Controllable Open-ended Question Generation with A New Question Type Ontology

Shuyang Cao and **Lu Wang**
Computer Science and Engineering
University of Michigan
Ann Arbor, MI
{caoshuy, wangluxy}@umich.edu

Abstract

We investigate the less-explored task of generating *open-ended questions* that are typically answered by multiple sentences. We first define a *new question type ontology* which differentiates the nuanced nature of questions better than widely used question words. A new dataset with $4,959$ questions is labeled based on the new ontology. We then propose a novel question type-aware question generation framework, augmented by a semantic graph representation, to jointly predict question focuses and produce the question. Based on this framework, we further use both exemplars and automatically generated templates to improve controllability and diversity. Experiments on two newly collected large-scale datasets show that our model improves question quality over competitive comparisons based on automatic metrics. Human judges also rate our model outputs highly in answerability, coverage of scope, and overall quality. Finally, our model variants with templates can produce questions with enhanced controllability and diversity.

1 Introduction

Question-asking has long served as an effective instrument for knowledge learning (Andre, 1979; Tobin, 1990) and assessing learning progress (Holme, 2003; Downing and Yudkowsky, 2009; Livingston, 2009). Compared to the widely studied task of generating factoid questions that inquire about "one bit" of information (Du et al., 2017; Duan et al., 2017; Li et al., 2019), this work is interested in *generating open-ended questions* that require deep comprehension and long-form answers (Labutov et al., 2015). Such open-ended questions are valuable in education, e.g., to facilitate complex knowledge acquisition (Lai et al., 2017) and nurture reasoning skills (Shapley, 2000), as well as in other applications like improving search engines (Han et al., 2019) and building open-domain dialogue systems (Shum et al., 2018).

Input: It's a difficult task to undertake. Teenagers tend to identify gangs with "fitting" in. Peer pressure plays a large part in it and sometimes teenagers have problems with their own identity being part of a gang deals with those issues. It also provides a little bit of respect on the street ...

BART SAMPLING:
- How do you stop a teen from joining a gang? (PROCEDURAL)
- How do you get teenagers to stop being in gangs? (PROCEDURAL)
- How do you get teens out of gangs? (PROCEDURAL)

BART + QWORD:
- **How** do you get a teenager out of a gang? (PROCEDURAL)
- **What** is the best way to get teenagers out of gangs? (PROCEDURAL)
- **Why** do teenagers join gangs? (CAUSE)

TPLGEN:
- How do I get `[NP]` to quit being in `[NP]`? ⇒ **How do I get** my son **to quit being in** a gang? (PROCEDURAL)
- What are `[NP]`? ⇒ **What are** some programs for teenagers involved in gangs? (EXAMPLE)
- Why do `[NP] [V] [NP]`? ⇒ **Why do** teenagers identify gangs? (CAUSE)

Figure 1: Open-ended questions generated by different models after reading the same input: (1) BART decoded with nucleus sampling, (2) BART that considers different question words, and (3) our type-aware generator TPLGEN, that predicts focuses and operates with generated templates (to the left of the arrows). Questions generated by our model have diverse TYPEs.

Significant progress has been made in generating factoid questions (Zhang and Bansal, 2019; Zhou et al., 2019b; Su et al., 2020), yet new challenges need to be addressed for open-ended questions. First, *specifying the question type* is crucial for constructing meaningful questions (Graesser et al., 1992). Question words such as "why" and "when" are generally seen as being indicative of types (Zhou et al., 2019b), but they underspecify the conceptual content of questions (Olney et al., 2012). Using Figure 1 as an example, different

question words, i.e., both "how" and "what", can be used for inquiring about procedures. It thus calls for a new question type ontology that can *precisely capture the conceptual nature of questions*. Second, constructing questions from a text with multiple sentences needs to focus on its *central concepts or phenomena that necessitate extensive descriptions*. New representations are needed to capture such content as question focus(es), to go beyond existing methods that rely on entities and their neighboring words (Du et al., 2017; Sun et al., 2018) even though they are effective for generating factoid questions. Third, encouraging the *diversity* of generated questions (Sultan et al., 2020; Wang et al., 2020) is less explored but critical for real world applications, e.g., various questions should be proposed to gauge how well students grasp the knowledge of complex subjects.

In this work, we aim to address the challenges of generating open-ended questions from input consisting of multiple sentences. We first introduce **a new question type ontology**, drawn upon researches in cognitive science and psychology (Graesser et al., 1992), to capture deeper levels of cognition, such as causal reasoning and judgments. Based on the new ontology, we collect and annotate a dataset of 4,959 questions to benefit research in both question generation and answering.[1]

We then design **a type-aware framework** to jointly predict question focuses (what to ask about) and generate questions (how to ask it). Different from pipeline-based approaches (e.g., Sun et al. (2018)), our framework is built on large pre-trained BART (Lewis et al., 2020), and uses shared representations to jointly conduct question focus prediction and question generation while learning task-specific knowledge. It is further augmented by a semantic graph that leverages both semantic roles and dependency relations, facilitating long text comprehension to pinpoint salient concepts.

Moreover, to achieve the goal of producing various types of questions from the same input, we investigate two model variants that **use templates to improve controllability and generation diversity**: one using pre-identified exemplars, the other employing generated templates to guide question writing, with sample outputs displayed in Figure 1.

For experiments, we collect two new large-scale datasets consisting of open-ended questions with answers from (1) Yahoo Answers[2] L6 dataset and (2) popular question-asking communities on Reddit[3], consisting of 291K and 720K question-answer pairs, respectively. Compared to existing popular QA datasets, such as SQuAD (Rajpurkar et al., 2016) and MS MARCO (Bajaj et al., 2016)), questions in our datasets ask about complex phenomena and perplexing social issues that seek solutions expressed in a long form. Automatic metrics show that our type-aware question generation model outperforms competitive comparisons, highlighting the effectiveness of semantic graph-augmented representation and joint modeling of focus prediction and question generation. Human judges also confirm that questions generated by our model have better overall quality. Adding templates further promotes question diversity, as evaluated by both automatic evaluation and human assessment.

2 Related Work

Question generation has long been studied to reduce human efforts in constructing questions for knowledge learning evaluation (Mitkov and Ha, 2003; Brown et al., 2005). Early work relies on syntactic transformation to convert declarative sentences to questions (Heilman and Smith, 2010; Chali and Hasan, 2015). Recent advancements rely on sequence-to-sequence models to generate a question from a given sentence or paragraph by considering the focus, type, and general-specific relations of questions (Sun et al., 2018; Zhou et al., 2019b; Krishna and Iyyer, 2019). In particular, question likelihoods and rewards are designed to steer them toward being addressed by the given answers (Zhou et al., 2019a; Zhang and Bansal, 2019). Attempts are also made toward creating complex questions that require multi-hop reasoning over the given text, and graph-based representations have been an enabling tool to facilitate the access to both entities and relations (Pan et al., 2020; Su et al., 2020). While our model also enhances the input with a semantic graph, it boasts a richer representation by including both dependency and semantic relations, with predicted question focuses highlighted via extra node embeddings. Moreover, we create a separate layer of cross attentions that is dedicated to the semantic graph, while prior work uses the same set of attentions to attend to the concatenated text and graph representations.

