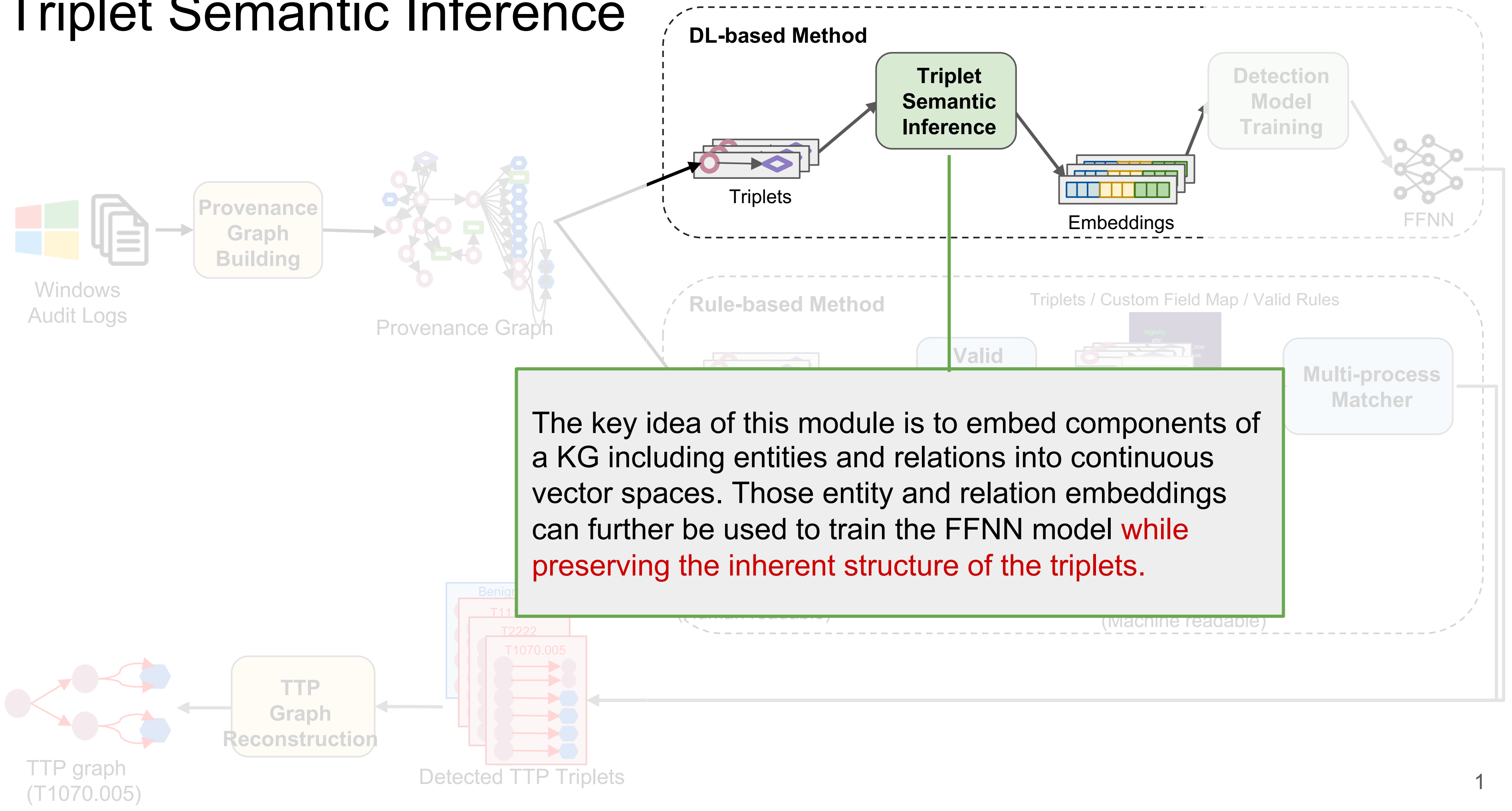
Triplet Semantic Inference



The key idea of this module is to embed components of a KG including entities and relations into continuous vector spaces. Those entity and relation embeddings can further be used to train the FFNN model while preserving the inherent structure of the triplets.

