Schema Generation for Narratives via Question Answering

Joongwon Kim, Paul Scott, Jong Min Choi
Department of Computer and Information Science, University of Pennsylvania
{jkim0118, pscott4, jongmin}@seas.upenn.edu

Abstract

Narratives contain sentences that perform different roles, particularly in relation to other sentences, within a given context. While current NLP models often assume input text to be a linear chain of sentences, this is often not the case, especially for narratives which contain diverse events. In this paper, we introduce a method to automatically construct schemata for any given textual narrative. Informed by the Question-Under-Discussion framework, we use questions as intermediary representations to connect pairs of sentences in narratives. First, we generate questions for each sentence using pre-trained seq2seq models. Second, we fine-tune classifiers that identify whether a given sentence functions as an anchor for a given question. Third, we iterate both models over each article to generate its schema. We demonstrate that our pipeline is able to generate realistic schemata representing narratives. Furthermore, we discuss the shortcomings of our current approach as well as future directions for extending our work beyond textual narratives.

1 Introduction

An important task in natural language processing (NLP) is for computers to understand the given text and derive its semantic and structural representation. One such application for textual understanding can be found in narratives. Narratives often contain sentences that perform different roles, particularly in relation to other sentences, within a given context. Therefore, it is possible to map the relationships between pairs of sentences in each narrative to construct a schema. We take a question-answering approach to investigate this hypothesis. More specifically, we observe that pairs of sentences in a given narrative can be connected by a question motivated from one sentence, for which the answer is found in another sentence. The resulting schema consists of sentences as nodes and questions as edges.

Instead of treating a narrative as a linear progression of sentences, the schema allows one to construct a richer representation of a narrative. This is especially useful in answering questions about a given narrative. As language models' performances do not scale well with long input text (Beltagy et al., 2020), it is possible that incorporating sentences from a particular subpath or a subgraph of the schema as condensed context allows models to focus on the most important parts of the narrative. We expect that this may lead to gain in performances, or insignificant drops in performance while using shorter context for question answering. This is important even in the latter case, as contemporary QA systems must handle questions with efficiency and providing shorter contexts reduce computation time for associated models. Hence we derive schemata not only to enrich narrative representations, but also to set a new direction for performing downstream NLP tasks.

In summary, the main contributions of our work include the following:

- We reproduce previous efforts to generate questions from a given sentence via language models such that the answer to the generated question can be found in the sentence.
- We demonstrate that language models can be used to classify the relevance of a question to a given sentence and perform anchor detection.
- We implement a novel method for generating schemata for any narrative by combining the aforementioned models in a pipeline.

2 Related Work

2.1 Question Answering

Question answering is a task in which a system must provide a response to natural questions posed by humans. On a high level, it is formulated either in a closed-domain setting where the reader

model produces answers based on a given context (Rajpurkar et al., 2016; Joshi et al., 2017; Lai et al., 2017), or in an open-domain setting where the system must first retrieve a relevant document from the Web and then produce answers via the reader (Yang et al., 2015; Chen et al., 2017; Kwiatkowski et al., 2019).

As this setup indicates, models used for QA tasks encode contextual information that provides background information necessary for answering the question. This holds true not only for the standard QA setup but also in other applications such as conversational QA (Choi et al., 2018; Reddy et al., 2019; Anantha et al., 2021), where the system encodes dialogue history as part of its context for answering the user's question. While this has been demonstrated to be effective, recent studies demonstrate its limitations due to language models' weaknesses to long inputs (Beltagy et al., 2020) as well as experiments suggesting insignificant benefits of using entire contexts (Ko et al., 2021). This observation motivates us to introduce a method that allows for more efficient usage of context in the case of narratives.

2.2 Language Models

Modern language models (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Brown et al., 2020) are built from transformer layers (Vaswani et al., 2017) and pre-trained on massive text corpora scraped from various sources such as news articles, blog posts and social media. They can be fine-tuned on task-specific datasets to perform a wide variety of downstream NLP tasks, including sentiment analysis, question answering, natural language inference, machine translation, summarization, and many others. With their advent, such pre-trained models have outperformed other neural baselines on the aforementioned tasks and demonstrated powerful learning capabilities.

