Information

- A Survey on Knowledge Graphs: Representation, Acquisition, and Applications
- Shaoxiong Ji , Shirui Pan , *Member, IEEE*, Erik Cambria , *Fellow, IEEE*, Pekka Marttinen , and Philip S. Yu , Life Fellow, IEEE
- URL: https://ieeexplore.ieee.org/abstract/document/9416312
- Code: no code
- other: this summary is mainly on the part of KGE over the survey.

Motivation

- Want to detect the new direction for Knowledge Graph, and the Knowledge Graph Embedding is a good direction, therefore, find this survey to learn and try to find some potential and interesting detailed directions.

Overview

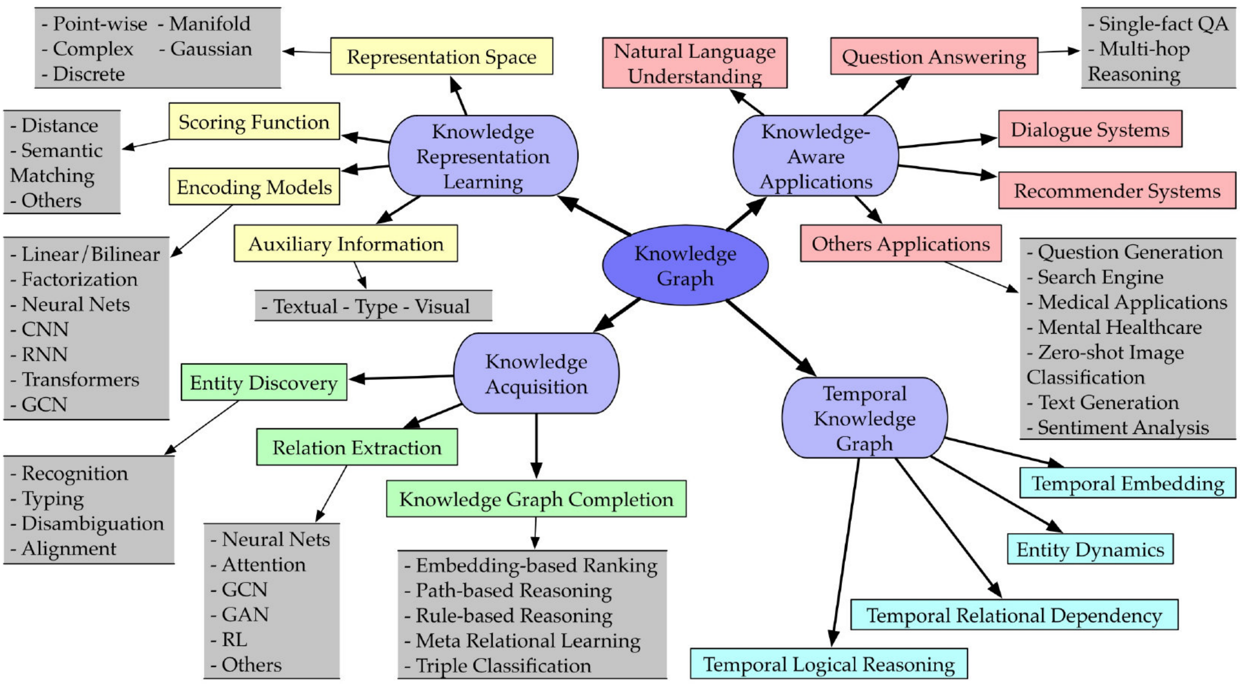
- The part of KGE(Knowledge Representation Learning) is in the left-top corner.
    - 

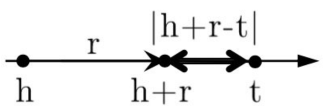
    Fig. 2. Categorization of research on knowledge graphs.

- this part can be divided into four sections:

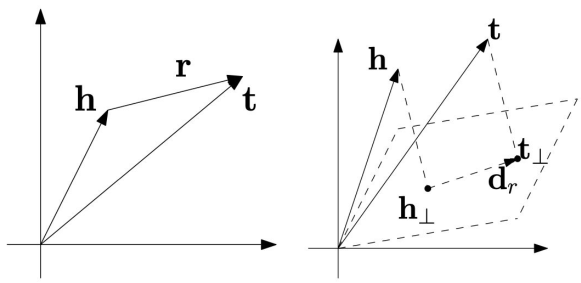
- **Representation Space**:
  - which space the relations and entities are represented.
  - The embedding space should follow three conditions, i.e., differentiability, calculation possibility, and definability of a scoring function [1].
  - Real-valued pointwise space: vector, matrix, and tensor space
  - other space: complex vector space, Gaussian space, and manifold.
- **Scoring Function**:
  - Measure the plausibility of factual triples
- **Encoding Models**:
  - represent and learn relational interaction.
- **Auxiliary Information**:
  - Incorporate other information to help KG Embedding

