# KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning

**Bill Yuchen Lin**, Xinyue Chen, Jamin Chen, Xiang Ren

University of California

EMNLP 2019

Presented by Koo hyeongseok
gudtjrdltka@korea.ac.kr
2021.01.08

How to empower machines with performing commonsense reasoning?

(Specifically, commonsenseQA task)

Where do adults use glue sticks?

A: classroom B: office C: desk drawer

# Previous work in commonsense reasoning

Using simply fine-tuning large, pre-trained language models

Such as GPT[1], BERT[2]

=> 1) *Much lower* than human performance

=> 2) Not providing *reusable* structures for *explainable* commonsense reasoning

[1] Radford et al., "Improving language understanding by generative pre-training", OpenAI technical report
[2] Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding", NAACL-HLT 2019

# Motivation

How about ***injecting external knowledge***?

[Entity/Event extraction tasks] – Keeping related entities form knowledge bases[1]

[Natural language understanding tasks] – Incorporating related knowledge triples at the word-level[2]

In this work, imposing graph-structured knowledge into models

[1] Bishan Yang and Tom Michael Mitchell.2017. Leveraging knowledge bases in lstms for improving machine reading. In Proc. of ACL.
[2] Dirk Weissenborn..2017. Dynamic integration of background knowledge in neural nlu systems. arXiv preprint arXiv:1706.02596.

# Contribution

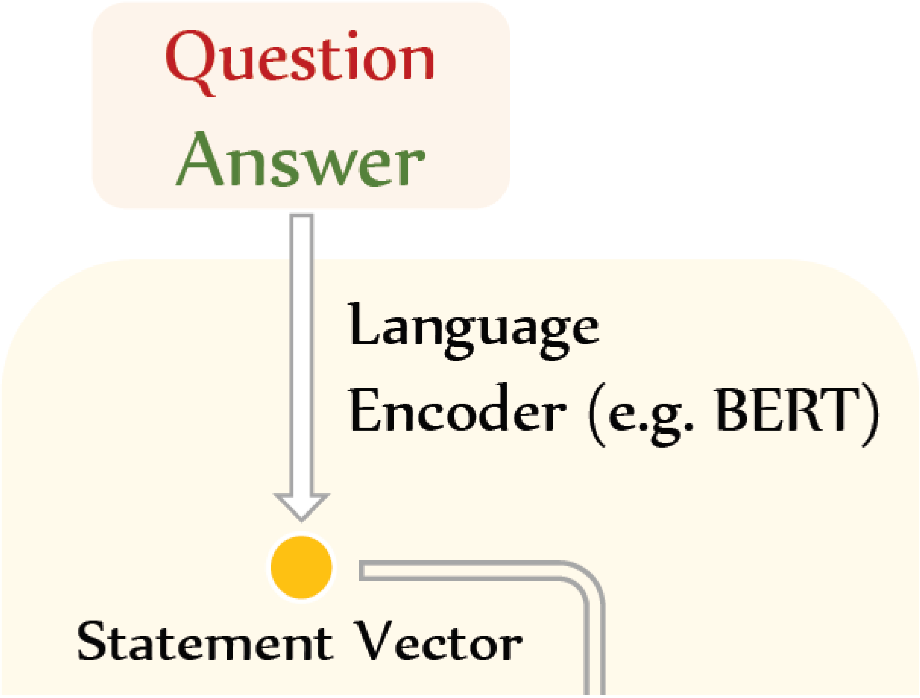
1) They effectively ***utilize external commonsense knowledge graphs*** in commonsense reasoning

2) They propose a ***knowledge-aware graph network(KagNet) module***, which model the graphs and perform explainable inferences

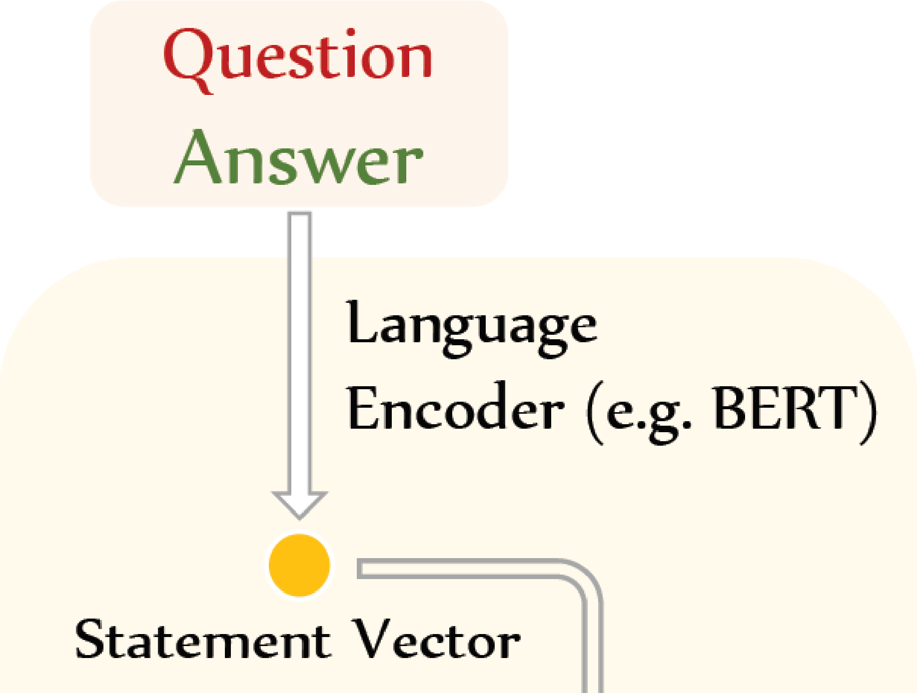
3) They yield a new ***state-of-the-art results on commonsenseQA task***

