Constraint-based Neural Question Generation using Sequence-to-Sequence and Transformer models for privacy policy documents

Deepti Lamba and William H. Hsu

***Abstract*—This paper presents the results of constraint-based automatic question generation for paragraphs from privacy policy documents. Existing work on question generation uses sequence-to-sequence and transformer-based approaches. This work introduces constraints to sequence-to-sequence and transformer based T5 model. The notion behind this work is that providing the deep learning models with additional background domain information can aid the system in learning useful patterns. This work presents three kinds of constraints – logical, empirical, and data-based constraint. The constraints are incorporated in the deep learning models by introducing additional penalty or reward terms in the loss function. Automatic evaluation results show that our approach significantly outperforms the state-of-the-art models.**

***Index Terms*— Question generation, constraints, privacy policy, Transformer, Sequence-to-Sequence model.**

TABLE I: SAMPLE CONTEXT-QUESTION-ANSWER TRIPLET FROM POLICYQA DATASET

Context
"The information that you provide is collected by TGI Fridays. In the case of links to our gift card and guest recognition sites, the information you voluntarily provide at those sites will only be shared with those service vendors who help TGI Fridays administer those websites or mobile application and the services they provide. In any case, TGI Fridays is the lawful "owner" of the information and each of these vendors may use the information only for the purpose of administering the digital or mobile application and its services for TGI Fridays and will take all necessary precautions to protect the information. Ownership of any information you provide us will be held solely by TGI Fridays. We **will not** sell ownership of this data to any other company or organization."
Question: Does the third party follow the privacy practice?
Answer: will not

I. INTRODUCTION

Question generation is defined as the task of automatically generating a grammatically, and syntactically correct interrogative sentence based on some context. Some popular use cases of question generation are in education [1]-[2], human-computer interaction [3], and question answering [4]-[6]. In addition to these applications, question generation can also aid in creating data sets for question answering systems. The motivation behind this work is to generate a better class of questions that can be used to build question answering systems for privacy policy domain. Table I shows an example from PolicyQA data set [7] that is used for generating questions in this work. The example is an excerpt from the privacy policy document of TGI Fridays.

Most existing work in question generation has focused on sequence-to-sequence models [9]-[19]. Recently, some researchers have used transformer models [20]-[23] for this task. This work on question generation focuses on incorporating additional knowledge, expressed using constraints, with existing sequence-to-sequence models and a transformer-based (T5) [24] approach. The performance of deep learning models is dependent on the amount of data available to them for learning useful patterns. Their ability to utilize vast amounts of data to learn patterns is a key factor behind their success. However, the usage of small data sets can lead to sub-optimal results. Since it is quite challenging to acquire well annotated data sets in domains such as privacy policy domain, we explore constraints as means to supplement the knowledge available to our models. The addition of constraints also contributes to the interpretability of deep learning models which otherwise are considered black boxes. The baselines discussed in Lamba and Hsu (2021) [8] were considered while designing constraints. These constraints served as input during the model training phase and the results produced in this work demonstrate their effectiveness. For each of the three models, an improvement in BLEU-n, METEOR, and ROUGE-L scores is seen.

The novel contributions of this research are listed below:

• To our best knowledge, this work is the first to use constraints for question generation. Our work studies the effects of different kinds of constraints: logical, empirical, and data based. It also analyses the effect of hybrid constraints.

• The approach presented in this paper has successfully boosted the performance of the models over the baselines by a significant margin. For each model, results indicate consistent improvement in the ROUGE-L, METEOR, and BLEU-n scores over the baseline results.

• This work is an extension of Lamba and Hsu (2021) [8] and is the first known attempt to generate questions in the privacy policy domain.

Manuscript received September 10, 2021.
Deepti Lamba was a Ph.D. student at Kansas State University, Manhattan, KS 66506 USA (e-mail: dlamba@ksu.edu); William H. Hsu is with Kansas State University, Manhattan, KS 66506 USA (e-mail: bhsu@ksu.edu).