Main Content

Representation Space

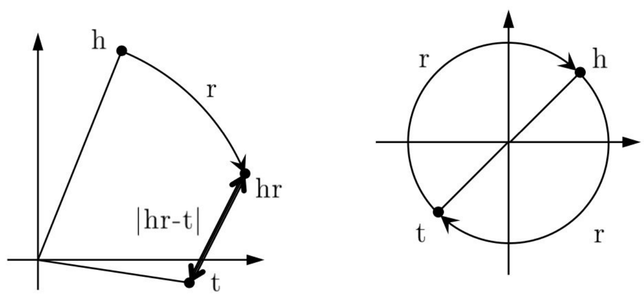
- **Pointwise Space** (Pointwise Euclidean Space)
  - TransE[2]: the relations and entities in d-dimension vector space: **h**+**r** ≈ **t**
    - 

      (a) TransE models **r** as translation in real line.
  - TransR[3]: divide the relation and entity space, and then project entity (**h**,**t** in R^k) into relation space by a projection matrix **M**.
  - NTN[4]: (topic about KGC, important)
    - $$h^T M t, M \in \mathbb{R}^{d*d*k}$$
    - M is a tensor which indicates the relational interaction between head and tail.
  - HAKE[5]: map the relations and entities into polar coordinates to capture the semantic hierarchies

    - concentric circles in the polar coordinate system can naturally reflect the hierarchy.
    - the radius coordinate aims to model entities at different levels of hierarchies.
    - the angular coordinate aims to distinguish entities at the same level of hierarchies.
    - $$e_m \in \mathbb{R}^d, e_p \in [0, 2\pi)^d$$
  - TransH[17]: models a relation as a hyperplane together with a translation operation on it.
    - TansE can not handle problems of reflexible, one-to-many, many-to-one, and many-to-many. TransH can do that.
    - TransH Utilizes the one-to-many/many-to-one mapping property of relation, they propose a simple trick to reduce the possibility of false-negative labeling to solve the incomplete KGs problem.
    - 

(a) TransE (b) TransH

Figure 1: Simple illustration of TransE and TransH.

- **Complex Vector Space**
  - $$h, t, r \in \mathbb{C}^d, h = Re(h) + i \cdot Im(h)$$
  - ComplEx[6]: can capture symmetric and antisymmetric relations, and use Hermitian dot product (dot product for complex).
  - RotatE[7]: use Euler's identity and elementwise Hadmard product
    - $$e^{i\theta} = \cos\theta + i\sin\theta, t = h \circ r$$
    - 

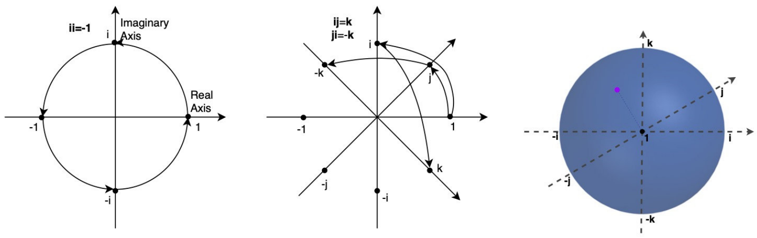
(b) RotatE models $\mathbf{r}$ as rotation in complex plane.

(c) RotatE: an example of modeling symmetric relations $\mathbf{r}$ with $r_i = -1$

    - capture the symmetric, antisymmetric, inversion, and composition.
    - defines each relation as a rotation from the head entity to the target entity in the complex vector space.

- a novel self-adversarial negative sampling technique for efficiently and effectively training the RotatE model.

- QuatE[8]: extend the complex-valued space into hypercomplex.
  - use Quaternion:
    - $$Q = a + bi + cj + dk$$
  -
  - Hamilton product is used as a compositional operator for head entity and relation:
    - $$h \otimes r$$
  -



(a) (b) (c)

Figure 1: (a) Complex plane; (b) Quaternion units product; (c) sterographically projected hypersphere in 3D space. The purple dot indicates the position of the unit quaternion.

