MSGE:A Multi-Step Gated Model for Knowledge Graph Completion

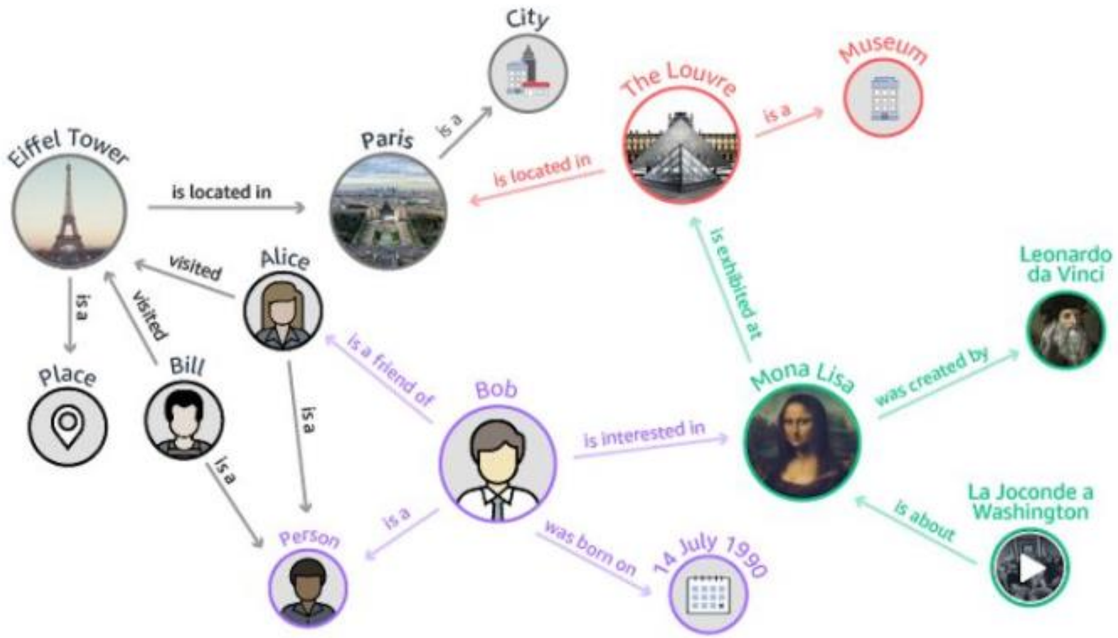
Chunyang Tan

Outline

- Introduction
- Related Work
- Motivation
- Our Model: MSGE
- Experiments
- Conclusion
- References

Introduction:Knowledge Graph Completion

- Knowledge Graph



Amazon's Neptune Database[1]

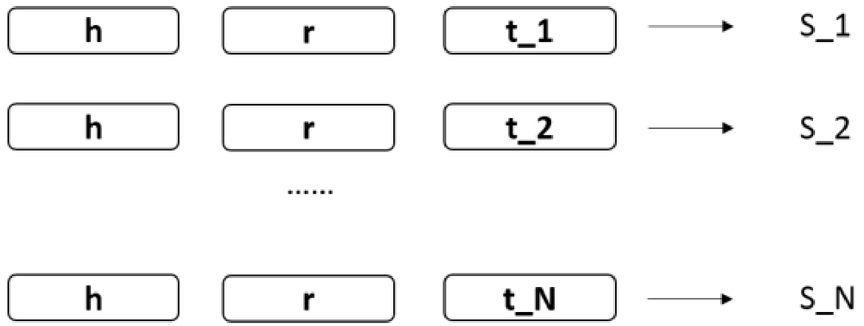
Real-world facts are stored in Knowledge Graph as triples.

Introduction:Knowledge Graph Completion

- Challenge in KGs
  - KGs are far from complete.

