Controlled Table-To-Text Generation

Nikhil Gupta* ngupta332@gatech.edu
Sreehari Sreejith* ssreejith3@gatech.edu
Shreesha Kulkarni* shrek@gatech.edu

Abstract

In our project, we explore the task of $controlled$ table to text generation: given a Wikipedia table and a set of highlighted cells from the table as the source x, the goal is to produce a single sentence y describing the same. Existing data generation models often generate text that is not faithful to source. The overarching goal behind addressing such a problem is to be able to perform better, more faithful Natural Language Generation. In our work, we experiment with two broad approaches to address this problem - a pragmatic reasoning approach and an encoder-decoder approach leveraging pretrained models fine-tuned on a relevant downstream task (i.e., summarization). Our experiments show that the pragmatic reasoning approach is beneficial when the model is under-confident about its predictions, and, using pre-trained encoder-decoder models that are fine-tuned on the summarization task serve as a good starting point for the downstream TOTTO task. Additionally, motivated by our qualitative results, we conclude with some key insights where we discuss how using innovative encoding schemes for the structured input in combination with the encoder-decoder setup, could serve as interesting starting points to be explored for future work.

1 Introduction

Natural Language Generation (NLG) is the task of generating text in natural language from structured data. Modern text generation approaches have become extremely good at generating fluent and grammatically correct sentences but face some major challenges - (i) Hallucination: The models sometimes generate words which fit the context but are not faithful to the input data. (ii) Rare Topics: Some topics have limited examples which cause models significant trouble to generate text. (iii) Logical Inference: Models who achieved good scores on NLG Benchmark have shown poor results on logical inference tasks suggesting that the model has not understood the data.

To truly help solve some of the above problems, we need to work on specific NLG tasks that aim to specifically address a subset of these problems. One such task is that of $controlled$ table-to-text generation. $Controlled$ table-to-text generation is a recently formulated task proposed by Parikh et al. [2020] along with a newly released Table-to-Text dataset referred to as ToTTo. The main goal is to be able to generate a single sentence which accurately captures the information presented in the highlighted cells in a table. We intend to work on this task to address two problems - *(1) Hallucination (2) Input encoding to capture information against different input schemas.*

In our work, we experiment with 2 broad approaches in order to generate target sentence conditioned on the highlighted section of the table. First, we explore a pragmatic reasoning approach and second, we experiment with a seq2seq based approach where we leverage the use of multiple pretrained models, which are fine-tuned on a related task (i.e., summarization). We provide more details in sections 3, 4 and 5. We show that our approaches produce useful results and serve as strong starting points that could bridge the gap seen in current NLG methods which tend to lack faithfulness to source data. We also explore our qualitative results to highlight both successful and failure cases of conditional generation that motivates interesting directions for future research in this area.

Table Title: Pune - Nagpur Humsafar Express
Section Title: Schedule
Table Description: *None*

Train Number	Station Code	Departure Station	Departure Time	Departure Day	Arrival Station	Arrival Time	Arrival Day
11417	PUNE	Pune Junction	22:00 PM	Thu	Nagpur Junction	13:30 PM	Fri
11418	NGP	Nagpur Junction	15:00 PM	Fri	Pune Junction	08:05 AM	Sat

Target sentence: The 11417 Pune - Nagpur Humsafar Express runs between Pune Junction and Nagpur Junction.

Figure 1: ToTTo example

2 Related Work

Traditional approaches to NLG include a standard multi-step system such as that proposed by Reiter & Dale (Reiter and Dale [2000]) which contains multiple phases like content determination, ordering and structuring, sentence planning, and surface realization. The drawback of the traditional approaches was that they were based on handcrafted rules. More recently, deep-learning based approaches have achieved state-of-the-art performance due to their high representational power.

Transfer learning has shown great potential on downstream NLP tasks like summarization, question answering, text generation etc. Various LSTM based encoder-decoder models are used to solve the text generation task. But the advent of transformer based architecture advanced the state-of-the-art for this task. In one of our approaches, we explore the use of these transformer based architectures to build encoder-decoder models to address controlled table-to-text generation.