  - advantages
    - Latent inter-dependencies (between all components) are aptly captured with Hamilton product.
    - Quaternions enable expressive rotation in four-dimensional space and have more degree of freedom than rotation in complex plane
    - The proposed framework is a generalization of ComplEx on hypercomplex space while offering better geometrical interpretations, concurrently satisfying the key desiderata of relational representation learning (i.e., modeling symmetry, anti-symmetry and inversion)

- **Gaussian Distribution**
  - KG2E[9]: introduces Gaussian distribution to deal with the (un)certainties of entities and relations.
    - embed entity(**H**) and relation(**T**) into multidimensional Gaussian distribution :
      - u means the position of entity and relation position in the complex space, E means their uncertainty.
      - $$H \sim N\ (u_h, \mathbf{\Sigma}_h)\ , T \sim N\ (u_t, \mathbf{\Sigma}_t)$$
    - the probability distribution of entity transformation H - T is donated as:
      - $$P_e \sim N(u_h - u_t, \mathbf{\Sigma}_h + \mathbf{\Sigma}_t)$$

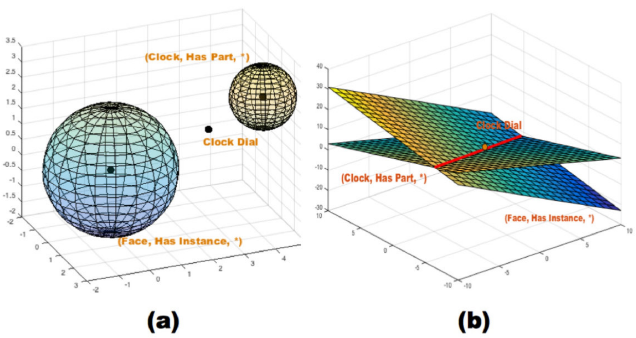
        - popular entities with fewer uncertainty, which contain more relations and facts than unpopular ones, high-frequency rela- tions with more uncertainty, which link more entity pairs than low frequency ones
    - TransG[10]: similar to KG2E, but this paper represents entities with Gaussian distributions, while it draws a mixture of Gaussian distribution for relation embedding, where the *m*th component translation vector of relation *r* is denoted as:
        - $$\mathbf{u}_{r,m} = t - h \sim \mathbf{N}(u_t - u_h, (\sigma_h^2 + \sigma_t^2)\mathbf{E})$$
        - TransG can discover the latent semantics of a relation automatically and leverage a mixture of relation components for embedding.
- **Manifold and Group** (KGE in manifold space, lie group, dihedral group)
    - basic definition:
        - Manifiold: A manifold is a topological space, which could be defined as a set of points with neighborhoods by the set theory
        - Group: The group is algebraic structures defined in abstract algebra.
    - ManifoldE[11]: for the link prediction.
        - the pointwise modeling has two problem:
            - the number of scoring equations is far more than the number of entities and relations, resulting in the problem of **ill-posed algebraic system**.
            - embeddings are restricted in an **overstrict geometric** form even in some methods with subspace projection.
        - this paper proposes two settings of manifold-based embedding:
            - Sphere: reproducing the Kernel Hilbert space.
            - Hyperplane: enhance the model with intersected embedding.
            - 

Figure 2: Visualization of embedding for Manifold-based models. (a) corresponds to the Sphere setting where all the tail entities are supposed to lay in the sphere. As Clock Dial is matched by the two facts, it should lay in both spheres. (b) corresponds to the Hyperplane setting where Clock Dial should lay and does lay in both hyperplanes, making embedding more precise.

    - MuRP[12]: epresents the multirelational knowledge graph in the Poincaré ball of hyperbolic space.

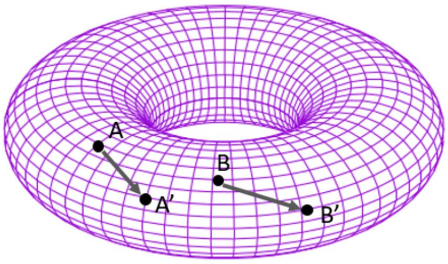
- ATTH[13]: utilize a class of hyperbolic KG embedding models that capture hierarchical and logical patterns.
- TorusE[14]: TransE has the regularization problem, TorusE try to use torus (Lie group) to solve it.
-
    - project from vector space into torus space.
    - $$[h] + [r] \approx [t]$$
    - 

      Figure 3: The image of embeddings on 2-dimensional torus obtained by TorusE. Embeddings of the triples (A, r, A') and (B, r, B') are illustrated. Note that $[\boldsymbol{A}'] - [\boldsymbol{A}]$ and $[\boldsymbol{B}'] - [\boldsymbol{B}]$ are similar on the torus.
- DihEdral[15]: a dihedral symmetry group preserving a 2-D polygon.