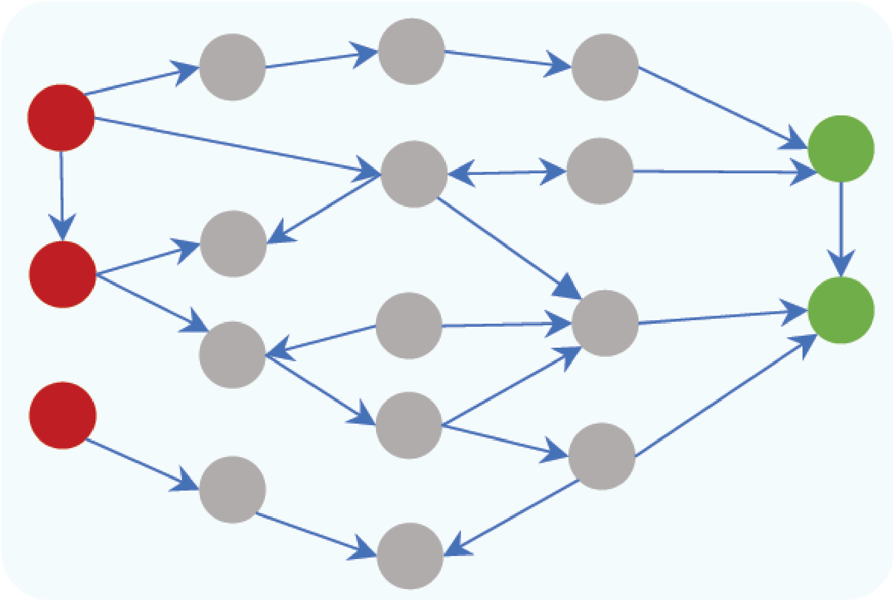
# Model

# Model: Overview



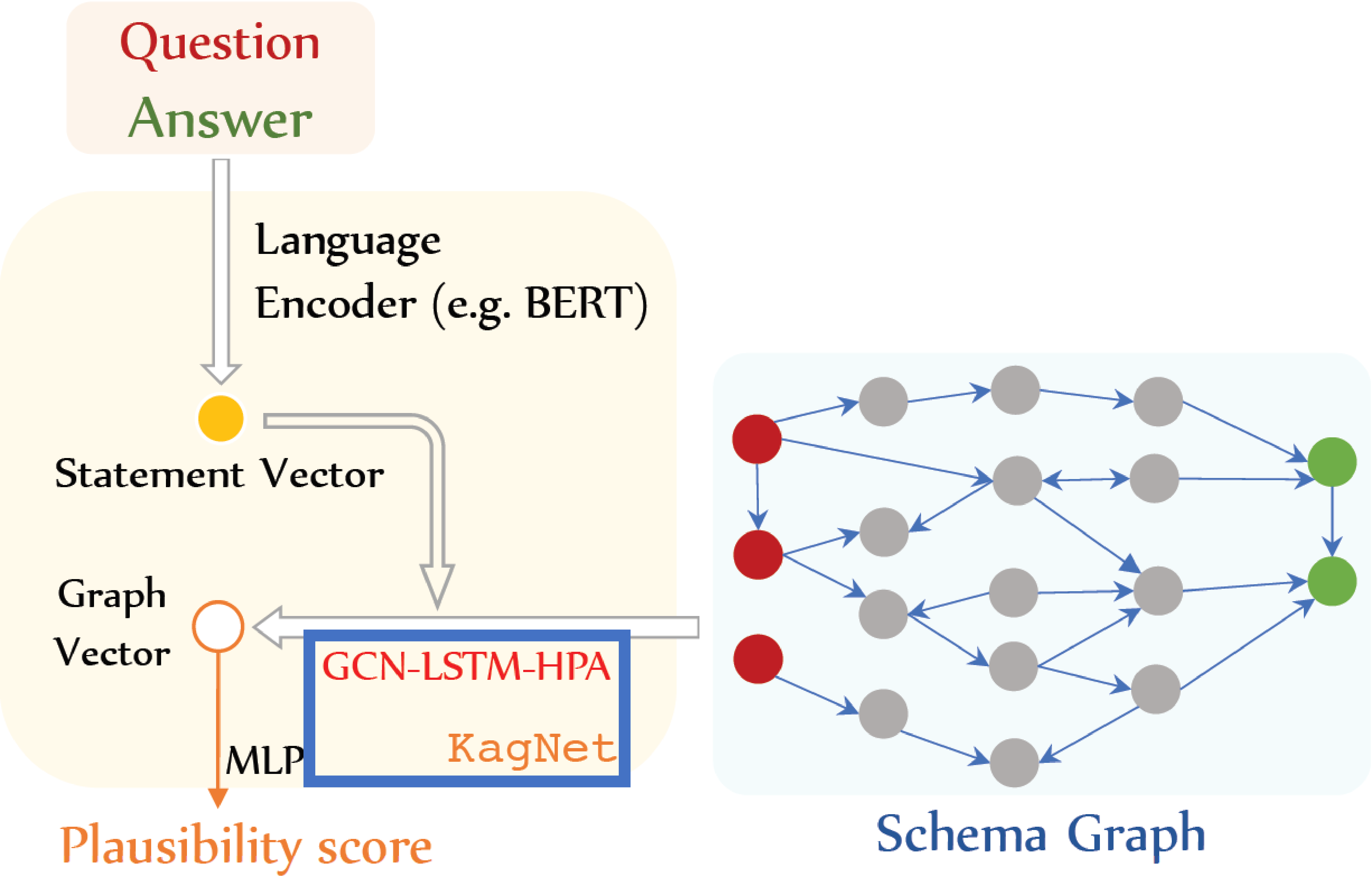
# Model: Overview



Schema Graph

# Model: Overview



## Model: Schema graph grounding

Question

Answer

Concept Recognition

$\mathcal{C}_q$

Question

Concepts

$\mathcal{C}_a$

Answer