Background of Knowledge Graph Research

- KG has become an increasingly popular research direction towards cognition and human-level intelligence.
- Real-world applications: recommendation systems, question answering etc.
- A survey paper provides a comprehensive review of knowledge graph covering overall research topics about:
    A. knowledge graph representation learning(KRL) or knowledge graph embedding (KGE)
    B. knowledge acquisition and completion
    C. temporal knowledge graph
    D. knowledge-aware applications

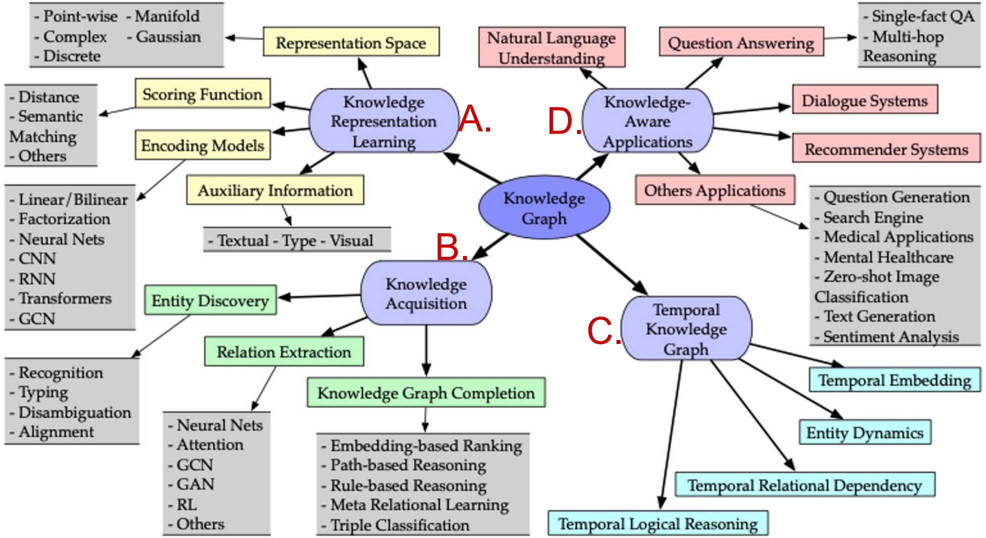


Fig: Categorization of research on knowledge graph

Background of Knowledge Graph Research

- KG has become an increasingly popular research direction towards cognition and human-level intelligence.
- Real-world applications: recommendation systems, question answering etc.
- A survey paper provides a comprehensive review of knowledge graph covering overall research topics about:
  A. knowledge graph representation learning(KRL) or knowledge graph embedding (KGE)
  B. knowledge acquisition and completion
  C. temporal knowledge graph
  D. knowledge-aware applications