Scoring Function

- **Basic information**
    - The scoring function is used to measure the plausibility of facts.
    - Two types:
        - Distance-based function (Euclidean distance):
            - $$h + r \approx t$$
        - Similarity-based function:
            - $$\mathbf{h}^\top \mathbf{M}_r \approx \mathbf{t}^\top$$
        -

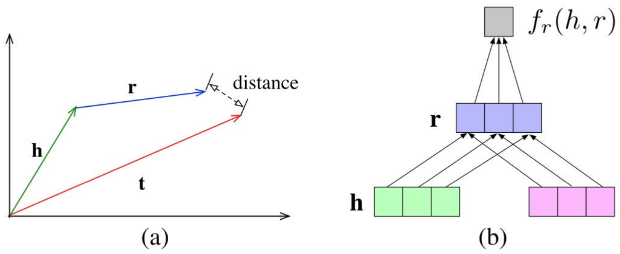
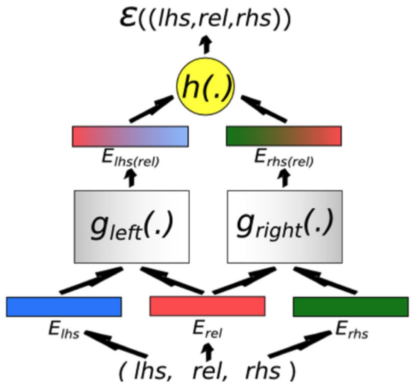


Fig. 4. Illustrations of distance-based and similarity matching-based scoring functions taking TransE [16] and DistMult [32] as examples. (a) Translational distance-based scoring of TransE. (b) Semantic similarity-based scoring of DistMult.

- **Distance-Based Scoring Function**:
  - TransE:
    - $$f_r(h,t) = \|h + r - t\|_{L_1/L_2}$$
  - TransF:
    - $$f_r(h,t) = (\mathbf{h} + \mathbf{r})^\top \mathbf{t}$$
  - ITransF[15]: enables **hidden concepts discovery** and **statistical strength transferring** by learning associations between relations and concepts via **sparse attention vectors**:
    - $$f_r(h,t) = \|\alpha_r^H \cdot \mathbf{D} \cdot \mathbf{h} + \mathbf{r} - \alpha_r^T \cdot \mathbf{D} \cdot \mathbf{t}\|_l$$
    - $$\mathbf{D} \in \mathbb{R}^{n*d*d}. \alpha_r^H, \alpha_r^T \in [0,1]^n$$
    - **D** is stacked concept projection matrices of entities and relations, **alpha** is attention vectors calculated by sparse softmax.
  - TransMS: trans- lates and transmits multidirectional semantics.
    - he semantics of head/tail entities and relations to tail/head entities with **nonlinear functions**
    - the semantics from entities to relations with **linear bias vectors**
    - $$f_r(\mathbf{h}, \mathbf{t}) = \| - \tanh(\mathbf{t} \circ \mathbf{r}) \circ \mathbf{h} + \mathbf{r} - \tanh(\mathbf{h} \circ \mathbf{r}) \circ \mathbf{t} + \alpha \cdot (\mathbf{h} \circ \mathbf{t})\|_{\ell_{1/2}}.$$
  - KG2E: (Gaussian Space) use 1, asymmetric KL-divergence. 2, symmetric expected likelihood.
  - ManifoldE:
    - $$f_r(h,t) = \left\| \mathcal{M}(h,r,t) - D_r^2 \right\|^2 \quad (5)$$
      where $\mathcal{M}$ is the manifold function, and D_r is a relation-specific manifold parameter.

- **Semantic Matching** (Similarity-based sccoring function)
  - SME[16]: match separate combinations of entity-relation pairs (h, r) and (r, t).

- **Fig. 1** Semantic matching energy function. A triple of entities (lhs, rel, rhs) is first mapped to its embeddings E_{lhs}, E_{rel} and E_{rhs}. Then E_{lhs} and E_{rel} are combined using $g_{left}(.)$ to output $E_{lhs(rel)}$ (similarly $E_{rhs(rel)} = g_{right}(E_{rhs}, E_{rel})$). Finally the energy $\mathcal{E}((lhs, rel, rhs))$ is obtained by matching $E_{lhs(rel)}$ and $E_{rhs(rel)}$ with the $h(.)$ function



- $f_r(h, t) = g_{\text{left}}(\mathbf{h}, \mathbf{r})^\top g_{\text{right}}(\mathbf{r}, \mathbf{t}).$

  - CrossE[18]: It not only learns one general embedding for each entity and relation as in most previous methods, but also generates multiple triple specific embeddings for both of them, named interaction embeddings.
    - use interaction matrix **C** to simulate the bidirectional interaction between entity and relation.
    - between entity and relation. The relation specific interaction is obtained by looking up interaction matrix as $\mathbf{c}_r = \mathbf{x}_r^\top \mathbf{C}$. By combining the interactive representations and matching with tail embedding, the scoring function is defined as

$$f(h, r, t) = \sigma\left(\tanh(\mathbf{c}_r \circ \mathbf{h} + \mathbf{c}_r \circ \mathbf{h} \circ \mathbf{r} + \mathbf{b})\mathbf{t}^\top\right). \tag{9}$$

