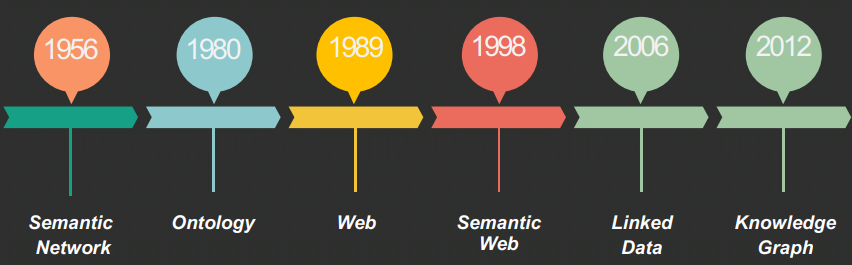
#### Lecture ONE (intro):

Ontology: An ontology is an explicit specification of a conceptualization.

— [Tom Gruber](https://en.wikipedia.org/wiki/Tom_Gruber), A Translation Approach to Portable Ontology Specifications

Semantic Web: Data + link

A process



Some famous KG:

Cyc: Concepts and facts such as “Every tree is a plant”

OpenCyc: A open source of Cyc

WordNet: Dictionary Knowledge Graph, useful in disambiguation. Has defined the semantic relationship between Noun. Verb. Adj. and Adv.

ConceptNet: Including DBPedia, Wikinary and Wordnet, use more natural language.

Freebase: owned by Google, multilanguage

Wikidata: editable, entity based

DBPedia: from Wikipedia, more strict, use RDF format

YAGO: including Wikipedia, WordNet and GeoNames, considered time and space

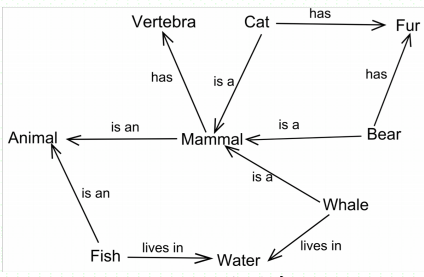
Others: NELL(self learning), Zhishi(Chinese),

Famous Companies: Palantir(location of [Bin](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [Laden](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;)), IBM waston(Jeopardy) Goolge KG(Searching Engine)

#### Lecture TWO (knowledge representation):

1. Semantic Network: use nodes and edges

For example,



Many others: Production Systems(If and Then), Frame System(Slot and Facet), Description Logic(Concept, Role, Individual, Axiom)

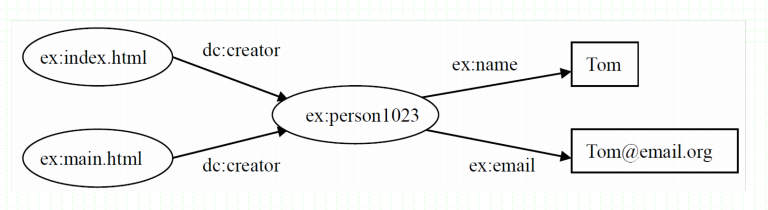
1. RDF (Resource Description Framework):

Structure: use triple graph model ( has Resource, Literal, Property, Statement)

Lexicon: URI

Language: XML

Its format,



Its schema,

Rdfs(Class, subclassOf, type, range, domain, subPropertyOF)

1. OWL(Web ontology Language):

Expand RDFs

Language: XML/RDFs

New::OWL2

1. Others:PSARQL, JSON-LD

**Lecture THERE(knowledge modeling):**

1. Manually Building

1,define a goal and sphere

2,try use available resources

3,define important terms

4,define class and class inherit

5,define attribute and relation

6,define range and restriction

7,building example

1. Machine learning based

1,entropy based inference learning

Tool:Protege

2,others

**Lecture FOUR(knowledge extraction):**

Information extraction: to get structured information

Knowledge extraction: to get machine-readable machine-readable and machine-interpretable knowledge

A, from relation database (structured)

1, Principles...

2, Method: Direct Mapping, R2RML

3, Tools: D2R, Virtuoso...

B, from Encyclopedia (semi-structured)

1, A format pattern, including abstract, infobox, category and page link...

2, from YAGO (infobox rules)

C, Unstructured data

1, key elements: data collection, entity identification , relation extract, event extract

2, Data collection: web scarping, manually collection

3, Entity identification:

3A, Rules and Dictionary based

3B, Machine learning based( Hidden Markov Model, Conditional Random Fields(CRF)...)

3C, Deeplearning based(CNN + CRF, Bi-LSTM + CRF)

3D, Many others: semi-supervised, inference learning, pre-training...

4, Relation extraction:

4A, Background

(Syntactic relations)

Relation of position: Syntagmatic Relations, Horizontal or Chain Relations

Relations of substitutability: Paradigmatic Relations, Vertical or Associative

Relations of co-occurrence: substitute of a group of words

(semantic relations)

Noun compounds

...

4B, Database

WordNet...

4C, Instance based relation extraction

High precision, domain-specific

4D, Many others

Supervised: Eigenvectors, kernel classify, Sequence labeling

Semi-supervised, Word-Embedding...

5,Event extraction

5A, Process:

Event Trigger Detection and Typing - Event Argument and Role Identification

5B, Rules and Instance based：

Define a semantic and word pattern for a specific domain

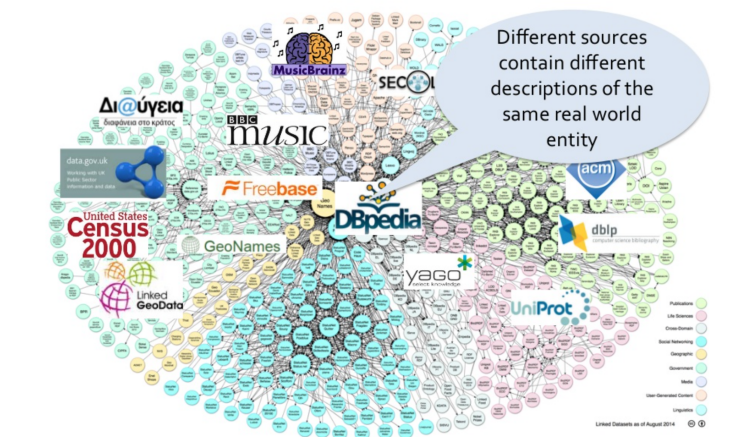
5C, Machine Learning based:

Classification Method, NLP...

