Knowledge Graph based Question Answering

+

Harman Singh

Indian Institute of Technology Delhi

4th April, 2022

Overview

- Knowledge Graphs
- Multiple Choice QA Task Solvers (combining KG and LM)
  - KagNet
  - QA-GNN
  - GreaseLM
- Why do we need such complex architectures?
  - GNN is a counter? paper
- Answer generation in natural language, or answer selection from KG
  - KGT5 (link prediction + QA using T5)

Reasoning with Knowledge

If it is ~~not~~ used for **hair**, a **round brush** is an example of what?

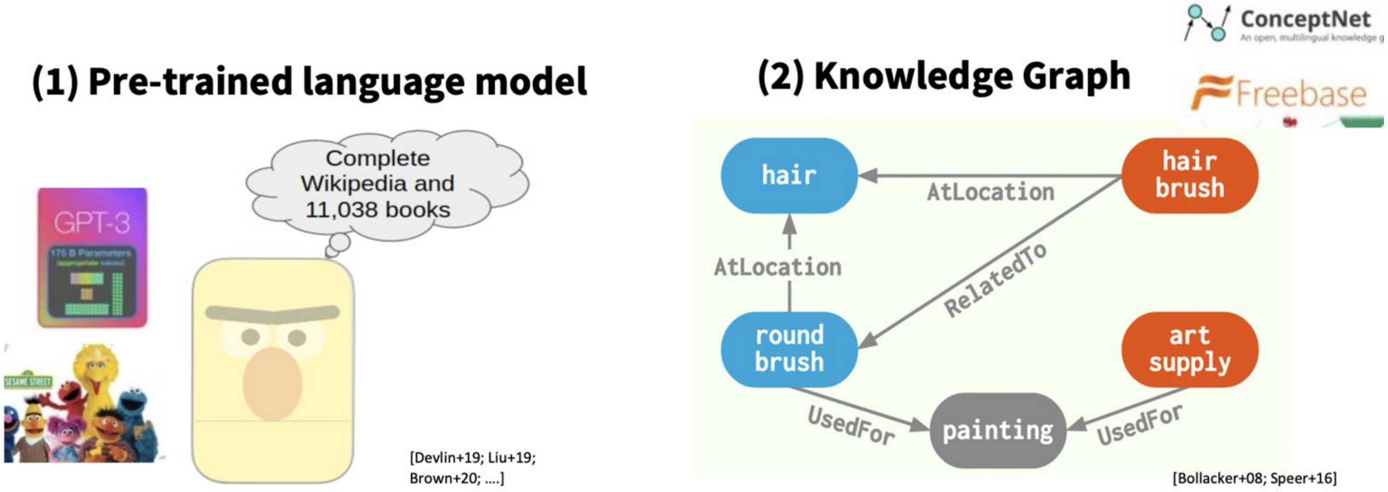
A. hair brush B. bathroom C. **art supplies*** D. shower

Q: How many of Warsaw's inhabitants spoke Polish in 1933?

Q: Who are current presidents of European countries who never held a world cup?

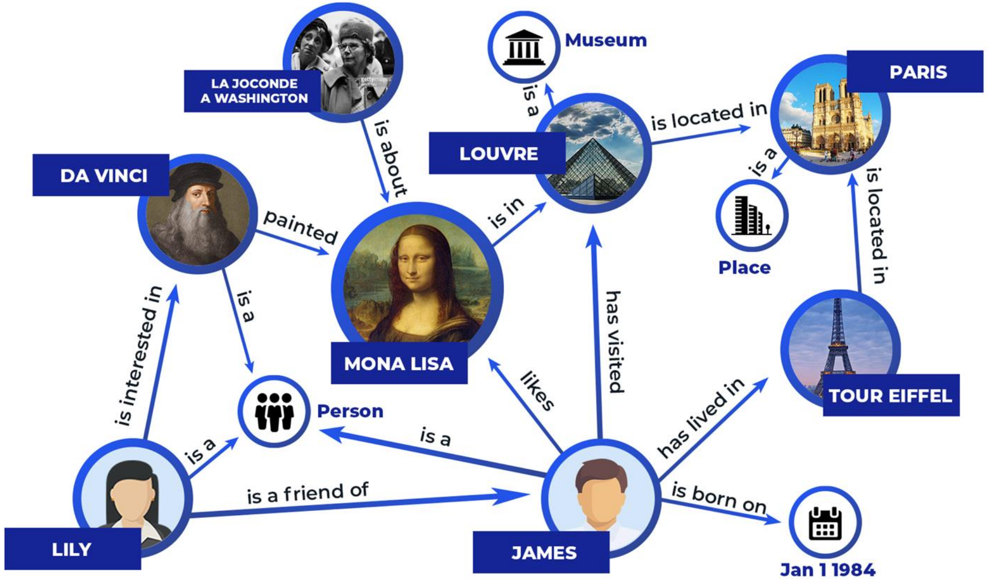
Where is the Knowledge?

Knowledge can be stored in a:



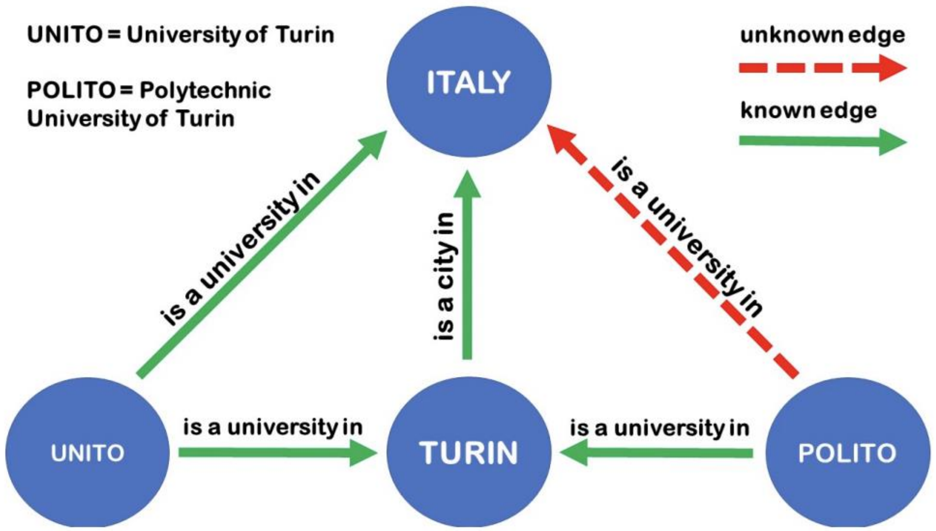
Knowledge Graphs

- Knowledge Graphs are heterogenous graphs
  - Multiple types of entities and relations exist
- Facts are represented as triples *(head, relation,tail)*
  - ('Paris', 'is_a', 'City')
  - ('India', 'population', '1.3B')
  - …



Benefits of KGs

- Explicitly stores knowledge
- Interpretable
- Easy to update and improve



Language Model's - Benefits

- Broad coverage
    - Trained over massive amounts of text
- Can encode practically

  anything that can

  be put in words
- Captures context

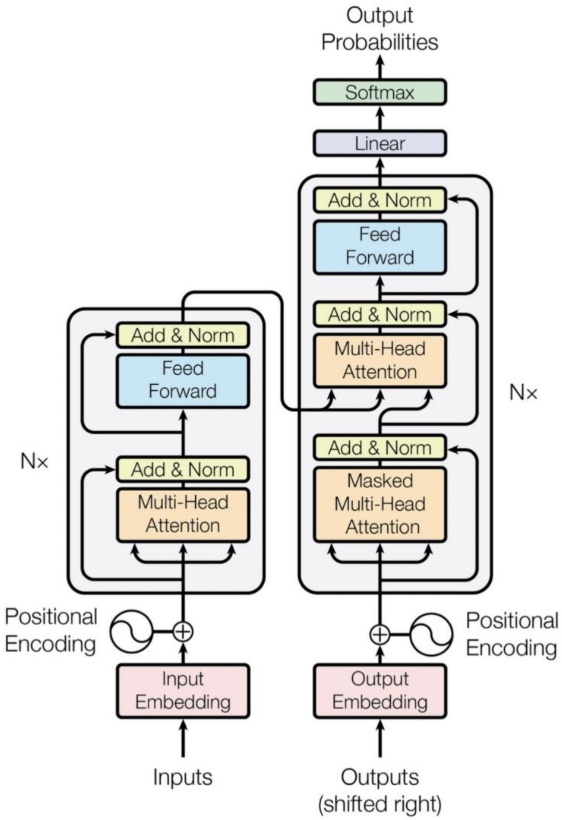
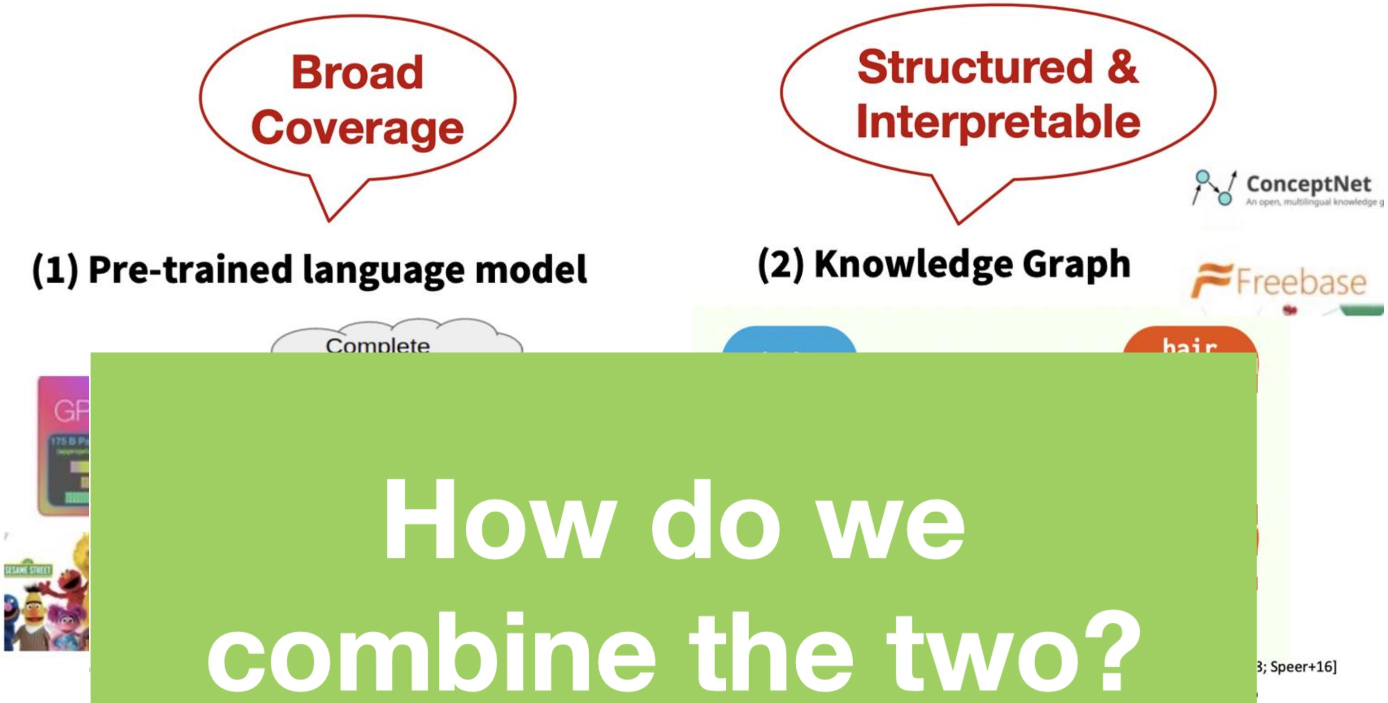


Figure 1: The Transformer - model architecture.

LM’s - Drawbacks

- Mysterious - Knowledge “hidden” in Weights
- Unclear how to improve them over time
- Not interpretable
- Cannot truly reason
    - For eg. BERT doing sophisticated string matching ?
- “Hidden” Biases

Leverage Both Knowledge



How do we combine the two?

Challenges in Knowledge Aware Reasoning

- How can we find a subgraph for reasoning?
  - KG/subgraphs are Noisy and Incomplete, also very large
  - Numerous subgraphs possible, how to select the most related ones?

- How do we encode the retrieved subgraph?
  - Complex Multi-Relational Graphs → how to model these?
  - No **supervision for aligning** graphs and Q-A pairs → distant supervision
  - Graph representation have to be compatible with Neural sentence encoders

KagNet, Lin et. al, EMNLP, 2019

KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning

Bill Yuchen Lin[†] **Xinyue Chen**[‡] **Jamin Chen**[†] **Xiang Ren**[†]

