WIKITABLET: A Large-Scale Data-to-Text Dataset for Generating Wikipedia Article Sections

Mingda Chen **Sam Wiseman** **Kevin Gimpel**
Toyota Technological Institute at Chicago, Chicago, IL, 60637, USA
{mchen,swiseman,kgimpel}@ttic.edu

Abstract

Datasets for data-to-text generation typically focus either on multi-domain, single-sentence generation or on single-domain, long-form generation. In this work, we cast generating Wikipedia sections as a data-to-text generation task and create a large-scale dataset, WIKITABLET, that pairs Wikipedia sections with their corresponding tabular data and various metadata. WIKITABLET contains millions of instances, covering a broad range of topics, as well as a variety of flavors of generation tasks with different levels of flexibility. We benchmark several training and decoding strategies on WIKITABLET. Our qualitative analysis shows that the best approaches can generate fluent and high quality texts but they struggle with coherence and factuality, showing the potential for our dataset to inspire future work on long-form generation.[1]

1 Introduction

Data-to-text generation (Kukich, 1983; McKeown, 1992) is the task of generating text based on structured data. Most existing data-to-text datasets focus on single-sentence generation, such as WIKIBIO (Lebret et al., 2016), LogicNLG (Chen et al., 2020), and ToTTo (Parikh et al., 2020). Other datasets are relatively small-scale and focus on long-form text generation, such as ROTOWIRE (Wiseman et al., 2017) and MLB (Puduppully et al., 2019). In this work, we cast generating Wikipedia sections as a data-to-text generation task and build a large-scale dataset targeting multi-sentence data-to-text generation with a variety of domains and data sources.

To this end, we create a dataset that we call WIKITABLET ("Wikipedia Tables to Text") that pairs Wikipedia sections with their corresponding tabular data and various metadata. The data resources we consider are relevant either to entire Wikipedia articles, such as Wikipedia infoboxes and Wikidata tables, or to particular sections. Data from the latter category is built automatically from either naturally-occurring hyperlinks or from named entity recognizers. This data construction approach allows us to collect large quantities of instances while still ensuring the coverage of the information in the table. We also perform various types of filtering to ensure dataset quality.

WIKITABLET contains millions of instances covering a broad range of topics and a variety of flavors of generation with different levels of flexibility. Figure 1 shows two examples from WIKITABLET. The first instance has more flexibility as it involves generating a fictional character biography in a comic book, whereas the second is more similar to standard data-to-text generation tasks, where the input tables contain all of the necessary information for generating the text. While the open-ended instances in WIKITABLET are to some extent similar to story generation (Propp, 1968; McIntyre and Lapata, 2009; Fan et al., 2018), the fact that these instances are still constrained by the input tables enables different evaluation approaches and brings new challenges (i.e., being coherent and faithful to the input tables at the same time).

Because of the range of knowledge-backed generation instances in WIKITABLET, models trained on our dataset can be used in assistive writing technologies for a broad range of topics and types of knowledge. For example, technologies can aid students in essay writing by drawing from multiple kinds of factual sources. Moreover, WIKITABLET can be used as a pretraining dataset for other relatively small-scale data-to-text datasets (e.g., ROTOWIRE). A similar idea that uses data-to-text generation to create corpora for pretraining language models has shown promising results (Agarwal et al.,

[1]Code, data, and pretrained models are available at https://github.com/mingdachen/WikiTableT

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 193–209

2021).

In experiments, we train several baseline models on WIKITABLET and empirically compare training and decoding strategies. We find that the best training strategies still rely on enforcing hard constraints to avoid overly repetitive texts. Human evaluations reveal that (1) humans are unable to differentiate the human written texts from the generations from our neural models; (2) while the annotations show that grammatical errors in the reference texts and the generations may prevent humans from fully understanding the texts, the best decoding strategy (i.e., beam search with n-gram blocking (Paulus et al., 2018)) does not have such a problem and shows the best performance on several aspects; (3) the degree of topical similarity between the generations and the reference texts depends on the open-endedness of the instances.

Our analysis shows that the generations are fluent and generally have high quality, but the models sometimes struggle to generate coherent texts for all the involved entities, suggesting future research directions. For example, when the instance has a high degree of flexibility, we find the models making mistakes about what a particular entity type is capable of. We also find errors in terms of the factuality of the generated text, both in terms of contradictions relative to the tables and commonsense violations.

2 Related Work

There have been efforts in creating data-to-text datasets from various resources, including sports summaries (Wiseman et al., 2017; Puduppully et al., 2019), weather forecasts (Liang et al., 2009), and commentaries (Chen and Mooney, 2008). Most of the recent datasets focus on generating single sentences given tables, such as WIKIBIO, ToTTo, LogicNLG, and WikiTableText (Bao et al., 2018), or other types of data formats, such as data triples (Vougiouklis et al., 2017; Gardent et al., 2017; Nan et al., 2021), abstract meaning representations (Flanigan et al., 2016), minimal recursion semantics (Hajdik et al., 2019), or a set of concepts (Lin et al., 2020). Other than single sentences, there have been efforts in generating groups of sentences describing humans and animals (Wang et al., 2018), and generating a post-modifier phrase for a target sentence given a sentence context (Kang et al., 2019). In this work, our focus is long-form text generation and we are interested in automatically creating a large-scale dataset containing multiple types of data-to-text instances. As shown in Table 1, WIKITABLET differs from these datasets in that it is larger in scale and contains multi-sentence texts. More details are in the next section.

Wikipedia has also been used to construct datasets for other text generation tasks, such as generating Wikipedia movie plots (Orbach and Goldberg, 2020; Rashkin et al., 2020) and short Wikipedia event summaries (Gholipour Ghalandari et al., 2020), and summarizing Wikipedia documents (Zopf, 2018; Liu* et al., 2018) or summaries of aspects of interests (Hayashi et al., 2020) from relevant documents.

As part of this work involves finding aligned tables and text, it is related to prior work on aligning Wikipedia texts to knowledge bases (Elsahar et al., 2018; Logan et al., 2019).

3 The WIKITABLET Dataset

The WIKITABLET dataset pairs Wikipedia sections[2] with their corresponding tabular data and various metadata; some of this data is relevant to entire Wikipedia articles ("article data") or article structure ("title data"), while some is section-specific ("section data"). Each data table consists of a set of **records**, each of which is a tuple containing an **attribute** and a **value**.

The instances in WIKITABLET cover a range of flavors of language generation. Some have more flexibility, requiring models to generate coherent stories based on the entities and knowledge given in the tables. The first instance in Figure 1 is such an example. The text is from the Wikipedia article entitled "Wolfsbane (comics)" and resides within two nested sections: the higher-level section "Fictional character biography" and the lower-level section "Messiah Complex". The task is challenging as models need to generate a coherent passage that can connect all the entities in the section data, and the story also needs to fit the background knowledge provided in the article data.

Other instances are more similar to standard data-to-text generation tasks, where the input tables contain all the necessary information for generating

[2]We define a Wikipedia section to be all text starting after a (sub)section heading and proceeding until the next (sub)section heading. We include Wikipedia sections at various nesting levels. For example, a top level section may start with a few paragraphs describing general information followed by two subsections with more specific information, in which case the example will be converted into three instances in our dataset.

