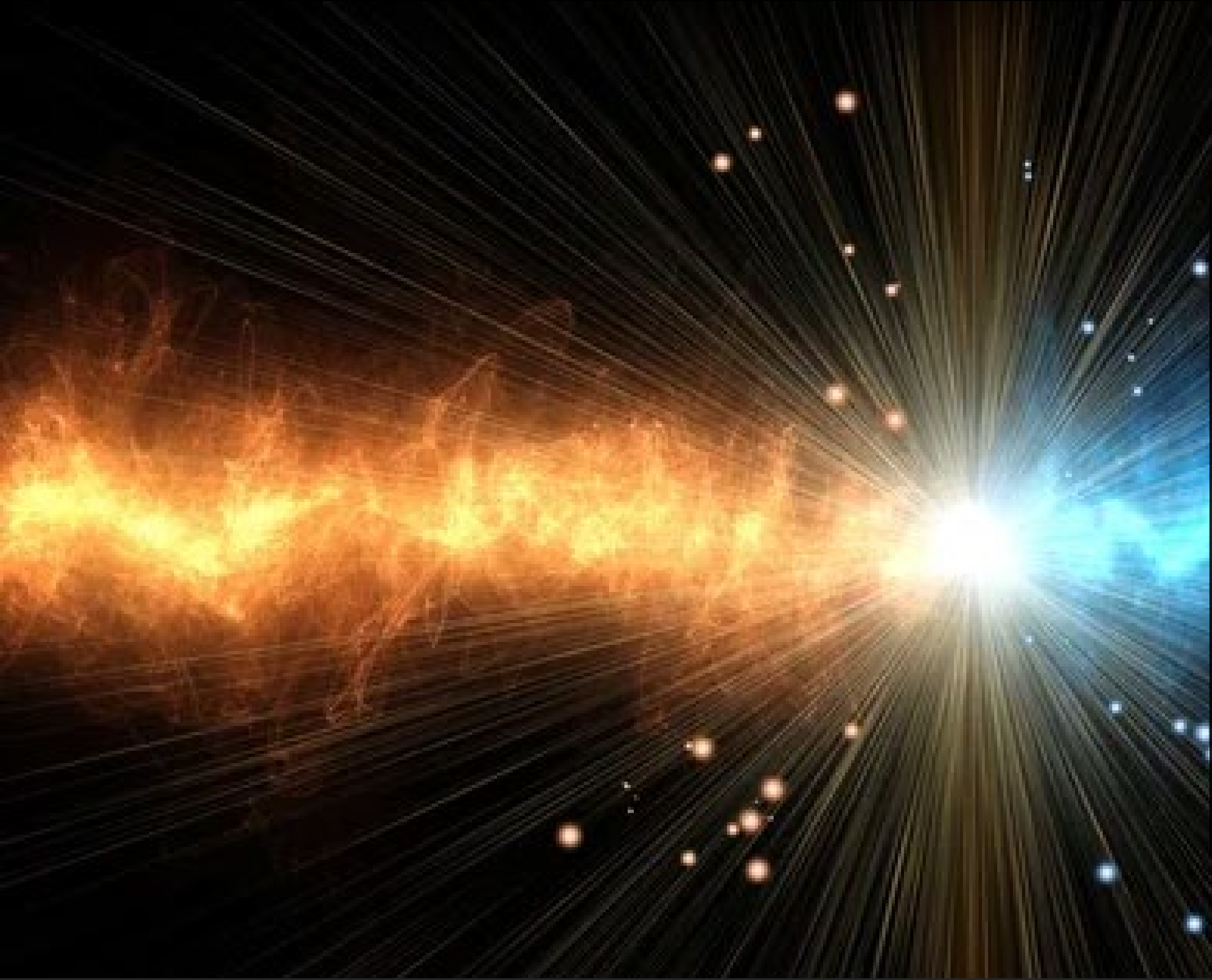


FusionMind QA-GNN: Improving question and answering using QA-GNN with external context fusion

CSE 6240 WSTM Project Presentation

TEAM 07:
MANOJ PARMAR, PALASH CHOUDHARY, SANCHITA PORWAL, SHREYAS VERMA

Introduction

Background

- ✓ The ability to reason and answer questions using observation and knowledge is a **crucial aspect of human intelligence**
- ✓ Answering questions using pre-trained language models (LMs) and knowledge graphs (KGs) **presents challenges in identifying relevant knowledge and performing joint reasoning**
- ✓ While LMs are effective in language understanding, **they suffer with handling knowledge and answering questions with underlying structural reasoning**. On the other hand, although KGs can encode topological information, **it lacks contextual information**