Prediction Task to Ranking Task

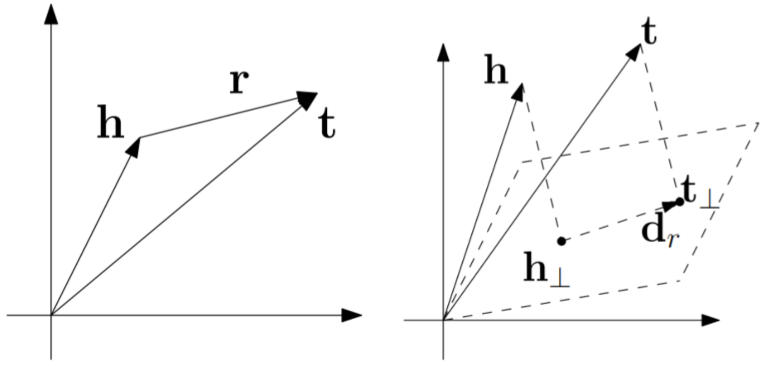
- Task: Knowledge Graph Completion
  - Predict tail entity given head entity and relation.
  - Score all triplets and rank them in order.



Related Work

- Distance-based Model
  - TransE[2]/TransH[3]/RotatE[4]…

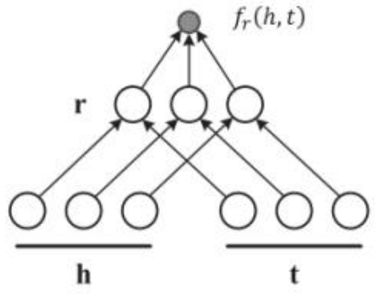
$$f_r(h,t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}$$

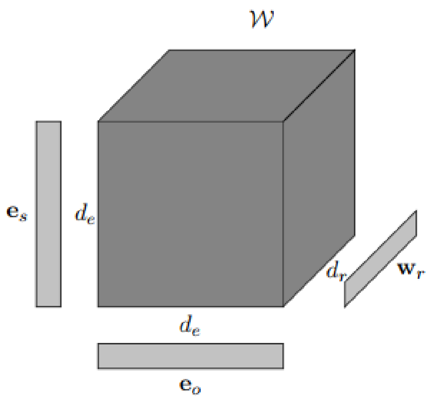


(a) TransE (b) TransH

- Tensor Decomposition Model
  - DistMult[5]/ComplEx[6]/SimplE[7]/TuckER[8]