{yuchen.lin,jaminche,xiangren}@usc.edu, kiwisher@sjtu.edu.cn

[†]Computer Science Department, University of Southern California

[‡]Computer Science Department, Shanghai Jiao Tong University

KagNet

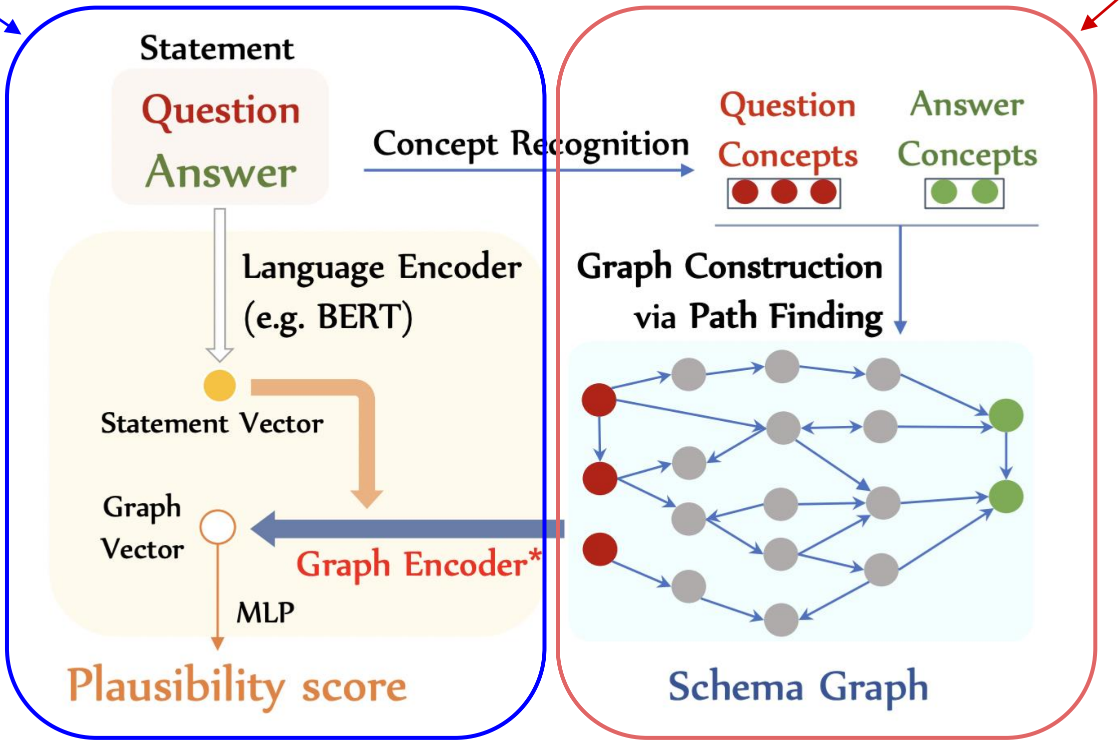
Overview

Reasoning

Subgraph construction

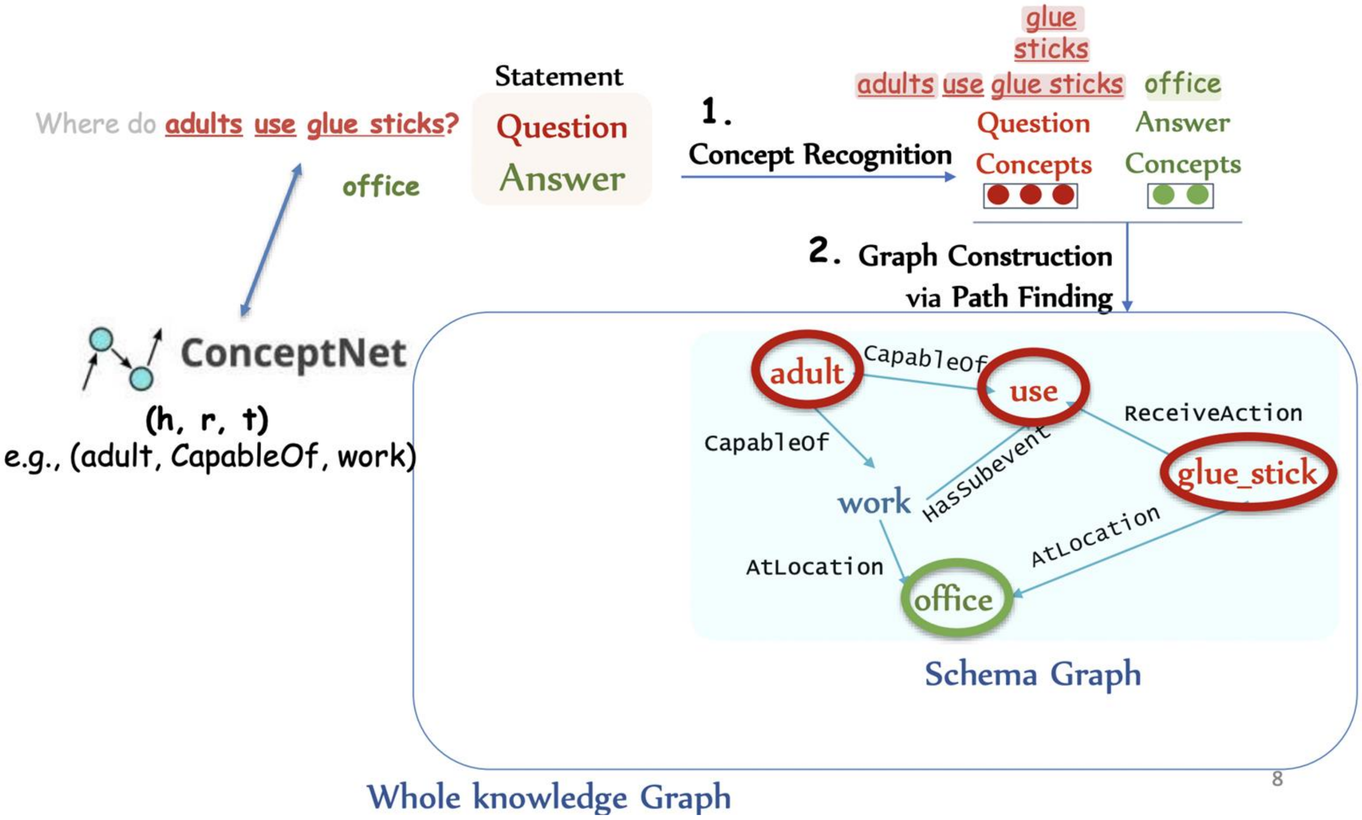
Given:

