


# Answer-Aware Question Generation from Tabular and Textual Data using T5

<https://doi.org/10.3991/ijet.v16i18.25121>

Saichandra Pandraju<sup>1\*</sup>, Sakthi Ganesh Mahalingam<sup>2\*</sup>

<sup>1</sup> QIS College of Engineering and Technology, Ongole, India

<sup>2</sup> VIT University, Vellore, India

\* Equal Contribution

sakthiganesh@icloud.com

**Abstract**— Automatic Question Generation (AQG) systems are applied in a myriad of domains to generate questions from sources such as documents, images, knowledge graphs to name a few. With the rising interest in such AQG systems, it is equally important to recognize structured data like tables while generating questions from documents. In this paper, we propose a single model architecture for question generation from tables along with text using “Text-to-Text Transfer Transformer” (T5) - a fully end-to-end model which does not rely on any intermediate planning steps, delexicalization, or copy mechanisms. We also present our systematic approach in modifying the ToTTo dataset, release the augmented dataset as TabQGen along with the scores achieved using T5 as a baseline to aid further research.

**Keywords**—Question Generation, T5, Table-to-Text, Transfer Learning

## 1 Introduction

The development of end-to-end supervised Question-Answering (QA) models has been accelerated with the advent of large-scale datasets. The Stanford Question Answering Dataset (SQUAD) [5] is a reading comprehension dataset composed of questions from Wikipedia articles, with the answer to each question being a part of the corresponding reading passage. Microsoft Machine Reading Comprehension (MS MARCO) [6] is a large-scale dataset focused on reading comprehension, question answering, passage ranking, Keyphrase Extraction, and Conversational Search Studies. TriviaQA [7] is a realistic text-based question-answer dataset with 950K question-answer pairings extracted from Wikipedia and the internet. Since the answers to questions may not be simply acquired via span prediction, TriviaQA is more challenging than traditional QA benchmark datasets such as SQuAD. DuoRC [8] comprises 186K distinct question-answer combinations derived from 7680 pairs of movie plots, each pair representing two different versions of the same film and highlights the challenges of combining knowledge and reasoning in neural architectures for reading comprehension.

As is the case, neural-network based solutions could always benefit from more training data, especially in domains where the existing datasets do not cater. Augmenting existing datasets or creating new datasets for specific domains is a time-consuming, tedious, and expensive task. To alleviate this problem and create more training data, there is a growing interest in developing new techniques that can automatically generate questions from a given source like a document [9] [10]. This task is referred to as Automatic Question Generation.

Given the practical importance of AQG systems, it is crucial to consider all forms of data from a document while generating questions. However, current AQG systems are ineffective in generating a large amount of high-quality question-answer pairs from structured data like tables. A tabular structure allows representing complex and vital information in a format that is a lot easier to interpret, ignoring which, would lead to a loss of potential high-quality questions. Let us consider an example of an organization that wants to create a QA database for its policies. It is more likely that financial policies such as ‘Per Diem’ would contain a substantial amount of information like the ‘per day’ expenses for the company across all branches represented as tables. In this case, a typical AQG system would struggle to generate a comprehensive QA dataset by ignoring tabular data. AQG also offers great value in computer-aided assessments [37, 38], allowing the instructors to generate a plethora of questions from existing materials such as presentations and documents.

In this paper, we emphasize that there is a need to improve the existing AQG systems and address the challenges involved while working with tabular data. To the best of our knowledge, the area of Question Generation from tables is not intensively studied, leading to a dearth of academic datasets to explore. But the recent advancements in NLP have led to representing a section of tabular data as natural language, also referred to as Table-to-Text Generation. In this paper, we propose a single model architecture for question generation from textual and tabular data using T5 and also release a modified version of the ToTTo [4] dataset as TabQGen that can be used to generate questions from tables based on the highlighted cells along with its baseline scores to aid further research.

