Transformer-Based Multi-Hop Question Generation (Student Abstract)

John Emerson and Yllias Chali
University of Lethbridge
4401 University Dr W
Lethbridge, AB T1K 3M4 Canada
john.emerson@uleth.ca, yllias.chali@uleth.ca

Abstract

Question generation is the parallel task of question answering, where given an input context and, optionally, an answer, the goal is to generate a relevant and fluent natural language question. Although recent works on question generation have experienced success by utilizing sequence-to-sequence models, there is a need for question generation models to handle increasingly complex input contexts to produce increasingly detailed questions. Multi-hop question generation is a more challenging task that aims to generate questions by connecting multiple facts from multiple input contexts. In this work, we apply a transformer model to the task of multi-hop question generation without utilizing any sentence-level supporting fact information. We utilize concepts that have proven effective in single-hop question generation, including a copy mechanism and placeholder tokens. We evaluate our model's performance on the HotpotQA dataset using automated evaluation metrics, including BLEU, ROUGE and METEOR and show an improvement over the previous work.

Introduction

Previous research on question generation has largely focused on generating questions that require limited reasoning to be answered. Typically, the input context used to generate these questions is limited to a single sentence or small paragraph. Multi-hop question generation expands the question generation task and attempts to generate questions that require multiple reasoning hops over multiple, related context paragraphs.

Proposed Model

Previous multi-hop question generation research has relied on supporting sentences being labelled within the training data (Gupta et al. 2020). This means that sentences within a context paragraph are labelled as to whether they are related to the reference question supplied by the dataset. More recent work has addressed the task in a setting free of this information by utilizing an RNN-based seq-to-seq model (Su et al. 2020). Such RNN-based models offer little opportunity for parallelization, which is problematic since multi-hop question generation utilizes input contexts considerably longer than the traditional question generation task.

We perform multi-hop question generation in a setting free of sentence-level supporting fact information by utilizing a transformer strengthened with components that have proven effective for the traditional question generation task. These components include placeholder tokens and a pointer-generator.

Dataset

The HotpotQA dataset (Yang et al. 2018) contains ∼92k samples which map two context paragraphs sampled from related Wikipedia articles to a multi-hop question. It contains two types of samples: bridge and comparison. Bridge samples form a question by identifying an entity that connects the two paragraphs and then utilize that connection to generate a question. In comparison samples, the context paragraphs each discuss a different named entity and generate a question that asks about a property they share.

Placeholder Tokens

Scialom, Piwowarski, and Staiano (2019) demonstrated that placeholder tokens are an effective way to improve the performance of transformers when applied to the single-hop question generation task.

∼52% of samples in their dataset contain named entity tokens, which results in a high number of out-of-vocabulary tokens. By utilizing placeholders, they greatly reduce this phenomenon. They replace named entity tokens within the input context and reference question with a placeholder token corresponding to the type of named entity. For example, the sentence "Taylor Swift can really sing." would be preprocessed into "*[person_1] [person_2]* can really sing." As a post-processing step, all placeholder tokens in the generated question are converted back into the original named entities they represent.

Within the HotpotQA dataset, ∼75% of answers contain at least one named entity token and ∼60% of answers contain only named entity tokens, so we employ the same placeholder technique.

Encoder

We utilize the standard transformer encoder detailed by Vaswani et al. (2017). First, all tokens in the input contexts are embedded. The embeddings that represent placeholder tokens are learned, while the embeddings corresponding to

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
MulQG (Su et al. 2020)	40.15	26.71	19.73	15.20	35.30	20.51
Proposed Model	**42.13**	**30.44**	**23.84**	**19.42**	**39.26**	**22.78**

Table 1: A performance comparison of the automatic evaluation metrics between our model and (Su et al. 2020).

non-placeholder (standard) tokens are frozen. We generate part-of-speech (POS) tags and answer position tags for each token using the BIO tagging scheme (Zhou et al. 2017). These tags are embedded with learnable vectors and concatenated with their token's embedding before the positional encoding is applied.

Before encoding the input contexts, we separate the context paragraphs with a *[SEP]* token. The motivation behind explicitly marking this boundary is to aid the model in encoding the connection between relevant entities across the two context paragraphs, resulting in a question that requires multiple reasoning hops.

Decoder and Pointer-Generator

We utilize the standard transformer decoder described by Vaswani et al. (2017), modified with a pointer generator, which gives the model the ability to choose between copying out-of-vocabulary tokens from the context paragraphs or selecting tokens from the model's vocabulary. Similar to the encoder, embeddings corresponding to the placeholder tokens are learned, while embeddings corresponding to the standard tokens are frozen.

The pointer-generator is based on the design introduced by See, Liu, and Manning (2017), but adapted for the transformer model. Such implementations are common in works utilizing transformer-based models (Prabhu and Kann 2020). For each position in the generated question, the pointer-generator utilizes a soft switch that determines whether to copy or generate a token.

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \tag{1}$$

$$P(w) = p_{gen} p_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \tag{2}$$

The soft switch $p_{gen} \in [0, 1]$ for position t in the generated question is shown in equation 1, where w_{h^*}, w_s, and w_x are learnable vectors, and b_{ptr} is a learnable scalar. s_t represents the decoder hidden state, x_t represents the input to the decoder, and h_t^* represents a context vector which is calculated by summing the encoder hidden states, which are weighted by an attention distribution a^t. We derive a^t by averaging the heads of the encoder-decoder attention layer from the last decoder block. The probability of a token w appearing at position t in the generated question is calculated by summing its generation probability and copy probability, as shown in Equation 2, where i represents each position in the input context where the token w occurs.

Evaluation and Conclusion

We utilize the automated metrics that are most common in the previous work on automatic question generation. Table 1 contrasts the performance of our model with Su et al. (2020), who also attempt the multi-hop question generation task in a setting free of sentence-level supporting fact information.

Our results show that transformer-based models are an effective solution to the multi-hop question generation task in a setting free of sentence-level supporting fact information. We also show that enhancements such as placeholder tokens and a copy mechanism, which have proven effective in the single-hop question generation task, can be successfully leveraged for the multi-hop variation. Finally, our results demonstrate a notable improvement in multiple evaluation metrics.

References

Gupta, D.; Chauhan, H.; Akella, R. T.; Ekbal, A.; and Bhattacharyya, P. 2020. Reinforced Multi-task Approach for Multi-hop Question Generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2760–2775. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Prabhu, N.; and Kann, K. 2020. Making a Point: Pointer-Generator Transformers for Disjoint Vocabularies. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, 85–92. Suzhou, China: Association for Computational Linguistics.

Scialom, T.; Piwowarski, B.; and Staiano, J. 2019. Self-Attention Architectures for Answer-Agnostic Neural Question Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6027–6032. Florence, Italy: Association for Computational Linguistics.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. arXiv:1704.04368.

Su, D.; Xu, Y.; Dai, W.; Ji, Z.; Yu, T.; and Fung, P. 2020. Multi-hop Question Generation with Graph Convolutional Network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. arXiv:1809.09600.

Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; and Zhou, M. 2017. Neural Question Generation from Text: A Preliminary Study. arXiv:1704.01792.