study on

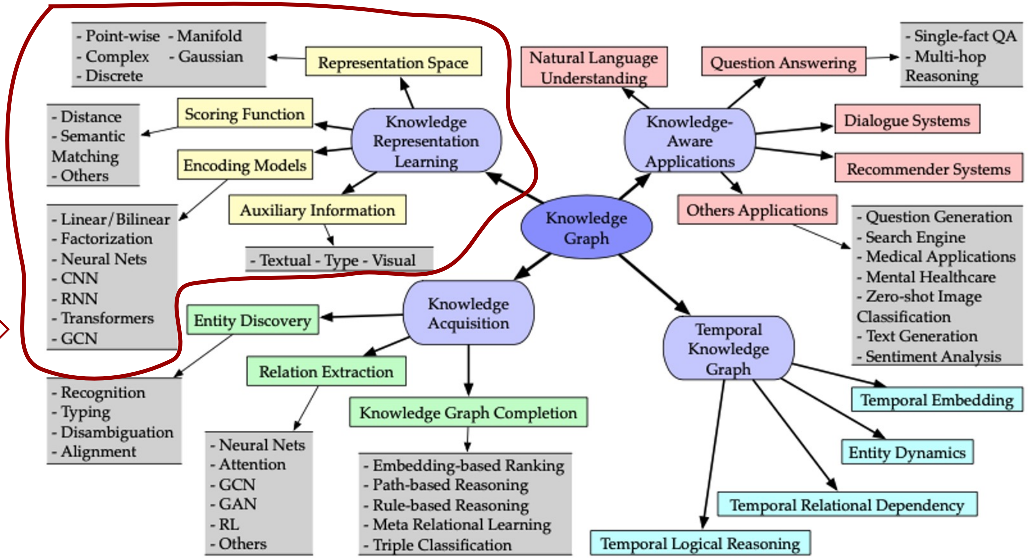
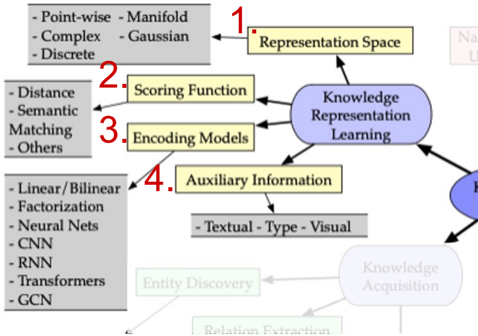


Fig: Categorization of research on knowledge graph

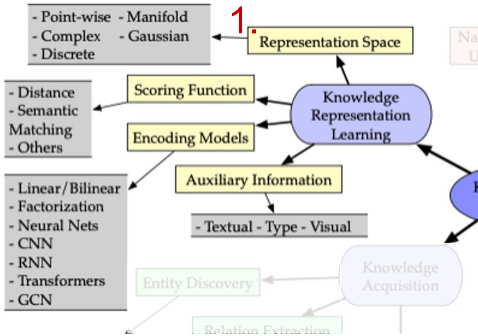
Background of Knowledge Graph Embedding



A clear workflow for developing a KGE model includes:

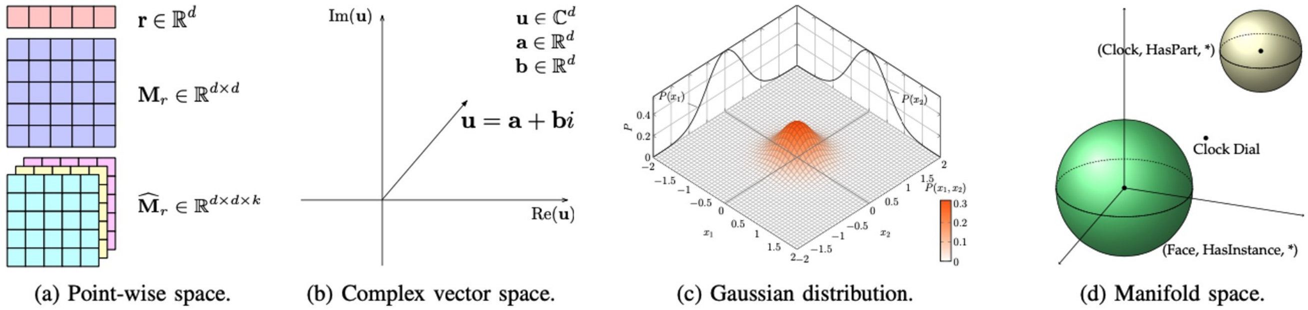
1. **Representation space** in which the relations and entities are represented
2. **Scoring function** for measuring the plausibility of factual triples;
3. **Encoding models** for representing and learning relational interactions;
4. **Auxiliary information** to be incorporated into the embedding methods.

