ToTTo: A Controlled Table-To-Text Generation Dataset

Ankur P. Parikh♠ Xuezhi Wang♠ Sebastian Gehrmann♠
Manaal Faruqui♠ Bhuwan Dhingra♣[*] Diyi Yang♠◇ Dipanjan Das♠
♠ Google Research, New York, NY
◇ Georgia Tech, Atlanta, GA
♣ Carnegie Mellon University, Pittsburgh, PA

totto@google.com

Abstract

We present ToTTo, an open-domain English table-to-text dataset with over 120,000 training examples that proposes a controlled generation task: given a Wikipedia table and a set of highlighted table cells, produce a one-sentence description. To obtain generated targets that are natural but also faithful to the source table, we introduce a dataset construction process where annotators directly revise existing candidate sentences from Wikipedia. We present systematic analyses of our dataset and annotation process as well as results achieved by several state-of-the-art baselines. While usually fluent, existing methods often hallucinate phrases that are not supported by the table, suggesting that this dataset can serve as a useful research benchmark for high-precision conditional text generation.[1]

1 Introduction

Data-to-text generation (Kukich, 1983; McKeown, 1992) is the task of generating a target textual description $\boldsymbol{y}$ conditioned on source content $\boldsymbol{x}$ in the form of structured data such as a table. Examples include generating sentences given biographical data (Lebret et al., 2016), textual descriptions of restaurants given meaning representations (Novikova et al., 2017), and basketball game summaries given boxscore statistics (Wiseman et al., 2017).

Existing data-to-text tasks have provided an important test-bed for neural generation models (Sutskever et al., 2014; Bahdanau et al., 2014). Neural models are known to be prone to *hallucination*, i.e., generating text that is fluent but not faithful to the source (Vinyals and Le, 2015; Koehn and Knowles, 2017; Lee et al., 2018; Tian et al., 2019) and it is often easier to assess faithfulness of the generated text when the source content is structured (Wiseman et al., 2017; Dhingra et al., 2019). Moreover, structured data can also test a model's ability for reasoning and numerical inference (Wiseman et al., 2017) and for building representations of structured objects (Liu et al., 2018), providing an interesting complement to tasks that test these aspects in the NLU setting (Pasupat and Liang, 2015; Chen et al., 2019; Dua et al., 2019).

However, constructing a data-to-text dataset can be challenging on two axes: task design and annotation process. First, tasks with open-ended output like summarization (Mani, 1999; Lebret et al., 2016; Wiseman et al., 2017) lack explicit signals for models on what to generate, which can lead to subjective content and evaluation challenges (Kryściński et al., 2019). On the other hand, data-to-text tasks that are limited to verbalizing a fully specified meaning representation (Gardent et al., 2017b) do not test a model's ability to perform inference and thus remove a considerable amount of challenge from the task.

Secondly, designing an annotation process to obtain natural but also clean targets is a significant challenge. One strategy employed by many datasets is to have annotators write targets from scratch (Banik et al., 2013; Wen et al., 2015; Gardent et al., 2017a) which can often lack variety in terms of structure and style (Gururangan et al., 2018; Poliak et al., 2018). An alternative is to pair naturally occurring text with tables (Lebret et al., 2016; Wiseman et al., 2017). While more diverse, naturally occurring targets are often noisy and contain information that cannot be inferred from the source. This can make it problematic to disentangle modeling weaknesses from data noise.

In this work, we propose ToTTo, an *open-domain* table-to-text generation dataset that introduces a novel task design and annotation process

[*] Work done during an internship at Google.

[1] ToTTo is available at https://github.com/google-research-datasets/totto

Table Title: Cristhian Stuani
Section Title: International goals
Table Description: As of 25 March 2019 (Uruguay score listed first, score column indicates score after each Stuani goal)

No.	Date	Venue	Opponent	Score	Result	Competition
1.	10 September 2013	Estadio Centenario, Montevideo, Uruguay	Colombia	2-0	2-0	2014 FIFA World Cup qualification
2.	13 November 2013	Amman International Stadium, Amman, Jordan	Jordan	2-0	5-0	2014 FIFA World Cup qualification
3.	31 May 2014	Estadio Centenario, Montevideo, Uruguay	Northern Ireland	1-0	1-0	Friendly
4.	5 June 2014		Slovenia	2-0	2-0	

Original Text: On 13 November 2013, he netted the Charruas' second in their 5 – 0 win in Jordan for the playoffs first leg, finishing Nicolas Lodeiro's cross at close range.
Text after Deletion: On 13 November 2013, he netted the second in their 5 – 0 win in Jordan.
Text after Decontextualization: On 13 November 2013, Cristhian Stuani netted the second in 5 – 0 win in Jordan.
Final Text: On 13 November 2013 Cristhian Stuani netted the second in a 5 – 0 win in Jordan.

Table 1: Example in the ToTTo dataset. The goal of the task is given the table and set of highlighted cells, to produce the final text. Our data annotation process revolves around annotators iteratively revising the original text to produce the final text.

to address the above challenges. First, ToTTo proposes a *controlled* generation task: given a Wikipedia table and a set of highlighted cells as the source $\boldsymbol{x}$, the goal is to produce a single sentence description $\boldsymbol{y}$. The highlighted cells identify portions of potentially large tables that the target sentence should describe, without specifying an explicit meaning representation to verbalize.

For dataset construction, to ensure that targets are natural but also faithful to the source table, we request annotators to **revise** existing Wikipedia candidate sentences into target sentences, instead of asking them to write new target sentences (Wen et al., 2015; Gardent et al., 2017a). Table 1 presents a simple example from ToTTo to illustrate our annotation process. The table and *Original Text* were obtained from Wikipedia using heuristics that collect pairs of tables $\boldsymbol{x}$ and sentences $\boldsymbol{y}$ that likely have significant semantic overlap. This method ensures that the target sentences are *natural*, although they may only be partially related to the table. Next, we create a clean and controlled generation task by requesting annotators to highlight a subset of the table that supports the original sentence and revise the latter iteratively to produce a final sentence (see §5). For instance, in Table 1, the annotator has chosen to highlight a set of table cells (in yellow) that are mentioned in the original text. They then deleted phrases from the original text that are not supported by the table, e.g., *for the playoffs first leg* and replaced the pronoun *he* with an entity *Cristhian Stuani*. The resulting final sentence (*Final Text*) serves as a more suitable generation target than the original sentence. This annotation process makes our dataset well suited for *high-precision* conditional text generation.

Due to the varied nature of Wikipedia tables, ToTTo covers a significant variety of domains while containing targets that are completely faithful to the source (see Figures 2- 6 for more complex examples). Our experiments demonstrate that state-of-the-art neural models struggle to generate faithful results, despite the high quality of the training data. These results suggest that our dataset and the underlying task could serve as a strong benchmark for controllable data-to-text generation models.

2 Related Work

ToTTo differs from existing datasets in both task design and annotation process as we describe below. A summary is given in Table 2.

