Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings

Published as a conference paper at ACL 2020

Approv Saxena, Aditay Tripathi, Partha Talukdar

2024-11-28

HoonUi Lee

Index

KGQA

EmbedKGQA

Experiment

Conclusion

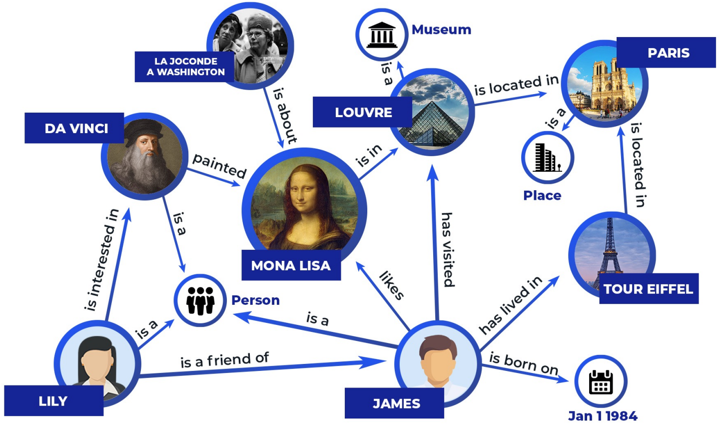
KGQA

❖ KGQA Definition

- Find a **set of nodes** of the knowledge graph to answer the question

- While given data

1. a natural language question
2. the topic entity
   - ➢ The start node in KG to find the answer about question
3. a knowledge graph

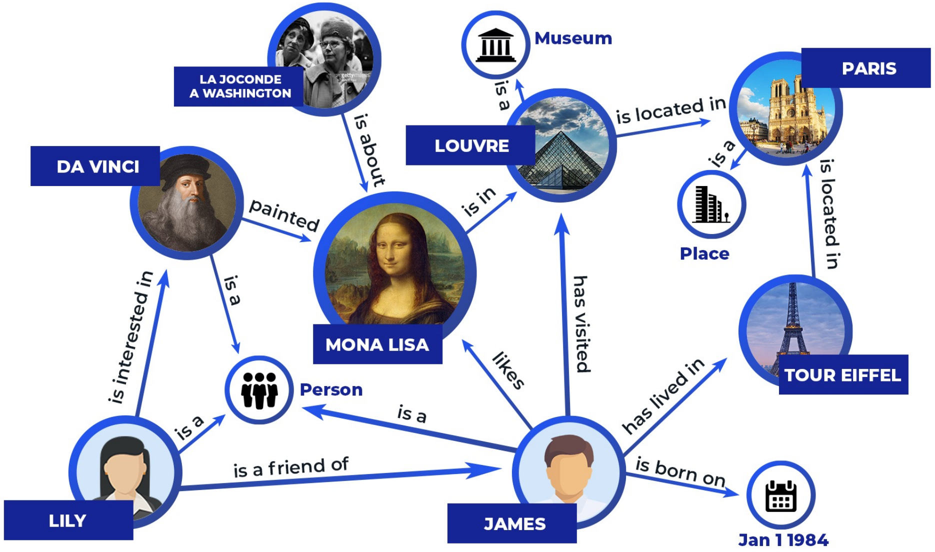


KGQA

❖ KGQA Definition

- **One-hop Question**
  - ➢ Where is the TOUR EIFFEL located in?

- **Multi-hop Question**
  - ➢ Where is the place DA VINCI's painting in?
  - ➢ DA VINCI -> MONA LISA -> LOUVRE
  - ➢ DA VINCI will be topic entity



KGQA

· Previous work

❖ Sparse & incomplete KG

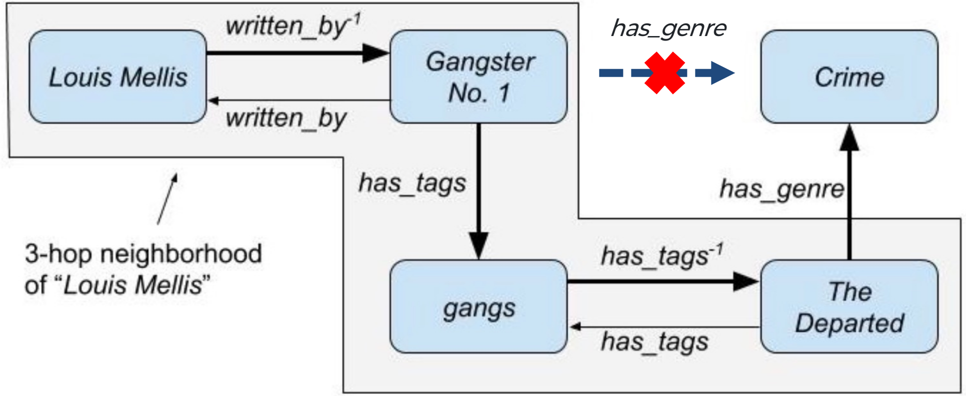
- Break of the reasoning chain
- Increase the length of reasoning path

❖ Two Approaches

- Use additional data (text corpus) to fill KG
  - ➢ Not always available
- Impose neighborhood limits to find few hop first in connected subgraph
  - ➢ Answer might be out of reach