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



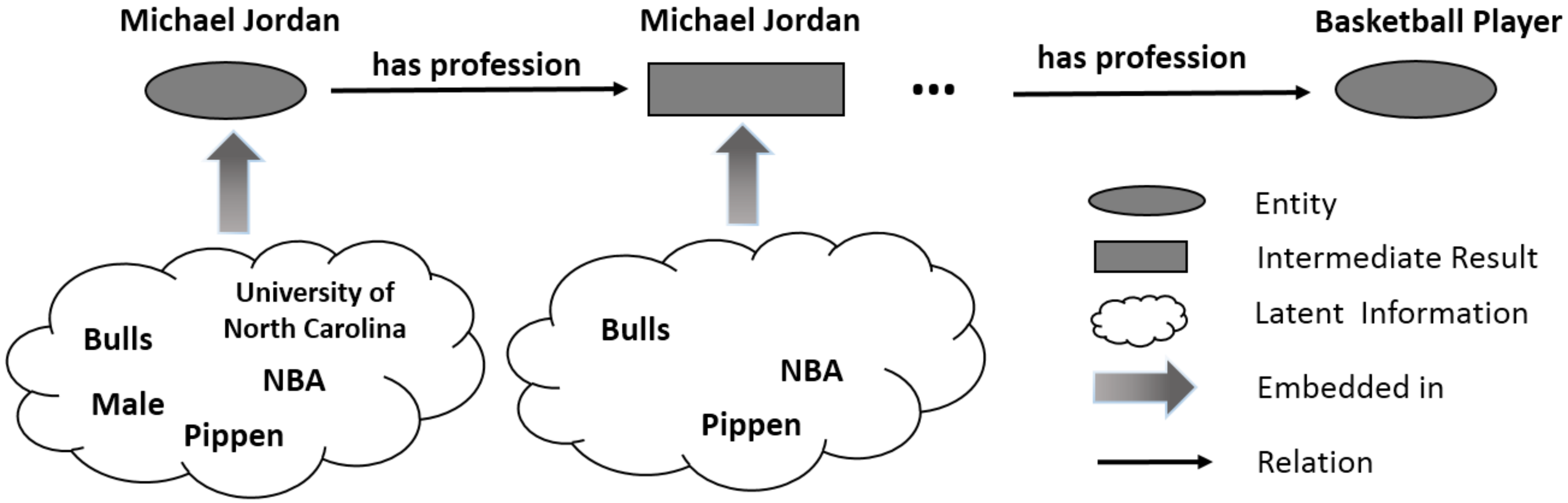


Motivation

- However, these models do not consider:
    1. Controlling information flow specifically
        - Keeping relevant information and filtering out useless ones
    2. The multi-step reasoning nature of a prediction process.

Motivation

1. Controlling information flow specifically.
   - Keeping relevant information and filtering out useless ones.
2. The multi-step reasoning nature of a prediction process.



Our Model: MSGE

- Multi-Step Gate Mechanism
  - Gate mechanism: In LSTM and GRU.
  - **Update** gate **z** and **Reset** gate **r**.

$$\boldsymbol{z} = \sigma(\boldsymbol{W_z}[\boldsymbol{e_r}, \boldsymbol{e_s}] + \boldsymbol{b_z})$$
$$\boldsymbol{r} = \sigma(\boldsymbol{W_r}[\boldsymbol{e_r}, \boldsymbol{e_s}] + \boldsymbol{b_r})$$

- Generate new embeddings
  - New information

$$\boldsymbol{e_s'} = tanh(\boldsymbol{W_s}[\boldsymbol{r} \odot \boldsymbol{e_s}, \boldsymbol{e_r}] + \boldsymbol{b})$$

  - Final output

$$\tilde{\boldsymbol{e_s}} = (1 - \boldsymbol{z}) \odot \boldsymbol{e_s'} + \boldsymbol{z} \odot \boldsymbol{e_s}$$

Our Model: MSGE

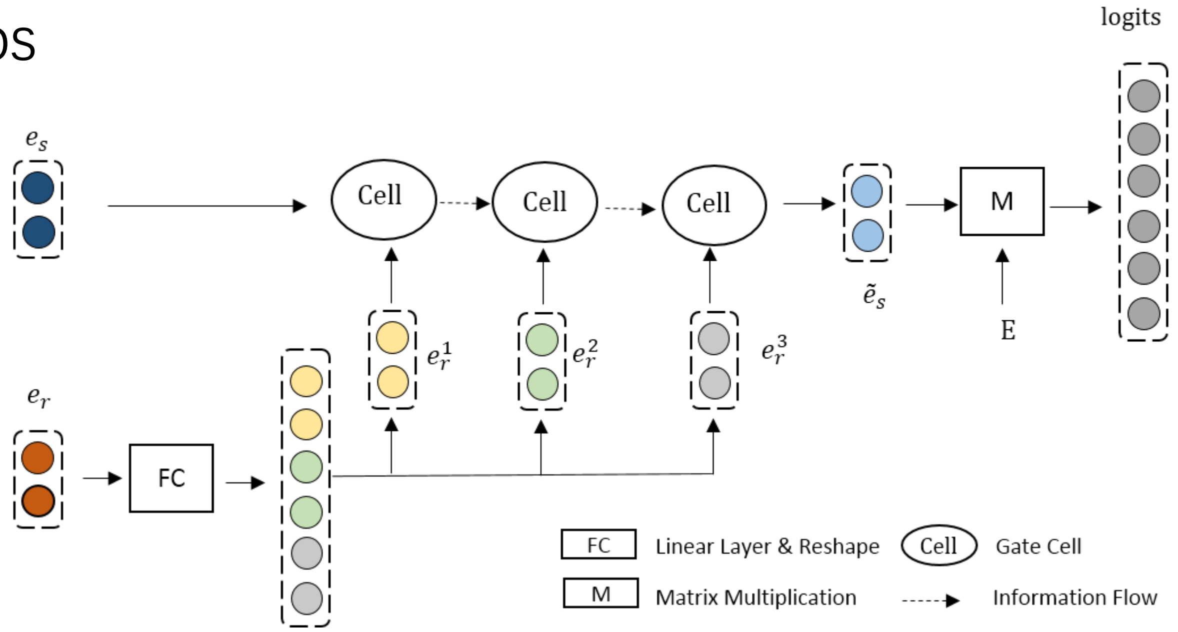
- Iterative Multi-step Architecture
  - Query relations for multiple steps