Concepts

<u>Example</u>

“Sitting too close to watch tv can cause what sort of pain?”

=> {sitting, close, watch_tv, watch, tv, sort, pain, etc.}

<u>Match</u>

Tokens in questions and answers

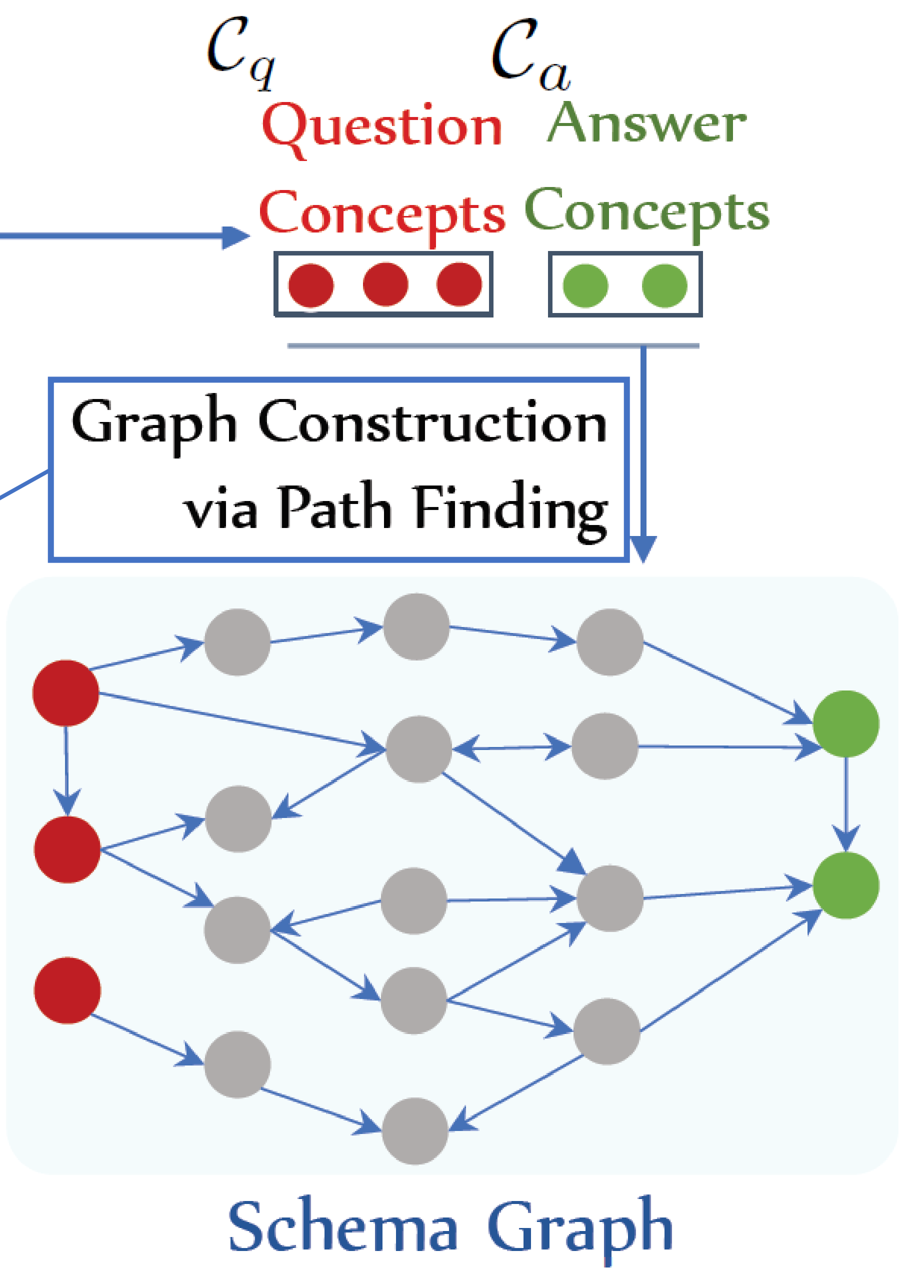
&

Mentioned concepts from the knowledge graph (specifically, ConceptNet)