KGQA

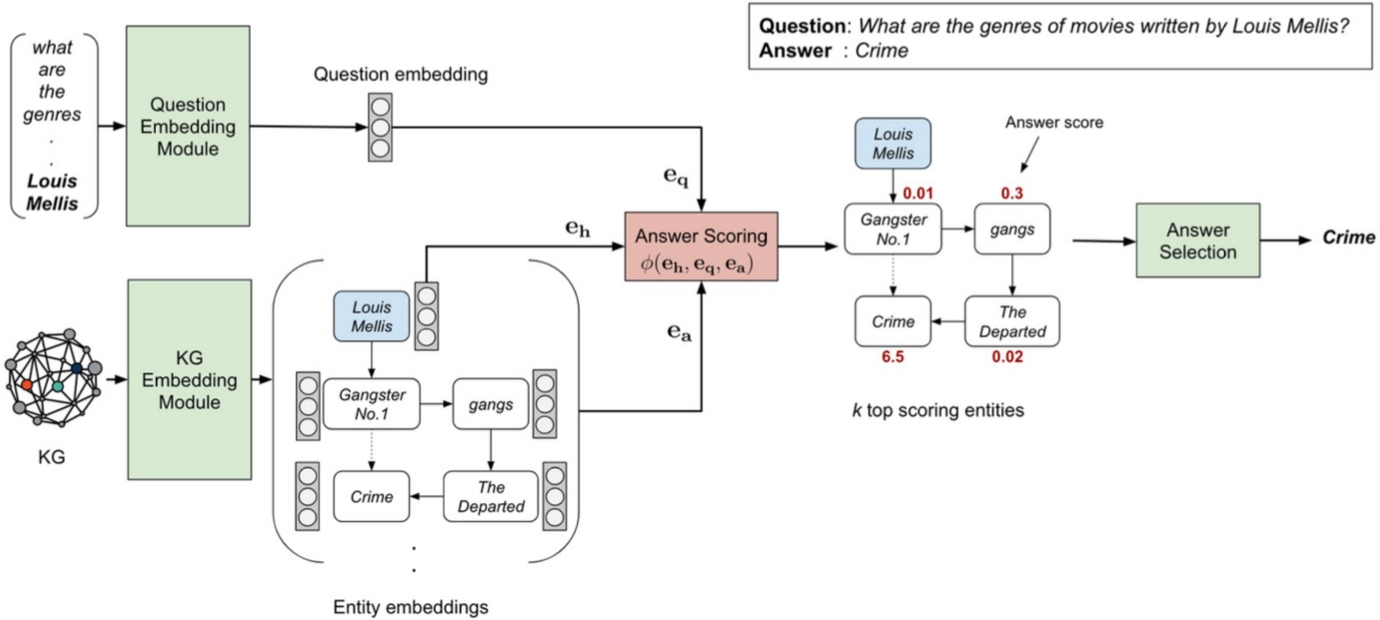
· Previous work



- Absence of the edge *has_genre(Gangster No. 1, Crime)*
  - ➢ needs to reason over a longer path
- Because of incompleteness, make the true answer out of reach
  - ➢ Can't reach for *"Crime"* in 3-hop in this example

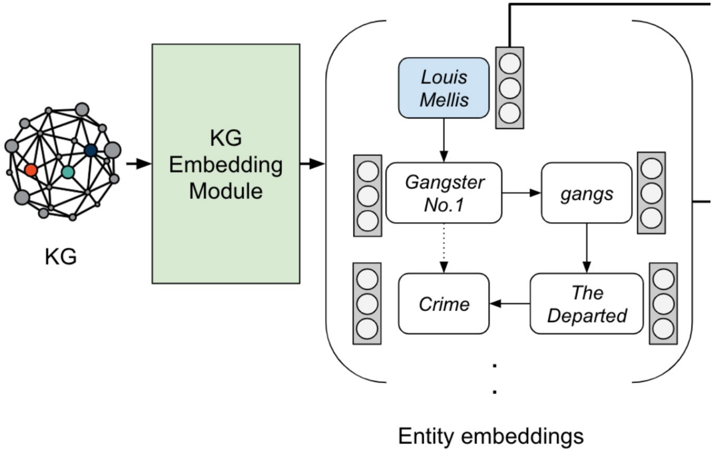
EmbedKGQA

· Overview



EmbedKGQA

· KG embedding

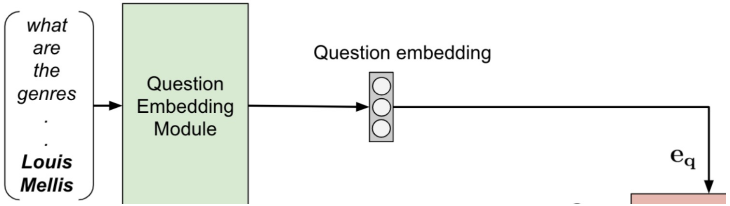


❖ **Find Entity Embeddings by KG Embedding**

- ComplEx used as the KG embedding module
- Trained for all $h, t \in \varepsilon$ and all $r \in R$ in the KG such that $e_h, e_r, e_a \in \mathbb{C}^d$
- The entity embeddings are used for learning a triple scoring function between the head entity, question, and answer entity

EmbedKGQA

· Question embedding



❖ Make Question Embedding

- Using RoBERTa, a variant of BERT, for the given question
- Get a **sentence embedding** that captures the meaning of the question
- Passing through 4 fully connected layers with ReLU activation and then projecting to the complex space C^d
- The dimension matches that of the entity

➢ $e^h, e^q, e^a \in C^d$

EmbedKGQA

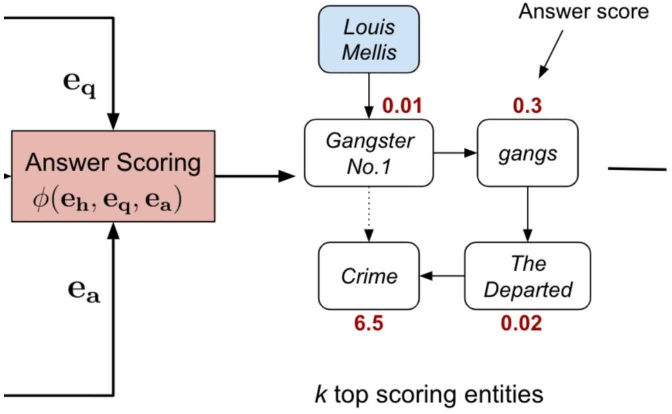
· Scoring function

❖ ComplEx Scoring Function

$$\phi(e_h, e_q, e_a) > 0 \quad \forall a \in \mathcal{A}$$

$$\phi(e_h, e_q, e_{\bar{a}}) < 0 \quad \forall \bar{a} \notin \mathcal{A}$$

➢ question q, topic entity $h \in \varepsilon$ and set of answer entities $A \subseteq \varepsilon$



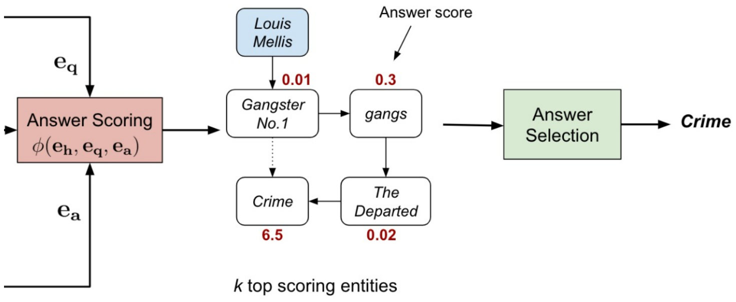
- For each question, the score φ(.) is calculated with all the candidate answer entities

- Learned by minimizing the **binary cross-entropy loss** between the sigmoid of the scores and the target labels