[1] Our data and code are available at: https://shuyangcao.github.io/projects/ontology_open_ended_question.

[2] https://answers.yahoo.com/

[3] https://www.reddit.com/

Given the data-driven nature of question generation and answering tasks, recent studies take advantage of the availability of large-scale QA datasets, such as SQuAD (Rajpurkar et al., 2016), MS MARCO (Bajaj et al., 2016), HotpotQA (Yang et al., 2018), DROP (Dua et al., 2019), inter alia. These corpora mainly contain factoid questions, while our newly collected datasets are not only larger in size but also comprise significantly more open-ended questions for querying reasons and procedures. A dataset closer to ours is ELI5 (Fan et al., 2019), which also obtains open-ended question-answer pairs from Reddit, while one of our datasets includes more Reddit communities and thus covers a wider range of topics.

Our work is more inline with generating deeper questions with responses that span over multiple sentences, where manually constructed templates are found effective (Olney et al., 2012). For example, Labutov et al. (2015) use crowdsourcing to collect question templates based on an ontology derived from Wikipedia and Freebase topics. Different from the topic-based ontology, our question types are more aligned with cognitive levels. Moreover, our templates are automatically learned from training data. Recent work (Rao and Daumé III, 2018, 2019) focuses on asking clarification questions based on both retrieval and generation models. As there has been no suitable framework for diverse types of questions, this work aims to fill the gap by introducing type-aware generation models which optionally leverage question templates for better controllability.

Generating diverse questions is much less studied, with existing approaches mainly focusing on entity replacement (Cho et al., 2019), sampling decoding (Sultan et al., 2020; Wang et al., 2020), and post-filtering (Liu et al., 2020). However, the produced diversity is driven by word choice and syntax variation, with little ability to control on question types, which is the focus of this work.

3 Data Collection and Question Type Annotation

3.1 Open-ended Question Datasets

To collect open-ended questions, we resort to online forums with active question-asking discussions. Concretely, we gather and clean *question-answer pairs* from Reddit and Yahoo Answers, to train generators that construct questions by taking the corresponding answer as input.

Question Type	**Description** (asking for...)
VERIFICATION	the truthfulness of an event or a concept.
DISJUNCTIVE	the true one given multiple events or concepts, where comparison among options is not needed.
CONCEPT	a definition of an event or a concept.
EXTENT	the extent or quantity of an event or a concept.
EXAMPLE	example(s) or instance(s) of an event or a concept.
COMPARISON	comparison among multiple events or concepts.
CAUSE	the cause or reason for an event or a concept.
CONSEQUENCE	the consequences or results of an event.
PROCEDURAL	the procedures, tools, or methods by which a certain outcome is achieved.
JUDGMENTAL	the opinions of the answerer's own.

Table 1: Our new question type ontology, which is adopted and modified from Olney et al. (2012). Types are sorted by levels of cognition (lower to higher).

We choose five popular **Reddit** communities: `r/AskHistorians`, `r/Ask_Politics`, `r/askscience`, `r/explainlikeimfive`, and `r/AskReddit`, where open-ended questions are actively asked. The original posts (OPs) are extracted, with their titles becoming questions. We also keep the best answer with the highest karma (i.e., upvotes minus downvotes) if it is greater than 1. A second dataset with question-answer pairs is collected from the **Yahoo** Answers L6 corpus[4], which covers a broader range of topics than the Reddit data. For each question, the best answer is rated by the user who raises the question.

Preprocessing. To ensure both questions and answers are well-formed, human inspection is conducted in multiple iterations to design rules to filter out improper samples. For instance, we discard samples whose answers have less than 15 content words to avoid the inclusion of factoid question. More details are provided in Table 6 in Appendix A. Ultimately, 719,988 question-answer pairs are kept for Reddit, and 290,611 for Yahoo. Each dataset is then divided into train, validation and test sets with a 90%/5%/5% split. The average lengths of questions and answers are 14.5 and 117.8 for Reddit, and 12.2 and 123.6 for Yahoo.

3.2 Question Type Ontology and Annotation

Our question type ontology is adopted and modified from Olney et al. (2012), where 18 categories are originally proposed for knowledge learning as-

[4] https://webscope.sandbox.yahoo.com/

sessment. We recruited 6 native English speakers for three rounds of question type annotation. Based on the annotators' feedback after each round, we refine the definitions, merge ambiguous types, and delete inapplicable categories. For example, an initial EXPECTATION type is merged into CAUSE due to their similarities in seeking causality. Finally, 10 types are preserved (Table 1). As can be seen, our ontology is designed to better capture the nature of questions than question words.

Annotating Questions with Types. After the annotation guideline is finalized, we ask the same set of annotators to label 5,000 ($2 \times 2{,}500$) randomly sampled questions from both Reddit and Yahoo's training sets. Each question is labeled by two annotators, with disagreements resolved through discussions. After removing samples without consensus, the final dataset consists of $4,959$ questions. EXAMPLE questions are most prevalent, comprising 23.4% of samples, while only 2.6% are CONSEQUENCE questions. A Krippendorff's α of 0.67 is obtained for all samples, indicating a reasonable agreement level. The annotation guideline and examples for each question type are shown in Table 12 in Appendix A.

Training Question Type Classifiers. Since our type-aware question generation model requires a specified type as input, here we describe how to build two question type classifiers: (1) γ_q, that labels a type by *reading the question* and is used to provide question type labels during *training*; (2) γ_a, that predicts a type for use by *taking the answer as input* and is used during *test*.

Both classifiers are based on RoBERTa (Liu et al., 2019), where a prediction layer is built on top of the contextual representation of the `[BOS]` token to output question type probabilities. γ_q achieves a macro F1 score of 0.80 on a reserved test set, with data splits detailed in Appendix B. To train γ_a, in addition to the annotated questions, we run γ_q on unlabeled questions in Reddit and Yahoo and include samples whose type prediction confidence score is above 0.9. We train one γ_a for each dataset. γ_a obtains macro F1 scores of 0.48 and 0.46 on the same reserved test set over all types after training on Yahoo and Reddit, respectively.