Question
+ 4 Answer options

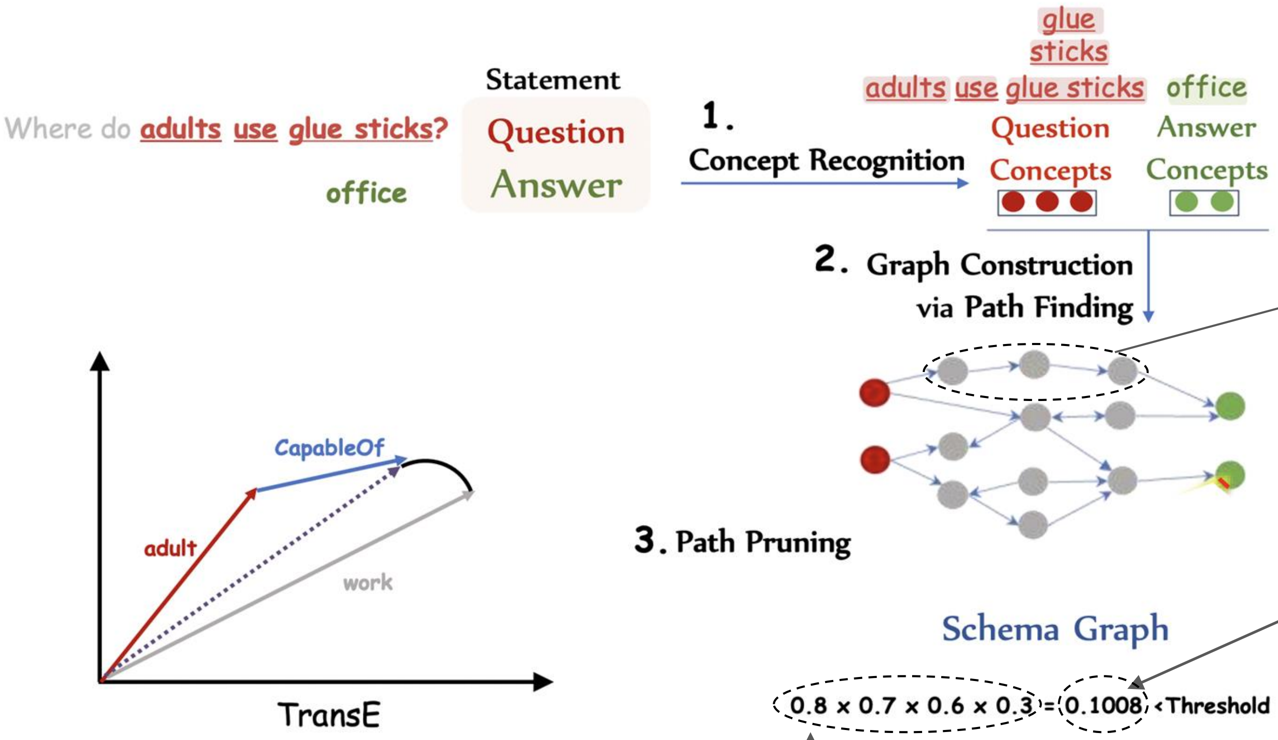


KagNet

Subgraph Construction



KagNet



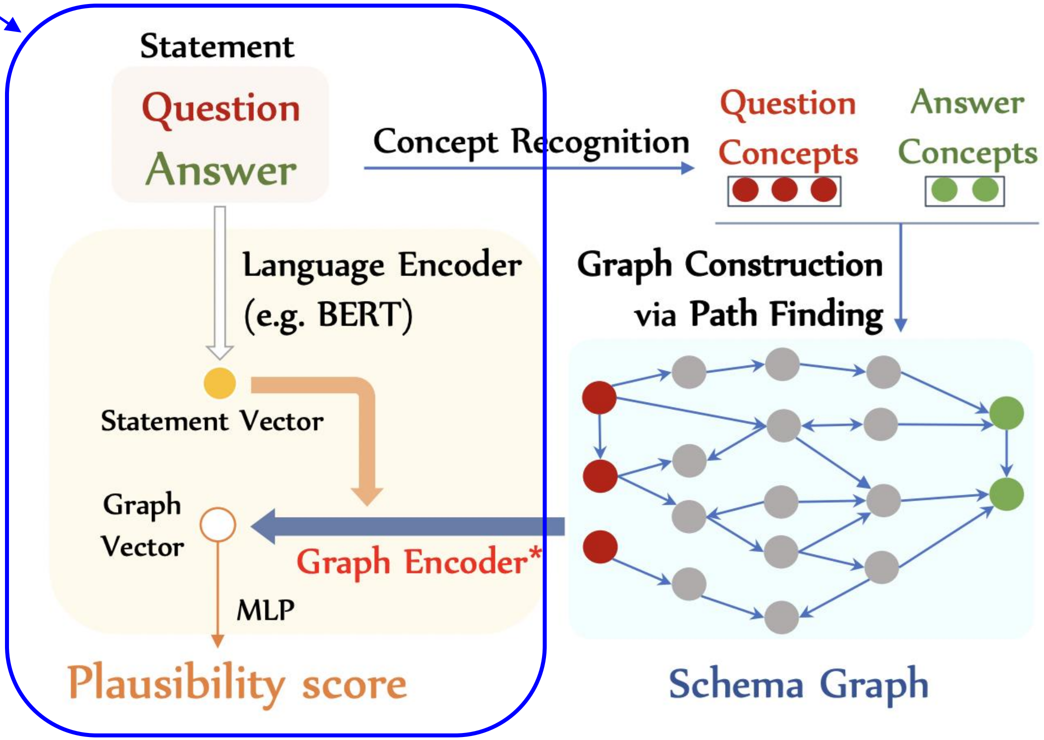
Path length < K =4

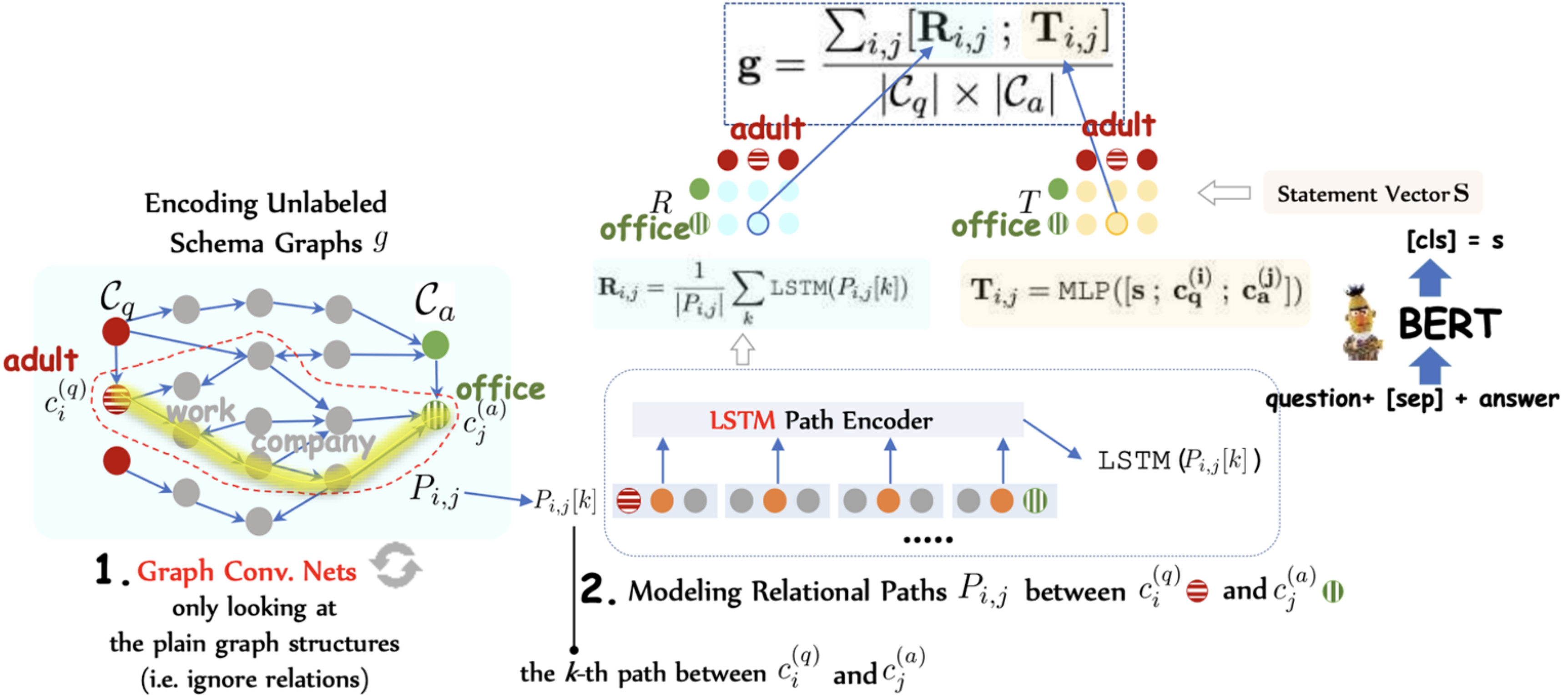
Path P score
= t1*t2*...tn

Scores for triples (t1, t2, …) in path P

KagNet

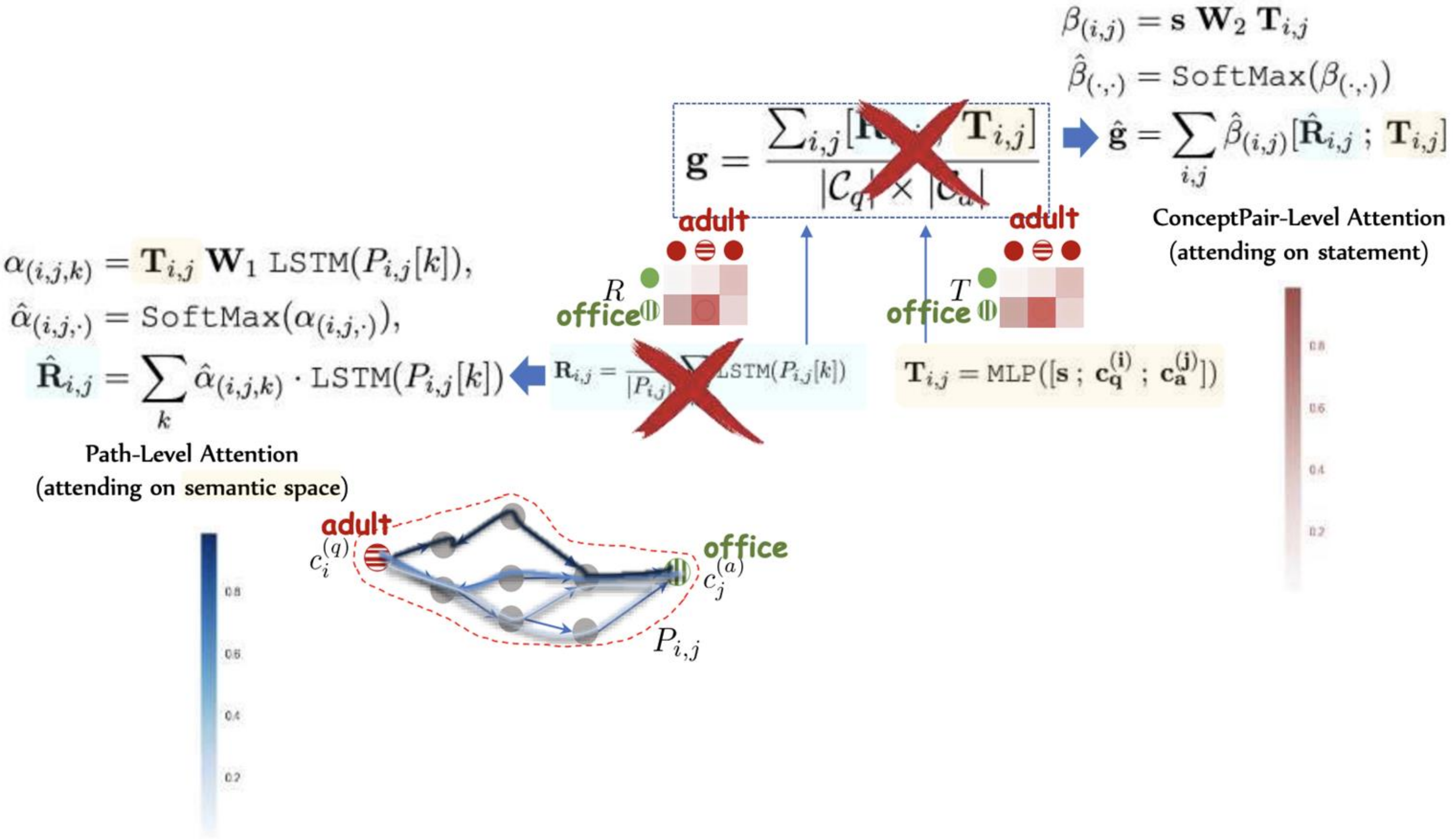
Reasoning



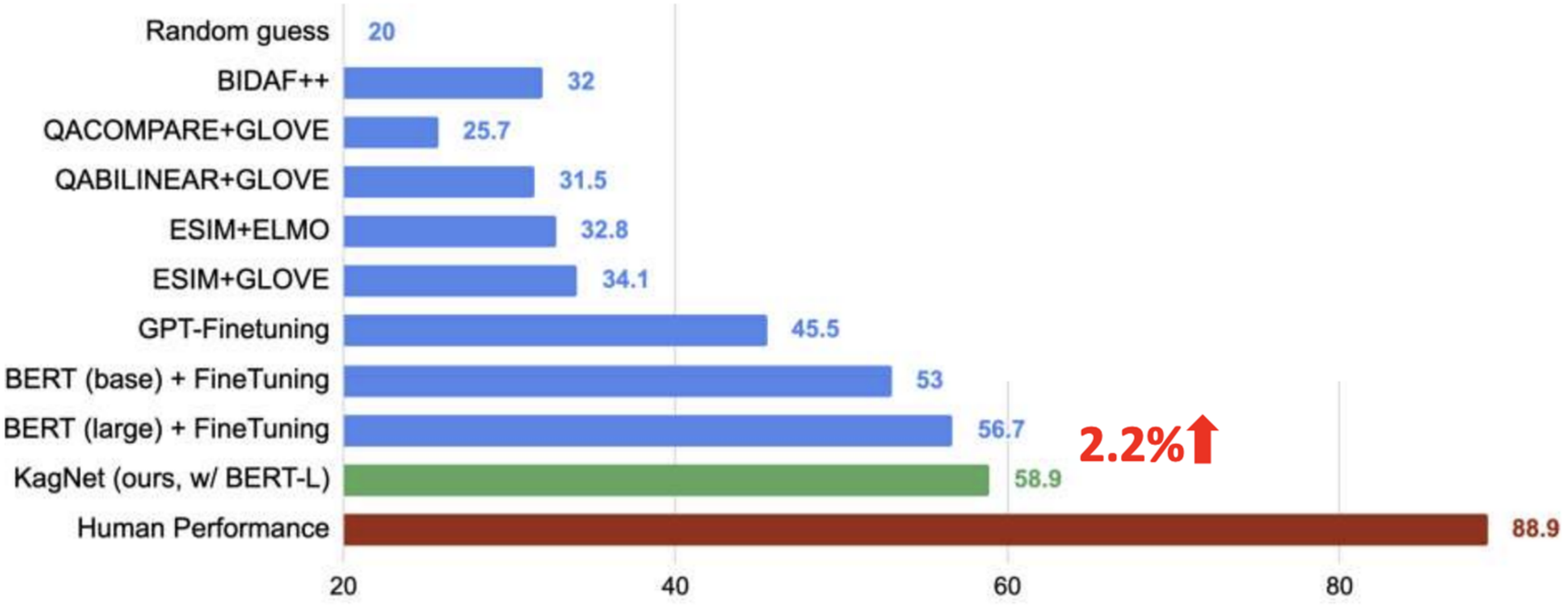


$\mathbf{g} = \frac{\sum_{i,j}[\mathbf{R}_{i,j}\ ;\ \mathbf{T}_{i,j}]}{|\mathcal{C}_q| \times |\mathcal{C}_a|}$
adult
adult
R
office
T
office
Statement Vector $\mathbf{s}$
[cls] = s
$\mathbf{R}_{i,j} = \frac{1}{|P_{i,j}|}\sum_k \mathtt{LSTM}(P_{i,j}[k])$
$\mathbf{T}_{i,j} = \mathtt{MLP}([\mathbf{s}\ ;\ \mathbf{c_q^{(i)}}\ ;\ \mathbf{c_a^{(j)}}])$
BERT
question+ [sep] + answer
Encoding Unlabeled Schema Graphs g
$\mathcal{C}_q$
$\mathcal{C}_a$
adult
$c_i^{(q)}$
office
$c_j^{(a)}$
work
company
$P_{i,j}$
$P_{i,j}[k]$
LSTM Path Encoder
$\mathtt{LSTM}\,(P_{i,j}[k])$
......
1. Graph Conv. Nets
only looking at
the plain graph structures
(i.e. ignore relations)
2. Modeling Relational Paths $P_{i,j}$ between $c_i^{(q)}$ and $c_j^{(a)}$
the k-th path between $c_i^{(q)}$ and $c_j^{(a)}$

Hierarchical Attention Mechanism



Comparison with standard baselines



Comparison with Knowledge Aware methods

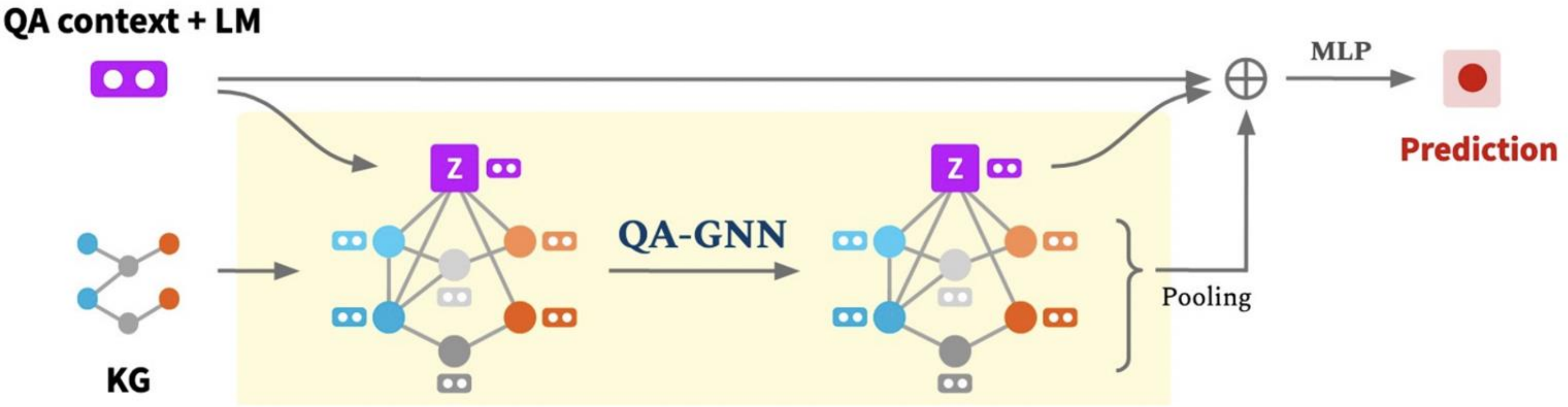
	Easy Mode		Hard Mode	
Model	IHdev.(%)	IHtest.(%)	IHdev.(%)	IHtest.(%)
Random guess	33.3	33.3	20.0	20.0
BLSTMs	80.15	78.01	34.79	32.12
+ KV-MN	81.71	79.63	35.70	33.43
+ CSPT	81.79	80.01	35.31	33.61
+ TEXTGRAPHCAT	82.68	81.03	34.72	33.15
+ TRIPLESTRING	79.11	76.02	33.19	31.02
+ **KAGNET**	83.26	**82.15**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

QA-GNN

Yasunaga et. al, NAACL, 2021

Key Innovations:

1. Language-conditioned KG node relevance scoring
2. Joint Reasoning:
   a. Connect text and KG to form a joint graph
   b. Mutually update representations via GNN



This and following QA-GNN slides are adapted/modified/taken directly from Jure Leskovec (NAACL HLT keynote)

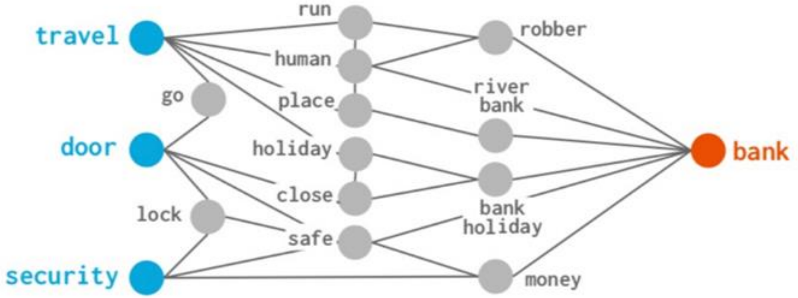
Existing Subgraph Retrieval Methods

QA Context

A **revolving door** is convenient for **two direction travel**, but also serves as a **security measure** at what?

A. **bank*** B. library C. department store
D. mall E. new york

Retrieved KG



Identify topic entities in the text:
travel, door, security, bank

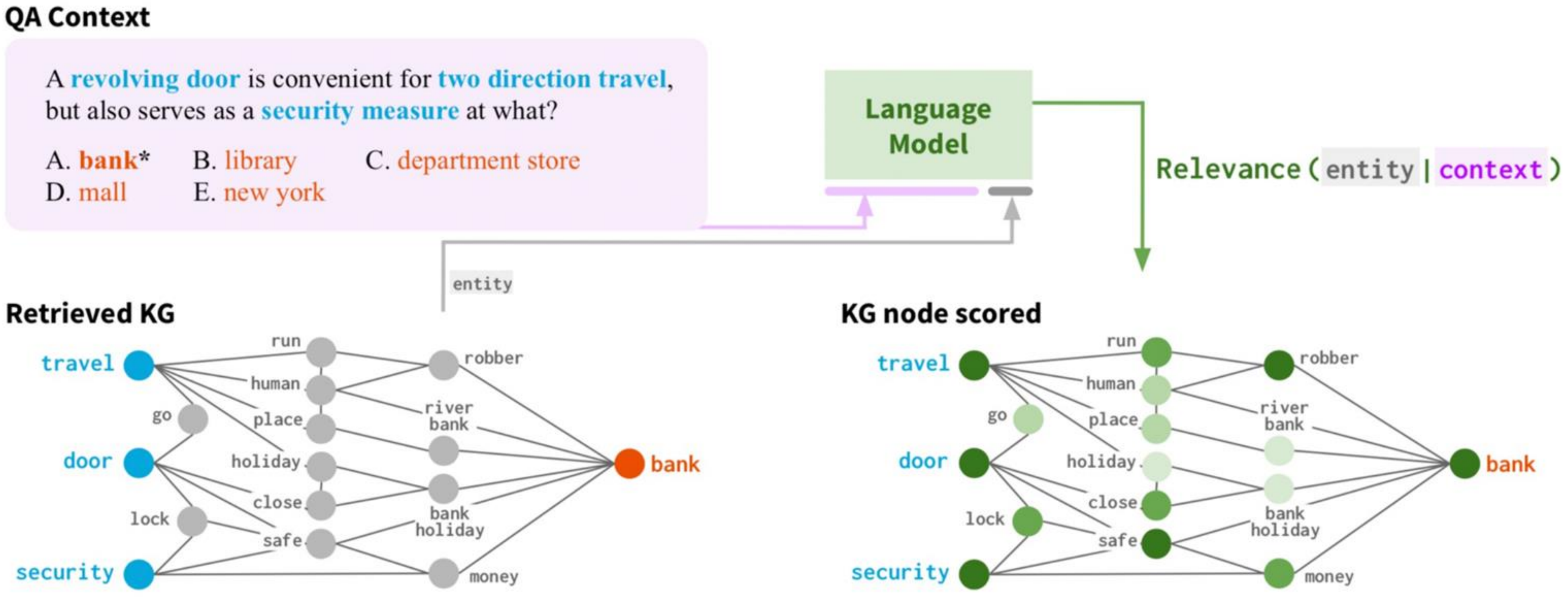
Retrieve *k*-hop heighbors/paths in KG

Some entities are irrelevant to the given QA context

- Off-topic - e.g. **holiday**
- Polysemy - e.g. **river_bank**
- Generic - e.g. **human, place**