II. Related Work

A. Question Generation

Prior to the year 2017, question generation was studied using rule-based approaches. Some prominent work that used rule-based systems was presented by Mitkov and Ha (2003) [26], Gates (2008) [27], Heilman and Smith (2009) [29], Chali and Hasan (2015) [30], Khullar et al. (2018) [28], and Dhole and Manning (2020) [31]. Rule-based systems required immense human effort to construct rules, which are seldom generalizable to other domains. However, these systems offer some benefits in terms of increased model interpretability and provide more control over the model to developers. Du et al. (2017) [9] presented the first work that used deep learning to generate questions. They used an attention-based [33] sequence-to-sequence model [32] with context-question pair as input, without making use of the answer sequence for training the model.

Broadly, answer-aware and answer-unaware models [34] are two categories of neural question generation systems. Du et al. (2017) [10] presented a two-step answer-unaware approach. The first step is identifying question worthy sentences and the second is using those sentences to generate questions. Zhou et al. (2017) [11] presented an attention-based sequence-to-sequence model that used additional information besides the context-question pair. This information includes answer position, and other linguistic features like part-of-speech [42] and named entity tags [41]. Answer-aware models tend to include answer words in the predicted question which is a major drawback. Kim et al. (2019) [12] used a special token to mask the answer in the paragraph. Song et al. (2018) [13] encoded the answer and source text separately. Hu et al. (2018) [14] focused on topic-specific question generation where the topic was provided as additional information to the model.

Sequence-to-sequence models for question generation have shown an improvement in model performance when additional information is provided as input to the model. Harrison and Walker (2018) [18] used additional information obtained from linguistic features like, word case, named entity tags and entity co-reference resolution. Cao et al. (2020) [19] also incorporated additional information from data to train an attention-based sequence-to-sequence model with copy mechanism [43]. Ma et al. (2020) [44] used answer position, alphabetic case, named entity and part of speech tags as auxiliary input.

Question generation has also been approached as a two-step process, where the generation of the interrogative word precedes the generation of the remaining text. One such work is presented by Sun et al. (2018) [15] who used the answer type to generate an interrogative word and then used the relative distance between words given in the context and answer to generate the remaining words of the question. Kang et al. (2019) [16] present a similar work with two modules, one for classifying the interrogative word and the second to generate the remaining question using the generated interrogative word. In our work, we focus on adding additional domain information articulated as constraints, to sequence-to-sequence and T5 transformer model, as opposed to using linguistic features like parts-of-speech tags, alphabetic case information, and answer text/position.

B. Constraint-based Approaches

Constraints can be designed by experts and enable the deep neural model to be on par with human reasoning and understanding. Domain knowledge can be expressed in different ways for deep learning networks. Dash et al. (2021) [45] presented two major categories for representing domain knowledge for deep leaning models: logical constraints and numerical constraints.

Borghesi et al. (2020) [46] presented a survey on ways to incorporate domain knowledge in neural networks. Their survey states that domain knowledge can be represented in multiple ways including algebraic equations and graphs. Experts can create specific features for any domain and express them using either propositional logic or first-order logic with the goal to constrain the neural network by means of parameters or structure. The earliest work that used propositional logic to express constraints in neural networks can be traced back to the early 90's by Towell et al. (1990) [47] and Fu (1193) [48]. Their work became popular, at the time, despite the inability of the proposed neural network models to learn new rules. A more recent approach was given by Xu et al. (2018) [49] to represent constraints using propositional rules. Their approach used a loss function to calculate the dissimilarity between constraints and model output.

Constraints have also been expressed using first-order logic rules: Sikka et al. (2020) [54] used it to represent declarative knowledge that is incorporated while training the model. However, incorporating rules into neural networks presents some challenges. Li and Srikumar (2019) [50] discussed three major challenges, which are: mapping rules to actual network nodes; cyclic dependencies in the network introduced by logical rules; and finally, the issue of logic not being differentiable. Yao et al. (2021)) [51] refined BERT using logical rules to include human explanations. Giunchiglia and Lukasiewicz (2021) [55] proposed an approach that expressed constraints as logic rules and defined a loss function to incorporate them. Recently, Silvestri et al. (2021) [61] combined semantics-based regularization [52] and constraint programming [53] to inject knowledge into a deep learning model.

