More Informative Relative Position Encoding for Table-to-Text Generation

Stanford CS224N Custom Project

Yuan Wang
Department of Computer Science
Stanford University
yuanw688@stanford.edu

Abstract

Table-to-text generation is an important task because a lot of valuable information is stored as tables. Recently, Kale and Rastogi (2020) showed that the T5 model can achieve SoTA results on table-to-text generation. Wang et al. (2022) introduced the Lattice model which further improved upon the standard T5 model by tailoring the T5 architecture (i.e. encoder self-attention and relative position encoding) to tabular input. The Lattice model encodes very limited information within the relative position encoding: whether the query token and key token are both in the metadata or same table cell, and if so, their relative position. We believe that encoding additional information within the relative position encoding (e.g. whether the query and token are in the same row or column) could further improve the model's understanding of tabular input and table-to-text generation quality. In this project, we showed that our approach ("Additional Positional Info") achieves slightly better results than the baseline Lattice model on the ToTTo dev set (Parikh et al., 2020). Furthermore, we showed that, with Additional Positional Info, the Structural Attention in the Lattice model is no longer necessary.

1 Key Information to include

- Mentor: Rishi Milind Desai
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Table-to-text generation seeks to generate natural language description for content and entailed conclusion in tables (Wang et al., 2022). See Figure 1 for an example task. It is an important task because a lot of valuable information is stored as tables (e.g. financial statements and economic & demographic statistics). Generating natural language descriptions for these tables make such data easier to discover. It also supports downstream tasks such as tabular semantic retrieval, reasoning, fact-checking and table-assisted question answering.

Compared to text-to-text generation tasks such as summarization, table-to-text generation is challenging and interesting because tabular data is non-linear. Furthermore, tables encode important semantic information within its layout. For example, the semantic relationship between tokens within the same cell is different than the semantic relationship between tokens in different cells. Finally, tabular data is semantically invariant to several transformation (e.g. permutation of row / column order). Capturing these properties of tabular input data within the model architecture could significantly improve table-to-text generation quality.

WMRQ-FM

Section Title: Translators
Table Section Text: *None*

Call sign	Frequency (MHz)	City of license	Facility ID	ERP W	Height m (ft)	Class	Transmitter coordinates	FCC info
W221CQ	92.1	Naugatuck, Connecticut	127817	125	17 m (56 ft)	D	41°29′45″N 73°2′35″W / 41.49583°N 73.04306°W	FCC
W246CC	97.1	Bolton, Connecticut	82412	100	189 m (620 ft)	D	41°48′10″N 72°26′30″W / 41.80278°N 72.44167°W	FCC
W258AL	99.5	Clinton, Connecticut	139348	200	159 m (522 ft)	D	41°34′11″N 73°01′5″W / 41.56972°N 73.01806°W	FCC
W272DO	102.3	New Haven, Connecticut	138034	250	112 m (367 ft)	D	41°20′58″N 72°58′22″W / 41.34944°N 72.97278°W	FCC
W283BS	104.5	Bridgeport, Connecticut	15398	250	69 m (226 ft)	D	41°13′10″N 73°12′6″W / 41.21944°N 73.20167°W	FCC

Sentence(s)

WMRQ-FM's translator stations were 97.1 W246CC in Bolton, 99.5 W258AL in Clinton, and 104.5 W283BS in Bridgeport.

Figure 1: Example ToTTo Task

Recently, Kale and Rastogi (2020) showed that linearizing the tabular input data into text (in a row-major order) and then applying the standard T5 text-to-text generation model achieves SoTA result. Wang et al. (2022) was able to further improve upon the standard T5 model by identifying the special properties of tabular data and tailoring the T5 model architecture to reflect these properties.