After running γ_q on both datasets, we find that Reddit has significantly more EXAMPLE questions (43.8% of all samples). Yahoo dataset is more balanced, with PROCEDURAL questions being the most frequent type (19.9% of all samples). Distributions of question types for the two datasets are listed in Table 8 in Appendix B.

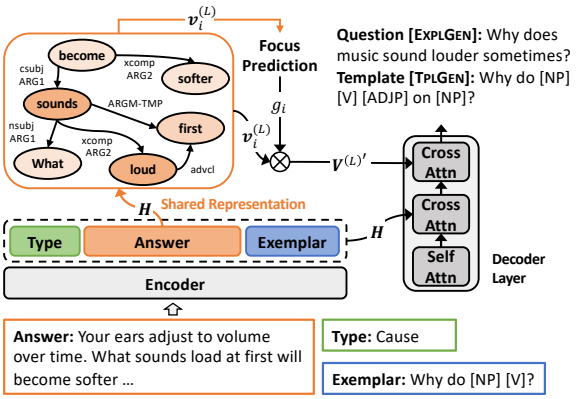


Figure 2: Our type-aware open-ended question generation framework. Detecting question focuses (nodes in darker color) and generating questions (or templates) are jointly learned. We only show a partial semantic graph. Special tokens are also inserted to segment different parts of the input. JOINTGEN uses the type and the answer for question generation; EXPLGEN further considers an exemplar; and TPLGEN uses all three for template generation.

4 Type-aware Open-ended Question Generation

In this section, we present our type-aware question generation framework. As shown in Figure 2, our model takes in a multi-sentence text and a predicted question type. Built on shared input representations, it first detects question focuses from a semantic graph, and then generates the question (§ 4.1). We also propose two model variants that consider automatically extracted template exemplars or generated templates to achieve controllability (§ 4.2), enabling the generation of diverse questions.

4.1 Joint Focus Prediction and Question Generation (JOINTGEN)

Our generator is built on top of BART (Lewis et al., 2020). To facilitate the detection of salient content (i.e., focuses) to raise questions, we first augment the encoder with a semantic graph that consists of both dependency relations and semantic roles, capturing semantic relations over different scopes with varying granularities. Question focuses are first detected based on the semantic graph, which then guide question generation via cross-attentions, as shown in Figure 2. Although the joint modeling of focus prediction and question generation has been studied before, our design differs by using shared

representations consisting of the input text and semantic graph, and the prediction of focuses are included through gating mechanisms, whereas previous work, e.g. Pan et al. (2020), simply employs multi-task learning. Below, we first describe constructing the semantic graph-augmented encoder, followed by the joint modeling of two tasks.

Improving Long Text Comprehension with Semantic Graph. To construct the semantic graph, for each sentence, we start with obtaining its dependency tree using Stanford CoreNLP (Manning et al., 2014). To better highlight core concepts, we discard less important relations, e.g., auxiliaries. The full list is included in Appendix C. Since our goal is to detect central concepts that are well connected with many other words, we can remove relations on the edges to minimize the number of parameters to learn. Moreover, as semantic roles can indicate main entities (Mannem et al., 2010), we extract semantic roles and their relations with AllenNLP (Shi and Lin, 2019). To merge the two sources of information, we add an edge in the dependency tree to connect the head word of the predicate and the head word of each semantic role. To build a connected graph from the multi-sentence input, we add an edge between each sentence's last token and the next sentence's first token. Finally, we merge nodes with the same surface forms or with corefered mentions. To the best of our knowledge, this is the first time that both dependency and semantic relations are encoded in the same graph for question generation, and with enhanced connectivity of the constructed graph, our design can better signal content salience.

Joint Modeling with Cross-attentions. Given a predicted question type t and a multi-sentence text $x = \{x_1, \cdots, x_n\}$, the BART encoder builds the contextual representation $\boldsymbol{H} = \{\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_n\}$ at the last layer, where $\boldsymbol{h}_0$ is for t.

To encode the semantic graph, we initialize the node representation for node v_i by taking the average contextual representations of its tokens and appending four bits encoding the number of nodes (capped at 10) that are merged into v_i, to add frequency information. This step yields new node representations $\boldsymbol{v}_i^{(0)}$. We then apply graph attention networks (GATs) (Veličković et al., 2018) of L layers to update the representations as follows:

$$\boldsymbol{v}_i^{(l)} = \sum_{j \in \mathcal{N}_i} a_{i,j} \boldsymbol{W}^{(l)} \boldsymbol{v}_j^{(l-1)} \quad (1)$$

where $\boldsymbol{W}^{(l)}$ is a learnable parameter for the l-th layer, and $\mathcal{N}_i$ denotes the neighbors of v_i. The attention score $a_{i,j}$ is calculated as in GATs. We use $L = 2$ for experiments.

To **predict focuses**, the final node representation $\boldsymbol{v}_i^{(L)}$ is fed into the following feedforward network, yielding the probability of v_i being a focus as:

$$p_{focus}(v_i = 1) = \sigma(\boldsymbol{W}_1 \tanh(\boldsymbol{W}_2 \boldsymbol{v}_i^{(L)})) \quad (2)$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are learnable parameters. Bias terms are omitted for simplicity. We construct ground-truth labels by treating a node as a focus if it contains words used in the question.

To **generate the question**, we use the gating mechanism to inform the focus prediction results, where new node representations after being weighted by the focus probability are:

$$\boldsymbol{v}_i^{(L)'} = g_i \odot \boldsymbol{v}_i^{(L)} \qquad g_i = p_{focus}(v_i = 1) \quad (3)$$

Our model benefits from both large pre-training and hybrid semantic graphs by adding a separate cross attention for node presentations in each BART decoder layer. We then design **separate cross attentions** to attend (1) the output of the BART encoder, yielding $\boldsymbol{z}_e$, and (2) the node representations $\boldsymbol{V}^{(L)'}$, producing $\boldsymbol{z}_v$, which are formulated as:

$$\boldsymbol{z}_e = \text{LN}(\boldsymbol{z}_s + \text{Attn}(\boldsymbol{z}_s, \boldsymbol{H})) \quad (4)$$

$$\boldsymbol{z}_v = \text{LN}(\boldsymbol{z}_e + \text{Attn}(\boldsymbol{z}_e, \boldsymbol{V}^{(L)'})) \quad (5)$$

$$\boldsymbol{z}' = \text{LN}(\boldsymbol{z}_v + \text{FFN}(\boldsymbol{z}_v)) \quad (6)$$

where $\boldsymbol{z}_s$ denotes the output of self attentions for the current layer, and $\boldsymbol{z}'$ is the output for the layer. $\text{Attn}(\cdot, \cdot)$ denotes the multi-head attention operation as in Vaswani et al. (2017), $\text{FFN}(\cdot)$ a feed-forward layer, and $\text{LN}(\cdot)$ is layer normalization.