➢ the target label is 1 for the correct answers and 0 otherwise

EmbedKGQA

· Answer selection module



❖ **Select the entity as the answer**

$$e_{ans} = \underset{a' \in \mathcal{E}}{\arg\max}\, \phi(e_h, e_q, e_{a'})$$

- In inference level,

  simply select entity with highest score in relatively smaller KGs

- if the knowledge graph is large, pruning the candidate entities can significantly improve the performance of EmbedKGQA

EmbedKGQA

· Answer selection module

❖ Select the entity as the answer

$$h_q = \mathrm{RoBERTa}(q')$$
$$S(r,q) = \mathrm{sigmoid}(h_q^T h_r)$$

$$\xrightarrow[S(r,q) > 0.5]{R_a}$$

$$\mathrm{RelScore}_{a'} = |\mathcal{R}_a \cap \mathcal{R}_{a'}|$$

$$e_{ans} = \underset{a' \in \mathcal{N}_h}{\arg\max}\, \phi(e_h, e_q, e_{a'}) + \gamma * \mathrm{RelScore}_{a'}$$

- Scoring between question and relations
- Select those relations which have **score greater than 0.5** It is denoted
- Relations as the set R_a

- $R_{a'}$ means **set of relations** in the shortest path between head entity h and **candidate entity a'**
- Relation score for each candidate answer entity is defined as the size of their intersection

Experiment

· Dataset

	Train	Dev	Test
MetaQA 1-hop	96,106	9,992	9.947
MetaQA 2-hop	118,948	14,872	14,872
MetaQA 3-hop	114,196	14,274	14,274
WebQSP	2,998	100	1,639

Table 1: Statistics for MetaQA and WebQuestionsSP datasets. Please refer section 5.1 for more details.

Statistic: # of Questions

MetaQA

- large scale multi-hop KGQA dataset with more than 400k questions in the movie domain
- has 1-hop, 2-hop, and 3-hop questions

WebQSP

- small QA dataset with 4,737 questions
- questions in this dataset are 1-hop and 2-hop
- Lesser questions in datasets

Experiment

· KGQA results

Randomly drop fact with p = 0.5

Model	MetaQA KG-Full			MetaQA KG-50		
	1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
VRN	**97.5**	89.9	62.5	-	-	-
GraftNet	97.0	94.8	77.7	64.0 (91.5)	52.6 (69.5)	59.2 (66.4)
PullNet	97.0	**99.9**	91.4	65.1 (92.4)	52.1 (90.4)	59.7 (85.2)
KV-Mem	96.2	82.7	48.9	63.6 (75.7)	41.8 (48.4)	37.6 (35.2)
EmbedKGQA (Ours)	**97.5**	98.8	**94.8**	**83.9**	**91.8**	**70.3**

With text corpus

metric: hits@1 (Is the model's answer same with real answer?)

- EmbedKGQA can outperform the state-of-the-art for 1-hop, 2-hop, 3-hop **in full dataset**
- In MetaQA KG-50 (when 50% of the triples are removed), graph becomes very sparse with an **average of only 1.66 links per entity node**
- Without text corpus, EmbedKGQA achieves state-of-the-art performance

Experiment

· KGQA results

Model	WebQSP KG-Full	WebQSP KG-50
KV-Mem	46.7	32.7 (31.6)
GraftNet	66.4	48.2 (49.7)
PullNet	**68.1**	50.1 (51.9)
EmbedKGQA	66.6	**53.2**

With text corpus

metric: hits@1 (Is the model's answer same with real answer?)

- EmbedKGQA can outperform the state-of-the-art for 2-hop **in incomplete dataset**
- Even with a small number of training examples,
  EmbedKGQA can learn good question embeddings
- They suppose KG embedding captured relevant and necessary information

Experiment

· Effect of answer selection module

Relation matching

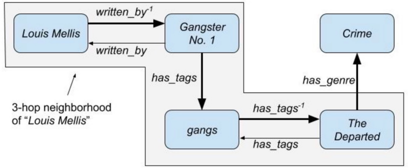
$$h_q = \text{RoBERTa}(q')$$
$$S(r, q) = \text{sigmoid}(h_q^T h_r)$$

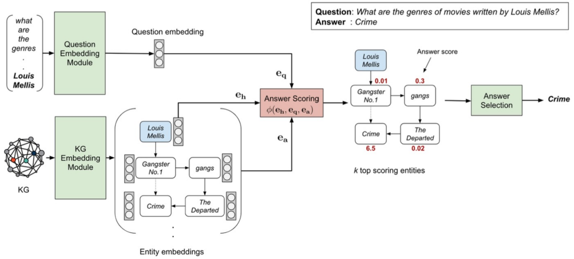
Model	**WebQSP KG-Full**	**WebQSP KG-50**
EmbedKGQA	66.6	53.2
{+ 2-hop filtering}	72.5	51.8
{+ 2-hop filtering, – Relation matching}	58.7	48.5
{– Relation matching}	48.1	47.4

- restricting the candidate set of answer entities to only the 2-hop neighborhood of the head entity
- Ablating the relation matching module to check effect of answer selection module

- In incomplete KG (KG-50), 2-hop neighborhood restriction causes degradation in performance
- Relation matching has a significant impact on the performance of EmbedKGQA on both WebQSP KG-full and WebQSP KG-50 settings

Conclusion





Previous work

There was some approaches to cover incomplete KG while answering about question

- using text corpus is not always available
- imposing neighborhood limits might cause the answer place out of range

EmbedKGQA

Using KG embedding while using ComplEx, and question embedding,

We can infer answer entity with scoring candidate answer, without neighborhood limitation and using text corpus

Experiments

Compare to existing models with limitation of using text corpus,

EmbedKGQA shows state-of-the-art performance in KGQA

However, when the training dataset is small (WebQsp), the model suffer to perform well