# Model: Schema graph grounding

Question

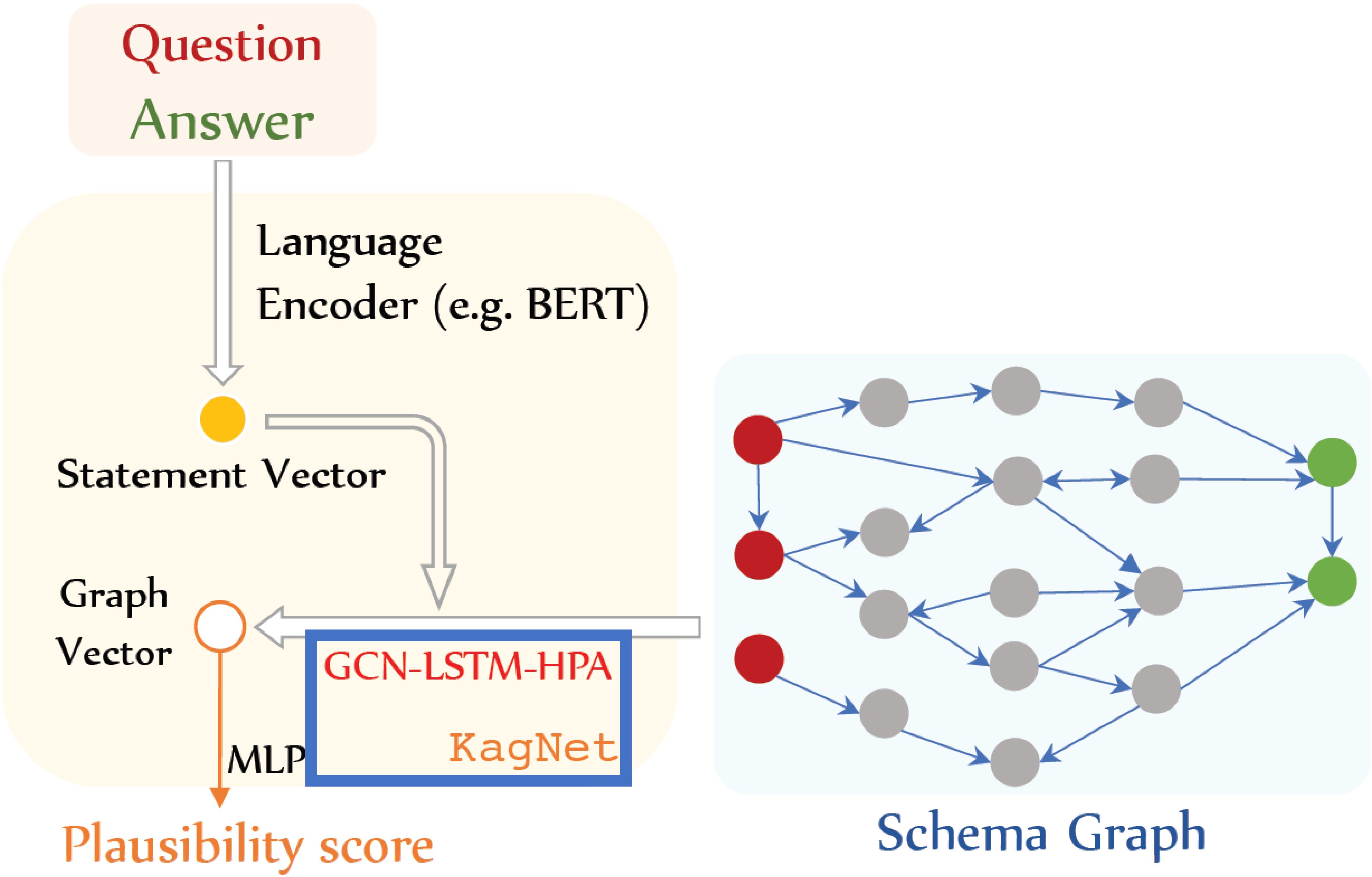
Answer




1)
<u>Find all paths</u> between question concepts and answer concepts, shorter than k concepts

2)
<u>Path pruning</u>
By utilizing knowledge graph embedding(KGE) techniques

# Model: GCN-LSTM-HPA



# Model: GCN(Graph Convolutional Networks)

Why applying GCN to Schema Graph?