## **2 Related Work**

### **2.1 Question Generation from Text**

Early work on question generation relied on heuristic algorithms to produce questions using manually constructed templates. In [14] proposed to generate self-questioning instructions automatically for a given text by decomposing strategy instruction into describing, modeling, scaffolding, and prompting the strategy. A template-based strategy to online learning [15] proposed the usage of semantic role labels into a system that generated natural language questions automatically. Later approaches also employed semantic pattern recognition to produce questions of various depths and types, as well as semantic role labeling of source sentences to produce both questions and responses in a domain-independent way [16]. Other approaches relied on generating deep

comprehension questions by breaking the task down into an ontology crowd-relevance workflow that included representing the original text in a low-dimensional ontology, crowdsourcing candidate question templates aligned with that space and ranking potentially relevant templates for a novel region of text [17]. Despite the fact that [18] used an over-generate-and-rank strategy followed by learning to rank the questions, the performance of the system is still heavily reliant on the manually constructed generation rules. To examine the profound relationship between language and image, [19] created a visual question generation task.

Attention-based neural networks are also used for question generation tasks such as training a sequence learning model using a seq2seq learning objective [9]. Formerly overlooked approaches such as using a hierarchical neural sentence-level sequence tagging model [10] helped bolster the use of attention-based neural networks for question generation tasks. In [13], high-quality question-answer pairs were generated using a system consisting of an information extractor, a neural question generator, and a neural quality controller. Some models [20] feed the generated questions to a QA system which then uses the QA system's performance as a measure of question quality. A few models consider question answering (QA) and question generation (QG) to be complementary tasks and focus on jointly training the two tasks. A generative machine comprehension model that encapsulates the text and creates a question based on the response using a seq2seq framework was proposed in [21]. Some approaches rely on probabilistic correlation to direct the training of both the QA and QG models simultaneously [22]. Other models focus only on the performance of the QA task and not explicitly on the quality of the generated questions. A Generative Domain-Adaptive Nets training framework [23], trains a generative model to produce questions based on unlabeled text and integrate model-generated questions with human-generated questions for question answering model training.

Textual Question and Answer Generation has been well studied in recent years after the introduction of transformer-based architectures which help with prolonged passages. Also, the recent success of large-scale transformer-based architectures such as BERT [24], RoBERTa [25], T5 [1], and PEGASUS [26] has further helped accelerate the research. Quiz-Style Question Generation for News Stories [11] formulated the problem into two distinct seq2seq tasks: question-answer generation (QAG) using PEGASUS [26] and distractor, or incorrect answer, generation (DG) using T5 [1]. In [12] proposed a Rough Answer and Key Sentence Tagging approach to discover answer-related contents and an Answer-guided graph to collect answer-focused structural information that supplements seq2seq models to construct exam-like questions using an extracted dataset from RACE [31].

## **2.2 Table-to-Text Generation**

Table-to-Text Generation is highly dominated by attention-based neural networks. While [3] presented an LSTM based seq2seq model that augments the seq2seq attentional model with a hybrid “pointer-generator” network, [2] utilized the seq2seq model in two stages – first generate a content plan that outlined the information to be included

along with the structure, followed by generating a document by keeping the content structure into consideration.

BERT-to-BERT [27] is a transformer-based encoder-decoder model, where both the encoder and decoder are initialized with publicly available checkpoints of BERT [24]. In particular, [28] used T5 [1] for a variety of Data-to-Text tasks which includes Table-to-Text generation.

### 3 Research Method

As with any machine learning problem, we need a reliable dataset to come up with a solution. While there are many techniques and datasets available for Table-to-Text Generation - which focuses on generating descriptive text based on the highlighted cells, there aren't many reliable datasets for question generation from tables. So, rather than creating an entirely new dataset, we resorted to modifying an existing Table-to-Text dataset into a Table-to-Question dataset for enhancing the performance of existing AQG systems, allowing it to generate questions from tables along with the text.

The ToTTo dataset serves as a great baseline for Table-to-Text tasks as it covers a significant variety of domains, contains highlighted cells that can be used by the model to attend, and meta-data for each table - providing additional context for the model to better formulate the output. In this section, we explain our entire pipeline in two parts, one focusing on the dataset creation and the other on question generation from tables using the generated dataset as shown in Figure 1.

