Evaluating QA-GNN: A Deep Dive into Question Answering with Language Models and Knowledge Graphs

Hayden Kwok

June 2024

1 Introduction

Natural Language Processing (NLP) is a sub-field of Artificial Intelligence (AI) that focuses on developing computer technology to comprehend, interpret, and generate human language, enabling machines to interact with and process natural language data. The branch utilizes a large span of techniques and algorithms that are applied in tasks such as machine translation, sentiment analysis, as well as text summarization. Here, I will be analyzing a particular approach within the subtopic of Question Answering (QA), noting its functionality, and exploring its limitations.

At a high level, the task of QA involves the development of systems and algorithms that automatically generate responses to user-generated questions in natural language. The objective of these algorithms is to effectively retrieve precise information within prompts to facilitate efficient interactions between humans and computers. This will foster improvements in access to knowledge for users, such as enhanced tutoring services, customer support, as well as the accuracy of LLMs.

I will be analyzing the QA-GNN model from the paper QA-GNN: Reasoning with language models and knowledge graphs for question answering (Michihiro Yasunaga, 2021), which is trained from existing language models (LMs) and knowledge graphs (KGs). The model attempts to address two major challenges: extracting pertinent knowledge from large KGs and jointly reasoning over the QA context and KG. QA-GNN links the QA context and KG to construct a joint graph, updating their representations through graph neural networks (joint reasoning), hence the GNN portion within the name of the model. In addition, it uses LMs to evaluate the importance of KG nodes in the provided QA context (relevance scoring). This combination of techniques, outlined in Figures 1 and 2, demonstrates the paper's approach to openly answering common-sense questions, as well as responding with prompted answer choices.

The associated codebase for this model can be found on GitHub in the directory hakwok/qagnn-Analysis-Study (Kwok, 2024) which is my forked and edited version of the original michiyasunaga/qagnn repository (Yasunaga et al., 2021). The QA-GNN model utilizes Transformers, PyTorch, Numpy, and Tqdm libraries implemented in Python scripts to facilitate its functioning.

While the original paper evaluates the model using Exact Match (EM) and F1 Score metrics, this analysis will employ Semantic Answer Similarity (SAS) with manual comparison oversight, as well as a big picture overview of the test/train accuracy to address potential limitations in the evaluation approach. I chose these forms of evaluation because EM and F1 scores may underestimate the model's performance due to a lack of semantic examination between ground truth and predicted answers. Furthermore, the concept of human-computer interaction encourages the insight of a "user" which in this case would be my own interpretations of the model's decision making.

2 Dataset

I plan to evaluate the QA-GNN model on two datasets: CommonsenseQA (CSQA) and OpenBookQA (OBQA), each accompanied by their corresponding knowledge graph datasets that are embedded within the repository. At a glance, CSQA comprises questions requiring commonsense knowledge to answer, where each question is linked to concepts from a commonsense knowledge graph. OBQA, on the other hand, evaluates open-book question-answering abilities by testing the model's capacity to combine scientific facts with commonsense knowledge for answering

multiple-choice questions.

CommonsenseQA (CSQA) consists of questions that demand commonsense reasoning to select the correct answer from multiple choices. Each question is associated with a commonsense knowledge graph. CSQA contains around 12,102 questions, with an average question length of 12 words and an average of 5 choices per question. Questions in this dataset often require reasoning beyond literal text comprehension, necessitating the integration of external knowledge.

OpenBookQA (OBQA) evaluates the model's ability to answer questions by combining scientific facts, or "open book", with commonsense knowledge. It comprises approximately 5,957 questions, with an average question length of 18 words and an average of 5 choices per question. OBQA questions require the integration of domain-specific scientific knowledge with commonsense reasoning, making it a challenging task for models.

These datasets offer diverse question types and require models to integrate information from various sources, including text and external knowledge, making them challenging benchmarks for evaluating QA systems, which could potentially explain the sub-optimal accuracy touched upon later. Here are examples of input/output pairs from both datasets:

CSQA:

- Input: "What might someone be using if they're putting it on a table?"
- Choices: A) food, B) clothes, C) water, D) dishes, E) flowers
- Output: D) dishes

OBQA:

- Input: "An object accelerates when"
- Choices: A) it gains mass, B) its mass increases, C) its speed increases, D) it moves towards the earth
- Output: C) its speed increases

3 Analysis Approach

Running the model was pretty straightforward as I followed the steps within the original documentation. I began by first modifying the dependencies to fit my hardware, such as changing the PyTorch/CUDA version and refractoring parts of the