Task Design Most existing table-to-text datasets are restricted in topic and schema such as WeathErGov (Liang et al., 2009), RoboCup (Chen and Mooney, 2008), Rotowire (Wiseman et al., 2017, basketball), E2E (Novikova et al., 2016, 2017, restaurants), KBGen (Banik et al., 2013, biology), and Wikibio (Lebret et al., 2016, biographies). In contrast, ToTTo contains tables with various schema spanning various topical categories all over Wikipedia. Moreover, ToTTo takes a different view of content selection compared to existing datasets. Prior to the advent of neural approaches, generation systems typically separated content selection (*what to say*) from surface realization (*how to say it*) (Reiter and Dale, 1997). Thus many generation datasets only focused on

Dataset	Train Size	Domain	Target Quality	Target Source	Content Selection
Wikibio (Lebret et al., 2016)	583K	Biographies	Noisy	Wikipedia	Not specified
Rotowire (Wiseman et al., 2017)	4.9K	Basketball	Noisy	Rotowire	Not specified
WebNLG (Gardent et al., 2017b)	25.3K	15 DBPedia categories	Clean	Annotator Generated	Fully specified
E2E (Novikova et al., 2017)	50.6K	Restaurants	Clean	Annotator Generated	Partially specified
LogicNLG (Chen et al., 2020)	28.5K	Wikipedia (open-domain)	Clean	Annotator Generated	Columns via entity linking
ToTTo	**120K**	**Wikipedia (open-domain)**	**Clean**	**Wikipedia (Annotator Revised)**	**Annotator highlighted**

Table 2: Comparison of popular data-to-text datasets. ToTTo combines the advantages of annotator-generated and fully natural text through a revision process.

the latter stage (Wen et al., 2015; Gardent et al., 2017b). However, this decreases the task complexity, since neural systems have already been quite powerful at producing fluent text. Some recent datasets (Wiseman et al., 2017; Lebret et al., 2016) have proposed incorporating content selection into the task by framing it as a summarization problem. However, summarization is much more subjective, which can make the task underconstrained and difficult to evaluate (Kryściński et al., 2019). We place ToTTo as a middle-ground where the highlighted cells provide some guidance on the topic of the target but still leave a considerable amount of content planning to be done by the model.

Annotation Process There are various existing strategies to create the reference target $\boldsymbol{y}$. One strategy employed by many datasets is to have annotators write targets from scratch given a representation of the source (Banik et al., 2013; Wen et al., 2015; Gardent et al., 2017a). While this will result in a target that is faithful to the source data, it often lacks variety in terms of structure and style (Gururangan et al., 2018; Poliak et al., 2018). Domain-specific strategies such as presenting an annotator an image instead of the raw data (Novikova et al., 2016) are not practical for some of the complex tables that we consider. Other datasets have taken the opposite approach: finding real sentences on the web that are heuristically selected in a way that they discuss the source content (Lebret et al., 2016; Wiseman et al., 2017). This strategy typically leads to targets that are natural and diverse, but they may be noisy and contain information that cannot be inferred from the source (Dhingra et al., 2019).To construct ToTTo, we ask annotators to revise existing candidate sentences from Wikipedia so that they only contain information that is supported by the table. This enables ToTTo to maintain the varied language and structure found in natural sentences while producing cleaner targets. The technique of editing exemplar sentences has been used in semiparametric generation models (Guu et al., 2018; Pandey et al., 2018; Peng et al., 2019) and crowd-sourcing small, iterative changes to text has been shown to lead to higher-quality data and a more robust annotation process (Little et al., 2010). However, to our knowledge, we are the first to use this technique to construct generation datasets.

Concurrent to this work, Chen et al. (2020) proposed LogicNLG which also uses Wikipedia tables, although omitting some of the more complex structured ones included in our dataset. Their target sentences are annotator-generated and their task is significantly more uncontrolled due to the lack of annotator highlighted cells.

3 Preliminaries

Our tables come from English Wikipedia articles and thus may not be regular grids.[2] For simplicity, we define a table $\boldsymbol{t}$ as a set of cells $\boldsymbol{t} = \{\boldsymbol{c}_j\}_{j=1}^{\tau}$ where τ is the number of cells in the table. Each cell contains: **(1)** a string value, **(2)** whether or not it is a row or column header, **(3)** the row and column position of this cell in the table, **(4)** The number of rows and columns this cell spans.

Let $\boldsymbol{m} = (m_{\text{page-title}}, m_{\text{section-title}}, m_{\text{section-text}})$ indicate table metadata, i.e, the page title, section title, and up to the first 2 sentences of the section text (if present) respectively. These fields can help provide context to the table's contents. Let $\boldsymbol{s} = (s_1, ..., s_\eta)$ be a sentence of length η. We define an annotation example[3] $\boldsymbol{d} = (\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{s})$ a tuple of table, table metadata, and sentence. Here, $\boldsymbol{D} = \{\boldsymbol{d}_n\}_{n=1}^{N}$ refers to a dataset of annotation examples of size N.

4 Dataset Collection

We first describe how to obtain annotation examples $\boldsymbol{d}$ for subsequent annotation. To prevent any overlap with the Wikibio dataset (Lebret et al.,

[2] In Wikipedia, some cells may span multiple rows and columns. See Table 1 for an example.

[3] An annotation example is different than a task example since the annotator could perform a different task than the model.

2016), we do not use infobox tables. We employed three heuristics to collect tables and sentences:

Number matching We search for tables and sentences on the same Wikipedia page that overlap with a non-date number of at least 3 non-zero digits. The numbers are extracted by regular expressions that capture most common number patterns, including numbers with commas and decimal points. This approach captures most of the table-sentence pairs that describe statistics (e.g., sports, election, census, science, weather).

Cell matching We extract a sentence if it has tokens matching at least 3 distinct cell contents from the **same row** in the table. The intuition is that most tables are structured, and a row is usually used to describe a complete event (e.g., a sports game, an election, census data from a certain year), which is likely to have a corresponding sentence description from the same page.

Hyperlinks The above heuristics only consider sentences and tables on the same page. We also find examples where a sentence s contains a hyperlink to a page with a title that starts with *List* (these pages typically only consist of a large table). If the table $\boldsymbol{t}$ on that page also has a hyperlink to the page containing s, then we consider this to be an annotation example. Such examples typically result in more diverse examples than the other two heuristics, but also add more noise, since the sentence may only be distantly related to the table.

Using the above heuristics we obtain a large set of annotation examples $\boldsymbol{D}$. We then sample random subset of the examples for annotation: 191,693 examples for training, 11,406 examples for development, and 11,406 examples for test. Among these examples, 35.8% were derived from number matching, 29.4% from cell matching, and 34.7% from hyperlinks.

5 Data Annotation Process

The collected annotation examples are noisy since a sentence s may be partially or completely unsupported by the table $\boldsymbol{t}$. We thus define a data annotation process that guides annotators through small, incremental changes to the original sentence. This allows us to measure annotator agreement at every step of the process, which is atypical in existing generation datasets.

5.1 Primary Annotation Task

The primary annotation task consists of the following steps: **(1)** Table Readability, **(2)** Cell highlighting, **(3)** Phrase Deletion, **(4)** Decontextualization. Each of these are described below and more examples are provided in Table 3.

Table Readability If a table is not readable, then the following steps will not need to be completed. This step is only intended to remove fringe cases where the table is poorly formatted or otherwise not understandable (e.g., in a different language). 99.5% of tables are determined to be readable.

Cell Highlighting An annotator is instructed to highlight cells that support the sentence. A phrase is supported by the table if it is either directly stated in the cell contents or meta-data, or can be logically inferred by them. Row and column headers do not need to be highlighted. If the table does not support any part of the sentence, then no cell is marked and no other step needs to be completed. 69.7% of examples are supported by the table.

For instance, in Figure 1, the annotator highlighted cells that support the phrases *second*, *13 November 2013*, *in Jordan*, and *5-0 win*. We denote the set of highlighted cells as a subset of the table: $\boldsymbol{t}_{\text{highlight}} \in \boldsymbol{t}$.