  - TorusE[14] and DihEdral[15] also use sementic matching:
    - for DihEdral：
    - $$f_r(h, t) = \mathbf{h}^\top \mathbf{R} \mathbf{t} = \sum_{l=1}^{L} \mathbf{h}^{(l)\top} \mathbf{R}^{(l)} \mathbf{t}^{(l)} \tag{11}$$

      where the relation matrix $\mathbf{R}$ is defined in block diagonal form for $\mathbf{R}^{(l)} \in \mathbb{D}_K$, and entities are embedded in real-valued space for $\mathbf{h}^{(l)}$ and $\mathbf{t}^{(l)} \in \mathbb{R}^2$.

Encoding Models

- **Basic information**: encode the interactions of entities and relations by using model architecture.
  - Linear/Bilinear models: regard relations as a linear/bilinear mapping by projecting head entities into a space close to tail entities.
  - Factorization: decompose relation into a low-rank matrices for representation learning.
  - Neural networks: encode relational with nonlinear neural activation and complex network to represent complex sementic similarity of entities and relations.
- **Linear/Bilinear Models**
  - Definition:

双线性函数

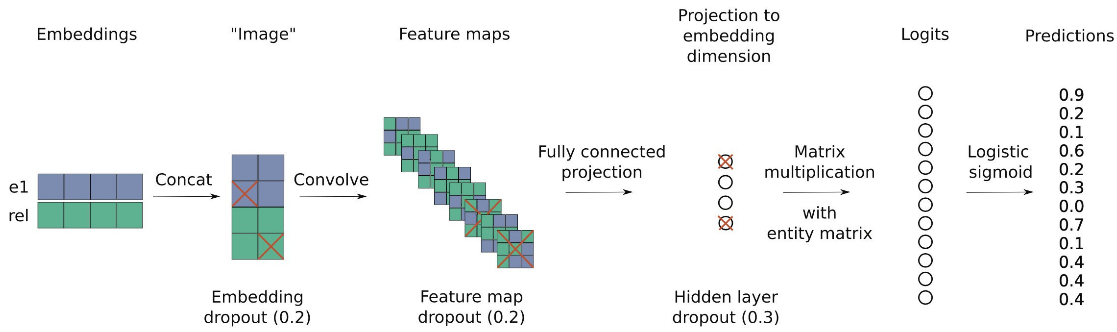
在线性函数的基础上, 我们可以讨论在代数学和几何学中都有重要作用的多元线性函数. 首先考虑二元情形.

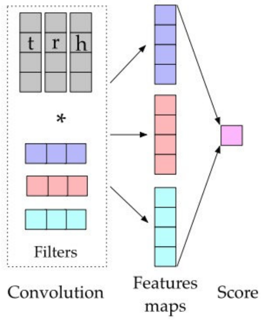
- 设线性空间 V 上的二元函数 $f(\alpha,\beta)$ 为**双线性函数**或**双线性型**, 如果对任意 $k,l\in\mathbb{F}$, $\alpha,\beta,\gamma\in V$, 有

$$f(k\alpha+l\beta,\gamma)=kf(\alpha,\gamma)+lf(\beta,\gamma);$$
$$f(\gamma,k\alpha+l\beta)=kf(\gamma,\alpha)+lf(\gamma,\beta).$$

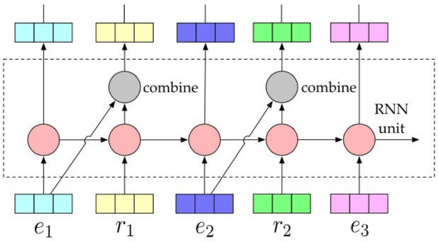
- traditional linear/bilinear model like SE, SME, DistMult, ComplEx, and ANALOGY, use:
  - $g_r(\mathbf{h},\mathbf{t})=\mathbf{M}_r^T\begin{pmatrix}\mathbf{h}\\\mathbf{t}\end{pmatrix}$
  - Bilinear: RESCAL, DistMult, HolE, ComplEx.
- TransE[2] use L2 regularization:
  - $\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2=2\mathbf{r}^T(\mathbf{h}-\mathbf{t})-2\mathbf{h}^T\mathbf{t}+\|\mathbf{r}\|_2^2+\|\mathbf{h}\|_2^2+\|\mathbf{t}\|_2^2.$
- SimplE[19]:
- - $f_r(h,t)=\frac{1}{2}(\mathbf{h}\circ\mathbf{r}\mathbf{t}+\mathbf{t}\circ\mathbf{r}'\mathbf{t})$
  - r' is the embedding of inversion relation.