QA-GNN

(1) Score KG nodes by LM

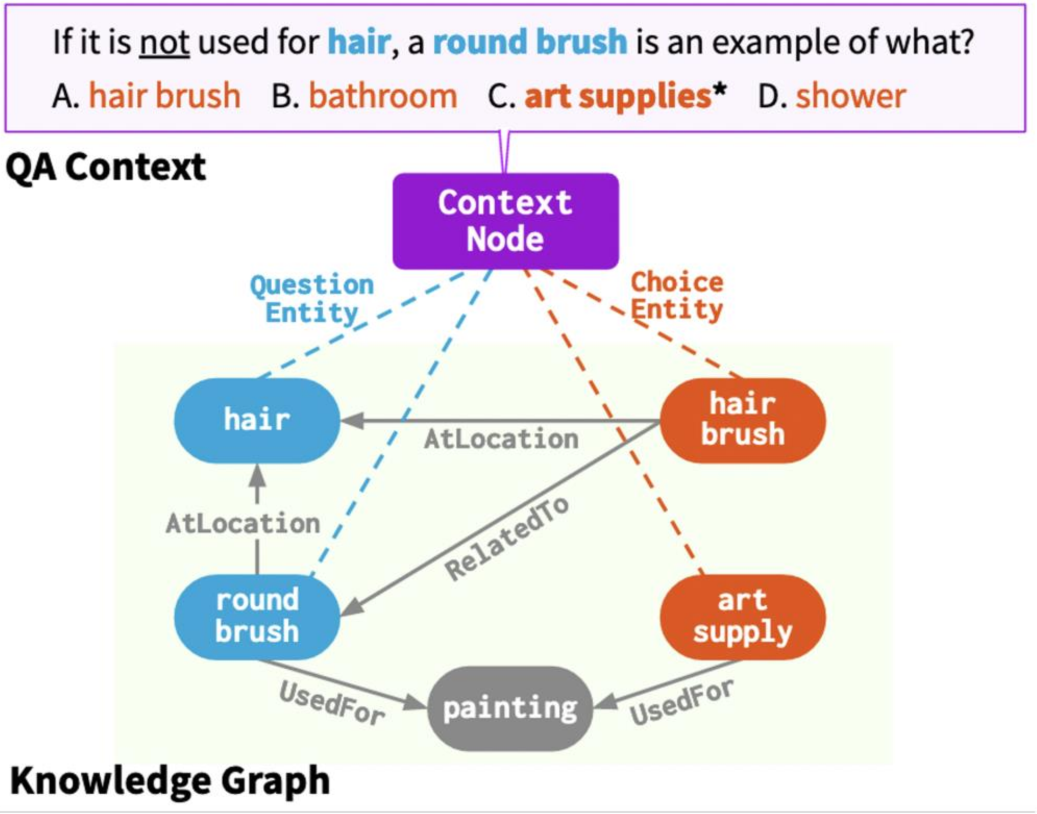


Some entities are irrelevant to the given QA context!

Entity relevance estimated by LM. Darker color indicates higher score.

QA-GNN

(2) Joint Reasoning



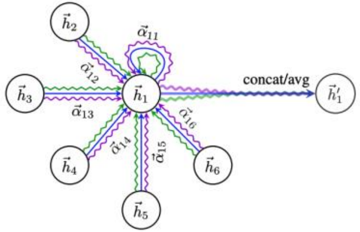
Joint Reasoning

QA-GNN Message Passing

$$\boldsymbol{h}_t^{(\ell+1)} = \boldsymbol{F}_n\left(\sum_{s\in\mathcal{N}_t\cup\{t\}} \alpha_{st}\boldsymbol{m}_{st}\right) + \boldsymbol{h}_t^{(\ell)}$$

Attention (s → t)

Message (s → t)



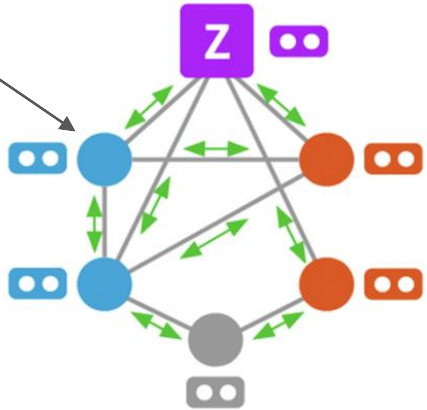
$$\boldsymbol{m}_{st} = f_m(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{r}_{st}),$$

$$\boldsymbol{q}_s = f_q(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{\rho}_s),$$

$$\boldsymbol{k}_t = f_k(\boldsymbol{h}_t^{(\ell)}, \boldsymbol{u}_t, \boldsymbol{\rho}_t, \boldsymbol{r}_{st})$$

$$\alpha_{st} = \frac{\exp(\gamma_{st})}{\sum_{t'\in\mathcal{N}_s\cup\{s\}}\exp(\gamma_{st'})}, \quad \gamma_{st} = \frac{\boldsymbol{q}_s^\top \boldsymbol{k}_t}{\sqrt{D}}.$$

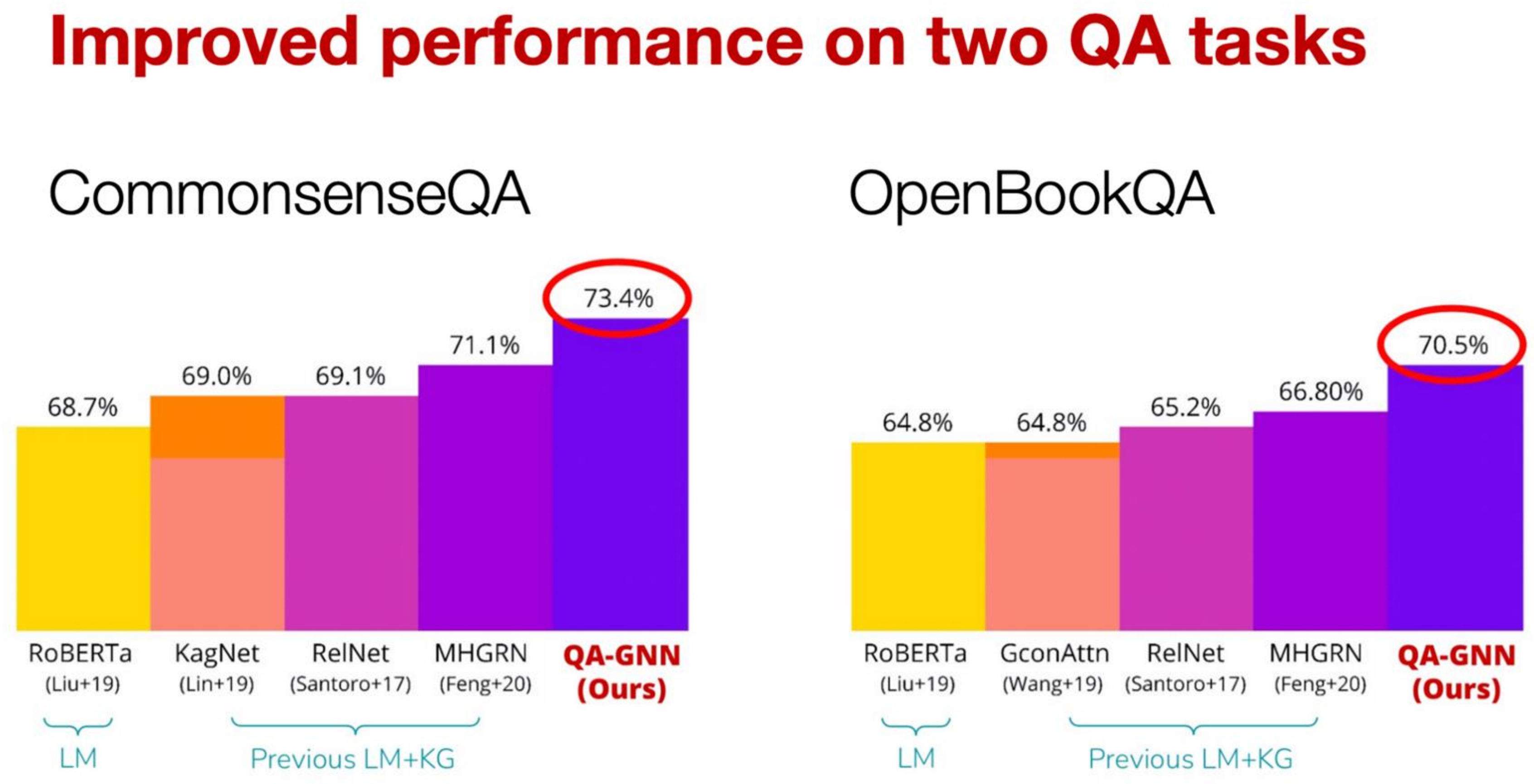
Initial Vector:
Mean pooled BERT embeddings
=> Refine with GNN



Node types

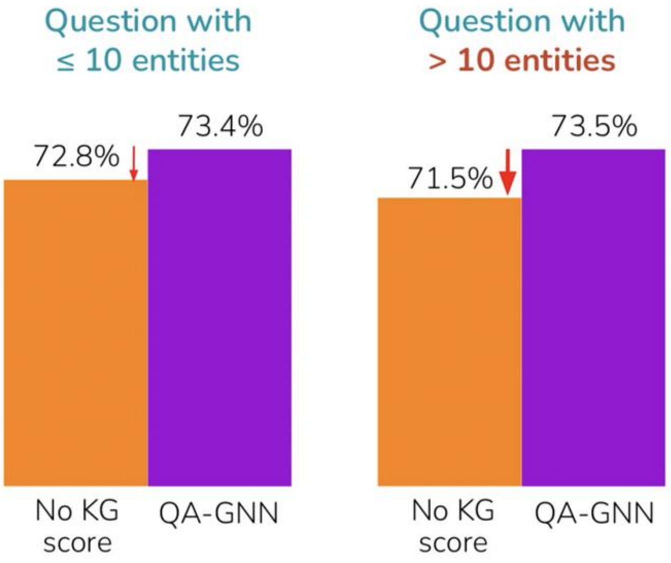
- Context
- Question entity
- Answer entity
- Other entity

