Framework for Recasting Table-to-Text Generation Data for Tabular Inference

Aashna Jena[1]*, **Vivek Gupta**[2]*†, **Manish Shrivastava**[1], **Julian Martin Eisenschlos**[3]
[1]LTRC, IIIT Hyderabad; [2]University of Utah; [3]Google Research, Zurich;
aashna.jena@research.iiit.ac.in; vgupta@cs.utah.edu;
m.shrivastava@iiit.ac.in; eisenjulian@google.com

Abstract

Prior work on constructing challenging tabular inference data centered primarily on human annotation or automatic synthetic generation. Both techniques have their own set of issues. Human annotation, despite its diversity and superior reasoning, struggles from scaling concerns. Synthetic data, on the other hand, despite its scalability, suffers from lack of linguistic and reasoning diversity. In this paper, we address both of these concerns by presenting a recasting approach that semi-automatically generates tabular NLI instances. We transform the table2text dataset ToTTo (Parikh et al., 2020) into a tabular NLI dataset using our proposed framework. We demonstrate the use of our recasted data as an evaluation benchmark as well as augmentation data to improve performance on TabFact (Chen et al., 2020b). Furthermore, we test the effectiveness of models trained on our data on the TabFact benchmark in the zero-shot scenario.

1 Introduction

Given a premise, Natural Language Inference (NLI) is the task of classifying a hypothesis as entailed (true), refuted (false) or neutral (cannot be determined from given premise). Several large scale datasets such as SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), and SQuAD (Rajpurkar et al., 2016) explore NLI with unstructured textual data as the premise. In recent times, efforts towards including structured data like tables as the premise for NLI began with TabFact (Chen et al., 2020b), InfoTabS (Gupta et al., 2020) and shared tasks like SemEval 2021 Task 9 (Wang et al., 2021a) and FEVEROUS (Aly et al., 2021). Tabular data differs from unstructured text due to its ability to capture information and relationships through structure instead of language.

While textual entailment is well studied, tabular entailment models require an immense amount of data to learn structural correlations. Recent work makes use of simple data augmentation techniques, context-free-grammar and templates to generate synthetic training data for NLP tasks in general (Alberti et al., 2019; Lewis et al., 2019; Wu et al., 2016; Leonandya et al., 2019), and Tabular NLI in specific (Geva et al., 2020; Eisenschlos et al., 2020). Even though synthetic data is highly scalable, it lacks the linguistic diversity, both structural and lexical, that human written data possesses. Since tabular entailment tasks require complex reasoning over data (e.g. ranking trends, aggregation, counting, etc), synthetic data limits the types and complexity of reasoning to only those mentioned in templates.

Human-annotated data, on the other hand, is fluent and diverse, but extremely hard to scale owing to its costly and time-consuming nature. Recent work shows that many human-annotated datasets for NLI contain annotation biases or artifacts (Gururangan et al., 2018; Geva et al., 2019). This enables NLI models to learn spurious patterns, meaning that they are able to classify correctly even with noisy, incomplete or fully absent premise (Poliak et al., 2018b). One reason behind this is the skewed distribution of words in the hypotheses. Negations like "no" and "never" are easily correlated with contradictions (Niven and Kao, 2019). Additionally, recent work also highlights the bias introduced by annotators through over-using certain types of keys or table cells (Gupta et al., 2021).

Can we generate data that is as scalable as synthetic data and yet contains human-like fluency and linguistic diversity? QA2D (Demszky et al., 2018) and SciTail (Khot et al., 2018) recast question answering data for entailment tasks on unstructured text. In this case, recasting refers to changing data intended for one task into data intended for

*Equal Contribution †Corresponding Author

other distinct task. Taking inspiration from this, we propose a framework to semi-automatically generate large-scale tabular NLI data by recasting existing table-to-text generation datasets (Parikh et al., 2020; Nan et al., 2021; Yoran et al., 2021; Chen et al., 2021, and others). This recasting framework is a middle route that permits us to leverage the advantages of both synthetic and human-annotated data generation methods. Recasting available data allows us to cut annotation time and cost. Since the source data is not originally intended for NLI, it eliminates the task-specific biases introduced by annotators. Recasting also allows us to create data that is not purely synthetic, owing to the human involvement in the source dataset creation process.