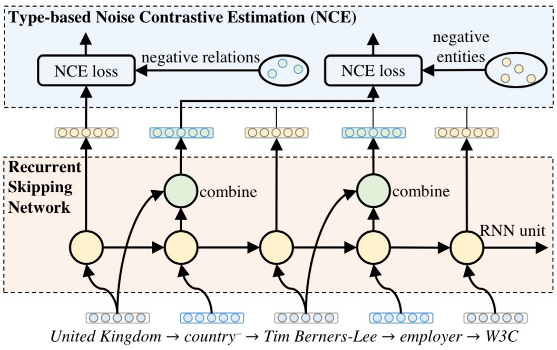
- **Factorization Models**
  - Definition:
    - tensor factorization can be denoted as:
    - $$\chi_{hrt}\approx\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$$
  - RESCAL[20]:
    - for the *k*th relation of m relations, the *k*th slice of **X** is factorized as:
      - $\mathcal{X}_k\approx\mathbf{A}\mathbf{R}_k\mathbf{A}^T.$
- **Neural Networks**
  - Basic:
    - Linear/bilinear model can be modeled by NN.
    - feed the entities and rellations into NN and get score output.
  - **CNN**
    - ConvE[21]: use 2D convelution over embedding.
      - $f_r(h,t)=\sigma(\mathrm{vec}(\sigma([\mathbf{M}_h;\mathbf{M}_r]*\boldsymbol{\omega}))\mathbf{W})\mathbf{t}$
      - ; means concatenate, * means conv operation, w means conv filter, Vec means reshape a tensor into a vector.

- 

- ConvKB[22]:
  - $f_r(h,t) = \text{concat}(\sigma([\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}] * \boldsymbol{\omega})) \cdot \mathbf{w}$
  - 

  - compared with ConvE, ConvKB can capture local relationships.

- **RNN**
  - RSN[23]:
    - 

    - 

- **Transformer**
  - In order to add contextual information into KG, CoKE[24] employs transformers to encode edges and path sequences:

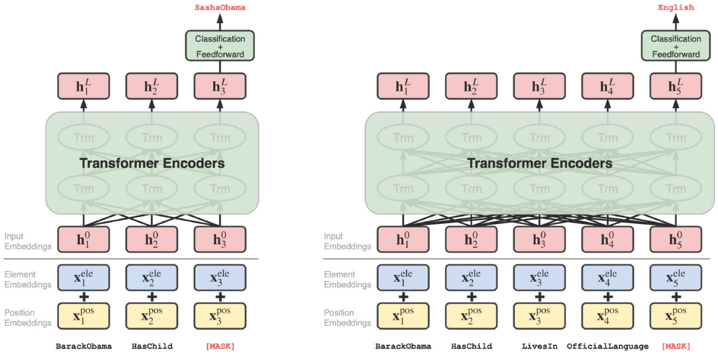
- 

  Figure 2: Overall framework of CoKE. An edge (left) or a path (right) is given as an input sequence, with an entity replaced by a special token [MASK]. The input is then fed into a stack of Transformer encoder blocks. The final hidden state corresponding to [MASK] is used to predict the target entity.

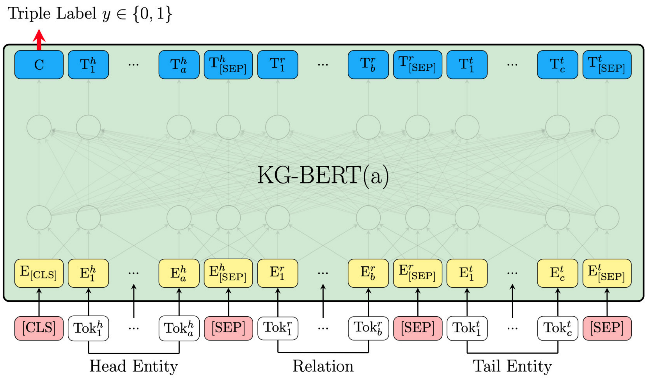
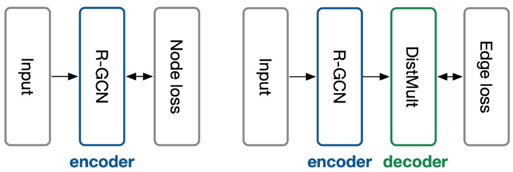
- KG-BERT[25] use BERT to encode entities and relations:
  - 

    Figure 1: Illustrations of fine-tuning KG-BERT for predicting the plausibility of a triple.