Approach

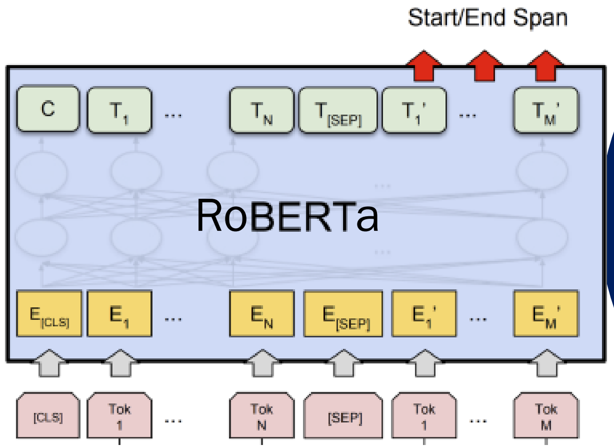
- ✓ In this project, we **built upon the previously published QA-GNN method**[1], which combines the strengths of LMs, KGs, and Graph Neural Networks (GNNs) for the Question-Answering (QA) objective
- ✓ Furthermore, we **experimented with improving the QA-GNN approach** by integrating additional information through the incorporation of knowledge facts from the data
- ✓ This enhancement aimed to **bolster** the model's performance in **answering complex questions** that necessitate a deeper understanding of the relationships between entities and contextual information

Impact

- ✓ By enhancing existing methodologies, our project has the potential to **advance the state-of-the-art question-answering** systems
- ✓ This, in turn, contributes to the development of more accurate and **context-aware AI applications**, benefiting virtual assistants, search engines, and educational tools.

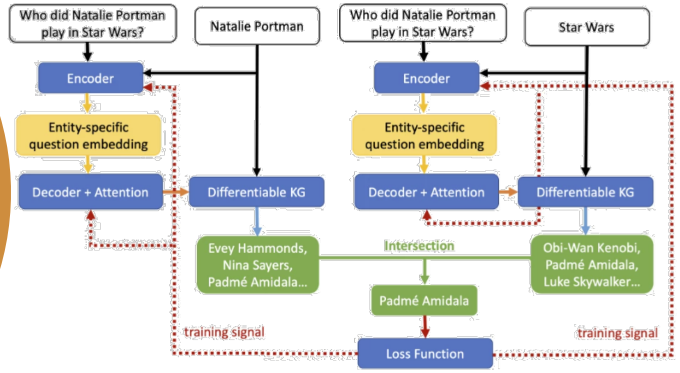
Approach overview

The 2 methods are: 1) Language models only & 2) Language models + knowledge graphs



Language models only

Language models + knowledge graphs (with and w/o context)



Data description and preprocessing

OpenBookQA

- ✓ Comprehensive set of **multiple-choice questions and related knowledge facts** to train and evaluate our model on elementary-level science question-answering tasks
- ✓ **Preprocessing steps**:
  - ✓ Convert the QA datasets into the required format and .jsonl files
  - ✓ Identify all mentioned concepts in the questions and answers to extract graph information for

ConceptNet

- ✓ **Structured semantic information** with relevant concepts and relationships to enhance our model's knowledge base
- ✓ **Preprocessing steps**:
  - ✓ Extract English relations from ConceptNet
  - ✓ Merging the original 42 relation types into 17 types

Data properties

We'll use the ConceptNet data for its knowledge graph and the OpenBookQA for its multiple-choice questions and related facts to train the language model

OpenBookQA

Important features:

- Each question comes with four possible answer choices, and the correct answer is labeled
- There are 1,326 core science facts
- Ground-truth labels: The correct answer choice is provided for each question

#Train	#Dev	#Test	Average Length of Q	Max token size for	
				Q + A	Q + A + Facts Context
4,957	500	500	~10.6 words	73	101