3 Problem Definition

Given a table $\mathbf{t}$ and related metadata $\mathbf{m}$ (page title, section title, table section text) and a set of highlighted cells $\mathbf{t_{highlight}}$, we are interested in producing a final descriptive sentence $\mathbf{s_{final}}$, which captures the highlighted cells in the context of the table. Mathematically this can be described as learning a function $\mathbf{f} : \mathbf{x} \rightarrow \mathbf{y}$ where $\mathbf{x} = (\mathbf{t}; \mathbf{m}; \mathbf{t_{highlight}})$ and $\mathbf{y} = \mathbf{s_{final}}$. Figure 1 shows an example of what the inputs and outputs look like for a single training example.

Motivation behind addressing such a problem is to be able to perform better, more faithful Natural Language Generation. Advances in NLG can help companies generate insightful content on a wide variety of topics in considerably lesser time. Understanding structured data can also be quite useful for services like Alexa and Siri and can be used as part of the pipeline in retrieving and translating to text/speech, relevant results from the web from any semi-structured sources such as Wikipedia tables. Some applications include - (i) generating sentences given biographical data (ii) textual descriptions of restaurants given meaning representations (iii) basketball game summaries given box score statistics (iv) generating fun facts from superlative tables in Wikipedia

4 Methods

We use 2 broad sets of approaches to address the controlled table to text generation problem - (i) a pragmatic approach and (ii) a sequence to sequence based approach where we explore leveraging pretrained models in the controlled table-to-text generation setting.

4.1 Pragmatic Approach

In pragmatic reasoning approach (Shen et al. [2019]), we have two independent models as shown in Figure 2. One is called speaker model and other is called listener model.

Speaker model is responsible for generating the output text conditioned on the input along with its probability. Mathematically, speaker model can be represented as $S(o|i)$ which generates the output o along with a probability score p when given input i.

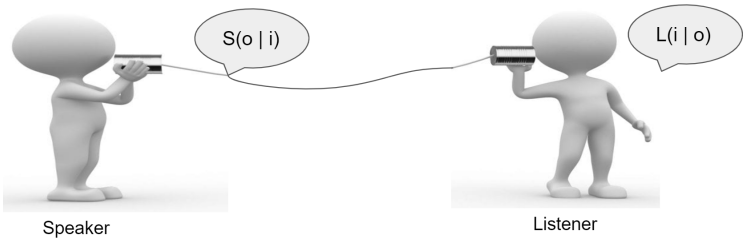


Figure 2: Pragmatic Reasoning Approach

Listener model, on the other hand, is responsible for identifying the inputs from the generated output along with a probability or a score. Mathematically, listener model can be represented as $L(i|o)$ which giving a score to the inputs given a generated output.

Since, both model independently, assign a score to generated output o, we can combine both the model's score to pick a best generated sentence. Mathematically, the best generated output o should be given as follows. λ is a hyper-parameter.

$$\underset{o}{\operatorname{argmax}}\, S(o|i)^{\lambda} \cdot L(i|o)^{1-\lambda} \tag{1}$$

4.1.1 Speaker and Listener Models

Speaker model, for TOTTO task, can be any encoder-decoder model that will take the table and the highlighted cell as the input and will generate an output text along with its probability as shown in Figure 3a. The probability can be beam search score or any equivalent raw score that can be converted to probability. We will use this speaker model to generate top K output texts where K would be a hyper-parameter. Their beam score can be used to compute their probabilities.

Listener model, for this task, can be any sequence classification model that can take a cell string and a generated text as input and can output the probability that the cell is highlighted. We have used BERT as our listener model as shown in Figure 3b.