Our generated data could be used for both evaluation and augmentation purposes for tabular inference tasks. Models pre-trained on our generated data show an improvement of 17 points from the TabFact baseline (Chen et al., 2020b) and 1.2 points from Eisenschlos et al. (2020), a synthetic data augmentation baseline. Additionally, we report a zero-shot accuracy of > 80% on TabFact's simple test set, which is 1.4 percent higher than the supervised baseline accuracy reported by Chen et al. (2020b) on its simple test set. We analyse the performance of various models on different splits of our data as evaluation set. Our main contributions are the following:

1. We propose a semi-automatic framework to generate tabular NLI data through recasting table-to-text generation data.
2. We build a large-scale, diversified, human-alike, and bias-free tabular NLI dataset using an existing table to text generation dataset, ToTTo (Parikh et al., 2020).
3. We show the effective of our generated data for augmented training for Tabular NLI task on TabFact (Chen et al., 2020b).
4. We demonstrate that our generated data could be used as an challenging evaluation set for tabular inference task.

The dataset and associated scripts are available at https://totto-to-tnli.github.io.

2 Framework

In this section, we will describe our semi-automatic framework to recast existing table-to-text generation data for the task of Table NLI. By recasting, we mean to transform data intended for one task into a form that meets the criteria of another distinct task.

Prerequisites Table-to-text generation datasets provide us with a premise i.e. a table and a description generated from it. This description (referred to as the **Base Entailment** here on), since derived from the table, entails it. The constraints for creating contradictions are fairly loose. Falsifying any one part of the Base Entailment that stems from the table creates contradictions. However, the constraints for creating entailments are very tight, since every part of the perturbed statement must be true for the overall statement to be an entailment. This warrants that we must find **all relevant entities** (i.e. entities stemming from the table) present in the Base Entailment. Only then can we decide what to replace, and deterministically call the resultant statement an entailment.

As shown in Table 1, alignments between a table and a Base Entailment aren't always explicit. In the example *"Party A won the **most** seats"*, we must know the alignment between *most* and the highest value of seats. While we can employ automatic matching techniques between the Base Entailment and the table to capture relevant entities, we cannot be sure of finding **all** of them unless explicitly given. For this reason, we require the source dataset to provide us with the following as prerequisites: (a) a table i.e. Premise, (b) a description i.e. Base Entailment and (c) explicit alignment information between the table and the Base Entailment for each relevant entity.

Once we have established the prerequisites, new NLI instances can be generated by perturbing existing data in following two ways: (a) Perturbing the hypothesis and (b) Perturbing the table i.e. premise.

2.1 Perturbing the Hypothesis

We make changes to the hypothesis i.e. the Base Entailment by replacing entities stemming from the table (referred to as **relevant entities** here on) with other potential candidates. We assume that the tables are oriented vertically, meaning that the first row contains headers, and each column has entities of the same type. A *potential candidate* for a relevant entity coming from table cell having coordinates $[rowX, columnY]$ can be any other entity from the same column (Y).

Creating Entailments (E) To create entailments, we replace *all* the relevant entities in the given

Original Table (OG)

Party	Votes(thou)	Seats
Party A	650	120
Party B	570	89
Party C	final count TBA	89
Total	1235	298

Counterfactual Table (CF - after swapping cells)

Party	Votes(thou)	Seats
~~Party A~~ **Party B**	650	120
~~Party B~~ **Party A**	570	89
Party C	final count TBA	89
Total	1235	298

$Annotation_{OG}$	Party A won 120 out of 298 seats.	Party A won the most seats.
$Entailment_{OG}$	Party B won 89 out of 298 seats.	Party B won the second most seats.
$Paraphrase_{OG}$	Out of a total of 298 available seats, Party B won 89.	Party B secured the second largest number of seats.
$Contradiction_{OG}$	~~Party A~~ Party B won 120 out of 298 seats.	~~Party A~~ Party B won the most seats. Party A won the ~~most~~ least seats.

*We swap **Party A** and **Party B** to create a counterfactual table. The contradictions mentioned above become the new base annotations ($Annotation_{CT}$)*

$Annotation_{CF}$	Party B won 120 out of 298 seats.	Party B won the most seats.
$Entailment_{CF}$	Party A won 89 out of 298 seats.	Party A won the second most seats.
$Paraphrase_{CF}$	89 of the 298 available seats were secured by Party A	Party A won next to the maximum number of seats.
$Contradiction_{CF}$	~~Party B~~ Party A won 120 out of 298 seats.	~~Party B~~ Party A won the most seats. Party B won the ~~most~~ least seats.