Joint Knowledge Graph Completion and Question Answering

Published as a conference paper at KDD 2022

Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, Hanghang Tong

2024-11-28

HoonUi Lee

Index

Previous work

KGQA and KGC

BiNet

Experiment

Conclusion

Previous work

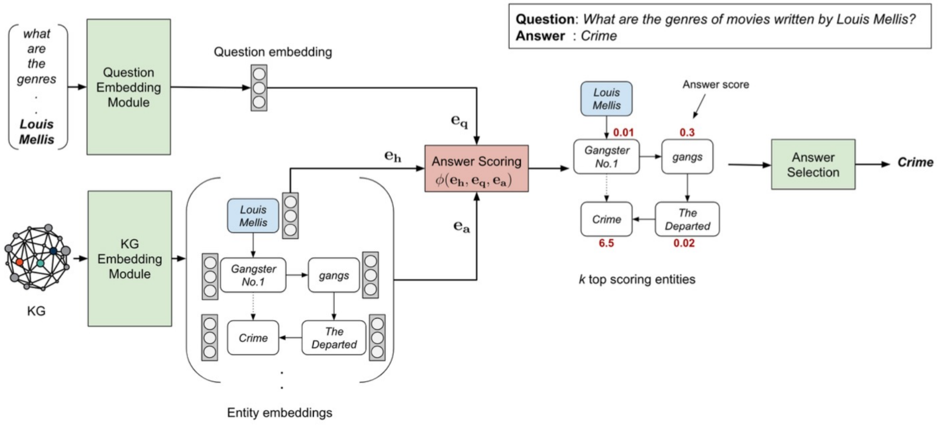
· EmbedKGQA

◆ KGC and KGQA have interchangeable properties

- KGC task treated as single-hop KGQA

➢ (Interstellar, hasGenre, ?) ⇔

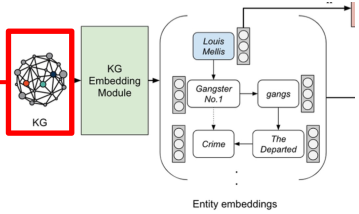
➢ "What is the genre of Interstellar?"



Overview of EmbedKGQA

Previous work

Is the KG always complete?



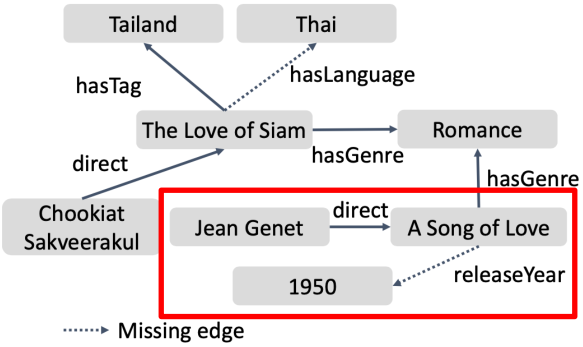
◆ **But treat them as two separate tasks**

- Most existing Multi-hop KGQA methods have implicitly assumed the background knowledge graph is complete

- Existing KGC methods only exploit the existing information of the input incomplete KGs

➔ **Multi-hop KGQA and KGC can help each other!**

KGQA and KGC

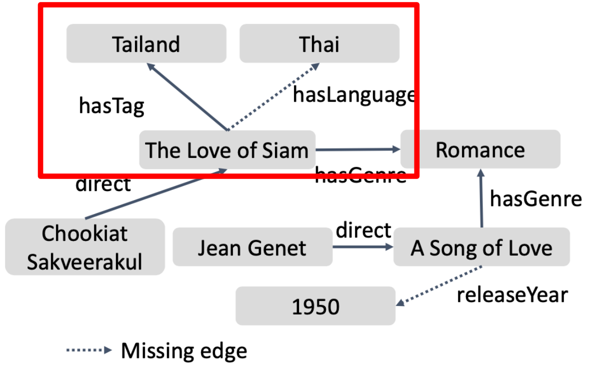


◆ KGQA helps KGC

- New knowledge can be inferred from the KGQA task
  - ➢ Want to answer *"which year was A song of Love released?"*
  - ➢ Can not answer only based on the existing knowledge graph

- Another question and answer can be used as new knowledge for KGC
  - ➢ Q: *"which years were all the films directed by Jean Genet released?"* A: 1950
  - ➢ we can infer that the release year of A song of Love is 1950
  - ➢ Using the hint that Jean Genet only directed one film in his life **in Question Context**

KGQA and KGC



◆ KGC helps KGQA

- KGC could help improve the performance of KGQA by providing a KG
  - ➢ The movie *The Love of Siam* is linked to Tailand via the hasTag relation
  - ➢ Ideal KGC model can infer that the movie might be intended for the Thai audiences

- "what is the language of the film The Love of Siam" can then be trivially answered
  - ➢ KGQA is provided more complete knowledge triples of high quality from KGC's inference

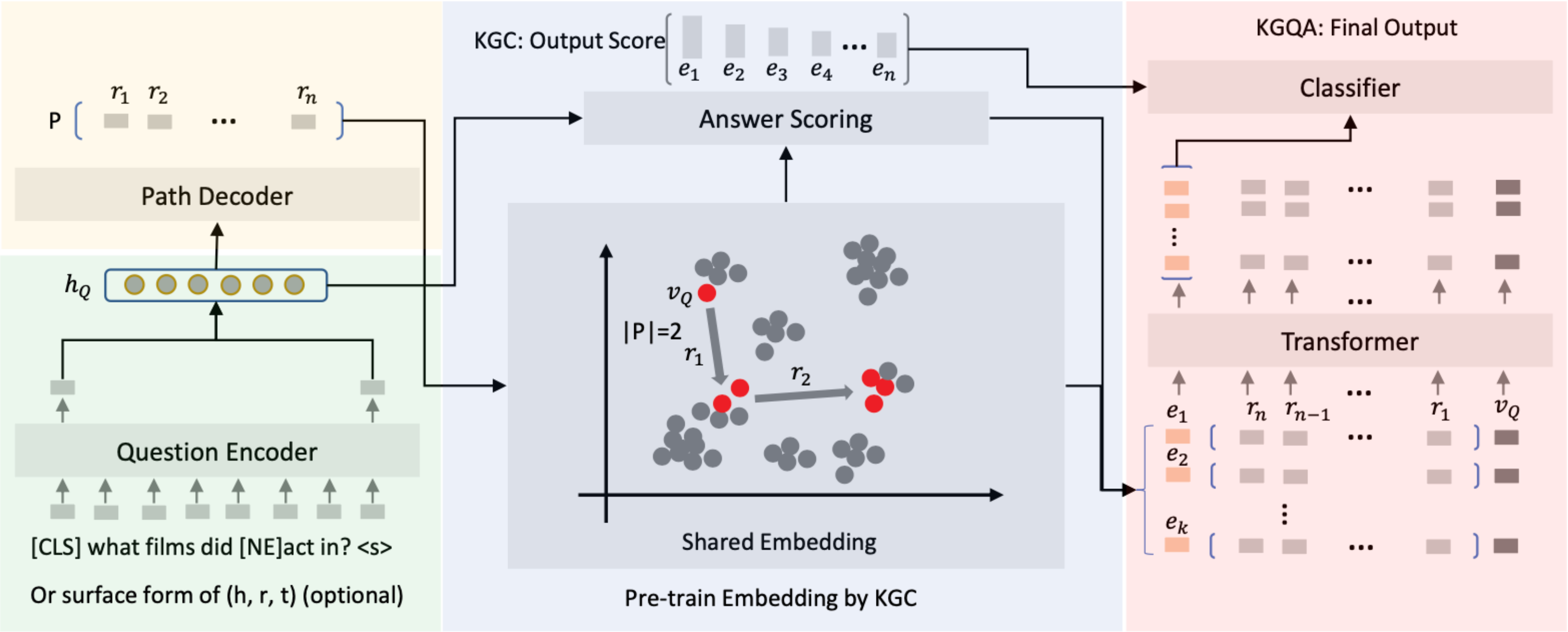
BiNet

◆ **Jointly address multi-hop KGQA and KGC tasks as multi-task learning problem**

