Leveraging Data Recasting to Enhance Tabular Reasoning

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

Creating challenging tabular inference data is essential for learning complex reasoning. Prior work has mostly relied on two data generation strategies. The first is human annotation, which yields linguistically diverse data but is difficult to scale. The second category for creation is synthetic generation, which is scalable and cost effective but lacks inventiveness. In this research, we present a framework for semi-automatically recasting existing tabular data to make use of the benefits of both approaches. We utilize our framework to build tabular NLI instances from five datasets that were initially intended for tasks like table2text creation, tabular Q/A, and semantic parsing. We demonstrate that recasted data could be used as evaluation benchmarks as well as augmentation data to enhance performance on tabular NLI tasks. Furthermore, we investigate the effectiveness of models trained on recasted data in the zero-shot scenario, and analyse trends in performance across different recasted datasets types.

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 text as the premise.

While textual inference is commonly researched, structured data (e.g. tables, knowledge-graphs and databases) enables the addition of more complex types of reasoning, such as ranking, counting, and aggregation. Creating challenging large scale supervision data is vital for research in tabular reasoning. In recent years, many initiatives to include tabular semi-structured data as the premise have been introduced e.g. tabular inference (TNLI) datasets such as TabFact (Chen et al., 2020b), InfoTabS (Gupta et al., 2020) and shared tasks like SemEval 2021 Task 9 (Wang et al., 2021b) and FEVEROUS (Aly et al., 2021). Tabular data differs from unstructured text in the way that it can capture information and relationships in a succinct manner through underlying structure (Gupta et al., 2020).

Example Table

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

***Q**: How many seats did Party B win? ; **A**: 89*

recast

Entailment: Party B won 89 seats.
Contradiction: Party B won 120 seats.

Table 1: Example of Tabular Data Recasting

Despite fluent, diverse, and creative, these human-annotated datasets are limited in scale owing to the costly and time-consuming nature of annotations. Furthermore, Gururangan et al. (2018) and Geva et al. (2019) show that many human-annotated datasets for NLI contain annotation biases or artifacts. This allows NLI models to learn spurious patterns (Niven and Kao, 2019), which enables models to predict the right label for the wrong reasons, sometimes even with noisy, incorrect or incomplete input (Poliak et al., 2018b). Recently, Gupta et al. (2021) revealed that tabular inference datasets also suffer from comparable challenges. Furthermore, Geva et al. (2019); Parmar et al. (2022) show that annotators introduce their own bias during annotation. For example, Gupta et al. (2021) demonstrates that annotators only generate hypothesis sentences from keys having numerical values, implying that some keys are either over or underutilized.

On the other hand, automatic grammar-based strategies , despite their scalability, lack linguistic

*Equal Contribution †Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4483–4496
December 7-11, 2022 ©2022 Association for Computational Linguistics

diversity (both structural and lexical) and necessary inference complexity. These approaches create examples with only naive reasoning. Recently, (Geva et al., 2020; Eisenschlos et al., 2020), create context-free-grammar and templates to generate augmentation data for Tabular NLI. Additionally, the use of a large language generation model (e.g., Zhao et al., 2022; Lewis et al., 2020; Raffel et al., 2020) for data generation has been proposed as well (e.g., Zhao et al., 2022; Ouyang et al., 2022; Mishra et al., 2022). However, such generation systems lack factuality, leading in hallucinations, inadequate fact coverage, and token repetition problem.

Can we generate challenging supervision data that is as scalable as synthetic data and yet contains human-like fluency and linguistic diversity? In this work, we attempt to answer the above question through the lens of data recasting. Data recasting refers to transforming data intended for one task into data intended for another distinct task. Although data recasting has been around for a long time, for example, QA2D (Demszky et al., 2018) and SciTail (Khot et al., 2018) effectively recast question answering data for inference (NLI), no earlier study has applied it to semi-structured data.

Therefore, we propose a semi-automatic framework for tabular data recasting. Using our framework, we generate large-scale tabular NLI data by recasting existing datasets intended for non NLI tasks such as Table2Text generation (T2TG), Tabular Question Answering (TQA), and Semantic Parsing on Tables (SPT). This recasting strategy is a middle road technique that allows us to benefit from both synthetic and human-annotated data generation approaches. It allows us to minimise annotation time and expense while maintaining linguistic variance and creativity via human involvement from the original source dataset. Table 1 shows an example of tabular data recasting. Note that while the steps we describe for recasting in this work are automatic, we call the overall end-to-end framework **semi**-automatic to include the manual effort gone into creation of the source datasets.