Table 1: Pipeline for generating recasted NLI data. We first create entailments and contradictions from the given base annotation. We then create a counterfactual table taking a contradiction to be the new base annotation. $\text{subscript}_{\text{OG}}$ represents the "Original" table and $\text{subscript}_{\text{CF}}$ represents the "Counterfactual" table. Note that in this example, $\text{Contradiction}_{\text{CF}}$ is an $\text{Entailment}_{\text{OG}}$ to the original table, but $\text{Entailment}_{\text{CF}}$ is a $\text{Contradiction}_{\text{OG}}$ to it.

Base Entailment with potential candidates. Two or more relevant entities coming from table cells in the same row, say $[rowX, columnA]$ and $[rowX, columnB]$ must be replaced with potential candidates from column A and B respectively, such that their row coordinate is equal i.e. $[rowY, columnA]$ and $[rowY, columnB]$ (refer Table 1). Entities coming from "aggregate rows" (such as the *Total* row in Table 1) or "headers" must be left intact.

Creating Contradictions (C) To create contradictions, we replace *one or more* of the relevant entities in the Base Entailment with other potential candidates. We observe that the resultant statement may accidentally be an entailment. In Table 1, consider the Base Entailment - *"**Party B** won 89 seats"*. Suppose we replace one relevant entity and get "*~~Party B~~ Party C* won 89 seats". The resultant statement is still an entailment. To ensure that this does not happen, we compare the non-replaced entities ("89") with corresponding cells in the row of the potential candidate. We also create contradictions by replacing words in the Base Entailment with their antonyms. This is especially useful for cases of superlatives and comparatives. An example is shown in Table 1.

2.2 Perturbing the Table (Premise)

In this subsection, instead of making changes to the Base Entailment, we change the premise, i.e. the tables by swapping two or more table cells. To further improve model generalization, similar to (Kaushik et al., 2020; Gardner et al., 2020) we create example pairs which differ minimally but have opposite inference labels. These perturbed tables no longer represent the real world, hence we call them **Counterfactual**. Addition of counterfactual data makes the model more robust by inhibiting it to learn spurious patterns between label and hypothesis/premise. Similar Counterfactual data also ensure that model is not biased and preferably grounds on the primary evidence rather than relying on it over-fitted pre-trained knowledge blindly. Similar observation was also observed by Müller

et al. (2021) for TabFact dataset.

Creating Counterfactual Tables (CF) We consider a contradiction C1 formed by replacing the highlighted entity from table cell $[rowX, columnY]$ with potential candidate $[rowA, columnY]$ in the original table (as described in Section 2.3). To create a counterfactual table, we swap cells present at $[rowX, columnY]$ and $[rowA, columnY]$ such that C1 becomes an entailment to the modified table, and the original Base Entailment becomes a contradiction to it. We then create more hypotheses from this as shown in Table 1.

Hypothesis Paraphrasing (HP) Dagan et al. (2013) demonstrate that paraphrasing data improves lexical and structural diversity, thus boosting model performance on unstructured NLI. Following Dagan et al. (2013), we paraphrase our data, since the hypotheses we derive from Base Entailments have similar structures. We use the publicly available T5 Model (Raffel et al., 2020) trained on the Google PAWS dataset (Zhang et al., 2019) for generating paraphrases. We generate the top five paraphrases and randomly select from them.

3 The `ToTTo-TNLI` Dataset

Using the framework described in Section 2, we create the ToTTo-TNLI dataset.

Source dataset We use ToTTo (Parikh et al., 2020), a controlled table to text generation dataset with over 120,000 training examples as our source dataset. ToTTo proposes a controlled generation task: given a Wikipedia table and a set of highlighted table cells, produce a one-sentence description. The dataset construction process requires annotators to revise existing candidate sentences from Wikipedia instead of constructing them, ensuring a natural and versatile style of writing while also eliminating annotator bias. Since we create contradictions by perturbing entailments, and we create counterfactual data such that a given hypothesis can be both an entailment and a contradiction to different premises, we ensure that the word distributions and syntactic structures for both entailments and contradictions are similar. This prevents the model from learning spurious cues in hypothesis sentence for predicting entailment inference.

Table Pre-processing ToTTo provides tables and a list of highlighted cells as the premise for its generation task. To make the table data ready for our use, we perform the following three-step pre-processing on tables:

1. We conduct all experiments assuming the tables to be vertically aligned as mentioned in Section 2. We find that some tables are horizontally aligned (with the first column containing headers). We apply heuristics to automatically identify such tables and flip them as a pre-processing step.