- Encoder-decoder-based model which transforms natural language questions into relation paths
  - In order to leverage multi-hop KGQA for the KGC task
- Multi-hop KGQA and KGC share both the embedding space and the answer scoring module
  - In order to leverage KGC for multi-hop KGQA
  - Automatically share latent features and reinforce each other

BiNet

· Overview



BiNet

· Preprocessing Text

◆ Pass the question context through pre-trained BERT

$$Q = (w_1, w_2, \dots, w_{|Q|}) \qquad v_Q \in \mathcal{V} \quad A_Q \subseteq \mathcal{V}$$

- Decodes a sequence of relations between the topic entity V_Q and an answer set A_Q in a natural language sentence Q
- Each question context Q could be mapped to a relation path in the KG distinctively

BiNet

· Preprocessing Text

◆ A special token [NE]

- Mask the topic entity inside the question context/surface form
- To mitigate the noise brought by the surface forms of the entities

➢ "Who starred Interstellar?" ➔ "Who starred [NE]?"

- It helps model generalize to similar questions involving other entities

BiNet

· Question Encoder

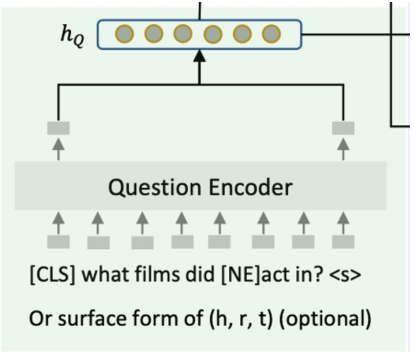
◆ **Pass the question context through pre-trained BERT**

$$[\mathbf{h}_{CLS}, \mathbf{w}_1, \ldots, \mathbf{w}_{|Q|}, \mathbf{h}_s] = \mathrm{BERT}([CLS], w_1, \ldots, w_{|Q|}, < s >)$$

✓ $\mathrm{h}_{\mathrm{CLS}}$ is the embedding of the [CLS] token and h_{S} is the embedding of the <s> token

$$\mathbf{h}_Q = \mathrm{FFN}([\mathbf{h}_{CLS}|\mathbf{h}_s])$$

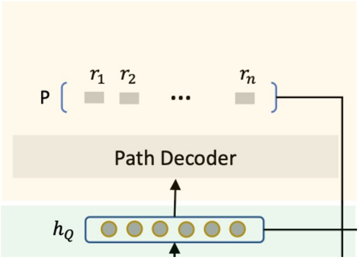
✓ FFN is a feed forward neural network, and | indicates concatenation



▪ Final question embedding is obtained from the combination $\mathrm{h}_{\mathrm{CLS}}$ of and h_{S}

BiNet

· Question Decoder



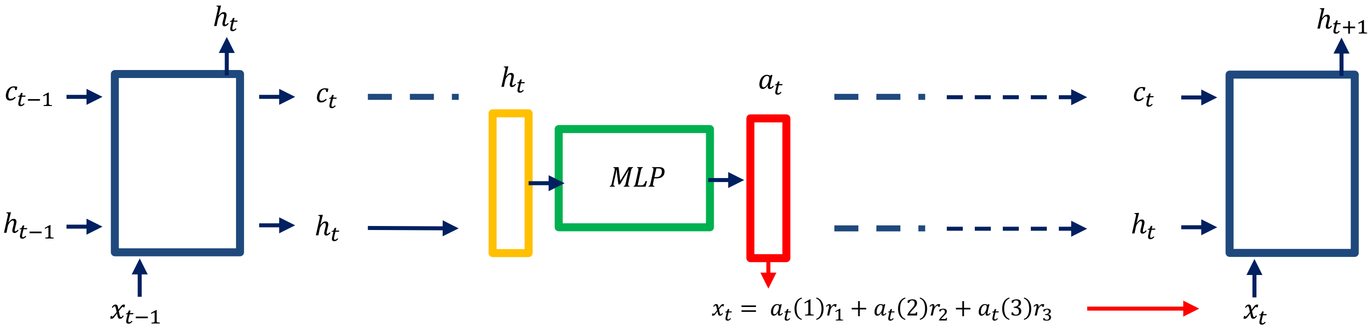
◆ **Generate generates a sequence of relations using LSTM**

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{o}_{t-1}) \qquad \mathbf{h}_0 = \mathrm{FFN}_h(\mathbf{h}_Q) \quad \mathbf{c}_0 = \mathrm{FFN}_c(\mathbf{h}_Q)$$

$$\mathbf{a}_t = \mathrm{softmax}(\mathrm{MLP}(\mathbf{h}_t))$$

✓ h_0, c_0 are obtained from question embedding h_Q passing it through two feed forward neural networks separately

✓ initial input embedding x_0 could be the question embedding h_Q or a zero vector



BiNet

· Training Question Encoder - Decoder

◆ Map the question context to its correct relation path

- To identify the correct path, find all the k-shortest paths between each entity pair (v_Q, v_i)

✓ $v_i \in A_Q$, A_Q is candidate entity set of Question

- Treat all these shortest paths as potentially correct path candidates

- Use Bayes' Rule to infer the probability of
  whether the shortest path is the correct mapping of the question context

BiNet

· Training Question Encoder - Decoder

- ✓ $PL_i = \{A_{Q_j} | Q_j \in S_i\}$: corresponding answer pool
- ✓ $S_1, S_2, .., S_m$: Divided question set after masking topic entity
- ✓ Each answer entity $v_i \in A_{Q_j}$ has corresponding candidate path set

$$PC(Q_j, v_i) = \{P_i | (v_{Q_j}, P_i, v_i)$$

◆ Bayes' Rule

S_j ➔ Who starred [NE]?
Q_{j1} → Who starred Interstellar?
Q_{j2} → Who starred Avengers?
Q_{j3} → Who starred Maze Runner?

$$Pr(P_i | S_j, \theta) = \frac{\sum_{Q_j \in S_j} \boxed{Pr(P_i, Q_j | \theta)}}{|PL_j|}$$