Phrase Deletion This step removes phrases in the sentence unsupported by the selected table cells. Annotators are restricted such that they are only able to delete phrases, transforming the original sentence: $s \rightarrow s_{\text{deletion}}$. In Table 1, the annotator transforms s by removing an individual word *Charras'*, and an entire phrase: *For the playoffs first leg, finishing nicols lodeiro's cross at close range*.

On average, s_{deletion} is different from s for 85.3% of examples and while s has an average length of 26.6 tokens, this is reduced to 15.9 for s_{deletion}. We found that the phrases annotators often disagreed on corresponded to verbs purportedly supported by the table. For instance, in Table 1, some annotators decided *netted* is supported by the table since it is about soccer, while others opted to delete it.

Decontextualization A given sentence s may contain pronominal references or other phrases that depend on context. We thus instruct annotators to identify the main topic of the sentence; if it is a pronoun or other ambiguous phrase, we ask them to replace it with a named entity from the table or metadata. To discourage excessive modification,

Original	After Deletion	After Decontextualization	Final
He was the first president of the Federal Supreme Court (1848–1850) and president of the National Council in 1850–1851.	He was the first president of the Federal Supreme Court (1848–1850) ~~and president of the National Council in 1850–1851~~.	Johann Konrad Kern was the first president of the Federal Supreme Court from 1848 to 1850.	Johann Konrad Kern was the first president of the Federal Supreme Court from 1848 to 1850.
He later raced a Nissan Pulsar and then a Mazda 626 in this series, with a highlight of finishing runner up to Phil Morriss in the 1994 Australian Production Car Championship.	He ~~later~~ raced a Nissan Pulsar and then a Mazda 626 ~~in this series, with a highlight of~~ finishing runner up ~~to Phil Morriss~~ in the 1994 Australian Production Car Championship.	Murray Carter raced a Nissan Pulsar and finished as a runner up in the 1994 Australian Production Car Championship.	Murray Carter raced a Nissan Pulsar and finished as runner up in the 1994 Australian Production Car Championship.
On July 6, 2008, Webb failed to qualify for the Beijing Olympics in the 1500 m after finishing 5th in the US Olympic Trials in Eugene, Oregon with a time of 3:41.62.	On July 6, 2008, Webb ~~failed to qualify for the Beijing Olympics in the 1500 m after~~ finishing 5th in the ~~US~~ Olympic Trials in Eugene, Oregon with a time of 3:41.62.	On July 6, 2008, Webb finishing 5th in the Olympic Trials in Eugene, Oregon with a time of 3:41.62.	On July 6, 2008, Webb finished 5th in the Olympic Trials in Eugene, Oregon, with a time of 3:41.62.
Out of the 17,219 inhabitants, 77 percent were 20 years of age or older and 23 percent were under the age of 20.	~~Out of the~~ 17,219 inhabitants ~~, 77 percent were 20 years of age or older and 23 percent were under the age of 20~~.	Rawdat Al Khail had a population of 17,219 inhabitants.	Rawdat Al Khail had a population of 17,219 inhabitants.

Table 3: Examples of annotation process. Deletions are indicated in red strikeouts, while added named entities are indicated in underlined blue. Significant grammar fixes are denoted in orange.

they are instructed to make at most one replacement.[4] This transforms the sentence yet again: $s_{\text{deletion}} \rightarrow s_{\text{decontext}}$. In Table 1, the annotator replaced *he* with *Cristhian Stuani*.

Since the previous steps can lead to ungrammatical sentences, annotators are also instructed to fix the grammar to improve the fluency of the sentence. We find that $s_{\text{decontext}}$ is different than s_{deletion} 68.3% of the time, and the average sentence length increases to 17.2 tokens for $s_{\text{decontext}}$ compared to 15.9 for s_{deletion}.

5.2 Secondary Annotation Task

Due to the complexity of the primary annotation task, the resulting sentence $s_{\text{decontext}}$ may still have grammatical errors, even if annotators were instructed to fix grammar. Thus, a second set of annotators were asked to further correct the sentence and were shown the table with highlighted cells as additional context. However, the annotators were not required to use the table. They were asked to determine the grammaticality and fluency of the provided sentence. If the sentence is not fluent or grammatical, they fix the errors to make it such. Annotators are also given an option to indicate that the sentence is not fixable.

This results in the final sentence s_{final}. On average, annotators edited the sentence 27.0% of the time, and the sentence length slightly increased to 17.4 tokens from 17.2. We found that for most of the cases, the table is not necessary to fix the sentence since the grammatical errors are due to surface syntax, such as a missing punctuation or a missing determiner. In a few cases, a verb may be missing, and in such instances, the table is needed to indicate the correct verb to use.

Property	**Value**
Training set size	120,761
Number of target tokens	1,268,268
Avg Target Length (tokens)	17.4
Target vocabulary size	136,777
Unique Tables	83,141
Rows per table (Median/Avg)	16 / 32.7
Cells per table (Median/Avg)	87 / 206.6
No. of Highlighted Cell (Median/Avg)	3 / 3.55
Development set size	7,700
Test set size	7,700

Table 4: ToTTo dataset statistics.

6 Dataset Analysis

Basic statistics of ToTTo are described in Table 4. The number of unique tables and vocabulary size attests to the open domain nature of our dataset. Furthermore, while the median table is actually quite large (87 cells), the median number of highlighted cells is significantly smaller (3). This indicates the importance of the cell highlighting feature of our dataset toward a well-defined text generation task.

6.1 Annotator Agreement

Table 5 shows annotator agreement over the development set for each step of the annotation process. We compute annotator agreement and Fleiss' kappa (Fleiss, 1971) for table readability and highlighted cells, and BLEU-4 score between annotated sentences in different stages, including (1) sentence after deletion; (2) sentence after decontextualization;

[4] Based on manual examination of a subset of 100 examples, all of them could be decontextualized with only one replacement. Allowing annotators to make multiple replacements led to excessive clarification.

Annotation Stage	Measure	Result
Table Readability	Agreement / κ	99.38 / 0.646
Cell Highlighting	Agreement / κ	73.74 / 0.856
After Deletion	BLEU-4	82.19
After Decontextualization	BLEU-4	72.56
Final	BLEU-4	68.98

Table 5: Annotator agreement over the development set. If possible, we measure the total agreement (in %) and the Fleiss' Kappa (κ). Otherwise, we report the BLEU-4 between annotators.

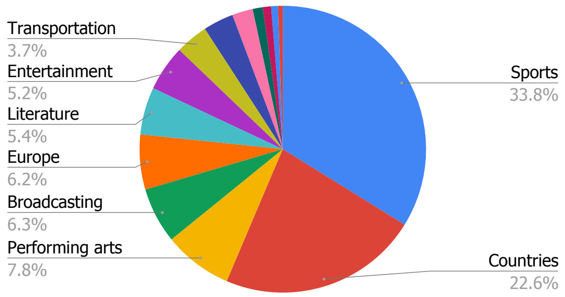

Figure 1: Topic distribution of our dataset.

and (3) final sentence after the secondary grammar correction task.

As one can see, the table readability task has an agreement of 99.38%. The cell highlighting task is more challenging. 73.74% of the time all three annotators completely agree on the set of cells which means that they chose the exact same set of cells. The Fleiss' kappa is 0.856, which is regarded as "almost perfect agreement" (0.81 - 1.00) according to (Landis and Koch, 1977).