<u>Updating node vectors via pooling features of their adjacent nodes</u>

=> 1) Contextually refine the concept vectors

=> 2) Capturing structural patterns of schema graphs

<u>Example</u>

"Close the door" – verb concept

Close

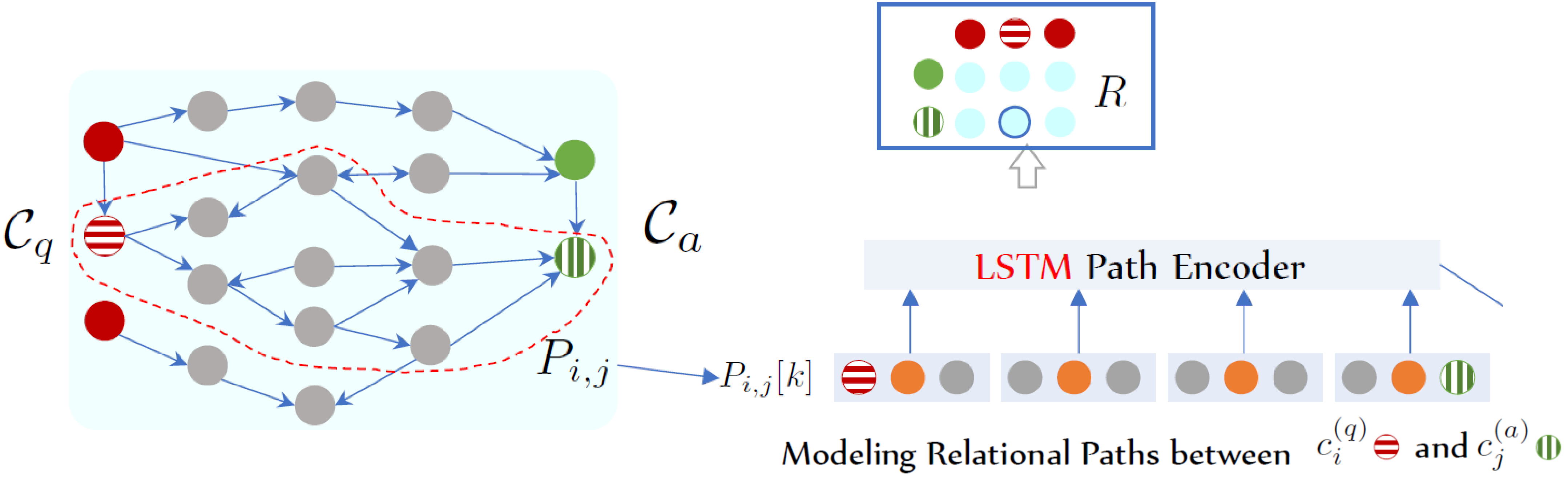
"A short distance apart" – adjective concept

* Applying GCNs on the plain version of schema graphs, ignoring relation types on the edges.

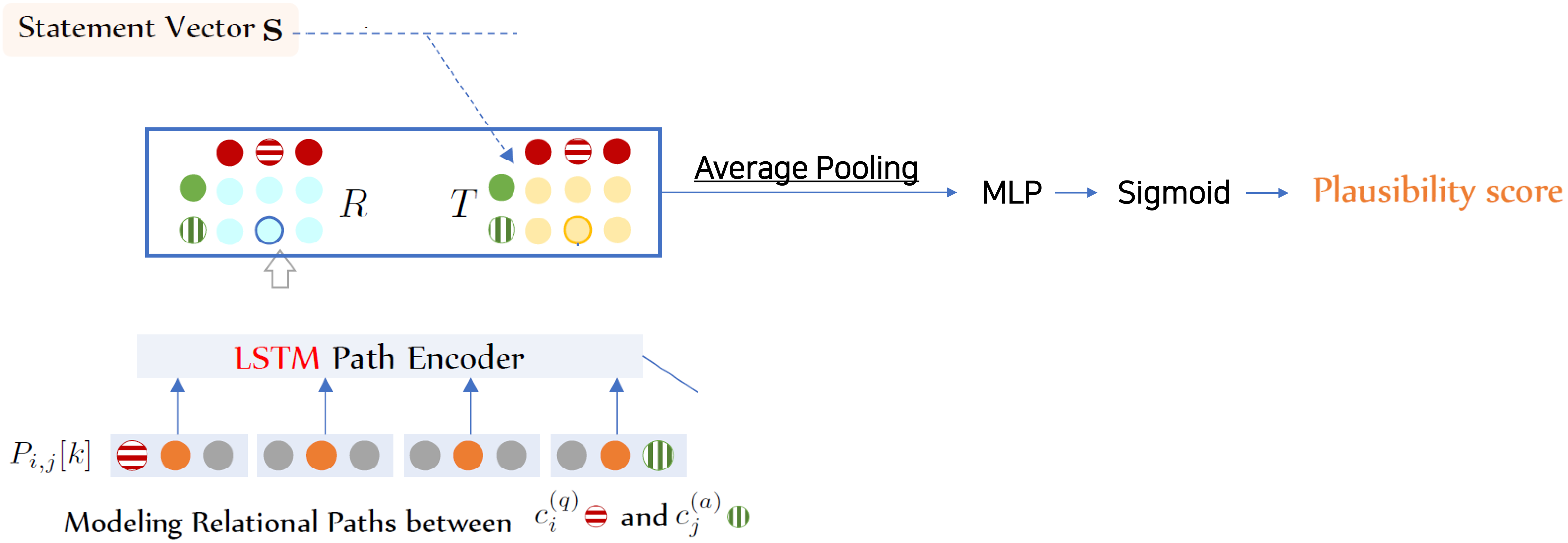
# Model: LSTM-based path encoder

Final Goal: "Measuring the plausibility of a candidate answer to a given question"