Our final **training objective** accounts for both focus prediction and question generation objectives with equal weights.

4.2 Diversifying Questions with Templates (ExplGen & TplGen)

An important goal of this work is to enable the generation of questions of diverse types. However, simply adding question type as input is insufficient (discussed in § 5). We thus propose to leverage question templates to gain stronger controllability. Below we first present how to automatically

extract templates from the training set, and then introduce two model variants that are built on the JOINTGEN framework: EXPLGEN uses exemplar templates to guide the model to generate questions of selected types, and TPLGEN adds an extra step to first generate type-specific templates.

Template Extraction. While collecting templates specific to a given type, we need to ensure they remain topic-independent to be generalizable to different domains. To this end, we replace a word in the question with a template token that indicates its syntax function, e.g., `[V]` for a verb, if it appears in the answer after lemmatization. We further consider topically related words in the questions, by calculating word-level semantic similarities based on Numberbatch word embeddings (Speer et al., 2017), which are found to perform better on our datasets than other embeddings. Concretely, for each word in the answer, we replace the most similar word in the question with the template token. This process is repeated until 80% of content words in questions are replaced. Finally, for each noun phrase, adjective phrase, and adverb phrase, if its head word has been replaced, the whole phrase is transformed into a phrase type token. For instance, a question "What are the differences between global warming and climate change?" becomes "What are the differences between `[NP]` and `[NP]`?"

Exemplars for Guidance (EXPLGEN). Our first model variant considers adding a template exemplar for the given type as additional input, which provide more specific information to control the type of generated questions. Figure 2 shows one such example. To identify exemplars, we use templates with frequencies above 20 on Yahoo and 50 on Reddit. We then manually inspect these templates and remove the ones with topic-specific words, resulting in 66 exemplars for all types. They are listed in Table 10 in Appendix D.

During training, we choose the exemplar that has the lowest edit distance with the question, which is also used for training an exemplar selector based on RoBERTa. During testing, the exemplar with the highest selector score is used. The accuracy of the exemplar selector for each question type on the test set is reported in Table 11 in Appendix D.

Generated Templates for Guidance (TPLGEN). We further propose another model variant where we generate a new template and feed it (instead of an exemplar template as in EXPLGEN) as part of the question generation input. Specifically, we reuse EXPLGEN to learn to generate a target template, as derived from the template extraction procedure. During question realization, TPLGEN uses a BART-based generator that takes as input the question type, the input text, the generated template, and the words that are predicted as focuses. We use separate cross attentions to attend the representations of the focused words, similar to how node representations are attended in JOINTGEN.

We recognize that having separate stages of exemplar selection and template generation introduces extra model training cost and potential errors in the pipeline. This work, however, focuses on improving the controllability as well as diversity of question generation, and we will leave the building of more efficient models in the future work.

5 Experiment Results

5.1 Automatic Evaluation

Comparisons and Metrics. We compare with DEEPQG (Pan et al., 2020), a model that uses dependency graphs for multi-hop question generation. We also compare with BART models that are fine-tuned on the same datasets as in our models, by using inputs of (1) the answer (BART), (2) the answer and a predicted question word (BART+QWORD), and (3) the answer and a predicted question type (BART+QTYPE). For BART+QWORD, the question word is predicted by a RoBERTa classifier that considers the answer and is trained on our training sets. We follow Liu et al. (2020) and use 9 categories of question words. For both our models and BART+QTYPE, the most confident type predicted by the classifier γ_a (described in § 3.2), which reads in the answer, is used as input. To test the efficacy of semantic graphs, we further compare with a variant of JOINTGEN that only uses the flat Transformer for focus prediction and question generation, denoted as JOINTGEN w/o graph.

We evaluate the generated questions with BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin, 2004).[5]

Results on both Yahoo and Reddit datasets are reported in Table 2. *Our* JOINTGEN *outperforms all comparisons on both datasets* over all automatic evaluation metrics except for METEOR on Reddit. When taking out the semantic graphs, model performance degrades substantially, which suggests that

[5]We do not consider using Q-BLEU (Nema and Khapra, 2018) since it weighs question words highly.

Model	Yahoo B-4	Yahoo MTR	Yahoo R-L	Reddit B-4	Reddit MTR	Reddit R-L
DEEPQG	6.53	25.92	27.56	–	–	–
BART	21.88	38.01	39.16	19.45	35.46	37.82
BART+QWORD	22.02	38.44	39.32	19.80*	**35.85**	38.48*
Type-aware Models						
BART+QTYPE	22.12	38.62	39.72	19.90*	35.83	38.68*
JOINTGEN (ours)	**22.56***	**38.63**	**40.40***	**20.09***	35.75	**39.07***
w/o graph	22.21	38.21	39.93	19.81*	35.60	38.47*
EXPLGEN (ours)	21.74	37.52	39.70	18.67	33.28	36.74
TPLGEN (ours)	21.51	36.55	39.63	17.83	31.69	36.05

Table 2: Automatic evaluation results on Yahoo and Reddit with BLEU-4 (B-4), METEOR (MTR) and ROUGE-L (R-L). JOINTGEN outperforms comparisons over all metrics except for METEOR on Reddit, but removing its graph degrades performance. We only report results by DEEPQG on Yahoo due to memory limitation. *: significantly better than BART ($p < 0.005$ with approximate randomization test).

Model	Yahoo Acc↑	Yahoo UnT↑	Yahoo Pair↓	Reddit Acc↑	Reddit UnT↑	Reddit Pair↓
BART	–	2.49	**23.94**	–	2.03	**21.11**
BART+QWORD	–	3.21	41.77	–	3.40	26.61
Type-aware Models						
BART+QTYPE	22.47	2.27	44.39	23.67	2.20	80.91
JOINTGEN	43.90	4.03	63.06	22.75	2.33	60.72
EXPLGEN	75.47*	6.98*	25.12	65.79*	6.23*	22.17
TPLGEN	**76.35***	**7.08***	24.93	**66.49***	**6.32***	22.13

Table 3: Automatic evaluation on controllability and diversity by specifying 9 different question types. We report type accuracy (Acc), number of unique types (UnT), and pairwise BLEU-4 (Pair). Our EXPLGEN and TPLGEN achieve stronger controllability by respecting the given question types more, as well as show higher diversity than comparisons except for BART with nucleus sampling. *: significantly better than all comparisons ($p < 0.005$).

having structured representation is useful for focus detection and the final question generation task. We also observe a huge performance gap between DEEPQG and systems based on BART, signifying the importance of leveraging pre-trained models for open-ended question generation. Meanwhile, *adding question types helps BART generate more relevant questions* than using question words, indicating the value of our new question type ontology.