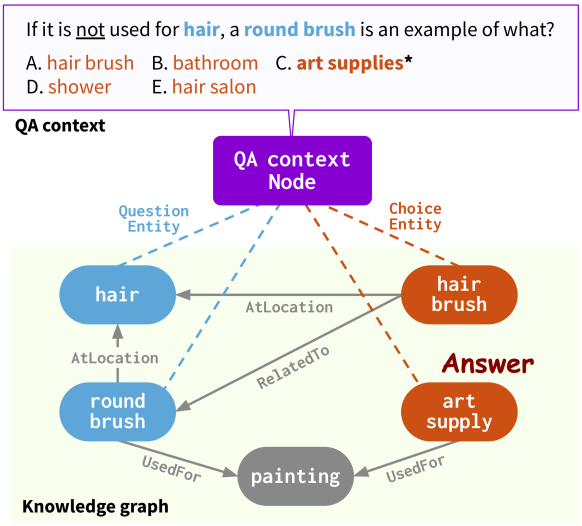

Figure 1: QA context and knowledge graph

Figure 2: QA-GNN Model

codebase. I then went on to download the CSQA and OBQA datasets and preprocess them through a python script. It began by extracting English relations from ConceptNet and merging the original 42 relation types into 17 types. Then, it goes on to convert the QA datasets into .jsonl files and identify the subsequent concepts mentioned in the questions and answers. Lastly, it extracts the subgraphs for each question-answer pair.

Afterwards, I trained and evaluated the two respective models with the default hyperparameters to ensure consistency.

CSQA:

- epochs: 15
- batch size: 64
- mini-batch size: 2
- GNN layers: 5
- encoder learning rate: 1e-5
- decoder learning rate: 1e-3
- number of relations: 38 (originally 17, add 2 relation types, and double because we add reverse edges)

- hidden dimension: 200
- number of unique concepts: 5

OBQA:

- epochs: 100
- batch size: 128
- mini-batch size: 1
- GNN layers: 5
- encoder learning rate: 1e-5
- decoder learning rate: 1e-3
- number of relations: 38 (same reasoning as CSQA)
- hidden dimension: 200
- number of unique concepts: 4

The analysis of the QA-GNN model involves several methods aimed at identifying its limitations and areas for improvement. One significant aspect of my approach is the utilization of alternative evaluation metrics, specifically Semantic Answer Similarity (SAS), in addition to the existing metrics of Exact Match (EM) and F1 Score used in the original paper. Since the original paper had already touched upon the EM and F1 metrics, this analysis will deviate from the consideration of those metrics and focus primarily on SAS and manual oversight. However, I will still use those metrics to help categorize the errors.

Semantic Answer Similarity (SAS) allows for a more nuanced evaluation by measuring the semantic similarity between the predicted and ground truth answers. This approach provides insights into the model's understanding of the questions and its ability to generate semantically relevant answers. This addresses a potential limitation of EM and F1 scores, which may overlook variations in answer phrasing. As a result, I wrote a series of Python scripts using BERT to retrieve SAS values from both datasets and plot them in distribution histograms.

By looking over the differences manually, we can better understand the subjective context that can be taken into account for the decisions the model makes in selecting its answers. In this way, we can analyze the ethical, and potentially dangerous, outcomes that may come out of error-prone QA models in deeper detail.

An interesting aspect of this approach is the emphasis on evaluating not only the accuracy of the model's predictions but also its understanding of the underlying semantics and its ability to rank answers based on relevance. By adopting a multi-layered evaluation strategy, I aim to gain a more comprehensive understanding of the QA-GNN model's strengths and limitations, ultimately facilitating informed decisions for its improvement and optimization.

4 Errors and their Categorization

The model yielded a train accuracy of 77.15%, and a test accuracy of 74.05% for the CSQA dataset. For the OBQA dataset model, the training accuracy was 69.60% and the test accuracy was 69.00%. This indicates that the model is not intensely overfitted nor is it trained to its peak capabilities. In the following subsections, I will go over a series of incorrectly classified errors and their subsequent statistics.

Since the dataset achieves normality as a result of CLT and LLN, I will only be manually assessing a 10% sample of the incorrect answers in light of the scope and time frame of this project. When doing so, I considered the differences in SAS to gauge the semantic errors of the model in addition to my manual oversight.