One potential limitation of Lattice model in (Wang et al., 2022) is that its relative position encoding only encodes whether the query and key token are both in the metadata or both in the same cell, and, if so, their relative position within the linearized sequence. Another potential limitation is that the Lattice model uses the simplified relative position encoding, which map each relative position bucket to a bias that is added to the logit used to compute attention weights. This means the relative position only affects the attention weights as a bias term, and it does not affect the value vectors in the attention output calculation. We believe that additional relative position information (e.g. whether the tokens are in the same row / column) could affect how the key/value token modifies the meaning of the query token. For example, in Figure 1, different cells within the same row pertains to different types of information for the same radio station while different cells within the same column pertains to same type of information pertaining to different radio stations. Therefore, we believe encoding such additional information in the relative position embedding could further improve the model's understanding of tabular input data and its table-to-text generation quality.

We investigated whether we can improve upon the Lattice model by encoding additional information in the relative position embedding ("Additional Positional Info"). We also investigate whether applying the original Relative Position Encoding approach (Shaw et al., 2018) ("Full Relative Position Encoding") could be beneficial. We achieved slightly better BLEU and PARENT score on the ToTTo dev set using Additional Position Info compared to the baseline Lattice model. We achieved poorer results using Additional Positional Info + Full Relative Position Encoding compared to the Lattice model. Furthermore, we showed that, with Additional Positional Info, the Lattice model's Structural Attention is no longer necessary.

3 Related Work

This project is related to recent efforts to apply the pretrained T5 models to the table-to-text generation task and relative position encoding.

3.1 Applying T5 to Table-to-Text Generation

T5 is a pretrained text-to-text generation model that achieves SoTA results on many NLP tasks (Raffel et al., 2020). Kale and Rastogi (2020) showed how to transform the table-to-text generation task into a text-to-text generation task by linearizing the table into a sequence of tokens (in row-major order) and then applying the T5 model. The T5 model achieved SoTA results without any modifications.

Wang et al. (2022) identified the special properties of tabular input data described in Section 2 and modified the standard T5 architecture to reflect these properties:

- **Structural Attention:** In the T5 model's encoder self-attention, a query token can attend to any of the key/value tokens. Wang et al. (2022) argues that this disregards an important structural information (i.e. a query token should only attend to tokens in the table metadata

or in the same row or column). Therefore, they propose Structural Attention, which prunes the other attention connections.

- **Transformation-Invariant Position Encoding:** In the T5 model's encoder self-attention, the relative position of the query and key/value token in the linearized sequence is mapped to one of N relative position bucket. Wang et al. (2022) argues this is not invariant to transformations that should not affect the table content (e.g. permuting rows or columns). Therefore, they propose Transformation-Invariant Position Encoding, which is invariant to such transformations.

The Lattice model further improved upon the standard T5 model and was more robust to content-invariant transformations of the table. Our work explores whether we can push the idea of Transformation-Invariant Position Encoding further by encoding additional positional information in the relative position embedding.

3.2 Relative Position Encoding

Shaw et al. (2018) introduced the Relative Position Representation. Instead of mapping the absolute position of each token in the sequence to an embedding vector as in the original Transformer paper (Vaswani et al., 2017), Relative Position Representation maps the relative position between the query and key/value token to one of the relative position buckets. Each bucket is then mapped to two embedding vectors. One embedding vector is added to the key vector in the attention-weight calculation. The other embedding vector is added to the value vector in the attention-output calculation.

In a subsequent work, Raffel et al. (2020) showed that for most standard NLP benchmark tasks, a simplified version of Relative Position Encoding worked very well. In this simplified version, each relative position bucket is mapped to a single-dimensional bias that is added to the logit in the attention-weight calculation. The Lattice model uses this simplified version.

Tabular data contains a lot of semantically relevant structural information (e.g. whether the query and key/value token are in the same cell). Therefore, its possible that table-to-text generation could benefit from the richer original Relative Position Representation proposed by Shaw et al. (2018). In particular, it might make sense to allow the relative position to interact with the query vector in the attention-weight calculation and also to allow the relative position to affect the value vector in the attention output calculation.