Background of Knowledge Graph Embedding



A clear workflow for developing a KGE model includes:

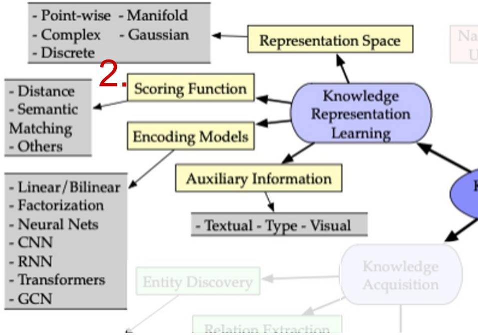
1. **Representation space** in which the relations and entities are represented
   - The most popularly used representation space is Euclidean point-based space by embedding entities in vector space and modeling interactions via vector, matrix, or tensor.



(a) Point-wise space. (b) Complex vector space. (c) Gaussian distribution. (d) Manifold space.

Fig. 3: An illustration of knowledge representation in different spaces.

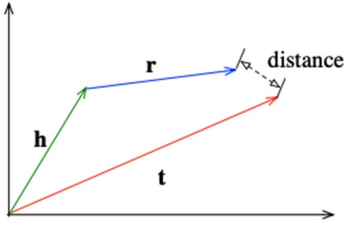
Background of Knowledge Graph Embedding



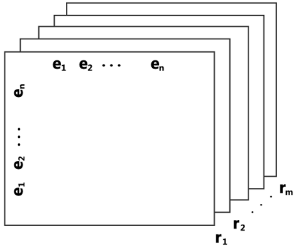
A clear workflow for developing a KGE model includes:

2. **Scoring function** $f_r(h,t)$ for measuring the plausibility of factual triplets;
    a. **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.
    b. **Pairwise interactions** scoring function measures the plausibility of facts by applying a tensor to express the inherent structure of a KG. The score can be captured by the interaction of head and relation.

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}$$



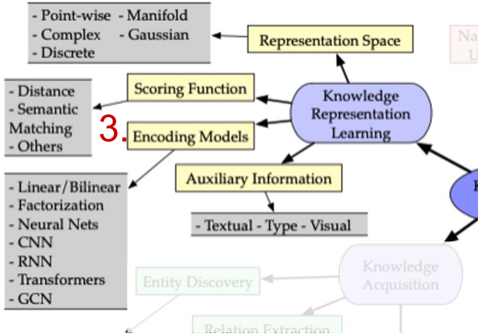
(a) Translational distance-based scoring of TransE.



(b) A tensor model of knowledge graph.

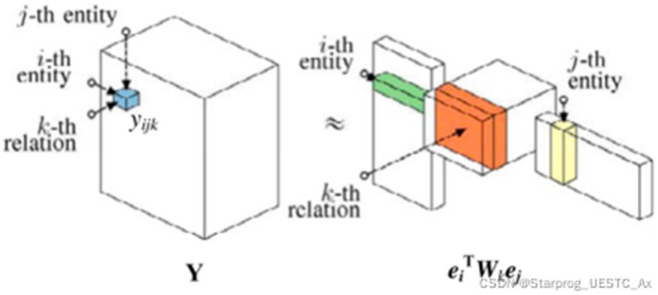
$$f_r(h,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$$

Background of Knowledge Graph Embedding

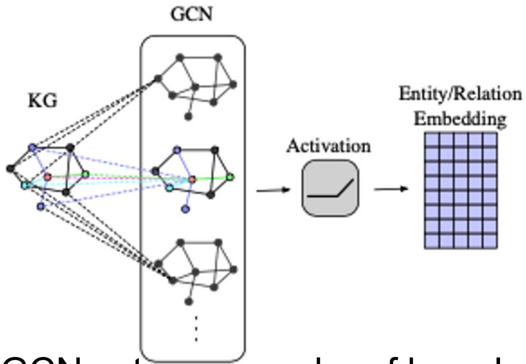


A clear workflow for developing a KGE model includes:

3. **Encoding models** that encode the interactions of entities and relations through specific model architectures.
   a. **Factorization models** formulates KRL models as three-way tensor X decomposition. For k-th relation of m relations, the k-th slice of X is factorized as $\mathcal{X}_k \approx \mathbf{A}\mathbf{R}_k\mathbf{A}^T$.
   b. **GCN-based model** utilizes complex graph convolution neural network structure to learn richer representation.