- **GNN**
  - use R-GCN[26] to do the Link prediction (recovery of missing facts, i.e. subject-predicate-object triples) and en- tity classification (recovery of missing entity attributes):
    - $$x_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}x_j^{(l)} + W_0^{(l)}x_i^{(l)}\right)$$
    - 

      (a) Entity classification (b) Link prediction

      Figure 3: (a) Depiction of an R-GCN model for entity classification with a per-node loss function. (b) Link prediction model with an R-GCN encoder (interspersed with fully-connected/dense layers) and a DistMult decoder that takes pairs of hidden node representations and produces a score for every (potential) edge in the graph. The loss is evaluated per edge.
  - SACN[27] use weight GCN(WGCN) as encoder, use Conv-TransE as decoder to combine the advantages of GCN and ConvE.

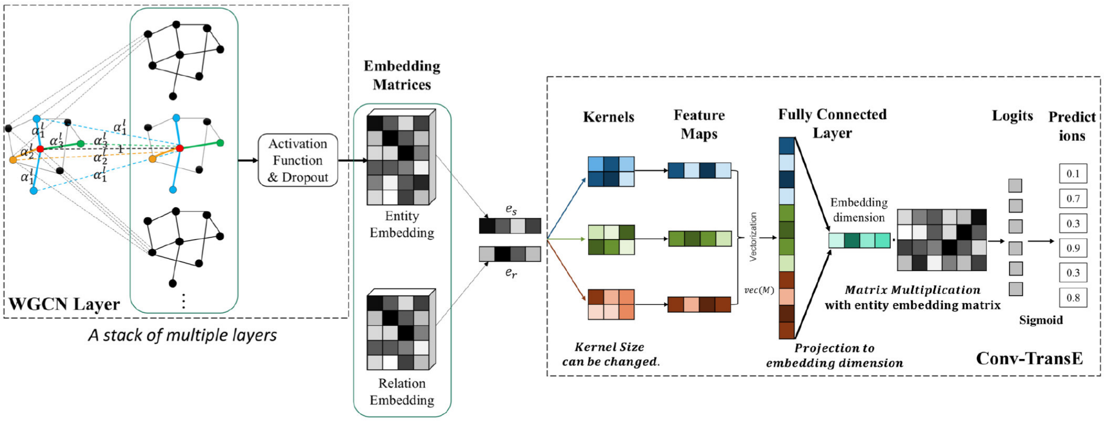
- 

  Figure 1: An illustration of our end-to-end Structure-Aware Convolutional Networks model. For encoder, a stack of multiple WGCN layers builds an entity/node embedding matrix. For decoder, e_s and e_r are fed into *Conv-TransE*. The output embeddings are vectorized and projected, and matched with all candidate e_o embeddings via inner products. A logistic sigmoid function is used to get the scores.

- CompGCN

Embedding With Auxiliary Information

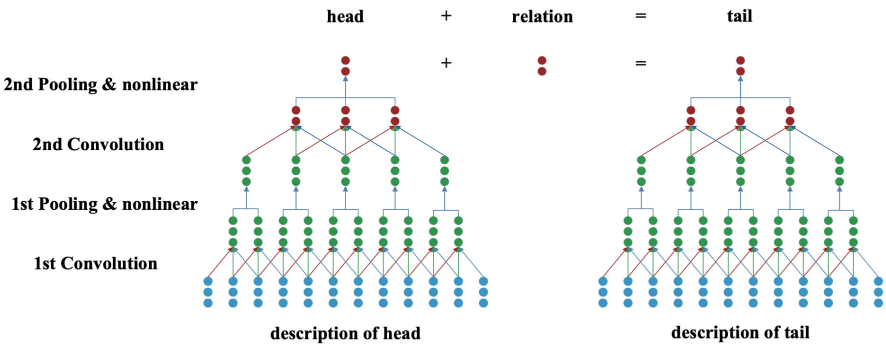
- **Textual Description**
  - DKRL[28]:
    - use Convelutional encoder.
    - 

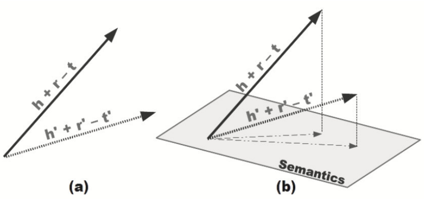
      Figure 3: The Convolutional Neural Network Encoder
  - SSP[29]:
    - project triples and textual information into semantic space:
    - 

      Figure 2: Simple illustration of TransE and SSP where $\mathbf{h} + \mathbf{r} - \mathbf{t}$ is the loss vector. The loss vectors of the two triples in (a) are length-equal, thus, it is hard to identify the correctness.In (b), we introduce a semantic hyperplane, and project the loss vectors to the hyperplane to consider the semantics of triples.