4 Approach

We will use the Lattice model introduced by Wang et al. (2022) as our baseline. On top of this baseline, we explore the following changes: (1) Additional Positional Info, (2) Full Relative Position Encoding, and (3) Ablating Structural Attention.

4.1 Additional Positional Info

In order to capture additional information about the relative position of the query and key/value tokens within the table, we use the following scheme to map the relative position information to one of the N relative position buckets:

1. If the query and key are both in the metadata or both in the same cell, map their relative position (i.e |i-j|) to a bucket in [0, N-6]. We use the same mapping as T5 with a cap at N-6.
2. Otherwise,
   (a) If the query is in the metadata and the key is in the table body, map it to bucket N-5.
   (b) If the query is in the table body and the key is in the metadata, map it to bucket N-4.
   (c) If the query and key are in the same row (but not same column), map it to bucket N-3.
   (d) If the query and key are in the same column (but not same row), map it to bucket N-2.
   (e) If the query and key are not in the same row or column, map it to bucket N-1.

We initialize the bias term associated with each relative position bucket using the pretrained T5 weights. We then update these bias terms during finetuning to adapt them to their new semantics.

4.2 Full Relative Position Encoding

We follow the original Relative Position Encoding approach proposed by Shaw et al. (2018). We map each relative position bucket to 2 embedding vectors (a_{ij}^K and a_{ij}^V) and add them to the key vector in the attention weight calculation (Equations 1, 2) and the value vector in the attention output calculation (Equation 3). Because the embedding vectors associated with each relative position bucket were not present during the pretraining of the T5 model, we will need to randomly initialize these vectors and then update them during finetuning.

$$e_{ij} = (x_i W^Q)(w_j W^K + a_{ij}^K)^T / \sqrt{d_z} \tag{1}$$

$$\alpha_{ij} = softmax_j(e_{ij}) \tag{2}$$

$$z_i = \sum_j \alpha_{ij}(x_j W^V + a_{ij}^V) \tag{3}$$

4.3 Ablating Structural Attention

To ablate the Structural Attention, we revert the Lattice model to use the standard T5 model's encoder self-attention mechanism. This means every query token can attend to every key/value token.

4.4 Notes on Implementation

We use the Lattice model implementation provided by the authors of Wang et al. (2022) as the starting point. We then implement the Additional Positional Info (Subsection 4.1) and Full Relative Position Encoding (Subsection 4.2).

5 Experiments

5.1 Data

We use the ToTTo dataset (Parikh et al., 2020) to train and evaluate our models. The input is a table with metadata (e.g. table title) with some cells highlighted and the target is text describing the highlighted cells. See Figure 1 for an example. We could only submit one model for evaluation on the ToTTo test set. Therefore, we used the ToTTo dev set for our evaluations. The ToTTo dev set has two subsets:

- **Overlap**: These examples use tables in the training set with different cells highlighted.
- **Non-Overlap**: These examples use tables not in the training set. This should be a better test of generalization to previously unseen tables.

5.2 Evaluation method

Following the approach in Parikh et al. (2020), Kale and Rastogi (2020) and Wang et al. (2022), we use the BLEU score and PARENT (F1) score to evaluate our models. We use these metrics to measure how well the model's prediction capture information within the input table and its fluency. According to Dhingra et al. (2019), BLEU score only compares the prediction against the reference sentences which may not accurately reflect the table. Therefore Dhingra et al. (2019) proposed the PARENT (F1) Score which compares the model's prediction against both the input table and the reference sentences. Dhingra et al. (2019) showed that the PARENT score correlated with human evaluations significantly better than BLEU score.

5.3 Experimental details

We fineutine a pretrained T5 model (with the architectural changes described in Section 4) using the same hyperparameters as Wang et al. (2022). See Appendix A for details.