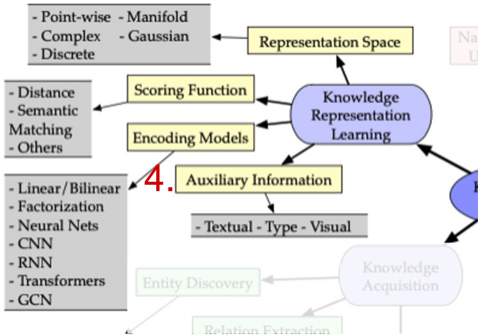


(a) Diagram of a three-way tensor



(b) GCN acts as encoder of knowledge graphs to produce entity and relation embeddings.

Background of Knowledge Graph Embedding



A clear workflow for developing a KGE model includes:

4. **Auxiliary information** to be incorporated into the embedding methods.
    - External information such as relation/entity types, image entity, path inference.
    - This thesis does not take this topic into account due to a limited research timeframe.

事實上，從 2010~2022 已經出現許多 KGE 的模型，參考近期論文（Watson, ShadeWatcher）以及考量實作可能性後，挑選了八個模型來實驗。

Year	Model	Technique	Year	Model	Technique
2022	RED-GNN [82]	GNN	2019	SACN [61]	GNN
2022	ConGLR [85]	GNN	2019	KBGAT [59]	GNN
2022	TripleRE [32]	Translational	2019	LAN [60]	GNN
2022	InterHT [33]	Translational	2019	CPL [87]	Relation Path
2022	HousE [31]	Translational	2019	IterE [88]	Logic Rule
2022	BERTRL [84]	GNN	2019	pLogicNet [89]	Logic Rule
2022	SNRI [73]	GNN	2019	DRUM [90]	Logic Rule
2022	TEMP [76]	GNN	2019	RLvLR [91]	Logic Rule
2022	RMPI [75]	GNN	2019	Neural-Num-LP [92]	Logic Rule
2022	Meta-iKG [72]	GNN	2018	SimplE [39]	Tensor Decompositional
2022	CSR [86]	GNN	2018	ConvKB [52]	CNN
2022	CURL [93]	Relation Path	2018	ConvE [51]	CNN
2022	GCR [94]	Logic Rule	2018	RGCN [57]	GNN
2021	PairRE [30]	Translational	2018	M-walk [95]	Relation Path
2021	HopfE [44]	Tensor Decompositional	2018	MultiHop [96]	Relation Path
2021	DualE [43]	Tensor Decompositional	2018	DIVA [97]	Logic Rule
2021	ConEx [56]	CNN	2018	RuleN [98]	Logic Rule
2021	KE-GCN [65]	GNN	2018	RUGE [99]	Logic Rule
2021	HRFN [68]				Tensor Decompositional
2021	GEN [67]				Traditional NN
2021	INDIGO [69]				Relation Path
2021	NBF-Net [81]				Relation Path
2021	CoMPILE [71]				Logic Rule
2021	TACT [70]	GNN	2017	NeuralLP [103]	Logic Rule
2021	RPC-IR [74]	GNN	2016	TranSparse [23]	Translational
2020	HAKE [28]	Translational	2016	TransG [22]	Translational
2020	TransRHS [29]	Translational	2016	HolE [37]	Tensor Decompositional
2020	LowFER [45]	Tensor Decompositional	2016	ComplEx [36]	Tensor Decompositional
2020	InteractE [55]	CNN	2016	NAM [49]	Traditional NN
2020	DPMPN [63]	GNN	2016	LogSumExp [104]	Relation Path
2020	RGHAT [64]	GNN	2016	KALE [105]	Logic Rule
2020	COMPGCN [66]	GNN	2015	TransD [20]	Translational
2020	GraIL [7]	GNN	2015	TransR [19]	Translational
2020	ExpressGNN [106]	Logic Rule	2015	KG2E [21]	Translational
2020	pGAT [83]	GNN	2015	DISTMULT [35]	Tensor Decompositional
2019	RotatE [27]	Translational	2015	RNNPRA [107]	Relation Path
2019	TransW [26]	Translational	2014	TransH [18]	Translational
2019	MuRP [25]	Translational	2014	ProPPR [108]	Relation Path
2019	QuatE [42]	Tensor Decompositional	2013	AMIE [109]	Logic Rule
2019	TuckER [40]	Tensor Decompositional	2013	SME [47]	Traditional NN
2019	CrossE [41]	Tensor Decompositional	2013	NTN [48]	Traditional NN
2019	ConvR [54]	CNN	2013	TransE [17]	Translational
2019	HypER [53]	CNN	2011	RESCAL [34]	Tensor Decompositional
2019	M-GNN [58]	GNN	2010	PRA [110]	Relation Path