Performance

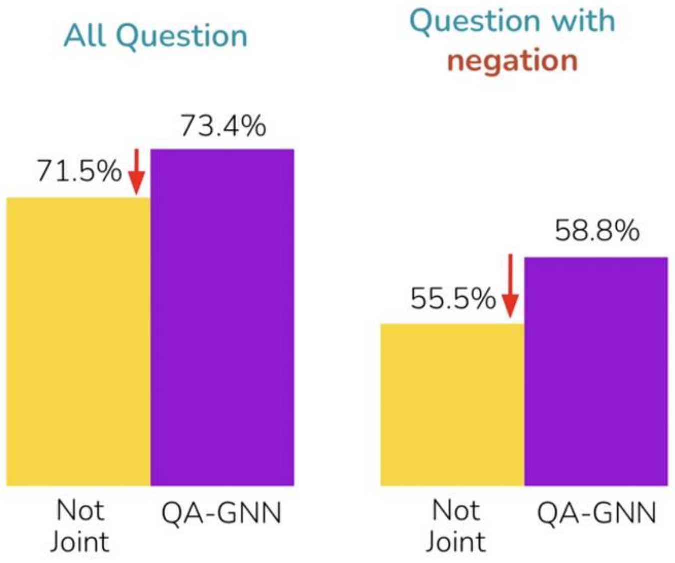


Analysis

- Node scoring tends to help when retrieved KG is big

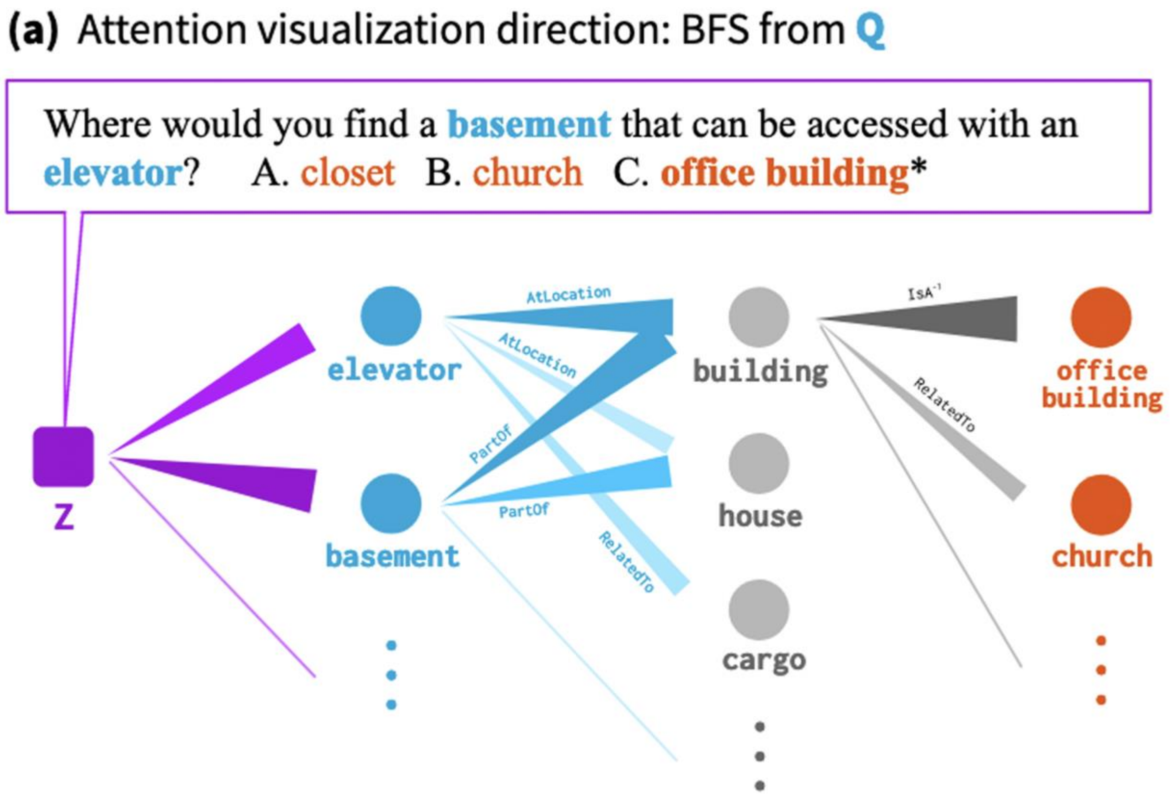


- Joint graph tends to help when question has negation



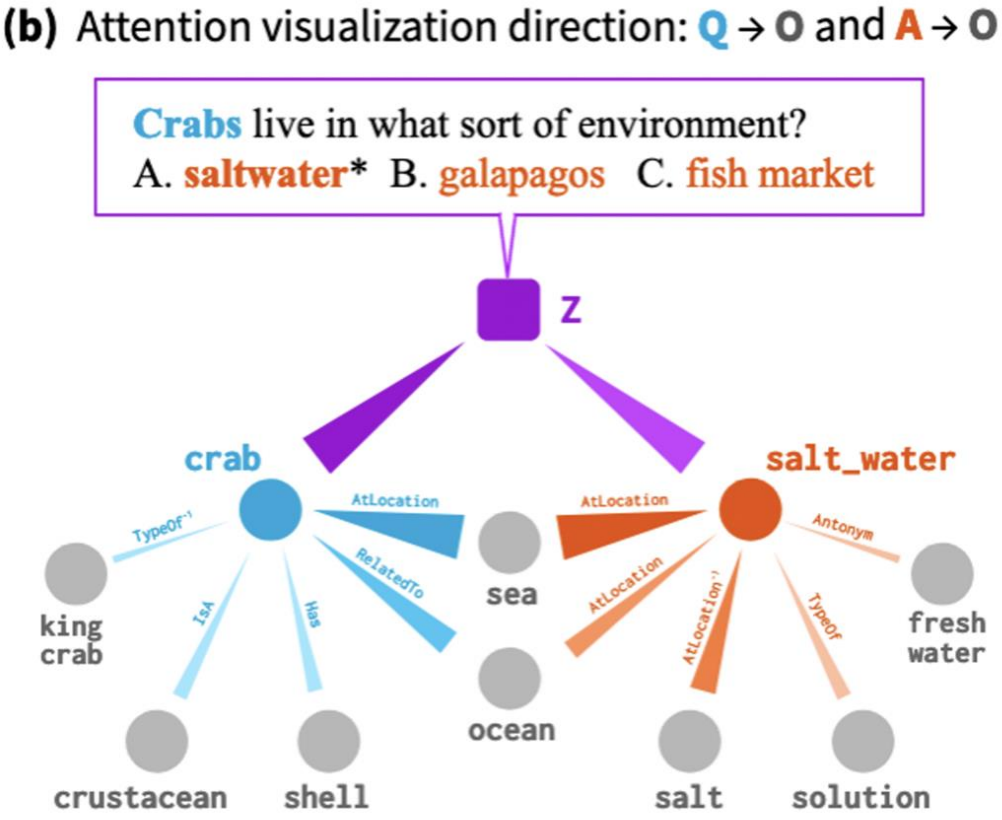
QA-GNN

Benefit 1: Interpretability



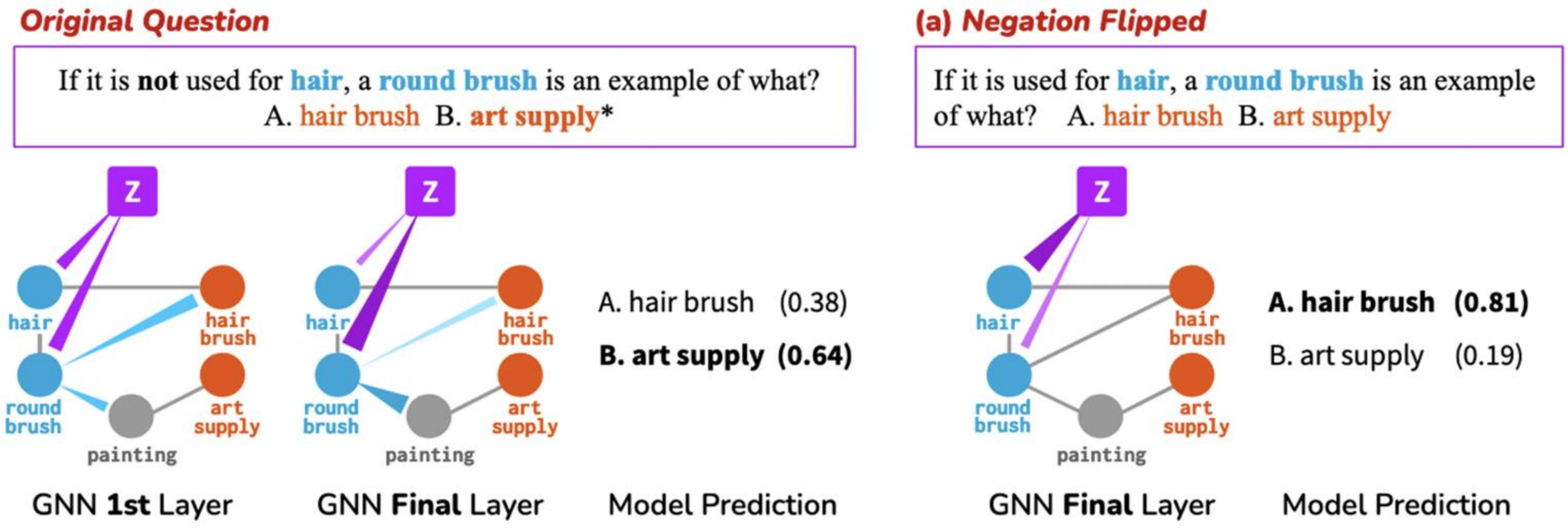
QA-GNN

Benefit 1: Interpretability



QA-GNN

Benefit 2: Structured Reasoning



Better Handling of negation or entity substitution

GreaseLM

Zhang et. al, ICLR, **2022**

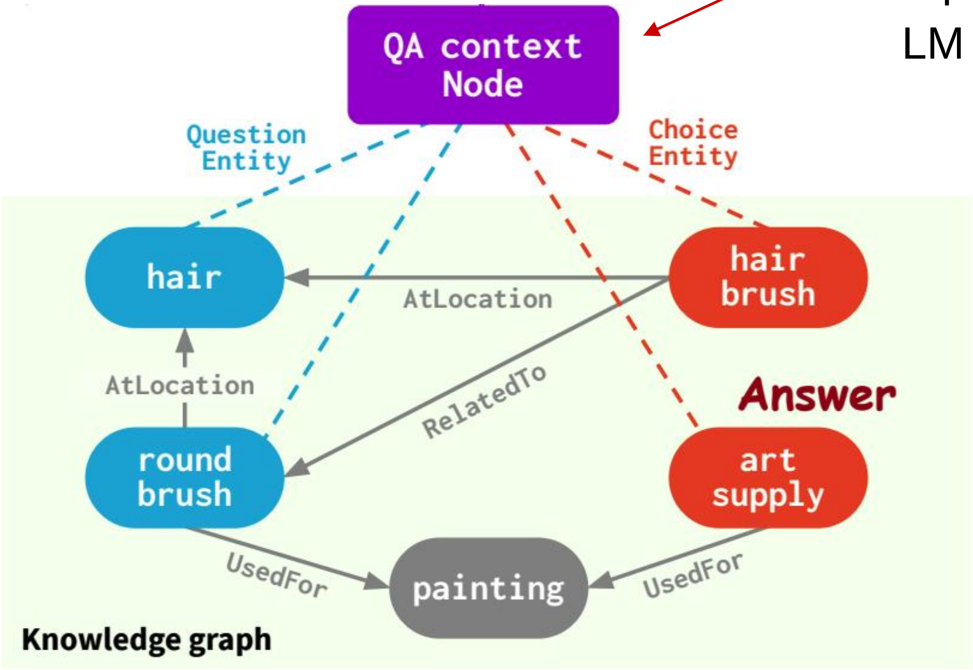
GreaseLM: Graph REASoning Enhanced Language Models for Question Answering

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren
Percy Liang, Christopher D. Manning, Jure Leskovec
Stanford University
{xikunz2,antoineb,myasu,hyren,pliang,manning,jure}@cs.stanford.edu

GreaseLM

“You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!”

Drawback of QA-GNN



single pooled representation of the LM

GreaseLM

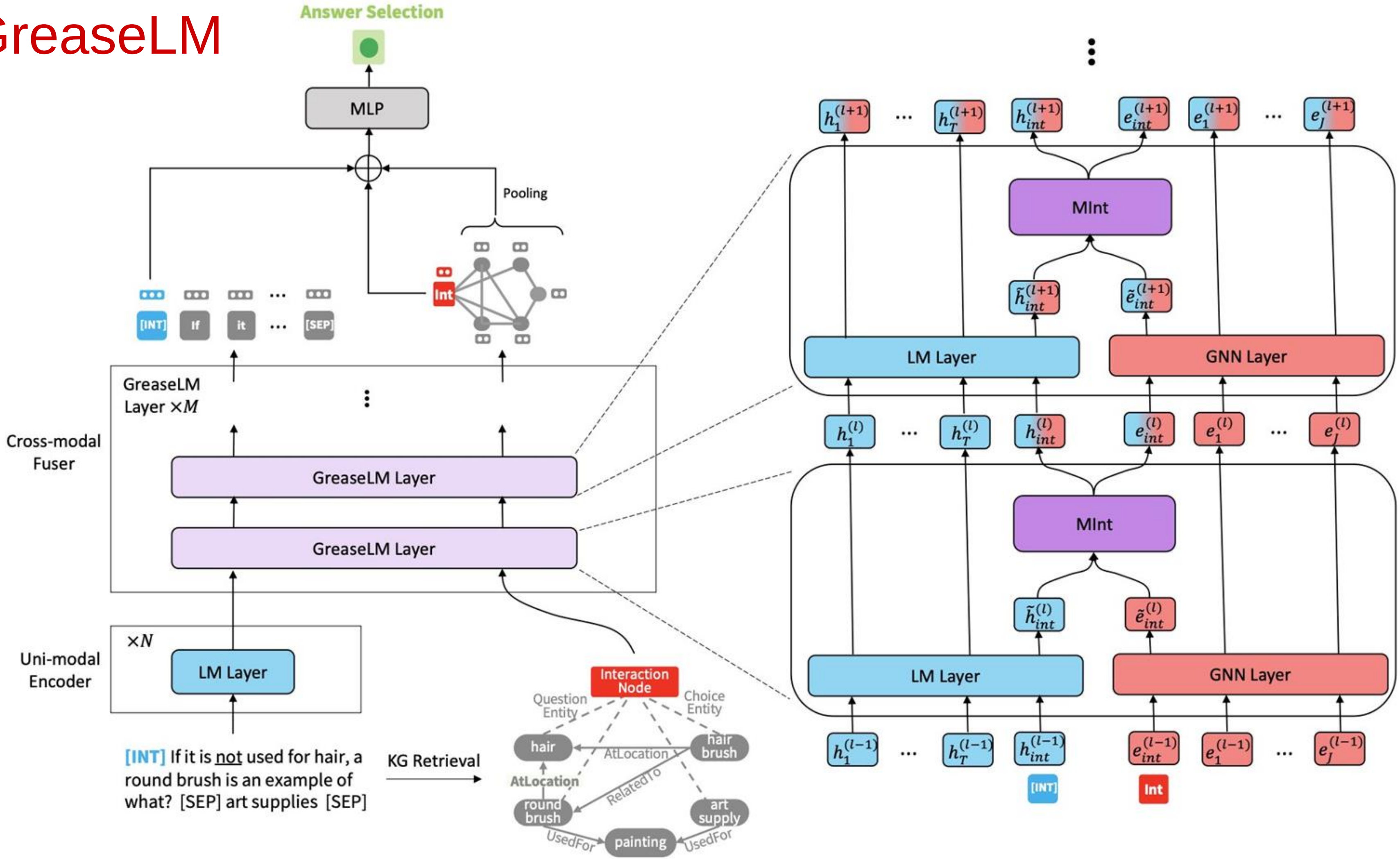
Key Innovation:

Individual token representations in the LM and node representations in the GNN **mix (interact) for multiple layers**

$$\{\boldsymbol{h}_{int}^{(\ell)}, \boldsymbol{h}_1^{(\ell)}, \ldots, \boldsymbol{h}_T^{(\ell)}\} \qquad \{\tilde{\boldsymbol{e}}_{int}^{(\ell)}, \tilde{\boldsymbol{e}}_1^{(\ell)}, \ldots, \tilde{\boldsymbol{e}}_J^{(\ell)}\}$$

We need more interaction!!

GreaseLM



Answer Selection
MLP
Pooling
[INT]
If
it
[SEP]
Int
GreaseLM
Layer ×M
Cross-modal
Fuser
GreaseLM Layer
GreaseLM Layer
×N
Uni-modal
Encoder
LM Layer
[INT] If it is not used for hair, a round brush is an example of what? [SEP] art supplies [SEP]
KG Retrieval
Interaction Node
Question Entity
Choice Entity
hair
hair brush
AtLocation
AtLocation
RelatedTo
round brush
art supply
UsedFor
painting
UsedFor
MInt
LM Layer
GNN Layer
MInt
LM Layer
GNN Layer
[INT]
Int

GreaseLM

Performance

- Better performance on standard datasets

CommonsenseQA

Methods	IHdev-Acc. (%)	IHtest-Acc. (%)
RoBERTa-Large (w/o KG)	73.1 ($\pm$0.5)	68.7 ($\pm$0.6)
RGCN (Schlichtkrull et al., 2018)	72.7 ($\pm$0.2)	68.4 ($\pm$0.7)
GconAttn (Wang et al., 2019)	72.6 ($\pm$0.4)	68.6 ($\pm$1.0)
KagNet (Lin et al., 2019)	73.5 ($\pm$0.2)	69.0 ($\pm$0.8)
RN (Santoro et al., 2017)	74.6 ($\pm$0.9)	69.1 ($\pm$0.2)
MHGRN (Feng et al., 2020)	74.5 ($\pm$0.1)	71.1 ($\pm$0.8)
QA-GNN (Yasunaga et al., 2021)	76.5 ($\pm$0.2)	73.4 ($\pm$0.9)
GreaseLM (**Ours**)	**78.5** ($\pm$0.5)	**74.2** ($\pm$0.4)

OpenbookQA

Model	Acc.	# Params
ALBERT (Lan et al., 2020) + KB	81.0	$\sim$235M
HGN (Yan et al., 2020)	81.4	$\geq$355M
AMR-SG (Xu et al., 2021)	81.6	$\sim$361M
ALBERT + KPG (Wang et al., 2020)	81.8	$\geq$235M
QA-GNN (Yasunaga et al., 2021)	82.8	$\sim$360M
T5* (Raffel et al., 2020)	83.2	**$\sim$3B**
T5 + KB (Pirtoaca)	85.4	**$\geq$11B**
UnifiedQA* (Khashabi et al., 2020)	**87.2**	**$\sim$11B**
GreaseLM (**Ours**)	84.8	$\sim$359M

