Team name: Little NLP learners

Project Proposal

Yuting Deng (yutingde@andrew.cmu.edu) Guoyao Li (guoyaol@andrew.cmu.edu)

Huiyi Zhang (huiyiz@andrew.cmu.edu) Eric Li (deyil@andrew.cmu.edu)

1 Technical Description

1.1 Question Generation

EQG: Ensemble Question Generation System

Input of our system

An context article and N (the number of question required to be generated)

Output of our system

Top N questions generated from the input text, half are W/H questions and half are T/F questions.

For W/H questions

Our W/H question model has an ensemble of two channels, which ensures that we can generate enough answerable questions and reduce the deviation caused by potential answer detecting.

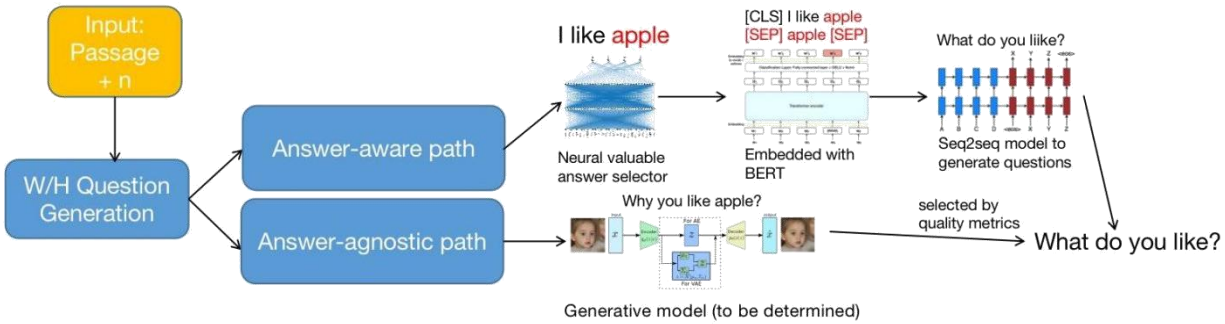

Figure 1. EQG architecture

First Channel: Answer-aware

We will build a potential answer detector to detect and label the potential answers existing in the paragraph. We intend to build the detector with a relatively shallow neural network. After labeling the potential answers. We will embed the context and answers using a pre-trained model(BERT), which will transform the tokens into a latent space to which ML models are easier to apply. Then, we will pass the embeddings into neural networks, thus generating questions.

Second channel: Answer-agnostic

We will make use of the Answer-agnostic question generator which has long been investigated. The advantage of the second channel is that it would not be affected by the error produced by detector(in channel 1). Through ensemble, we can help the system not be subject to the error in the detector, thus increasing the performance of the whole system.

At last, we will propose a novel discriminator, which allows us to pick out the best N/2 questions generated from the paragraph.

Yes/No-questions

The basic idea for Yes/No-questions is the same as W/H questions. We will leverage the power of ensemble models to generate both high-quality and diverse Y/N questions. The passages will first go through a pre-processing stage, such as constituency parsing. Then we will use MLP to select question-worthy contents. Next, these contents will go through both a template-based framework and a generative framework to generate n binary questions. At last, we will use statistical methods to select top n/2 questions from it.

Technical Tools

We plan to use Pytorch to build our novel model. Pytorch is the most frequently used deep learning framework in machine learning research, supporting multiple kinds of operations. We will make use of BERT pretrained models from the open source package transformers. We plan to do data pre-processing using the library of NumPy.

Moreover, we will utilize the GPUs in AWS and Google Colab to apply training and fine-tuning of our model. GPUs can significantly accelerate the computation of tensors and matrices, thus will greatly enhance the development of our new model.
About the management of code, we plan to use Git and GitHub.

1.2 Question Answering

Baseline Model

For question answering, we plan to build our model upon a generative model FID (Fusion-in-Decoder) [1]. The FID approach is divided into two parts, support passage retrieval and text generation through a sequence-to-sequence model. It uses BM25 or DPR (Deep Passage Retrieval) to retrieve support passages. Then it takes the pre-trained T5 model from Huggingface's transformer library and fine-tunes it at each dataset. In our case, we will fine-tune it on the Wikipedia dataset to minimize the cross-entropy loss for generating answer tokens from the decoder.

The encoder will process the concatenation of each support passage, its title, and the input question independently, with special tokens inserted. The decoder then takes the concatenation of all representations of support passages and performs attention to generate the answer sentence.