5.4 Results

5.4.1 Main Results

Our main findings (Table 1) are:

1. Encoding additional positional information ("Additional Positional Info") seem to slightly improve the BLEU and PARENT score compared to the baseline Lattice model on the ToTTo dev set. However, our evaluation on the ToTTo test set (Table 2)) showed similar metrics compared to the baseline Lattice model. This suggests that encoding additional positional information may have a slight benefit which is consistent with our expectations.
2. Using full relative position encoding ("Additional Positional Info + Full Relative Position Encoding") seem to significantly decrease the BLEU and PARENT score compared to both Additional Positional Info and baseline Lattice model. This is not very surprising because the embedding vectors for the Full Relative Position Encoding are not present during the pretraining and need to be randomly initialized. This is also consistent with the training curve in Figure 2, which showed the Additional Positional Info + Full Relative Position Encoding started the finetuning with a lower BLEU score on the dev set due to randomly initializing new embedding vectors and never catching up.
3. With Additional Positional Info, we can ablate Structural Attention ("Additional Positional Info w/o Structural Attn") and achieve similar metrics as "Additional Info."

Model	**Overall**		**Overlap**		**Non-Overlap**	
	B	P	B	P	B	P
Lattice (Baseline)	47.5	58.3	55.7	63.3	39.4	53.5
Additional Positional Info (Ours)	**47.8**	58.7	**56.1**	**63.5**	39.7	54.1
Additional Positional Info + Full Relative Position Encoding (Ours)	45.9	57.0	53.7	61.6	38.2	52.6
Additional Positional Info w/o Structural Attn (Ours)	**47.8**	**59.0**	55.9	**63.5**	**40.0**	**54.6**

Table 1: Evaluations on ToTTo Dev Set (B: BLEU, P: PARENT)

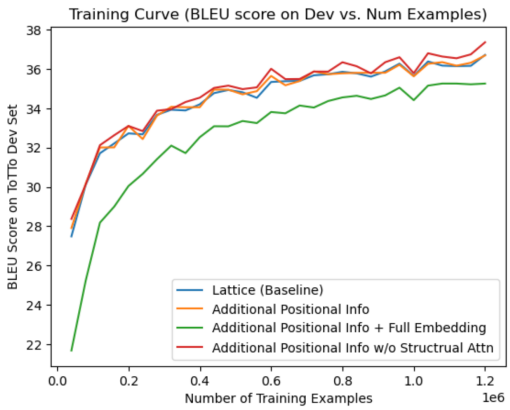


Figure 2: Training Curve

5.4.2 Ablation Study

We investigate the relative contributions of Structural Attention and Additional Positional Info through ablation (Table 3). We see that, by adding Additional Positional Info to the T5 model, it's no longer

Model	Overall			Overlap			Non-Overlap		
	B	P	BR	B	P	BR	B	P	BR
Lattice Baseline	47.4	57.8	0.207	55.6	62.3	0.337	39.1	53.3	0.077
Additional Positional Info (Ours)	47.3	57.9	0.210	55.5	62.6	0.341	39.1	53.3	0.079

Table 2: Evaluation on ToTTo Test Set (B: BLEU, P: PARENT, BR: BLEURT)

Model	Att	Pos	Overall	Overlap	Non-Overlap
T5	-	-	45.7	53.7	37.7
T5 + Additional Positional Info	-	✓	47.8	55.9	40.0
T5 + Structural Attention	✓	-	47.0	54.4	39.6
Lattice + Additional Positional Info	✓	✓	47.8	56.1	39.7

Table 3: Ablation Study on ToTTo Dev Set using BLEU Score

necessary to use Structural Attention. This not surprising because the Additional Positional Info captures whether the query and key tokens are not in the same row or column. So the model can learn a large negative bias for such situation which is equivalent to the Structural Attention.