5D, Deep-learning based:

Pipeline, Joint Model...

**Lecture FIVE(Knowledge Integration):**



1. Ontology Matching

1, Basis matching

(use least edition distance, or similarity. Use N-Gram to split the text)

2, Text matching

(the importance of a word(TF-TDF))

3, Structure matching

(Converted to vectors.Considering neighbors, context, ext... Use similarity propagation)

4, Representation learning

(Converted to vectors. Use machine learning based method. Tool: Lily)

1. Instance Matching

(==Entity==Record==Object==)

1, Challenging

(Polysemy and Synonym)

2, Rules based matching

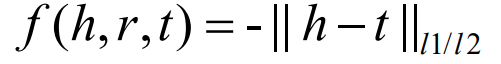
(use ontology semantic to set rules)

3,learning based

4,Blocking

**Lecture SIX(Representation learning):**

(Transfer entity and relationship into low and dense vector, as a representation)

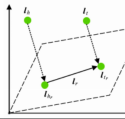
1. Distance based Model

UM

SE

1. Translation based Model

TransE



TransH(project into a hyper-plane)

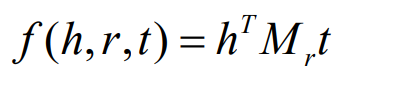
TransR(embed into matrices)

TransD(individually embed entity and relation)

TranSparse(use sparse matrices to deal with the imbalance)

TransM(use weights)

Others: ManiFoldE, TransF, TransA, KG2E, TransG,

1. Semantic Matching Model

Latent Factor Model(LFM)

DistMUlt(LFM in diagonal matrices)

ComplEX(use complex number)

ANALOGY(use analogy method)

Others:RESCAL, HolE, SLM, SME, NTN, MLP, NAM, ConvE

1. Multi-sources and others Models

ILP, GAKE, TransC,

1. Performance evaluations

MRR: Ave( 1 / correct\_labeled\_Instance)

Hits@N: Proportion\_of( correct\_labeled\_Instance < Num\_N )

**Lecture SEVEN(Knowledge Storage):**

1. Graph Data Management System (Graph DBMS)

A1, Property Graph:

Node(have labels)

Edge(have direction and a type)

A2, Tools:

Neo4j(most popular, query language Cypher,declarative )

JanusGraph(most flexible, query language Gremlin, procedural)

ArangoDB

...

1. RDF stores

(from W3C, academic)

B1, Its Property:

Three elements: Subject, Predicate, Object

Statements(a triple), Description(triples)

Strict Semantic Web standard follower

Query language: SPARQL

B2, Tools:

Virtuoso, RDF4J

1. Relational DBMS

Horizontal table, Vertical table...

Too much join operation, expensive to look though

**Lecture EIGHT(Entity linking)**

**Lecture NINE(Knowledge Reasoning)**

## A Survey on Knowledge Graphs: Representation, Acquisition and Applications

https://arxiv.org/pdf/2002.00388.pdf

**Abstract**—Human knowledge provides a formal understanding of the world. Knowledge graphs that represent structural relations between entities have become an increasingly popular research direction towards cognition and human-level intelligence. In this survey, we provide a comprehensive review of knowledge graph covering overall research topics about 1) knowledge graph representation learning, 2) knowledge acquisition and completion, 3) temporal knowledge graph, and 4) knowledge-aware applications, and summarize recent breakthroughs and perspective directions to facilitate future research. We propose a full-view categorization and new taxonomies on these topics. Knowledge graph embedding is organized from four aspects of representation space, scoring function, encoding models, and auxiliary information. For knowledge acquisition, especially knowledge graph completion, embedding methods, path inference, and logical rule reasoning, are reviewed. We further explore several emerging topics, including meta relational learning, commonsense reasoning, and temporal knowledge graphs. To facilitate future research on knowledge graphs, we also provide a curated collection of datasets and open-source libraries on different tasks. In the end, we have a thorough outlook on several promising research directions

1. **INTRODUCTION**

One definition,

A knowledge graph is a structured representation of facts, consisting of entities, relationships, and semantic descriptions

Some examples,

Examples of knowledge base and knowledge graph are illustrated in Fig. 1. Knowledge can be expressed in a factual triple in the form of (head, relation, tail) or (subject, predicate, object) under the resource description framework (RDF), for example, (Albert Einstein, WinnerOf, Nobel Prize)

Abbrvations,

Recent advances in knowledge-graph-based research focus on knowledge representation learning (KRL) or knowledge graph embedding (KGE) by mapping entities and relations into low-dimensional vectors while capturing their semantic meanings

1. **OVERVIEW**

Following previous literature, we define a knowledge graph as G = {E, R, F}, where E, R and F are sets of entities, relations and facts, respectively. A fact is denoted as a triple (h, r, t) ∈ F.

Categorization of Research on Knowledge Graph,

1. Knowledge Representation Learning

We categorize KRL into four aspects of representation space, scoring function, encoding models and auxiliary information, providing a clear workflow for developing a KRL model. Specific ingredients include:

1. representation space in which the relations and entities are represented;

Representation learning includes point-wise space, manifold, complex vector space, Gaussian distribution, and discrete space

1. scoring function for measuring the plausibility of factual triples;

Scoring metrics are generally divided into distance-based and similarity matching based scoring functions.

1. encoding models for representing and learning relational interactions;

Current research focuses on encoding models, including linear/bilinear models, factorization, and neural networks.

1. auxiliary information to be incorporated into the embedding methods.

Auxiliary information considers textual, visual, and type information.

1. Knowledge Acquisition
2. KGC(knowledge Graph Complete, expanding existing knowledge graphs)

embedding-based ranking, relation path reasoning, rule-based reasoning, and meta relational learning

1. relation extraction,

Entity discovery includes recognition, disambiguation, typing, and alignment.

1. entity discovery.