Second category of constraints is numerical constraints [45] which can be expressed in multiple ways, including: as a loss function; as constraints on weight; and through regularization. Aghaebrahimian (2017) [56] applied a constraint on the number of shared patterns between sentences and questions for a deep learning-based question answering system by defining a loss function. In the case of our deep learning models, we have used loss function augmented with additional penalty or reward terms to impose constraints.

Knowledge has also been added to neural networks through weight-based constraints. One such work is presented by Hu et al. (2016) [57] who developed a method to transfer information from logic rules to the actual neural network via network weights. Jiang et al. (2020) [58] combined regular expressions with neural networks to create

weighted finite state automata. Another technique that can add numerical constraints to a deep learning model is called regularization. It is applied by adding penalty terms, like L1 norm and L2 norm, to the objective function. This limits the capacity of the deep learning model.

III. Methodology

A. Problem Statement

More formally, question generation task is defined as: Given a passage from a privacy policy document as input, $X_{passage} = (x_1, x_2, \dots, x_n)$, the deep learning model generates a question represented by $Y = (y_1, y_2, \dots, y_T)$. The goal of this task is to find the best $\bar{Y}$ using Equation 1:

$$\bar{Y} = \underset{x}{\operatorname{argmax}} P\left(Y \mid X_{passage}\right) \quad (1)$$

where $P\left(Y \mid X_{passage}\right)$ is the conditional log-likelihood of the predicted question Y, given input $X_{Passage}$. This work does not use the answer sequence as input in any way, neither the actual sequence nor its position. This work uses domain knowledge to train several deep learning models to produce semantically and syntactically correct questions. The domain knowledge is expressed as constraints that the deep learning network needs to obey. A violation of the constraint penalizes the loss function, or in case of data constraint a reward function is defined to reward the network for satisfying the constraint. The objective function is modified to introduce a new knowledge-based loss term $Loss_C$ for each constraint being used. The new objective function is given below:

$$Loss = \underset{x}{\operatorname{argmin}}\, Loss(Y, \bar{Y}) + \lambda_1 x Loss_{C1}(\bar{Y}) + \lambda_2 y Loss_{C2}(\bar{Y}) - \lambda_3 z Loss_{C3}(\bar{Y}) \quad (2)$$

where, x and y have a value of 0 when a constraint is satisfied and 1 otherwise. The λ terms in Equation (2) are hyperparameters that denote the weights of x, y and z in the objective function. z denotes the function given by Equation (3) below:

$$z = \frac{1}{n}\left[\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n\right] \quad (3)$$

where λ_i is an empirically learnt term that provides weight to the i^{th} term in the itemset Z. x_i is assigned the value 1 if the i^{th} term is present in a generated question, otherwise x_i is set to 0. This is explained further in the paragraph on domain constraints below.

B. Proposed Constraints

Logical Constraint

The first type of constraint that has been designed for this work is a logical constraint that is the outcome of human knowledge. This constraint uses named entity tags presented in Lamba and Hsu (2021) [8] and forces the model to have at least one named entity term in the generated question. The core idea here is to focus on the key entity terms identified in the passage and formulate questions focusing on those terms.

The absence of a named entity term imposes the logical constraint on the objective function which assigns a value of one to function, zero otherwise. For brevity, this constraint will be referred to as C1 and is defined as follows: Let N be a set of all named entities and Y be a set of all words in a predicted question, such that $Y = (y_1 + y_2 + \cdots + y_n)$. Then, C1 is given by the following binary function in Equation (4):

$$f(c_1) = \begin{cases} 0 & if\ \exists t \in [1,n],\ such\ that\ y_t \in N \\ 1 & if\ \forall t \in [1,n],\ such\ that\ y_t \notin N \end{cases} \quad (4)$$

Empirical Constraint

Based on empirical results presented in Lamba and Hsu (2021) [8], it was observed that underperforming models are predisposed to output the same word in consecutive positions in the question. For example, "Do you you collect collect my data?". This constraint checks for duplicated consecutive terms and if found, the value of the function is 1, otherwise it is 0. The constraint is represented by Equation (5).

$$f(c_2) = \begin{cases} 1 \text{ if } \exists t \in [2,n], \text{such that } y_{t-1} = y_t \\ 0 \text{ if } \forall t \in [2,n], \text{such that } y_{t-1} \neq y_t \end{cases} \quad (5)$$

where Y is a set of all words in a predicted question such that $Y = (y_1 + y_2 + \cdots + y_n)$. This constraint will hereon be referred to as C2.