Our recasted data can be used for both evaluation and augmentation purposes for tabular inference tasks. Models pre-trained on our data show an improvement of 17% from the TabFact baseline (Chen et al., 2020b) and 1.1% from Eisenschlos et al. (2020), a synthetic data augmentation baseline. Additionally, we report a zero-shot accuracy of 71.1% on the TabFact validation set, which is 5% percent higher than the supervised baseline accuracy reported by Chen et al. (2020b).

Our main contributions are the following:

1. We propose a semi-automatic framework to generate tabular NLI data from other non-NLI tasks such as T2TG, TQA, and SPT.
2. We build five large-scale, diversified, human-alike, and complex tabular NLI datasets sourced from datasets as shown in Table 4.
3. We present the usage of recasted data as TNLI model evaluation benchmarks. We demonstrate an improvement in zero-shot transfer performance on TabFact using recasted data. We demonstrate the efficacy of our generated data for data augmentation on TabFact.

The dataset and associated scripts are available at `https://recasting-to-nli.github.io`

2 Why Table Recasting?

Tables are structured data forms. Table cells define a clear boundary for a standalone independent piece of information. These defined table entries facilitate the task of drawing alignments between relevant table cells and given hypotheses. If both the premise and hypothesis were plain text, any n-gram in the premise could be aligned to any n-gram in the hypothesis. Finding alignments between the premise and hypothesis is crucial in the recasting process. This is because, in order to modify statements, we must understand which entities influence their truth value.

Moreover, in tables, entries of the same type (same part of speech type, named entity type, domain etc) are clubbed under a common column header. This allows us to easily identify a group of candidates which are interchangeable in a sentence without disrupting its coherence. Frequently, the column header also indicates the data type (e.g. Name, Organization, Year etc). This is incredibly beneficial when modifying source data by substituting entities.

3 Tabular Recasting Framework

In this section, we describe a general semi-automatic framework to recast tabular data for the task of Table NLI. By recasting, we imply converting data meant for one task into a format that satisfies the requirements of a different task.

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 cells swaps)

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

Base Entailment$_{OG}$	Party A won 120 out of 298 seats.	Party A won the most seats.
New 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$_{CT}$)*

Base Entailment$_{CF}$	Party B won 120 out of 298 seats.	Party B won the most seats.
New 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 2: 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. subscript$_{OG}$ represents the "Original" table and subscript$_{CF}$ represents the "Counterfactual" table. Note that Base Entailment$_{OG}$ contradicts Table$_{CF}$ and Base Entailment$_{CF}$ contradicts Table$_{OG}$. This pair will always exhibit this property, but there can be statements which entail (or contradict) both OG and CF tables.

3.1 Prerequisites

A Table NLI data instance consists of (a) a **table**, (b) some **entailments**, i.e. true claims based on the table, and (c) some **contradictions**, i.e. false statements based on the table. To be able to generate these, a table is the initial prerequisite. Since we utilise tabular data as our source, this need is met.

In addition to the table, we require at least one reference statement that validates the table. We utilize the structure of this reference statement (henceforth referred to as the **Base Entailment**) to generate further entailments and contradictions. Once we have the Base Entailment, contradictions can be formed fairly easily. Falsifying any part of the Base Entailment that is linked to the table creates contradictions.

To create an entailment, however, every portion of the perturbed statement must hold true for the entire statement to constitute an entailment. This means that all entities originating from the table (henceforth referred to as **relevant entities**) must be found in the Base Entailment. Then and only then can we know with certainty how perturbations affect the truth value of a given assertion.

Alignments between a table and a Base Entailment are not always apparent, as demonstrated in Table 2. In the example *"Party A won the **most** seats*, the alignment between *most* and the greatest number of seats must be determined. Although we can employ automatic matching techniques between the Base Entailment and the table to extract relevant entities, we cannot be certain of detecting **all** of them unless they are explicitly provided. Therefore, we must be able to extract the following from source datasets: (a) a **table** i.e. the Premise, (b) a **reference statement** i.e. the Base Entailment, (c) **relevant entities** and (d) their **alignments** with the reference statement.

Once the prerequisites are met, new NLI instances can be formed by perturbing existing data in two ways: (a) by perturbing the hypothesis and (b) by perturbing the table, i.e. the premise.