$$e_r^{'} = \boldsymbol{W}\boldsymbol{e_r} + \boldsymbol{b}$$

$$[\boldsymbol{e_r^0}, \boldsymbol{e_r^1}, ..., \boldsymbol{e_r^k}] = Reshape(\boldsymbol{e_r^{'}})$$

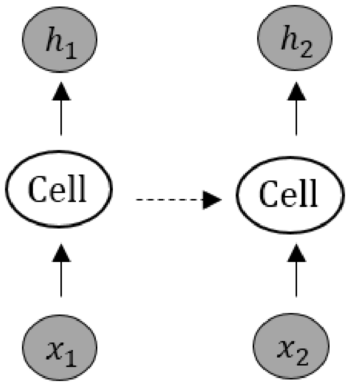
  - Score function
    - C means Gate Cell

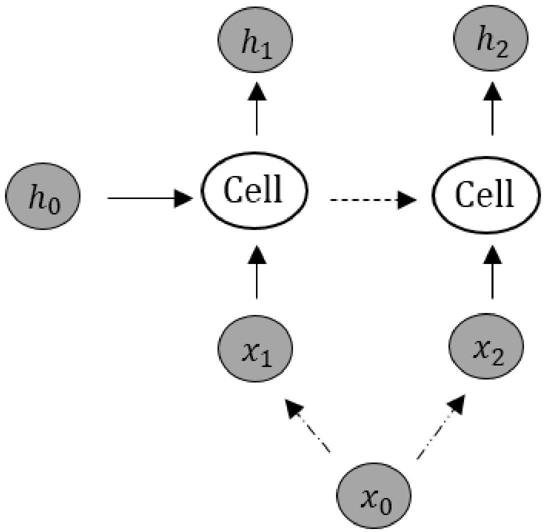
$$\phi = (C^k(\boldsymbol{e_s}, [\boldsymbol{e_r^0}, \boldsymbol{e_r^1}, ..., \boldsymbol{e_r^k}]))\boldsymbol{e_o}$$



Our Model: MSGE

- Differences with RNN-like model
  - In RNN: An input sequence is fed.
    - e.g. words.
  - In MSGE: Two kind of inputs.
    - Entities and Query relations.





Experiments

- Datasets
  - WN18RR and FB15k-237 are datasets without inverse relations

Dataset	#Entity	#Rel	#Train	#Valid	#Test
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134
FB15k	14951	1345	483142	50000	59071
FB15k-237	14541	237	212115	17535	20466
UMLS	135	49	5216	652	661
KINSHIP	104	26	8544	1068	1074

Experiments

- Link Prediction
  - MSGE can achieve SOTA performance.