Domain Constraint

The third type of constraint is learnt directly from the context data, and it is referred to as domain constraint. In this work, we mine frequent item-sets using the Apriori algorithm [35] and then use 1 of the item-sets to design a constraint to inject information into the models. We experimented with several values of support and confidence and the best values were obtained empirically. This constraint, unlike the other two, does not penalize the objective function. Instead, it rewards the objective function if it is satisfied. This constraint will be referred to as C3 and is given by Equation (3). The value of $z \in [0,1]$ such that if all words of the item set of size n, represented by Z, are present in a predicted question then the objective function will receive maximum reward.

C. Deep Learning Models

This section first discusses the Sequence-to-sequence models, followed by the transformer model used in this work. The sequence-to-sequence models use a gated recurrent unit (GRU) [59] encoder-decoder with paragraph level encoding. This work uses T5 Transformer model [24] that has been pretrained on C4 corpus [24] and is fine tuned for our data.

Encoder

Similar to work of Zhou et al. (2017) [11], we provide a concatenation of the word vector with the label embedding vector as input to the encoder. The input is used to compute the encoder hidden state at time t, which is represented by h_t, using the equation below:

$$h_t = f(W^h h_{t-1} + W^x x_t) \quad (6)$$

where, h_{t-1} represents the previous hidden state of the encoder, x_t is the word at time t, and W represents the weight matrices for the hidden states and the input, respectively. We use a unidirectional GRU encoder to produce the hidden vector for the decoder.

Decoder

The decoder accepts the hidden state from the previous unit to predict a sequence of words in the question. The decoder output at time t is represented by y_t and can be computed using the equation below:

$$y_t = f(W^h h_t) \quad (7)$$

where, h_t represents decoder hidden state at time t and W^h represents the weight matrix.

Decoder with attention

This work also uses a GRU decoder with attention [33] to generate the sequence of words in the question.

T5 Model

T5 stands for Text-to-Text Transfer Transformer and was proposed by Raffel et al. (2020) [24]. This work uses the implementation of T5 provided by the Hugging Face library [64] for training T5-small model [24]. The architecture of T5 is similar to one proposed by Vaswani et al. (2017) [60] with a few architectural modifications.

D. Data Preparation

The data set used in this work is PolicyQA [7] which consists of over 25000 examples (context-question-answer tuples). The data set was first shuffled and then divided into training (~80%), development (~10%), and test (~10%) sets. As part of data preprocessing, the following steps were performed: (1) The entire data was changed to lowercase; (2) Named entities [8] consisting of a pair of words were hyphenated to make it a single word; (3) SOS and EOS tokens were appended to all questions in the data; (4) Spelling disparities in the data were fixed. For example, the word "parties" was spelled as "parities"; (5) Shortened words were expanded to their full length, for example, the word "information" was written as "info" in many places. This step was performed to ensure consistency throughout data; and (6) The last letter of the question and question mark were intentionally separated by a space for consistency.

IV. Experimental Setup

This research uses PyTorch version 1.7.1. for all deep learning models. The models have been trained on Nvidia Tesla v100. The encoder-decoder hidden sizes were alternated between 256, 500, 1000, and 2000. Results presented in Lamba and Hsu (2021) [8] showed that greedy decoding produces better results for this task, hence greedy decoding has been used for all models. SGD with learning rate 0.001 was used for optimization. For training, teacher forcing [62] for sequence-to-sequence models was used. Lowest perplexity on development set was used to select the best model.

A. Baseline Models

The baselines for this work have already been discussed in Lamba and Hsu (2021) [8]. These baselines are listed below:

Seq2Seq: It is a basic sequence-to-sequence model [32] that uses a GRU encoder-decoder model. The input to this model was the context concatenated with named entity tags and the text was not reversed for the experiments. This model does not use any pre-trained word embedding.