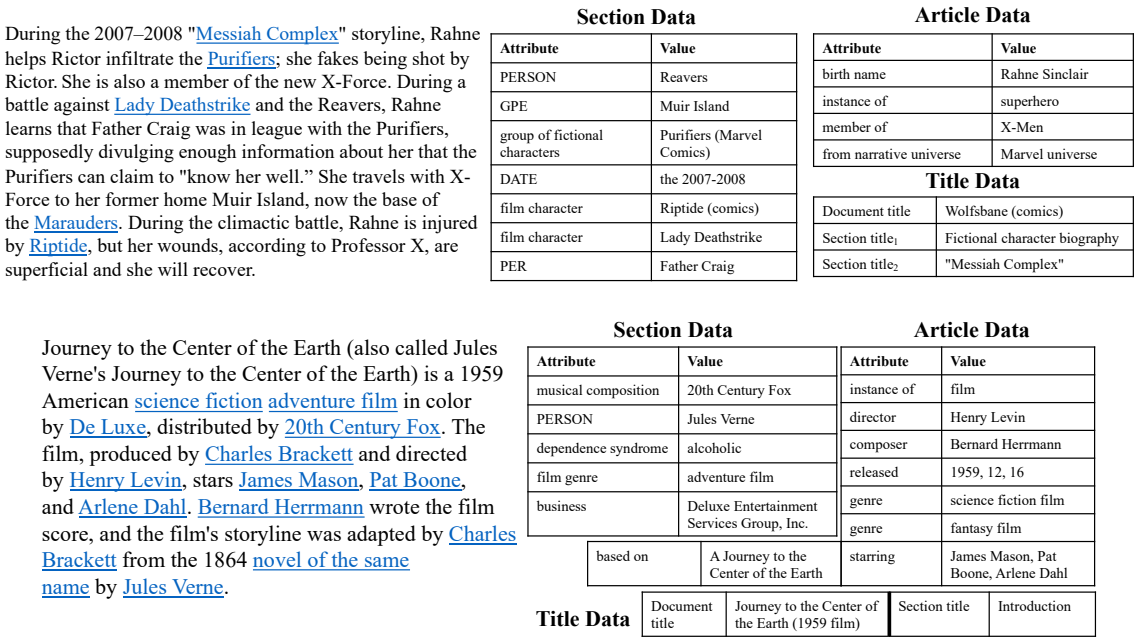

Figure 1: Two examples from WIKITABLET. Only parts of the tables are shown due to space constraints. Underlined texts are hyperlinks. Records with the attributes "DATE", "PER", "PERSON", or "GPE" are from NER. The subscripts for section titles indicate the ordering of nesting, where smaller numbers are for higher level sections.

the text. The second instance in Figure 1 is an example of this sort of task. However, these tasks are still challenging due to the wide variety of topics contained in WIKITABLET.

3.1 Dataset Construction

We begin by describing the steps we take to construct WIKITABLET. More details are in the supplementary material. In general, the steps can be split into two parts: collecting data tables and filtering out texts. When collecting data, we consider five resources: Wikidata tables, infoboxes in Wikipedia pages,[3] hyperlinks in the passage, named entities in the passage obtained from named entity recognition (NER), and Wikipedia article structure. For a given Wikipedia article, we use the same infobox and Wikidata table for all sections. These tables can serve as background knowledge for the article. For each section in the article, we create a second table corresponding to section-specific data, i.e., section data. The section data contains records constructed from hyperlinks and entities identified by a named entity recognizer.[4]

We form records for named entities by using the type of the entity as the attribute and the identified entity as the value. We form records for hyperlinks as follows. For the attribute, for a hyperlink with surface text t and hyperlinked article ℓ, we use the value of the "instance of" or "subclass of" tuple in the Wikidata table for ℓ. For example, the first instance in Figure 1 will be turned into a record with attribute "superhero" and value "Wolfsbane (comics)". If ℓ does not have a Wikidata table or no appropriate tuple, we consider the parent categories of ℓ. For the value of the tuple, we use the document title of ℓ rather than the actual surface text t to avoid giving away too much information in the reference text.

Complementary to the article data, we create a title table that provides information about the position in which the section is situated, which includes the article title and the section titles for the target section. As the initial sections in Wikipedia articles do not have section titles, we use the section title "Introduction" for these.

We also perform various filtering to ensure the quality of the data records, the coverage of the input data, and the length of the reference text. The final dataset contains approximately 1.5 million instances. We randomly sample 4533 instances as the development set and 4351 as the test set. We

[3]Wikidata is a consistently-structured knowledge base (e.g., has a fixed set of attributes), whereas infoboxes are not consistently-structured and this flexibility sometimes allows the infobox to contain extra information. Therefore, we consider using infoboxes as extra resources.

[4]We use the NER tagger from spaCy (Honnibal and Montani, 2017) and a BERT model (Devlin et al., 2019) finetuned on CoNLL03 data (Tjong Kim Sang and De Meulder, 2003).

also ensure that there are no overlapping Wikipedia articles among splits.

3.2 Dataset Characteristics

Table 1 shows statistics for WIKITABLET and related datasets. While the average length of a WIKITABLET instance is not longer than some of the existing datasets, WIKITABLET offers more diverse topics than the sports-related datasets ROTOWIRE and MLB, or the biography-related dataset WIKIBIO. Compared to the prior work that also uses Wikipedia for constructing datasets, WIKIBIO, LogicNLG, ToTTo, and DART (Nan et al., 2021) all focus on sentence generation, whereas WIKITABLET requires generating Wikipedia article sections, which are typically multiple sentences and therefore more challenging. WIKITABLET is also much larger than all existing datasets.

To demonstrate the diversity of topics covered in WIKITABLET, we use either the "instance of" or "subclass of" relation from Wikidata as the category of the article.[5] We show the top 10 most frequent document categories in Table 2. Due to the criteria we use for filtering, only 1.05% of articles in WIKITABLET do not have these relations or Wikidata entries, and we omit these articles in the table. As the table demonstrates, more than 50% of the articles in WIKITABLET are not about people (i.e., the topic of WIKIBIO), within which the most frequent category covers only 4.61%.

3.3 Dataset Challenges

In this subsection, we highlight two challenges of WIKITABLET.

1. In contrast to work on evaluating commonsense knowledge in generation where reference texts are single sentences describing everyday scenes (Lin et al., 2020), WIKITABLET can serve as a testbed for evaluating models' abilities to use world knowledge for generating coherent long-form text.

2. Compared to other long-form data-to-text datasets such as ROTOWIRE where the input tables are box scores, the input tables in WIKITABLET are more diverse, including both numbers (e.g., economy and population data of an area throughout years), and short phrases. This makes WIKITABLET more challenging and applicable to various scenarios.

[5] When there are multiple values in these two relations, we pick the one that has the smallest number of words, as it often is the most generic phrase, suitable for representing the topic.

4 Methods

In this section, we describe details of models that we will benchmark on WIKITABLET.