做一個自己的版本。

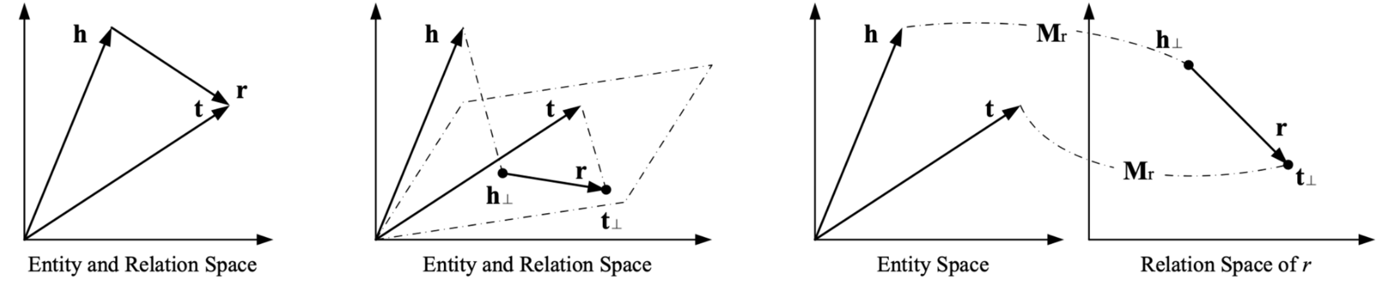
Category	Model	Citation
Translation Distance Models	TransE(2013)	6772
	TransH(2014)	3362
	TransR(2015)	3380
Tensor Factorization Models	Rescal(2011)	2334
	DistMult(2014)	2611
	ComplEx(2016)	2412
NN-based Model	R-GCN(2018)	3583
	CompGCN(2020)	530

Characteristics on Selected KGE models

Category	Model	Characteristics
Translation Distance Models	TransE(2013)	● Precursory translation method
	TransH(2014)	● Performs translation in relation-specific hyperplane ● Improve the performance of TransE on 1-to-N, N-to-1, and N-to-N relations.
	TransR(2015)	● Converts entity space to relation space ● Relational space projection
Tensor Factorization Models	RESCAL(2011)	● Precursory semantic matching method
	DistMult(2014)	● RESACL + diagonal matrices
	ComplEx(2016)	● DistMult + Complex-valued embeddings
GCN-based Model	R-GCN(2018)	● Basis decomposition; block-diagonal-decomposition; ● end-to-end framework: ○ encoder: R-GCN ○ decoder: DistMult
	CompGCN(2020)	● Entity-relation-composition operators ● end-to-end framework: ○ encoder: COMPGCN, ○ decoder: DistMult, etc.