Notably, our template-based generators, EXPLGEN and TPLGEN, which are trained to comply with the given templates, still produce comparable scores. This highlights *the possibility to control the generated questions' types and syntax as demonstrated by the templates, without performance loss.*

Question Diversity Evaluation. Next, we examine the controllability of models by specifying different question types as input. The top 9 confident types[6] predicted by our type predictor γ_a are used as input to our models, producing 9 questions for evaluation. For BART, we use nucleus sampling (Holtzman et al., 2020) with $k = 10$ and $p = 0.7$ to sample diverse questions.

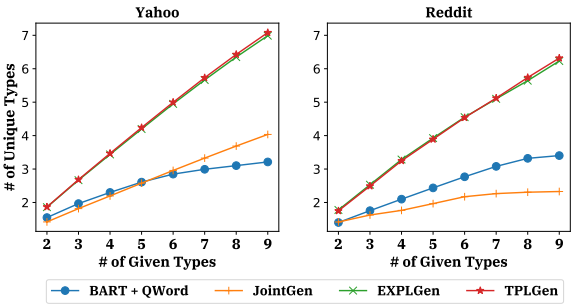


Figure 3: Number of unique types of the generated questions (Y-axis), when different numbers of question types are specified (X-axis).

To evaluate, we first calculate the **question type accuracy** by comparing whether the types of the generated questions match the specified ones, with types labeled by our classifier γ_q (§ 3.2). We then report the average numbers of **unique question types** in the 9 generated questions per sample, with higher number indicating better controllability. Finally, we consider **pairwise BLEU-4** (Cho et al., 2019) by computing the BLEU-4 between pairwise generated questions per sample, where lower values suggest higher content diversity.

First, *our* EXPLGEN *and* TPLGEN *can generate questions with diverse types and content*, as shown by the significantly higher numbers of unique types than all comparisons and lower pairwise BLEU scores than comparisons except for BART with nucleus sampling in Table 3. This implies stronger type control by template-based generators, compared to BART+QTYPE and JOINTGEN which only use the question type token as input. Results on numbers of unique types by varying numbers of question types specified in the input are displayed in Figure 3, where EXPLGEN and TPLGEN maintain steady controllability.

Second, *our question type ontology provides a new perspective for question diversity evaluation.* Among the comparisons, although BART with nucleus sampling and BART+QWORD both have low pairwise BLEU, the types of questions they can generate are limited.

[6] 9 types are chosen because we only have 9 categories of question words for BART+QWORD.

Model	Type (%)	Syntax (%)	Content (%)
BART	23.3	11.7	36.7
BART+QWORD	38.3	57.5	36.7
EXPLGEN	77.9	**61.2**	58.3
TPLGEN	**81.2**	58.3	**64.2**

(a) Yahoo

Model	Type (%)	Syntax (%)	Content (%)
BART+QWORD	40.0	**50.0**	22.9
JOINTGEN	63.3	31.2	20.4
EXPLGEN	**84.2**	35.0	45.8
TPLGEN	78.7	37.9	**60.4**

(b) Reddit

Table 4: Percentage of samples marked as having the most diverse question types, syntax structures, and content of answers. Ties are allowed. The Krippendorf's αs for the three aspects are 0.56, 0.43, and 0.25 on Yahoo, and are 0.54, 0.38, and 0.38 on Reddit.

Answer: My sister in law and her husband "genetically modified" their second child because the first has EB. They eliminated that and had a baby that gets to live pain free. Under the right circumstances, I'm all for it ...

JOINTGEN

~~[PROCEDURAL]~~ How would you feel about genetically modified babies?

[JUDGMENTAL] What are your thoughts on genetically modified babies?

[VERIFICATION] Is it possible to genetically modify a child?

EXPLGEN

[PROCEDURAL] How do people genetically modify their children?

[JUDGMENTAL]What do you think about genetically modified babies?

[VERIFICATION] Can you genetically modify a baby to have a better chance of survival?

TPLGEN

[PROCEDURAL] How do people genetically modify their children?

[JUDGMENTAL] What do you think about genetically modified babies?

[VERIFICATION] Can you genetically modify a baby?

Figure 4: Sample outputs of our models given different question types. Spans that belong to the exemplars or the generated templates are colored with blue. Generated questions that do not match the given type are marked by ~~strikethrough~~.

5.2 Human Evaluation

Question Diversity. We hire three annotators who have participated in our question type annotation study to evaluate 80 groups of questions generated by four selected models on each dataset. For each group, we randomly sample an answer and indicate three most probably question types to each model, to generate three corresponding questions.

For each sample, the annotators are asked to rank the four models from 1 (highest) to 4 (lowest) on three aspects of diversities: **type**–whether the three generated questions have different types, **syntax**–whether they use different syntax, and **answer content**–whether the three questions need to be addressed with different answers. Ties are allowed.

We find that *human judges rate questions generated by our* EXPLGEN *and* TPLGEN *as having greater diversities over all aspects*, except for syntax diversity on Reddit, as shown in Table 4. Among the two model variants, questions by TPLGEN yield more diverse answers. Based on our observation, TPLGEN uses automatically generated templates to produce more focused questions with different answers, compared to EXPLGEN which employs exemplars. This shows the promise of using automatically generated templates to create questions that need to be addressed with different answers. Besides Figure 1, we show more sample outputs in Figure 4, where EXPLGEN and TPLGEN exhibit stronger controllability than JOINTGEN.

Question Content Quality. We use the same set of human judges to evaluate another 80 groups of questions output by five selected models and the reference. Three aspects are rated from 1 (worst) to 5 (best): **appropriateness**–whether the question is semantically correct, *without* considering the answer; **answerability**–whether the question can be addressed by the given answer; and **scope**–whether the question is related to a longer span of the answer (global scope) or focuses on local content (e.g., one phrase or one sentence). We further ask the annotators to rank questions based on their overall quality and preferences, with ties allowed.

As shown in Table 5, our JOINTGEN model produces questions with better answerability and that cover broader content in the answers. It is also rated as the best in more than half of the evaluation instances on both datasets. Between BART+QWORD and BART+QTYPE, human judges rate the system outputs that conditioned on our question types to have better overall quality.