Model	WN18		WN18RR		FB15k		FB15k-237	
	MRR	*Hit@10*	*MRR*	*Hit@10*	*MRR*	*Hit@10*	*MRR*	*Hit@10*
TransE (2013)	-	0.892	-	-	-	0.471	-	-
DistMult[†] (2015)	0.822	0.936	0.430	0.490	0.654	0.824	0.241	0.419
ComplEx[†] (2016)	0.941	0.947	0.440	0.510	0.692	0.840	0.247	0.428
R-GCN (2017)	0.814	**0.964**	-	-	0.696	0.842	0.248	0.417
TransAt (2018)	-	0.951	-	-	-	0.782	-	-
MINERVA (2018)	-	-	0.448	0.513	-	-	0.293	0.456
ConvE (2018)	0.943	0.956	0.430	0.520	0.657	0.831	0.325	0.501
TorusE (2018)	0.947	0.954	-	-	0.733	0.832	-	-
RotatE (2019)	-	-	-	-	-	-	0.297	0.480
SimplE (2018)	0.942	0.947	-	-	0.727	0.838	-	-
TuckER (2019)	**0.953**	0.958	**0.470**	<u>0.526</u>	<u>0.795</u>	<u>0.892</u>	**0.358**	<u>0.544</u>
MSGE(Ours)	<u>0.951</u>	<u>0.961</u>	<u>0.464</u>	**0.547**	**0.806**	**0.894**	<u>0.357</u>	**0.545**

Experiments

- Link Prediction
  - Perform better compared to Non-Embedding models.

Model	UMLS				Kinship			
	MRR	*Hit@10*	*Hit@3*	*Hit@1*	*MRR*	*Hit@10*	*Hit@3*	*Hit@1*
ComplEx $^{\triangleleft}$ (2016)	0.894	**0.995**	0.962	0.824	0.838	0.980	0.910	0.754
ConvE $^{\triangleleft}$ (2018)	0.933	0.992	0.964	0.894	0.797	0.974	0.886	0.697
NTP (2017)	0.872	0.970	0.906	0.817	0.612	0.777	0.700	0.500
NeuralLP (2017)	0.778	0.962	0.869	0.643	0.619	0.912	0.707	0.475
MINERVA (2018)	0.825	0.968	0.900	0.728	0.720	0.924	0.812	0.605
MSGE(Ours)	**0.946**	0.993	**0.973**	**0.914**	**0.865**	**0.988**	**0.941**	**0.785**

Experiments

- Ablation Study
  - **No gate**: Remove the gates in our model to verify the necessity of controlling information flow.
  - **Concat**: Concatenate information extracted in every step together and feed them into a fully connected layer to obtain another kind of final information.
  - **Replicate**: Replicate the relation to gain k same query relations for training .