One notable model is GPT-3, which is a language model created by OpenAI. The model is capable of performing many tasks, such as text generation, machine translation, and even code generation (Brown et al., 2020). GPT-3 also features great capability for question answering as well as question generation, in which the model generates questions that can be answered by the input text.

2.3 Discourse Structures

Recent studies have utilized question answering to study discourse structures. One way to represent the relationship between various parts within a sentence or pairs of sentences is to formulate a question connecting the two segments, also known as the Question Under Discussion (QUD) framework (Pyatkin et al., 2020). This has resulted in efforts to derive questions inspired from text (Scialom and Staiano, 2019; Ko et al., 2020).

Another recent work presents a dataset containing annotations of news articles with human-written questions connecting sentences within the articles (Ko et al., 2021). Each question is motivated by an anchoring sentence from the article, and the answer to the question can be found in a subsequent sentence in the same article. In our work, we extensively use this dataset to train classifiers and generators that are used in our schema generation pipeline.

3 Methods

Given an input narrative text, our goal is to generate a schema consisting of sentences as nodes and questions as edges connecting pairs of sentences in the narrative. More formally, each pair of sentence (s_1, s_2) is connected by a question q_{12} which is motivated from s_1 and finds its answer from s_2. Here, we define s_1 to be the *anchor* of q_{12}. An overview of our pipeline is given below, along with a figure that illustrates the process:

1. Train a generator that produces a question which can be answered by the input sentence.
2. Train a classifier that identifies whether a given sentence functions as an anchor for a given question.
3. For each article, run the generator to produce questions for each sentence in the narrative.
4. Run the classifier for each generated question along with previous sentences and construct a schema based on their scores.

Figure 1 provides a visual overview of our pipeline which enables the generation of schemata from narratives. Note that the process can be fully automated as the fine-tuned models can be run on any given sentence or pair of sentences.

3.1 Question Generation

Question generation is a task which aims to generate a question which can be answered by the given sentence. While previous works employ complex