With respect to the sentence revision tasks, we see that the agreement slightly degrades as more steps are performed. We compute single reference BLEU among all pairs of annotators for examples in our development set (which only contains examples where both annotators chose $\boldsymbol{t}_{\text{highlight}} \neq \emptyset$). As the sequence of revisions are performed, the annotator agreement gradually decreases in terms of BLEU-4: 82.19 → 72.56 → 68.98. This is considerably higher than the BLEU-4 between the original sentence $\boldsymbol{s}$ and $\boldsymbol{s}_{\text{final}}$ (45.87).

6.2 Topics and Linguistic Phenomena

We use the Wikimedia Foundation's topic categorization model (Asthana and Halfaker, 2018) to sort the categories of Wikipedia articles where the tables come from into a 44-category ontology.[5] Figure 1 presents an aggregated topic analysis of our dataset. We found that the *Sports* and *Countries* topics together comprise 56.4% of our dataset, but the other 44% is composed of a much broader set of topics such as *Performing arts*, *Transportation*, and *Entertainment*.

Types	Percentage
Require reference to page title	82%
Require reference to section title	19%
Require reference to table description	3%
Reasoning (logical, numerical, temporal etc.)	21%
Comparison across rows / columns / cells	13%
Require background information	12%

Table 6: Distribution of different linguistic phenomena among 100 randomly chosen sentences.

Table 6 summarizes the fraction of examples that require reference to the metadata, as well as some of the challenging linguistic phenomena in the dataset that potentially pose new challenges to current systems. Please refer to Figures 2- 6 in the Appendix for more complex examples.

6.3 Training, Development, and Test Splits

Each annotation consists of the set of highlighted cells $\boldsymbol{t}_{\text{highlight}}$ and the modified sentences $\boldsymbol{s}_{\text{deletion}}, \boldsymbol{s}_{\text{decontext}}, \boldsymbol{s}_{\text{final}}$. After the annotation process, we only consider examples where the sentence is related to the table, i.e., $\boldsymbol{t}_{\text{highlight}} \neq \emptyset$. This initially results in a training set $\boldsymbol{D}_{\text{orig-train}}$ of size 131,849 that we further filter as described below.

For more robust evaluation, each example in the development and test sets was annotated by three annotators. Since the machine learning task uses $\boldsymbol{t}_{\text{highlight}}$ as an input, it is challenging to use three different sets of highlighted cells in evaluation. Thus, we only use a single randomly chosen $\boldsymbol{t}_{\text{highlight}}$ while using the three $\boldsymbol{s}_{\text{final}}$ as references for evaluation[6]. We only use examples where at least 2 of the 3 annotators chose $\boldsymbol{t}_{\text{highlight}} \neq \emptyset$. This results in a development set $\boldsymbol{D}_{\text{dev}}$ of size 7,700 and a test set $\boldsymbol{D}_{\text{test}}$ of size 7,700.

Overlap and Non-Overlap Sets Without any modification $\boldsymbol{D}_{\text{orig-train}}$, $\boldsymbol{D}_{\text{dev}}$, and $\boldsymbol{D}_{\text{test}}$ may contain many similar tables. Thus, to increase the generalization challenge, we filter $\boldsymbol{D}_{\text{orig-train}}$ to remove some examples based on overlap with $\boldsymbol{D}_{\text{dev}}, \boldsymbol{D}_{\text{test}}$.

For a given example $\boldsymbol{d}$, let $\boldsymbol{h}(\boldsymbol{d})$ denote its set of

[5] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Council/Directory

[6] We don't use union or intersection because this may result in a set of highlighted cells that doesn't directly correspond to any of the references.

header values and similarly let $\boldsymbol{h}(\boldsymbol{D})$ be the set of header values for a given dataset $\boldsymbol{D}$. We remove examples $\boldsymbol{d}$ from the training set where $\boldsymbol{h}(\boldsymbol{d})$ is both rare in the data as well as occurs in either the development or test sets. Specifically, $\boldsymbol{D}_{\text{train}}$ is defined as:

$$\boldsymbol{D}_{\text{train}} := \{\boldsymbol{d} : \boldsymbol{h}(\boldsymbol{d}) \notin (\boldsymbol{h}(\boldsymbol{D}_{\text{dev}}) \cup \boldsymbol{h}(\boldsymbol{D}_{\text{test}})) \text{ or } \\ \text{count}\big(\boldsymbol{h}(\boldsymbol{d}), \boldsymbol{D}_{\text{orig-train}}\big) > \kappa\}.$$

The $\text{count}(\boldsymbol{h}(\boldsymbol{d}), \boldsymbol{D}_{\text{orig-train}})$ function returns the number of examples in $\boldsymbol{D}_{\text{orig-train}}$ with header $\boldsymbol{h}(\boldsymbol{d})$. To choose the hyperparameter κ we first split the test set as follows:

$$\boldsymbol{D}_{\text{test-overlap}} := \{\boldsymbol{d} : \boldsymbol{h}(\boldsymbol{d}) \in \boldsymbol{h}(\boldsymbol{D}_{\text{train}})\}$$
$$\boldsymbol{D}_{\text{test-nonoverlap}} := \{\boldsymbol{d} : \boldsymbol{h}(\boldsymbol{d}) \notin \boldsymbol{h}(\boldsymbol{D}_{\text{train}})\}$$

The development set is analogously divided into $\boldsymbol{D}_{\text{dev-overlap}}$ and $\boldsymbol{D}_{\text{dev-nonoverlap}}$.

We then choose $\kappa = 5$ so that $\boldsymbol{D}_{\text{test-overlap}}$ and $\boldsymbol{D}_{\text{test-nonoverlap}}$ have similar size. After filtering, the size of $\boldsymbol{D}_{\text{train}}$ is 120,761, and $\boldsymbol{D}_{\text{dev-overlap}}$, $\boldsymbol{D}_{\text{dev-nonoverlap}}$, $\boldsymbol{D}_{\text{test-overlap}}$, and $\boldsymbol{D}_{\text{test-nonoverlap}}$ have sizes 3784, 3916, 3853, and 3847 respectively.

7 Machine Learning Task Construction

In this work, we focus the following task:

Given a table $\boldsymbol{t}$ and related metadata $\boldsymbol{m}$ (page title, section title, table section text) and a set of highlighted cells $\boldsymbol{t}_{\text{highlight}}$, produce the final sentence $\boldsymbol{s}_{\text{final}}$. Mathematically this can be described as learning a function $\boldsymbol{f} : \boldsymbol{x} \rightarrow \boldsymbol{y}$ where $\boldsymbol{x} = (\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{t}_{\text{highlight}})$ and $\boldsymbol{y} = \boldsymbol{s}_{\text{final}}$.

Note that this task is different from what the annotators perform, since they are provided with a starting sentence requiring revision. Therefore, this task is more challenging, as the model must generate a new sentence instead of revising an existing sentence. Since we use several stages in our annotation mechanism, one could design several other tasks for machine learning models given the data such as sentence revision or cell highlighting, but we leave this out of the scope of this work.

8 Experiments

We present baseline results on ToTTo by examining three existing state-of-the-art approaches:

- BERT-to-BERT (Rothe et al., 2019): A Transformer encoder-decoder model (Vaswani et al., 2017) where the encoder and decoder are both initialized with BERT (Devlin et al., 2018). The original BERT model is pre-trained with both Wikipedia and the Books corpus (Zhu et al., 2015), the former of which contains our (unrevised) test targets. Thus, we also pre-train a version of BERT on the Books corpus only, which we consider a more correct baseline. However, empirically we find that both models perform similarly in practice (Table 7).
- Pointer-Generator (See et al., 2017): A Seq2Seq model with attention and copy mechanism (our implementation).
- Puduppully et al. (2019): A Seq2Seq model with an explicit content selection and planning mechanism designed for data-to-text.