Seq2Seq+attention: A GRU encoder-decoder model with Bahdanau attention [33]. This model does not use any pre-trained embedding and uses context concatenated with named entities as input.

NQG (2017) [9]: This uses an attention-based LSTM encoder-decoder model to generate a question from a given context. Experiments were conducted with sentence level and paragraph level encoder. The best results were obtained using paragraph level encoder with GloVe [63] pre-trained embedding.

Transformer-based model (T5): T5-small model is fine-tuned for the task of question generation on privacy policy documents. This model consists of 60 million parameters. The input to the model is composed of a concatenation of policy passage with named entities described in Lamba and Hsu (2021) [8].

B. Evaluation Metrics

The evaluation package by Chen et al. (2015) [40] is used for evaluating the predicted questions. The questions are evaluated using BLEU-n (Papineni et al., 2002) [37], METEOR (Lavie & Denkowski, 2009) [38], and ROUGE-L (Lin, 2004) [39].

V. Results and Discussion

The baseline results of Lamba and Hsu (2020) [8] along with results of constraint-based models are presented in Table II. The table is divided into 4 sections for ease of understanding. The first section presents the results of baseline NQG (2017) from Lamba and Hsu (2021) [8]. The next part of the table shows the results of baselines Seq2Seq, followed by Seq2Seq

TABLE II: Evaluation Results (in percentage) for Sequence-to-Sequence and Transformer models with Greedy Decoding

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
NQG (2017)	32.66	18.27	12.73	9.94	15.54	30.63
Seq2Seq (Lamba and Hsu, 2021 [8])	27.22	15.52	8.63	5.12	14.23	30.16
Seq2Seq+NER (Lamba and Hsu, 2021 [8])	24.66	14.33	8.96	5.80	14.93	33.07
Seq2Seq+NER+C1	28.41	16.38	9.67	7.04	16.22	**35.85**
Seq2Seq+NER+C2	31.08	**18.67**	**11.68**	**8.53**	17.79	35.18
Seq2Seq+NER+C1+ C2	**31.11**	18.66	11.63	8.49	**17.80**	35.13
Seq2Seq+Attn (Lamba and Hsu, 2021 [8])	28.09	16.00	9.86	6.30	14.96	31.84
Seq2Seq+Attn+NER (Lamba and Hsu, 2021 [8])	28.68	16.24	9.95	6.87	15.81	31.51
Seq2Seq+Attn+NER+C1	31.11	**18.66**	11.62	8.48	**17.80**	35.15
Seq2Seq+Attn+NER+C2	28.40	16.38	9.67	7.04	16.22	**35.85**
Seq2Seq+Attn+NER+C3	**31.12**	**18.66**	11.62	8.48	**17.80**	35.15
Seq2Seq+Attn+NER+All	31.10	**18.66**	**11.63**	**8.49**	**17.80**	35.12
T5-small (Lamba and Hsu, 2021 [8])	31.32	17.14	11.51	8.53	18.29	31.02
T5-small+NER (Lamba and Hsu, 2021 [8])	**32.98**	**18.22**	11.89	**8.49**	**18.74**	32.16
T5-small+NER+C1	29.02	14.88	9.26	6.48	18.23	28.38
T5-small+NER+C2	29.18	14.43	9.01	6.42	17.40	27.96
T5-small+NER+All 3	31.99	18.10	**11.92**	8.42	18.56	**32.85**

+NER (input concatenated with named entity tags) from Lamba and Hsu (2021) [8], and finally the results showing the effects of constraints C1, C2, and a combination of the two constraints on the model.

The results in section two show that the application of C1 produces an improvement across all evaluation metrics with the best ROUGE-L score. ROUGE-L obtained in this case shows an 8.4% increase over Seq2Seq+NER and an 18.9% improvement over the basic Seq2Seq model. The highest METEOR score is achieved when both C1 and C2 are applied to the Seq2Seq model. This brings a 25.1% improvement over the basic Seq2Seq model. The application of C1 and C2 also shows improvement across all BLEU-n scores. The application of any kind of constraint shows a considerable improvement over Seq2Seq and outperforms NQG (2017) [9] model for ROUGE-L and METEOR.