By applying LSTM-based path encoder,
<u>Representing schema graphs with respect to the paths between question concepts and answer concepts</u>



# Model: LSTM-based path encoder



# Model: HPA(Hierarchical path-based Attention Mechanism)

Average Pooling => Hierarchical path-based Attention Mechanism

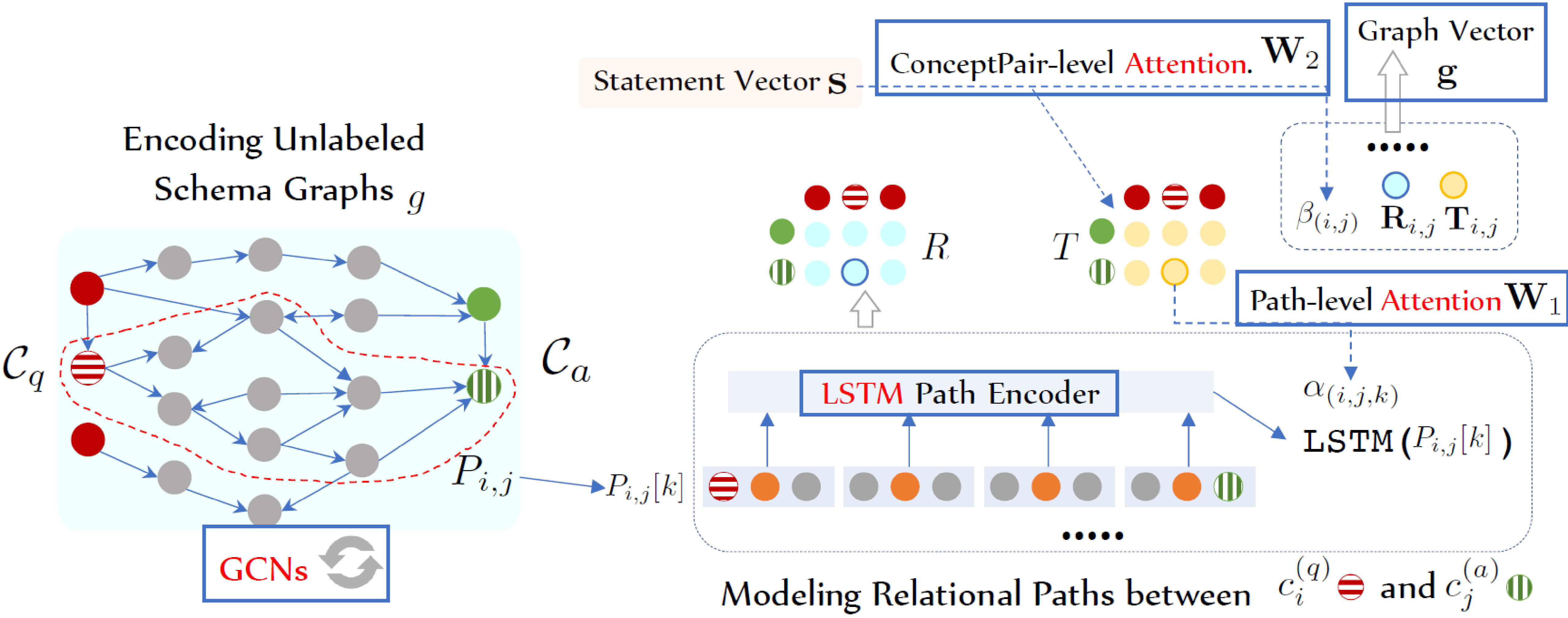
Two-levels attention mechanisms

1) Not all paths are equally important: *Path-level attention scores*

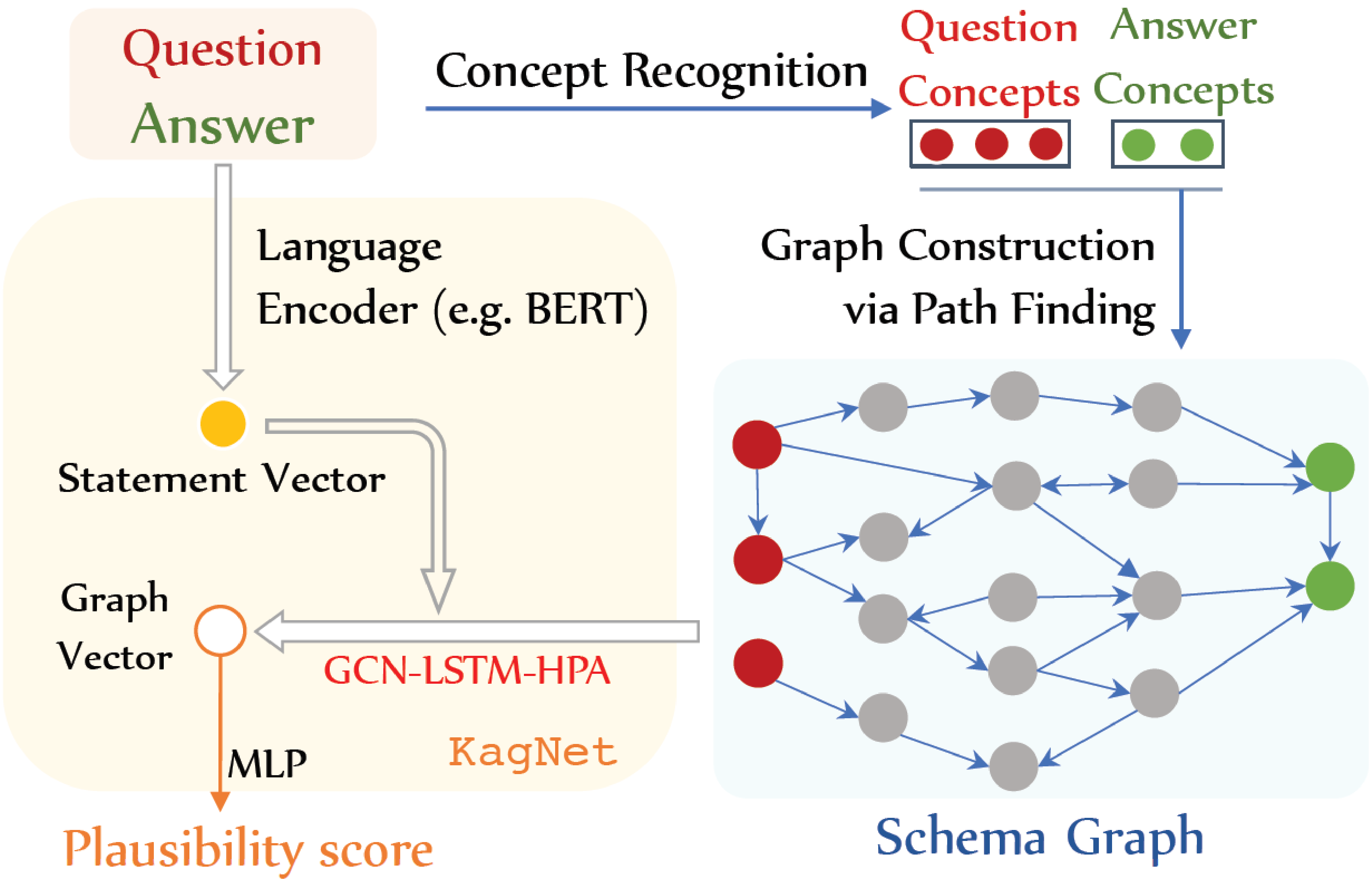
2) Not all question-answer concept pairs equally important: *Concept-pair-level attention scores*

~~Average Pooling~~ Hierarchical path-based Attention Mechanism → MLP → Sigmoid → Plausibility score

# Model: GCN-LSTM-HPA



## Model: KagNet(Knowledge-Aware Graph Networks)



# Q & A about model

# Dataset and code

Dataset(CommonsenseQA)
: https://www.tau-nlp.org/commonsenseqa
⇒ Official split: OFtrain/OFdev/OFtest
⇒ In-house split: IHtrain/ IHdev/ IHtest

Code
: https://github.com/INK-USC/KagNet

# Parameter settings

GCN: 2 Layers (100 dim, 50 dim)

LSTM: 1 bidirectional (128dim)

Pre-train KGE: TransE (100dim)(Initialized with GloVe embeddings)

Statement encoder: BERT-Large

Optimizer: Adam

Path-score threshold: 0.15