To assign the listener score to a generated text, we will call the listener model for all possible (c, o) pairs where c is the cells in a given table and o is all top K generated sentences. So, the listener probability can be as follows.

$$L(i|o) = \prod_{c \in \text{H}} P(\text{c=1}; o) \prod_{c \in \text{NH}} P(\text{c=0}; o)$$

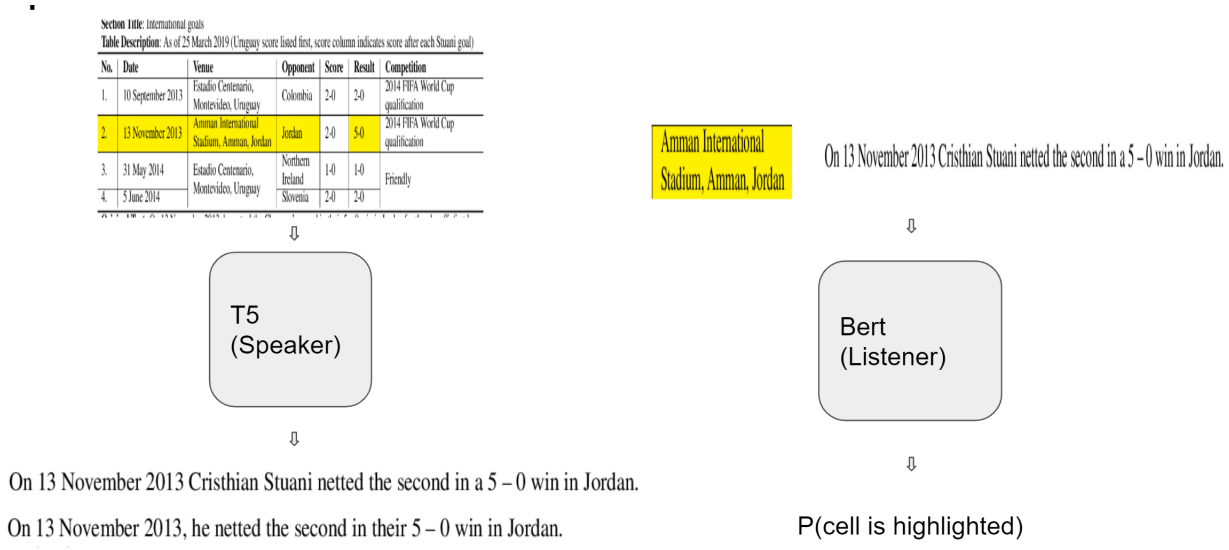
where H and NH represents group of highlighted cells and non highlighted cells respectively whereas $P(c = 1)$ and $P(c = 0)$ represents the probability that cell is highlighted and not highlighted respectively.

We will use the equation 1 to compute the aggregate score for a generated text sequence o and will pick the one with the highest score.

4.2 Leveraging pretrained models for controlled table-to-text generation

Rothe et al. [2019] showed that pre-trained checkpoints can be used to build powerful encoder-decoder models that result in superior performance on on Sequence Generation tasks like Summarization and Machine Translation. *We extend their idea one step further by further using a model that is trained on the summarization task before finetuning on our downstream TOTTO task*. Our hypothesis was that the summarization task being similar to our TOTTO task can benefit the model on TOTTO task and help us see improved results with lesser fine-tuning. Also, we note that one other key motivation behind using such large pretrained models is to be able to generate coherent sentences around the various topics that may appear in the diverse tables that we encounter in our task.

We use combinations of encoders and decoders using Bert, RoBERTa and GPT as suggested by Rothe et al. [2019]. We specifically used the fine-tuned encoder-decoder models on summarization



(a) Speaker Model

(b) Listener Model

Figure 3: Speaker and Listener Models

task from huggingface publicly available models and trained on our downstream task. To further clarify this, the end-to-end procedure looks as follows: these encoder-decoder models are built by first choosing the pretrained publicly available endpoints of the model (say, BERT). Both encoder and decoder models are initialised with these pretrained weights and the configurations are set so that encoder hidden states are sent to the decoder which is now configured to use cross-attention and also has an additional language modelling head for the final sequence generation. This encoder-decoder model then gets trained on the summarization task and we use the pretrained weights following this training as the starting point when we train on our ToTTo dataset.