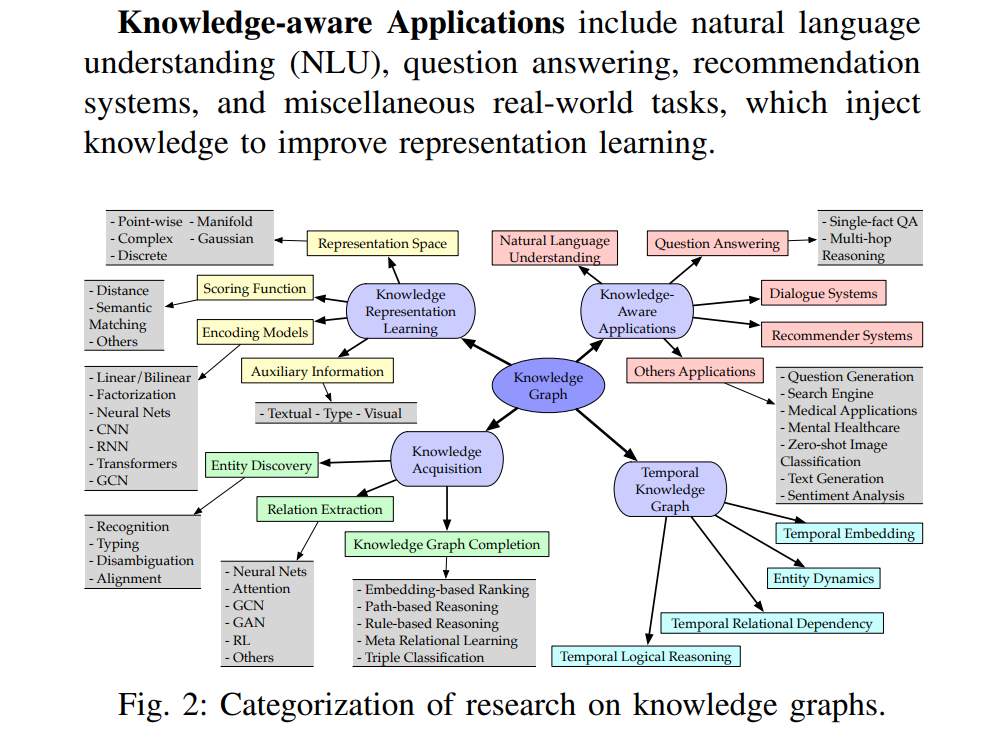
Relation extraction models utilize attention mechanism, graph convolutional networks (GCNs), adversarial training, reinforcement learning, deep residual learning, and transfer learning.

1. Temporal Knowledge Graphs

incorporate temporal information for representation learning. This survey categorizes four research fields, including temporal embedding, entity dynamics, temporal relational dependency, and temporal logical reasoning

1. Knowledge-aware Applications

include natural language understanding (NLU), question answering, recommendation systems, and miscellaneous real-world tasks, which inject knowledge to improve representation learning.



1. **KNOWLEDGE REPRESENTATION LEARNING**
2. Representation Space

The key issue of representation learning is to learn lowdimensional distributed embedding of entities and relations. Current literature mainly uses real-valued point-wise space (Fig. 3a) including vector, matrix and tensor space, while other kinds of space such as complex vector space (Fig. 3b), Gaussian space (Fig. 3c), and manifold (Fig. 3d) are utilized as well. The embedding space should follow three conditions, i.e., differentiability, calculation possibility, and definability of a scoring function [15].

1. Point-wise Space( Projecting into vectors or matrix space)

TransE [16] represents entities and relations in d-dimension vector space, i.e., h, t, r ∈ R d , and makes embeddings follow the translational principle h+r ≈ t.

TransR [17] then further introduces separated spaces for entities and relations. The authors projected entities (h, t ∈ R k ) into relation (r ∈ R d ) space by a projection matrix Mr ∈ R k×d

NTN [18] models entities across multiple dimensions by a bilinear tensor neural layer. The relational interaction between head and tail h TMt c is captured as a tensor denoted as Mc ∈ R d×d×k

HAKE [19] captures semantic hierarchies by mapping entities into the polar coordinate system, i.e., entity embeddings em ∈ R d and ep ∈ [0, 2π) d in the modulus and phase part, respectively.

1. Complex Vector Space （ i.e., h = Re(h)+iIm(h).）

ComplEx [23] firstly introduces complex vector space shown in Fig. 3b which can capture both symmetric and antisymmetric relations

RotatE [24] proposes a rotational model taking relation as a rotation from head entity to tail entity in complex space as t = h◦r （capture anti-symmetry）

QuatE [25] extends the complex-valued space into hypercomplex h, t, r ∈ Hd by a quaternion Q = a + bi + cj + dk with three imaginary components,

1. Gaussian Distribution

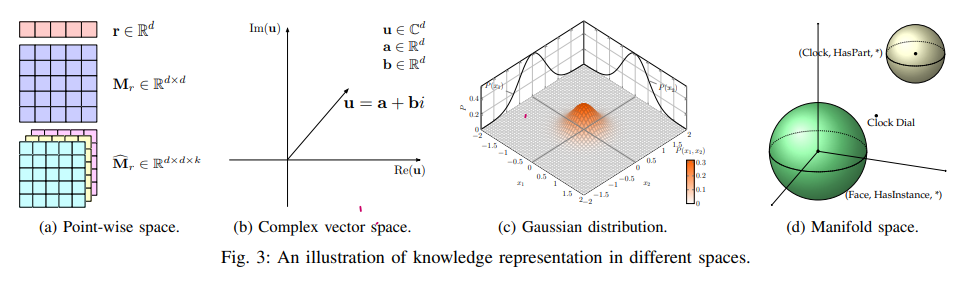
KG2E [26] introduces Gaussian distribution to deal with the (un)certainties of entities and relations.

TransG [27] represents entities with Gaussian distributions, while it draws a mixture of Gaussian distribution for relation embedding

1. Manifold and Group

ManifoldE [28] The authors introduced two settings of manifold-based embedding, i.e., Sphere and Hyperplane. Sphere: it relaxes the real-valued point-wise space into manifold space with a more expressive representation from the geometric perspective. Hyperplane: good at capturing hierarchical information.