Our base model is based on the transformer (Vaswani et al., 2017). To encode tables, we linearize the tables by using special tokens to separate cells and using feature embeddings to represent records in tables. For the title table in the first instance in Figure 1 the linearized table will be

$$\begin{aligned}&\langle\text{boc}\rangle_1 \text{Doc.}_1\ \text{title}_1 \langle\text{bov}\rangle_1\ \text{Wolfsbane}_1\ (\text{comics})_1\\ &\langle\text{boc}\rangle_2 \text{Sec.}_2\ \text{title}_2 \langle\text{bov}\rangle_2\ \text{Fictional}_2\ \text{character}_2\\ &\text{biography}_2 \langle\text{boc}\rangle_3 \cdots \langle\text{eoc}\rangle\end{aligned} \tag{1}$$

As shown in Eq. 1, we employ several techniques when encoding tables: (1) we use special tokens $\langle$boc$\rangle$ and $\langle$bov$\rangle$ to separate attributes and values, and $\langle$eoc$\rangle$ to indicate the end of a sequence; (2) we use subscript indices to indicate unique ID embeddings that are added to the embeddings for each record, which helps models align attributes with values; and (3) we restart the positional embeddings at each $\langle$boc$\rangle$, such that models will not use the ordering of the input records. In addition, we add a special embedding to each record to indicate if it is from the section table or the article/title table. In Wikidata, there could be multiple qualifiers attached to a record, in which case we replicate the record for each qualifier separately.

Similar linearization approaches have been used in prior work (Dhingra et al., 2019; Hwang et al., 2019; Herzig et al., 2020; Yin et al., 2020). With linearized tables, training and inference become similar to other sequence-to-sequence settings. We train our models with teacher-forcing and standard cross entropy loss unless otherwise specified.

4.1 Training Strategies

We experiment with three types of modifications to standard sequence-to-sequence training:

α-entmax. α-entmax (Peters et al., 2019) is a mapping from scores to a distribution that permits varying the level of sparsity in the distribution. This mapping function has been used in machine translation (Peters et al., 2019) and text generation (Martins et al., 2020). When using α-entmax in the decoder, we also replace the cross entropy loss with the α-entmax loss (Peters et al., 2019). Both

	Vocab.	Tokens	Examples	Avg. Len.	Record Types	Avg. Records	Domain
WikiTableText	-	185.0k	13.3k	13.9	3.0k	4.1	Wikipedia
WIKIBIO	400.0k	19.0M	728.0k	26.1	1.7k	19.7	Biography
ROTOWIRE	11.3k	1.6M	4.9k	337.1	39.0	628.0	Sports
MLB	38.9k	14.3M	26.3k	542.1	53.0	565.0	Sports
LogicNLG	122.0k	52.7k	37.0k	14.2	11.7k	13.5	Wikipedia
ToTTo	136.0k	1.3M	136.0k	17.4	41.8k	32.7	Wikipedia
DART	33.2k	717.1k	82.2k	21.6	-	-	Wikipedia+Restaurant
WIKITABLET	1.9M	169.0M	1.5M*	115.9	147.4k†	51.9	Wikipedia

Table 1: Statistics for several data-to-text datasets. WIKITABLET combines a large number of examples, moderate generation length (typically more than one sentence), and a large variety of record types. We omit record types and avg. records for DART as its input units are triple sets instead of table records. *887.7k unique Wikipedia articles. †Number of record types for each resource: 31.8k (Infobox), 1.7k (Wikidata), 115.6k (Hyperlinks), 17 (NER).

Category	Fraction (%)
human	45.62
film	4.61
single (music)	1.74
human settlement	1.53
album	1.41
sports season	1.26
television series	1.17
village	1.12
taxon	0.89

Table 2: Top 10 most frequent article categories and their corresponding proportions in WIKITABLET.

α-entmax and the α-entmax loss have a hyperparameter α. We follow Martins et al. (2020) and use $\alpha = 1.2$ as they found it to be the best value for reducing repetition in generation.

Copy Mechanism. Similar to prior work on data-to-text generation (Wiseman et al., 2017; Puduppully et al., 2019), we use pointer-generator network style copy attention (See et al., 2017) in the decoder.

Cyclic Loss. Cyclic losses have been shown to be effective in textual style transfer (Shetty et al., 2018; Pang and Gimpel, 2019) and neural machine translation (Cheng et al., 2016; He et al., 2016; Tu et al., 2017). Wiseman et al. (2017) also used this for data-to-text and found it helpful for generating long sequences. In this work, we experiment with adding the cyclic loss to our transformer models, where the backward model can be seen as an information extraction system. We expect that adding the cyclic loss should enable a data-to-text model to generate sentences that are more faithful to the conditioned tables. The cyclic loss is used during training only and does not affect the models during inference. More details are in the appendix.

4.2 Decoding Strategies

Massarelli et al. (2020) showed that the choice of decoding strategy can affect the faithfulness or repetitiveness of text generated by language models. We are also interested in these effects in the context of data-to-text generation, and therefore benchmark several decoding strategies on WIKITABLET. Our models use byte-pair encoding (BPE; Sennrich et al., 2016) and for all of the following strategies, we always set the minimum number of decoding steps to 100 as it improves most of the evaluation metrics, and the maximum number of decoding steps to 300.

Specifically, we benchmark (1) greedy decoding; (2) nucleus sampling (Holtzman et al., 2020) with threshold 0.9 as suggested by Holtzman et al. (2020); (3) beam search; and (4) beam search with n-gram blocking (Paulus et al., 2018) where we set the probabilities of repeated trigrams to be 0 during beam search. We set the beam size to be 5 by default. The appendix has more details about the decoding strategies.

5 Experiments

5.1 Setup

We experiment with two sizes of transformer models. One is "Base", where we use a 1-layer encoder and a 6-layer decoder, each of which has 512 hidden size and 4 attention heads. The other one is "Large", where we use a 1-layer encoder and a 12-layer decoder, each of which has 1024 hidden size and 8 attention heads. Models similar to the base configuration have shown strong performance on ROTOWIRE (Gong et al., 2019).[6] Due to limited

[6]When training the base model with entmax on WIKIBIO, it achieves BLEU-4 45.75 and ROUGE-4 39.39 on the test set using greedy decoding, which are comparable to the current state-of-the-art results of Liu et al. (2018).

	REP	BLEU	RL	MET	PAR-P	PAR-R	PAR-F1
References	1.2	100.0	100.0	100.0	100.0	59.2	72.9
Linearized article tables	8.0	2.2	14.7	9.3	100.0	16.3	25.6
Linearized section tables	1.0	1.9	27.9	15.5	100.0	20.9	33.4
Linearized tables	7.9	6.4	22.0	18.3	100.0	48.3	63.0
Linearized tables + references	7.6	36.5	61.3	56.5	99.9	100.0	100.0
Base models trained on the 500k training set (beam search)							
Base	33.0	15.6	36.9	20.3	66.3	28.8	37.7
Base + entmax	25.9	15.4	36.2	20.3	64.6	29.0	37.7
Base + copy	30.1	15.9	37.5	20.7	67.1	29.4	38.5
Base + copy + cyclic loss	28.0	15.7	37.5	20.8	67.5	29.7	38.9
Large models trained on the full training set (different decoding strategies)							
Large + greedy	26.8	18.9	38.5	23.5	60.4	33.1	40.4
Large + nucleus sampling	2.3	18.3	36.1	23.7	54.2	32.5	38.7
Large + beam search	18.8	19.5	39.9	23.9	65.8	34.3	42.8
Large + beam search + n-gram blocking	1.9	19.3	39.3	24.4	62.2	35.3	43.0

Table 3: Test set results for our models. When training the large models, we use the "copy + cyclic loss" setting as it gives the best performance for the base models for most of the metrics.

computational power, we parameterize our backward model as a transformer model with a 2-layer encoder and a 2-layer decoder.[7]

We use BPE with 30k merging operations. We randomly sample 500k instances from the training set and train base models on them when exploring different training strategies. We train a large model with the best setting (using the copy mechanism and cyclic loss) on the full training set. We train both models for 5 epochs. During training we perform early stopping on the development set using greedy decoding.