5.4.3 Robustness to Content-Invariant Transformations

Similar to Wang et al. (2022), we also investigated whether the models are robust to content invariant transformations of the tabular input data (e.g. permuting the rows or columns). We found that, similar to the baseline Lattice model, our model was also robust to these content-invariant transformations (see Table 4).

Model	Overall			Overlap			Non-Overlap		
	O	T	Δ	O	T	Δ	O	T	Δ
Lattice (Baseline)	47.5	47.5	0	55.5	55.5	0	39.5	39.5	0
Additional Positional Info	47.8	47.8	0	56.1	56.1	0	39.7	39.7	0
Additional Positional Info w/o Structural Attn	47.8	47.8	0	55.9	55.9	0	40.0	40.0	0

Table 4: Robustness evaluation on ToTTo dev set. We use the BLEU score. O is the original dev set. T is the dev set after content-invariant transformation (e.g. randomly permuting rows and columns). Δ is the difference between O and T.

6 Analysis

6.1 Analysis of Per-Instance Differences

We first consider the histogram of per-instance differences in BLEU score and PARENT score for the "Additional Positional Info w/o Structural Attn" versus the baseline Lattice model (Figure 3). We see that most instances had zero or minimal difference in scores. This suggests that two model's outputs are similar. Furthermore, we see that the differences are greater on the non-overlap subset compared to the overlap subset. This suggests that both models memorized certain patterns from the tables in the training set which resulted in more similar output on the Overlap subset.

We then randomly sampled 20 examples from the ToTTo dev set (10 from overlap and 10 from non-overlap) and manually rated them from 0-3 (0: mostly incorrect; 1: somewhat correct; 2: mostly correct). See Table 5. We observed that:

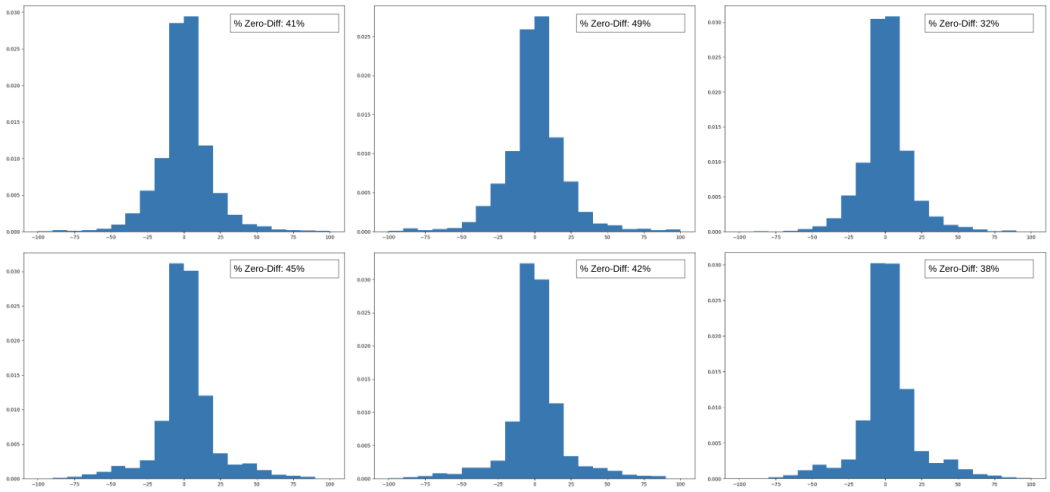


Figure 3: Histogram of Non-Zero Per Instance Score BLEU Score Differences on ToTTo Dev Set. Top: BLEU; Bottom: PARENT (F1). Left: Overall; Middle: Overlap; Right: Non-overlap.