5.3 Further Analyses

Does focus prediction correlate with question quality? We first investigate the relationship between focus prediction and question generation by using our joint model JOINTGEN. As can be seen from Figure 5, there is a strong correlation between F1 scores of focus prediction and BLEU-4 as well

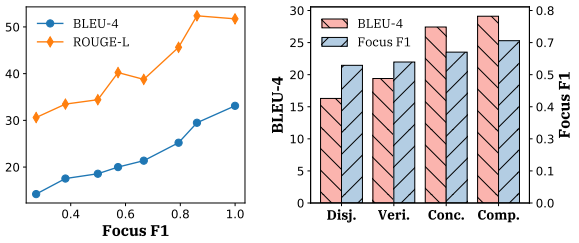
Model	Appro.	Ans.	Scp.	Top 1
REFERENCE	4.77	3.96	3.79	34.5%
BART	**4.93**	4.02	3.81	39.7%
BART+QWORD	4.86	4.14	3.85	40.8%
BART+QTYPE	4.92	4.23	3.94	48.7%
JOINTGEN	4.90	**4.25**	**3.96**	**50.5%**
TPLGEN	4.92	4.19	3.87	46.4%

(a) YAHOO

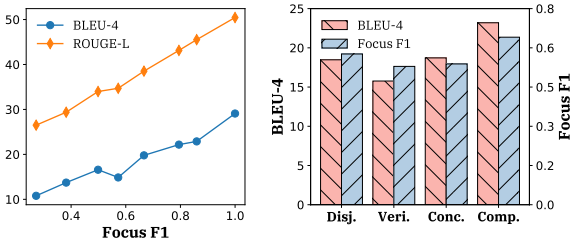
Model	Appro.	Ans.	Scp.	Top 1
REFERENCE	4.90	4.43	4.37	47.1%
BART	**4.89**	4.27	4.21	43.9%
BART+QWORD	4.88	4.29	4.21	46.7%
BART+QTYPE	4.88	4.39	4.26	49.6%
JOINTGEN	4.84	**4.45**	**4.38**	**50.3%**
TPLGEN	4.81	4.21	4.19	33.1%

(b) REDDIT

Table 5: Human evaluation on appropriateness (Appro.), answerability (Ans.), scope of answers (Scp.), and percentage of questions rated as having the best overall quality (Top 1), with ties allowed. JOINTGEN is rated as with higher answerability and with answers requiring a broader scope. The Krippendorff's αs for the four aspects are 0.42, 0.56, 0.50, 0.39 on Yahoo, and are 0.34, 0.46, 0.43, 0.35 on Reddit.



(a) Yahoo



(b) Reddit

Figure 5: On both Yahoo and Reddit, we find positive effect of focus prediction on question quality (measured by BLEU-4 and ROUGE-L) (left), which is also broken down by question types (right). Disj.: DISJUNCTIVE, Veri.: VERIFICATION, Conc.: CONCEPT, Comp.: COMPARISON.

as ROUGE-L, where samples in the Yahoo and Reddit test sets are grouped into 8 bins based on the F1 scores. The Pearson correlation coefficients between BLEU-4 and focus F1 are 0.29 on Yahoo and 0.26 on Reddit. For ROUGE-L, the correlation coefficients are 0.35 on Yahoo and 0.34 on Reddit. All the correlations have $p < 10^{-5}$. The strong positive correlations imply the importance of accurate focus prediction for open-ended question generation. We also show the F1 scores and BLEU-4 for selected question types on the right of Figure 5, again demonstrating the effect of focus detection on question quality.

When do our models fail to respect the given types? Next, we provide insights into which types of questions are challenging to generate by using our template-based models EXPLGEN and TPLGEN. Both variants frequently fail to respect the given question type of VERIFICATION, in which cases they often produce JUDGEMENTAL questions. They also tend to confuse EXAMPLE and EXTENT with CONCEPT questions. After manually inspecting 50 generated questions for the aforementioned three types, we find that many of them can be labeled with both types, thus creating confusion for our classifier. For instance, *"What are the import restrictions in the US?"* can be considered as either asking for a definition or for examples. Therefore, future work should include designing multi-class type identification models.

6 Conclusion

We present a new question type ontology which better captures the nuances of questions to support the study of open-ended question generation. We further annotate a new dataset with 4,959 questions based on the proposed ontology. We describe a joint question focus detection and question generation framework with a novel semantic graph-augmented representation, which is directly built on large pre-trained models. Based on this framework, we also enhance the controllability and diversity of generated questions by employing template exemplars or automatically generated templates. Experiments on two large datasets show that questions generated by our models have better quality and higher diversity than non-trivial comparisons, with similar results rated by human judges.

Acknowledgements

This research is supported in part by National Science Foundation through Grants IIS-1813341 and a CAREER award IIS-2046016. We thank three anonymous reviewers, area chair, and senior area chairs for their valuable suggestions for improving various aspects of this work.

Ethics Statement

Large models that are pre-trained on heterogeneous web data are shown to encode biases and can be potentially harmful for marginalized populations.

While the automatically learned templates improve controllability in question generation, we also recognize that our system might be misused to create questions that contain objectionable content. We therefore advocate cautious and responsible practices in real-world deployment.

Our data collection process for the two new datasets involves removing samples with abusive languages and human inspection on random samples. Given the data volume, however, we cannot exhaustively verify that all records are free of potentially offensive content.

References

Thomas Andre. 1979. Does answering higher-level questions while reading facilitate productive learning? *Review of Educational Research*, 49(2):280–318.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 819–826.

Yllias Chali and Sadid A Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*, 41(1):1–20.

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3121–3131, Hong Kong, China. Association for Computational Linguistics.

Steven M Downing and Rachel Yudkowsky. 2009. *Assessment in health professions education*. Routledge.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Arthur C Graesser, Natalie Person, and John Huber. 1992. Mechanisms that generate questions. *Questions and information systems*, 2:167–187.

Fred X Han, Di Niu, Kunfeng Lai, Weidong Guo, Yancheng He, and Yu Xu. 2019. Inferring search queries from web documents via a graph-augmented sequence to attention network. In *The World Wide Web Conference*, pages 2792–2798.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California. Association for Computational Linguistics.

Thomas Holme. 2003. Assessment and quality control in chemistry education. *Journal of Chemical Education*, 80(6):594.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2321–2334, Florence, Italy. Association for Computational Linguistics.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for 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.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019. Improving question generation with to the point context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3216–3226, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, pages 2032–2043.

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*.

Samuel A Livingston. 2009. Constructed-response test questions: Why we use them; how we score them. r&d connections. number 11. *Educational Testing Service*.

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 84–91.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. *Dialogue & Discourse*, 3(2):75–99.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. 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 143–155, Minneapolis, Minnesota. Association for Computational Linguistics.