- The probability that a specific path P_i correctly interprets the question context

$$Pr(P_i, Q_j | \theta) = \frac{\sum_{v_i \in A_{Q_j}} |PC(Q_j, v_i)|^{\mathbb{1}(P_i \in PC(Q_j, v_i))}}{|A_{Q_j}|}$$

✓ $\mathbb{1}()$ is the indicator function

- The probability that a specific path P_i is associated with the question Q_j

BiNet

· Training Question Encoder – Decoder: Loss function

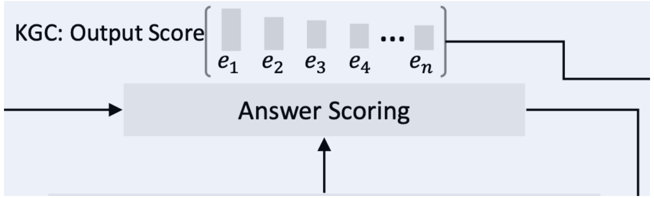
◆ Binary cross-entropy loss function

$$\mathcal{L}(\hat{P}, P) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{|P|} \mathbb{1}(P[j] = r_i) log(Pr(r_i|\mathcal{M}))$$
$$+(1 - \mathbb{1}(P[j] = r_i)) log(1 - Pr(r_i|\mathcal{M}))$$

- Train is performed at a relation level within the path
- Train the model to assign a high probability to the correct relation r_i and a low probability to the incorrect relation r_i

BiNet

· Answer Scoring



◆ Using learning-based methods, scoring function

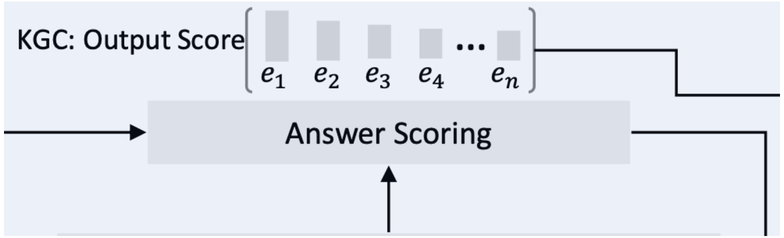
- Existing knowledge graph traversal methods or subgraph matching approaches are likely to fail on incomplete or incorrect knowledge graphs

- TransE, RotatE
  - ➢ noise often exists in the embedding space
  - ➢ increase of the path length, the cascading error will become larger

Path embedding $\mathbf{p} = \sum_{r_i \in P} \mathbf{r}_i$: TransE $\mathbf{p} = \mathbf{r}_1 \odot \mathbf{r}_2 \odot \cdots \odot \mathbf{r}_n$: RotatE

BiNet

· Answer Scoring



◆ Probabilistic Reasoning Model

$$Pr(v|P, v_Q, \mathcal{G}) \propto \prod_{i=1}^{|P|} \Theta(r_i, v_i | P_{1\rightarrow i-1}, v_Q, \mathcal{G})$$

- Considering a relation sequence $P = (r_1, \ldots, r_{|P|})$ originated from topic entity v_Q to A_Q's node
- Compute the likelihood of v by multiplying the likelihood of all intermediate steps traversed by P

➔ Finding the best answer is equivalent to maximizing the probability function

BiNet

· Answer Scoring

◆ **Consider incompleteness of the KG**

- Iterating all the intermediate candidates can find the correct answer with a high probability

➔ It could **hamper the efficiency**

- Each step, we select the top-k candidates with maximum likelihood

➢ Using efficient search algorithm, such as beam search

➢ Strike a good balance between effectiveness and efficiency, mitigate the cascading error

- In the last step, we choose the candidate with the highest probability

$$o = \max_{v_i \in \mathcal{V}}(Pr(v_i|P, v_Q, \mathcal{G}))$$

BiNet

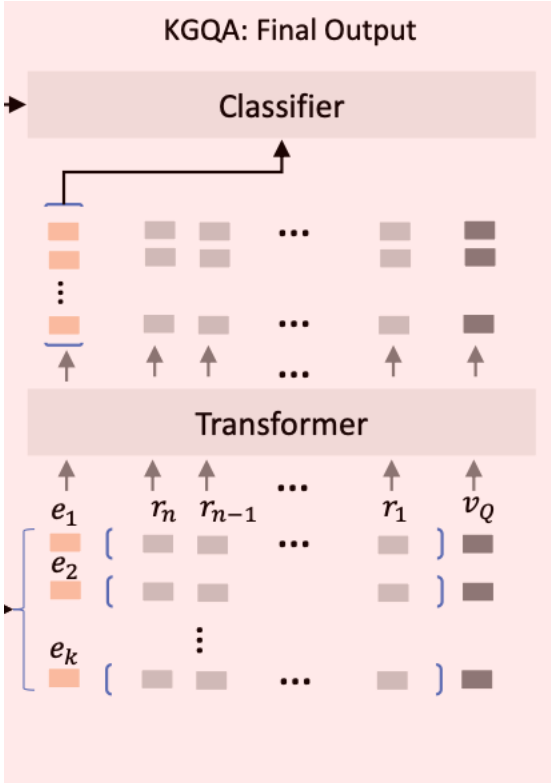
· Answer Refinement

◆ Refine noise in the candidate answer set

- Noise may has a higher probability than true answers

- The incompleteness and complexity of the KG

➢ It is almost impossible to find perfect model which can satisfy all the properties and find answers without errors



BiNet

· Answer Refinement

◆ Re-order the top-k candidates of the answer scoring module

$$h_i = \text{TRANSFORMER}([\mathbf{e}_{v_Q} | \mathbf{r}_1 | ... | \mathbf{r}_n | \mathbf{e}_{v_i}])$$

- given a topic entity v_Q and a path P, we concatenate them with each of the candidates to get k sequences
- h_i is the output embedding of entity v_i

$$Pr(v_i | P, v_Q, \mathcal{G}) = \text{Sigmoid}(\text{FFN}(h_i))$$

- The final score is predicted by passing h_i through a feed forward neural network with Sigmoid function

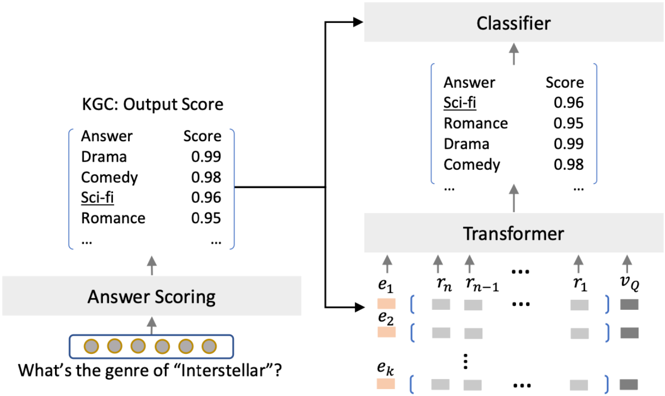


Figure 3: Answer Refinement.