Moreover, we explore different strategies of representing the source content that resemble standard linearization approaches in the literature (Lebret et al., 2016; Wiseman et al., 2017).

- **Full Table** The simplest approach is simply to use the entire table as the source, adding special tokens to mark which cells have been highlighted. However, many tables can be very large and this strategy performs poorly.
- **Subtable** Another option is to only use the highlighted cells $\boldsymbol{t}_{highlight} \in \boldsymbol{t}$ with the heuristically extracted row and column header for each highlighted cell. This makes it easier for the model to only focus on relevant content but limits the ability to perform reasoning in the context of the table structure (see Table 10). Overall though, we find this representation leads to higher performance.

In all cases, the selected cells are linearized with row and column separator tokens. We also experiment with prepending the table metadata to the source table.[7]

8.1 Evaluation metrics

The model output is evaluated using two automatic metrics. Human evaluation is described in § 8.3.

BLEU (Papineni et al., 2002): A widely used metric that uses n-gram overlap between the reference $\boldsymbol{y}$ and the prediction $\hat{\boldsymbol{y}}$ at the corpus level. BLEU does not take the source content $\boldsymbol{x}$ into account.

PARENT (Dhingra et al., 2019): A metric recently proposed specifically for data-to-text evaluation

[7] The table section text is ignored, since it is usually missing or irrelevant.

that takes the table into account. PARENT is defined at an instance level. For a given example $(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ PARENT is defined as:

$$PARENT(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = \frac{2 \times E_p(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) \times E_r(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)}{E_p(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) + E_r(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)}$$

$E_p(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ is the PARENT precision computed using the prediction, reference, and table (the last of which is not used in BLEU). $E_r(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ is the PARENT recall and is computed as:

$$E_r(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = R(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)^{(1-\lambda)} R(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)^{\lambda}$$

where $R(\boldsymbol{x}_n, \boldsymbol{y}_n, \hat{\boldsymbol{y}}_n)$ is a recall term that compares the prediction with both the reference and table. $R(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)$ is an extra recall term that gives an additional reward if the prediction $\hat{\boldsymbol{y}}_n$ contains phrases in the table $\boldsymbol{x}_n$ that are not necessarily in the reference (λ is a hyperparameter).

In the original PARENT work (Dhingra et al., 2019), the same table $\boldsymbol{t}$ is used for computing the precision and both recall terms. While this makes sense for most existing datasets, it does not take into account the highlighted cells $\boldsymbol{t}_{highlight}$ in our task. To incorporate $\boldsymbol{t}_{highlight}$, we modify the PARENT metric so that the additional recall term $R(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)$ uses $\boldsymbol{t}_{highlight}$ instead of $\boldsymbol{t}$ to only give an additional reward for relevant table information. The other recall and the precision term still use $\boldsymbol{t}$.

8.2 Results

Table 7 shows our results against multiple references with the subtable input format. Both the BERT-to-BERT models perform the best, followed by the pointer generator model.[8] We see that for all models the performance on the non-overlap set is significantly lower than that of the overlap set, indicating that slice of our data poses significant challenges for machine learning models. We also observe that the baseline that separates content selection and planning performs quite poorly. We attest this to the fact that it is engineered to the Rotowire data format with fixed size tables and predefined column names.

Table 8 explores the effects of the various input representations (subtable vs. full table) on the BERT-to-BERT model. We see that the full table format performs poorly even if it is the most knowledge-preserving representation. Using table metadata significantly helps under different input.

8.3 Human evaluation

For each of the 2 top performing models in Table 7, we take 500 random outputs and perform human evaluation using the following axes:

- **Fluency** - A candidate sentence is fluent if it is grammatical and natural. The three choices are *Fluent*, *Mostly Fluent*, *Not Fluent*.
- **Faithfulness** (Precision) - A candidate sentence is considered faithful if all pieces of information are supported by either the table or one of the references. Any piece of unsupported information makes the candidate unfaithful.
- **Covered Cells** (Recall) - The percentage of highlighted cells the candidate sentence covers.
- **Coverage with Respect to Reference** (Recall) - We ask whether the candidate is strictly more or less informative than each reference (or neither, which is referred to as neutral).

In addition to evaluating the model outputs, we compute an oracle upper-bound by treating one of the references as a candidate and evaluating it compared to the table and other references. The results, shown in Table 9, attest to the high quality of our human annotations since the oracle consistently achieves high performance. All the axes demonstrate that there is a considerable gap between the model and oracle performance.

This difference is most easily revealed in the last column when annotators are asked to directly compare the candidate and reference. As expected, the oracle has similar coverage to the reference (61.7% neutral) but both baselines demonstrate considerably less coverage. According to an independent-sample t-test, this difference is significant at a $p < 0.001$ level for both baselines. Similarly, we observe a significantly lower percentage of covered cells for the baselines compared to the reference according to a χ^2 test. Comparing the baselines to each other, we do not observe a significant difference in either coverage metric.

Furthermore, the baselines are considerably less faithful than the reference. The faithfulness of both models is significantly lower than the reference (χ^2 test with $p < 0.001$). The models do not differ significantly from each other, except for the non-overlap case, where we see a moderate effect

[8] Note the BLEU scores are relatively high due to the fact that our task is more controlled than other text generation tasks and that we have multiple references.

Model	Overall		Overlap Subset		Nonoverlap Subset	
	BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
BERT-to-BERT (Books+Wiki)	**44.0**	**52.6**	**52.7**	**58.4**	**35.1**	**46.8**
BERT-to-BERT (Books)	43.9	**52.6**	**52.7**	**58.4**	34.8	46.7
Pointer-Generator	41.6	51.6	50.6	58.0	32.2	45.2
Puduppully et al. (2019)	19.2	29.2	24.5	32.5	13.9	25.8

Table 7: Performance compared to multiple references on the test set for the subtable input format with metadata.

Data Format	**BLEU**	**PARENT**
subtable w/ metadata	43.9	52.6
subtable w/o metadata	36.9	42.6
full table w/ metadata	26.8	30.7
full table w/o metadata	20.9	22.2

Table 8: Multi-reference performance of different input representations for BERT-to-BERT Books model.

favoring the book model. While it is well known that neural methods struggle with faithfulness in the presence of noisy references (Wiseman et al., 2017; Tian et al., 2019), our results indicate it is a problem even when the references are clean.

9 Model Errors and Challenges

In this section, we visualize some example decoder outputs from the BERT-to-BERT Books model (Table 10) and discuss specific challenges that existing approaches face with this dataset. In general, the model performed reasonably well in producing grammatically correct and fluent sentences given the information from the table, as indicated by Table 10. Given the "full table", the model is not able to correctly select the information needed to produce the reference, and instead produces an arbitrary sentence with irrelevant information. Note the model corrects itself with highlighted cell information ("subtable"), and learns to use the metadata to improve the sentence. However, we also observe certain challenges that existing approaches are struggling with, which can serve as directions for future research. In particular:

Hallucination As shown in Table 10 (examples 1-4) the model sometimes outputs phrases such as *first scottish*, *third* that seem reasonable but are not faithful to the table. This hallucination phenomenon has been widely observed in other existing data-to-text datasets (Lebret et al., 2016; Wiseman et al., 2017). However, the noisy references in these datasets make it difficult to disentangle model incapability from data noise. Our dataset serves as strong evidence that even when the reference targets are faithful to the source, neural models still struggle with faithfulness.