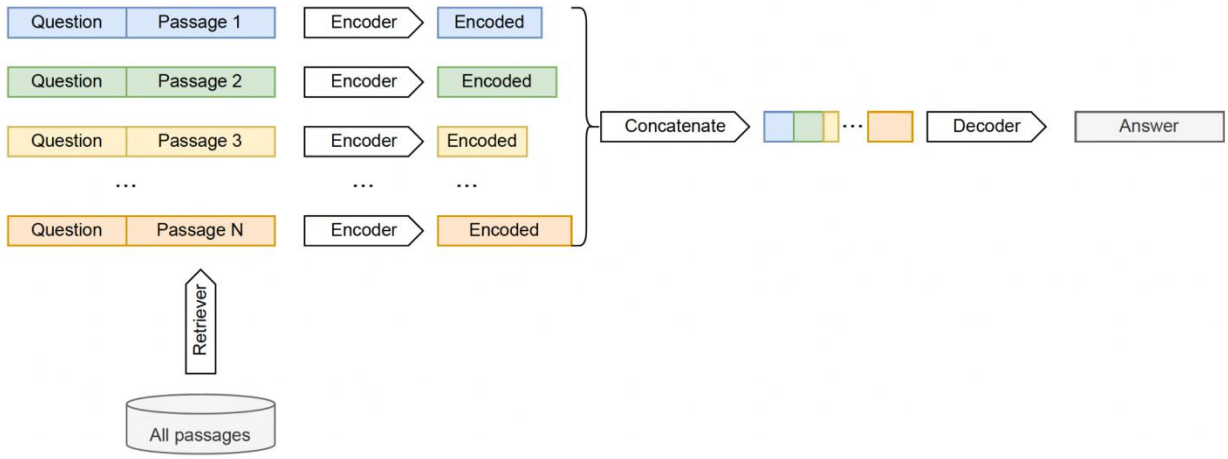

Figure 2. FID architecture

Interpretability

Our main goal is to interpret the FID model using post hoc explanation methods. Details are introduced in section 2.

2 Innovation

2.1 Question Generation

Improve run-time efficiency

In the paper of BERT for Question Generation[7] the authors use BERT sequentially, which means multiple BERT models need to be trained, which greatly increases the training time. With our ensemble approach, we do not have to fine-tune multiple BERT models at a time, which will greatly save run-time.

Improving performance: through 2-channel ensemble

Our W/H question model will have an ensemble of two channels, which ensures that we can generate enough answerable questions and reduce the deviation caused by potential answer detecting.

The first channel is answer-aware. We will build a potential answer detector to detect and label the potential answers existing in the paragraph. The second channel is answer-agnostic. We will make use of the answer-agnostic question generator.
Combining the advantages of the two channels, we can help the system generate enough question pairs that have answers in the paragraph, and not be subject to the error in the detector, thus increasing the performance of the whole system.

2.2 Question Answering

Improving interpretability

While FID is a powerful generative model that is capable of answering more complex questions by scaling to a large number of passages and aggregating evidence from multiple sources, it is a rather opaque model. In contrast to extractive models, which directly take answers from support passages, generative models are difficult to interpret and often suffer from producing hallucinated answers. As a result, we would like to further improve the interpretability of the baseline model by adapting the SOTA post hoc explanation methods such as attention and LIME.

Attention mechanisms have recently dominated many NLP tasks. Since the attention layers in attention mechanisms assign weights to inputs, it is often assumed that attention can be used to identify the information that models found important. We would like to check the attention weights in our task and test whether this assumption holds true [5].
LIME (Local Interpretable Model-Agnostic Explanations) is a popular local model-agnostic explanation method by learning an interpretable model locally around the prediction. [6]
We would like to adapt LIME to our task by generating local explanations to improve the interpretability of our model.

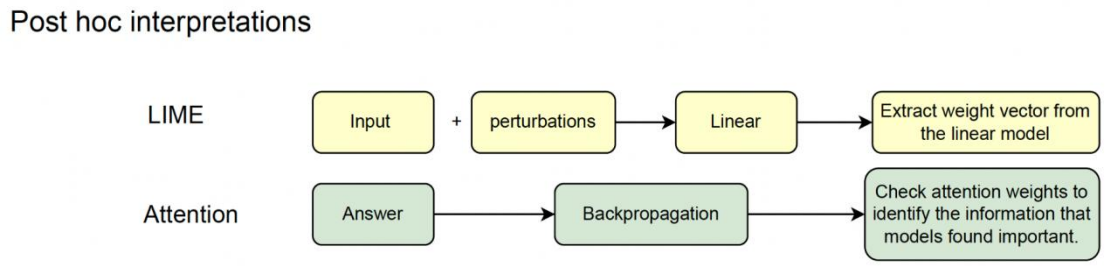

Figure 3. Post hoc interpretations

Improving Accuracy

For better accuracy as well as interpretability, we plan to reproduce the FID–PGN model [2], which is built upon FID and solves the problem of generating hallucinated answers.

To reproduce FID–PGN, we will add an additional pointer–generator network by having a one–layer MLP predicting probability of generation. Besides the probability distribution of generated vocabulary, we can get a copy distribution from the cross–attention score of the last decoder layer, which indicates the probability of copying a word from original passage to the answer. The final probability of a token y_t to be included in the answer is

$$P(y_t) = p_{\text{gen}} P_{\text{vocab}}(y_t) + (1 - p_{\text{gen}}) P_{\text{ctx}}(y_t). \qquad \textit{(Eq.1)}$$

This approach integrates both extractive and generative readers. We plan to use the same method in section 2 to analyze whether explicitly modeling the decision of directly copying or generating answer tokens leads to improvement in interpretability. In addition, we will compare the results of it with FID to validate the improvement in accuracy.