The third part of the table shows the results of attention based Seq2Seq models. The trends observed here are like the ones presented in the second part of the table. The application of constraints provides a boost to results as compared to the baseline models. The application of all three constraints individually and when combined beats the baselines. The application of C2 gives the best METEOR score which shows a 12.59% improvement over Seq2Seq+attention and 13.77% improvement over Seq2Seq+attention with named entity labels. Based on the results produced by the sequence-to-sequence models, we can see that the application of constraints provides a boost to the results.

The last section of Table II shows the results of T5 model. Results show that application of C1 and C2 by themselves do not show any improvement in scores. The last row in the table shows that when all three constraints are applied to the model then ROUGE-L beats the T5 baselines and METEOR is close to the T5+NER baseline. However, the improvement produced by the application of constraints to T5 are not at par with the improvements produced in the sequence-to-sequence models. We further analyze the questions produced by T5 model and for this we randomly select some of the predicted questions. Table III shows two examples of predicted questions from our study set along with the actual questions. The predicted questions are correct, complete, and syntactically correct. The first ground truth question in Table III asks whether the organization shares customer data with others and the generated question is clearer and more precise and is an improvement over the actual question. The two questions given in Table III are evaluated using the evaluation metrics and they produce a METEOR score of 30% and ROUGE-L of 45.35%. There are many more examples from the predicted questions that suggest that low scores of T5 can

TABLE III: Questions predicted using Transformer Model

Ground Truth: Do you share my information with others?
Predicted Question: Does the company share user's information with a third-party?
Ground Truth: Does you collect my information to enhance or personalize my experience?
Predicted Question: Does the company use user's information for customized services?

be attributed to the inability of existing metrics not being able to capture the nuances of question generation.

VI. Conclusion and Future Work

The experimental results discussed in this work establish that a constraint-based approach to question generation is effective. The results also provide a performance comparison between sequence-to-sequence models and a T5 model. This work is a continuation of the work discussed in Lamba and Hsu (2021) [8]. The addition of constraints to three different models shows an improvement in performance as measured by ROUGE-L, METEOR, and BLEU-n scores. Thus, this work paves the way for future research in question generation. This work can be extended in the future to incorporate more constraints and experiment exhaustively with combination of constraints presented in this work. This research uses an out-of-the-box pre-trained T5 model [24], which in future could be redesigned for the domain and task. The pre-training could be done on legal documents that are close to privacy policies. Another logical extension of this work is the exploration of different ways of representing constraints and incorporating them in deep learning models.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

This research has already been published as part of Deepti Lamba's Ph.D. dissertation [25] conducted at Kansas State University under the supervision of Dr. William H. Hsu.

References

[1] Heilman, Michael, and Noah A. Smith. "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*, pp. 609-617. 2010.

[2] Kurdi, Ghader, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. "A systematic review of automatic question generation for educational purposes." *International Journal of Artificial Intelligence in Education* 30, no. 1 (2020): 121-204.

[3] Mostafazadeh, Nasrin, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. "Generating Natural Questions About an Image." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1802-1813. 2016.

[4] Wang, Tong, Xingdi Yuan, and Adam Trischler. "A joint model for question answering and question generation." *arXiv preprint arXiv:1706.01450* (2017).

[5] Tang, Duyu, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. "Question answering and question generation as dual tasks." *arXiv preprint arXiv:1706.02027* (2017).

[6] Duan, Nan, Duyu Tang, Peng Chen, and Ming Zhou. "Question generation for question answering." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 866-874. 2017.

[7] Ahmad, Wasi, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. "PolicyQA: A Reading Comprehension Dataset for Privacy Policies." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 743-749. 2020.

[8] Lamba, Deepti and Hsu, William H. Answer-Agnostic Question Generation in Privacy Policy Domain using Sequence-to-Sequence and Transformer Models. 2021 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2021. To appear.

[9] Du, Xinya, Junru Shao, and Claire Cardie. "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), pp. 1342-1352. 2017.

[10] Du, Xinya, and Claire Cardie. "Identifying where to focus in reading comprehension for neural question generation." In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2067-2073. 2017.