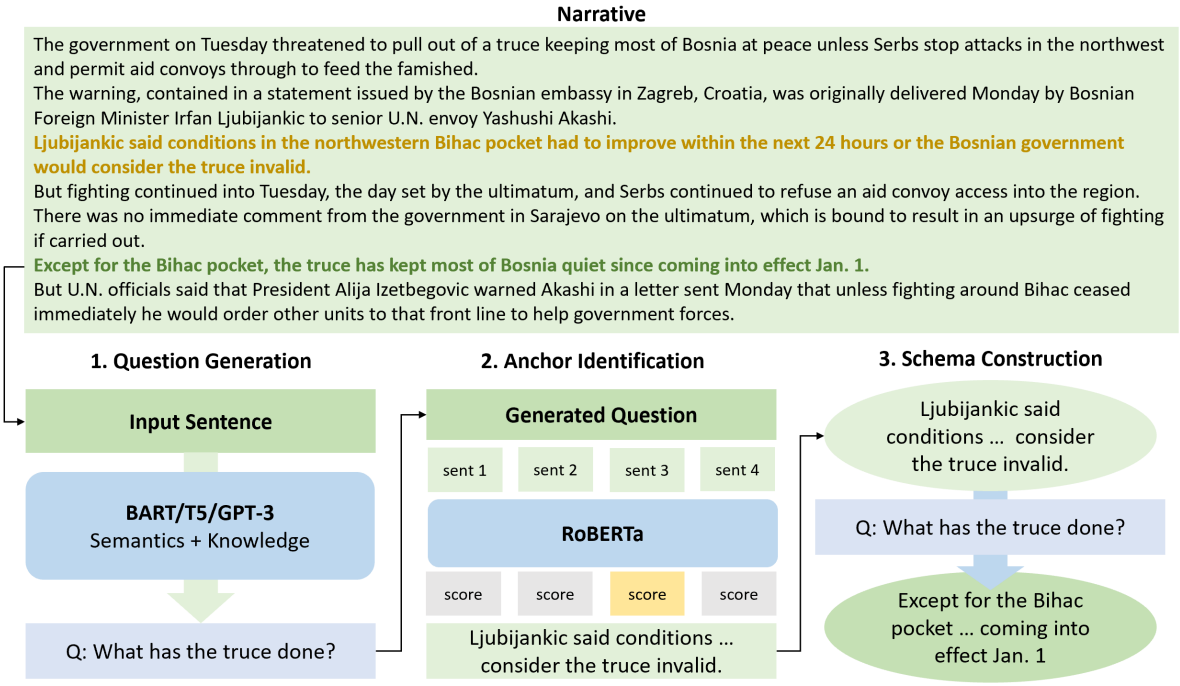

Figure 1: Overview of our schema generation pipeline. Given an input text of narrative, we first process a fine-tuned question generator on each sentence of the narrative. Then, we run the generated question along with each previous sentence through a fine-tuned anchor classifier. Finally, we take the highest-scoring pair and connect the anchor sentence to the original sentence via the generated question. Repeating this process for all sentences in the narrative, we obtain a schema that details the discourse structure of the narrative.

methods on top of pre-trained language models (Sun et al., 2018; Chan and Fan, 2019; Cao and Wang, 2021), or jointly train with the question answering objective (Dong et al., 2019; Dugan et al., 2022) to improve the quality and diversity of generated questions, we focus on obtaining questions of sufficient quality that can be answered by the sentence. Therefore, our question generation models are directly fine-tuned on pairs of questions and sentences that contain their answers.

We fine-tune three different seq2seq models: BART (Lewis et al., 2020), T5 (Raffel et al., 2019) and GPT-3 (Brown et al., 2020). Our choice of BART and T5 is based on the fact that both models display high performance across many subtasks in natural language generation, and are easily accessible. Meanwhile, we choose to use GPT-3 for its state-of-the-art performance on natural language generation benchmarks.

3.2 Anchor Identification

We define the task of anchor sentence identification, where given a pair of a question and a sentence, the objective is to determine whether the question is *anchored* upon the sentence. In other words, the goal is to identify whether the sentence can sufficiently motivate the question. Note the relative ease of this task compared to finding the *answer* sentence, as the classifier merely needs to identify cues that relate the question to the sentence rather than factual consistency. We frame this as a classification task, for which a normalized score between 0 to 1 is returned to indicate the relatedness of the question and sentence.

To perform anchor identification, we fine-tune RoBERTa (Liu et al., 2019) on pairs of questions and their anchors and evaluate its performance. We experiment with both RoBERTa-base and RoBERTa-large. Our choice is based on the fact that RoBERTa demonstrates high performance

Algorithm 1: Schema Construction

F_{QG}: question generator
F_{AC}: anchor classifier
input: document D, given as a list of sentences
output: schema G, initialized as a set of $\|D\|$ nodes

for i **in** $[1...\|D\|]$:
  $L_i \leftarrow$ list()
  $Q_i \leftarrow F_{QG}(D_i)$
  for j **in** $[1...i-1]$:
    L_i.append$(F_{AC}(Q_i, D_j))$
  $i^* \leftarrow$ argmax(L_i)
  G.add$(e(D_{i^*}, D_i))$
return G

Category	Train	Dev	Test
# of questions (total)	11,247	244	398
# of questions (avg)	37.62	34.86	20.95
length of articles (# sent)	30.31	23.0	23.32
length of questions (# words)	9.05	8.33	6.75

Table 1: Dataset statistics. We report 1) the total number of questions in each article which is either paired with an *anchor* sentence or *answer* sentence for our classification/generation tasks, 2) the average number of questions from each article, 3) the average number of sentences in each article, and 4) the average number of words in each question.

across many subtasks in natural language understanding, and is easily accessible online.

3.3 Schema Construction

Using the fine-tuned question generator and anchor classifier, we present a method to automatically construct schemata from a given narrative. The detailed algorithm can be found in Algorithm 1. Given a document D comprised of individual sentence units, the algorithm iterates through the sentences in a sequential order. For each sentence, a question is generated with the question generator F_{QG}, resulting in a question Q_i. Then, Q_i is paired with each preceding sentence and scored by the anchor classifier F_{AC}. Of the preceding sentences, the maximum scoring sentence D_{i^*} is paired with D_i and the corresponding edge is added to G. Since none of the steps outlined in Algorithm 1 require human supervision, this process can be fully automated as long as F_{QG} and F_{AC} are provided.