Overall, 321 of the 9741 prompts were incorrect in the CSQA dataset, and 154 was incorrect out of the 500 in the OBQA dataset. I took a random sample of 32 responses in the CSQA dataset and 15 in the OBQA dataset.

In addition, during my manual oversight, I came across a sizeable amount of grammar and typo mistakes within the prompts itself which could be a contributing factor to the errors.

CSQA:
Average SAS Score: 0.9291
Average of Wrong SAS Score: 0.7261
OBQA:
Average SAS Score: 0.9090
Average of Wrong SAS Score: 0.7053

After manually reviewing the sample, I sorted each into four categories of errors: ambiguous priority, verbal reasoning, semantic misunderstandings, and reversed logic.

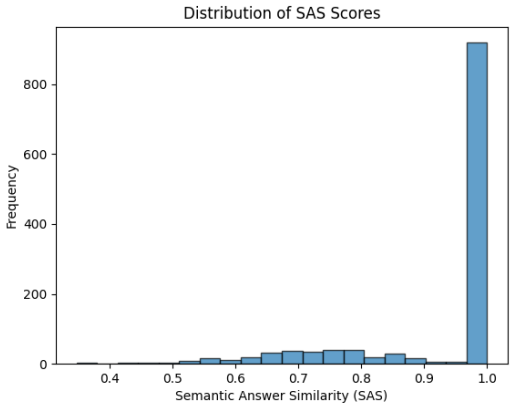

Figure 3: CSQA Distribution of SAS Scores

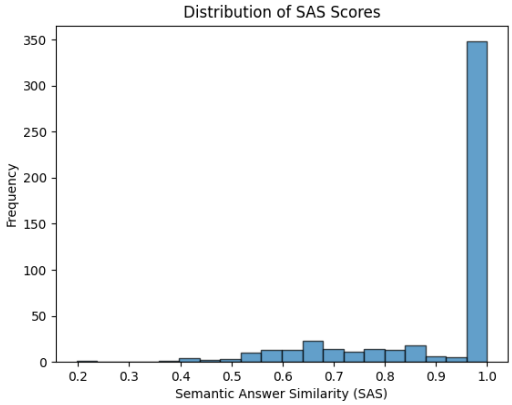

Figure 4: OBQA Distribution of SAS Scores

4.1 Ambiguous Priority

This section featured answers by the QA-GNN model that did not equate to the ground truth but were in a sense potentially correct. For example, one of the prompts in the CSQA dataset that came up was ”What do humans take in while breathing?”. The ground truth answer was ”oxygen” but the model chose ”air”. Technically speaking, humans do in fact breathe air, however, oxygen was the *better* response as it is a more accurate choice of the specific type of air we breathe. Therefore, the model does not choose the answer with the highest priority but still opts for a debatably reasonable choice. Overall, this category was the most frequent and featured the highest min, max, and median SAS values.

4.2 Verbal Reasoning

These errors were very similar to that of ambiguous priority in that the answer that the model chose was correct to a degree but could have intuitively reasoned to choose the better answer given the context. For example, in the OBQA dataset, one of the questions was ”Where would you only need a shopping bag if you did not finish what you got there?”. The ground truth answer was ”restaurant” but the model opted for ”supermarket”. Technically speaking, the model isn't entirely wrong for choosing a supermarket since a shopper wouldn't necessarily need a bag. However, with the context of the word ”finish”, one would be able to deduce that the restaurant would be the better choice as people who do not finish their food would opt to ask for a to-go bag. This subsection featured the largest spread in SAS score.

4.3 Semantic Misunderstandings

These answers were just flat-out wrong in terms of classifying the answer choices in relation to the discussed subject and likely misidentified the semantics of either the question or answer choices. In the OBQA dataset, one of the questions was ”What is used for sensing visual things?” in which the model answered ”tibia” instead of the Ground Truth Answer, ”cornea”. These errors often exuberated confusion and were often unrelated in terms of answer choices which led to its spread being concentrated at lower SAS values.

4.4 Reversed Logic

Lastly, this category was often rare and essentially meant that the model had answered the question as if it were the opposite expected result. In the CSQA dataset, one of the prompts that came up was ”Bill did not abandon the fight, but did what to the enemy?” which featured a ground truth answer of ”engage”. On the contrary, the model answered with ”embrace” which is completely the opposite of ”engage” in terms of semantics and what Bill was intending to do (continue fighting as he did not abandon it).