1. **Bert2Bert**: Bert is a transformer based architecture as presented in Devlin et al. [2018] that advanced the state-of-the-art at the time of its release on many downstream tasks. In Bert2Bert we use Bert as both encoder and decoder where the decoder part contains cross-attention layers and is autoregressive whereas the Bert encoder remains truly bidirectional as in the original Bert architecture.
2. **Roberta2Roberta**: Roberta2Roberta is an encoder-decoder model in which both the encoder and decoder are RoBERTa models. In comparison to the original Bert model, RoBERTa is better trained, using larger batch size with dynamically changing masking pattern on a larger dataset and is thus expected to perform better.
3. **Bert2GPT**: In BERT2GPT2, we use a BERT encoder model and use GPT2 as our autoregressive decoder model. Bert2GPT2 is slightly different from the BERT based decoders in that GPT2 is inherently autoregressive and so we decided that it was important to see if there was any difference in training using the same.
4. **T5**: The T5 model Raffel et al. [2019] is a task-agnostic text-to-text model which is based on the original transformer model Vaswani et al. [2017]. T5 is pretrained on the large-scale C4 dataset (i.e., colossal, cleaned version of Common Crawl's web crawl corpus). T5 is trained to be applied to any text-to-text task by simply appending a prefix indicating the task. In our task, we use T5 specifically for conditional generation.

5 Experiments and Results

5.1 TOTTO Dataset

For our experiments, we use the ToTTo dataset which was released in 2020. The task contains a table with highlighted cells along with some metadata as input and model needs to generate faithful text

Split	Number of examples
Training set	1000
Test set	500

Table 1: Dataset Stats

that describes the highlighted cells in the input.
For instance, as shown in the Figure 1, the table along with Table Title, Section Title and Table Description (metadata) and the highlighted cells are given as input and the model needs to generate the target statement or a statement similar to that, which is specifically conditioned on the highlighted cells in the table. We note here that it is the presence of these highlighted cells and the need for the output to be conditioned on them that differentiates our task from simply summarizing the table or its metadata.

We use this dataset because - (i) It can serve as a benchmark for high precision text generation because it tells the model the specific information to look for in input (via highlighting cells). (ii) The dataset is very diverse in terms of the topics, information and the table structure.

Due to compute limitations, we used a subset of the data and perform all our experiments on that. Table 1 shows our data statistics.

Evaluation Metrics: We use the BLEU metric Papineni et al. [2002] for evaluation. This metric is widely used and uses n-gram overlap between the reference y and generated $\hat{y}$. This does not involve input in the calculation. In our work, we have also shown some of the limitations of this metric.

5.2 Results

5.2.1 Pragmatic Reasoning

We split the training set as mentioned in Table 1 into training and validation set. Table 2 shows the results of pragmatic reasoning.

split	Bert2Bert	Bert2Bert + Pragmatic	% change
Validation set	0.1853	0.1914	**+3**%
Test set	0.2210	0.2273	**+2.8**%

Table 2: Pragmatic Reasoning results on Bert2Bert model

The Table 3 shows a case where pragmatic reasoning helped.

Target	In 2017, Anupam Roy composed for the Hindi films Running Shaadi (one song) and Dear Maya.
Bert2Bert	Anupam Roy was a Lyricist-Music Director at Runningshaadi.com in 2017.
Bert2Bert + Pragmatic	Anupam Roy was a Lyricist-Music Director at Runningshaadi.com in 2017 for Hindi and Dear Maya.

Table 3: Pragmatic reasoning improved the BLEU score for this example by 50%.

5.3 Bert2Bert, Roberta2Roberta, Bert2GPT2 and T5

We show our quantitative results on the 4 different encoder decoder models in table 4. We see that the Roberta2Roberta model performs the best out of the four models with T5 also generating close to a similar BLEU score.