We report BLEU (Papineni et al., 2002), ROUGE-L (RL) (Lin, 2004), METEOR (MET) (Banerjee and Lavie, 2005), and PARENT (Dhingra et al., 2019), including precision (PAR-P), recall (PAR-R), and F1 (PAR-F1) scores. The first three metrics consider the similarities between generated texts and references, whereas PARENT also considers the similarity between the generation and the table. When using PARENT, we use all three tables, i.e., the section, article, and title tables.

As we are also interested in the repetitiveness of generated texts, we define a metric based on n-gram repetitions which we call "REP". REP computes the ratio of the number of repeated n-grams to the total number of n-grams within a text, so when REP has higher value, it indicates that the text has more repetitions. Here we consider n-grams that appear 3 or more times as repetitions and the n-grams we consider are from bigrams to 4-grams. When reporting REP scores for a dataset, we average the REP scores for each instance in the dataset. Similar metrics have been used in prior work (Holtzman et al., 2020; Welleck et al., 2020).

[7]We did not experiment with pretrained models because they typically use the entirety of Wikipedia, which would presumably overlap with our test set.

5.2 Results

In Table 3, we report the test results for both our base models and large models. We also report a set of baselines that are based on simply returning the linearized tables and their concatenations with the references. The linearized table baselines show how much information is already contained in the table, while the reference baselines show the upper bound performance for each metric.

In comparing training strategies, we find that using α-entmax improves REP significantly but not other metrics. Adding the cyclic loss or the copy mechanism helps improve performance for the PAR scores and REP, and combining both further improves these metrics.

When comparing decoding strategies, we find that both nucleus sampling and n-gram blocking are effective in reducing repetition. Nucleus sampling harms the PAR scores, especially PAR-P, but has less impact on the other metrics, indicating that it makes the model more likely to generate texts that are less relevant to the tables. Using beam search improves all metrics significantly when compared to greedy decoding, especially the PAR-P and REP scores. Adding n-gram blocking further reduces the REP score, pushing it to be even lower than that from nucleus sampling, but still retains the improvements in PAR scores from beam search. The best overall decoding strategy appears to be beam search with n-gram blocking.

	Grammar	Coherence	Faithfulness
Reference	4.0 (1.0)	4.1 (0.9)	3.8 (0.8)
Beam search	4.0 (1.0)	4.0 (1.0)	3.9 (1.0)
Nucleus sampling	4.0 (0.8)	4.1 (0.9)	3.9 (0.8)
n-gram blocking	4.2 (0.9)	4.2 (0.9)	3.9 (1.0)

Table 4: Average human ratings (standard deviations in parentheses) for grammaticality, coherence, and faithfulness to the input article table.

	Relevance	Support
Beam search	3.8 (1.1)	3.6 (1.2)
Nucleus sampling	3.7 (1.2)	3.8 (1.1)
n-gram blocking	3.9 (1.0)	3.8 (1.0)

Table 5: Average human ratings (standard deviations in parentheses) of relevance and support when comparing to the reference text.

6 Analysis

We now describe a manual evaluation and analyze some generated examples. All results in this section use the development set. We also conduct experiments on analyzing the effect of using the section data and the article data during training, finding that the benefits that they bring to the model performance are complementary. See the appendix for more details.

6.1 Human Evaluation

We conduct a human evaluation using generations from the large model on the development set. We choose texts shorter than 100 tokens and that cover particular topics as we found during pilot studies that annotators struggled with texts that were very long or about unfamiliar topics.[8]

We design two sets of questions. The first focuses on the text itself (i.e., grammaticality and coherence) and its faithfulness to the input article table. Since this set does not involve the reference, we can ask these questions about both generated texts and the reference texts themselves. The second set of questions evaluates the differences between the generations and the reference texts (i.e., relevance and support), allowing us to see if the generated text matches the human written section text. Specifically, relevance evaluates topical similarity between generations and references, and support evaluates whether the facts expressed in the generations are supported by or contradictory to those in the references. The full questions and numerical answer descriptions are in the appendix.

[8]We did not find the filtering to change the observed trends for the automatic metrics and provide the list of selected topics in the appendix.

We report results in Tables 4 and 5. The scores are on a 1-5 scale with 5 being the best. For the first set, we collect 480 annotations from 38 annotators. For the second set, we collect 360 annotations from 28 annotators. We also ensure that each system has the same number of annotations.[9]

It is interesting to note from Table 4 that human annotators are unable to differentiate the human written texts from the generations from our neural models. Since the Wikipedia section texts are parts of Wikipedia articles, showing the section texts in isolation can make them difficult to understand, potentially resulting in noisy annotations. As shown by the first instance in Table 6, the text uses the pronoun "he" without clarifying what the pronoun refers to. The paragraph is rated 3 for coherence, presumably due to this ambiguity. Also, Wikipedia texts are sometimes grammatically complex and annotators can mistake them for being ungrammatical, e.g., the second instance in Table 6.

On the other hand, the coherence errors in the generated texts are not always easy to spot. See, for example, the last two instances in Table 6, where the incoherence lies in the facts that (1) it is impossible to marry a person before the person is born, and (2) senior year takes place after junior year. These details are embedded in long contexts, which may be overlooked by annotators and lead to results favorable to these neural models.

To study the relationship between coherence and grammaticality, we compute Spearman's correlations between the human annotations for coherence and grammaticality after removing the ones with perfect scores for coherence. Table 7 shows the results. The correlations are much higher for references, beam search, and nucleus sampling than for n-gram blocking. This trend suggests that the imperfect coherence scores for the reference texts are likely because annotators find the texts to contain grammatical errors (or to possess grammatical complexity) which may prevent them from fully understanding the texts. However, n-gram blocking does not have this problem and thus achieves the best results for both coherence and grammaticality. We hypothesize that n-gram blocking is able to avoid the types of grammatical errors that

[9]We used Amazon Mechanical Turk. To ensure annotation quality, we only recruited annotators with master qualification. We collected one annotation for each instance (so that we can cover more instances) and paid 30 cents per annotation. The amount of wage per annotation is decided by (1) the amount of time each annotator spent on the task during our pilot study and (2) a target hourly wage of approximately $11.