2. We create a column-wise list of "potential candidates" for each table while pre-processing. For each column, we assign it a data type (date, alpha, alphanumeric, numeric or ordinals), which is the data type of the majority of its cells. We ensure that the potential candidates for each column fit its data type. *NULL* values and values like "TBA"/ "Undecided" are removed.

3. We label certain table rows to indicate that their values must not be replaced as they represent aggregate values like "Grand total" or "Average". We also label rows containing middle headers.

Hypothesis Pre-processing ToTTo (Parikh et al., 2020) provides a one-sentence description i.e. Base Entailment derived from the given highlighted cells i.e. relevant entities. We perform the following two-step pre-processing on the sentences:

US presidential inaugurations (A table Row)	
President #	44
Name	Barack Obama
Inauguration Date	January 20, 2009
Location	West Front, US Capitol
Base Entailment : **Obama's** inauguration as the **forty fourth** president took place at the **United States Capitol** in **2009**.	

Table 2: An example of cases requiring partial matching.

1. While ToTTo (Parikh et al., 2020) provides a list of relevant cells, it does not provide their alignments with the Base Entailment. We try to match every relevant cell with n-grams in the Base Entailment. We also handle cases of partial matching. Table 2 shows examples of names, ordinals, locations and dates that require partial matching.

2. We label the Base Entailment to indicate whether they are entailed from single or multiple rows. We also label sentences containing superlatives, comparatives and aggregate words like total, average, mean, etc. Table 1 shows an example of a Base Entailment containing a superlative.

Numbers (#)	Train	Test	Total
Entailments	2.5M	280K	2.9M
Contradictions	2.7M	301K	3M
Unique Tables	103K	6.7K	109K
Counterfactual Tables	284K	17K	301k

Table 3: Statistics for ToTTo-TNLI dataset

We use the method described in Section 2 to create NLI data from the ToTTo dataset. Since contradictions are easier to create, we limit their number such that the ratio of entailments: contradictions remains close to 1:1. We also limit the generation of counterfactual tables to three per original table. Table 3 shows the statistics for the resultant ToTTo-TNLI dataset.

4 Experiments and Analysis

In this section, we explore the significance of our data in various capacities. Overall, we aim to answer the following research questions:

1. **RQ1**: Can ToTTo-TNLI be used as a challenging test set for existing NLI models?
2. **RQ2**: Can ToTTo-TNLI be used as augmented data for existing Table NLI tasks like TabFact (Chen et al., 2020b)?
3. **RQ3**: How well can ToTTo-TNLI perform in a zero shot setting?

4.1 Experimental Setup

Model In all experiments, we start with the Table NLI model developed by Eisenschlos et al. (2020) as synthetic data augmentation baseline (referred to as TAPAS + Table-NLI model from here on). The model is based on TAPAS (Herzig et al., 2020), a table-based BERT model, and intermediately pre-trained on automatic rule based generated synthetic and counterfactual NLI data to recognize entailment. Following, Eisenschlos et al. 2020, we further pre-train the TAPAS + Table-NLI model on ToTTo-TNLI before fine-tuning on the downstream task.

Dataset We use TabFact (Chen et al., 2020b), a benchmark Table NLI dataset, as the end task to report results. TabFact is a binary classification task (with labels: Entail, Refute) on Wikipedia derived tables. We use the standard train and test splits in our experiments, and report the official accuracy metric. TabFact gives simple and complex tags to each example in its test set, referring to statements derived from single and multiple rows respectively. Complex statements encompass a range of aggregation functions applied over multiple rows of table data. We report and analyze our results on simple and complex test data separately.

4.2 Results and Analysis

The following sections describe the results of our experiments with respect to the research questions outlined above.

ToTTo-TNLI as Evaluation benchmark (RQ1) We randomly sample small subsets of our data, including counterfactual tables, to create three test sets. We test the publicly available TAPAS+TNLI model fine-tuned on TabFact (but not pre-trained on ToTTo-TNLI) on the randomly sampled test sets. We find that even though TabFact contains both simple and complex training data, the model gives a mean accuracy of 65.6% on our test set, more than 14 points behind its accuracy on the TabFact test set. It further drops to 58.8% when tested on counterfactual data.