3.2 Perturbing the Hypothesis

We modify the hypothesis, i.e. the Base Entailment, by substituting relevant entities with other potential candidates. We presume the tables are vertically aligned, which means that the top row contains headers and each column contains entities of the same kind. A *potential candidate* for a relevant entity coming from table cell C_{XY} having coordinates [row X, column Y] can be any other non-null entity from the same column Y.

Creating Entailments (E) To create entailments, we replace *all* the relevant entities in the given Base Entailment with potential candidates. Two or more relevant entities coming from table cells in the same row, say $C_{\text{XA}}, C_{\text{XB}}$, must be substituted with potential candidates from column A and B respectively, such that their row coordinate is equivalent i.e. $C_{\text{XA}}, C_{\text{XB}} \rightarrow C_{\text{ZA}}, C_{\text{ZB}} \mid Z \neq X$ (refer Table 2). Entities originating from "aggregate rows" (such as the *Total* row in Table 2) or "headers" must be left intact.

Creating Contradictions (C) To create contradictions, we substitute **one or more** relevant entities from the Base Entailment with alternative candidates. We note that the ensuing statement may be an entailment by accident. In Table 2, consider the Base Entailment - *"**Party B** won 89 seats"*. Suppose we replace one key entity (Party B) with a potential candidate to arrive at "***~~Party B~~ Party C*** won 89 seats". The resultant statement remains an entailment. To prevent this from occurring, the non-replaced entities are compared. Assume $C_{\text{XA}}, C_{\text{XB}}$ represent the relevant entities in the Base Entailment. If we replace $C_{\text{XA}} \rightarrow C_{\text{ZA}}$ then we must guarantee that $C_{\text{XB}} \neq C_{\text{ZB}}$ to avoid unintentional entailments.

We also generate contradictions by substituting antonyms for words in the Base Entailment. This is particularly helpful for scenarios involving superlatives and comparatives (refer Table 2). We use NLTK Wordnet to find antonyms. We enforce equality of POS tags for word-antonym pairs. We create a dictionary of such pairs across datasets, count their frequency and broadly filter (manually) the word-antonym pairs for keeping the frequent and sensible ones. Majority of these are comparative/superlative word-pairs (higher-lower, most-least, best-worst).

3.3 Perturbing the Table (Premise)

In this subsection, instead of modifying the Base Entailment, we swap two or more table cells to modify the premise, i.e. the tables. Similar to Kaushik et al. (2020) and Gardner et al. (2020), we build example pairs with minimal differences but opposing inference labels in order to improve model generalisation. These modified tables no longer reflect the actual world. Hence, we refer to them as **Counterfactual**. The addition of counterfactual data increases the model's robustness by preventing it from learning spurious correlations between label and hypothesis/premise. Minimally varying counterfactual data also ensures that the model is not biased and preferably grounds on primary evidence, as opposed to depending blindly on its pre-trained knowledge. Similar findings were made by Müller et al. (2021) for TabFact.

Creating Counterfactual Tables (CF) We consider a contradiction C1 formed by replacing the relevant cell $C_{\text{XA}} \rightarrow C_{\text{ZA}}$ in the original table (as described in Table 2). To create a counterfactual table, we swap cells $C_{\text{XA}} \leftrightarrow C_{\text{ZA}}$ such that C1 becomes an entailment to the modified table, and the original Base Entailment becomes a contradiction to it. Based on this, we generate further hypotheses, as illustrated in Table 2.

Hypothesis Paraphrasing (HP) Dagan et al. (2013) demonstrates that data paraphrasing increases lexical and structural diversity, thus boosting model performance on unstructured NLI. In accordance with Dagan et al. (2013), we paraphrase our data because the hypotheses derived from Base Entailments have similar structures. For producing paraphrases, we employ the publicly available T5 Model (Raffel et al., 2020) trained on the Google PAWS dataset (Zhang et al., 2019). We produce the top five paraphrases and then select at random from among them.

4 Addressing Tabular Recasting Constraints

Dataset specific implementations pose several challenges. We address them in the following ways:

Table Orientation. As stated in Section 3, we conduct all experiments assuming the tables are vertically aligned. We observe several horizontally aligned tables (with the first column containing headers) in source datasets. As a preliminary processing step, we employ heuristics to automatically recognise such tables and flip them. For example, we check for frequently occurring header names in the first column or consistency in data types (numeric, alpha, etc.) across rows rather than columns.