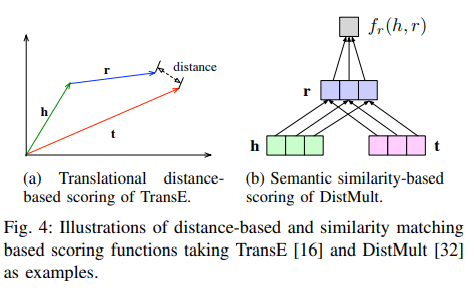
TorusE [15] solves the regularization problem of TransE via embedding in an n-dimensional torus space which is a compact Lie group



1. Scoring Function

The scoring function is used to measure the plausibility of facts, also referred to as the energy function in the energy-based learning framework. There are two typical types of scoring functions, i.e., distance-based (Fig. 4a) and similarity-based (Fig. 4b) functions, to measure the plausibility of a fact

Distance-based scoring function measures the plausibility of facts by calculating the distance between entities, where addictive translation with relations as h+r ≈ t is widely used. Semantic similarity-based scoring measures the plausibility of facts by semantic matching. It usually adopts a multiplicative formulation, i.e., , hT >Mr ≈ t T, to transform head entity near the tail in the representation space.



1. Distance-based Scoring Function:

Structural Embedding (SE) [8] uses two projection matrices and L1 distance to learn structural embedding

TransE, translation-based scoring function that aims to learn embeddings by representing relations as translations from head to tail entities



TransH projects entities and relations into a hyperplane,

TransR introduces separate projection spaces for entities and relations,

TransD constructs dynamic mapping matrices

TransA [34] uses Mahalanobis distance to enable more adaptive metric learning

TransF [35] relaxes the strict translation and uses dot product as Semantic Matching: 

ITransF [36] enables hidden concepts discovery and statistical strength transferring by learning associations between relations and concepts via sparse attention vectors,

KG2E [26] in Gaussian space and ManifoldE [28] with manifold also use the translational distance-based scoring function

1. Semantic Matching

SME [39] proposes to semantically match separate combinations of entity-relation pairs of (h, r) and (r, t).( linear and bilinear block)

DistMult [32] proposes a simplified bilinear formulation by restricting Mr to be diagonal.

HolE [21] introduces a circular correlation of embedding, which can be interpreted as a compressed tensor product, to learn compositional representations

HolEx [40] interpolates the HolE and full tensor product method.

ANALOGY [22] models analogical structures of relational data. HolE with ComplEx , Fourier transformed in the frequency domain can be viewed as a special case of ComplEx

CrossE , Crossover interactions are introduced by CrossE [42] with an interaction matrix C ∈ R nr×d to simulate the bi-directional interaction between entity and relation.

1. Encoding Models

This section introduces models that encode the interactions of entities and relations through specific model architectures, including linear/bilinear models, factorization models, and neural networks.

1. Linear/Bilinear

Linear models formulate relations as a linear/bilinear mapping by projecting head entities into a representation space close to tail entities.

Including SE [8], SME [39], DistMult [32], ComplEx [23], and ANALOGY [22]. SimplE [48], bilinear family such as RESCAL, DistMult, HolE and ComplEx can be transformed from one into another with certain constraint.

1. Factorization

Factorization aims to decompose relational data into low-rank matrices for representation learning. Factorization methods formulated KRL models as three-way tensor X decomposition. A general principle of tensor factorization can be denoted as Xhrt ≈ h >Mrt, with the composition function following the semantic matching patter

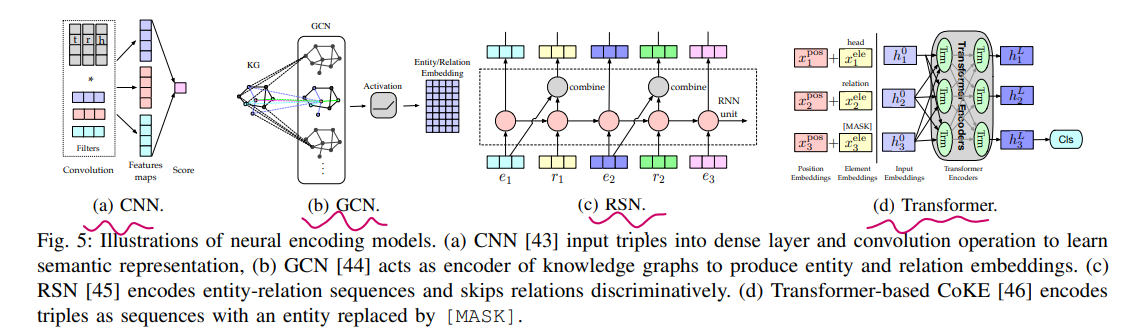
RESCAL, Nickel et al. [49] proposed the three-way rank-r factorization RESCAL over each relational slice of knowledge graph tenso

TuckER, By introducing threeway Tucker tensor decomposition, TuckER [52] learns to embed by outputting a core tensor and embedding vectors of entities and relations

LowFER [53] ,proposes a multimodal factorized bilinear pooling mechanism to better fuse entities and relations.

1. Neural models

Neural networks encode relational data with non-linear neural activation and more complex network structures by matching semantic similarity of entities and relations. Several neural models are illustrated in Fig. 5.



Representative neural models include

multi-layer perceptron (MLP) [3], encodes entities and relations together into a fully-connected layer, and uses a second layer with sigmoid activation for scoring a triple

neural tensor network (NTN) [18], NTN [18] takes entity embeddings as input associated with a relational tensor

neural association model (NAM) [54].

They generally feed entities or relations or both into deep neural networks and compute a semantic matching score

ConvE [55] uses 2D convolution over embeddings and multiple layers of nonlinear features to model the interactions between entities and relations by reshaping head entity and relation into 2D matrix

ConvKB [43] adopts CNNs for encoding the concatenation of entities and relations without reshaping

HypER [56] utilizes hypernetwork H for 1D relation-specific convolutional filter generation to achieve multi-task knowledge sharing, and meanwhile simplifies 2D ConvE

RNN-based, The MLP- and CNN-based models, as mentioned above, learn triplet-level representations. In comparison, the recurrent networks can capture long-term relational dependencies in knowledge graphs

RSN [45] (Fig. 5c) designs a recurrent skip mechanism to enhance semantic representation learning by distinguishing relations and entities

CoKE, (transformers-based), To utilize contextual information in knowledge graphs, CoKE [46] employs transformers to encode edges and path sequences.

KG-BERT [59] (transformers-based), borrows the idea form language model pretraining and takes Bidirectional Encoder Representations from Transformer (BERT) model as an encoder for entities and relations.