Patricia Shapley. 2000. On-line education to develop complex reasoning skills in organic chemistry. *Journal of Asynchronous Learning Networks*, 4(2):43–52.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop question generation with graph convolutional network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4636–4647, Online. Association for Computational Linguistics.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. 2020. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Kenneth Tobin. 1990. Research on science laboratory activities: In pursuit of better questions and answers to improve learning. *School science and Mathematics*, 90(5):403–418.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Zhen Wang, Siwei Rao, Jie Zhang, Zhen Qin, Guangjian Tian, and Jun Wang. 2020. Diversify question generation with continuous content selectors and question type modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2134–2143, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019a. Multi-task learning with language modeling for question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3394–3399, Hong Kong, China. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019b. Question-type driven question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6032–6037, Hong Kong, China. Association for Computational Linguistics.

A Data Collection

Data Filtering. After collecting the raw data from Yahoo and Reddit, we design rules to filter out ill-formed answers and questions. These rules are listed in Table 6. Finally, we conduct human inspection on random samples from the two datasets and confirm that samples are all clean and contain open-ended questions.

Rules for Data Cleaning
- The question has URL links.
- The question has more than 1 sentence or does not end with a question mark.
- The question has less than 4 words or less than 1 content word.
- The question does not start with wh-words: *what, why, how, which, where, who, when*; yes-no words: *is, are, was, were, will, would, do, does, did, can, could, should, has, have*; or frequent words for conditions: *if, in, for, to, as, at*.
- The answer has less than 15 content words.
- The answer has less content words than the question.
- The answer has more than 30% of the words as digit letters.
- The question and the answer have less than 2 overlapping content words.
- The question or the answer contains abusive words from Google's "what do you need" project[7].
- The question or the answer has emoticons[8].
- The question or the answer has 3 consecutive punctuation.
- The question or the answer has 3 consecutive fully uppercased words.
- The question has more than 90% of title-case words or the answer has more than 30% of title-case words.
- The question has more than 1 unique word not in the English dictionary or the answer has more than 2 unique words not in the English dictionary[9].

Table 6: Rules for filtering out ill-formed question-answer pairs on both Yahoo and Reddit.

Question Type Annotation. We include the definition and corresponding examples for each question type in the annotation guideline, as shown in Table 12. We allow annotators to label a question with two types if they cannot decide between the two. All recruited annotators are U.S. college students, and are paid $15 per hour for the task. On average, it takes 3.5 hours to annotate 1000 questions.

[7] https://gist.github.com/jamiew/1112488

[8] https://en.wikipedia.org/wiki/List_of_emoticons

[9] https://github.com/dwyl/english-words

For samples with disagreed labels, we check whether agreement can be reached by considering both labeled types. For example, if annotator A labels VERIFICATION and JUDGMENTAL, and annotator B labels JUDGMENTAL, the agreed-upon type is JUDGMENTAL. We then resolve outstanding disagreements by discussion.

B Details for Question Type Classifiers

To train the question type classifier γ_q that reads the question as input, we split the collected question type dataset into training, validation, and test sets. Sample counts and question type distributions for different data splits are shown in Table 7.

Question Type	**Training**	**Validation**	**Test**
VERIFICATION	445	58	72
DISJUNCTIVE	156	36	36
CONCEPT	289	46	54
EXTENT	274	49	48
EXAMPLE	871	152	139
COMPARISON	162	25	30
CAUSE	475	69	91
CONSEQUENCE	102	14	11
PROCEDURAL	469	63	85
JUDGMENTAL	476	68	94
ALL	3719	580	660

Table 7: Sample counts and question type distributions for the newly labeled dataset with question types.

We then use γ_q to identify types for unlabeled questions in Yahoo and Reddit. The question type distributions for the two datasets are shown in Table 8.

	Yahoo		*Reddit*	
Question Type	**#**	**%**	**#**	**%**
VERIFICATION	48,577	16.7	43,801	6.1
DISJUNCTIVE	5,131	1.8	7,179	1.0
CONCEPT	44,963	15.5	34,744	4.8
EXTENT	16,811	5.8	19,217	2.7
EXAMPLE	29,494	10.1	315,478	43.8
COMPARISON	13,167	4.5	14,727	2.0
CAUSE	35,010	12.0	128,204	17.8
CONSEQUENCE	5,547	1.9	16,542	2.3
PROCEDURAL	57,762	19.9	66,662	9.3
JUDGMENTAL	34,149	11.8	73,434	10.2

Table 8: Distributions of question types for the two datasets as labeled by γ_q.

C Details for Graph Construction

We discard secondary dependency relations for graph construction, including *case, mark, cc, cc:preconj, aux, aux:pass, cop, det, discourse, expl,*

det:predet, punct, ref. The definition for each dependency can be found in Universal Dependency.[10]

D Details for Templates and Exemplars

Template Construction. To avoid replacing words that are representative of question types during template construction, we maintain a list of words not to be replaced for each question type, as shown in Table 9. These words are identified by frequency with additional manual inspection.

Question Type	Words Not to Be Replaced
VERIFICATION	-
DISJUNCTIVE	*or*
CONCEPT	*mean*
EXTENT	*many, much, long, take, get*
EXAMPLE	*good, best, find, anyone, get*
COMPARISON	*difference, best, better, and, or*
CAUSE	*people*
CONSEQUENCE	*happen, happens, would, affect, effect, effects*
PROCEDURAL	*get, way, make, best, know*
JUDGMENTAL	*think, would, like, anyone*

Table 9: Words not to be replaced during template construction, per question type.

Exemplar Collection. Table 10 lists the collected template exemplars for different question types.

Exemplar Classifiers. To predict the exemplars used for question decoding, we train one exemplar classifier for each question type, on each dataset. Accuracy values of these exemplar classifiers on the reserved test sets are listed in Table 11.

E Details for Implementation

We use Fairseq (Ott et al., 2019) to build our models and conduct training and decoding. For the Graph Attention Networks (GATs) in our focus predictor, we adopt the implementation by PyTorch Geometric (Fey and Lenssen, 2019). All our experiments are conducted on a Quadro RTX 8000 GPU with 48 GB of memory.