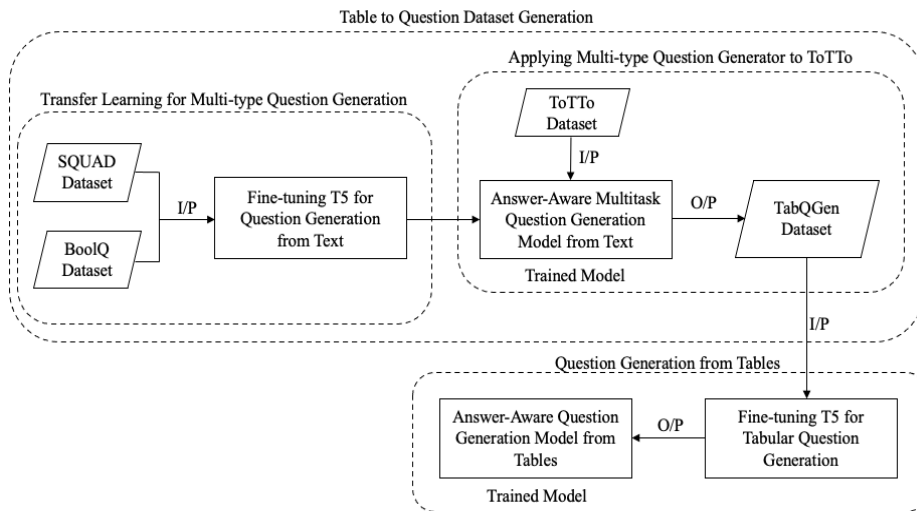


Fig. 1. End-to-End Question Generation Pipeline

### 3.1 Table-to-Question Dataset Generation

ToTTo offers over 120K training examples and puts forward a controlled generation task of constructing a one-sentence description given a Wikipedia table and highlighted cells. Each training example also has around 2-3 different descriptions totaling ~240K-360K descriptions which could potentially be converted into its question form. Given the size of the dataset, it would not be viable to generate a question for each description manually. To tackle this problem, we fine-tuned a model to generate a wide variety of questions given a context and answer.

**Transfer Learning for Multi-Type Question Generation.** We rely on the Stanford Question Answering Dataset [5] (SQuAD1.1) dataset which has over 100K question-answer pairs on 500+ articles for generating semantically accurate questions with a high variance - ranging from a single word to a sentence length question-answer pair. For Boolean style questions, we rely on the BoolQ [36] (Boolean Questions) dataset which consists of 9427 and 3270 labeled training and validation samples respectively. Also, the ability of T5 to perform multiple tasks based on its prefix allows us to use a single model for multiple question-generation tasks. Using this approach, we fine-tune a single T5 model to generate an extensive range of questions with task-specific prefixes as shown below.

*For questions with Boolean Answers:* Input- boolqgen answer: True context: In 2020, among the top 5 leagues in Europe, the Portuguese forward Cristiano Ronaldo scored 33 goals for the Italian club Juventus making him the top goal scorer of the season.

Generated Question- did cristiano ronaldo score more goals than anyone else in europe?

*For questions with one-word answers:* Input- qgen answer: Cristiano Ronaldo context: In 2020, among the top 5 leagues in Europe, the Portuguese forward Cristiano Ronaldo scored 33 goals for the Italian club Juventus making him the top goal scorer of the season.

Generated Question- Who was the top goal scorer in 2020?

*For questions with sentence length answer:* Input- qgen answer: The Portuguese forward Cristiano Ronaldo scored 33 goals. context: In 2020, among the top 5 leagues in Europe, the Portuguese forward Cristiano Ronaldo scored 33 goals for the Italian club Juventus making him the top goal scorer of the season.

Generated Question- How many goals did Cristiano Ronaldo score for Juventus in 2020?

*For questions with summary answers:* Input- qgen answer: Among top 5 leagues in Europe, Cristiano Ronaldo is the top goal scorer with 33 goals in 2020. context: In 2020, among the top 5 leagues in Europe, the Portuguese forward Cristiano Ronaldo scored 33 goals for the Italian club Juventus making him the top goal scorer of the season.

Generated Question- Who is the top goal scorer in the 2020 European football season?