R-GCN [60] proposes relation-specific transformation to model the directed nature of knowledge graphs. GNNs are introduced for learning connectivity structure under an encoder-decoder framework

SACN [44] introduces weighted GCN (Fig. 5b), which defines the strength of two adjacent nodes with the same relation type, to capture the structural information

Conv-TransE adopts ConvE model as semantic matching metric and preserves the translational property.

CompGCN [63] proposes entity-relation composition operations over each edge in the neighborhood of a central node and generalizes previous GCN-based models.

1. Embedding with Auxiliary Information
2. Textual Description

The challenge of KRL with textual description is to embed both structured knowledge and unstructured textual information in the same space

DKRL [65] extends TransE [16] to learn representation directly from entity descriptions by a convolutional encoder

SSP [66] captures the strong correlations between triples and textual descriptions by projecting them in a semantic subspace.

1. Type Information

Entities are represented with hierarchical classes or types, and consequently, relations with semantic types.

SSE [67] incorporates semantic categories of entities to embed entities belonging to the same category smoothly in semantic space.

TKRL [68] proposes type encoder model for projection matrix of entities to capture type hierarchy.

KREAR [69] categorizes relation types into attributes and relations and modeled the correlations between entity descriptions.

1. Uncertain Information

In contrast to classic deterministic knowledge graph embedding, uncertain embedding models aim to capture uncertainty representing the likelihood of relational facts.

Chen et al. [75] proposed an uncertain knowledge graph embedding model to simultaneously preserve structural and uncertainty information, where probabilistic soft logic is applied to infer the confidence score

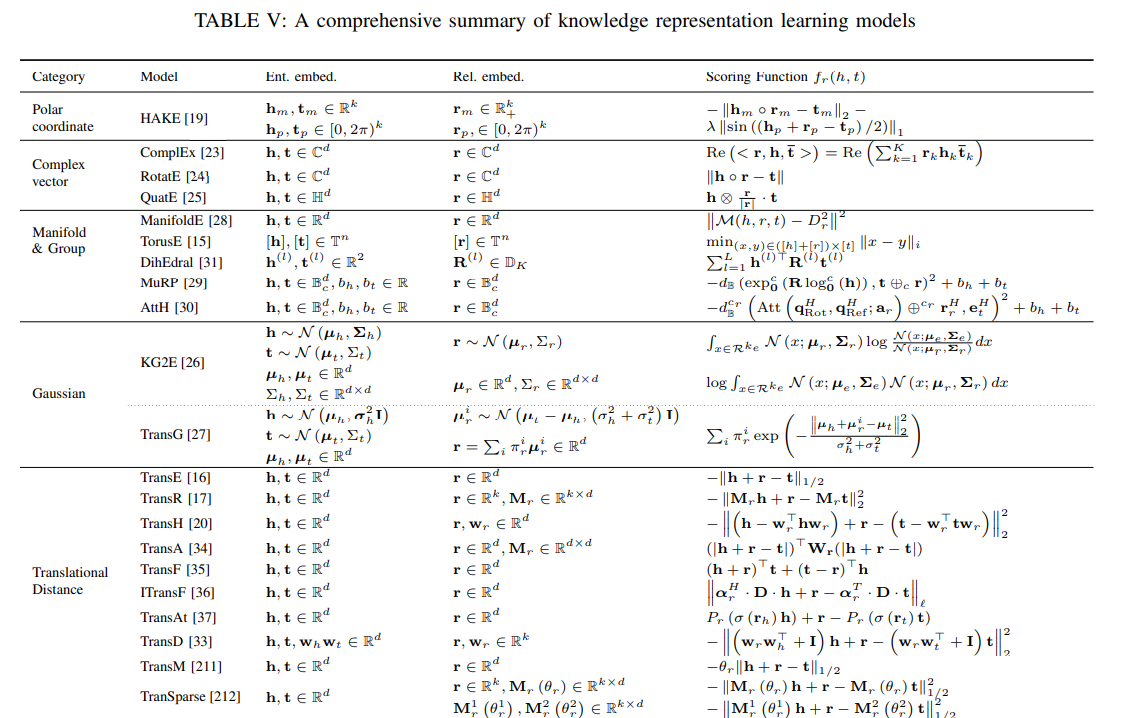
1. Visual information (e.g., entity images)

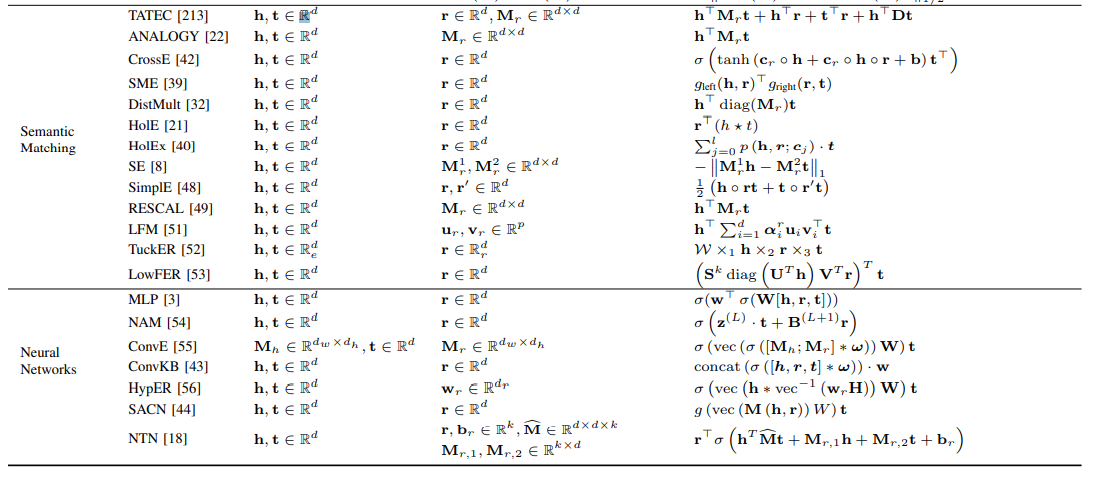
Image-embodied IKRL [71], containing cross-modal structure-based and image-based representation, encodes images to entity space and follows the translation principle.

1. Summary

Overall, developing a novel KRL model is to answer the following four questions: 1) which representation space to choose; 2) how to measure the plausibility of triplets in a specific space; 3) which encoding model to use for modeling relational interactions; 4) whether to utilize auxiliary information.