Partial Matching. We observe that some datasets provide relevant cells, but do not provide their alignments with the Base Entailment. We attempt to match every relevant cell with n-grams in the Base Entailment. Of particular interest is the sample row shown in Table 3 that contains names, numbers, locations and dates that are not exact, but partial matches to n-grams in the Base Entailment. We handle such cases of partial matching.

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 3: An example of cases requiring partial matching.

Irreplaceable Entities. We observe that **not all** relevant entities are replaceable by potential candidates. Table 2 presents an example of a table with a **Total** row. Relevant entity 298 cannot be replaced while creating *New Entailment*$_{OG}$ because it is an aggregate entity that whose substitution will disrupt the truth value of the statement.

Similar observation is made while swapping table cells to create counterfactual tables. Suppose we swap the aggregate cell 298 with 120 . The resultant table would be logically flawed since the "Seats" column won't add up to its Total. To prevent this, aggregate rows and header cells are marked as non-replaceable entities.

5 Dataset Recasting

Using the framework outlined in Section 3, we recast the five datasets listed in Table 4. All datasets utilise open-domain Wikipedia tables, comparable to TabFact. In addition, these datasets and TabFact share reasoning kinds such as counting, minimum/-maximum, ranking, superlatives, comparatives, and uniqueness, among others. Table 5 summarises the statistics of recasted datasets. [1]

5.1 Table2Text Generation to Table NLI

Given a table and a set of highlighted cells, the Table2Text generation task is to create a description derived from the highlighted cells. We presume this description to be the *Base Entailment* given that it is true based on the table. The highlighted cells become the *relevant entities*. An example is shown in Table 2, where *Base Entailment* $_{OG}$ is a description generated from OG Table's highlighted cells .

[1] Some of these datasets contain examples that are shared, but because the derivation procedure for NLI data is unique for each task type, generated statements are also different and regarded as individual instances.

Source Dataset	**Task**
WikiTableQuestions (Pasupat and Liang, 2015a)	TQA
FeTaQA (Nan et al., 2022)	TQA
Squall (Shi et al., 2020b)	SPT
WikiSQL (Zhong et al., 2017)	SPT
ToTTo (Parikh et al., 2020)	T2TG

Table 4: Source datasets used for creating tabular NLI data

Dataset	**Entail**	**Contradict**	**Total**
QA-TNLI	32k	77k	109k
WikiSQL-TNLI	300k	385k	685k
Squall-TNLI	105k	93k	198k
ToTTo-TNLI	493k	357k	850k

Table 5: Statistics for various recasted datasets. QA-TNLI combines recasted data from both FeTaQA and WikiTableQuestions. Test splits are created by randomly sampling 10% samples from each dataset.

Recasting ToTTo (Parikh et al., 2020). ToTTo comprises of over 120k training examples derived from Wikipedia tables. Annotators edit freely written Wikipedia text to produce table descriptions. Annotators also mark relevant cells, but not their alignments with the Base Entailment. To link relevant cells with tokens in the Base Entailment, we apply partial matching techniques. If all relevant cells are successfully matched, we proceed to build new entailments. In either scenario, contradictions are generated using any relevant cells for which alignments can be found. Table 3 illustrates this with an example.

5.2 Table Question Answering To Table NLI

Given a table and a question, the Table Question Answering task is to generate a long form (sentence) or short form (one word/phrase) answer to the question. A long form answer is a *Base Entailment* in itself. Table 1 depicts an example of recasting QA data.

Recasting WikiTableQuestions (Pasupat and Liang, 2015a). WikiTableQuestions provides 22k questions over Wikipedia tables, with short-form answers. We use a T5 based pre-trained model developed by Chen et al. (2021a) to convert $\{Question, Answer\} \rightarrow Statement$ (refer Table 1). We presume this to be our Base Entailment. Unless it is an aggregate value, the short-form answer is likely to be an entity from the table. We search for matches between the answer and table cells as well as n-grams in the question. We create contradictions from any relevant entities found.

Recasting FeTaQA (Nan et al., 2022). FeTaQA provides 10k question-answer pairs on Wikipedia tables. These long form answers are Base Entailments in themselves. Since supporting cell information is provided as well, we create both entailments and contradictions wherever possible.