4 Data

Our dataset is taken from DCQA (Ko et al., 2021), which consists of hundreds of news documents with annotated questions connecting pairs of sentences in each article. Each data sample can be treated as a triple of (q, s_{anc}, s_{ans}), where q denotes the question, s_{anc} denotes the anchor sentence, and s_{ans} denotes the answer sentence. Hence the input to the classification task is given as (q, s_{anc}) and the input to the generation task is given as (s_{ans}, q).

A summary of the dataset statistics is given in Table 1. The news documents contain around 20-30 sentences on average, with 20-40 associated questions written by human annotators. We use this dataset to build the training and evaluation sets for the models outlined above.

5 Evaluation

5.1 Question Generation

BART/T5. To generate a question from a given sentence, we utilize seq2seq models, more specifically BART (Lewis et al., 2020) and T5 (Raffel et al., 2019) due to their success in downstream generation subtasks. We preprocess DCQA (Ko et al., 2021) to obtain pairs of answer sentences and questions, and fine-tune the models on the training set consisting of 11K such pairs[1]. Then, we perform evaluation with the fine-tuned model on the test split of the dataset and use the ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019) metrics to evaluate the quality of the generated questions.

GPT-3. Meanwhile, we also investigate the ability of GPT-3 for generating text from a smaller number of demonstrations. More specifically, we use the statement as the prompt for GPT-3 and use the question as the completion for training. For example, if the statement *Her time was 6.95 seconds.* is used as a prompt, then *What was her time?* is a possible completion for the given prompt.

We perform experiments in two settings: a few-shot learning setting where the model is provided five examples to learn from, and a fine-tuning setting where we provide 1000 randomly sampled training data. To account for financial budget, we use the Curie model for both experiments. After providing examples or fine-tuning on the training set, we generate questions for each prompt in the test and validation sets. These results are then compared to the true question completions from both sets using ROUGE and BERTScore.

5.2 Anchor Identification

We fine-tune RoBERTa (Liu et al., 2019) on the training set of DCQA comprised of 11K pairs of questions and their anchor sentences. Intuitively, questions are most likely related to sentences that have many overlapping entities and actions. As such, we add a baseline which computes a heuristic measure based on lexical overlap. This baseline computes the fraction of words, in their lemma forms, in the question that overlap with the candidate anchor. Meanwhile, textual similarity is another signal that may be helpful for detecting anchors. Hence we add an SBERT (Reimers et al., 2019) model based on the MPNet (Song et al., 2020) architecture as another baseline, with the

[1]We perform fine-tuning with Fairseq and Huggingface.

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
GPT-3 (few-shot)	21.22 / 20.94	3.99 / 6.12	19.42 / 20.22	89.55 / 89.51
GPT-3 (fine-tune)	21.05 / 21.37	4.31 / 5.61	19.87 / 20.70	89.62 / 89.53
BART-base	27.41 / 27.09	8.09 / 9.15	26.25 / 26.39	90.71 / 90.56
T5-base	26.70 / 27.47	7.84 / 9.54	25.19 / 26.73	90.39 / 90.48
BART-large	27.73 / 27.99	8.26 / 10.05	26.34 / 27.23	90.74 / 90.74
T5-large	27.78 / 28.50	7.55 / 9.93	26.32 / 27.72	90.56 / 90.45

Table 2: Results for question generation. Each cell contains metrics computed for development/test set.

Model	Accuracy	Precision	Recall	F1
Lexical Overlap	75.81 / 71.82	74.88 / 71.18	74.25 / 69.34	74.51 / 69.74
SBERT (MPNet-base)	72.84 / 70.96	72.23 / 70.56	73.01 / 67.95	72.33 / 68.32
RoBERTa-base	83.33 / 80.34	82.85 / 79.97	82.20 / 78.98	82.48 / 79.36
RoBERTa-large	84.68 / 80.35	84.12 / 79.67	85.33 / 80.23	84.38 / 79.88

Table 3: Results for anchor classification. Each cell contains metrics computed for development/test set.

score indicating the similarity between the question and the candidate anchor. We use accuracy, precision, recall and F1 metrics to evaluate the performance of the anchor classifier.