More importantly, we take a close look at our qualitative results to get a good sense for how each model performs and identify various success and failure cases. These results are shown in Figures 4 and 5.

We discuss both the quantitative and qualitative results in depth in the discussion section that follows.

Model	Target	Generated
Bert2GPT2	Coolie is a 1995 Tamil language film directed by, P. Vasu.	'Coolie' is a 1995 Tamil film directed by P. Vasu, and produced by Pr.V Vasu. "Greenie" is an episode of the 1995 Kooli movie, Coolie.P.
Roberta2Roberta	Elisha Oscar Crosby served as the United States Minister Resident to Guatemala from 1861 to 1864.	Elisha Oscar Crosby was the Ambassador of the United States to Guatemala from May 28, 1861.
Bert2GPT2	Mayra Rosales (born 1980) is the heaviest living woman.	raquel Rosales was the heaviest person in the 1980s, with a height of 1.7 metres.Rosa Rosales is one of the richest people in the world. She has an estimated net worth of $1.
Bert2Bert	The Gates was a television series that aired from June 20, 2010 to September 19, 2010.	the gates (tv series) aired on june 20, 2010. in september 2010, the gates starred as the gates of the gates episode and season finale of " the gates : the gates ". it\'s the second season of the gate series

Figure 4: Qualitative results on the Bert2Bert, Roberta2Roberta and Bert2GPT2 models: In the first 2 rows, the generated text resembles the target text closely. In the 3rd and 4th row, there's minor resemblance, but the model seems to hallucinate a lot of information. For eg, in the 3rd row, the generated text fails to capture the detail that the person is a woman and that she was born in 1980, which get passed via the highlighted cells in the input.

Model	Target	Generated
T5	Utada's success has made her one of Japan's top-selling recording artists of all time.	Utada Hikaru is one of the best-selling music artists in Japan.
T5	Maja Neuenschwander had a victory at the 2015 Vienna Marathon.	Maja Neuenschwander won the Vienna Marathon in 2015.
T5	In the California Clásico in June 2016, the attendance record was 50,816.	California Clásico finished with a record 50,816 points.
T5	West Bromwich Albion lost 0–2 to Blackburn Rovers in front of 16,393 spectators.	Blackburn Rovers won the FA Cup in 1884–85.

Figure 5: Qualitative results on the T5 model: In the first 2 rows, we see very promising, well conditioned text generation. The 3rd and 4th row shows that even though the generated text is relevant, it is not well conditioned on the highlighted cells in the table. For eg., model fails to capture the score and attendance of the soccer game in the 4th row.

Encoder-Decoder Model:	Bert2Bert	Roberta2Roberta	Bert2GPT2	T5 (Conditional Generation)
BLEU Scores:	0.2768	**0.3731**	0.0369	**0.3337**

Table 4: Results using fine-tuned models pretrained on the summarization task

6 Discussions

We carefully analyse the results from our experiments and discuss them in detail and draw a few key insights based on the same

6.1 Pragmatic Reasoning

We found that performance of Pragmatic reasoning depends on various factors such as capability of the speaker model, listener model, similarity between the generated output sentences etc.

In general, we found that efficacy of the Pragmatic approach may reduce as the speaker model becomes powerful. It is intuitive as powerful model tend to generate good sentences, thus, minimizing the need for this approach. The Table 5 shows the Pragmatic reasoning performance with 2 models - Bert2Bert and T5. The pragmatic reasoning did not work well with T5.

	val	test
Bert2Bert + Pragmatic	0.1914(**+3%**)	0.2273(**+2.8%**)
T5 + Pragmatic	0.2978(**-2.4%**)	0.3060(**-2.2%**)

Table 5: Pragmatic Reasoning vs Speaker model

On investigating, we found that Pragmatic reasoning worked when T5 model was not confident in its predictions i.e. the output probabilities by speaker model were low. However, T5 was confident in majority of the cases and its generated sentences were quite similar to each other, thus, confusing the listener model. Thus, training the listener model on larger dataset and including some model generated sentences in it might benefit the listener model.