Simple Illustration of Selected Translation-based KGE Model

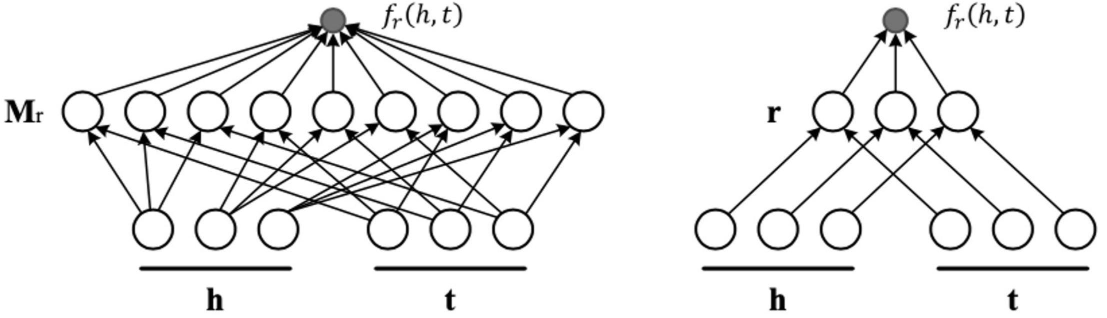
Category	Model	Characteristics	Ent. embedding	Rel. embedding	Scoring Function $f_r(h,t)$
Translation Distance Models	TransE (2013)	• Precursory translation method	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$-\lVert \mathbf{h} + \mathbf{r} - \mathbf{t} \rVert_{1/2}$
	TransH (2014)	• Performs translation in relation-specific hyperplane • 1-to-N, N-to-1, and N-to-N relations	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r}, \mathbf{w}_r \in \mathbb{R}^d$	$-\lVert (\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{r} - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r) \rVert_2^2$
	TransR (2015)	• Converts entity space to relation space • Relational space projection	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^k$, $\mathbf{M}_r \in \mathbb{R}^{k \times d}$	$-\lVert \mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t} \rVert_2^2$



(a) TransE. (b) TransH. (c) TransR.

Simple Illustration of Selected Semantic Matching KGE Model

Category	Model	Characteristics	Ent. embedding	Rel. embedding	Scoring Function $f_r(h,t)$
Tensor Factorization Models	RESCAL (2011)	• Precursory semantic matching method	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{M}_r \in \mathbb{R}^{d \times d}$	$\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$
	DistMult (2014)	• RESACL + diagonal matrices • Faster than RESACL	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$	$\mathbf{r} \in \mathbb{R}^d$	$\mathbf{h}^\top \mathrm{diag}(\mathbf{r})\mathbf{t}$
	ComplEx (2016)	• DistMult + Complex-valued embeddings	$\mathbf{h}, \mathbf{t} \in \mathbb{C}^d$	$\mathbf{r} \in \mathbb{C}^d$	$\mathrm{Re}\left(\mathbf{h}^\top \mathrm{diag}(\mathbf{r})\bar{\mathbf{t}}\right)$



(a) RESCAL. (b) DistMult.

Simple Illustration of GCN-based KGE Model

Category	Model	Characteristics	Ent. embedding	Rel. embedding	Scoring Function $f_r(h,t)$
GCN-based Model	R-GCN (2018)	● Basis decomposition; block-diagonal-decomposition; ● end-to-end framework: ○ encoder: R-GCN ○ decoder: DistMult			
	CompGCN (2020)	● Entity-relation-composition operators ● end-to-end framework: ○ encoder: COMPGCN, ○ decoder: DistMult, etc.			

實驗設計

KGE models demonstrated their superiority over others by utilizing benchmark datasets (such as WN18 and FB15K) and conducting downstream tasks like **Link Prediction(LP)** and **Triplet Classification(TC)**.

To identify the most suitable KGE model for our next stage, we input our synthesized triplets into eight selected models and evaluate them on LP and TC tasks. The final decision regarding the KGE model was based on the outcome measures, including MRR, Hit rate, f1-score, and accuracy.