The most popularly used representation space is Euclidean point-based space by embedding entities in vector space and modeling interactions via vector, matrix, or tensor. Other representation spaces, including complex vector space, Gaussian distribution, and manifold space and group, are also studied. Manifold space has an advantage over pointwise Euclidean space by relaxing the point-wise embedding. Gaussian embeddings can express the uncertainties of entities and relations, and multiple relation semantics. Embedding in complex vector space can effectively model different relational connectivity patterns, especially the symmetry/anti-symmetry pattern. The representation space plays an essential role in encoding the semantic information of entities and capturing the relational properties





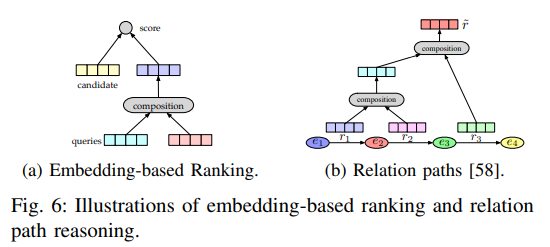
**IV. KNOWLEDGE ACQUISITION**

Knowledge acquisition aims to construct knowledge graphs from unstructured text and other structured or semi-structured sources, complete an existing knowledge graph, and discover and recognize entities and relations.

The main tasks of knowledge acquisition include relation extraction, KGC, and other entity-oriented acquisition tasks such as entity recognition and entity alignment.

1. Knowledge Graph Completion

Because of the nature of incompleteness of knowledge graphs, KGC is developed to add new triples to a knowledge graph. Typical subtasks include link prediction, entity prediction, and relation prediction



1. Embedding-based Models

Preliminary research on KGC focused on learning lowdimensional embedding for triple prediction.

Taking entity prediction as an example, embedding-based ranking methods, as shown in Fig. 6a, firstly learn embedding vectors based on existing triples. By replacing the tail entity or head entity with each entity e ∈ E, those methods calculate scores of all the candidate entities and rank the top k entities.

Including aforementioned KRL methods (e.g., TransE [16], TransH [20], TransR [17], HolE [21], and RGCN [60]) and joint learning methods like DKRL [65] with textual information

ProjE [82] proposes a combined embedding by space projection of the known parts of input triples, and another candidate entities. (does not support relation prediction)

SENN [83] distinguishes three KGC subtasks explicitly by introducing a unified neural shared embedding with adaptively weighted general loss function to learn different latent features

ConMask [84] proposes relationship-dependent content masking over the entity description to select relevant snippets of given relations, and CNN-based target fusion to complete the knowledge graph with unseen entities. Tend to capture evolution of Knowledge.

REMEDY [85], for medical domain.

1. Relation Path Reasoning:

Embedding learning of entities and relations has gained remarkable performance in some benchmarks, but it fails to model complex relation paths. Relation path reasoning turns to leverage path information over the graph structure.

Path-Ranking Algorithm (PRA) [86] chooses a relational path under a combination of path constraints and conducts maximum-likelihood classification.

Neelakantan et al. [58] developed an RNN model to compose the implications of relational paths by applying compositionality recursively (in Fig. 6b)

Chainof-Reasoning [87], a neural attention mechanism to enable multiple reasons, represents logical composition across all relations, entities, and text

DIVA [88] proposes a unified variational inference framework that takes multi-hop reasoning as two sub-steps of path-finding (a prior distribution for underlying path inference) and path-reasoning (a likelihood for link classification).

1. RL-based Path Finding:

DeepPath [89] firstly applies RL into relational path learning and develops a novel reward function to improve accuracy, path diversity, and path efficiency

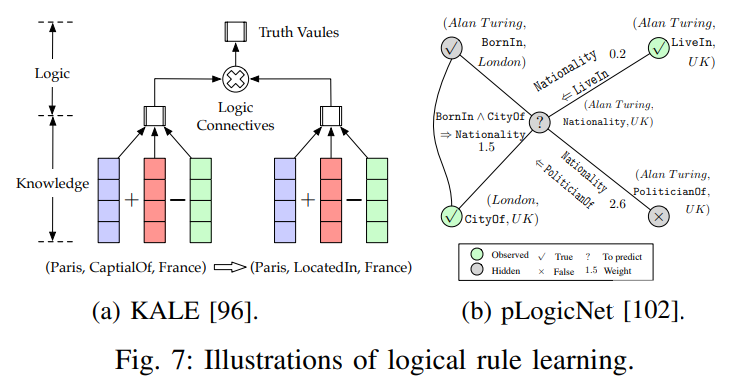
MINERVA [90] takes path walking to the correct answer entity as a sequential optimization problem by maximizing the expected reward

MultiHop [91] proposes a soft reward mechanism

M-Walk [92] applies an RNN controller to capture the historical trajectory and uses the Monte Carlo Tree Search (MCTS) for effective path generation

CPL [93] proposes collaborative policy learning for pathfinding and fact extraction from text.

1. Rule-based Reasoning(better make use of the symbolic nature of knowledge)



For example, given relations sonOf, hasChild and gender, and entities X and Y , there is a rule in the reverse form of logic programming as: (Y, sonOf, X) ← (X, hasChild, Y) ∧ (Y, gender, Male)

RLvLR [95] proposes a scalable rule mining approach with efficient rule searching and pruning, and uses the extracted rules for link prediction

KALE [96] proposes a unified joint model with t-norm fuzzy logical connectives defined for compatible triples and logical rules embedding. Specifically, three compositions of logical conjunction, disjunction, and negation are defined to compose the truth value of a complex formula. Fig. 7a illustrates a simple first-order Horn clause inference.

RUGE [97] proposes an iterative model, where soft rules are utilized for soft label prediction from unlabeled triples and labeled triples for embedding rectification.

IterE [98] proposes an iterative training strategy with three components of embedding learning, axiom induction, and axiom injection.

The logical rule is one kind of auxiliary information; meanwhile, it can incorporate prior knowledge, enabling the ability of interpretable multi-hop reasoning and paving the way for generalization even in few-shot labeled relational triples.

Neural Theorem Provers (NTP) [99] learns logical rules for multi-hop reasoning, which utilizes a radial basis function kernel for differentiable computation on vector space.