> **Example of FeTaQA Recasting**
>
> *Q: What was the total number of seats?*
>
> *Long answer: There were* 298 *seats in total.*
>
> ↓ **replace 298 with 89** ↓
>
> *Entailment: There were* 298 *seats in total.*
>
> *Contradiction: There were* 89 *seats in total.*

5.3 Semantic Parsing to Table NLI

Given a table and a question, the Semantic Parsing task is to generate the underlying logical/SQL form of the question. Since the datasets provide a logical query form, we execute this query to obtain a "short-form answer". We combine the question and short form answer as mentioned in 5.2 to get the *Base Entailment*. SQL queries are parsable, allowing for easy identification of column names and cell values. Owing to the fact that the reasoning depends on these entities, we infer these are the *relevant entities*.

Recasting WikiSQL (Zhong et al., 2017). WikiSQL provides 67.7k annotated [SQL query, textual question] pairs on Wikipedia tables. To augment this data, we parallelly replace values in an SQL query and its corresponding question. We execute the new query, and combine the answer with the perturbed question to create a new entailment.

> **Example of WikiSQL Recasting**
>
> *Q: Which party won* 120 *seats?*
>
> *SQL: Select party from T where seats =* 120
>
> *Executed answer: Party A*
>
> *Base Entailment: Party A won* 120 *seats.*
>
> ↓ **replace 120 with 89** ↓
>
> *Q': Which party won* 89 *seats?*
>
> *SQL': Select party from T where seats =* 89
>
> *Executed answer: [Party B, Party C]*
>
> *Entailment': Party B won* 89 *seats.*
>
> *Party C won* 89 *seats.*

Note that when executing a query, the answer can be a single entity or a list of multiple entities. If we have a list of entities satisfying the query, any of these entities can be used to create entailments, while none of these entities should be used to create contradictions (we find other potential candidates from the answer column). Consider the OG table given in Table 2.

Recasting Squall. Squall provides 11k [sql query, textual question] pairs with table metadata. We augment it similar to WikiSQL. Furthermore, table metadata enables us to identify column kinds and, in some circumstances, reduce SQL queries and questions to skeletons. These skeletons may subsequently be used to generate hypotheses on additional tables that meet the column type specifications of the skeleton in question. Consider the example from Table 2 with columns Party (text) and Seats (numeric).

> **Example of Squall Recasting**
>
> *Q: Which party has the maximum seats?*
>
> *SQL: select party from T where seats=max(seats)*
>
> ↓ **extract skeleton** ↓
>
> *Q': Which $C1_{text}$ has the maximum $C2_{num}$?*
>
> *SQL': select $C1_{text}$ from T where $C2_{num}$=max($C2_{num}$)*

This can now be used on another table, suppose one about countries and their populations to ask "Which country has the maximum population?".

5.4 Human Evaluation

We asked five annotators to annotate fifty samples from each dataset on two fronts:

- Inference label: Label each sample as entail, refute or neutral. Neutral samples can either be those which can't be derived from the table, or which don't make sense.
- Coherence score: Score each sample on a scale of 1 to 3 based on its semantic coherence and grammatical correctness, 1 being incoherent and 3 being coherent with minor or no grammatical issues. A score of 2 is given to statements who's meaning can be understood, but the structure or grammar is incorrect in more than one place.

We compare our generated label with the majority annotated inference label, and if no majority was reached, we consider the sample inconclusive. For Coherence score, we calculate the average of the five annotators.

Analysis. Results are summarized in Table 6. We observe high label match scores for our datasets, with QA-TNLI at 90%, Squall-TNLI at 87% and WikiSQL-TNLI at 84%. ToTTo-TNLI is slightly

Dataset	% Label Match	Coherence Score
QA-TNLI	90%	2.68
Squall-TNLI	87%	2.54
WikiSQL-TNLI	84%	2.55
ToTTo-TNLI	78%	2.46

Table 6: Results for human evaluation of our generated data. Please note that the verification labels are considered to be matched only if annotators have reached a majority **and** it matches our generated label.

behind at 78%, which is largely due to samples marked as "neutral" or samples where no majority was reached. We also observe a consistently above average coherence score, largely between 2.5 and 3. This implies that most of our data is logical, coherent, and grammatical. Since the sources of our data are human-written (Wikipedia text/human annotations), we expect our generated sentences to be fluent and semantically correct.