**Applying Multi-Type Question Generator to ToTTo.** Given the ability of our Multi-Type Question Generator model, we apply this on the ToTTo dataset's

descriptions to generate questions. Each *final\_sentence* in the ToTTo dataset is passed as both the context and answer for the above model to generate a question by considering the entire sentence as an answer. We appended the generated question with its respective entry using the question key. This modified ToTTo dataset is referred to as TabQGen in this paper. Figure 2 shows the sample meta-data of the TabQGen dataset, where “...” represents values inherited from ToTTo. Though ToTTo features a test set containing Overlap and Non-Overlap samples, it is not made publicly available, leading us to utilize the ToTTo’s dev set as TabQGen’s test set.

```
{
  "table_page_title": "...",
  "table_webpage_url": "...",
  "table_section_title": "...",
  "table_section_text": "...",
  "table": "...",
  "highlighted_cells": ...,
  "example_id": ...,
  "sentence_annotations": [{
    "original_sentence": "...",
    "sentence_after_deletion": "...",
    "sentence_after_ambiguity": "...",
    "final_sentence": "...",
    "question": <<generated question>>
  }],
}
```

Fig. 2. Sample meta-data of TabQGen

### 3.2 Question Generation from Tables

Now that we have our dataset containing tables, corresponding highlighted cells, and questions, the challenge is to make T5 understand the tabular data. Encoding the entire table would require special embeddings as in TAPAS [29], but since TAPAS is based on the BERT architecture, adding a decoder model to generate text would not only add to the complexity but also deviate from our aim of using a single model architecture. We instead followed the linearization approach by [28] and used special tags to represent the tabular data and meta-data. We also resized the embeddings of the T5 model with the newly added tags for which the weights would be learned during the training. As mentioned in [4], though the full-table approach utilizes the entire table as the source along with additional tokens for highlighted cells, it performs poorly due to large table sizes. Instead, the sub-table approach focuses solely on the highlighted cells along with their respective column and row headers, leading to better performance as the model focuses only on the relevant content.

Figure 3 shows the disparity between the tokenized lengths of full-table and sub-table representations. Only 36% of the training data is  $\leq 600$  tokens for the full-table whereas almost all samples are  $< 600$  tokens for sub-table representation which can be

of utmost benefit for contemporary transformer architectures as most of them were trained with 512 as maximum input length. Furthermore, it was observed that using a sub-table with metadata achieved far better results than using without metadata, which corroborates the findings mentioned by [4]. The metadata provides the necessary context to the model about the table thus preventing it from noisy predictions due to hallucination [30]. This further allows the model to focus on a set of highlighted cells rather than the entire table, allowing more directed and precise questions as shown in Table 1.

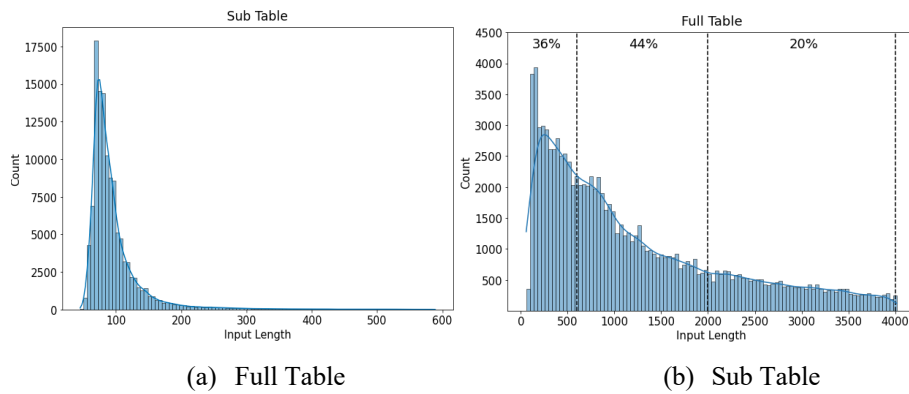


Fig. 3. Sample meta-data of TabQGen

Table 1. Sample of Full Table with highlighted cells

Table Page Title: Top Goal Scorers  
 Table Section Title: Top 5 leagues in 2020

Ranking	Player	Goals	Nationality	Club
1	Cristiano Ronaldo	33	Portugal	Juventus
2	Robert Lewandowski	32	Poland	Bayern Munich
3	Ciro Immobile	28	Italy	SS Lazio
4	Erling Haaland	23	Norway	Borussia Dortmund
5	Mohamed Salah	23	Egypt	Liverpool
6	Romelu Lukaku	22	Belgium	Inter Milan
7	Zlatan Ibrahimovic	20	Sweden	AC Milan
8	Kylian Mbappé	19	France	Paris Saint-Germain
9	Francesco Caputo	19	Italy	US Sassuolo
10	Lionel Messi	19	Argentina	FC Barcelona