Our experiments have shown that Pragmatic approach can be very helpful in picking the best sentence. *It has been shown more useful with weaker models or where model are not confident about its predictions*. However, if model is confident, it has shown to have less performance because model's generated sentences becomes similar to each other.

6.2 Bert2Bert, Roberta2Roberta, Bert2GPT2 and T5

From our quantitative results in Table 4, we found that the **Roberta2Roberta model performs the best in terms of Bleu score, with T5 being the next best performing model.**

We note a few key observations here. First, we note that the difference observed between the Roberta2Roberta model and the Bert2Bert model is as expected as Roberta is in many ways an improved version of Bert after having gone through better, more extensive pretraining Liu et al. [2019]. But we see that bert2gpt2 model has a significantly lower score than both. Our intuition behind why this happens is that both bert and roberta are trained on wikipedia data which overlaps with our dataset, whereas this is not the case for GPT which is trained on a large books corpus.

Secondly, and more importantly, we note that the quantitative performance of Roberta2Roberta and that of T5 are close to equal, whereas there seems to significant difference in terms of the qualitative results. In our qualitative results, we found that T5 tends to produce crisp, short sentences, that are reasonably well conditioned on the highlighted cells. We show a few examples here, such as those in the first 2 rows in figure 5. An important observation here was that **good qualitative results do not necessarily translate to good Bleu scores**. For eg., in the first row in figure 5, we see that the generated sentence means the exact same thing as the target, but we fail to see a significant overlap in the actual n-grams in the sentence. Thus, such cases will not be contributing positively to our quantitative evaluation using Bleu scores. We thus note that it becomes important to perform human evaluation on such tasks to correctly gauge their true performance, in addition to using automated metrics such as Bleu as they may only carry limited information.

One final observation based on our qualitative results, and perhaps the most important, was that the **failure cases seem to stem from improper conditioning on the highighted cells in the table**. For eg., in the 3rd row in figure 4, the generated text fails to capture the detail that the person is a woman and that she was born in 1980, which get passed via the highlighted cells in the input. Similarly, in figure 5, the model fails to capture the score and attendance of the soccer game in the 4th row, which are again passed via the highlighted cells in the input. This directly points us towards the need for better representations of the tables itself and we use this to motivate directions for future work in this area.

7 Future Work and Conclusion

In conclusion, we have explored 2 main themes to address controlled table to text generation.

We found that the pragmatic approach is promising and can improve results on models that are underconfident about its predictions. We see the pretrained encoder-decoder models, as good starting points for the task of conditional generation that help us explore the task and identify key areas of improvement.

Additionally, we took a careful look at our qualitative results, which lead us to some key insights (shared below), which could drive future work.

One major observation was that a lot of failure cases were due to improper conditioning on the table and the highlighted cells. To overcome this, we propose the use of better table representations beyond encoding them as html strings. We found 2 relevant recent work that include strategies to embed tabular data i.e., TAPAS and TABERT which is work from google and facebook respectively.

Next, we noticed that the encoder-decoder checkpoints that were fine-tuned on summarization, were easier to train and converged faster for our task in comparison to using standalone checkpoints.

We also saw that it was important to look beyond Bleu scores and investigate qualitative results as the metric doesn't capture the full picture for our task.

8 Labour Division

Nikhil focused on Pragmatic Reasoning approach and training Bert2Bert model. Sreehari focused on training T5, Bert2GPT2 and exploring TAPAS encoding scheme. Shreesha focused on training RoBERTa2RoBERTa model and setting up the starter code and baseline. Rest of the work like presentation, report etc. has been equally shared among the 3.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

Kishore Papineni, 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*, pages 311–318, 2002.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*, 2020.

Colin Raffel, 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.

Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge university press, 2000.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks, 2019.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. Pragmatically informative text generation. *arXiv preprint arXiv:1904.01301*, 2019.

Ashish Vaswani, 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*, pages 5998–6008, 2017.