Rare topics Another challenge revealed by the open domain nature of our task is that models often struggle with rare or complex topics. For instance, example 5 of Table 10 concerns microdrive capacities which is challenging. As our topic distribution indicates (Figure 1), certain topics have relatively limited training examples. This calls for the development of models that can be learned with limited examples and better generalization-ability.

Diverse table structure In example 6, inferring *six* and *five* correctly requires counting table rows and columns. Similarly, in the last example of Table 10, the phrases *last* and *career highs* can be deduced from the table structure and with comparisons over the columns. However, the models are unable to easily make these inferences from the simplistic source representation that we used. Our dataset presents a unique challenge for learning better table representation due to its various types of table schema. Please see Figures 2-6 for more example tables.

Numerical reasoning As discussed above, reasoning over the table structure often requires counting rows or columns, or comparing numbers over a set of cells. In addition to examples 6 and 7, example 4 requires comparing numbers to conclude that *third* is an incorrect relation. The model errors indicate numerical reasoning remains challenging for generation systems. Recent attention to this problem in question answering (Dua et al., 2019; Andor et al., 2019) may be relevant for our task.

Evaluation metrics Many of the above issues are difficult to capture with metrics like BLEU since the reference and prediction may only differ by a word but largely differ in terms of semantic meaning. Furthermore, it is unclear how to correctly reward models that produce output with more inferences, urging for better metrics possibly

	Model	Fluency (%)	Faithfulness (%)	Covered Cells (%)	Less/Neutral/More Coverage w.r.t. Ref
Overall	*Oracle*	*99.3*	*93.6*	*94.8*	*18.3 / 61.7 / 20.0*
	BERT-to-BERT (Books)	88.1	76.2	89.0	49.2 / 36.2 / 14.5
	BERT-to-BERT (Books+Wiki)	87.3	73.6	87.3	53.9 / 32.9 / 13.2
Overlap	*Oracle*	*99.6*	*96.5*	*95.5*	*19.8 / 62.8 / 17.4*
	BERT-to-BERT (Books)	89.6	78.7	92.1	42.0 / 43.7 / 14.3
	BERT-to-BERT (Books+Wiki)	89.8	81.1	91.0	47.8 / 39.2 / 13.1
Non-overlap	*Oracle*	*99.1*	*91.4*	*94.3*	*17.0 / 60.9 / 22.1*
	BERT-to-BERT (Books)	86.9	74.2	86.4	55.5 / 29.8 / 14.7
	BERT-to-BERT (Books+Wiki)	84.8	66.6	83.8	60.1 / 26.6 / 13.3

Table 9: Human evaluation over references (to compute *Oracle*) and model outputs. For Fluency, we report the percentage of outputs that were completely fluent. In the last column $X/Y/Z$ means X% and Z% of the candidates were deemed to be less and more informative than the reference respectively and Y% were neutral.

ID	Reference	Decoder output (w/ metadata)		w/o metadata
		Full table	Subtable	Subtable
1	in the 2012 film pizza bagel, michael pillarella portrays tommy.	in 2012, groff played the role of tommy in the film pizza bagel.	in 2012, pillarella appeared as tommy in the film pizza bagel.	harris played the role of tommy in the 2012 film pizza bagel.
2	the album shari addison placed at no. 176 on the billboard 200 along with no. 5 on the gospel albums.	shari addison's " 5", reached number 176 on the billboard 200.	shari addison charted at number 176 on the us chart and at number 5 on the us billboard 200.	the shari addison peaked at number 176 on the billboard 200 chart.
3	in the 1939 currie cup, western province lost to transvaal by 176 in cape town.	northern transvaal and western province were drawn by 16 in 1989 and in 1995, western province were defeated by the sharks in durban.	the first currie cup was played in 1939 in transvaal1 at newlands, with western province winning 176.	the first scottish cup was played in 1939 in transvaal1, where transvaal1 defeated western province 176 in the final at newlands in cape town.
4	the bnp secured their best general election result in oldham west and royton where nick griffin secured 16.4% of the votes.	bnp results ranged from 278 to 6,552 votes.	in the british national party election, nick griffin placed third with 16.4% of the vote.	in oldham west and royton, nick griffin won 16.4% of the vote.
5	a second generation of microdrive was announced by ibm in 2000 with increased capacities at 512 mb and 1 gb.	the microdrive models formed 512 megabyte and 1 gigabyte in 2000.	there were 512 microdrive models in 2000: 1 gigabyte.	cortete's production was 512 megabyte.
6	the 1956 grand prix motorcycle racing season consisted of six grand prix races in five classes: 500cc, 350cc, 250cc, 125cc and sidecars 500cc.	the 1966 grand prix motorcycle racing season consisted of seven grand prix races in five classes: 500cc, 350cc, 250cc, 125cc and sidecars 500cc.	the 1956 grand prix motorcycle racing season consisted of eight grand prix races in five classes: 500cc, 350cc, 250cc, 125cc and sidecars 500cc.	the 1955 grand prix motorcycle racing season consisted of eight grand prix races in five classes: 500cc, 350cc, 250cc, 125cc and sidecars 500cc.
7	in travis kelce's last collegiate season, he set personal career highs in receptions (45), receiving yards (722), yards per receptions (16.0) and receiving touchdowns (8).	during the 2011 season, travis kelceum caught 76 receptions for 1,612 yards and 14 touchdowns.	travis kelce finished the 2012 season with 45 receptions for 722 yards (16.0 avg.) and eight touchdowns.	kelce finished the 2012 season with 45 catches for 722 yards (16.0 avg.) and eight touchdowns.

Table 10: Decoder output examples from BERT-to-BERT Books models on the development set. The "subtable with metadata" model achieves the highest BLEU. Red indicates model errors and blue denotes interesting reference language not in the model output.

built on learned models (Wiseman et al., 2017; Ma et al., 2019; Sellam et al., 2020) for appropriate evaluation.

10 Conclusion

In this work, we presented ToTTo, a large, English table-to-text dataset that presents both a controlled generation task and a data annotation process based on iterative sentence revision. We also provided several state-of-the-art baselines, and demonstrated ToTTo could be a useful dataset for modeling research as well as for developing evaluation metrics that can better detect model improvements. ToTTo is available at https://github.com/google-research-datasets/totto.

Acknowledgements

The authors wish to thank Ming-Wei Chang, Jonathan H. Clark, Kenton Lee, and Jennimaria Palomaki for their insightful discussions and support. Many thanks also to Ashwin Kakarla and his team for help with the annotations.

References

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *Proc. of EMNLP*.

Sumit Asthana and Aaron Halfaker. 2018. With few eyes, all hoaxes are deep. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):21.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proc. of EMNLP*.

Eva Banik, Claire Gardent, and Eric Kow. 2013. The kbgen challenge. In *Proc. of European Workshop on NLG*.

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proc. of ICML*.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proc. of ACL*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. TabFact: A large-scale dataset for table-based fact verification. In *Proc. of ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017a. Creating training corpora for NLG micro-planning. In *Proc. of ACL*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017b. The WebNLG challenge: Generating text from RDF data. In *Proc. of INLG*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proc. of NAACL*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *TACL*, 6:437–450.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proc. of WMT*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proc. of EMNLP*.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *Proc. of ACL*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proc. of EMNLP*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. In *Open Review*.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proc. of ACL*.

Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. 2010. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 57–66.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proc. of AAAI*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90.

Inderjeet Mani. 1999. *Advances in automatic text summarization*. MIT press.

Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proc. of SIGDIAL*.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing nlg data: Pictures elicit better data. In *Proc. of INLG*.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proc. of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proc. of ACL*.