[11] Zhou, Qingyu, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. "Neural question generation from text: A preliminary study." In National CCF Conference on Natural Language Processing and Chinese Computing, pp. 662-671. Springer, Cham, 2017.

[12] Kim, Yanghoon, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. "Improving neural question generation using answer separation." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6602-6609. 2019.

[13] Song, Linfeng, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. "Leveraging context information for natural question generation." In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 569-574. 2018.

[14] Hu, Wenpeng, Bing Liu, Rui Yan, Dongyan Zhao, and Jinwen Ma. "Topic-Based Question Generation." (2018).

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

[16] Kang, Junmo, Haritz Puerto San Roman, and Sung-Hyon Myaeng. "Let Me Know What to Ask: Interrogative-Word-Aware Question Generation." In Proceedings of the 2nd Workshop on Machine Reading for Question Answering, pp. 163-171. 2019.

[17] Ma, Xiyao, Qile Zhu, Yanlin Zhou, and Xiaolin Li. "Improving question generation with sentence-level semantic matching and answer position inferring." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8464-8471. 2020.

[18] Harrison, Vrindavan, and Marilyn Walker. "Neural Generation of Diverse Questions using Answer Focus, Contextual and Linguistic Features." In Proceedings of the 11th International Conference on Natural Language Generation, pp. 296-306. 2018.

[19] Cao, Zhen, Sivanagaraja Tatinati, and Andy WH Khong. "Controllable Question Generation via Sequence-to-Sequence Neural Model with Auxiliary Information." In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, 2020.

[20] Matsumori, Shoya, Kosuke Shingyouchi, Yuki Abe, Yosuke Fukuchi, Komei Sugiura, and Michita Imai. "Unified Questioner Transformer for Descriptive Question Generation in Goal-Oriented Visual Dialogue." arXiv preprint arXiv:2106.15550 (2021).