BiNet

· Answer Refinement

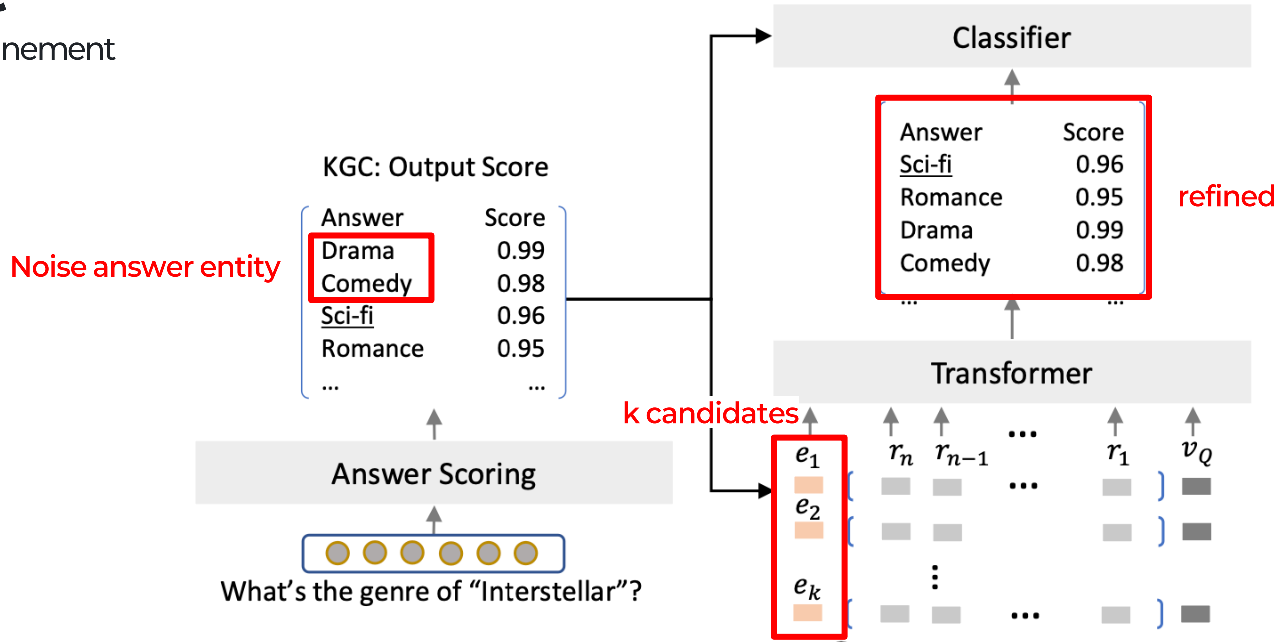


Figure 3: Answer Refinement.

- In the last step, a classifier will be used to return the answer predicted by the answer scoring module or the transformer module (Optional, ablation study in Experiment)

BiNet

· Overall loss function

$$\mathcal{L} = \mathcal{L}_{KGQA} + \mathcal{L}_{KGC} + \mathcal{L}_{Path} + \mathcal{L}_{REG}$$
$$= \sum_{Q \in \overline{Q}} \mathcal{J}(\hat{y}, y) + \lambda_1 \sum_{(h,r,t) \in \mathcal{G}} \mathcal{J}(\hat{t}, t) + \lambda_2 \sum_{Q \in \overline{Q}} \mathcal{L}(\hat{P}, P) + \lambda_3 ||\mathbf{W}||_2^2$$

- $\mathcal{L}_{KGQA}$: Trains the model to generate correct answers for the question answering task
- $\mathcal{L}_{KGC}$: Trains the model to learn the correct entities for the KGC task
- $\mathcal{L}_{Path}$: Trains the path decoder to improve its ability to predict paths include measuring the effectiveness of the Answer Refinement process
- $\mathcal{L}_{REG}$: Applies regularization to prevent the model from overfitting

BiNet

· Overall loss function

$$\mathcal{L} = \mathcal{L}_{KGQA} + \mathcal{L}_{KGC} + \mathcal{L}_{Path} + \mathcal{L}_{REG}$$
$$= \sum_{Q \in \overline{Q}} \mathcal{J}(\hat{y}, y) + \lambda_1 \sum_{(h,r,t) \in \mathcal{G}} \mathcal{J}(\hat{t}, t) + \lambda_2 \sum_{Q \in \overline{Q}} \mathcal{L}(\hat{P}, P) + \lambda_3 ||\mathbf{W}||_2^2$$

- Composed as **Multi-task Learning**
- Each loss component works complementarily to help BiNet optimize performance in both Knowledge Graph Question Answering and Knowledge Graph Completion

Experiment

· Dataset

Table 8: Summary of datasets. Coverage is the accuracy of subgraph matching. As we can see, simply applying edge traverse on the complete knowledge graph could achieve nearly 100% accuracy.

Dataset	Train	Valid	Test	Coverage
MetaQA 1-hop	96,106	9,992	9,947	100%
MetaQA 2-hop	118,948	14,872	14,872	100%
MetaQA 3-hop	114,196	14,274	14,274	99%
WebQSP	2,950	-	1,560	99%
SimpleQA	15,3188	2,105	4,345	99%

MetaQA

- large scale multi-hop KGQA dataset with more than 400k questions in the movie domain
- has 1-hop, 2-hop, and 3-hop questions

WebQSP

- small QA dataset with 4,737 questions
- questions in this dataset are 1-hop and 2-hop
- Lesser questions in datasets

SimpleQuestions

- 100,000 1-hop questions with corresponding triplets in FB

Experiment

· KGQA results

Randomly drop fact with p = 0.5

Randomly drop fact with p = 0.3

Table 2: KGQA Hits@1 results of MetaQA on 50% and 30% incomplete knowledge graphs.

	50% KG				30% KG			
Model	MetaQA-1	MetaQA-2	MetaQA-3	Avg	MetaQA-1	MetaQA-2	MetaQA-3	Avg
GraftNet	64.0	52.6	59.2	58.6	48.4			48.4
PullNet	65.1	52.1	59.7	59.0	-	-	-	-
KV-Mem	63.6	41.8	37.6	47.7	44.7			44.7
EmbedKGQA	83.1	91.8	70.3	81.7	77.7	81.2	69.0	76.0
BiNet	84.2	92.8	75.9	84.3	77.8	86.4	74.3	79.5

Subgraph based

metric: hits@1 (Is the model's answer same with real answer?)