# Experiments

## Comparison with standard baselines

| Model | OFdev-Acc.(%) | OFtest-Acc.(%) |
|---|---|---|
| Random guess | 20.0 | 20.0 |
| BIDAF++ | - | 32.0 |
| QACompare+GloVe | - | 25.7 |
| QABilinear+GloVe | - | 31.5 |
| ESIM+ELMo | - | 32.8 |
| ESIM+GloVe | - | 34.1 |
| GPT-FineTuning | 47.11 | 45.5 |
| BERT-Base-FineTuning | 53.57 | 53.0 |
| BERT-Large-FineTuning | 62.34 | 56.7 |
| CoS-E (w/ additional annotations) | - | 58.2 |
| **KagNet** (Ours) | 64.46 | **58.9** |
| Human Performance | - | *88.9* |

Knowledge-agnostic Methods
(no external resources)

<u>Increment of 2.2%</u>

# Experiments

## Comparison with knowledge-aware baselines

Bidirectional LSTM

Utilizing external knowledge

| Model | Easy Mode IHdev.(%) | Easy Mode IHtest.(%) | Hard Mode IHdev.(%) | Hard Mode IHtest.(%) |
|---|---|---|---|---|
| Random guess | 33.3 | 33.3 | 20.0 | 20.0 |
| BLSTMs | 80.15 | 78.01 | 34.79 | 32.12 |
| + KV-MN | 81.71 | 79.63 | 35.70 | 33.43 |
| + CSPT | 81.79 | 80.01 | 35.31 | 33.61 |
| + TextGraphCat | 82.68 | 81.03 | 34.72 | 33.15 |
| + TripleString | 79.11 | 76.02 | 33.19 | 31.02 |
| + **KagNet** | 83.26 | **82.15** | 36.38 | **34.57** |
| Human Performance | - | *99.5* | - | *88.9* |