5 Discussion

The previous plots present insight into the potential reasoning to which the model fails on select categories in terms of SAS values. One major observation is the overall higher concentration of the spread of SAS values for the errors found in ambiguous priority indicates that the model often comes very close to choosing the ground truth answer. The chosen answer and the ground truth answer share a high semantic similarity, which tells

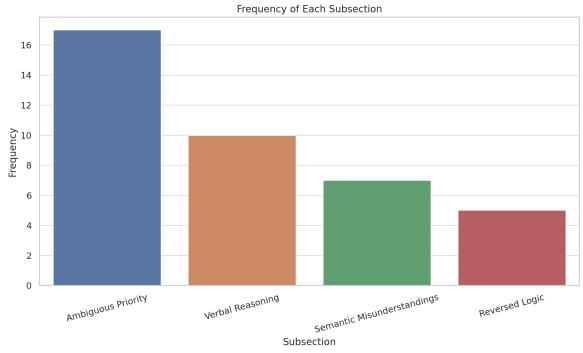

Figure 5: Frequency of Subsections

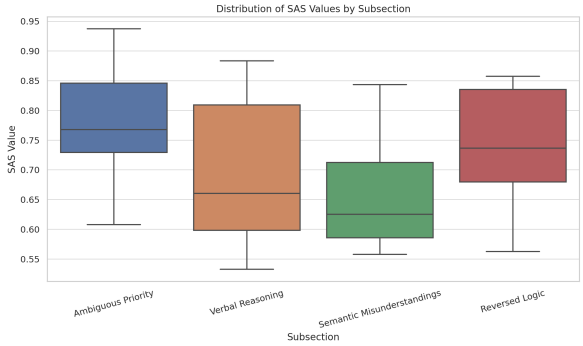

Figure 6: Distribution of SAS each subsection

us that the meaning behind the answer choices are very closely aligned (i.e. ”ocean” and ”river” from the question ”Where would using a boat require navigation skills?”) and that it might just be coming down to missing context within the question to help the model make a more accurate decision. However, without knowledge of the ranking scores, it would be quite difficult to know if the model consistently chose the second-best answer for this category and should be something to analyze in future work.

A special case for this category was the question ”What causes someone to laugh from something surprising?” to which the model responded ”accidents” instead of ”funny”. This darker response to the prompt could touch on the ethics side of NLP, as the reality of the answer choice is that many may reflect the choice of the model but ethically it is the incorrect choice as it promotes negative sentiment. An interesting approach to filtering out these special cases may lie within the preprocessing of the datasets, which could better rank the more ethically challenging questions to fit those of choices with higher positive sentiment.

The verbal reasoning category featured pretty straightforward errors but a questionable spread in SAS values. Highlighting the discrepancy between the contrasting values of SAS may outline the crux of the issue. The minimum SAS achieved, 0.5213, had the question ”What explains the characteristic lunar formations?” in which the ground truth and chosen answers were ”many collisions that have occurred” and ”remains of ancient ponds”. At first look, the two answer choices have very little semantic similarity and may appear unrelated. However, if we dive into the question we will come to realize that ”remains of ancient ponds” may have some scientific reasoning to be utilized in predicting the lunar formations in previous eras. Realistically, they should have been able to intuitively work out the idea that ”many collisions that have occurred” was the more appropriate answer; this may spark the idea of a future work that analyzes the Semantic similarity between the question and both the ground truth and chosen answers, in relation how the model determines its ranking.

On the contrary, the prompt with the highest SAS, 0.8832, featured the question ”What do people look for when competing against someone?” which was answered with ”skill” instead of the ground truth ”edge”. The two answer choices are very related in that they are both traits of an opponent, but realistically the model should reason that the *player* would be more likely to desire an edge on their opponent to seek victory instead of skill from their opponent to seek competitiveness. These kinds of errors could be attributed to similar reasons as that of ambiguous priority and should be addressed with ranking described in the first example of verbal reasoning.

Moreover, the prompt may be a major factor in the output of the model in Verbal Reasoning answers. For example, one of the prompts was ”The fat man refused to accept what was possible, he complained that he what the simplest activities?” in which the model responded with ”no go”. For context, the correct answer was ”can’t do” and at a glance, the answer was supposed to replace ”what” in the prompt to complete the sentence. Prompts like these from the CSQA dataset may confuse the model as it had been previously trained primarily with direct question-answer prompts instead of one that requires more intuitive thinking.