Analysis: Our data proves to be challenging when used as evaluation benchmark. We attribute this to the quality of our data. Our data, unlike TabFact, is derived from freely written Wikipedia text, ensuring no repetitive patterns or styles of writing. Moreover, counterfactual test data contains several pairs of tables and hypotheses which differ minimally but have opposite labels. The NLI model must be extremely robust to spurious patterns in order to correctly classify such examples. The drop in accuracy from 65.6% (on normal test set) to 58.8% (on counterfactual test set) shows lack of robustness in the model. A natural extension to this would be to introduce counterfactual data into training, which is shown to improve generalization in models (Gardner et al., 2020; Kaushik et al., 2020).

ToTTo-TNLI as Augmented data for TabFact (RQ2) Since TabFact is a binary classification task with Entail and Refute labels, our data fits the setting. We pre-train the TAPAS + Table-NLI

Model	$\text{Test}_{\text{mean of 3}}$	$\text{Test}_{\text{counterfactual}}$
Base	64.9	58.1
Large	65.6	58.8

Table 4: Accuracies for base and large TAPAS-TNLI model trained on TabFact and tested on ToTTo-TNLI simple and counterfactual elavulation sets

model on ToTTo-TNLI data following Eisenschlos et al. 2020, and then fine-tune on TabFact. We first show results on the downstream task after pre-training on raw ToTTo-TNLI data. We show further improvements with paraphrasing, and addition of counterfactual tables (and corresponding counterfactual statements) as described in section 2. Our best model outperforms the TabFact baseline (Chen et al., 2020b) by 17 points and the TAPAS + Table-NLI model by 1.2 points (refer Table 6).

Analysis: Our data, although recasted from a non-NLI task, is able to boost model performance. Paraphrasing and addition of counterfactual data push the accuracies even further, showcasing the importance of diverse and robust data.

Zero shot performance of ToTTo-TNLI (RQ3) The TAPAS+TNLI model, once pre-trained on ToTTo-TNLI, is in principle already a complete table NLI model. Since we create a versatile and large scale dataset, we look at the zero-shot accuracy of the model on the TabFact test set before fine-tuning on TabFact. We find that the model gives >80% accuracy on the simple test set before fine-tuning. This is 1.4 percent ahead of the baseline in a *supervised* setting for simple test data. Our model also outperforms TAPAS-Row-Col-Rank (Dong and Smith, 2021), which is a model trained on synthetic NLI data, by 4 points in the zero-shot setting.

Model	$\text{Test}_{\text{simple}}$	$\text{Test}_{\text{full}}$
TabFact $\text{Baseline}_{\text{sup}}$	79.1	65.1
$\text{Tapas-row-col-rank}_{\text{w/o sup}}$	76.4	63.3
$\text{ToTTo-TNLI (OG)}_{\text{w/o sup}}$	79.8	64.9
$\text{+ CF + Paraphrase}_{\text{w/o sup}}$	80.5	65.3

Table 5: Zero-shot accuracies for TAPAS+TNLI model trained on ToTTo-TNLI and tested on TabFact simple and full test set. Baseline is taken from TabFact (Chen et al., 2020b). (Dong and Smith, 2021) gives the zero-shot accuracy of TAPAS-Row-Col-Rank on TabFact. $\text{Subscript}_{\text{sup}}$ indicates that model is supervised i.e. fine-tuned on TabFact. $\text{Subscript}_{\text{w/o sup}}$ indicates that model is not fine-tuned on TabFact i.e. it is tested in a zero-shot setting.

Analysis: These results indicates that ToTTo-TNLI data generalizes well for NLI tasks that do not require complex mathematical operations. One reason behind this could be the nature of the source dataset. In this paper, we use generation datasets as the source, which naturally focus more on construction and coherence than complex reasoning. Recasting other types of datasets (e.g. Q/A, Semantic Parsing) for the purpose of including complex reasoning are discussed in Section 5.

5 Discussion

Why did we choose ToTTo? We specifically chose ToTTo to be our source dataset for several reasons. First, the dataset uses open-domain Wikipedia tables, which gives a good generalization ability to the recasted dataset. It is also the most common source for tables for downstream tasks. Second, to create table descriptions, ToTTo picks sentences from freely-written Wikipedia text and allows annotators to only edit them instead of constructing them. This ensures that the data is not just human-like, but also naturally sourced, eliminating annotator bias such as skewed word distributions and repetitive patterns in writing. Third, the dataset is large (with over 120k data points) even before scaling, ensuring a good variety of tables and descriptions to begin with. Fourth, ToTTo provides highlighted cell information. ToTTo highlights cells which are both explicitly and implicitly relevant to the description, which is essential for recasting. Having implicitly relevant cells also shows that this data is fairly complex, and requires the model to learn pattern recognition beyond token-matching.