- Both models performed well on both the Overlap and Non-Overlap set. Majority of our human ratings are 2 (mostly correct) and the rest are 1 (somewhat correct). There were no 0 (mostly incorrect) predictions.
- Despite the large difference in BLEU and PARENT score, the quality of the output of both models looked similar for most instances. Compared to the Lattice baseline, the "Additional Info w/o Structural Attn" had one significant win in the Non-Overlap samples and 1 marginal win and 2 marginal losses in the Overlap-subset.
- The quality of the output on the Overlap and Non-Overlap set appear similar despite the large difference in BLEU and PARENT score. This suggests that both models generalize reasonably well to unseen tables. The large difference in quantitative evaluation metrics appears to be mostly due to challenges in automatic evaluation.

Model	**Overlap**			**Non-Overlap**		
	H	B	P	H	B	P
Lattice (Baseline)	1.7	59.2	69.1	1.6	30.2	43.7
Additional Positional Info w/o Structural Attn	1.7	65.8	73.0	1.7	38.5	53.6

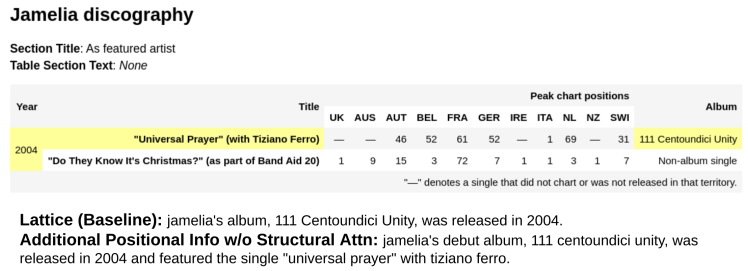
Table 5: Human evaluation of a sample of model outputs (H: human rating, B: BLEU, P: PARENT). We sampled 10 examples in the Overlap set and 10 examples in the Non-overlap set. The human ratings are 0-3 (0: mostly incorrect, 1: somewhat correct, 2: mostly correct). Note all of the actual human rating values are 1 or 2.

6.2 Analysis of Wins and Losses Patterns

We observed a few win and loss patterns for the Additional Positional Info w/o Structural Attention model versus the baseline Lattice model.

1. **[Win Pattern]** Given an input table with multiple highlighted columns, the Additional Positional Info w/o Structural Attention model seem to capture information from more columns than the baseline Lattice model. See Figure 4 for an example.
2. **[Loss Pattern]** Given an input table with multiple highlighted rows, the Additional Positional Info w/o Structural Attention model seems to miss the information on some rows that is captured by the baseline Lattice model. See Figure 5 for an example.

Both the wins and losses may be caused by the Additional Positional Info w/o Structural Attention model's relative position encoding mapping query and key tokens being in the same row versus same column to different buckets (N-3 and N-2 respectively) while the baseline Lattice model map both situations to the same bucket (N-1).



Jamelia discography

Section Title: As featured artist
Table Section Text: *None*

Year	Title	Peak chart positions											Album
		UK	AUS	AUT	BEL	FRA	GER	IRE	ITA	NL	NZ	SWI	
2004	"Universal Prayer" (with Tiziano Ferro)	—	—	46	52	61	52	—	1	69	—	31	111 Centoundici Unity
	"Do They Know It's Christmas?" (as part of Band Aid 20)	1	9	15	3	72	7	1	1	3	1	7	Non-album single

"—" denotes a single that did not chart or was not released in that territory.

Lattice (Baseline): jamelia's album, 111 Centoundici Unity, was released in 2004.
Additional Positional Info w/o Structural Attn: jamelia's debut album, 111 centoundici unity, was released in 2004 and featured the single "universal prayer" with tiziano ferro.