6 Experiments and Analysis

In this section, we examine the relevance of our recast data across various settings. Overall, we aim to answer the following research questions:

1. **RQ1**: How challenging is recast data as a TNLI benchmark?
2. **RQ2**: How effective are models trained on recasted data in a zero shot setting?
3. **RQ3**: How beneficial is recasted data for TNLI data augmentation?

6.1 Experimental Setup

In all experiments, we follow the pre-training pipeline similar to Eisenschlos et al. (2020). We start with TAPAS (Herzig et al., 2020), a table-based BERT model, and intermediately pre-train it on our recasted data. We then fine-tune the model on the downstream tabular NLI 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.

6.2 Results and Analysis

We describe the results of our experiments with respect to the research questions outlined above.

1. As Evaluation benchmarks (RQ1) We randomly sample small subsets from each dataset, including counterfactual tables, to create test sets. We evaluate the publicly available TAPAS-TNLI model (Eisenschlos et al., 2020) fine-tuned on TabFact on the randomly sampled test sets, as shown in Table 7. We find that even though TabFact contains both simple and complex training data, the model gives a best accuracy of 68.6%, more than 12 points behind its accuracy on the TabFact set.

Analysis. The TAPAS-TNLI model performs best on WikiSQL-TNLI data, showing either that WikiSQL is most comparable to TabFact (in terms of domain, reasoning, and writing) or that WikiSQL is relatively trivial to address. Squall-TNLI is the hardest, as expected, as Squall was designed specifically to include questions that execute complex SQL logic. QA-NLI and ToTTo-NLI lie in-between, showing that they have some similarities with TabFact, but also incorporate complementary reasoning instances.

2. Zero Shot Inference Performance (RQ2) Once we pre-trained our model on recasted TNLI data, it is in principle already a table NLI model. Since we create a versatile and large scale dataset, we look at the zero-shot accuracy of our models on the TabFact test set before fine-tuning, as shown in Table 8. Our best model gives 83.5% accuracy on the simple test set before fine-tuning. Its performance is 6.0% percent ahead of Table-BERT, a *supervised* baseline. Our best model also outperforms TAPAS-Row-Col-Rank (Dong and Smith, 2021), which is a model trained on synthetic NLI data, by 7% in the zero-shot setting.

Analysis. QA-TNLI achieves the best zero-shot performance of 71.1%. We speculate that joining two datasets (FeTaQA and WikiTableQuestions) helps the model learn a variety of linguistic structures and reasoning. This is closely followed by Combined-TNLI, a model trained on the mixture of all the datasets. We speculate that the model's training may have been negatively impacted by integrating too many distinct data kinds. Squall-NLI noticeably gives 62.6% accuracy on the complex test set, indicating its utility for learning complex reasoning. The zero-shot accuracy of TabFact trained models in Squall-NLI (i.e. Table Table 7) and that

Test Set	Model	
	Base	Large
QA-TNLI	56.1	58.0
WikiSQL-TNLI	66.8	68.6
Squall-TNLI	53.7	55.1
ToTTo-TNLI	64.9	65.6

Table 7: Accuracies for base and large TAPAS-TNLI model trained on TabFact and tested on recasted datasets

Model	TabFact		
	simple	complex	full(s+c)
Table-BERT$_{sup}$	79.1	58.2	65.1
LPA Ranking$_{sup}$	78.7	58.5	65.3
Tapas-RC-Rank	76.4	57.0	63.3
QA-TNLI	**83.5**	**64.9**	**71.1**
WikiSQL-TNLI	79.0	57.7	64.9
Squall-TNLI	82.0	62.6	69.1
ToTTo-TNLI	80.3	59.6	66.7
Combined-TNLI	83.0	62.9	69.7

Table 8: Zero-shot accuracies for models trained on recasted data and tested on TabFact simple, complex and full dev set. Table-BERT and LPA Ranking are supervised baselines taken from TabFact (Chen et al., 2020b). (Dong and Smith, 2021) gives the zero-shot accuracy of TAPAS-Row-Col-Rank on TabFact.

of Squall-NLI trained model on TabFact (55.1% vs 69.1%) clearly show that Squall-NLI is a superior dataset in terms of complexity of reasoning. ToTTo-TNLI performs fairly well on simple data (80.3%) but is not well equipped to handle complex examples. This is due to the "descriptive" nature of generation data, which includes limited inferential assertions.