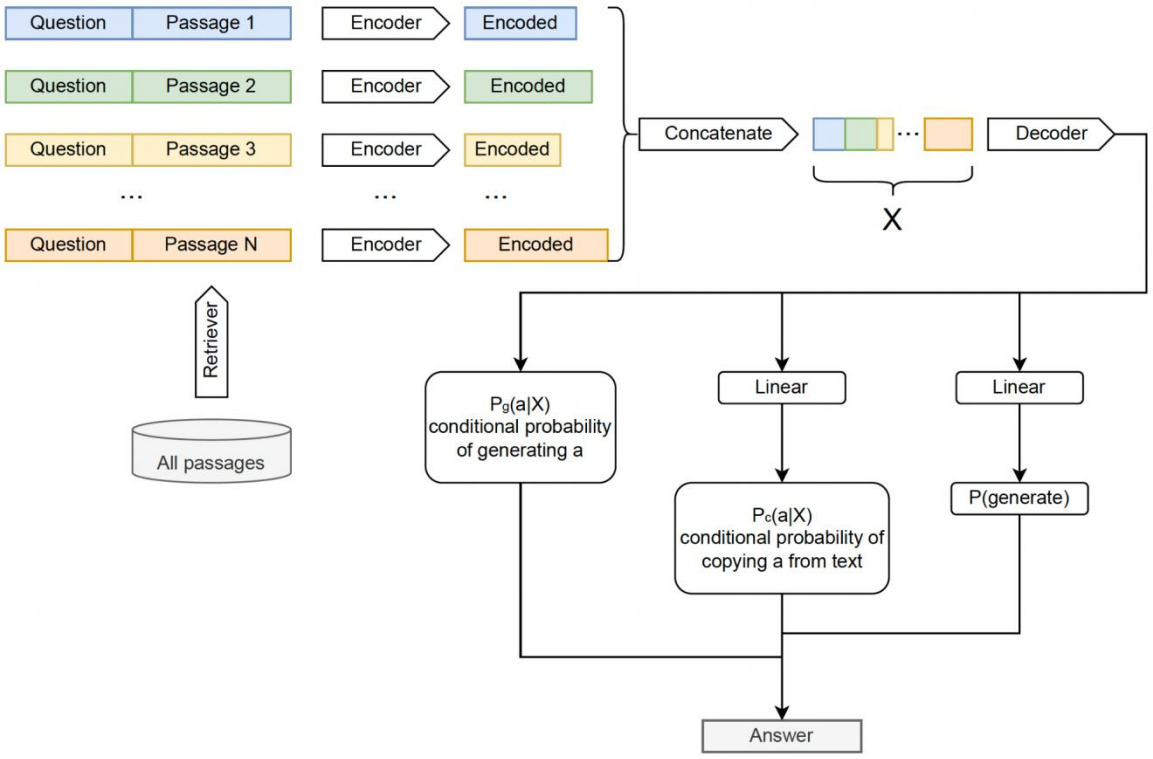

Figure 4. FID-PGN architecture

3 Management Plan

1. How responsibilities will be allocated:

Guoyao Li and Eric Li are responsible for building QG models

Huiyi Zhang and Yuting Deng are responsible for building QA models

2. How often the team will meet (typically at least once per week):

The team will meet once a week on weekends

3. How the team will communicate between meetings:

We will use Slack to communicate

We will use Zoom to help each other code and debug

4. How the codebase will be shared and managed:

We will use one public google drive and a github repository to store dataset and code.

5. How disputes between team members will be settled:

Since we divided the team into 2 groups and each responsible for one part, disputes can only happen in a group. Thus, we can find another group to participate in the discussion together.

Reference:

[1] Gautier Izacard and Edouard Grave. 2021. a. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 874–880, Online. Association for Computational Linguistics. (github, paper)

[2] Shuang Liu, Dong Wang, Xiaoguang Li, Minghui Huang, and Meizhen Ding. 2022. A Copy-Augmented Generative Model for Open-Domain Question Answering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 435–441, Dublin, Ireland. Association for Computational Linguistics. (paper)

[3]Alona Sydorova, Nina Poerner, and Benjamin Roth. 2019. Interpretable Question Answering on Knowledge Bases and Text. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4943–4951, Florence, Italy. Association for Computational Linguistics.

[4] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5540–5552, Online. Association for Computational Linguistics.

[5] Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable?. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

[7] Chan, Ying-Hong, and Yao-Chung Fan. "BERT for question generation." Proceedings of the 12th International Conference on Natural Language Generation. 2019.