Hao Peng, Ankur P Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. 2019. Text generation with exemplar-based adaptive decoding. In *Proc. of NAACL*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In **SEM@NAACL-HLT*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proc. of AAAI*.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2019. Leveraging pre-trained checkpoints for sequence generation tasks. *arXiv preprint arXiv:1907.12461*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proc. of ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NIPS*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proc. of ICML Deep Learning Workshop*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. of EMNLP*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. In *Proc. of EMNLP*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proc. of ICCV*.

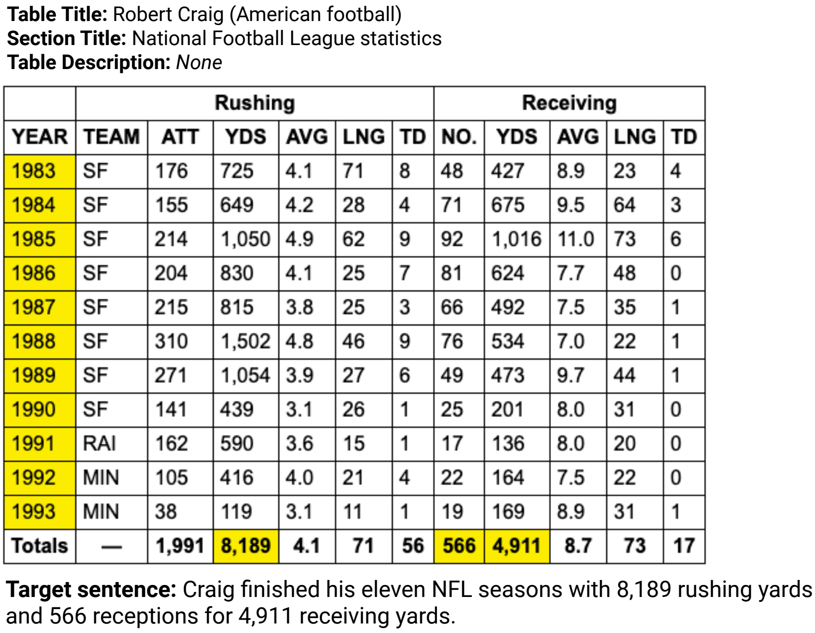

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description: *None*

	Rushing						Receiving				
YEAR	**TEAM**	**ATT**	**YDS**	**AVG**	**LNG**	**TD**	**NO.**	**YDS**	**AVG**	**LNG**	**TD**
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1,050	4.9	62	9	92	1,016	11.0	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1,502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1,054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**—**	**1,991**	**8,189**	**4.1**	**71**	**56**	**566**	**4,911**	**8.7**	**73**	**17**

Target sentence: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Figure 2: ToTTo example with numerical reasoning about table cells.

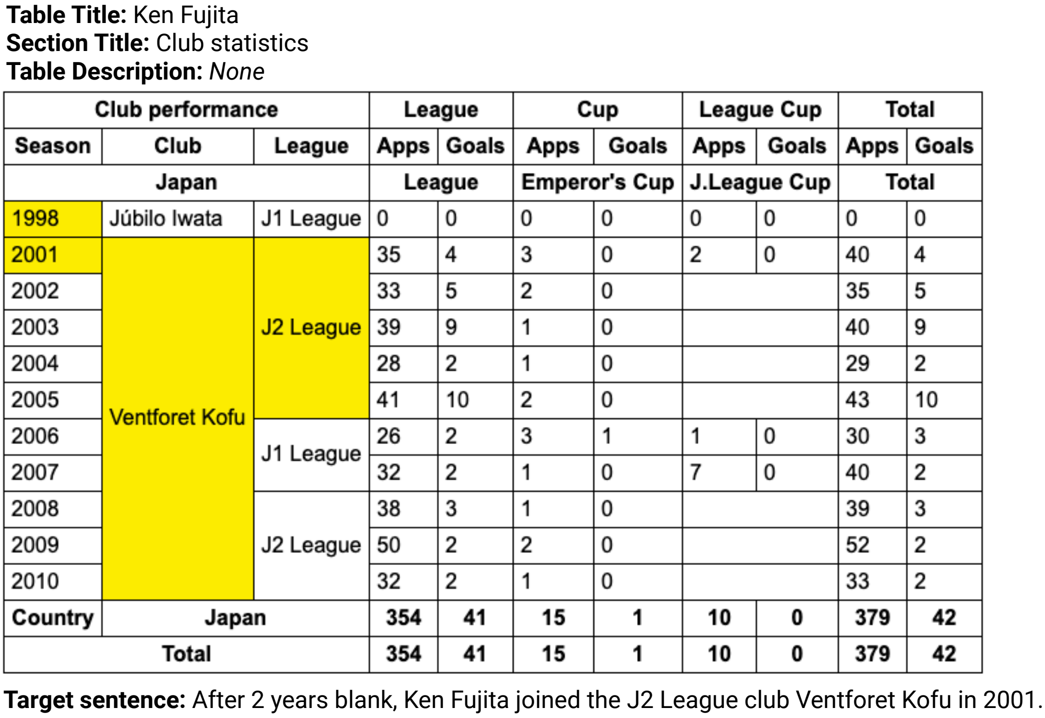

Table Title: Ken Fujita
Section Title: Club statistics
Table Description: *None*

Club performance			**League**		**Cup**		**League Cup**		**Total**	
Season	**Club**	**League**	**Apps**	**Goals**	**Apps**	**Goals**	**Apps**	**Goals**	**Apps**	**Goals**
Japan			**League**		**Emperor's Cup**		**J.League Cup**		**Total**	
1998	Júbilo Iwata	J1 League	0	0	0	0	0	0	0	0
2001	Ventforet Kofu	J2 League	35	4	3	0	2	0	40	4
2002			33	5	2	0			35	5
2003			39	9	1	0			40	9
2004			28	2	1	0			29	2
2005			41	10	2	0			43	10
2006		J1 League	26	2	3	1	1	0	30	3
2007			32	2	1	0	7	0	40	2
2008		J2 League	38	3	1	0			39	3
2009			50	2	2	0			52	2
2010			32	2	1	0			33	2
Country	**Japan**		**354**	**41**	**15**	**1**	**10**	**0**	**379**	**42**
Total			**354**	**41**	**15**	**1**	**10**	**0**	**379**	**42**

Target sentence: After 2 years blank, Ken Fujita joined the J2 League club Ventforet Kofu in 2001.

Figure 3: ToTTo example with complex table structure and temporal reasoning.

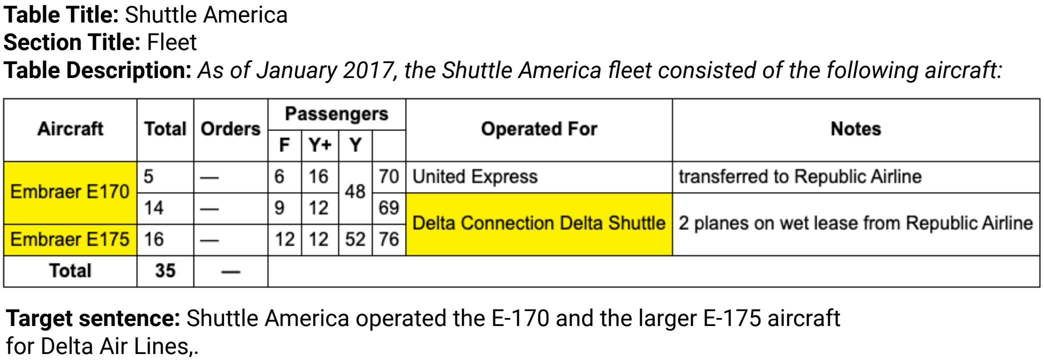

Table Title: Shuttle America
Section Title: Fleet
Table Description: *As of January 2017, the Shuttle America fleet consisted of the following aircraft:*

Aircraft	Total	Orders	Passengers				Operated For	Notes
			F	Y+	Y			
Embraer E170	5	—	6	16	48	70	United Express	transferred to Republic Airline
	14	—	9	12		69	Delta Connection Delta Shuttle	2 planes on wet lease from Republic Airline
Embraer E175	16	—	12	12	52	76		
Total	35	—						

Target sentence: Shuttle America operated the E-170 and the larger E-175 aircraft for Delta Air Lines,.