- **Type Information**
    - entities are represented with heirarchical classes or types, and relations with sementic types.
- **Visual Information**
    - utilize visual information to embed entities.
- **Uncertain Information**
    - uncertain embedding models aim to capture uncertainty rep- resenting the likelihood of relational facts.

Summary

- Which representation space to choose.
- how to measure the plausibility of triples in a specific space.
- which encoding model to use for modeling relational interactions.
- whether to ultilize auxiliary information.

Review

- for the representation space part, we can use several space like normal Euclidean space, complex space, manifold, etc. Each kind of space has its own benefits.
- For the scoring part, we need to use semantic matching now.
- but for the encoding models, we need to use NN models, but not the traditional linear models. LMs and transformers need to be utilized well in the KGE.
- if we can, we need to use extention auxiliary information to enhance the KGE, especially for large textual information.
- In conclusion, I need to find some paper about KGE recently, grasp the detail about the KGE，and undderstand the benchmark and how to evaluate the KGE from the experiments.
- a good KGE, and a suitbale representation space, encoding model can help us to do NLP task which need to use KG and Graph Structure.

Reference

1. T. Ebisu and R. Ichise, “TorusE: Knowledge graph embedding on a lie group,” in Proc. AAAI, 2018, pp. 1819–1826.
2. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in Proc. NIPS, 2013, pp. 2787–2795.
3. Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in Proc. AAAI, 2015, pp. 2181–2187.
4. R. Socher, D. Chen, C. D. Manning, and A. Ng, “Reasoning with neural tensor networks for

knowledge base completion," in Proc. NIPS, 2013, pp. 926–934.
5. Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in Proc. AAAI, 2020, pp. 3065–3072.
6. T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in Proc. ICML, 2016, pp. 2071–2080.
7. Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in Proc. ICLR, 2019, pp. 1–18.
8. S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embedding," in Proc. NeurIPS, 2019, pp. 2731–2741.
9. S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with Gaussian embedding," in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage., Oct. 2015, pp. 623–632.
10. H. Xiao, M. Huang, and X. Zhu, "TransG: A generative model for knowledge graph embedding," in Proc. ACL, vol. 1, 2016, pp. 2316–2325.
11. H. Xiao, M. Huang, Y. Hao, and X. Zhu, "From one point to a manifold: Orbit models for knowledge graph embedding," in Proc. IJCAI, 2016, pp. 1315–1321.
12. I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," in NeurIPS, 2019, pp. 4463–4473.
13. I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Lowdimensional hyperbolic knowledge graph embeddings," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 6901–6914.
14. T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," in Proc. AAAI, 2018, pp. 1819–1826.
15. Q. Xie, X. Ma, Z. Dai, and E. Hovy, "An interpretable knowledge transfer model for knowledge base completion," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2017, pp. 950–962.
16. A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," Mach. Learn., vol. 94, no. 2, pp. 233–259, Feb. 2014.
17. Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in Proc. AAAI, 2020, pp. 3065–3072.
18. W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in Proc. 12th ACM Int. Conf. Web Search Data Mining, Jan. 2019, pp. 96–104.
19. S. M. Kazemi and D. Poole, "SimplE embedding for link prediction in knowledge graphs," in Proc. NeurIPS, 2018, pp. 4284–4295.
20. M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in Proc. ICML, vol. 11, 2011, pp. 809–816.
21. T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in Proc. AAAI, vol. 32, 2018, pp. 1811–1818.
22. D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol., vol. 2, 2018,
23. L. Guo, Z. Sun, and W. Hu, "Learning to exploit long-term relational dependencies in knowledge graphs," in Proc. ICML, 2019, pp. 2505–2514.
24. Q. Wang et al., "CoKE: Contextualized knowledge graph embedding," 2019, arXiv:1911.02168. [Online]. Available: http://arxiv.org/abs/1911.02168.

25. L. Yao, C. Mao, and Y. Luo, “KG-BERT: BERT for knowledge graph completion,” 2019, arXiv:1909.03193. [Online]. Available: http://arxiv.org/abs/1909.03193
26. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in Proc. ESWC, 2018, pp. 593–607.
27. C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, “Endto-end structure-aware convolutional networks for knowledge base completion,” in Proc. AAAI, vol. 33, 2019, pp. 3060–3067.
28. R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in Proc. AAAI, 2016, pp. 2659–2665.
29. H. Xiao, M. Huang, L. Meng, and X. Zhu, “SSP: Semantic space projection for knowledge graph embedding with text descriptions,” in Proc. AAAI, 2017, pp. 3104–3110.