- When the background knowledge graph becomes sparse, the Hits@1 accuracy decreases

➔ the quality of the background KG has significant impact on the KGQA task

- While the KG becomes sparse, because of less ability to cover the answer entities, subgraph retrieval-based methods' performance suffer from incomplete KG

- BiNet achieves the best results for all situation

Experiment

· KGQA results

Table 3: KGQA Hits@1 results of WQSP and SimpleQA on 50% and 30% incomplete knowledge graphs.

	50% KG		30% KG	
Model	Webqsp	SimpleQA	Webqsp	SimpleQA
GraftNet	32.7	39.8	34.9	25.7
PullNet	48.2	-	34.6	-
KV-Mem	50.1	28.9	25.8	22.8
EmbedKGQA	47.3	41.7	38.8	33.5
BiNet	49.4	42.6	40.5	33.9

metric: hits@1

- Compare to MetaQA, WebQSP and SimpleQA have less performance than MetaQA

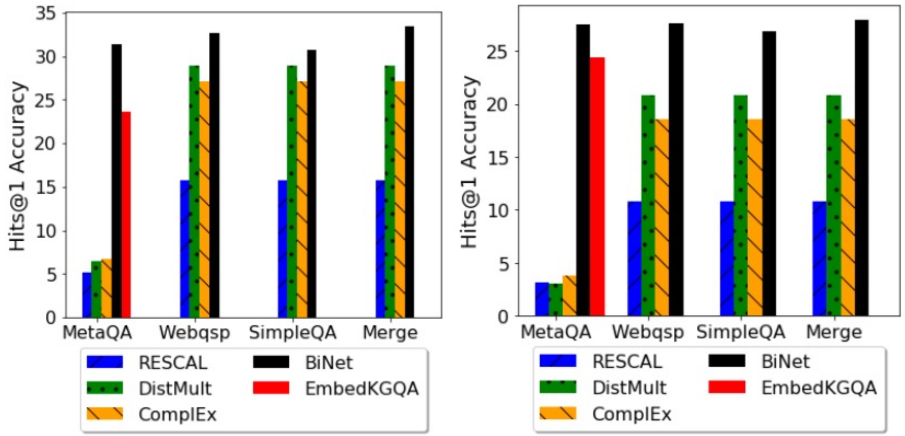
- WebQSP has small training set, SimpleQA has only 1-hop question

➔ Small training set was still insufficient to significantly improve performance

➔ Simple question training set made it difficult for the model to predict answers from incomplete KG

Experiment

· KGC results



(a) Accuracy on 50% Incomplete KG (b) Accuracy on 30% Incomplete KG

- Because of MetaQA's sparsity, traditional KG embedding methods do not perform very well on MetaQA
- For EmbedKGQA, transform KG triple to natural language question and trained EmbedKGQA
- BiNet has the highest performance, that shows that information from the question can help KGC

Experiment

· Ablation studies – Answer Refinement

Table 4: Ablation study of Answer Refinement.

50% KG			
Model	MetaQA-3hop	Webqsp	SimpleQA
BiNet without refinement	70.3	47.2	41.8
BiNet with refinement	75.9	49.4	42.6
30% KG			
Model	MetaQA-3hop	Webqsp	SimpleQA
BiNet without refinement	71.2	39.1	33.2
BiNet with refinement	74.3	40.5	33.9

- Refinement module could improve the prediction accuracy by about 2% on average on both 50% and 30% incomplete knowledge graphs

➔ Refinement model indeed alleviates the sparsity of the background knowledge graph

- The accuracy improvement on long path questions is more significant

➔ When the path becomes longer, the refinement module is even more effective

Experiment

· Ablation studies – Power of KGC

Table 5: The power of knowledge graph completion.

50% KG			
Model	EmbedKGQA	KGC + EmbedKGQA	BiNet
MetaQA-1hop	83.1	83.2	84.2
MetaQA-2hop	91.8	92.4	92.8
MetaQA-3hop	70.3	73.5	75.9
Webqsp	47.3	47.7	49.4
SimpleQA	41.7	41.9	42.6
30% KG			
Model	EmbedKGQA	KGC + EmbedKGQA	BiNet
MetaQA-1hop	77.7	77.8	77.8
MetaQA-2hop	81.2	85.1	86.4
MetaQA-3hop	69.0	71.1	74.3
Webqsp	38.8	39.1	40.5
SimpleQA	33.5	33.7	33.9

- Using ComplEx to predict the answer of (h, r ,?) in BiNet

➔ keep those triples which satisfy $Pr(vt|ri,vh,$G)>= 0.99 where 1 is the highest score

- On average, completing the knowledge graph first could improve about 1.2% Hits@1 accuracy

➔ Completing the knowledge graph first can indeed improve the KGQA performance

Experiment

· Ablation studies – Path Prediction

Table 11: Results of Path Decoder.

Question	Path
the movies starred by [Tanner Maguire] were in which genres	starred_actors_reverse | has_genre
when did the movies written by [Cristian Nemescu] release	written_by_reverse | release_year
the films acted by [Benjamin Pitts] were released in which years	Starred_actors_reverse | release_year
who are movie co-writers of [Ray Ashley]	written_by_reverse | written_by
who co-starred with [Mary McDonnell]	starred_actors_reverse | starred_actors
who are movie co-directors of [Jack Hazan]	directed_by_reverse | directed_by

- Before training the path decoder, add ground-truth paths to the training data
- With probability α (α = 0.5), the decoder uses the actual ground-truth relation as the input to the decoder during the next time-step
- With probability 1 −α, it uses the relation that the model predicts as the next input to the model, even if it does not match the actual next relation in the ground-truth

Conclusion

The existing KGQA models that **extracting subgraph** do not perform well in incomplete KG while not using text corpus
EmbedKGQA do not leverage the complementary nature of **Knowledge Graph Completion (KGC) and Knowledge Graph Question Answering (KGQA)**

Proposed model BiNet jointly address multi-hop KGQA and KGC tasks as **multi-task learning problem**
KGQA's question embedding helps KGC by serving additional information,
KGC helps KGQA while KGQA **use KG completed by KGC**

Experiment shows state-of-the-art performance in KGQA and KGC task
and **Answer refining, KGQA with KGC task** improve performance of KGQA task while using BiNet

However, when the training dataset is small or the questions in the given dataset are simple (1-hop),
the model still suffer to perform well.