GreaseLM

Performance

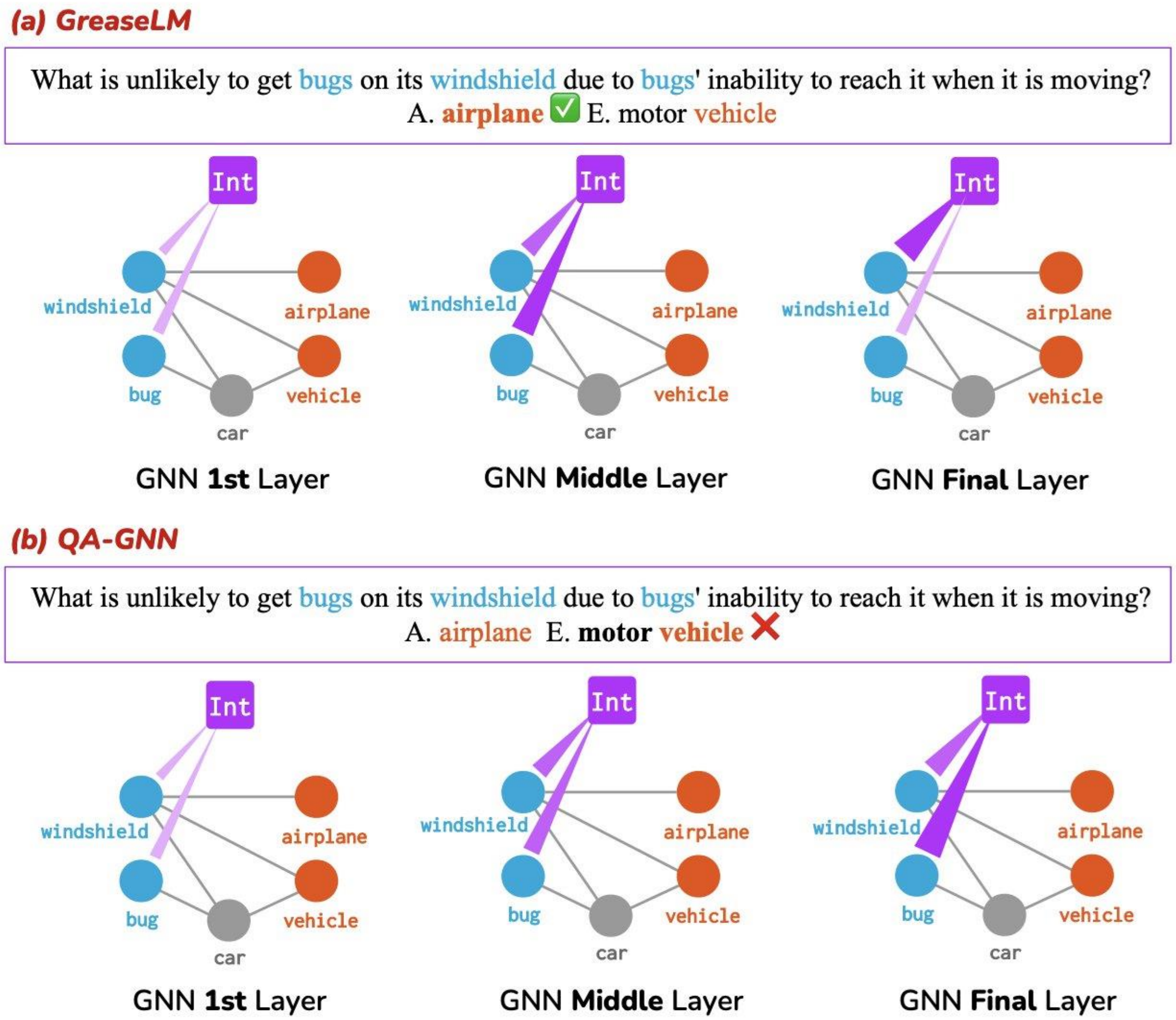
- Better performance on complex questions

Table 5: Performance of GREASELM on the *CommonsenseQA* IH-dev set on complex questions with semantic nuance such as prepositional phrases, negation terms, and hedge terms.

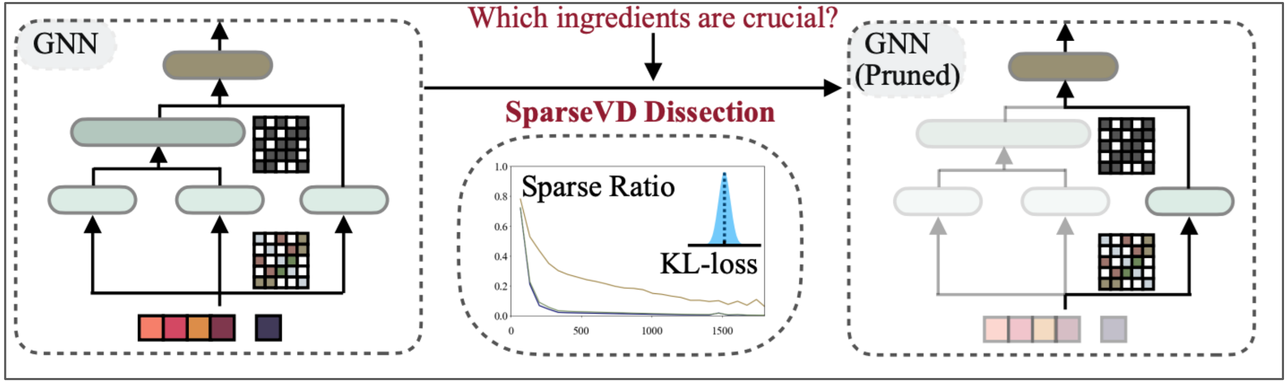
Model	**# Prepositional Phrases**					**Negation**	**Hedge**
	0	1	2	3	4	**Term**	**Term**
n	210	429	316	171	59	83	167
RoBERTa-Large	66.7	72.3	76.3	74.3	69.5	63.8	70.7
QA-GNN	**76.7**	76.2	79.1	74.9	81.4	66.2	76.0
GREASELM (**Ours**)	75.7	**79.3**	**80.4**	**77.2**	**84.7**	**69.9**	**78.4**

GreaseLM

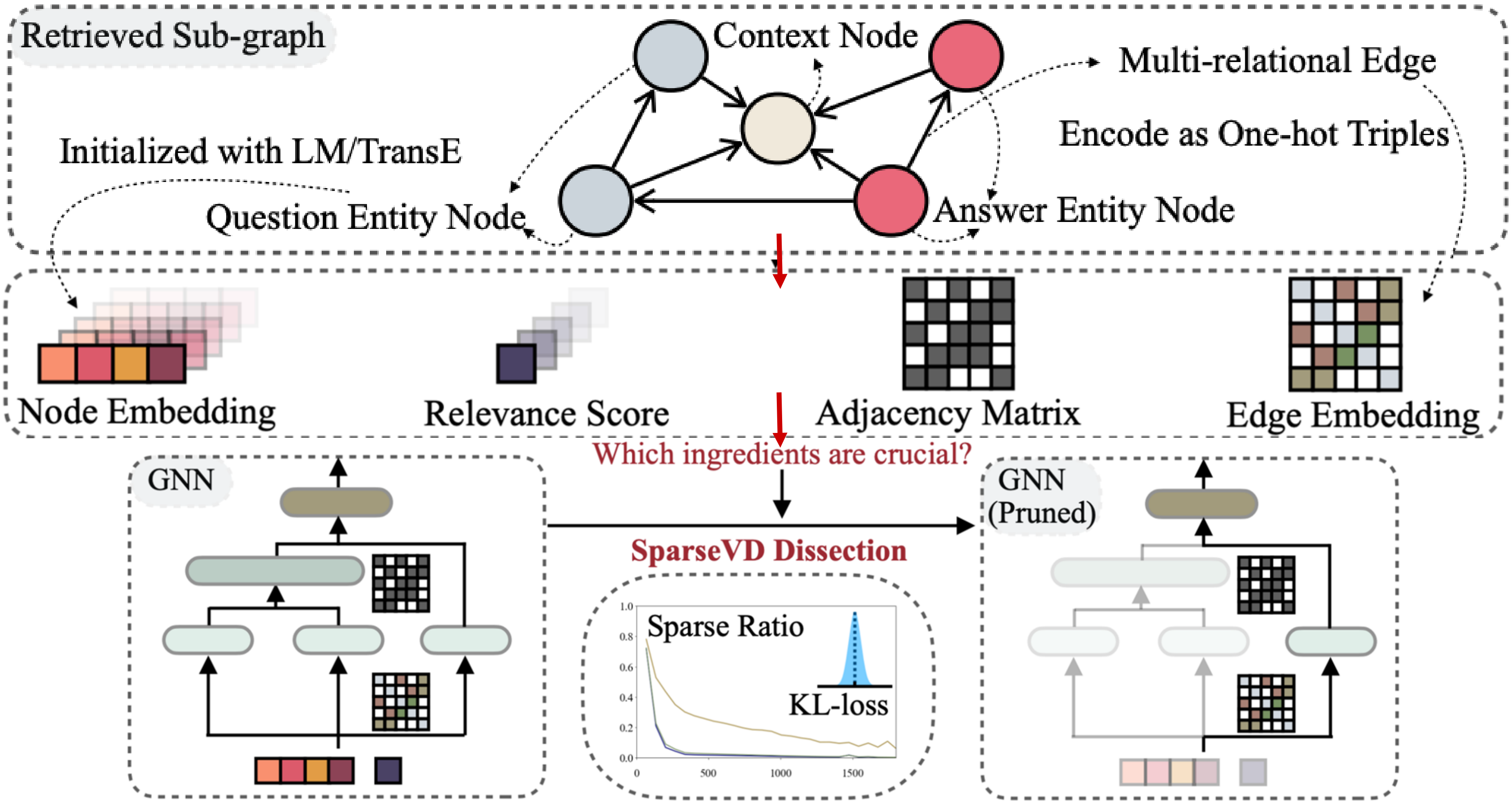
Benefit:
Better Attention patterns than QA-GNN



GNN as a counter? Revisiting GNN for QA Wang et. al, ICLR, 2022

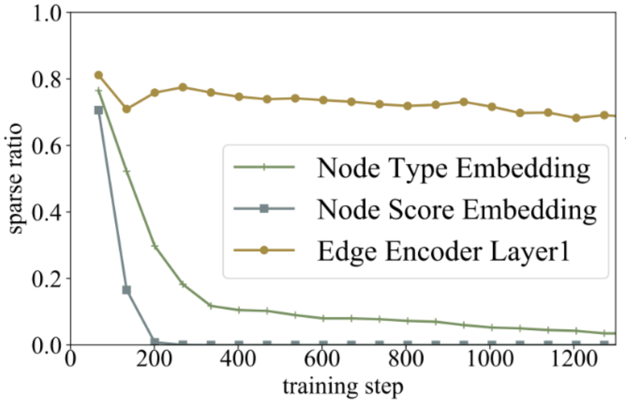


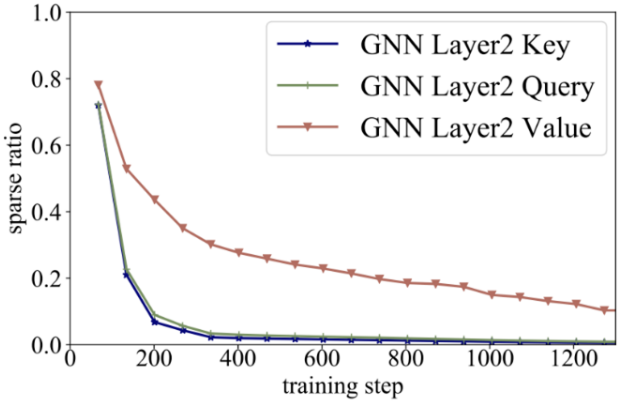
- Analysis of existing GNN modules
  - Used SparseVD (pruning) to analyse importance of different parts of GNN architectures for QnA
- Importance of edge counting
  - Counting edges in a graph => important for qNa
- Design of a GSC (Graph soft counter)
  - Replace complex GNN with a “very-very” simple GNN

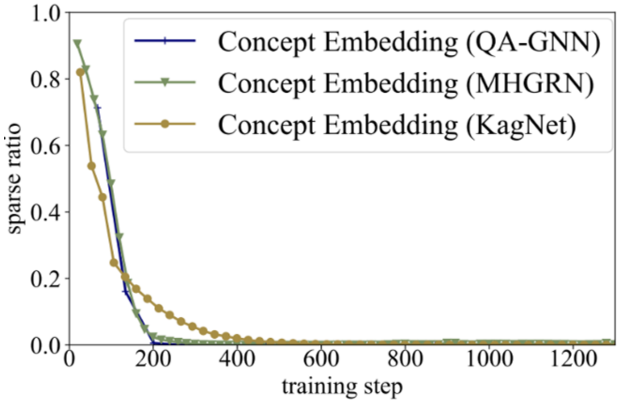

Retrieved Sub-graph
Context Node
Multi-relational Edge
Encode as One-hot Triples
Initialized with LM/TransE
Question Entity Node
Answer Entity Node
Node Embedding
Relevance Score
Adjacency Matrix
Edge Embedding
Which ingredients are crucial?
GNN
GNN (Pruned)
SparseVD Dissection
Sparse Ratio
KL-loss

Pruning → Prune different NN layers

Y axis → Sparse Ratio (lower means the weights **can be made sparse**)







-Node type
-Node score embeddings
-Edge Encoder layer

GNN Layers

$$\boldsymbol{h}_t^{(\ell+1)} = f_n\left(\sum_{s\in\mathcal{N}_t\cup\{t\}} \alpha_{st}\boldsymbol{m}_{st}\right) + \boldsymbol{h}_t^{(\ell)},$$

Initial Node embedding layers f_embed(N0)