3. Data Augmentation for TabFact (RQ3) Since TabFact is a binary classification task with Entail and Refute labels, our recasting data can also be used for data augmentation. We pre-train the model with our recasted data, similar to Eisenschlos et al. (2020) (refer 6.1), before final fine-tuning on the TabFact dataset. Table 9 shows the performance after data augmentation. Our best model outperforms the Table-BERT and LPA Ranking baselines (Chen et al., 2020b) by 17 points, and Eisenschlos et al. (2020) by 1.1 points.

Analysis. Following the zero-shot results (Table 8), QA-TNLI performs well as expected in the fine-tuned setting. We speculate that ToTTo-TNLI outperforms QA-TNLI due to their dataset size disparity (nearly 8x more, refer Table 5). The fact that WikiSQL-TNLI achieved the highest accuracy with TabFact-trained models (Table 7) and the lowest zero-shot accuracy (Table 8) on TabFact indicates that the data is relatively non-complex. Squall-TNLI does not improve model performance after augmention despite its remarkable zero-shot performance (Table 7). We suspect that this is because the domains and types of underlying logic (a.k.a. reasoning types) are quite distinct. We also combine all datasets (in equal rates) to train a composite TNLI model. Its accuracies are not at par with our best model. There can be several reasons behind this, one being that our mixing strategy isn't optimal. We could, for example, train for one dataset at a time and then slowly go on to the next, instead of mixing all datasets at each stage in equal proportions. This can be further investigated in the future. Another possibility is that the datasets include distinct types of data, such that merging them all has a detrimental effect.

7 Related Work

Inference on 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, 2015b), WikiQAA (Abbas et al., 2016), FinQA (Chen et al., 2021b) and HybridQA (Chen et al., 2020c) perform question answering on tables. ToTTo (Parikh et al., 2020), Yoran et al. (2021), LogicNLG (Chen et al., 2020a) and Logic2Text (Chen et al., 2020d) 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. Yang and Zhu (2021) decomposes NLI statements into subproblems to enhance inference on TabFact.

Data Augmentation. Generating cheap and scalable data for the purpose of training and evaluation has given rise to the use of augmentation

Model	Dev	Test$_{\text{full}}$	Test$_{\text{simple}}$	Test$_{\text{complex}}$	Test$_{\text{small}}$
Table-BERT-Horizontal(Chen et al., 2020b)	66.1	65.1	79.1	58.2	68.1
LPA-Ranking (Chen et al., 2020b)	65.1	65.3	78.7	58.5	68.9
Logical-Fact-Checker (Zhong et al., 2020)	71.8	71.7	85.4	65.1	74.3
HeterTFV (Shi et al., 2020a)	72.5	72.3	85.9	65.7	74.2
Structure-Aware TF (Zhang et al., 2020)	73.3	73.2	85.5	67.2	-
ProgVGAT (Yang et al., 2020)	74.9	74.4	88.3	67.6	76.2
TableFormer (Yang et al., 2022)	82.0	81.6	93.3	75.9	84.6
TAPAS+Salience (Wang et al., 2021a)	82.7	82.1	93.3	76.7	84.3
TAPAS + CF + Syn (Eisenschlos et al., 2020)	81.0	81.0	92.3	75.6	83.9
QA-TNLI (Question Answering)	81.4	**81.8**	92.6	**76.4**	84.0
WikiSQL-TNLI (Semantic Parsing)	78.3	78.6	91.2	72.4	80.9
Squall-TNLI (Semantic Parsing)	80.6	80.5	91.9	74.9	82.3
ToTTo-TNLI (Table2Text Generation)	**81.9**	**82.1**	**93.7**	**76.4**	**85.4**
Combined-TNLI	81.0	80.5	92.0	74.8	83.7
Human -	-	-	-	-	92.1

Table 9: Accuracies on TabFact, including the Human Performance. Table-BERT-Horizontal and LPA-Ranking (w/ discriminator) are baselines taken from TabFact (Chen et al., 2020b). CF means CounterFactual data, TF means TansFormers, LPA means Latent Program Algorithm. ToTTo-TNLI, QA-TNLI (WikiTQ + FeTaQA), WikiSQL - TNLI and Squall - TNLI are table NLI models pre-trained on CF + Synthetic data (Eisenschlos et al., 2020) followed by respective re-casted datasets. Combined - TNLI is a model trained on all of the data, starting with CF + Synthetic data and then mixing data from recasted datasets in equal rates.

techniques. Synthetic data generation for augmentation 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 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. 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.