NeuralLP [100] enables gradient-based optimization to be applicable in the inductive logic programming

pLogicNet [102] proposes probabilistic logic neural networks (Fig. 7b) to leverage first-order logic and learn effective embedding by combining the advantages of Markov logic networks and KRL methods while handling the uncertainty of logic rules.

ExpressGNN [103] generalizes pLogicNet by tuning graph networks and embedding and achieves more efficient logical reason

1. Meta Relational Learning

The long-tail phenomena exist in the relations of knowledge graphs. Meanwhile, the real-world scenario of knowledge is dynamic, where unseen triples are usually acquired. The new scenario, called as meta relational learning or few-shot relational learning, requires models to predict new relational facts with only a very few samples.

GMatching [104] develops a metric based few-shot learning method with entity embeddings and local graph struct

Meta-KGR [105], an optimization-based meta-learning approach, adopts model agnostic meta-learning for fast adaption and reinforcement learning for entity searching and path reasoning

1. Triple Classification

Triple classification is to determine whether facts are correct in testing data, which is typically regarded as a binary classification problem.

Including ding translational distance-based methods like TransH [20] and TransR [17] and semantic matching-based methods such as NTN [18], HolE [21] and ANALOGY.

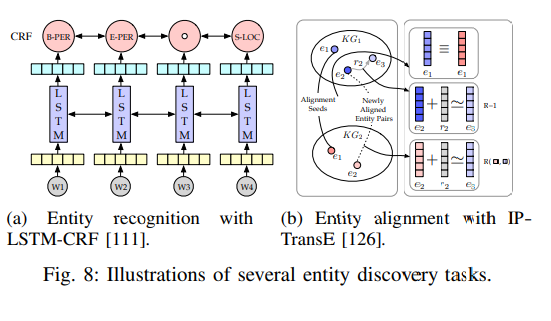
Dong et al. [79] extended the embedding space into region-based n-dimensional balls where the tail region is in the head region for 1-to-n relation using fine-grained type chains. Better at dealing 1-to-n relations

1. Entity discovery
2. entity recognition

Entity recognition or named entity recognition (NER), when it focuses on specifically named entities, is a task that tags entities in text. Hand-crafted features such as capitalization patterns and language-specific resources like gazetteers are applied in many pieces of literature.

LSTM-CNN [110] for learning character-level and word-level features and encoding partial lexicon matches

Stack-LSTM , Lample et al. [111] proposed stacked neural architectures by stacking LSTM layers and CRF layers, i.e., LSTM-CRF (in Fig. 8a) and Stack-LSTM



MGNER [112] proposes an integrated framework with entity position detection in various granularities and attention-based entity classification for both nested and non-overlapping named entities

NER , Recently, Li et al. [114] formulated flat and nested NER as a unified machine reading comprehension framework by referring annotation guidelines to construct query questions

1. entity disambiguation

DSRM [121], The contemporary end-to-end learning approaches have made efforts through representation learning of entities and mentions, for example, DSRM [121] for modeling entity semantic relatedness and EDKate [122] for the joint embedding of entity and text.

Ganea and Hofmann [123] proposed an attentive neural model over local context windows for entity embedding learning and differentiable message passing for inferring ambiguous entities.

Le and Titov [124], By regarding relations between entities as latent variables, Le and Titov [124] developed an end-to-end neural architecture with relation-wise and mention-wise normalization

1. entity typing

Entity typing includes coarse and finegrained types, while the latter uses a tree-structured type category and is typically regarded as multi-class and multilabel classification.

PLE [117] focuses on correct type identification and proposes a partial-label embedding model with a heterogeneous graph for the representation of entity mentions, text features, and entity types and their relationships. Which reduced label noise.

Ma et al. [118]

JOIE [119], use embedding approaches

Connect-E [120], enhanced joint embedding learning

1. entity alignment

entity alignment (EA) aims to fuse knowledge among various knowledge graphs

MTransE [125], Embedding-based alignment calculates the similarity between the embeddings of a pair of entities. MTransE [125] firstly studies entity alignment in the multilingual scenario.

IPTransE [126] proposes an iterative alignment model by mapping entities into a unified representation space under a joint embedding framework

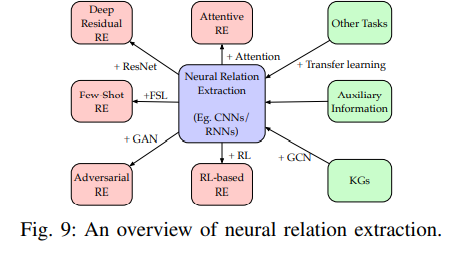
Additional information of entities is also incorporated for refinement, for example, JAPE [128] capturing the correlation between cross-lingual attributes, KDCoE [129] embedding multi-lingual entity descriptions via co-training, MultiKE [130] learning multiple views of the entity name, relation, and attributes, and alignment with character attribute embedding [131].

1. Relation extraction

Relation extraction is a key task to build large-scale knowledge graphs automatically by extracting unknown relational facts from plain text and adding them into knowledge graphs.

Some techniques here includes weak supervision, feature engineering and inner correlation between features.

This section reviews recent advances of neural relation extraction (NRE), with an overview illustrated in Fig. 9.



1. Neural Relation Extraction

CNNs with position features of relative distances to entities [136] are firstly explored for relation classification, and then extended to relation extraction by multi-window CNN [137] with multiple sized convolutional filters.

1. Attention Mechanism
2. Graph Convolutional Networks

GCNs are utilized for encoding a dependency tree over sentences or learning KGEs to leverage relational knowledge for sentence encoding.

1. Adversarial Training

Adversarial Training (AT) is applied to add adversarial noise to word embeddings for CNNand RNN-based relation extraction under the MIML learning setting

1. Reinforcement Learning

RL has been integrated into neural relation extraction recently by training instance selector with policy networks.