What are the limitations of using a generation dataset? Section 2 highlights the prerequisites for creating tabular NLI data. Table-to-text generation datasets fit right into it, making them a good data source for recasting. One limitation of using generation data is that the "generated text" will always entail the premise, never contradict, so contradiction data is is hard to source naturally. We also draw some observations from the experimental results. In the zero shot setting (RQ3 in subsection 4.2) where we train our model only on ToTTo-TNLI and test on TabFact, we observe that data recasted from ToTTo is not best suited for test examples requiring complex mathematical reasoning. One reason behind this is that ToTTo picks statements from freely written text on Wikipedia as

Model		$\text{Test}_{\text{full}}$	$\text{Test}_{\text{simple}}$	$\text{Test}_{\text{complex}}$	$\text{Test}_{\text{small}}$
Table-BERT-Horizontal	Chen et al. 2020b	65.1	79.1	58.2	68.1
Logical-Fact-Checker	Zhong et al. 2020	71.7	85.4	65.1	74.3
HeterTFV	Shi et al. 2020	72.3	85.9	65.7	74.2
Structure-AwareTransformer	Zhang et al. 2020	73.2	85.5	67.2	-
ProgVGAT	Yang et al. 2020	74.4	88.3	67.6	76.2
TAPAS-Row-Col-Rank	Dong and Smith 2021	76.0	89.0	69.8	-
TAPAS + CF + Synthetic	Eisenschlos et al. 2020	81.0	92.3	75.6	83.9
Ours - Large	ToTTo-TNLI (OG)	81.9	92.8	75.8	84.1
	+ CF + Paraphrase	**82.1**	**93.4**	**76.1**	**84.4**
Human	-	-	-	-	92.1

Table 6: Accuracies on TabFact, including the Human Performance. OG denotes Original data and CF denotes Counterfactual data. Baseline and human results are taken from (Chen et al., 2020b) and (Zhong et al., 2020).

table descriptions. Freely written descriptive text is less likely to contain mathematical reasoning as very few contexts would demand it. This is one of the limitations of selecting a freely-written natural data source when compared to task-specific annotations.

Why recasting over human-annotation or automatic templates? Manual human annotation and automatic data generation through templates/ context-free-grammar are the two extremes of data creation approaches. Each has its own advantages. Human annotation allows generation of high-quality data in terms of creativity, diversity and fluency. Fully automatic data generation approaches produce synthetic but highly scalable cost-effective data in a very short time. In this paper, we aim to leverage the advantages of both approaches while eliminating their disadvantages as much as possible. We want to create data in large volumes without compromising on its quality. This is where recasting steps in - it allows us to eliminate annotation cost and time by picking previously annotated data. Moreover, sourcing data from manually annotated datasets guarantees human-involvement, which in turn ensures a creative and linguistically diverse end product. In our recasting framework, we use automatic techniques for finding alignments, making perturbations to the hypotheses and creating counterfactual data. The "automatic" nature of this part of the framework affirms high scalability. While the creation of new hypotheses from given source data is an automatic process, we call our overall recasting framework *semi-automatic* to indicate the human-involvement in the source dataset.

Future Directions Based on the observations and discussions, we identify the future directions as follows. *(1) Recasting other non-NLI datasets :* We make our case for choosing ToTTo as our source dataset in this paper. However, other generation datasets such as LogicNLG (Chen et al., 2020a) and Logic2Text (Chen et al., 2020d) can also be recasted for tabular NLI using our proposed framework, with dataset-specific implementation details. Tasks such as Question answering or semantic parsing on tables may also prove to be useful sources for recasting. Question-Answering has been used previously to create NLI data for unstructured text (Demszky et al., 2018; Khot et al., 2018). Our framework can be extended to handle various kinds of source datasets. *(2) Creating complex data from SQL derived data :* Since reasoning over structured tabular data is similar to performing SQL queries over databases, datasets for semantic parsing can be used to create complex data. An advantage of having structured SQL queries aligned with the hypotheses is that we can easily determine what kind of reasoning/ combination of reasoning underlies each hypothesis. TaPex (Liu et al., 2021) learns to execute SQL queries as part of pre-training and is shown to improve model performances on downstream tasks like NLI and Question Answering.