6 Results

6.1 Question Generation

Table 2 summarizes the results for the question generation task. The scores indicate that both BART and T5 are able to generate reasonable questions which can be answered by their input sentences. While BART tends to exhibit higher scores on the validation set, T5 exhibits higher scores on the test set. As one would expect, large models perform overall better than base models according to the automatic metrics.

Meanwhile, we observe that the smaller seq2seq models supervised on larger portions of the training data score better than GPT-3 with few-shot learning or fine-tuning. This is likely due to the fact that the smaller models are better able to learn the distribution of the question data by observing much larger numbers of examples, resulting in higher scores on the automatic metrics. Further analysis of this result is explored in section 7.1.

6.2 Anchor Identification

Table 3 reports our results for the anchor classification task. Our fine-tuned RoBERTa model outperforms both the lexical overlap and semantic similarity baselines. Again, RoBERTa-large outperforms RoBERTa-base on most metrics except for precision on the test set. The F1 score of 80-85% on the development and test sets indicate the reliability of the anchor classifier.

6.3 Schema Construction

We demonstrate examples of the schemata generated by our pipeline in Figures 2 and 3. Each graph consists of a set of nodes which represent the sentences in the associated narrative, and edges which represent connections between sentences in the narrative via the generated questions. The schemata begins with a root node corresponding to the first sentence of the article, and connects all sentences of the narrative via a tree structure. Further description of our examples can be found in the captions of Figures 2 and 3.

7 Discussion

The results of our experiments corroborate that language models are good at performing semantic tasks which require one to reason over the relationship between two different sentences. However, upon closer inspection the schemata contain errors propagated from pre-trained models in our pipeline. We provide an analysis of these errors, along with future directions for applying our schemata.

7.1 Error Analysis

The errors observed in the generated schemata can be traced into two main sources.

Error from generator. One source of error observed in the generated schemata is the question generator. Current language models exhibit behaviors of generating text that is factually incorrect and unfaithful to the given contextual input (Cao et al., 2018), and we observe this behavior in our question generation modules. As shown in Table 4, the first example demonstrates how our model (T5-large) generates an unexpected entity *soldiers* which is not even mentioned in the original arti-

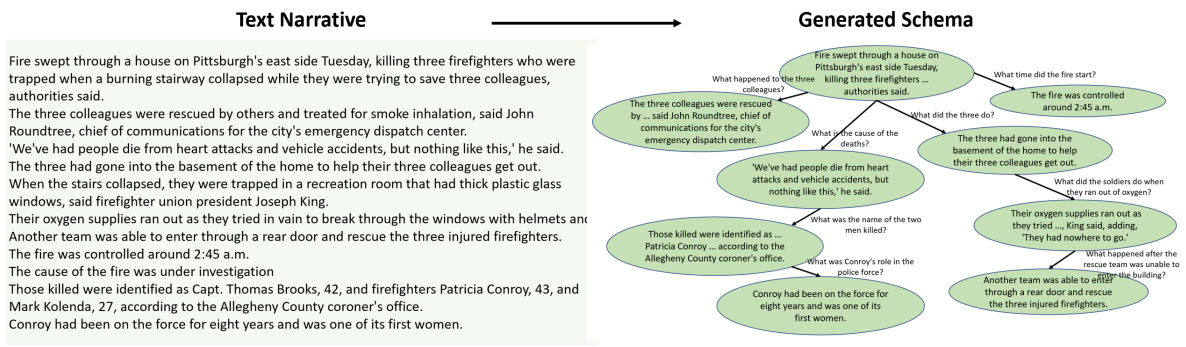

Figure 2: Example schema generated by our system. The article reports a fire accident killing three firefighters. One of the paths in the graph focuses the death of the firemen and subsequent investigation, while the other path examines the history of events that occurred during the moment of the accident.