Model Input: <page\_title> Top Goal Scorers </page\_title> <section\_title> Top 5 leagues in 2020 </section\_title> <table> <cell> Mohamed Salah <col\_header> Player </col\_header> </cell> <cell> 23 <col\_header> Goals </col\_header> </cell> <cell> Liverpool <col\_header> Club </col\_header> </cell> </table>

Predicted Question: How many goals did Salah score for Liverpool?

## 4 Results

We trained the T5-small, T5-base, and T5-large models using a single NVIDIA Tesla T4 machine with batch sizes of 32, 8, and 4 respectively. We used a linear schedule warmup with an AdamW optimizer of  $1e-4$  as the learning rate. We used 91.72% of the TabQGen dataset for training and the remaining 8.28% (~10K samples) as validation. Table 2 shows the performance of the T5 models on the hold-out test dataset of TabQGen.

The following metrics are used to assess the performance of question generation from tables:

- NIST [32] measures the information gain from each n-gram considered. This allows in giving more credit if a system gets a difficult n-gram match but less credit for an easy n-gram match.
- BLEU [33] scores measure the quality of text that has been translated by a machine from one natural language to another using n-grams. We used a cumulative 4-gram BLEU score (B4) as an evaluation metric.
- ROUGE-L [34] uses statistics based on the Longest Common Subsequence (LCS) to evaluate recall by how many words in reference sentences are used in predicted sentences.
- METEOR [35] is a precision-based metric for evaluating machine-translation output.

**Table 2.** Performance of T5 models on TabQGen

Model	BLUE	NIST	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
<b>T5 Large</b>	72.78	9.41	<b>f:</b> 52.20 <b>p:</b> 53.38 <b>r:</b> 52.84	<b>f:</b> 35.55 <b>p:</b> 36.40 <b>r:</b> 35.91	<b>f:</b> 51.77 <b>p:</b> 52.82 <b>r:</b> 52.33	60.44
<b>T5 Base</b>	72.66	9.37	<b>f:</b> 52.11 <b>p:</b> 53.34 <b>r:</b> 52.72	<b>f:</b> 35.46 <b>p:</b> 36.37 <b>r:</b> 35.76	<b>f:</b> 51.69 <b>p:</b> 52.82 <b>r:</b> 52.17	60.15
<b>T5 Small</b>	71.4	9.21	<b>f:</b> 50.71 <b>p:</b> 51.97 <b>r:</b> 51.30	<b>f:</b> 32.96 <b>p:</b> 34.86 <b>r:</b> 34.28	<b>f:</b> 50.27 <b>p:</b> 51.45 <b>r:</b> 50.73	58.32

The TabQGen dataset along with relevant scripts can be found at <https://github.com/saichandrapandraj/T5QGen> [or] <https://github.com/msakthiganesh/T5QGen>.



## 5 Conclusion

In this paper, we emphasize the need for AQG systems to effectively utilize all the available data in source documents and propose an Answer-Aware Question Generation system using T5 to generate questions from both tabular and textual data. To do so, we augmented each entry of the ToTTo dataset with its respective questions and named this augmented ToTTo as TabQGen. This TabQGen dataset is further used for fine-tuning a T5 model to generate questions from tables. With this approach, utilizing TabQGen in coalescence with existing AQG approaches can effectively generate questions from source documents considering both the textual and tabular data. The findings demonstrate that the model can generate a wide range of high-quality questions from tabular data. Even though we concentrated on automated metrics such as BLEU, NIST, ROUGE, and METEOR, confirming our findings through human inspection is a critical next step.