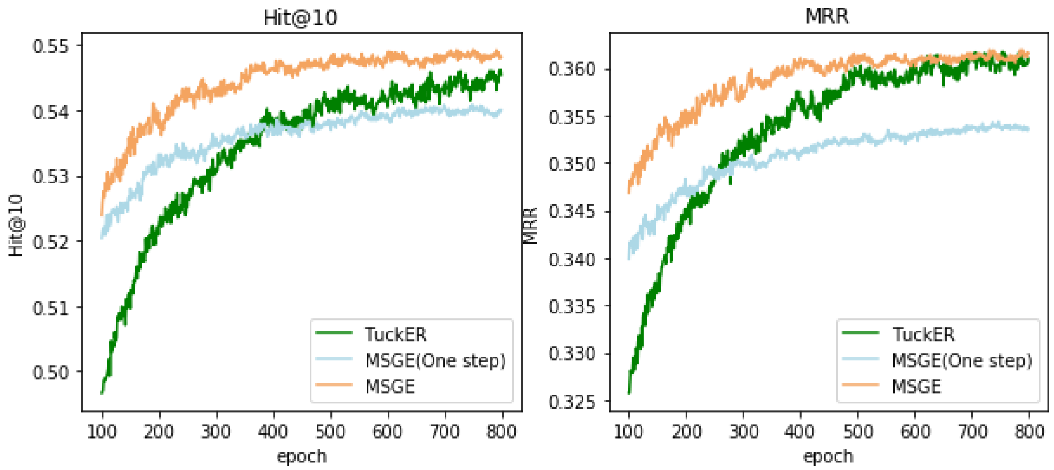
Experiments

- Ablation Study
  - **No gate**
  - **Concat**
  - **Replicate**

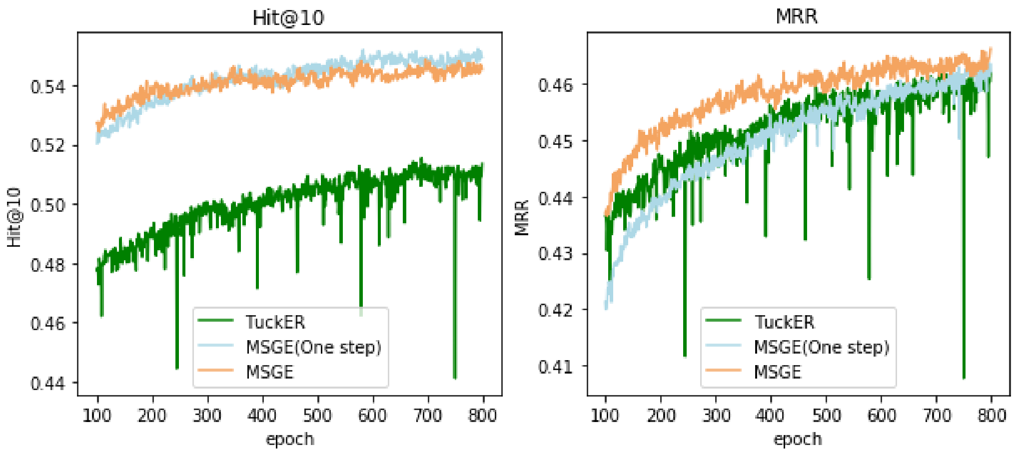
Model	FB15k-237			
	MRR	*Hit@10*	*Hit@3*	*Hit@1*
MSGE	**0.357**	**0.544**	**0.392**	**0.264**
No gate	0.301	0.459	0.327	0.222
Concat	0.349	0.534	0.384	0.256
Replicate	0.351	0.537	0.388	0.257

Experiments

- Convergence Study



FB15k-237



WN18RR

Experiments

- Efficiency Analysis
  - Space Complexity

Dataset	ConvE	TuckER	MSGE
WN18	10.32M	9.39M	8.48M
WN18RR	10.31M	9.39M	8.84M
FB15k	6.16M	11.53M	3.81M
FB15k-237	5.19M	11.00M	3.57M

- Time Efficiency
  - On Telsa K40c.
  - On FB15k-237/WN18RR
  - TuckER needs 29s/28s to run an epoch respectively
  - MSGE needs 17s/24s respectively

Conclusion

- In this paper, we propose a **M**ulti-**S**tep **G**ated **E**mbedding model **MSGE** for link prediction task in knowledge graph completion.
- We do link prediction on many benchmark datatsets comparing with many baseline methods, MSGE achieves SOTA performance.
- Analysis show the effectiveness of MSGE.
- In future, we'd like to incorporate more information for knowledge graph completion.

References

- [1] Source : https://geomarketing.com/amazons-neptune-database-will-expand-the-knowledgegraph-heres-how
- [2] Bordes A., Usunier N., Garcia-Duran A., Weston J., and Yakhnenko O. Translating embeddings for modeling multi-relational data. NIPS, pages 2787-2795, 2013.
- [3] Wang Z., Zhang J., Feng J., and Chen Z. Knowledge graph embedding by translating on hyperplanes. AAAI, 14:1112{1119, 2014.
- [4] Sun Z., Deng Z. H., and Nie J.Y. Rotate: Knowledge graph embedding by relational rotation in complex space. ICLR, 2019.
- [5] Yang B., Yih W., He X., Gao J., and Deng L. Embedding entities and relations for learning and inference in knowledge bases. ICLR, 2015.
- [6] Trouillon T., Welbl J., Riedel S., Gaussier E., and Bouchard G. Complex embeddings for simple link prediction. ICML, pages 2071-2080, 2016.
- [7] Kazemi S.M. and Poole D. Simple embedding for link prediction in knowledge graphs. NIPS, 2018.
- [8] Balaˇzevi´c I., Allen C., and Hospedales T. Tensor factorization for knowledge graph completion. EMNLP, 2019

Q&A