# Experiments

## Ablation study

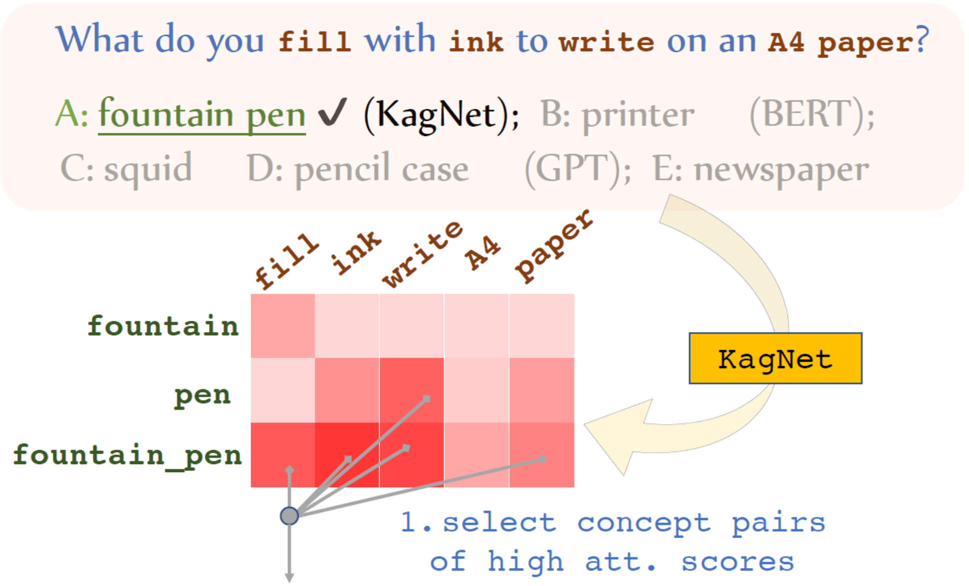
| Model | IHdev.(%) | IHtest.(%) |
|---|---|---|
| KAGNET (STANDARD) | 62.35 | 57.16 |
| : replace GCN-HPA-LSTM w/ R-GCN | 60.01 | 55.08 |
| : w/o GCN | 61.84 | 56.11 |
| : #GCN Layers = 1 | 62.05 | 57.03 |
| : w/o Path-level Attention | 60.12 | 56.05 |
| : w/o QAPair-level Attention | 60.39 | 56.13 |
| : using all paths (w/o pruning) | 59.96 | 55.27 |

Relational GCN

# Experiments

## Interpretability: transparent and interpretable inference process

*1. Concept-pair-level attention scores*




*2. Path-level attention scores*

```
ink —PartOf—> fountain_pen
ink —RelatedTo—> container <—IsA— fountain_pen

fill <—HasSubEvent— ink <—AtLocation— fountain_pen
fill —RelatedTo—> container <—IsA— fountain_pen

write <—UsedFor— pen
write <—UsedFor— pen <—IsA— fountain_pen

paper <—RelatedTo— write <—UsedFor— fountain_pen

····· 2. Ranking via path-level attn.
```

# Experiments

**Transferability:** testing with another task while fixing its parameters

Kagnet

BERT-Large

Trained by CommonsenseQA dataset

SWAG dataset: 59.01% vs 56.53%

WSC dataset: 53.51% vs 51.23%

# Conclusion

1) They proposed ***a knowledge-aware graph network (KagNet)***, which is based on GCN and LSTMs, with a hierarchical path-based attention mechanism.

2) ***Effectively utilize external knowledge graphs*** to model the relations of concepts in the questions and answers from both semantic and symbolic spaces.

3) KagNet achieved ***a new SOTA*** and provided results in ***transparent, interpretable way***

# Limitation

1. Negative reasoning: not sensitive to negation words

2. Comparative reasoning strategy

3. Subjective reasoning
   ex) *"Traveling from new place to new place is likely to be what?" "exhilarating" or "exhausting" ?*

# Limitation

## RoBERTa, not using ConceptNet

| | | | |
|---|---|---|---|
| RoBERTa (ensemble model) | Facebook AI | 08/13/2019 | 72.5 |
| FreeLB-RoBERTa (single model) | Microsoft Dynamics 365 AI Research & UMD | 10/03/2019 | 72.2 |
| RoBERTa + IR (single model) | Microsoft STCA-NLP team | 08/23/2019 | 72.1 |
| RoBERTa (single model) | Facebook AI | 08/13/2019 | 72.1 |

## ALBERT, using ConceptNet

| Model | Affiliation | Date | Accuracy | Accuracy (*Uses ConceptNet) |
|---|---|---|---|---|
| Human | | 03/10/2019 | 88.9 | |
| ALBERT+DESC-KCR (ensemble model) | Microsoft Cognitive Services Research | 12/02/2020 | | 83.3 |
| Albert+KD (ensemble model) | HIT-SCIR-QA | 12/30/2020 | | 80.9 |
| ALBERT+DESC-KCR (single model) | Microsoft Cognitive Services Research | 12/02/2020 | | 80.7 |
| ALBERT+KD (single model) | HIT-SCIR-QA | 12/10/2020 | | 80.3 |

# Q & A