Training Settings. We use Adam (Kingma and Ba, 2014) for the training of all our models. Our question type classifiers and template exemplar classifiers are trained with a maximum learning rate of 1×10^{-5} and a batch size of 32. For training generation models, the maximum learning rate is 3×10^{-5} and each batch contains at most 32,768 tokens. Mixed-precision training is adopted for all

[10] https://universaldependencies.org/

Question Type	Template Exemplars
VERIFICATION	"Is [NP] [NP]?", "Is there [NP]?", "Is [NP] [ADJP]?", "Can [NP] [V] [NP]?", "Do [NP] [V] [NP]?", "Does anyone have [NP]?", "Is it [ADJP] to [V] [NP]?"
DISJUNCTIVE	"Is [NP] [NP] or [NP]?", "Is [NP] [ADJP] or [ADJP]?", "Who is [NP] or [NP]?", "What came [ADVP] [NP] or [NP]?", "Which is [NP] or [NP]?", "What is [NP] or [NP]?"
CONCEPT	"What is [NP]?", "What does [NP] mean?", "Who is [NP]?", "Where is [NP]?", "What is the meaning of [NP]?", "What does [NP] do?", "What do you know about [NP]?", "When is [NP]?", "What is meant by [NP]?", "Where did [NP] come from?", "Which is [NP]?", "When was [NP] [V]?", "What is the definition of [NP]?", "How is [NP]?", "Does anyone know anything about [NP]?", "What happened to [NP]?"
EXTENT	"What is [NP]?", "How [OTHER] is [NP]?", "How many [OTHER] are in [NP]?", "How many [NP]?", "How much does [NP]?"
EXAMPLE	"What are [NP]?", "What is a good [NP]?", "What is the best [NP]?", "Where can I [V] [NP]?", "What are some good [NP]?", "Does anyone have [NP]?"
COMPARISON	"What is the difference between [NP] and [NP]?", "What is the best [NP]?", "What is [NP]?", "Which is better [NP] or [NP]?", "Who is the best [NP]?"
CAUSE	"Why is [NP]?", "Why is [NP] [ADJP]?", "Why do [NP]?", "Why do [NP] [V] [NP]?", "What causes [NP]?", "Why do [NP] [V]?"
CONSEQUENCE	"What are [NP]?", "How does [NP] affect [NP]?", "What are [NP] effects [NP]?", "What are the benefits of [NP]?"
PROCEDURAL	"How do I [V] [NP]?", "How to [V] [NP]?", "How do you [V] [NP]?", "What is the best way to [V] [NP]?", "How is [NP] [V]?", "How does [NP] [V] [NP]?", "How do [NP] work?"
JUDGMENTAL	"What is [NP]?", "What do you think of [NP]?", "Do you believe in [NP]?", "Do you like [NP]?", "Should I [V] [NP]?", "Is [NP] [NP]?", "Do you [V] [NP]?", "Who is [NP]?", "Are you [NP]?", "Should [NP] [V] [NP]?"

Table 10: Template exemplars for different question types.

Question Type	Yahoo Acc	Reddit Acc
VERIFICATION	43.04	38.09
DISJUNCTIVE	65.21	47.50
CONCEPT	43.43	43.84
EXTENT	45.56	26.99
EXAMPLE	46.06	50.26
COMPARISON	53.25	54.33
CAUSE	54.58	43.77
CONSEQUENCE	38.94	26.57
PROCEDURAL	43.66	29.55
JUDGMENTAL	28.31	28.24

Table 11: Accuracy of exemplar classifiers for different question types on Yahoo and Reddit.

models except for models with GATs.

Decoding Settings. We use beam search for decoding. A beam size of 5 and a length penalty of 1.5 are used for all models. Repeated trigram blocking is applied to question generation. The minimum and maximum lengths for generation are set to 1 and 100, respectively.

Model Parameters. The question type classifiers and template exemplar classifiers are based on RoBERTa$_{\text{Large}}$, which has 355M parameters. Our generation model builds a GAT upon the BART model, containing 430M parameters in total.

Running Time. Training question type classifiers takes 23 hours. Due to the difference in training data size, the training time for template exemplar classifiers ranges from 20 minutes to 3 hours. For our generation model with focus prediction, it takes 6 hours to train on Yahoo and 12 hours to train on Reddit. Decoding on the test set of Yahoo and Reddit takes 8 minutes and 15 minutes, respectively.

In this study, you are asked to annotate the question types for 1000 questions. The question type reflects the nature of the question. It is **not determined by the interrogative word** of the question. There are 10 question types in total. The definition for each type is shown in the following Table, along with examples per question type. During annotation, you can label two most-confident types when no clear decision can be made for the most probable type.
VERIFICATION: Asking for the truthfulness of an event or a concept. - *"Is Michael Jackson an African American?"* - *"Does a Mercedes dealer have to unlock a locked radio?"* - *"Could stress, anxiety, or worry cause cholesterol levels to rise?"*
DISJUNCTIVE: Asking for the true one given multiple events or concepts, where comparison among options is not needed. - *"Is Michael Jackson an African American or Latino?"* - *"Is a DVI to HDMI cable supposed to transmit audio and video or just video?"* - *"When you get a spray-on tan does someone put it on you or does a machine do it?"*
CONCEPT: Asking for a definition of an event or a concept. - *"Who said the sun never sets on the British empire?"* - *"Where do dolphins have hair at?"* - *"What is the origin of the phrase "kicking the bucket"?"*
EXTENT: Asking for the extent or quantity of an event or a concept. - *"How long does gum stay in your system?"* - *"What is Barry Larkin's hat size?"* - *"To what extent is the Renewable Fuel Standard accurate nationwide?"*
EXAMPLE: Asking for example(s) or instance(s) of an event or a concept. - *"What are some examples to support or contradict this?"* - *"Where can I get my teeth examined around Los Angeles?"* - *"What countries/regions throughout the world do not celebrate the Christmas holidays?"* - *"What is the best goal or win you have ever made in a sport?"*
COMPARISON: Asking for comparison among multiple events or concepts. - *"How does an electric violin "play" differently than an acoustic violin?"* - *"What is the best tinted facial moisturizer?"* - *"In what hilariously inaccurate ways is your job/career portrayed on television or in movies?"* - *"Which is better, Nike or Adidas?"*
CAUSE: Asking for the cause or reason for an event or a concept. - *"How does the D.M.V. decide the first letter of the California driver's license?"* - *"Why are parents strick on girls than boys?"* - *"What makes nerve agents like "Novichok" so hard to produce and why can only a handful of laboratories create them?"* *"Why is the sky blue?"*
CONSEQUENCE: Asking for the consequences or results of an event. - *"What are the negative consequences for the services if they do not evaluate their programs?"* - *"In the US, what is the benefit of having a red left-turn arrow?"* - *"What would happen if employers violate the legislation?"* - *"What if the Hokey Pokey is really what it's all about?"*
PROCEDURAL: Asking for the procedures, tools, or methods by which a certain outcome is achieved. - *"Why YM 7.5 BETA always stupidly shows me available, although I initially set it to invisible?"* - *"How did the Amish resist assimilation into the current social status in the U.S?"* - *"How astronomers detect a nebula when there are no stars illuminating it?"*
JUDGMENTAL: Asking for the opinions of the answerer's own. - *"Do you think that it's acceptable to call off work for a dying-dead pet?"* - *"Should I date a guy that has an identical twin?"* - *"How old is too old for a guy to still live with his mother?"*

Table 12: Question type annotation guideline.