6 Related Work

Inference on Structured and Semi Structured Data In recent times, inference tasks such as NLI, Question Answering and Text generation have been applied to structured data sources like tables. TabFact (Chen et al., 2020b) and InfoTabs (Gupta et al., 2020) explore inference as an entailment task. WikiTableQuestions (Pasupat and Liang, 2015), Wik-

iQAA (Abbas et al., 2016) and HybridQA (Chen et al., 2020c) perform question answering on tables. ToTTo (Parikh et al., 2020), Yoran et al. (2021) and LogicNLG (Chen et al., 2020a) explore logical text generation on tables. Most of these datasets derive tables from Wikipedia.

Early work on structured data modeling classify tables into structural categories and embed tabular data into a vector space (Ghasemi-Gol and Szekely, 2018; Trabelsi et al., 2019; Deng et al., 2019). Recent work like TAPAS (Herzig et al., 2020), TAPAS-Row-Col-Rank (Dong and Smith, 2021) TaBERT (Yin et al., 2020), TABBIE (Iida et al., 2021), Tables with SAT (Zhang et al., 2020), TabGCN (Pramanick and Bhattacharya, 2021) and RCI (Glass et al., 2021) use more sophisticated methods of encoding tabular data. TAPAS (Herzig et al., 2020) encodes row/column index and order via specialized embeddings and pre-trains a MASK-LM model on co-occurring Wikipedia text and tables.

Data Augmentation and Recasting Generating cheap and scalable data for the purpose of training and evaluation has given rise to the use of augmentation and recasting techniques. Previous work done on recasting data for NLI on unstructured data includes White et al. (2017) and Poliak et al. (2018a), which use semantic classification data as their source. Multee (Trivedi et al., 2019) and SciTail (Khot et al., 2018) recast Question Answering data for entailment tasks. Demszky et al. (2018) proposes a framework to recast QA data for NLI for unstructured text. For tabular text, Dong and Smith (2021) present an effort to re-use text generation data for evaluation.

Data augmentation in NLP refers to methods used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. Synthetic data generation for augmentation of training data for unstructured text is explored in Alberti et al. (2019); Lewis et al. (2019); Wu et al. (2016); Leonandya et al. (2019), and for Tabular NLI in specific is shown in Geva et al. (2020); Eisenschlos et al. (2020). Salvatore et al. (2019) and Dong and Smith (2021) generate synthetic data for evaluation purposes. Kaushik et al. (2020); Gardner et al. (2020) show that providing counterfactual data, especially "minimal pairs" of examples (examples that differ only slightly but have opposite labels) can help to improve generalization in models. Müller et al. (2021) demonstrate that adding counterfactual hypotheses enhances model performance on the TabFact dataset. Data augmentation for tables is also explored for computer vision tasks such as structure recognition. TabAug (Khan et al., 2021) uses deletion/replication operations on rows and columns as perturbations. Due to varying height/width of different cells, TabAug performs deletion/replication on entire rows or columns. Since we don't deal with the visual structure but the textual content of the tables, we are able to make perturbations on a cellular level with operations like swapping. Closer to our work, Sellam et al. (2020) use perturbations of Wikipedia sentences for intermediate pre-training for BLEURT(a metric for text generation) and Xiong et al. (2020) replace entities in Wikipedia by others with the same type for a MASK-LM model objective.

Table Pre-training Existing works explore pre-training through several tasks such as Mask Column Prediction in TaBERT (Yin et al., 2020), Multi-choice Cloze at the Cell Level in TUTA (Wang et al., 2021b), Structure Grounding (Deng et al., 2021) and SQL execution (Liu et al., 2021). Our work is closely related to Eisenschlos et al. (2020), which uses two pre-training tasks over Synthetic and Counterfactual data to drastically improve accuracies on downstream tasks. Pre-training data is either synthesized using templates (Eisenschlos et al., 2020), mined from co-occuring tables and NL sentence contexts (Yin et al., 2020; Herzig et al., 2020), or directly taken from human-annotated table-NLI datasets (Deng et al., 2021; Yu et al., 2021). In our study, we employ pre-training data that has been automatically scaled from existing non-NLI data.

7 Conclusion

In this paper we introduced a semi-automatic framework for recasting existing table-to-text generation data for tabular NLI. We make the case for choosing the recasting route due to its cost effectiveness, scalability and ability to retain human-like diversity in the resultant data. Finally, we leverage our framework to generate NLI data for existing table to text dataset namely ToTTo (Parikh et al., 2020). In addition, we demonstrated that our created dataset could be utilized as an evaluation set as well as for data augmentation to enhance performance on the Tabular NLI task on TabFact (Chen et al., 2020b).