## 6 References

- [1] Colin Raffel, et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1-67, 2020.
- [2] R. Puduppully, L. Dong, and M. Lapata, “Data-to-Text Generation with Content Selection and Planning”, *AAAI*, vol. 33, no. 01, pp. 6908-6915, Jul. 2019. <https://doi.org/10.1609/aaai.v33i01.33016908>
- [3] Abigail See, Peter J Liu, and Christopher D Manning, “Get to the point: Summarization with pointer generator networks”, In *Proceedings of the 55th Annual Meeting of the ACL*, pp. 1073–1083, Jul. 2017.
- [4] Ankur P. Parikh, et al., “ToTTo: A Controlled Table-To-Text Generation Dataset”, *Proceedings of EMNLP*, 2020.
- [5] Rajpurkar, Pranav and Zhang, Jian and Lopyrev, Konstantin and Liang, Percy, “SQuAA: 100, 000+ Questions for Machine Comprehension of Text”, In *Proceedings of the 2016 Conference on EMNLP*, pp. 2383–2392, 2016.
- [6] Tri Nguyen, et al., “MS MARCO: A Human Generated MACHine Reading COMprehension Dataset”, In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with NIPS 2016*, 2016.
- [7] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer, “TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension”, In *Proceedings of the 55th Annual Meeting of the ACL*, pp. 1601–1611, Jul. 2017. <https://doi.org/10.18653/v1/p17-1147>
- [8] Amrita Saha, Rahul Aralikkatte, Mitesh M. Khapra, and Karthik Sankaranarayanan, “DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension”, In *Proceedings of the 56th Annual Meeting of the ACL*, pp. 1683–1693, Jul. 2018. <https://doi.org/10.18653/v1/p18-1156>
- [9] Xinya Du, Junru Shao, and Claire Cardie, “Learning to Ask: Neural Question Generation for Reading Comprehension”, In *Proceedings of the 55th Annual Meeting of the ACL*, pp. 1342–1352, Jul. 2017. <https://doi.org/10.18653/v1/p17-1123>
- [10] Xinya Du and Claire Cardie, “Identifying Where to Focus in Reading Comprehension for Neural Question Generation”, In *Proceedings of the 2017 Conference on EMNLP*, pp. 2067–2073, Sep. 2017. <https://doi.org/10.18653/v1/d17-1219>

- [11] Adam D. Lelkes, Vinh Q. Tran, and Cong Yu. 2021, “Quiz-Style Question Generation for News Stories”, In Proceedings of the Web Conference 2021, pp. 2501–2511, 2021. <https://doi.org/10.1145/3442381.3449892>
- [12] Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu, “EQG-RACE: Examination-Type Question Generation”, arXiv preprint arXiv:2012.06106, 2020.
- [13] Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He, “Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus”, In Proceedings of the Web Conference 2020, pp. 2032–2043, 2020. <https://doi.org/10.1145/3366423.3380270>
- [14] Jack Mostow, and Wei Chen, “Generating Instruction Automatically for the Reading Strategy of Self-Questioning”, In Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling, pp. 465–472, 2009.
- [15] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne, “Generating Natural Language Questions to Support Learning On-Line”, In Proceedings of the 14th European Workshop on Natural Language Generation, Aug. 2013.
- [16] Karen Mazidi and Rodney D. Nielsen, “Linguistic Considerations in Automatic Question Generation”, In Proceedings of the 52nd Annual Meeting of the ACL, pp. 321–326, 2014. <https://doi.org/10.3115/v1/p14-2053>
- [17] Igor Labutov, Sumit Basu, and Lucy Vanderwende, “Deep Questions without Deep Understanding”, In Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing, pp. 889–898, 2015. <https://doi.org/10.3115/v1/p15-1086>
- [18] Michael Heilman 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 ACL, pp. 609–617, 2010.
- [19] [19] Nasrin Mostafazadeh, 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 ACL, pp. 1802–1813, Aug. 2016. <https://doi.org/10.18653/v1/p16-1170>
- [20] Xingdi Yuan, et al., “Machine Comprehension by Text-to-Text Neural Question Generation”, In Proceedings of the 2nd Workshop on Representation Learning for NLP, pp. 15–25, Aug. 2017. <https://doi.org/10.18653/v1/w17-2603>
- [21] Tong Wang, Xingdi Yuan, and Adam Trischler, “A Joint Model for Question Answering and Question Generation”, In Proceedings of Learning to generate natural language workshop, ICML, Aug. 2017.
- [22] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan and Ming Zhou, “Question answering and Question Generation as Dual Tasks”, arXiv preprint arXiv:1706.02027, 2017.
- [23] Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W. Cohen. “Semi-Supervised QA with Generative Domain-Adaptive Nets”, In Proceedings of the 55th Annual Meeting of the ACL, pp.1040–1050, Jul. 2017. <https://doi.org/10.18653/v1/p17-1096>
- [24] Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, In Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, pp. 4171–4186, Jun. 2019.
- [25] Yinhan Liu, et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach”, arXiv preprint arXiv:1907.11692, 2019.
- [26] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu, “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization”, In Proceedings of the 37th International Conference on Machine Learning, PMLR, vol. 119, pp. 11328–11339, 2020.