Figure 4: TOTTO example with rare topics and complex table structure.

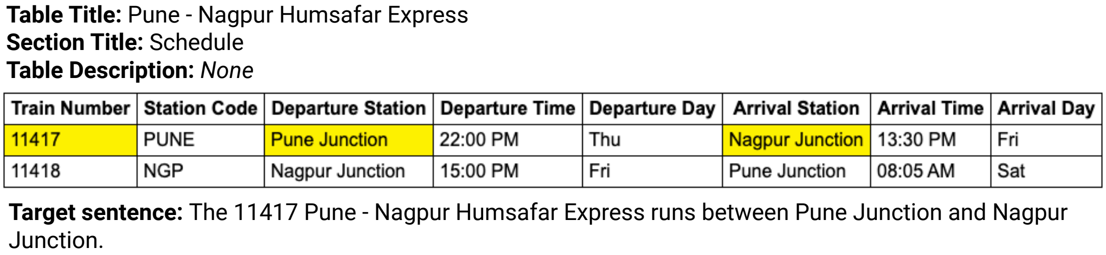

Table Title: Pune - Nagpur Humsafar Express
Section Title: Schedule
Table Description: *None*

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 5: TOTTO example with rare topic.

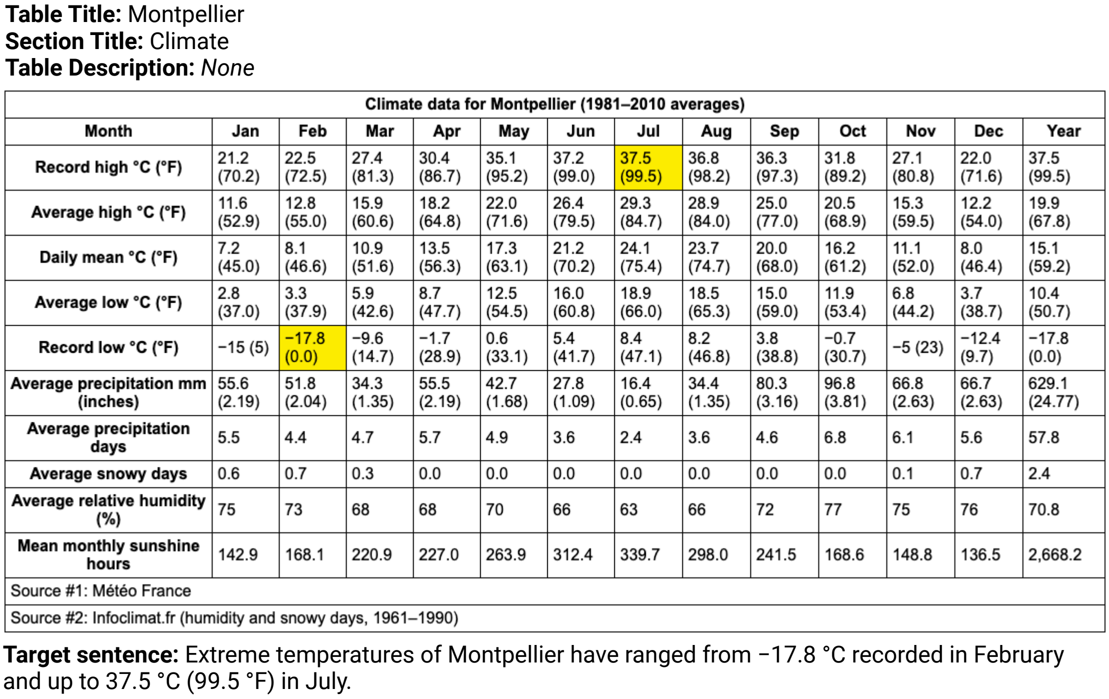

Table Title: Montpellier
Section Title: Climate
Table Description: *None*

Climate data for Montpellier (1981–2010 averages)													
Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
Record high °C (°F)	21.2 (70.2)	22.5 (72.5)	27.4 (81.3)	30.4 (86.7)	35.1 (95.2)	37.2 (99.0)	37.5 (99.5)	36.8 (98.2)	36.3 (97.3)	31.8 (89.2)	27.1 (80.8)	22.0 (71.6)	37.5 (99.5)
Average high °C (°F)	11.6 (52.9)	12.8 (55.0)	15.9 (60.6)	18.2 (64.8)	22.0 (71.6)	26.4 (79.5)	29.3 (84.7)	28.9 (84.0)	25.0 (77.0)	20.5 (68.9)	15.3 (59.5)	12.2 (54.0)	19.9 (67.8)
Daily mean °C (°F)	7.2 (45.0)	8.1 (46.6)	10.9 (51.6)	13.5 (56.3)	17.3 (63.1)	21.2 (70.2)	24.1 (75.4)	23.7 (74.7)	20.0 (68.0)	16.2 (61.2)	11.1 (52.0)	8.0 (46.4)	15.1 (59.2)
Average low °C (°F)	2.8 (37.0)	3.3 (37.9)	5.9 (42.6)	8.7 (47.7)	12.5 (54.5)	16.0 (60.8)	18.9 (66.0)	18.5 (65.3)	15.0 (59.0)	11.9 (53.4)	6.8 (44.2)	3.7 (38.7)	10.4 (50.7)
Record low °C (°F)	−15 (5)	−17.8 (0.0)	−9.6 (14.7)	−1.7 (28.9)	0.6 (33.1)	5.4 (41.7)	8.4 (47.1)	8.2 (46.8)	3.8 (38.8)	−0.7 (30.7)	−5 (23)	−12.4 (9.7)	−17.8 (0.0)
Average precipitation mm (inches)	55.6 (2.19)	51.8 (2.04)	34.3 (1.35)	55.5 (2.19)	42.7 (1.68)	27.8 (1.09)	16.4 (0.65)	34.4 (1.35)	80.3 (3.16)	96.8 (3.81)	66.8 (2.63)	66.7 (2.63)	629.1 (24.77)
Average precipitation days	5.5	4.4	4.7	5.7	4.9	3.6	2.4	3.6	4.6	6.8	6.1	5.6	57.8
Average snowy days	0.6	0.7	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.7	2.4
Average relative humidity (%)	75	73	68	68	70	66	63	66	72	77	75	76	70.8
Mean monthly sunshine hours	142.9	168.1	220.9	227.0	263.9	312.4	339.7	298.0	241.5	168.6	148.8	136.5	2,668.2
Source #1: Météo France													
Source #2: Infoclimat.fr (humidity and snowy days, 1961–1990)													

Target sentence: Extreme temperatures of Montpellier have ranged from −17.8 °C recorded in February and up to 37.5 °C (99.5 °F) in July.

Figure 6: TOTTO example with interesting reference language.