[21] Scialom, Thomas, Benjamin Piwowarski, and Jacopo Staiano. "Self-attention architectures for answer-agnostic neural question generation." In Proceedings of the 57th annual meeting of the Association for Computational Linguistics, pp. 6027-6032. 2019.
[22] Chan, Ying-Hong, and Yao-Chung Fan. "A recurrent BERT-based model for question generation." In Proceedings of the 2nd Workshop on Machine Reading for Question Answering, pp. 154-162. 2019.
[23] Varanasi, Stalin, Saadullah Amin, and Günter Neumann. "CopyBERT: A Unified Approach to Question Generation with Self-Attention." In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pp. 25-31. 2020.
[24] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." Journal of Machine Learning Research 21 (2020): 1-67.
[25] Lamba, D. Deep learning with constraints for answer-agnostic question generation in legal text understanding. PhD thesis, Department of Computer Science, Kansas State University, USA,2021.
[26] Mitkov, Ruslan. "Computer-aided generation of multiple-choice tests." In Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing, pp. 17-22. 2003.
[27] Gates, Donna M. Automatically generating reading comprehension look-back strategy: Questions from expository texts. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2008.
[28] Khullar, Payal, Konigari Rachna, Mukul Hase, and Manish Shrivastava. "Automatic question generation using relative pronouns and adverbs." In Proceedings of ACL 2018, Student Research Workshop, pp. 153-158. 2018.
[29] Heilman, Michael, and Noah A. Smith. Question generation via overgenerating transformations and ranking. Carnegie-Mellon Univ Pittsburgh pa language technologies insT, 2009.
[30] Chali, Yllias, and Sadid A. Hasan. "Towards topic-to-question generation." Computational Linguistics 41, no. 1 (2015): 1-20.
[31] Dhole, Kaustubh, and Christopher D. Manning. "Syn-QG: Syntactic and Shallow Semantic Rules for Question Generation." In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 752-765. 2020.
[32] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." In Advances in neural information processing systems, pp. 3104-3112. 2014.
[33] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[34] Cao, Zhen, Sivanagaraja Tatinati, and Andy WH Khong. "Controllable Question Generation via Sequence-to-Sequence Neural Model with Auxiliary Information." In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, 2020.
[35] Agrawal, Rakesh, and Ramakrishnan Srikant. "Fast algorithms for mining association rules." In Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, pp. 487-499. 1994.
[36] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).
[37] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation." In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311-318. 2002.
[38] Lavie, Alon, and Michael J. Denkowski. "The METEOR metric for automatic evaluation of machine translation." Machine translation 23, no. 2-3 (2009): 105-115.
[39] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81. 2004.
[40] Chen, Xinlei, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco captions: Data collection and evaluation server." arXiv preprint arXiv:1504.00325 (2015).
[41] Sang, Erik F., and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." arXiv preprint cs/0306050 (2003).
[42] Brill, Eric. A simple rule-based part of speech tagger. PENNSYLVANIA UNIV PHILADELPHIA DEPT OF COMPUTER AND INFORMATION SCIENCE, 1992.
[43] Gulcehre, Caglar, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. "Pointing the unknown words." arXiv preprint arXiv:1603.08148 (2016).
[44] Ma, Xiyao, Qile Zhu, Yanlin Zhou, and Xiaolin Li. "Improving question generation with sentence-level semantic matching and answer position inferring." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8464-8471. 2020.
[45] Dash, Tirtharaj, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. "Incorporating Domain Knowledge into Deep Neural Networks." arXiv preprint arXiv:2103.00180 (2021).
[46] Borghesi, Andrea, Federico Baldo, and Michela Milano. "Improving deep learning models via constraint-based domain knowledge: a brief survey." arXiv preprint arXiv:2005.10691 (2020).
[47] Towell, Geoffrey G., and Jude W. Shavlik. "Knowledge-based artificial neural networks." Artificial intelligence 70, no. 1-2 (1994): 119-165.
[48] Fu, Li-Min. "Knowledge-based connectionism for revising domain theories." IEEE Transactions on Systems, Man, and Cybernetics 23, no. 1 (1993): 173-182
[49] Xu, Jingyi, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. "A semantic loss function for deep learning with symbolic knowledge." In International conference on machine learning, pp. 5502-5511. PMLR, 2018.
[50] Li, Tao, and Vivek Srikumar. "Augmenting neural networks with first-order logic." arXiv preprint arXiv:1906.06298 (2019).
[51] Yao, Huihan, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. "Refining neural networks with compositional explanations." arXiv preprint arXiv:2103.10415 (2021).
[52] Diligenti, Michelangelo, Marco Gori, and Claudio Sacca. "Semantic-based regularization for learning and inference." Artificial Intelligence 244 (2017): 143-165.
[53] Rossi, Francesca, Peter Van Beek, and Toby Walsh, eds. Handbook of constraint programming. Elsevier, 2006.
[54] Sikka, Karan, Andrew Silberfarb, John Byrnes, Indranil Sur, Ed Chow, Ajay Divakaran, and Richard Rohwer. "Deep adaptive semantic logic (dasl): Compiling declarative knowledge into deep neural networks." arXiv preprint arXiv:2003.07344 (2020).
[55] Giunchiglia, Eleonora, and Thomas Lukasiewicz. "Multi-Label Classification Neural Networks with Hard Logical Constraints." arXiv preprint arXiv:2103.13427 (2021).
[56] Aghaebrahimian, Ahmad. "Constrained deep answer sentence selection." In International Conference on Text, Speech, and Dialogue, pp. 57-65. Springer, Cham, 2017.
[57] Hu, Zhiting, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. "Harnessing deep neural networks with logic rules." arXiv preprint arXiv:1603.06318 (2016).
[58] Jiang, Chengyue, Yinggong Zhao, Shanbo Chu, Libin Shen, and Kewei Tu. "Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3193-3207. 2020.
[59] Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).
[60] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in neural information processing systems, pp. 5998-6008. 2017.
[61] Silvestri, Mattia, Michele Lombardi, and Michela Milano. "Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem." In International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pp. 266-282. Springer, Cham, 2021.
[62] Williams, Ronald J., and David Zipser. "A learning algorithm for continually running fully recurrent neural networks." Neural computation 1, no. 2 (1989): 270-280.
[63] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
[64] Wolf, Thomas, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac et al. "Transformers: State-of-the-art natural language processing." In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38-45. 2020.