Acknowledgement

We thank members of the Utah NLP group for their valuable insights and suggestions at various stages of the project; and reviewers their helpful comments. We thanks Chaitanya Agarwal for valuable feedback on paraphrasing. Additionally, we appreciate the inputs provided by Vivek Srikumar and Ellen Riloff. Vivek Gupta acknowledges support from Bloomberg's Data Science Ph.D. Fellowship.

References

Faheem Abbas, Muhammad Kamran Malik, Muhammad Umair Rashid, and Rizwan Zafar. 2016. Wikiqa—a question answering system on wikipedia using freebase, dbpedia and infobox. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 185–193. IEEE.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2021. Open question answering over tables and text. *Proceedings of ICLR 2021*.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. 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.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. TabFact : A Large-scale Dataset for Table-based Fact Verification. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. HybridQA: A dataset of multi-hop question answering over tabular and textual data. pages 1026–1036.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020d. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6:1–220.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.

Li Deng, Shuo Zhang, and Krisztian Balog. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. Structure-grounded pretraining for text-to-SQL. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.

Rui Dong and David Smith. 2021. Structural encoding and pre-training matter: Adapting BERT for table-based fact verification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2366–2375, Online. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Majid Ghasemi-Gol and Pedro A. Szekely. 2018. Tabvec: Table vectors for classification of web tables. *ArXiv*, abs/1802.06290.

Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1212–1224, Online. Association for Computational Linguistics.

Vivek Gupta, Riyaz A. Bhat, Atreya Ghosal, Manish Srivastava, Maneesh Singh, and Vivek Srikumar. 2021. Is my model using the right evidence? systematic probes for examining evidence-based tabular reasoning. *CoRR*, abs/2108.00578.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Umar Khan, Sohaib Zahid, Muhammad Asad Ali, Adnan Ul-Hasan, and Faisal Shafait. 2021. Tabaug: Data driven augmentation for enhanced table structure recognition. In *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II*, page 585–601, Berlin, Heidelberg. Springer-Verlag.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Rezka Leonandya, Dieuwke Hupkes, Elia Bruni, and Germán Kruszewski. 2019. The fast and the flexible: Training neural networks to learn to follow instructions from small data. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 223–234, Gothenburg, Sweden. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor.

Thomas Müller, Julian Eisenschlos, and Syrine Krichene. 2021. TAPAS at SemEval-2021 task 9: Reasoning over tables with intermediate pre-training. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 423–430, Online. Association for Computational Linguistics.

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.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. 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.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018a. Towards a unified natural language inference framework to evaluate sentence representations. *CoRR*, abs/1804.08207.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018b. Hypothesis only baselines in natural language inference. pages 180–191.

Aniket Pramanick and Indrajit Bhattacharya. 2021. Joint learning of representations for web-tables, entities and types using graph convolutional network. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1197–1206, Online. Association for Computational Linguistics.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Felipe Salvatore, Marcelo Finger, and Roberto Hirata Jr. 2019. A logical-based corpus for cross-lingual evaluation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 22–30, Hong Kong, China. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Qi Shi, Yu Zhang, Qingyu Yin, and Ting Liu. 2020. Learn to combine linguistic and symbolic information for table-based fact verification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5335–5346, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mohamed Trabelsi, Brian D. Davison, and Jeff Heflin. 2019. Improved table retrieval using multiple context embeddings for attributes. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1238–1244.

Harsh Trivedi, Heeyoung Kwon, Tushar Khot, Ashish Sabharwal, and Niranjan Balasubramanian. 2019. Repurposing entailment for multi-hop question answering tasks. 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 2948–2958, Minneapolis, Minnesota. Association for Computational Linguistics.

Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021a. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021b. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Changxing Wu, Xiaodong Shi, Yidong Chen, Yanzhou Huang, and Jinsong Su. 2016. Bilingually-constrained synthetic data for implicit discourse relation recognition. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language*

Processing, pages 2306–2312, Austin, Texas. Association for Computational Linguistics.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *ICLR*.

Xiaoyu Yang, Feng Nie, Yufei Feng, Quan Liu, Zhigang Chen, and Xiaodan Zhu. 2020. Program enhanced fact verification with verbalization and graph attention network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7810–7825, Online. 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.

Ori Yoran, Alon Talmor, and Jonathan Berant. 2021. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. *arXiv preprint arXiv:2107.07261. Version 1.*

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir R. Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. In *ICLR*.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1629, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. LogicalFactChecker: Leveraging logical operations for fact checking with graph module network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6065, Online. Association for Computational Linguistics.