N0 → initial embeddings

Pruning Results

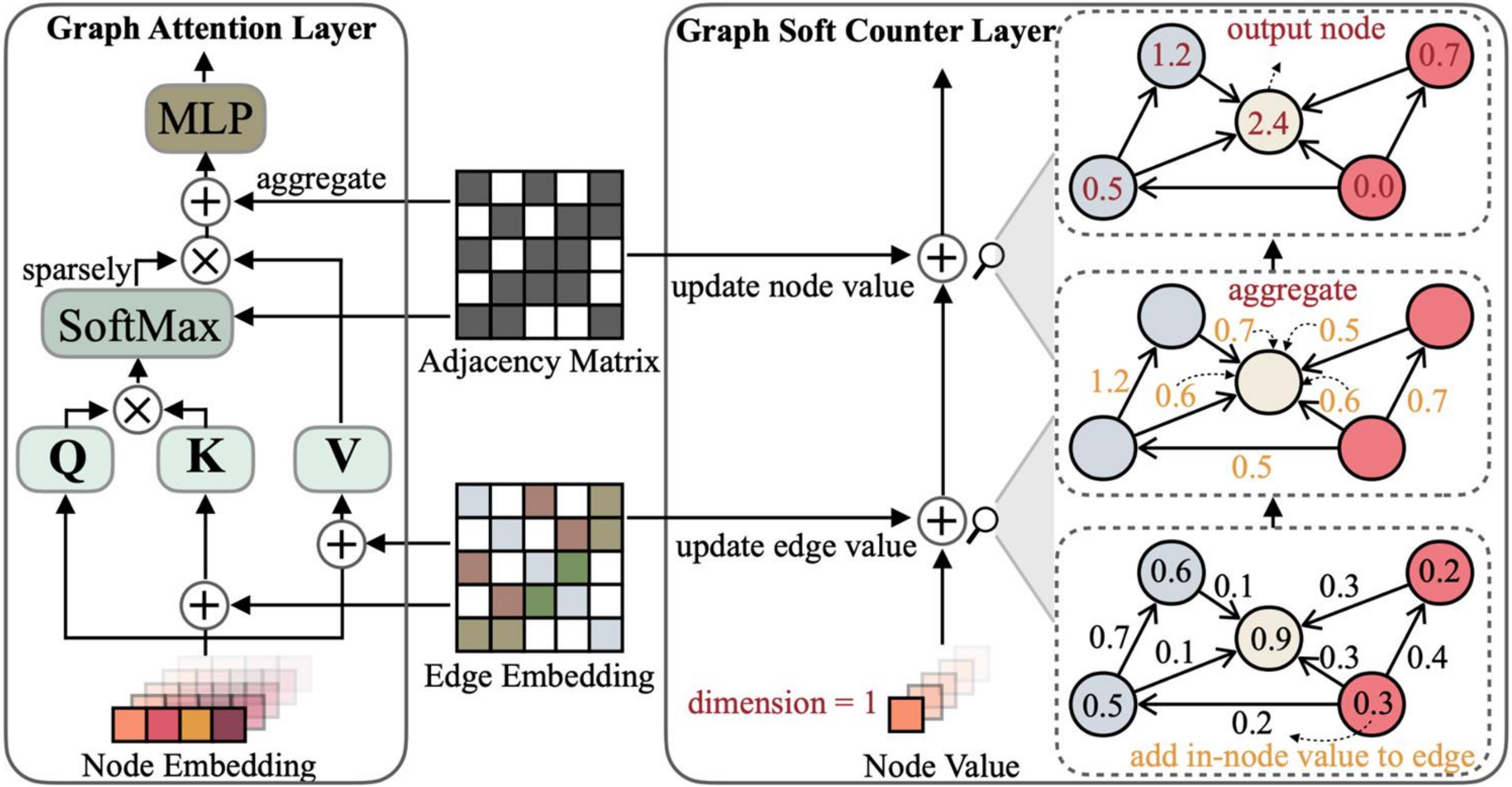
2 loss terms →

Maximize accuracy on CommonsenseQA + Minimize KL divergence

	w/o SparseVD		w/ SparseVD	
Methods	**IHdev-Acc. (%)**	**IHtest-Acc. (%)**	**IHdev-Acc. (%)**	**IHtest-Acc. (%)**
KagNet (Lin et al., 2019)	73.47 (±0.22)	69.01 (±0.76)	75.18 (±1.05)	70.48 (±0.77)
MHGRN (Feng et al., 2020)	74.45 (±0.10)	71.11 (±0.81)	77.15 (±0.32)	72.66 (±0.61)
QAGNN (Yasunaga et al., 2021)	76.54 (±0.21)	73.41 (±0.92)	77.64 (±0.50)	73.57 (±0.48)

Table 1: To preserve the reasoning ability for analysis, our SparseVD tool prunes the GNN-based models without loss of accuracy on *CommonsenseQA* in-house split.

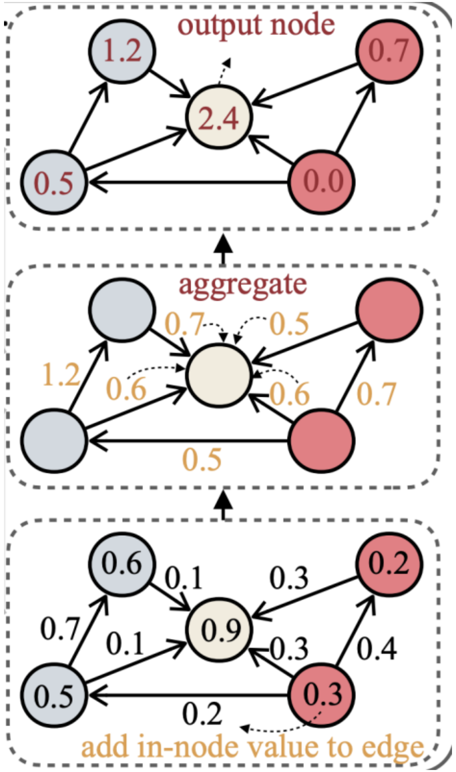
GSC (graph soft counter)



- Node embedding not needed, only have 1 dim node value
- Edge embedding replaced with 1 dim edge value (output by edge encoder)
- Only 32 retrieved nodes are enough !!! (QA-GNN uses 200)
- GAttNet reduced to 2 simple steps

  1) update the edge value with in-node

  2) update the node value by aggregating the edge



	KagNet	MHGRN	QAGNN	**GSC** (Ours)
Adj-matrix	✓	✓	✓	✓
Edge-type	✓	✓	✓	✓
Node-type	×	✓	✓	✓
Node-embedding	✓	✓	✓	×
Relevance-score	×	×	✓	×
#Learnable Param	700k	547k	2845k	**3k**
Model size	819M	819M	821M	**3k**

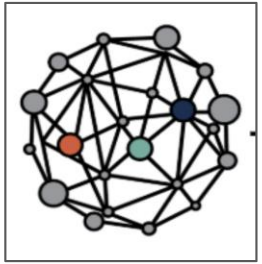
Number of parameters used by different models

KGT5

Saxena et. al, ACL, **2022**

TASK: KBQA (Knowledge Base Question Answering)

- Given: Question + KB (entities and relations)



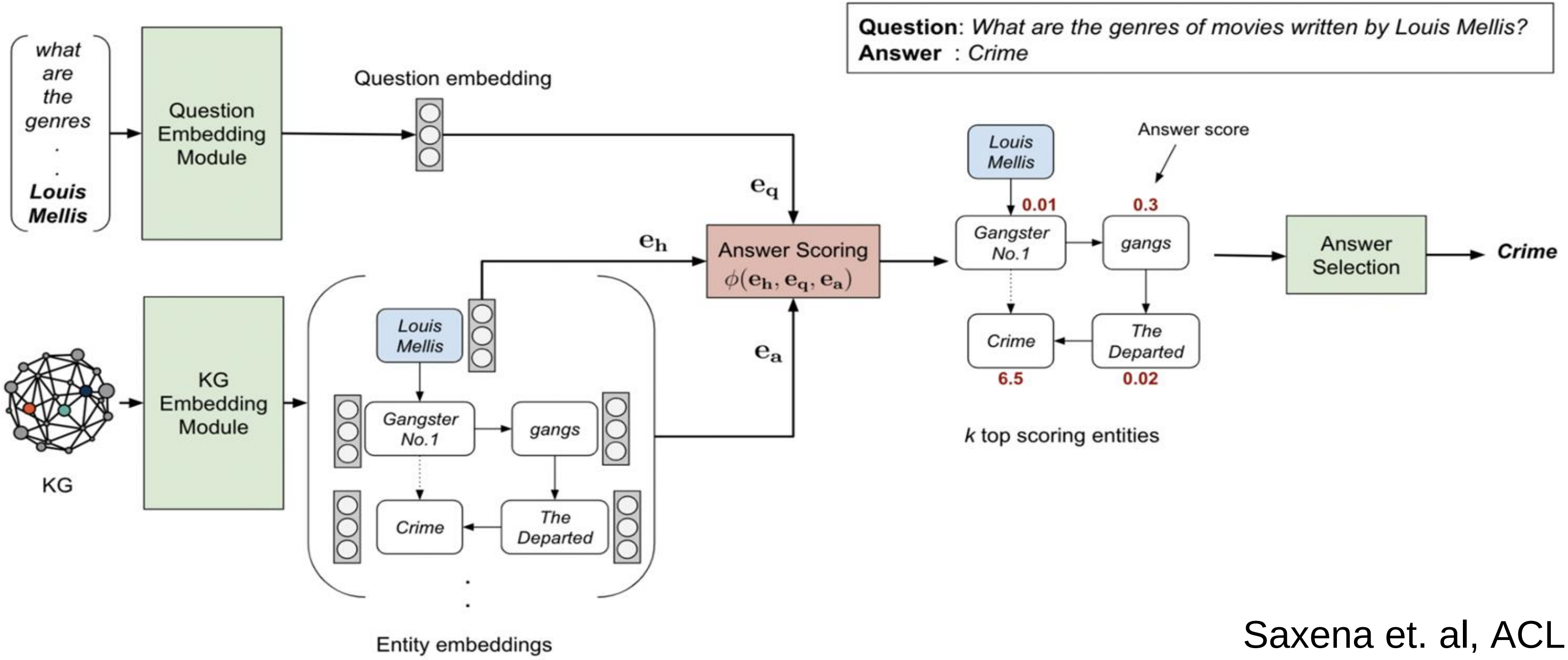
Answer: Crime

Question: *What are the genres of movies written by Louis Mellis?*

Note: KB=Knowledge Graph for us

Prior Approaches

- Get Question embedding and KG embedding (i.e entity and relation embedding)
- Score entities in KG and output answer



Saxena et. al, ACL, 2020

KGT5

- KBQA as a a Seq-2-Seq task → using a unified T5 model
- Pretrain on Link prediction → this helps learn the KG relations and entities
- Finetune on QA task → but without KG

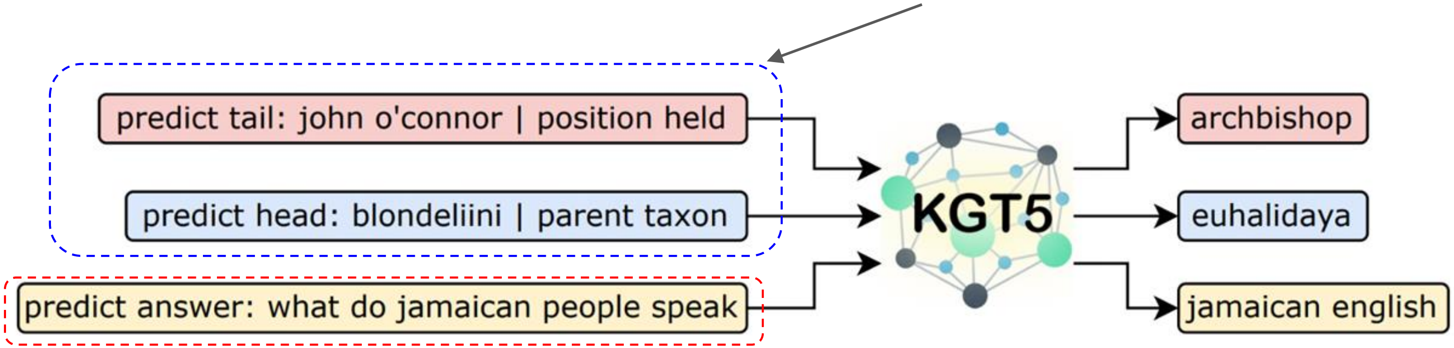


Figure 1: Overview of our method KGT5. KGT5 is first trained on the link prediction task (predicting head/tail entities, given tail/head and relation). For question answering, the same model is further finetuned using QA pairs.

Saxena et. al, ACL, 2022

KGT5

Finetuning

Pretraining (Note: start from untrained weights)

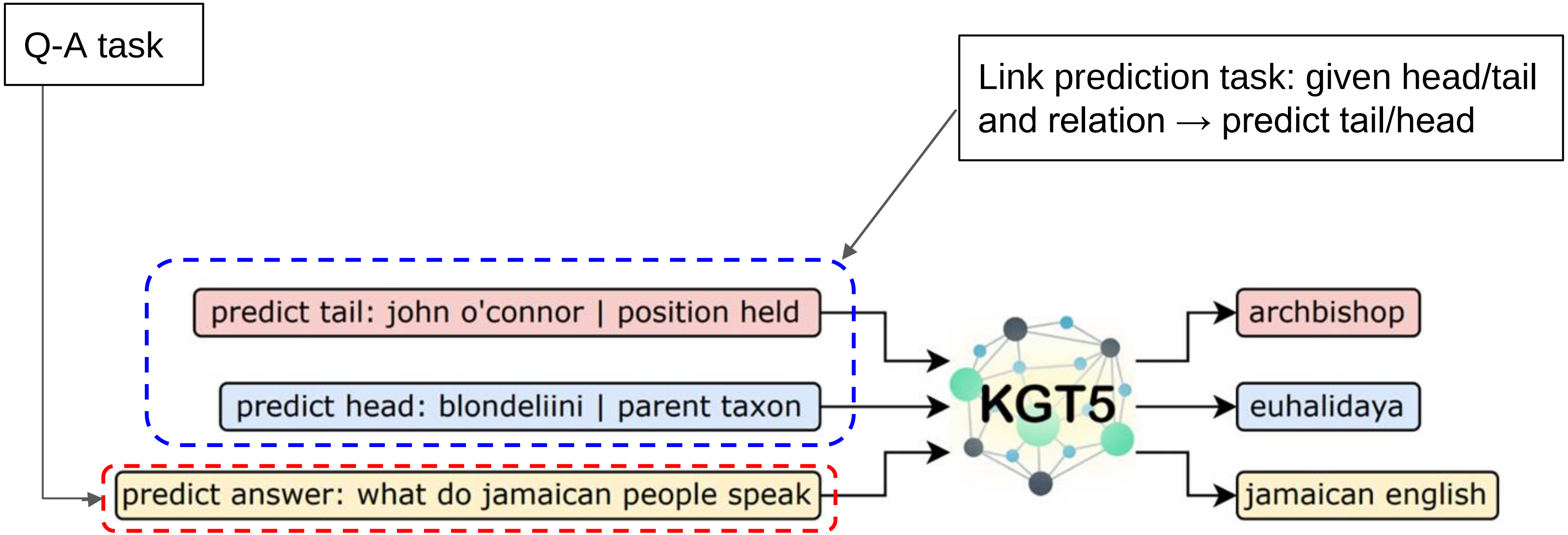


Figure 1: Overview of our method KGT5. KGT5 is first trained on the link prediction task (predicting head/tail entities, given tail/head and relation). For question answering, the same model is further finetuned using QA pairs.