Method	Text	G	C
Reference	He contested the parliamentary seat of Meriden at the 1987 general election, where he was defeated by the sitting Conservative MP Iain Mills by a margin of 16,820. He was then selected to fight the Conservative-held marginal seat of Birmingham Northfield ...	3	3
Reference	Boscawen married on 23 April 1700 in Henry VII's Chapel, Westminster Abbey, Charlotte Godfrey elder daughter and coheir of Colonel Charles Godfrey, master of the jewel office and his wife Arabella Churchill ...	3	4
Sampling	7th Marquess of Exeter married, firstly, Edith Csanady de Telegd (born **1 September 1935** in England; died 16 June 1956 in London), on **17 January 1934** ...	4	5
Blocking	... He averaged 10.9 rebounds and 3.0 assists per game **as a senior in 1987-88**. He was selected to the Sweet 16 of the NCAA Tournament **as a junior in 1988-89** ...	5	5

Table 6: Human annotation examples for grammaticality (G) and coherence (C). Due to space constraints, only parts of the texts are shown. We highlight texts that are incoherent.

	Ref.	Beam	Samp.	Block.
Spearman corr.	39.6	39.7	40.8	16.4
# annotations	67	80	76	67

Table 7: Spearman correlations between the human evaluation results for grammaticality and coherence. We omit annotations with perfect scores for coherence.

	1	2	3	4	5
Relevance	24.2	19.2	13.6	12.0	8.9
# annotations	10	48	65	124	113
Support	17.0	11.0	17.5	12.5	9.4
# annotations	13	47	68	135	97

Table 8: Averaged perplexities and the corresponding numbers of annotations for each option for the relevance and support questions (5 is the best option). We aggregate annotations for different decoding algorithms. We note that the perplexities are computed based on the reference texts using the large model.

prevent understanding because (1) unlike nucleus sampling, n-gram blocking does not rely on randomness to avoid repetition; (2) n-gram blocking does not suffer from repetitions like beam search.

We report results for the second set of questions in Table 5. The three evaluated systems show similar performance. To investigate the relationship between the degree of open-endedness of a WIKITABLET instance and its corresponding evaluation scores, we compute the averaged perplexities (based on our large models) for each option in Table 8. The most relevant generations are typically from more closed-ended or constrained instances.[10] Similarly for the support scores, more open-ended instances are distributed at score 3, which means that there is no fact supported by or contradictory to the shown tables. While the open-endedness of an instance usually depends on its topics (e.g., movie plots are open-ended), there are many cases where the models can benefit from better entity modeling, such as understanding what a particular entity type is capable of (e.g., see the last example in Sec. 6.3).

Recent work has also found conducting human evaluation for long-form generation to be challenging, for example in the context of question answering (Krishna et al., 2021) and story generation (Akoury et al., 2020). Our observations for data-to-text generation complement theirs and we hope that our dataset can inspire future research on human evaluation for long-form text generation.

[10] Li and Hovy (2015) use entropy as a proxy to quantify complexity of tasks. In this work, we use perplexity to measure how open-ended the instances are.

percentile	train perp.	dev perp.
10	2.3	2.5
20	3.1	3.6
30	4.0	4.7
40	4.9	6.1
50	6.1	7.6
60	7.6	9.8
70	9.8	12.9
80	13.3	18.2
90	20.6	29.3
100	278.8	129.4

Table 9: Percentiles of perplexities for training and development splits of our dataset. We use the large model.

6.2 Distribution of Perplexity

To determine the fraction of WIKITABLET that can be seen as constrained, we report the percentiles of perplexities for training and development splits in Table 9. From Table 8, it can be observed that instances with perplexities around 9.0 generally lead to model generations that are closely relevant to the reference texts and mostly supported by the input tables, and therefore are likely to be the constrained instances. From Table 9, we see that at least half of our dataset has perplexities lower than 9.0, so we conjecture that half of our dataset consists of constrained instances.

6.3 Generation Examples

Table 10 shows generation examples for nucleus sampling and beam search with n-gram blocking.

Method	Generated Text
Nucleus sampling (threshold 0.9)	In a two-issue mini-series titled "Messiah Complex", portrayed by Maisie Williams, Father Craig is ordered to rescue the world from Reavers and run out of the X-Force causing the X-Force to permanently withdraw. The original humans were replaced by slots and Muir Island being pulled away. They also possessed the entire X-Force after issue #9 in which the X-Force appeared. Claremont and Sinclair later reprised their parts in "The Donderful World of Wolverine" with Wolfsbane as one of the new X-Men and her group taken back to their reservation for unknown reasons.
n-gram blocking	In the "Messiah Complex" storyline, Wolfsbane is a member of the X-Men, a group of mutants recruited by Father Craig to fight the Reavers in Muir Island. Father Craig is killed in the battle with Reavers, but Father Craig manages to rescue the mutant from the Rain of Gesham and retrieve the mutant. She is later revealed to have been killed by Rain in a battle with Father Craig. In the aftermath, Father Craig tells Father Craig that she is the only mutant to be killed by a mutant.
Nucleus sampling (threshold 0.9)	In May 2013, McCoy allegedly assault a Spoiler business official. The incident motivated him to take a business trip to Harrisburg, Pennsylvania and sign a letter from the Buffalo Bills paying him over $50,000 as well as verbally prohibiting legal kickoff work. In September 2014, McCoy shared the issue with some former players of the first two film franchises. McCoy played a leading role in the 2012 Marvel Cinematic Universe film "Erab of the Press", and was named one of the film's biggest stars on Twitter. On September 9, 2014, McCoy filed a lawsuit against PYT to change its Twitter name to LeSean McCoy Productions.
n-gram blocking	On September 9, 2014, McCoy was arrested in Harrisburg, Pennsylvania on suspicion of assault. He was charged with assault and battery. In May 2013, he was fined over $50,000 by the Buffalo Bills. In September 2014, he was suspended for two games by the PYT for violating the Marvel Cinematic Universe. He was released by the Bills in October of the same year. He was cleared of all charges on Twitter, and was banned from playing in the 2014 Pro Bowl due to his Twitter account.

Table 10: Generation examples from the large model. The first example corresponds to the first instance in Figure 1. The complete set of generations is in the appendix.

We observe very different trends between the two instances in Figure 1. For the first instance about the X-Men, although both generations look fluent, their stories differ dramatically. The generated text for nucleus sampling describes a story that starts by saying Father Craig rescues the world from Reavers and ends with Wolfsbane joining as one of the new X-Men. On the other hand, n-gram blocking generates a story where Wolfsbane already is a member of X-Men, and the story says Father Craig fought and was killed by the Reavers, but manages to rescue the mutant. For the less open-ended instances (e.g., the second instance in Figure 1), different decoding strategies mostly generate similar details (see the appendix for generations).

Despite having different details, these generations appear to try to fit in as many entities from the tables as possible, in contrast to beam search (shown in the appendix) which mostly degenerates into repetition for more open-ended instances. This explains our previous observation that n-gram blocking helps with the PAR-R score.

Even though the generations are of good quality for most instances, their implausibility becomes more apparent when readers have enough background knowledge to understand the involved entities. For example, the second instance in Table 10 comes from the Wikipedia page "LeSean McCoy" (a football player) under the sections "Personal life" and "Controversies" (details in the appendix). The generation from nucleus sampling is implausible/nonsensical in some places ("assault a Spoiler business official") and factually incorrect elsewhere (McCoy did not play a leading role in any film, and "Erab of the Press" is not an actual film). The fourth generation is implausible because a player is unlikely to be suspended for "violating the Marvel Cinematic Universe", and it is unlikely for a person to be cleared of all charges on Twitter. Our models have limited access to knowledge about entities, e.g., the capabilities of a social media company like Twitter. Future research may incorporate extra resources, make use of pretrained models, or incorporate factuality modules to solve these problems.