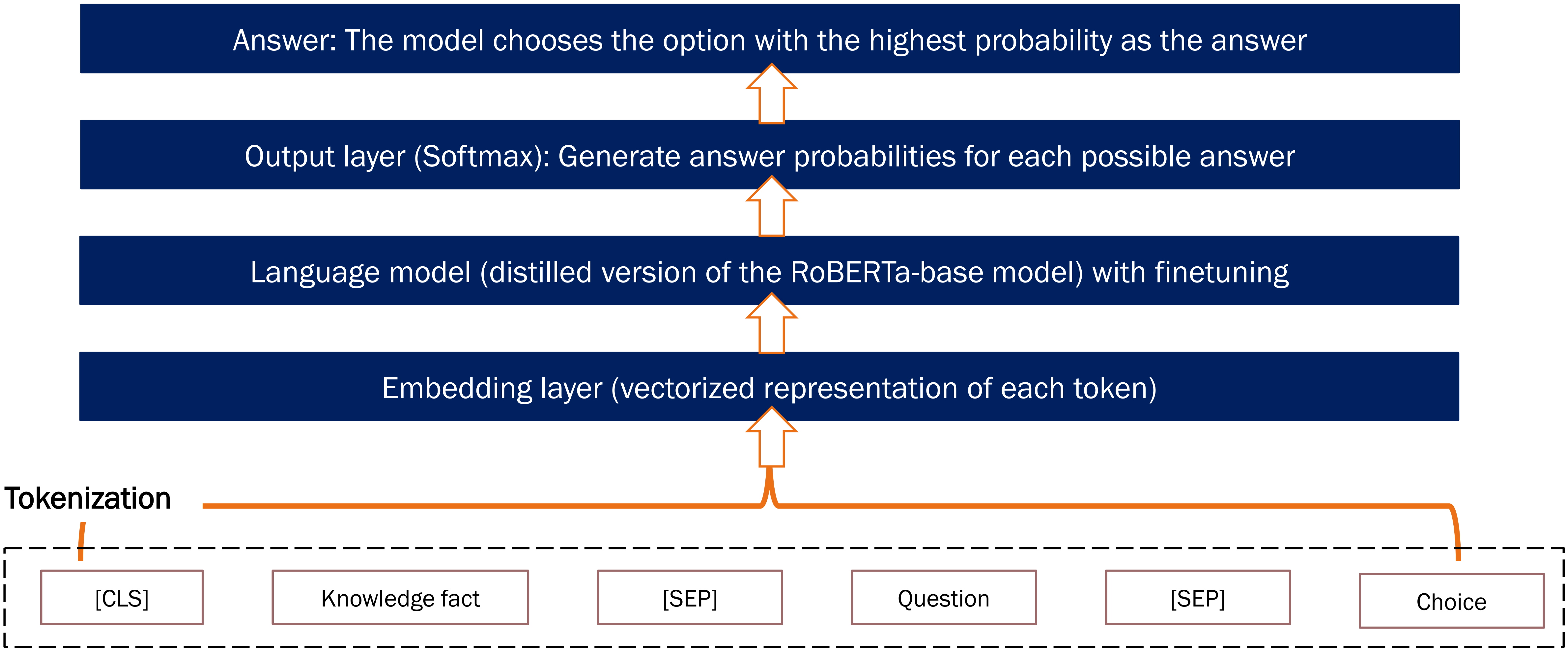
ConceptNet

Important features:

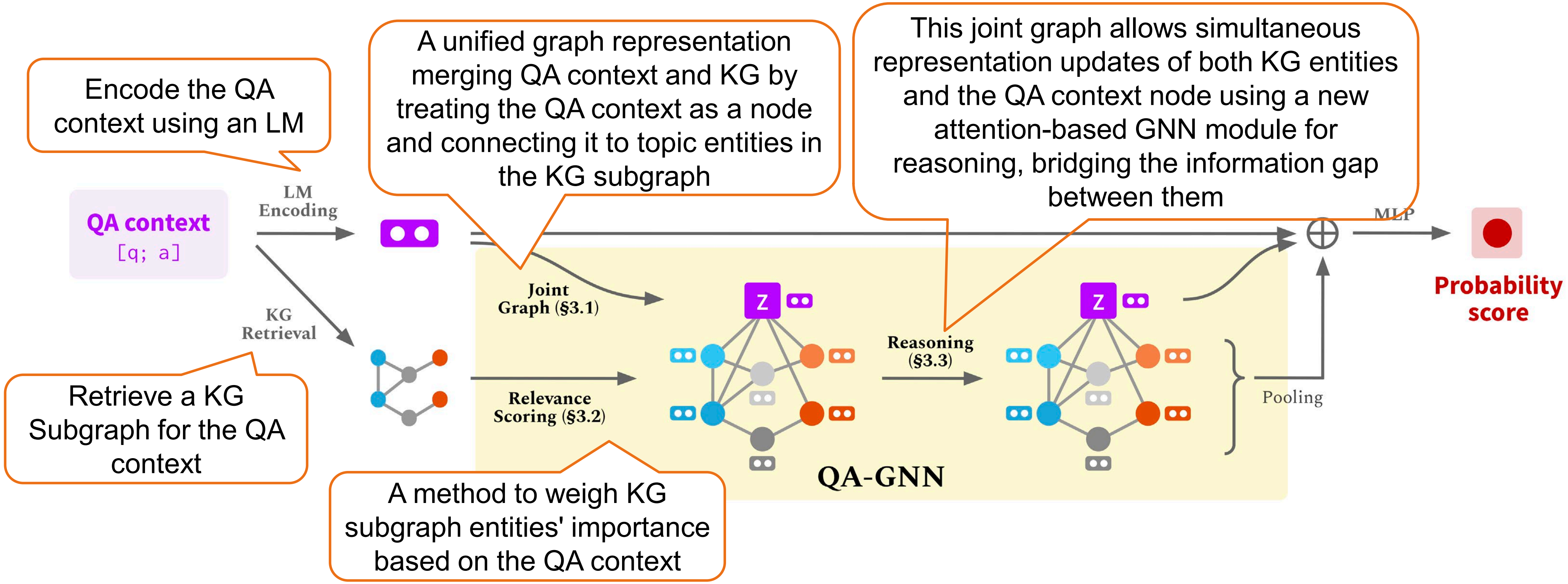
- Important features: Each concept and relationship has associated properties, such as weight and frequency
- Number of labels/classes: There are no explicit labels or classes in ConceptNet