The Semantic Misunderstandings category was the hardest to identify the root cause of the issue

as most question-answer pairs were quite different and confusing. For example, the question "The animals were not thirsty, so what did they do when they were by the river?" was answered with "fly" but in reality should have been "pass water". This example shows that the model may be misunderstanding the meaning of "pass water", and as a result, ranking it lower and choosing "fly" instead. Therefore, it would be interesting to see what the SAS between the question and the ground truth answer would be in relation to that of the chosen answer to clearly identify the root of the misunderstanding.

The reversed logic questions often resulted in the model choosing the choice that would be completely incorrect (almost directly converse) but was in a sense related to the question topic as if the logic of the question was *not* (i.e.) "choose the best answer". Therefore, combining the contrasting elements of opposing logic and closely related concepts, the SAS scores were often concentrated on the higher end but could potentially drop all the way to approximately the 60% margin. These results could be attributed to a misunderstanding of logic from the questions if they were potentially worded unconventionally. For example, "Bill's arm got cold when he put it inside the" was one of the questions from the OBQA dataset which had a ground truth answer of "refrigerator". The model responded with "jacket" which has the opposite logic as "refrigerator". Perhaps, one of the words had been improperly tokenized, such as that of "when" or "cold" and instead gave it a semantic of something along the lines of "therefore" or "warm".

6 Architectural Improvements

In light of these findings, it is crucial to the concept of human-computer interaction that the model is able to accurately answer prompts from humans regardless of the clarity of input. In order to improve the QA-GNN model to further its progress in achieving this reality, I propose three essential steps to address the previous errors: improving the data preprocessing, improve ranking, and implementing a Natural Language Inference model.

6.1 Detailed data preprocessing

Improved data cleaning (reduction of typos and improved clarity of questions) could potentially lessen the ambiguity of answer choices and reduce the confusion in the model's ability to rank answer choices in relation to the prompt. However, this is something to lightly consider as although the model would be fed more clean data, realistically, most applicable scenarios could see humans incorrectly inputting their data (i.e. a student misspells a few words in the prompt). Therefore, it may be wise to stray away from this as it is essential to the advancement of the model's predictive abilities in a more natural manner, but if quick and efficient improvements to the model are desired then this could be an easy fix.

6.2 Ranking using Neural Networks

For the Ambiguous Priority and Verbal Reasoning categories, it would be best to first analyze the SAS between the question and answers to see if the model is properly recognizing similar semantics. Afterward, employing measurements of MMR (ranking the answer choices and training the model to re-rank their answer choices based on previous rankings of questions with similar SAS) would address the tighter errors that were primarily failing to choose the *best* answer. This can be implemented using a neural network as it can best reflect the continuous re-ranking based on previous weights and biases.

6.3 Natural Language Inference

The other errors besides the Ambiguous Priority and Verbal Reasoning errors represented a much lesser portion of the mistakes and therefore are much harder to address. However, one way to attempt to fix the issue would be to add confirmation of understanding of the semantics through the implementation of the Natural Language Inference (NLI) model. This will verify if the model's predicted answer follows the logic of the question and, after combining with the QA-GNN confidence score with weighted-sum averaging (or potentially even a small neural network), the model can more accurately select its answer.

7 Conclusion

Throughout the course of this analysis study, I have identified existing errors in the QA-GNN model, including logic-based misunderstandings and improper reasoning for the ranking of answers. Future works may see the implementation of the concepts mentioned in the way forward, further improving the model and approaching the

goal of safe question-answering in the realm of human-computer interaction.

8 Acknowledgements

I utilized Github Co-pilot to assist in resolving dependencies when running the model. Overall, the outputs were not really changed, except for choosing specific versions suitable for my device. Within the report, I used ChatGPT to format certain aspects of the LaTeX document, such as the imports of the images and proofreading. Here, the outputs of the imports had little to no change except for the input of specific fields. For the proofreading, there was little change to the output as it mainly corrected grammar errors from the prewritten text I wrote and only served as a minor enhancement of my writing; I also restructured it to better flow with the paper.

References

Kwok, H. (2024). Qa-gnn analysis study. https://github.com/hakwok/qagnn-Analysis-Study.

Michihiro Yasunaga, Hongyu Ren, A. B. P. L. J. L. (2021). Qa-gnn: Reasoning with language models and knowledge graphs for question answering.

Yasunaga, M. et al. (2021). Qa-gnn: Reasoning with language models and knowledge graphs for question answering. https://github.com/michiyasunaga/qagnn.