7 Conclusion

We created WIKITABLET, a dataset that contains Wikipedia article sections and their corresponding tabular data and various metadata. WIKITABLET contains millions of instances covering a broad range of topics and kinds of generation tasks. Our manual evaluation showed that humans are unable to differentiate the references and model generations, and n-gram blocking performs the best on grammaticality and coherence. However, qualitative analysis showed that our models sometimes struggle with coherence and factuality, suggesting several directions for future work.

Acknowledgments

This work was supported in part by a Google Fellowship to M. Chen.

Impact Statement

We highlight a few limitations as follows: (1) Wikipedia texts are generally written in objective tones, but some of the texts may contain controversial content that even the community contributors do not agree upon; (2) models trained on our dataset may generate deceitful texts that are unfaithful to what actually happened to particular entities; (3) though the instances in WIKITABLET cover various topics, the writing style is almost always the same. Future work may explore more diverse writing styles.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. In *AAAI Conference on Artificial Intelligence*.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 128–135, New York, NY, USA. Association for Computing Machinery.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1965–1974, Berlin, Germany. Association for Computational Linguistics.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG

challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

Li Gong, Josep Crego, and Jean Senellart. 2019. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong. Association for Computational Linguistics.

Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. 2019. Neural text generation from rich semantic representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2259–2266, Minneapolis, Minnesota. Association for Computational Linguistics.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2020. WikiAsp: A dataset for multi-domain aspect-based summarization. *ArXiv*, abs/2011.07832.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 820–828. Curran Associates, Inc.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Wonseok Hwang, Jinyeung Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on WikiSQL with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Jun Seok Kang, Robert Logan, Zewei Chu, Yang Chen, Dheeru Dua, Kevin Gimpel, Sameer Singh, and Niranjan Balasubramanian. 2019. PoMo: Generating entity-specific post-modifiers in context. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 826–838, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Karen Kukich. 1983. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Jiwei Li and Eduard Hovy. 2015. The NLP engine: A universal turing machine for NLP.

Percy Liang, Michael Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Peter J. Liu*, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4881–4888. AAAI Press.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2020. Sparse text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4252–4273, Online. Association for Computational Linguistics.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

Neil McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 217–225, Suntec, Singapore. Association for Computational Linguistics.

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

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Eyal Orbach and Yoav Goldberg. 2020. Facts2Story: Controlling text generation by key facts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2329–2345, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 138–147, Hong Kong. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Philadelphia, Pennsylvania, USA. 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.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. Association for Computational Linguistics.

Vladimir Propp. 1968. *Morphology of the Folktale*, volume 9. University of Texas Press.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4nt: author attribute anonymity by adversarial training of neural machine translation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1633–1650.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Pavlos Vougiouklis, Hady ElSahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon S. Hare, and Elena Simperl. 2017. Neural Wikipedian: Generating textual summaries from knowledge base triples. *CoRR*, abs/1711.00155.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, Tilburg University, The Netherlands. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Markus Zopf. 2018. Auto-hMDS: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

A Dataset Construction

When collecting data, we consider five resources: Wikidata tables, infoboxes in Wikipedia pages, hyperlinks in the passage, named entities in the passage obtained from named entity recognition (NER), and Wikipedia article structure. For each article in Wikipedia, we use the same infobox and Wikidata table for all sections. These tables can serve as background knowledge for the article. For each section in the article, we create a second table corresponding to section-specific data, i.e., section data. The section data contains records constructed from hyperlinks and entities identified by a named entity recognizer. Section data contributes around 25% of the records in WIKITABLET.

We filter out several entity types related to numbers[11] as the specific meanings of these numbers in the section of interest are difficult to recover from the information in the tables. After filtering, we use the identified entities as the values and the entity types as the attributes. This contributes roughly 12% of the records in our final dataset.

We also create records from hyperlinks in the section of interest. We first expand the hyperlinks available for each section with hyperlinks available in the parent categories. We first group hyperlinks across all Wikipedia articles with those same categories, and then we perform string matching between these hyperlinks and the text in the section. If there are exact matches, we will include those hyperlinks as part of the hyperlinks in this section.

Details for constructing a record with attribute a and value v for a hyperlink with surface text t and hyperlinked article ℓ are as follows. To set a, we use the value of the "instance of" or "subclass of" tuple in the Wikidata table for ℓ. If ℓ does not have a Wikidata table or no appropriate tuple, we consider the parent categories of ℓ as candidates for a. If there are multiple candidates for a, we first embed these candidates and a using GloVe (Pennington et al., 2014) embeddings and then choose the one that maximizes cosine similarity between the document titles or section titles and the candidates for a. For the value v of the tuple, we use the document title of ℓ rather than the actual surface text t to avoid giving away too much information in the reference text. The records formed by hyperlinks contribute approximately 13% of the records in WIKITABLET.

[11] List of filtered entity types: PERCENT, TIME, QUANTITY, ORDINAL, CARDINAL.

We shuffle the ordering of the records from NER and the hyperlinks to prevent models from relying on the ordering of records in the reference text.

The records from the section data can be seen as section-specific information that can make the task more solvable. Complementary to the article data, we create a title table that provides information about the position in which the section is situated, which includes the article title and the section titles for the target section. As the initial sections in Wikipedia articles do not have section titles, we use the section title "Introduction" for these.[12]

As the records in our data tables come from different resources, we perform extra filtering to remove duplicates in the records. In particular, we give Wikidata the highest priority as it is a human-annotated well-structured data resource (infoboxes are human-annotated but not well-structured due to the way they are stored on Wikipedia) and the entities from NER the lowest priority as they are automatically constructed. That is, when we identify duplicates across different resources, we will keep the records from the higher priority resource and drop those from the lower one. More specifically, the duplicates between Wikidata records and infoboxes are determined by whether there are duplicate values or duplicate attributes: for hyperlinks and infoboxes or Wikidata, they are judged by duplicate values; for NER and hyperlinks, they are based on whether there is any token overlapping between values.

After table collection, we have the following criteria for filtering out the texts: (1) we limit the text length to be between 50 and 1000 word tokens; (2) to ensure that there is sufficient information in the table, we only keep data-text pairs that contain more than 2 records per sentence and more than 15 records per 100 tokens from Wikidata and infoboxes; (3) to avoid texts such as lists of hyperlinks, we filter out texts where more than 50% of their word tokens are from hyperlink texts.

B Human Evaluation

The selected topics for human evaluations are: human (excluding the introduction and biography section), film, single (song), song, album, television series. When evaluating grammaticality and coherence, only the generated text is shown to annotators.

[12] Among millions of section titles in Wikipedia, there are only 4672 sections, including nested sections, that are called "Introduction". Therefore, we believe this process will not introduce much noise into the dataset.