#Nodes	#Edges	#Relationship Types	Most frequent relationship type
~ 784K	~ 4.3M	17	"HasSubevent"

Existing approach (1) deep-dive



Existing approach (2) deep-dive



We explored variations of this method by testing with and without supplemental context (scientific facts from the OpenBookQA dataset) as additional input to the QA context

Experiments and evaluations

Dataset

- ✓ Evaluating model performance on OpenBookQA dataset
- ✓ Dataset split: 4957 (train) + 500 (dev) + 500 (test)

Evaluation Metric

- ✓ Supervised QA (classification) task
- ✓ Using accuracy to compare different approaches

Hyperparameter Experimentation

- ✓ Architecture: usage of a final fully connected layer
- ✓ Number of neighbors for knowledge graph subgraph retrieval
- ✓ Usage of additional facts context

Training Environment

- ✓ Primarily using Google Colab and PACE COCICE clusters
- ✓ COC-ICE: 23 CPU nodes, multiple GPU nodes with RTX 6000 and Tesla V100

Results

LM only (finetuned on OpenBookQA)

Method	Accuracy	
	Dev	Test
DistilRoBERTa	0.536	0.522
DistilRoBERTa with context	0.658	0.670

LM (no finetuning) + KG

Method	K	FC Layer	Accuracy	
			Dev	Test
QAGNN	2	0	0.526	0.484
	2	1	0.534	0.498
	3	0	0.552	0.516
	3	1	0.546	0.486
QAGNN with Context	2	0	0.650	0.648
	2	1	0.636	0.640
	3	0	0.642	0.662
	3	1	0.652	0.676
	4	0	0.652	0.682
	4	1	0.669	0.694

Predictions

We observe how the usage of context improves the model's ability to reason in QAGNN

Question 1

Question

There is most likely going to be fog around:

A. A marsh B. A tundra C. The plains D. A desert

Context added:

Fog is formed by water vapor condensing in the air

Answer prediction

Without context: The plains

With Context: A marsh

Reason: The context helped the model understand that the answer entity was related to water

Question 2

Question

A positive effect of burning biofuel is:

A. shortage of crops for the food supply B. an increase in air pollution C. powering the lights in a home D. deforestation in the amazon to make room for crops

Context added:

Biofuel is used to produce electricity by burning

Answer prediction

Without context: D

With Context: C

Reason: Without context, the model is not able to understand the sentiment for the answer and gives the wrong answer.

Future work

We observe that supplemental context helps improve the performance even without finetuning the LM model

Additional context

Inspired by Yichong Xu et al[2], we would like to explore additional context such as Wiktionary data

To further refine using context

Effectively combine context

As many BERT based models have a limitation of maximum 512 input token size, we can experiment on how we can effectively combine different context together to be most informative

References

1. Michihiro Yasunaga et al. “QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering”. In: CoRR abs/2104.06378 (2021). arXiv: 2104.06378. url: https://arxiv.org/abs/2104.06378.
2. Yichong Xu et al. “Fusing Context Into Knowledge Graph for Commonsense Reasoning”. In: CoRR abs/2012.04808 (2020). arXiv: 2012 . 04808. url: https ://arxiv.org/abs/2012.04808.