- [27] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn, “Leveraging Pre-Trained Checkpoints for Sequence Generation Tasks”, Transactions of the ACL, vol. 8, pp. 264–280, 2020. [https://doi.org/10.1162/tacl\\_a\\_00313](https://doi.org/10.1162/tacl_a_00313)
- [28] Mihir Kale and Abhinav Rastogi, “Text-to-Text Pre-Training for Data-to-Text Tasks”, In Proceedings of the 13th International Conference on Natural Language Generation, pp. 97–102, Dec. 2020.
- [29] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, Julian Eissenschlos, “TaPas: Weakly Supervised Table Parsing via Pre-training”, In Proceedings of the 58th Annual Meeting of ACL, pp. 4320–4333, Jul. 2020. <https://doi.org/10.18653/v1/2020.acl-main.398>
- [30] Sam Wiseman, Stuart M Shieber, and Alexander M Rush, “Challenges in Data-to-Document Generation”, In Proceedings of the 2017 Conference on EMNLP, pp. 2253–2263, Sep. 2017.
- [31] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang and Eduard Hovy, “RACE: Large-scale ReAding Comprehension Dataset from Examinations”, In Proceedings of the 2017 Conference on EMNLP, pp. 785–794, Sep. 2017. <https://doi.org/10.18653/v1/d17-1082>
- [32] Mark Przybocki, Kay Peterson, and Sébastien Bronsart, “NIST Metrics for Machine Translation Challenge (MetricsMATR)”, NIST Multimodal Information Group, Apr. 2008. <https://doi.org/10.1007/s10590-009-9065-6>
- [33] K. Papineni, S. Rouskos, T. Ward, and W. J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation”, In Proceedings of the 40th Annual Meeting of ACL, pp. 311–318, Jul. 2002. <https://doi.org/10.3115/1073083.1073135>
- [34] Lin, Chin-Yew, “ROUGE: A Package for Automatic Evaluation of Summaries”, Association for Computational Linguistics, pp. 74–81, Jul. 2004.
- [35] S. Banerjee A. and Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”, In Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72, Jun. 2005.
- [36] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova, “BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions”, In Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, pp. 2924–2936, Jun. 2019.
- [37] Hosam Farouk El-Sofany, Samir A. El-Seoud, F. F. M. Ghaleb, Shaima Ibrahim, Noor Al-Jaidah, “Questions-Bank System to Enhance E-Learning in School Education”, International Journal of Emerging Technologies in Learning – iJET, vol. 4, pp. 8-19, 2009. <https://doi.org/10.3991/ijet.v4i3.978>
- [38] Sasko Ristov, Marjan Gusev, Goce Armenski, “Massive Development of E-Testing Questions”, International Journal of Emerging Technologies in Learning – iJET, vol. 10, pp. 46-53, 2015. <https://doi.org/10.3991/ijet.v10i4.4688>

## 7 Authors

**Saichandra Pandraju** graduated with a bachelor's degree in Electronics and Communications Engineering from QIS College of Engineering and Technology, Ongole, India. At present, he is working in the Research & Development wing at Infosys Ltd., mainly engaged in the research of Natural Language Processing.

**Sakthi Ganesh Mahalingam** graduated from VIT University, Vellore, India with a bachelor's degree in Electronics and Communications Engineering. At present, he is

working in the Research & Development wing at Infosys Ltd., mainly engaged in deep learning research.

Article submitted 2021-06-28. Resubmitted 2021-08-01. Final acceptance 2021-08-01. Final version published as submitted by the authors.