Figure 4: Win Example: Multiple Highlighted Columns

List of World Rally Championship Constructors' champions

Section Title: By season
Table Section Text: *None*

Year	Manufacturers' Championship	Car(s) used	Wins	Podiums	Points	Margin
2018	Japan Toyota	Toyota Yaris WRC	5	14	368	27
2017	United Kingdom M-Sport (Ford)	Ford Fiesta WRC	5	19	428	83
2016	Germany Volkswagen	Volkswagen Polo R WRC	7	14	377	62
2015	Germany Volkswagen	Volkswagen Polo R WRC	11	17	413	183
2014	Germany Volkswagen	Volkswagen Polo R WRC	12	18	447	237
2013	Germany Volkswagen	Volkswagen Polo R WRC	10	18	425	145

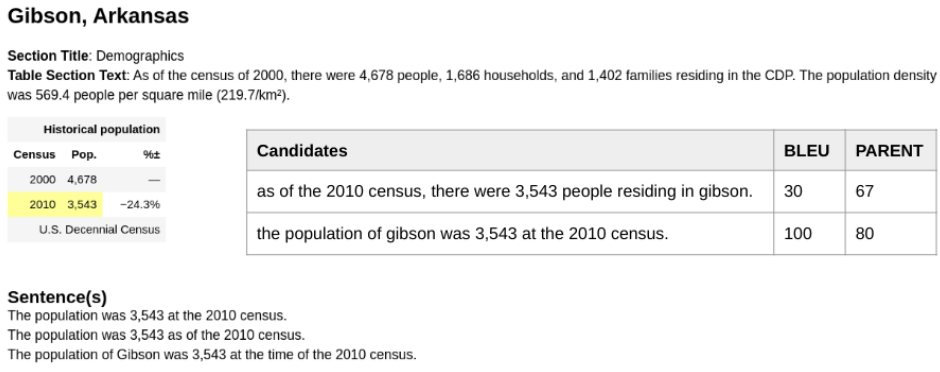
Lattice (Baseline): Volkswagen won the world rally championship constructors' championship in 2013, 2014, 2015 and 2016.
Additional Positional Info w/o Structural Attn: Volkswagen won the world rally championship constructors' championship three times in 2013 and 2015.

Figure 5: Loss Example: Multiple Highlighted Rows

6.3 Analysis of Automatic Evaluation Issues

We observed a few issues with the automatic evaluation metrics:

1. **Reference Phrasing:** The reference outputs often do not cover the different ways to express the information in the table. As a result the BLEU score may penalize a candidate that is valid but phrased differently than the reference sentences. The PARENT score is generally better but still affected by reference sentence phrasing. See Figure 6 for example.
2. **Outside Info:** The reference sentences sometimes use information outside of the table metadata and highlighted cells. This can significantly penalize valid candidates in both BLEU score and PARENT score. See Figure 7 for example.



Gibson, Arkansas

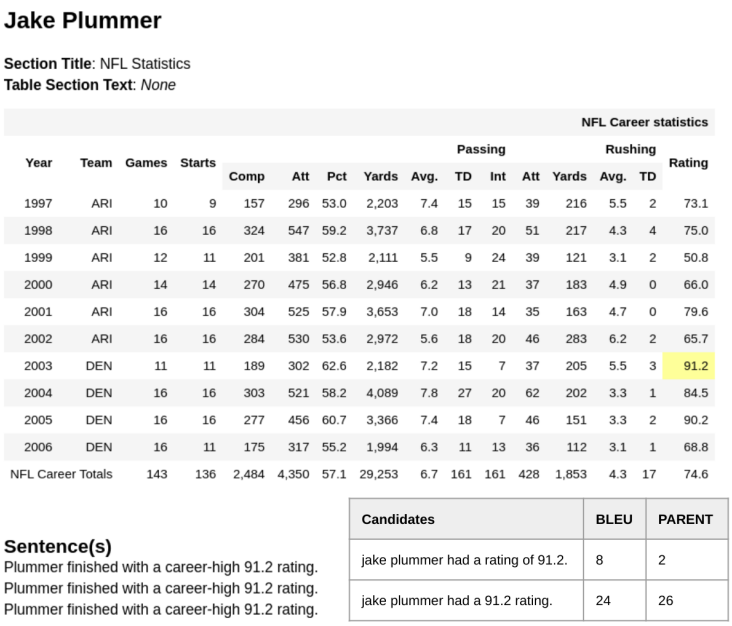
Section Title: Demographics
Table Section Text: As of the census of 2000, there were 4,678 people, 1,686 households, and 1,402 families residing in the CDP. The population density was 569.4 people per square mile (219.7/km²).