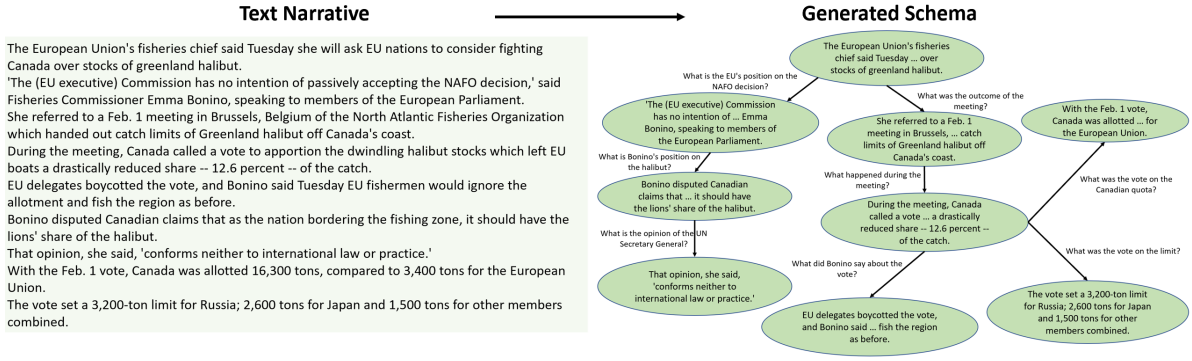

Figure 3: Example schema generated by our system. The article reports about the European Union's policy decision. One of the paths focuses on the position of EU and related officials on Canada's actions, while the other dives into the sequence of events during the meeting.

cle. Meanwhile, the second example shows that our model generates a verb *start* when the correct verb for this sentence would be *end*.

Error from classifier. Another source of error in our schemata is the anchor classifier. The performance of our classifier, while relatively high in terms of automatic metrics, is not perfect. As seen in Table 4, the third row associates the question asking about *two men killed*, which is not even true in the first place according to the narrative, with the sentence containing an unrelated comment.

To improve the quality of both question generation and anchor classification outputs, it may be beneficial to run coreference resolution (Lee et al., 2017) or decontextualization (Choi et al., 2021) models to ground pronouns and other components.

7.2 Future Work

We consider future work that can be performed with the schemata generated by our pipeline.

Are narratives structured linearly as treated by most NLP models? Our generated schemata demonstrate that this is not the case. Rather, the schemata contain branches of sentences focusing on different aspects of the given narrative. Therefore, it could be beneficial in downstream tasks to incorporate sentences not based on their *textual* positions but their *structural* positions as displayed in our schemata. More specifically, for a given question, the system could identify the anchor in the schema and use sentences along the path leading to the anchor as contextual input.

Can this pipeline be extended beyond textual narratives? Yes, one exciting direction is conversational QA, where the models incorporate previous QA turns as context to produce the answer to a question. In a similar approach to building schemata for narratives, one could identify anchors for questions in a conversation to previous answers returned by the QA system and build a corresponding schema. Then, one could use the questions and answers along a particular path in the schema to answer a given question, rather than simply using the preceding k turns as contextual input.

Module	sentence	question
Question Generation	Their oxygen supplies ran out as they ... adding, 'They had nowhere to go.'	What did the soldiers do when they ran out of oxygen?
	The fire was controlled around 2:45 a.m.	What time did the fire start?
Anchor Classification	'We've had people die from heart attacks and vehicle accidents, but nothing like this,' he said.	What was the name of the two men killed?

Table 4: Examples of errors for generated outputs. The green text indicates correct outputs, and the red text indicates incorrect outputs. The errors can be traced to either the question generator or the anchor classifier.

8 Attribution

Joongwon was in charge of planning the project, executing fine-tuning experiments for BART and T5, and designing the schemata generation algorithm. Paul executed fine-tuning experiments for GPT-3 and helped with visualizations. Jong-Min performed literature review, preprocessed the dataset and performed error analysis.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shuyang Cao and Lu Wang. 2021. Controllable open-ended question generation with a new question type ontology. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6424–6439.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *thirty-second AAAI conference on artificial intelligence*.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, Dayheon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-unaware question generation for education. *arXiv preprint arXiv:2203.08685*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 6544–6555.

Wei-Jen Ko, Cutter Dalton, Mark Simmons, Eliza Fisher, Greg Durrett, and Junyi Jessy Li. 2021. Discourse comprehension: A question answering framework to represent sentence connections. *arXiv preprint arXiv:2111.00701*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. Qadiscourse-discourse relations as qa pairs: Representation, crowdsourcing and baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 671–688. Association for Computational Linguistics.

Thomas Scialom and Jacopo Staiano. 2019. Ask to learn: A study on curiosity-driven question generation. *arXiv preprint arXiv:1911.03350*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.