1. Other Advances
2. Joint Entity and Relation Extraction

Traditional relation extraction models utilize pipeline approaches by first extracting entity mentions and then classifying relations. However, pipeline methods may cause error accumulation. Several studies show better performance by joint learning

1. Summary

Knowledge graph completion completes missing links between existing entities or infers entities given entity and relation queries. Embedding-based KGC methods generally rely on triple representation learning to capture semantics and do candidate ranking for completion. Embedding-based reasoning remains in individual relation level, and is poor at complex reasoning because it ignores the symbolical nature of knowledge graph, and lack of interpretability. Hybrid methods with symbolics and embedding incorporate rulebased reasoning, overcome the sparsity of knowledge graph to improve the quality of embedding, facilitate efficient rule injection, and induce interpretable rules. With the observation of the graphical nature of knowledge graphs, path search and neural path representation learning are studied. However, they suffer from connectivity deficiency when traverses over largescale graphs. The emerging direction of meta relational learning aims to learn fast adaptation over unseen relations in lowresource settings

Entity discovery acquires entity-oriented knowledge from text and fuses knowledge between knowledge graphs. There are several categories according to specific settings. Entity recognition is explored in a sequence-to-sequence manner, entity typing discusses noisy type labels and zero-shot typing, and entity disambiguation and alignment learn unified embeddings with iterative alignment model proposed to tackle the issue of a limited number of alignment seeds.

Relation extraction suffers from noisy patterns under the assumption of distant supervision, especially in text corpus of different domains. Thus, weakly supervised relation extraction must mitigate the impact of noisy labeling. For example, multi-instance learning takes bags of sentences as inputs and attention mechanism [146] reduce noisy patterns by soft selection over instances, and RL-based methods formulate instance selection as a hard decision. Another principle is to learn richer representation as possible. As deep neural networks can solve error propagation in traditional feature extraction methods, this field is dominated by DNN-based models, as summarized in Table IV

1. **TEMPORAL KNOWLEDGE GRAPH**

Current knowledge graph research mostly focuses on static knowledge graphs where facts are not changed with time, while the temporal dynamics of a knowledge graph is less explored.

1. Temporal Information Embedding

Temporal information is considered in temporal-aware embedding by extending triples into temporal quadruple as (h, r, t, τ ), where τ provides additional temporal information about when the fact held.

1. Entity Dynamics

Real-world events change entities’ state, and consequently, affect the corresponding relations. To improve temporal scope inference, the contextual temporal profile model [181] formulates the temporal scoping problem as state change detection and utilizes the context to learn state and state change vectors.

1. Temporal Relational Dependency

There exists temporal dependencies in relational chains following the timeline, for example, wasBornIn → graduateFrom → workAt → diedIn

**D.** Temporal Logical Reasoning

1. **KNOWLEDGE-AWARE APPLICATIONS**

The application of knowledge graphs includes two folds: 1) in-KG applications such as link prediction and named entity recognition; and 2) out-of-KG applications, including relation extraction and more downstream knowledge-aware applications such as question answering and recommendation systems. This section introduces several recent DNN-based knowledge-driven approaches with the applications on natural language processing and recommendation. More miscellaneous applications such as digital health and search engine are introduced in Appendix E.

1. Language Representation Learning

Language representation learning via self-supervised language model pretraining has become an integral component of many NLP systems. Traditional language modeling does not exploit factual knowledge with entities frequently observed in the text corpus. How to integrate knowledge into language representation has drawn increasing attention.

1. Question Answering

Knowledge-graph-based question answering (KG-QA) answers natural language questions with facts from knowledge graphs. Neural network-based approaches represent questions and answers in distributed semantic space, and some also conduct symbolic knowledge injection for commonsense reasoning

1. Recommender Systems

**VII. FUTURE DIRECTIONS**

1. Complex Reasoning
2. Unified Framework
3. Interpretability

**D**. Scalability

**E.** Knowledge Aggregation

**F.**  Automatic Construction and Dynamics

Current knowledge graphs rely highly on manual construction, which is labor-intensive and expensive. The widespread applications of knowledge graphs on different cognitive intelligence fields require automatic knowledge graph construction from large-scale unstructured content. Recent research mainly works on semi-automatic construction under the supervision of existing knowledge graphs. Facing the multimodality, heterogeneity, and large-scale application, automatic construction is still of great challenge.

**APPENDIX E More Application**

**A.** Text Classification and Task-Specific Applications

**B**. Dialogue Systems

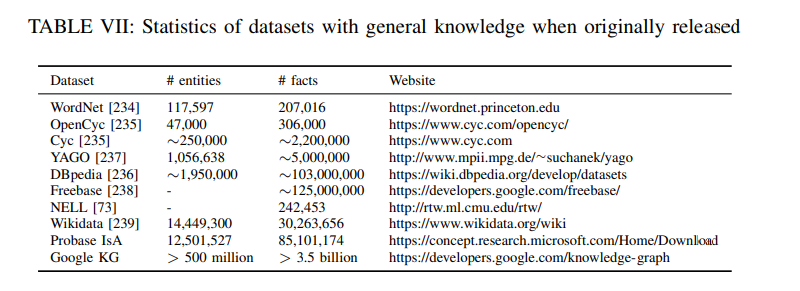
**C**. Medicine and Biology

**APPENDIX F Datasets and Library**

**A.**

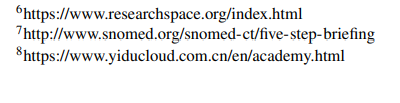
**a)** General Datasets:

Datasets with general ontological knowledge include WordNet [234], Cyc [235], DBpedia [236], YAGO [237], Freebase [238], NELL [73] and Wikidata [239]. It is hard to compare them within a table as their ontologies are different. Thus, only an informal comparison is illustrated in Table VII, where their volumes kept going after their release.



**b)** Domain-Specific Datasets:

Some knowledge bases on specific domains are designed and collected to evaluate domainspecific tasks. Some notable domains include life science, health care, and scientific research, covering complex domains and relations such as compounds, diseases, and tissues. Examples of domain-specific knowledge graphs are ResearchSpace6 , a cultural heritage knowledge graph; UMLS [240], a unified medical language system; SNOMED CT7 , a commercial clinical terminology; and a medical knowledge graph from Yidu Research8 .



**B. Open-Source Libraries**

Recent research has boosted the open-source campaign, with several libraries listed in Table IX. They are AmpliGraph [252] for knowledge representation learning, Grakn for integration knowledge graph with machine learning techniques, and Akutan for knowledge graph store and query. The research community has also released codes to facilitate further research. Notably, there are three useful toolkits, namely scikit-kge and OpenKE [253] for knowledge graph embedding, and OpenNRE [254] for relation extraction. We provide an online collection of knowledge graph publications, together with links to some open-source implementations of them, hosted at https://shaoxiongji.github.io/knowledge-graphs/.