Data Recasting. Data generation through recasting has been previously explored for NLI on unstructured data. White et al. (2017) uses 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. Poliak et al. (2018a) presents a collection of recasted datasets originating from seven distinct tasks. For tabular text, Dong and Smith (2021) present an effort to re-use text generation data for evaluation.

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., 2021c), 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.

8 Conclusion

In this paper we introduced a semi-automatic framework for recasting tabular data. We made our case for choosing the recasting route due to its cost effectiveness, scalability and ability to retain human-alike diversity in the resultant data. Finally, we leveraged our framework to generate NLI data for five existing tabular datasets. In addition, we demonstrated that our recasted datasets could be utilized as evaluation benchmarks as well as for data augmentation to enhance performance on the Tabular NLI task on TabFact (Chen et al., 2020b).

9 Limitations

While our work on tabular data recasting produces intriguing outcomes, we observe the following limitations in our approach:

1. Source datasets are designed for tasks different than the target. While our methodology

assures that recasted data retains the strengths and positive qualities of its original source, we have observed that some of these traits may not necessarily coincide with the targeted task. For instance, generation tasks provide “descriptions”, therefore the annotated data is *descriptive* in nature, but it is unlikely to contain complicated reasoning involving common sense and table-specific knowledge. In addition, any faults in the original data (e.g. bias issue) may get transferred to the recasted version.

2. Although the domains of source and target tasks can be comparable (in our example, open-domain Wikipedia tables), their distributions of categories, themes, and so on are likely to vary. When we train models using recasted augmentation data, we unintentionally introduce a domain transfer challenge. As a result, the final model’s performance is influenced to some extent by domain alignment.

3. Tables are semi-structured data representations that differ not just in domains and writing style, but also in structure. For example, InfoTabS (Neeraja et al., 2021) is a collection of Infoboxes, which are tables that describe a single entity (person, organisation, location). These are very different from the database-style tables that we use in our research. Tables can also be chronological, nested, or segmented which makes them more challenging. While we can employ our current heuristics to identify such tables, our current recasting strategy is prone to failure with tables that do not have database-like structures.

4. Annotated data sometimes relies on common sense and implicit knowledge that is not explicitly mentioned in the premise. Such data instances might be difficult to interpret automatically, making them challenging to recast. For example, in Table 10, to compare "Gold" with "Silver", the association of "Silver medal" with 2_{nd} place and "Gold medal" with 1_{st} place must be known. This implicit common-sense like knowledge makes this example hard to recast.

5. Our work on data recasting is done only on English language data. However, our proposed framework is easily extensible to other languages, high resource and low resource alike. Since we depend on identifying and aligning entities (between premise and hypothesis), morphologically analytic languages are easier to work with. Highly agglutinative languages may require additional efforts such as morph-analysis.

Micheal Phelps - 100m Butterfly

Vanue	Year	Medal
Olympics, Beijing	2008	Gold
Olympics, London	2012	Gold
Olympics, Rio	2016	Silver

Label: *Entailment:*
H: Micheal Phelps ranked better in 2012 than in 2016 for the 100m Butterfly event.

Table 10: An example table and an entailment derived from the same.

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. 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*.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021a. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

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, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, 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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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.

Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2022. Don't blame the annotator: Bias already starts in the annotation instructions. *arXiv preprint arXiv:2205.00415*.

Panupong Pasupat and Percy Liang. 2015a. 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.

Panupong Pasupat and Percy Liang. 2015b. 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. 2020a. 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.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020b. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for 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.

Fei Wang, Kexuan Sun, Jay Pujara, Pedro Szekely, and Muhao Chen. 2021a. Table-based fact verification with salience-aware learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4025–4036, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021b. 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. 2021c. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and; 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*.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

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.

Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring decomposition for table-based fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1045–1052, Punta Cana, Dominican Republic. 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 *International Conference of Learning Representation*.

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.

Mengjie Zhao, Fei Mi, Yasheng Wang, Minglei Li, Xin Jiang, Qun Liu, and Hinrich Schuetze. 2022. LM-Turk: Few-shot learners as crowdsourcing workers in a language-model-as-a-service framework. pages 675–692.

Victor Zhong, Caiming Xiong, and R. Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *ArXiv*, abs/1709.00103.

Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. Logical-FactChecker: 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.