1 = it is completely ungrammatical, as it is impossible to understand the text.
2 = it has many grammatical errors, and these errors make the text very difficult to understand.
3 = it has grammatical errors, and some of them make part of the text difficult to understand.
4 = it has some grammatical errors, but they are minor errors that do not affect reading.
5 = it is completely grammatical, as it does not have any grammatical errors.

Table 11: Rating explanations for grammaticality.

1 = it is completely incoherent, as it is impossible to piece together information in the text.
2 = it is incoherent in most places. You can only understand part of the story.
3 = it is incoherent in many places, but if you spend time reading it, you still can understand the whole story.
4 = it is mostly coherent. Although the text is incoherent in some places, it does not affect reading.
5 = it is completely coherent.

Table 12: Rating explanations for coherence.

The question for grammaticality is "On a scale of 1-5, how much do you think the text is grammatical? (Note: repetitions are grammatical errors.)" (option explanations are shown in Table 11), and the question for coherence is "On a scale of 1-5, how much do you think the text is coherent? (Coherence: Does the text make sense internally, avoid self-contradiction, and use a logical ordering of information?)" (rating explanations are in Table 12).

When evaluating faithfulness, we show annotators the article data and the generation. The question is "On a scale of 1-5, how much do you think the text is supported by the facts in the following table?" (rating explanations are in Table 13).

When evaluating coherence and relevance, annotators were shown the reference text and the generation, as well as the Wikipedia article title and section titles for ease of understanding the texts. Annotators were asked two questions, with one being "On a scale of 1-5, how much do you think the text is relevant to the reference" (Table 14), and the other being "On a scale of 1-5, how much do you think the text is supported by the facts in the reference?" (Table 15).

C Effect of α-entmax

In this section, we disentangle the effect of α-entmax and that of α-entmax loss. We note that (1) when not using the α-entmax loss, we use standard cross entropy loss (e.g., in the case of "base+ent." we maximize the log probabilities generated by α-entmax); (2) when combining α-entmax and copy mechanism, we aggregate the probabilities generated by α-entmax and those from softmax. This is because we use the first attention head in the transformer decoder as the copy attention, following the implementation in OpenNMT (Klein et al., 2017). While it is feasible to combine the α-entmax and α-entmax loss with the copy mechanism if we use the sparse transformer (Correia et al., 2019), we leave this for future study. We report the results in Table 16. It is interesting to see that when using greedy decoding, "ent. + ent. loss" outperforms the baseline model by a significant margin on all the metrics, however the improvement disappears (except for repetition) after we switch to use beam search as the decoding strategy. This is likely because α-entmax promotes sparsity in the generated probabilities, making beam search decoding unnecessary. Removing the α-entmax loss hurts the performance, but its gains become larger in switching to beam search decoding. Adding copy mechanism improves the performance, leading to comparable performance to the baseline model. Although "base+ent.+copy" still underperforms "base+copy" when using beam search, we believe that combining α-entmax and α-entmax loss with the copy mechanism is promising as (1) α-entmax is not used in our large models and the initial results have shown that α-entmax and the copy mechanism are complementary, so it may further improve our current best performance; (2) α-entmax already shows the best performance when using greedy decoding, which has speed and optimization advantages compared to the beam search based decoding strategies especially considering the long-form characteristic

1 = it is completely contradictory to what is described in the table.
2 = it has some facts contradictory to what is described in the table.
3 = it is not supported by the table, and it does not contradict the table.
4 = some of the text is supported by the facts in the table, and the rest of it does not contradict the facts in the table.
5 = it is completely supported by the table.

Table 13: Rating explanations for faithfulness.

1 = the text is completely irrelevant to the reference.
2 = most of the text is irrelevant to the reference.
3 = some of the text is relevant to the reference.
4 = most of the text is relevant to the reference.
5 = the text is talking about the same thing as the reference.

Table 14: Rating explanations for relevance.

1 = it has quite a few facts contradictory to what is described in the reference.
2 = it has some facts contradictory to what is described in the reference.
3 = it is not supported by the reference, and it does not contradict the reference.
4 = some of the text is supported by the facts in the reference, and the rest of it does not contradict the reference.
5 = it is completely supported by the reference.

Table 15: Rating explanations for supportedness.

	REP	BLEU	PAR-P	PAR-R	PAR-F1
Greedy decoding					
base	38.1	14.7	61.6	27.7	35.8
+ ent. + ent. loss	36.0	16.2	62.2	28.9	37.0
+ ent.	44.5	13.9	63.5	25.5	33.9
+ ent. + copy	43.7	14.8	64.2	26.6	35.2
+ copy	37.8	15.8	61.3	28.3	36.3
Beam search (beam size 5)					
base	33.0	15.6	66.3	28.8	37.7
+ ent. + ent. loss	25.9	15.4	64.6	29.0	37.7
+ ent.	34.7	13.8	67.2	26.6	35.8
+ ent. + copy	34.1	15.0	69.4	28.1	37.6
+ copy	30.1	15.9	67.1	29.4	38.5

Table 16: Effect of using α-entmax and α-entmax loss. When not using the α-entmax loss, we use standard cross entropy loss.

of WIKITABLET.

D Details of Cyclic Loss

In this section, we will denote the linearized table where the values are replaced with a special $\langle$mask$\rangle$ token by $u_1, \cdots, u_n$, and denote the reference text by $x_1, \cdots, x_m$. Formally, the training loss is

$$\sum_{w \in S} -\log p(w|u_1, \cdots, u_n, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_m) \quad (2)$$

where S represents the set of masked tokens, and $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m$ is the sequence of token-level probabilities predicted by the forward model (in our experiments, these could either come from the softmax function, or the α-entmax function). Specifically, we multiply the backward transformer's input embedding matrix by the $\boldsymbol{v}$ probability vectors to obtain the input representations to the first encoder layer. We find that it is helpful to add a "reference loss" while training with the cyclic loss, defined as

$$\sum_{w \in S} -\log p(w|u_1, \cdots, u_n, x_1, \cdots, x_m) \quad (3)$$

This loss does not contain the generation model in it explicitly, but it does lead to an improved backward model by training it with clean inputs. Improving the backward model then increases the benefits of the cyclic loss.[13]

	REP	BLEU	PAR-P	PAR-R	PAR-F1
Both	38.1	14.7	61.6	27.7	35.8
Art. only	60.9	8.4	55.2	14.7	20.8
Sec. only	39.0	13.4	56.1	24.3	31.7

Table 17: Effect of dropping section or article data from the input (using the "base" setting).

	REP	BLEU	PAR-P	PAR-R	PAR-F1
None	37.8	15.8	61.3	28.3	36.3
Both	35.9	15.8	62.0	28.5	36.7
Art. only	37.2	15.8	61.7	28.1	36.2
Sec. only	34.8	15.9	61.9	28.2	36.2

Table 18: Effect of dropping section or article data when using cyclic training. The results are based on the "base + copy" and "base + copy + cyclic loss" settings.

E Effect of Article Data and Section Data

We report results in Table 17 for the models that are trained with partial data input, where art. only and sec. only indicate that we use only article data or section data, respectively. We always use title data. Section data contributes the most to the BLEU and PAR scores, but using section data and article data together is the best setting.