Historical population		
Census	Pop.	%±
2000	4,678	—
2010	3,543	−24.3%
U.S. Decennial Census		

Candidates	BLEU	PARENT
as of the 2010 census, there were 3,543 people residing in gibson.	30	67
the population of gibson was 3,543 at the 2010 census.	100	80

Sentence(s)
The population was 3,543 at the 2010 census.
The population was 3,543 as of the 2010 census.
The population of Gibson was 3,543 at the time of the 2010 census.

Figure 6: Automatic Evaluation Issues Example for Reference Phrasing



Jake Plummer

Section Title: NFL Statistics
Table Section Text: *None*

NFL Career statistics																
				Passing							Rushing					
Year	Team	Games	Starts	Comp	Att	Pct	Yards	Avg.	TD	Int	Att	Yards	Avg.	TD	Rating
1997	ARI	10	9	157	296	53.0	2,203	7.4	15	15	39	216	5.5	2	73.1
1998	ARI	16	16	324	547	59.2	3,737	6.8	17	20	51	217	4.3	4	75.0
1999	ARI	12	11	201	381	52.8	2,111	5.5	9	24	39	121	3.1	2	50.8
2000	ARI	14	14	270	475	56.8	2,946	6.2	13	21	37	183	4.9	0	66.0
2001	ARI	16	16	304	525	57.9	3,653	7.0	18	14	35	163	4.7	0	79.6
2002	ARI	16	16	284	530	53.6	2,972	5.6	18	20	46	283	6.2	2	65.7
2003	DEN	11	11	189	302	62.6	2,182	7.2	15	7	37	205	5.5	3	91.2
2004	DEN	16	16	303	521	58.2	4,089	7.8	27	20	62	202	3.3	1	84.5
2005	DEN	16	16	277	456	60.7	3,366	7.4	18	7	46	151	3.3	2	90.2
2006	DEN	16	11	175	317	55.2	1,994	6.3	11	13	36	112	3.1	1	68.8
NFL Career Totals		143	136	2,484	4,350	57.1	29,253	6.7	161	161	428	1,853	4.3	17	74.6

Sentence(s)
Plummer finished with a career-high 91.2 rating.
Plummer finished with a career-high 91.2 rating.
Plummer finished with a career-high 91.2 rating.

Candidates	BLEU	PARENT
jake plummer had a rating of 91.2.	8	2
jake plummer had a 91.2 rating.	24	26

Figure 7: Automatic Evaluation Issues Example for Outside Info

7 Conclusion

Our main findings are:

1. Encoding additional positional information results in similar or slightly better table-to-text generation quality than the baseline Lattice model.
2. Using the original Full Relative Position Encoding results in significantly worse table-to-text generation quality than the baseline Lattice model. This may be due to adding new embedding vectors to the model that is not pre-trained.
3. With additional position information, Structural Attention is no longer needed to achieve the Lattice model's SoTA results.

The main limitations and areas for future work are:

1. Explore different initialization strategies that might improve Full Relative Position Encoding.
2. Qualitatively analyze a larger sample to identify common win and loss patterns. Develop and implement strategies for addressing the loss patterns.
3. Evaluate the models on additional benchmarks.
4. Hyperparamter tuning.

References

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

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

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

A Appendix - Model Training Hyperparameters

We use the same model training hyperparameters as Wang et al. (2022) for all of the models studied in this project:

- Model architecture and pre-trained weights: T5-small
- Learning rate: 2e-4
- Max number of steps: 150,000 (8 examples per batch)
- Max input sequence length: 512
- Max target sequence length: 128