Saxena et. al, ACL, 2022

KGT5

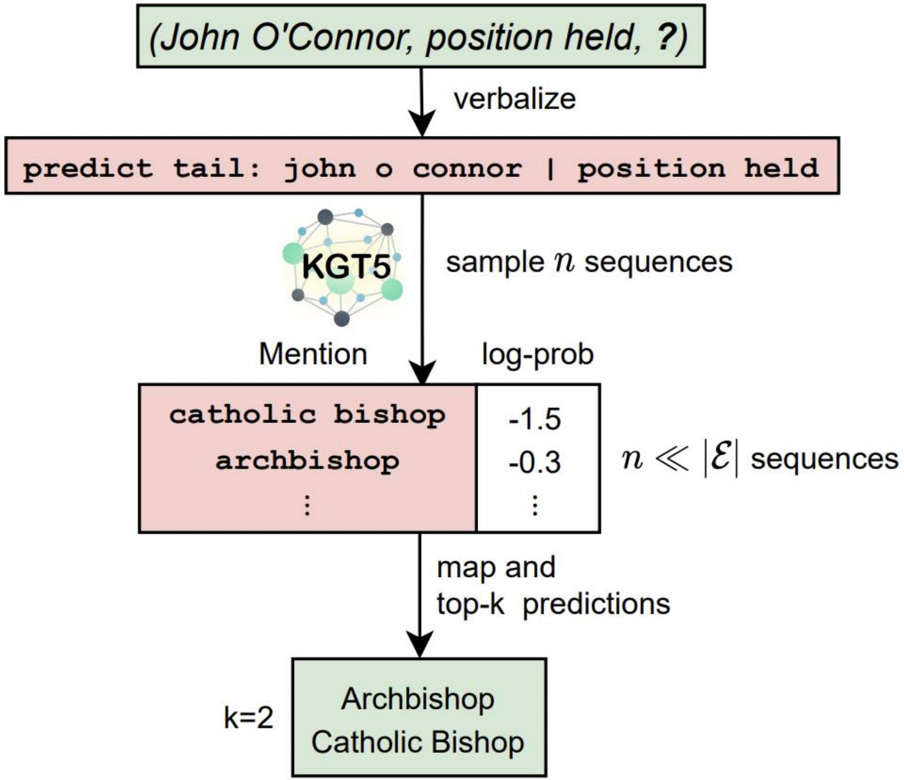
Link prediction

- Eg: given **(h, r, ?)** we need to find the tail, **t**
- **Sample n sequences** from T5 model
    - Sample from output probability of words (from T5)
    - Do this multiple times to get different outputs
    - Log-prob of any output entity=

$$\sum_{t=1}^{T} \log(\mathbb{P}(w_t|input, w_1, w_2, ..., w_{t-1}))$$

where $\mathbb{P}$ is the model's output distribution.

- Get Top - K predictions as answer



"model almost always outputs an entity mention"

Performance: Link prediction

Model	**MRR**	**Hits@1**	**Hits@3**	**Hits@10**	**Params**
TransE (Bordes et al., 2013) †	0.253	0.170	0.311	0.392	2,400M
DistMult (Yang et al., 2015) †	0.253	0.209	0.278	0.334	2,400M
SimplE (Kazemi and Poole, 2018) †	0.296	0.252	0.317	0.377	2,400M
RotatE (Sun et al., 2019b) †	0.290	0.234	0.322	0.390	2,400M
QuatE (Zhang et al., 2019) †	0.276	0.227	0.301	0.359	2,400M
ComplEx (Trouillon et al., 2016) $	**0.308**	**0.255**	-	**0.398**	614M
KGT5 (Our method)	**0.300**	**0.267**	**0.318**	**0.365**	60M
ComplEx 14-dim ‡	0.201	0.161	0.211	0.275	67M
ComplEx 26-dim ‡	0.239	0.187	0.261	0.342	125M
KEPLER (Wang et al., 2021) ††	0.210	0.173	0.224	0.277	125M
DKRL (Xie et al., 2016a) ††	0.160	0.120	0.181	0.229	20M
MLMLM (Clouatre et al., 2021) ‡‡	0.223	0.201	0.232	0.264	355M
KGT5-ComplEx Ensemble	**0.336**	**0.286**	**0.362**	**0.426**	674M

KGT5

Question Answering

Model	CWQ	WQSP
GT query	25.2	56.9
Pullnet	26.8 (+1.6)	47.4 (-9.5)
EmbedKGQA	-	42.5 (-14.4)
LEGO	29.4 (+4.2)	48.5 (-8.4)
GT query	24.5	56.9
KGT5	**34.5 (+10.0)**	**50.5 (-6.4)**

Results from 50% KG setting:
I.e Randomly drop 50% edges

Method	WQSP	CWQ
T5-small + QA finetuning	31.3	27.1
KGT5 (50% KG pretraining)	50.5	34.5
KGT5 (full KG pretraining)	56.1	36.5
EmbedKGQA	66.6	-
CBR-KGQA (Das et al., 2021b)	**73.1**	**70.4**

Results from Full KG setting:

Conclusion: T5 is good at generating the entities **not present** in KG (50 %), but bad at memorizing the KG entities (from 100% KG)

Comments + Papers/Pointers/Discussion:

Pros:

- It addresses the challenges through two key innovations:

  (i) relevance scoring, in which they employ LMs to determine the relative value of KG nodes in a specific QA context, and

  (ii) joint reasoning, where they connect the QA context and KG to form a joint graph, and mutually update their representations through graph-based message passing

- Good ablations
- Interpretable
- Method is quite general
- Outperforms baselines

Cons

1. The inherent scalability problem of GNN is an issue (Rocktim)
    a. Graph soft counter has some solutions
    b. Retrieve less number of nodes
2. The approach seems to be limited to MCQ questions (Rocktim)
    a. Use T5 kind of an approach to “generate answers”
3. Paper generates a sub graph G for each answer option and then does its predictions (Jai)
    a. The QA-GNN paper generates one graph (paths from Q entities to A entitites)
4. Unified QA and T5 beat QA GNN because of their size and amount of data trained. The paper could have done a study in which they increase their model size

    (Jai, Rohit)

    a. Yes!
    b. I believe QA-GNN was concerned about parameter efficiency (30x smaller than Unified QA)

Cons

5. They use different LMs for different data sets (Rohit)
    a. Possibly study: use same LM for all and compare
    b. Some intuition: AristoRoberta is finetuned on RACE dataset (reading comprehension): may have something common with OpenbookQA (also has extra science facts input, apart from the Q and A)
6. Choice of number of GNN layers not clear (Rohit)
    a. There are a few ablations w.r.t number of layers (acc increases then decreases as layers increase) → may or maynot be satisfactory
7. The baselines used for comparison are old (from 2018,2019) More recent baselines should be used for comparison. (Shivangi)
8. Test accuracy on MedQA-USMLE is marginal. More datasets from different domains can be used to check the generalizability and domain adaptation of the model (Shivangi)
    a. Yes in general there don’t seem to be many datasets having relevant KG+QA.

Cons

9. Graph connection to all nodes performs comparatively to just joining just the QA entities which shows that the edges do not have much relevance and only the node incorporated matters (Shreya)
10. Next sentence prediction NLI model instead of MLM model for relevance scoring (Vishal Saley)
    a. Possibly can improve since it would be able to better capture the entailment
11. Node relevance is definitely important but it does not protect against creating a partitioned sub-graph. Instead, path relevance could have been good measure and it is an straightforward extension of the proposed method. (Vishal)
    a. Something like what was done in the KagNet paper → maybe we can try it for QA GNN

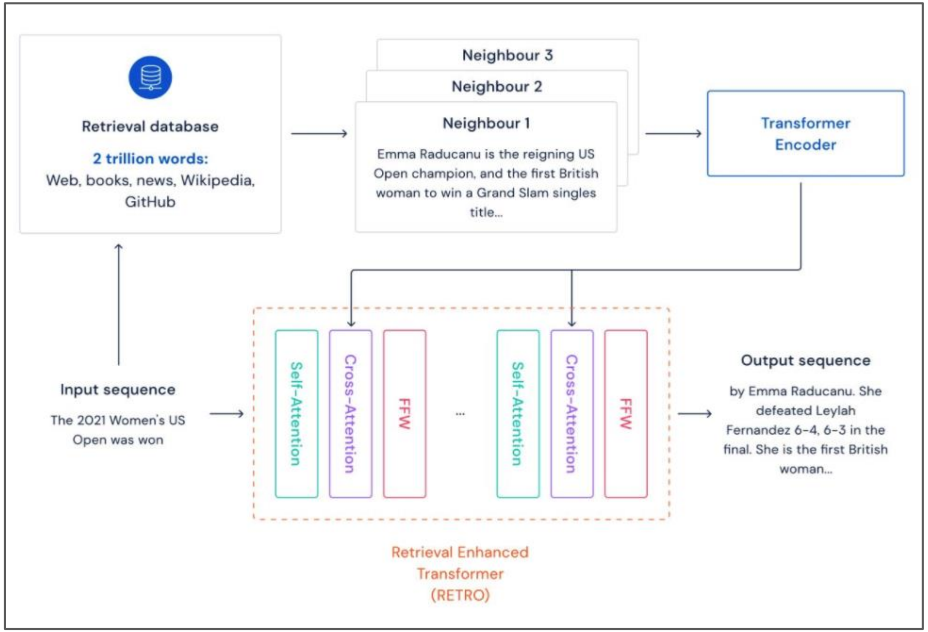
Extensions

1. Extending it to handle at least short answer-type questions/subjective QA where the answers are generated (Rocktim, Jai, Rohit)
    a. T5, GPT like model?
2. Question answering on a KG without Multiple Choices (Jai)
    a. Papers like EmbedKGQA (Saxena et. al, 2020) do this
    b. Can see the KBQA line of work
    c. Interesting method proposed by Jai (get entities, choose among them, also finetune LM → looks like EmbedKGQA and KGT5 type of method)
3. Double Negation (Rohit, Shreya)/Theorem proving using the explainability module (Rohit)
    a. Logical reasoning in NLP seems to be tough (there are a few datasets/papers like ProofWriter/Ruletaker which can be looked at)

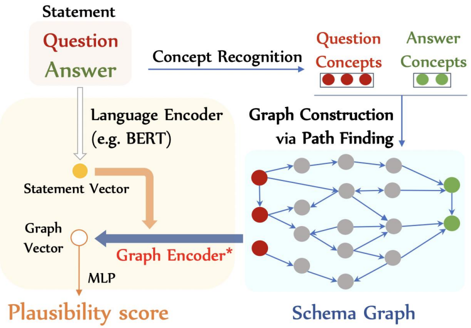
Extensions

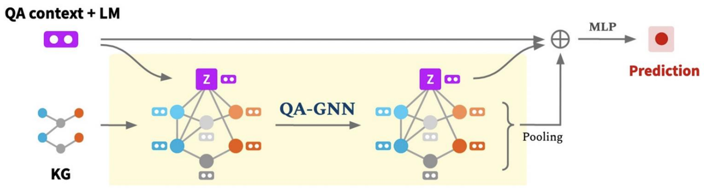
4. Similar methods can be applied in table understanding (Rohit)
5. Multimodal setting (Shivangi, Rohit)
    a. Images → scene graphs (symbolic!, objects and relation) → retrieve similar nodes from KG → Do QA
    b. Can see the GraphVQA paper
6. Given method is general enough to be extended to different reasoning problems. For example, in case of document grounded QA we can form contextualized representation for all the documents given a question. (Vishal Saley)
7. Model can be tested with other Language models other than Roberta (Shreya)

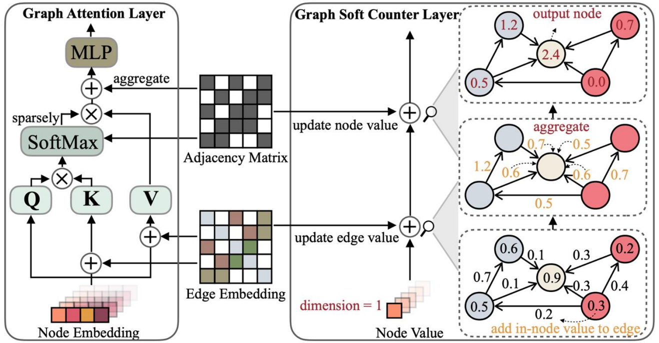
8. At some point in the future, we wouldn't need to train bigger language models but would need bigger knowledge bases, which can be updated each day easily
    a. Retro paper by deepmind?

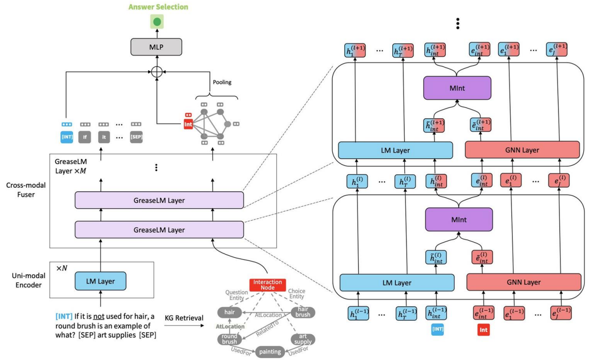


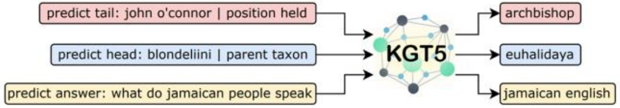
Thank you!


Statement
Question
Answer
Concept Recognition
Question Concepts
Answer Concepts
Language Encoder (e.g. BERT)
Graph Construction via Path Finding
Statement Vector
Graph Vector
Graph Encoder*
MLP
Plausibility score
Schema Graph


QA context + LM
KG
QA-GNN
MLP
Prediction
Pooling


Graph Attention Layer
MLP
aggregate
sparsely
SoftMax
Q
K
V
Node Embedding
Adjacency Matrix
Edge Embedding
Graph Soft Counter Layer
update node value
update edge value
dimension = 1
Node Value
output node
aggregate
add in-node value to edge


Answer Selection
MLP
Pooling
GreaseLM Layer ×M
Cross-modal Fuser
GreaseLM Layer
GreaseLM Layer
Uni-modal Encoder
LM Layer
KG Retrieval
Interaction Node
MInt
LM Layer
GNN Layer


predict tail: john o'connor | position held
predict head: blondeliini | parent taxon
predict answer: what do jamaican people speak
KGT5
archbishop
euhalidaya
jamaican english

References

1. Sequence-to-Sequence Knowledge Graph Completion and Question Answering - Saxena et. al, 2022
2. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering - Yasunaga et. al 2021
3. GREASELM: Graph Reasoning Enhanced Language Models for Question Answering - Zhang et. al, 2022
4. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings - Saxena et. al, 2022
5. Gnn is a counter? revisiting gnn for question answering - Wang et. al, 2022
6. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning - Lin et. al, 2019