We also investigate the effect of partial data input for the cyclic loss in Table 18, where "None" is the model that is not trained with the cyclic loss. We note that in this setting, we still use both data resources as the input to the forward model, but vary the input data and the gold standard for the backward model. Although using only section data gives the best REP score and improves the PAR-P score, it does not help the model in other metrics. Combining the article data with the section data gives significant improvements to the PAR-F1 score compared to section data alone.

Both experiments show that there are interactions between these two data resources that can help models to learn better from both kinds.

F Generation Examples

We show the full set of generations in Table 19. The part of input data and reference text for Table 19 is shown in Figure 2.

[13] We experimented with initializing the backward model with pretrained checkpoints, but did not find it helpful.

… In September 2014, McCoy received criticism for leaving a $0.20 tip at the PYT burger restaurant in Philadelphia. McCoy claimed the low tip was due to "bad service." The owner responded on September 9, 2014, saying that McCoy had been "verbally abusive" and made "derogatory comments about women." In April 2019, McCoy tweeted an important plot point in the film *Avengers: Endgame* shortly after watching the film, angering his followers and Marvel fans to the point that some blocked him and even called for the termination of his contract with the Buffalo Bills or for him to get injured. McCoy later called this the "worst decision" of his life in an interview with *Good Morning Football*.

Section Data

Attribute	**Value**
information	Spoiler (media)
violent crime	assault
ORG	the Buffalo Bills
DATE	September 9, 2014
MONEY	0.200
social networking service	Twitter
film series	Marvel Cinematic Universe

Article Data

Attribute	**Value**
date of birth	1988-07-12
instance of	human
place of birth	Harrisburg
occupation	American football player

Title Data

Document title	LeSean McCoy
Section $title_1$	Personal life
Section $title_2$	Controversies

Figure 2: An example from WIKITABLET. Only parts of the tables and reference are shown due to space constraints. Underlined texts are hyperlinks. Records with the attributes "DATE", "ORG', "MONEY" are from NER. Subscripts for section titles indicate the ordering of nesting, where smaller numbers indicate higher level sections.

Method	Generated Text
sampling 0.9	In a two-issue mini-series titled "Messiah Complex", portrayed by Maisie Williams, Father Craig is ordered to rescue the world from Reavers and run out of the X-Force causing the X-Force to permanently withdraw. The original humans were replaced by slots and Muir Island being pulled away. They also possessed the entire X-Force after issue #9 in which the X-Force appeared. Claremont and Sinclair later reprised their parts in "The Donderful World of Wolverine" with Wolfsbane as one of the new X-Men and her group taken back to their reservation for unknown reasons.
beam search (size=5)	In the "Messiah Complex" storyline, Wolfsbane is a member of the X-Men. She is a member of the X-Factor Investigations team and a member of the X-Force/
blocking (size=5)	In the "Messiah Complex" storyline, Wolfsbane is a member of the X-Men, a group of mutants recruited by Father Craig to fight the Reavers in Muir Island. Father Craig is killed in the battle with Reavers, but Father Craig manages to rescue the mutant from the Rain of Gesham and retrieve the mutant. She is later revealed to have been killed by Rain in a battle with Father Craig. In the aftermath, Father Craig tells Father Craig that she is the only mutant to be killed by a mutant.
sampling 0.9	Journey to the Center of the Earth is a 1959 American science fiction film based on Jules Verne's 1959 novel of the same name. The film, directed by Henry Levin, stars James Mason as Professor Otto Lidenbrock, a scientist who gets entangled in the power of the archenemy as he tries to impress his closest allies and compromises them with the future. The film marks the first screen appearance of future CinemaScope of 20th Century Fox lead singer Peter Ronson. "The 20th Century Fox" earned critical acclaim and was a box-office success, grossing $10 million worldwide against a production budget of $3.44 million .
beam search (size=5)	Journey to the Center of the Earth is a 1959 American science fiction film directed by Henry Levin and starring James Mason, Pat Boone, and Arlene Dahl. The screenplay by Charles Brackett and Walter Reisch is based on Jules Verne's novel "A Journey to the Center of the Earth" (1959). The film was produced by Charles Brackett and Walter Reisch and distributed by Twentieth Century-Fox Film Corporation. The film was nominated for an Academy Award for Best Art Direction. It was also nominated for an Academy Award for Best Art Direction and a Golden Globe Award for Best Visual Effects. It was also nominated for a Golden Globe Award for Best Motion Picture - Musical or Comedy, but lost to 20th Century Fox's "CinemaScope".
blocking (size=5)	Journey to the Center of the Earth is a 1959 American science fiction film directed by Henry Levin and starring James Mason, Pat Boone, and Arlene Dahl. The screenplay by Charles Brackett and Walter Reisch is based on Jules Verne's novel of the same name. The film was produced by CinemaScope and 20th Century Fox and distributed by Twentieth Century-Fox Film Corporation and Joseph M. Schenck Enterprises, Inc. It was nominated for an Academy Award for Best Art Direction.
sampling 0.9	In May 2013, McCoy allegedly assault a Spoiler business official. The incident motivated him to take a business trip to Harrisburg, Pennsylvania and sign a letter from the Buffalo Bills paying him over $50,000 as well as verbally prohibiting legal kickoff work. In September 2014, McCoy shared the issue with some former players of the first two film franchises. McCoy played a leading role in the 2012 Marvel Cinematic Universe film "Erab of the Press", and was named one of the film's biggest stars on Twitter. On September 9, 2014, McCoy filed a lawsuit against PYT to change its Twitter name to LeSean McCoy Productions.
beam search (size=5)	On September 9, 2014, McCoy was arrested in Harrisburg, Pennsylvania on suspicion of assault. He was charged with assault and battery. In May 2013, McCoy was fined over $50,000 by the Buffalo Bills. In September 2014, McCoy was arrested in Harrisburg, Pennsylvania on suspicion of assaulting a woman who had been raped by McCoy. McCoy was charged with assault and possession of marijuana. McCoy was suspended from the PYT for the first two games of the Marvel Cinematic Universe.
blocking (size=5)	On September 9, 2014, McCoy was arrested in Harrisburg, Pennsylvania on suspicion of assault. He was charged with assault and battery. In May 2013, he was fined over $50,000 by the Buffalo Bills. In September 2014, he was suspended for two games by the PYT for violating the Marvel Cinematic Universe. He was released by the Bills in October of the same year. He was cleared of all charges on Twitter, and was banned from playing in the 2014 Pro Bowl due to his Twitter account.

Table 19: Top: generation examples for the first instance in Figure 1. Middle: generation examples for the second instance in Figure 1. Bottom: generation examples that correspond to the instance in Figure 2.

G Details of Decoding Strategies

Nucleus Sampling. Generating long sequences usually suffers from repetitions. Nucleus sampling (Holtzman et al., 2020) aims to reduce the repetitions in generations by sampling from truncated probability distributions. The truncation is based on whether the cumulative probability is above a threshold. We set the threshold to be 0.9 as suggested in Holtzman et al. (2020).

Beam Search with n-gram Blocking. Paulus et al. (2018) found it effective to reduce the repetitions during beam search by "blocking" n-grams that have been generated in previous decoding steps. We follow their approach by using trigram blocking and setting the probability of repeated trigrams to be 0 during beam search.