Training Process for KGE models(1/2)

- The concept of a KGE model is to solve an optimization problem that maximizes the total plausibility of **observed triplets** in KG.
- That is, minimizing the loss function($\mathcal{L}_{softplus}$ or $\mathcal{L}_{margin}$) constituted by scoring function of both observed triplets $\mathcal{T}$ and corrupted(non-observed) triplets $\mathcal{T}'$.

$$\mathcal{L}_{margin} = \sum_{\tau \in \mathcal{T}} \sum_{\tau' \in \mathcal{T}'} [\gamma + \underbrace{\boxed{f_{r'}(h', t')}}_{\text{plausibility of corrupted triplets}} - \underbrace{\boxed{f_r(h, t)}}_{\text{plausibility of observed triplets}}]$$

$$\mathcal{L}_{softplus} = \sum_{\tau \in \mathcal{T}} log(1 + exp(-\boxed{f_r(h, t)})) + \sum_{\tau' \in \mathcal{T}'} log(1 + exp(\boxed{f_{r'}(h', t')}))$$

Training Process for KGE models(2/2)

Algorithm 1: Learning KGE models(Simplified)

Input : The training set $\mathcal{T} = (h, r, t)$, entity set $\mathcal{E}$, relation set $\mathcal{R}$, embedding dimension d

Output: Entity and relation embeddings

Initialize the entity embeddings e and relation embeddings r
for $i \leftarrow 1$ **to** $num\_epoches$ **do**
    `// Sample a subset from` $\mathcal{T}$ `with batch size` b
    $\mathcal{T}_{batch} \leftarrow sample(\mathcal{T}, b)$
    for $\tau \in \mathcal{T}_{batch}$ **do**
        `// Negative Sampling`
        Sample a observed triplet $\tau = (h, r, t)$
        Sample a non-observed triplet $\tau' = (h', r', t')$
        `// Update embeddings by minimizing the loss function`
        `// TBM ->` $\mathcal{L}_{margin}$`, SMM ->` $\mathcal{L}_{softplus}$
        Compute the loss function $\mathcal{L}$
        Update the gradient $\nabla \mathcal{L}$
        `// Handle additional constraints and regularization terms`
        ...
    end for
    `// Optimize hyperparameters, such as embedding dimension, learning rate etc.`
    ...
end for

Latex

（還在做，請韻蓁幫忙研究）

模型參數

Hyperparameter tuning using Bayesian optimization
Note. The hyper-parameter range is based on my own experience.

Category	**Model**	**Embedding Dimension {64,128,256}**	**Learning Rate (log scale) [0.001, 0.1)**	**Batch Size {128, 256, 512}**	**Epochs (Applied early stopping)**
Translation Distance Models	TransE(2013)				
	TransH(2014)				
	TransR(2015)				
Tensor Factorization Models	RESCAL(2011)				
	DistMult(2014)				
	ComplEx(2016)				
GCN-based Model	R-GCN(2018)				
	CompGCN(2020)				

實驗結果

Category	Downstream Task \Model	Link Prediction		Triplet Classification		Entity Dimension	Relation Dimension	Training Time (500 epoches)
		MRR	hit@10	Accuracy	F1 score			
Translation Distance Models	TransE(2013)	0.64	0.73	0.978	-	50	50	1 hr. 55 mins
	TransH(2014)	0.64	0.73	0.981		50	50	2 hr. 40 mins
	TransR(2015)	0.69	0.77	0.985	-	50	50	4 hr. 45 mins
Tensor Factorizati on Models	RESCAL(2011)				-	50	50*50	11 hr. 40 mins
	DistMult(2014)	0.002	0.002	0.56	-	50	50	2 hr. 02 mins
	ComplEx(2016)	0.11	0.22	0.901	-	50	50	3 hr. 30 mins
NN-based Model	R-GCN(2018)							
	CompGCN(2020)							

Observation and Discussion

- Translation Distance Models is much better than Semantic Matching Models
- 待補